Modelling complexity and redundancy in endocytic actin polymerisation

Lewis Peter Hancock

BSc Biochemistry

A thesis submitted for the degree of Doctor of Philosophy

University of Sheffield

School of Biosciences

September 2023

I would like to dedicate this thesis to my Great, great Uncle Bill (1924-2020). Having grown up in a working-class mining village, he cycled a 40-mile round trip to study at the University of Sheffield whilst spending his evenings helping the war effort. Never did he dwell on the challenges he faced, only his passion for learning. It came as quite a shock to him that the man he chatted with over tea, Hans Krebs, was actually a rather influential figure. He was the embodiment of dedication and serves as an inspiration to my life.

Abstract

Actin is one of the most ubiquitous proteins of life and can form filaments which play crucial roles in a wide range of processes from cell division to intracellular trafficking. Formation of these filament networks is tightly controlled using a wide array of protein types, chief among them being nucleators. Nucleators facilitate the unfavourable first steps of filament formation and thus their regulation dictates when and where filamentous networks are produced. The central proline-rich region of Las17 (yeast homologue of human WASp) is thought to nucleate actin "mother filaments" at the endocytic sites. Arp2/3 – a potent nucleator activated by Las17 – can branch these mother filaments. Proline also constitutes the core binding region of SH3 domains which leaves the nucleating region of Las17 open to competitive regulation. Eleven Las17-binding SH3 domains are recruited to yeast endocytic sites. Five of these bind via a tandem of domains (three SH3s in Sla1 and two SH3s in Bzz1). We hypothesise that this "cloud" of SH3 domains can regulate the access of actin to the proline-rich region of Las17. However, the high number of proteins and interactions involved renders a purely experimental approach challenging.

Throughout this thesis, two agent-based models are built (one being a progression of the other) to test the veracity of our regulatory cloud hypothesis. Binding affinities were experimentally obtained to build the model, demonstrate the power of avidity conferred through tandem SH3 binding, and refine our Las17 nucleating mechanism. We identify that the weak interactions of the SH3 cloud can combine in effect – particularly complemented by the tandem binding of Sla1 and Bzz1 – to define a window of Las17 nucleating activity. This work suggests how endocytic SH3 domains can regulate endocytic progression whilst also furthering our understanding of the relatively unexplored nucleating mechanisms employed by Las17.

Table of Contents

Acknowledgements

This candidate confirms the work submitted is their own. Science is always a collaborative endeavour, and I certainly wouldn't be submitting a thesis without the amazing support of my supervisors Mike Williamson and Kathryn Ayscough. They provided me with the warmest and most encouraging environment I could ever ask for. None of this would be possible without them and the generosity of my funder White Rose BBSRC.

I would like to thank Ellen Allwood for making me feel at home in the lab and picking me up when I was otherwise too stressed to move forward. Whenever progress on my programming stalled, Nick Fowler always jumped in to help and calm my mind over an evening game of Warhammer – thanks Nick! And thank you to John Palmer and Rosie Staniforth for always encouraging me even when I felt out of my depth.

Even outside the university, the support held strong. Thanks to Jeremy Craven for offering advice and an outlet for my many queries. The time spent at Certara was one of my proudest moments so thank you for the opportunity you gave me. On the topic of people who have always been there for me, I will never forget Christine Breaky for her support and dedication.

And thank you to my family and friends too. I am so lucky to have such warm people in my life. Mum, thank you for providing me a loving home that afforded me the opportunities to pursue an academic dream. Dad, your trip through to Sheffield when I was upset was one of the things that kept me going. My siblings, Matthew and Emily, the joy you gave me on my visits home always helped keep my spirits high. And finally, my friends and lab colleagues for keeping a smile on my face and Amy for dealing with my long nights absorbed by the lab.

Abbreviations

CME Clathrin-Mediated Endocytosis
ATP/ADP Adenosine Triphosphate/Adenosine Diphosphate
G-actin Globular actin
F-actin Filamentous actin
M2 Myosin II
ABP actin-binding protein
WASp Wiskott-Aldrich syndrome protein
Hsp70 Heat Shock Protein 70
WH1/WH2 Wiskott–Aldrich syndrome protein homology domain ½
NPF Nucleation Promoting Factor
WDS WISH/DIP/SPIN90 family proteins
FH1/2 Formin Homology ½
TMBN Tandem-Monomer-Binding Nucleator
JMY Junction Mediating and Regulatory Protein
Cobl Cordon-bleu WH2 Repeat Protein
ABS1/2 Actin Binding Site ½
BAR Bin/Amphiphysin/Rvs
WCA/CA WH2 Central Acidic/Central Acidic
PP Polyproline
IDP Intrinsicaly Disordered Protein
YAB Ysc84 actin binding domain
K_{d} Dissociation Constant
k_{on} Association Rate Constant
k_{off} Dissociation Rate Constant
MD Molecular Dynamic
ABM Agent Based Modelling
ODE Ordinary Differential Equation
FLAME Flexible Large-scale Agent Modelling Environment
GPU Graphical Processing Unit
CPU Central Processing Unit
PBS Phosphate Buffered Saline
KME KCl MOPS EGTA
SDS Sodium Dodecyl-sulfate
DTT Dithiothreitol
PCR Polymerase chain reaction
MST Microscale Thermophoresis
BLI Bio-Layer Interferometry
GST Glutathione-S-Transferase
Ni-NTA Nickel-nitrilotriacetic Acid
NMR Nuclear Magnetic Resonance
pI Isoelectric Point
SEC Size Exclusion Chromatography
TIRF Total internal reflection fluorescence
BSA Bovine Serum Albumin

1. Introduction

It is almost impossible to find a biology textbook that does not mention actin. This protein is one of the most abundant in eukaryotic cells and can provide a wide range of functions from driving cell motility to providing an intracellular highway for transport (Pollard and Borisy, 2003; Colonne, Winchell and Voth, 2016). A key part of their functionality is the ability for actin monomers to interact with one other and form filaments (nucleation) which can be elongated (polymerisation). Actin remains one of the most highly conserved proteins despite its three-billion-year history with human and budding yeast actin orthologs sharing a ~90% sequence similarity. This similarity even allows co-polymerisation between parasitic protozoan and mammalian actin which are separated by over a billion years of evolution (Kotila *et al.*, 2022).

The important conservation of this protein family was expanded with the discovery of orthologs in Archaea and Bacteria. Both Archaeal actin (Crenactin) and Bacterial actin (e.g., MreB and MreC) have been shown to regulate cellular structure and division through the formation of filamentous structures (Divakaruni *et al.*, 2007; Bernander, Lind and Ettema, 2011). This function is consistent with eukaryotic actin and demonstrates the important role actin plays throughout all three domains of life.

1.1 Cytoskeleton

Eukaryotic and many prokaryotic cells employ their actin orthologs to maintain a large, internal network of actin filaments which makes up part of the cytoskeleton. The cytoskeleton – essentially the cellular skeleton – was first theorised over a hundred years ago and can be constituted from a variety of proteins including actin, intermediate filament (including keratin and vimentin), and microtubules. The network of actin filaments which constitutes the cytoskeleton is termed the "actin cytoskeleton" and is now known to be an effective method for maintaining cell shape, compartmentalising functions, organising the location of proteins, and providing mechanical forces for cellular process such as cytokinesis and endocytosis (Pollard and Goldman, 2018).

This conserved function makes the actin cytoskeleton a frequent target for pathogenic species. For example, the protozoan *L. donovani* has been shown to promote excessive actin polymerisation in phagosomes through their cell-surface lipophosphoglycans to prevent the application of lysosomal markers to the protozoa. Conversely, the toxoplasma *T. gondii* can reduce host actin polymerisation by releasing a G-actin binding, and sequestering, toxin named Toxofilin to evade immune capture (Poupel *et al.*, 2000; Holm A *et al.*, 2001).

The cytoskeleton can form myriad different networks to serve different purposes and there exists functional overlap between the different cytoskeletal proteins. For example, intracellular transport via molecular motors occurs along both microtubule and actin networks. This transport employs different proteins to select individual filament types with kinesin and dynein moving along microtubules and myosin moving along actin. However, despite clear overlaps in many cellular functions, actin remains the dominant cytoskeletal component proving the mechanical force during Clathrin-Mediated Endocytosis (CME) (Pollard and Goldman, 2018). The formation of this force-driving network is the subject of this thesis and consequently, the following overview of the cytoskeleton will primarily focus on the role of actin.

1.1.1 Cytoskeleton: Filament processivity

As previously mentioned, both actin and microtubules play a major role in the intracellular transport of internal vesicles. This transport is executed by vesicle-bound motor proteins which turn over cellular energy stores to generate movement along actin and/or microtubule filaments. In the case of the actin-binding myosins, this directional movement is achieved by hydrolysing ATP to shift the location of their binding head domain(s). Most myosins possess two heads that essentially “walk” along the filament, although some myosins possess only a single head (Mermall, Post and Mooseker, 1998).

This processive activity can also be employed by other mechanisms such as in human muscle contraction. Here, myosin II (M2) form bundles which run along the length of the cell. The myosin head domains face outwards into the cytosol and can interact with the parallel actin filaments. M2 hydrolyses ATP to move along bundled actin filaments (termed thin filaments) which generates a movement force against the actin cytoskeleton. This is sufficient to move the filaments in opposite, relative directions to cause a muscle contraction (Squire, 2019).

Contractile forces between actin filaments and myosin II also play a role in cytokinesis by helping to separate the cell following the formation of two distinct nuclei. Actin and M2 filaments form a ring-like structure around the cortex of the cell (termed the contractile ring) in-between the two daughter nuclei. Actin filaments within the contractile ring can interact with the N-terminal “curly” region of membrane-associated IQGAP proteins to increase filament bending, marking a potentially important link to membrane geometry (Palani *et al.*, 2021). The myosin heads then process along the actin filaments to contract the region reminiscent of a muscle contraction. The resultant membrane deformation is termed a cleavage furrow. This furrow deepens and the cell eventually splits in two (Pollard and O’Shaughnessy, 2019). Interestingly, *Dictyostelium* and some mammalian cells can form these furrows even in the absence of M2 due to an alternative pathway driven by SCAR/WAVE (King *et al.*, 2010).

1.1.2 Cytoskeleton: Structural cortical networks

Actin not only serves as a tract for myosin movement, but can also provide structural support for organisms. Eukaryotic cells are compartmentalised from their environment using a lipid bilayer. These bilayers form spontaneously and rely on the cytoskeleton to maintain their shape against external forces such as fluid dynamics (Watson, 2015; Pegoraro, Janmey and Weitz, 2017). Actin filaments are anchored to the plasma membrane to form stress fibres which provide a structural framework to help maintain cell shape. Microtubules, the most rigid cytoskeletal polymer, extend outwards from the centre of the cell to provide further reinforcement of cell shape and structural integrity (Pollard and Goldman, 2018).

Cells must frequently adapt the shape of their membrane to react to external stimuli and this requires the cytoskeleton to possess both viscoelastic and dynamic properties. Structural elasticity is crucial for rapid recovery following membrane deformation and is aided by the filaments' simultaneous properties of high flexibility and propensity to adopt a linear arrangement with a high persistence length (the length required for the correlation between initial tangent and direction to be lost). Actin filaments can even recruit stabilising actin-binding factors to further increase their persistence length such as cross-linking proteins, bundling proteins, and actin-binding proteins (ABPs) which reinforce the inherent stability of filaments (Isambert *et al.*, 1995). These features are critical in resisting the mechanical forces applied upon cells without succumbing to shearing of the supporting cytoskeleton (Pegoraro, Janmey and Weitz, 2017).

1.1.3 Cytoskeleton: Localised force-driving networks

Relatively short-lived networks (often located cortically) can be rapidly developed to provide a directional mechanical force. Key examples of this are found in cell movement and endocytosis (Pollard and Goldman, 2018). As explained in the previous section, a network of cortical actin is usually maintained across the entire cell to reduce external mechanical forces impacting cell shape and integrity. However, this region of the cytoskeleton can also be rapidly expanded to actively direct cellular movement.

When cells move, the frontal region is termed the leading-edge. Lamellipodia are membranous projections which extend from the leading edge of the cell to facilitate cell movement and mechanosensing. A dynamic network of filamentous actin serves as the major cytoskeletal component of the lamellipodium and is the principal driving force. In human cells, development of the leading edge begins with a membrane signal which leads to the recruitment of proteins including WASp and the Arp2/3 complex to increase the concentration of filamentous actin by branching pre-existing filaments. These branches are depolymerised once the network extends into the cell (after around 1 µm) to provide more actin monomers for the leading-edge and direct the force-generating network to a narrow point on the membrane(Pollard and Borisy, 2003).

This bears a striking resemblance to the mechanism of pulling the membrane inwards during endocytosis. Here, internalisation signals also facilitate the recruitment of proteins including WASp and the Arp2/3 complex to catalyse the expansion of a branched, actin cytoskeletal network (Carlsson, 2018). Mechanisms such as transport can employ relatively long stretches of unbranched actin. However, when employed for the purpose of generating force, cells heavily favour either a branched network or filament bundles. This is because as actin monomers are recruited to filament ends contacting the membrane, the filament bends to make room for the addition. The resultant release of thermal energy as the filament returns to a straight conformation is applied to the membrane. This rachet mechanism "pushes" against the membrane. It has been theorised that shorter filaments are more efficient in applying this particular force and thus networks of many short, branched filaments are one way of increasing force (Pollard and Borisy, 2003). On the other hand, networks can also bindle Factin to increase filament rigidity and thereby force transmission (Chandrasekaran, Upadhyaya and Papoian, 2019).

1.1.4 The multifunctional protein

The actin cytoskeleton can form a range of functionally distinct networks owing to its uniquely combined properties of rapid polymerisation, interactions with many proteins, structural flexibility, and long persistence length. Many actin-binding proteins are likewise represented across multiple networks such as Myosin V which can transport vesicles along actin filaments, interact with actin to knit plasma membranes together and facilitate vesicle scission during endocytosis (Langford, 2002; Barker *et al.*, 2007). Another example is the WASp family proteins which help drive branched actin networks through activation of Arp2/3 (Pollard and

Borisy, 2003). WASp can relocate to from endocytic sites to pseudopodia following knockout of SCAR/WAVE in *Dictyostelium* to assume the functional dynamics of the latter, including activation of Arp2/3 (Veltman *et al.*, 2012). A sharing of proteins and mechanical phenomena between actin networks illustrates the high degree of interconnectivity between different actin-related processes. The large range of mechanisms which utilise actin, from cell division to endocytosis, reveal how the unique properties of the actin family are critical for organisms and thus explains why this protein is so highly conserved (Pollard and O'Shaughnessy, 2019). This all highlights the prevailing theme of regulatory complexity and redundancy. Study of a specific actin network may shed light upon other cytoskeletal processes which utilise the same functional or regulatory elements.

1.2 Actin

Actin monomers (globular or G-actin) are just over 40 kilodaltons in size and fold to form a globular structure resembling a cuboid with dimensions of 55 Å × 55 Å × 35 Å. Resolved structures look almost identical despite often being bound by completely different proteins to prevent polymerisation, highlighting its high stability (Dominguez and Holmes, 2011).

A simple glance at the structure of actin reveals two clear domains (domains 1 and 2). These are separated by a deep cleft and are connected by only two passes of the polypeptide chain. One of these interdomain connections adopts a helical structure and serves as the point of rotation during conformational change. Domains 1 and 2 can each be further subdivided into subdomains (1 and 2, and 3 and 4 respectively; **figure 1.1**) (Dominguez and Holmes, 2011).

The deep cleft is occupied by a nucleotide and Mg^{2+} cation which contact both domains and reinforce their positioning. Interdomain contacts made by these factors are required for correct actin folding (Stuart, Leatherbarrow and Willison, 2011). Hydrolysis of bound ATP to ADP can induce conformational change through the interdomain axis. However, the ATPase activity of G-actin is extremely weak and it generally adopts an ATP-bound conformation. This cleft is also lined with hydrophobic residues and is the primary interaction surface for a range of actin binding proteins including Spire and profilin (Dominguez and Holmes, 2011).

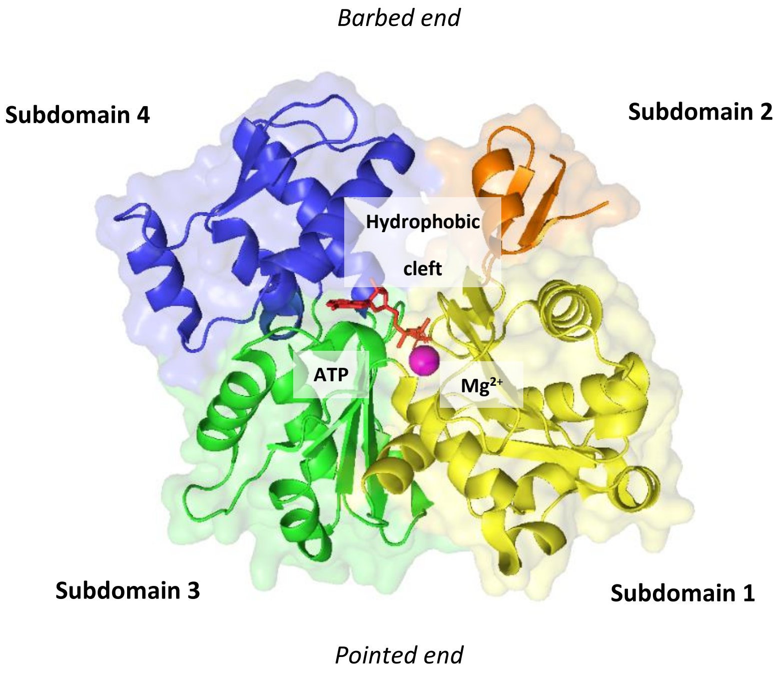


Figure 1.1: The structure of G-actin. Actin monomers contain four subdomains which are each illustrated by a separate colour (PDB 3HBT). Subdomains 1 and 2 can be grouped together to define domain 1 whilst subdomains 3 and 4 are grouped to define domain 2. The magenta ball represents the Mg2+ cation while ATP is shown in red. The hydrophobic cleft exists in between subdomains 4 and 2 and is often the target for actin-binding proteins. This figure was produced in PyMOL v4.6 using the PDB file 3HBT.

1.2.1 Actin paralogues

Throughout evolution, genes are often duplicated which can give rise to paralogues. Numerous proteins belong to the same structural superfamily as actin including Hsp70s, hexokinases, and sugar kinases (Bork, Sander and Valencia, 1992). The structure of Hsp70, a more relevant paralogue for this thesis, possesses a nucleotide binding domain which bears a striking resemblance to actin. This is then connected to a base and lid domain via a short linker. The Hsp70 family of proteins primarily functions to bind exposed hydrophobic regions at high affinity to prevent the aggregation of unfolded proteins and to enhance folding. Hydrolysis of ATP causes the base and lid domains to clamp together which greatly enhances binding affinity (Ambrose and Chapman, 2021). When working with actin binding proteins, possessing a knowledge of structural paralogues can help to identify copurified proteins.

1.2.2 Actin filament structure

Actin proteins can interact both laterally and longitudinally to form linear actin filaments. These filaments are constituted from two parallel strands which twist around one another to form a helical structure (**figure 1.2**). Lateral actin interactions occur between the sub filaments and involve the sides of subdomains 1, 3 and 4. Longitudinal contacts within the sub filament contribute the strongest actin-actin interaction via the hydrophobic residues in the deep cleft. The end of the filament where these clefts are exposed is referred to as the "barbed end" and is associated with a more rapid recruitment of monomers than the opposite pole of the filament, referred to as the "pointed end" (Dominguez and Holmes, 2011).

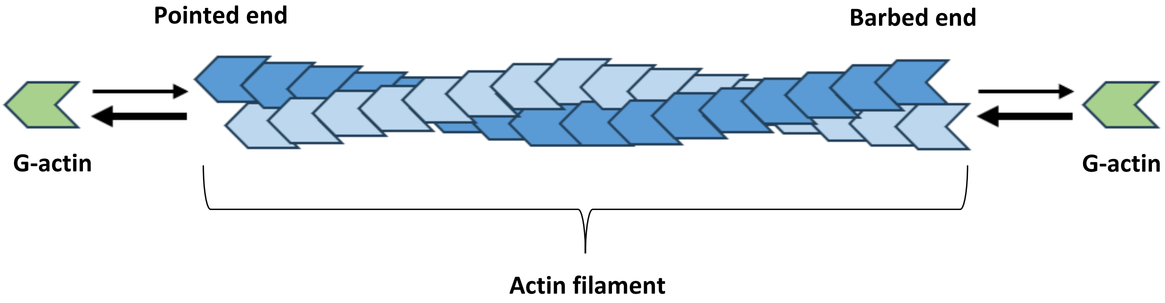


Figure 1.2: Cartoon representation of an actin filament. Actin subunits/monomers are shown as arrow heads with the point representing the pointed end of the protein and the indent representing the barbed end (hydrophobic cleft). The two sub filaments are shown using different shades of blue to illustrate how they intertwine to give actin filaments their helical shape. Actin association/loss is given by the arrows with the relative magnitude of the rate corresponding to the size. As shown by the figure, the barbed end of a filament has a higher ratio of association/loss than the pointed end contributing to a faster barbed end growth rate.

1.2.3 Actin filament bundling

Actin filaments can be branched by Arp2/3, though they are typically characterised as long and linear. These linear filaments can be bundled to increase structural integrity and increase the amount of force applied to a point of membrane contact. A classic example of actin bundles are the thin filaments which control muscle contraction (Squire, 2019). Actin bundling proteins such as fascin, members of the WASp family proteins, and alpha actinin facilitate this process by linking filaments through numerous F-actin binding domains (Winder and Ayscough, 2005).

1.2.4 Structural and mechanistic differences between G-actin and F-actin

Actin is referred to as F-actin after it is incorporated into a filament. Transition between G-actin and F-actin is accompanied by a conformational change which re-orientates the angles between the subunits to give protein a “flatter” appearance. This change is most reflected in the re-positioning of an alpha helix in subdomain 2 and a region of sub domain 4. The F-actin conformational change significantly increases the ATPase activity of subunits. Computational techniques predict a rate increase of around 40,000 times (McCullagh, Saunders and Voth, 2014). Consequently, F-actin structures more frequently occupy an ADP bound state than G-actin (Dominguez and Holmes, 2011).

ATP hydrolysis and the subsequent P_i release induces a further change as observed by X-ray crystallography (Oda *et al.*, 2009), although recent structural data suggests this change may not as large as previously expected (Dominguez, 2019). Cytosolic G-actin is usually bound to ATP resulting in the subunits at the fast-growing barbed end of filaments mostly occupying an ATP state. The likelihood of undergoing hydrolysis and entering an ADP state is time-dependant resulting in subunits at the slow-growing, pointed end more often being bound by ADP. This nucleotide state is less stable within a filament than ATP-actin which results in a higher dissociation rate. This difference between the relative stability of the barbed and pointed end results in a process referred to as treadmilling (**figure 1.3**) (Pollard, 1986; Dominguez and Holmes, 2011).

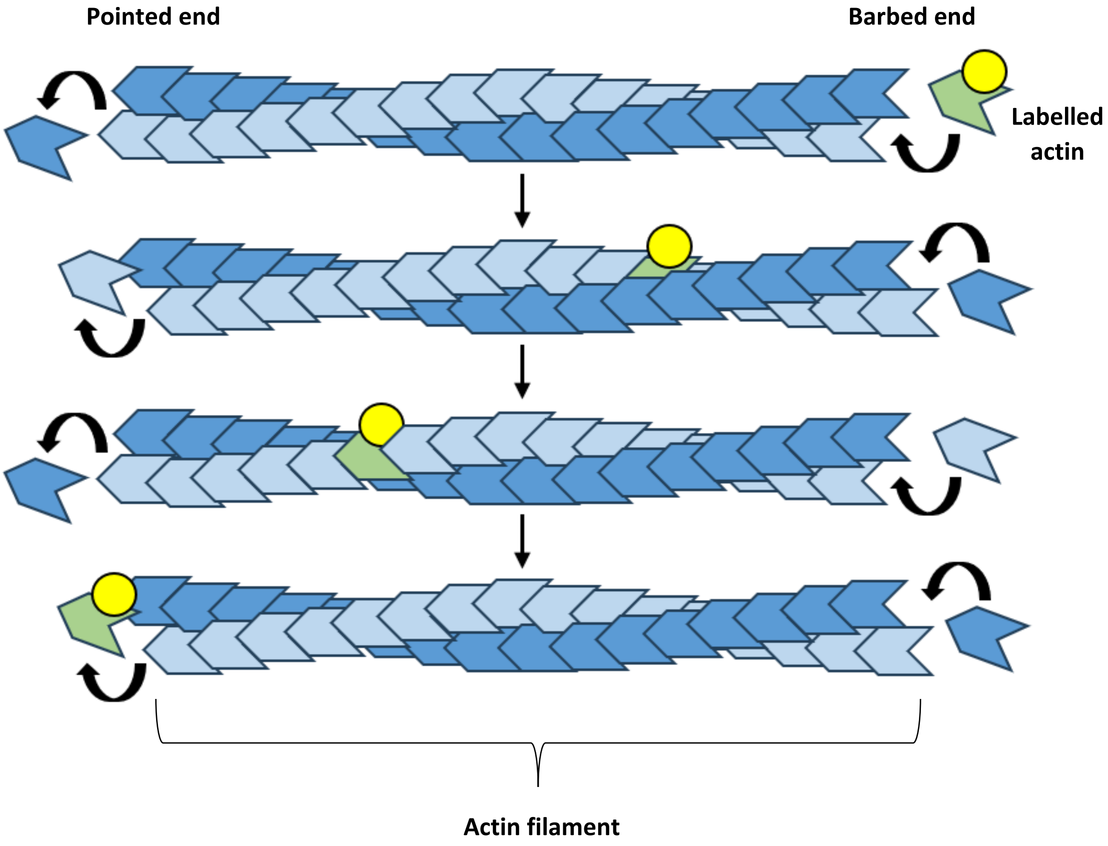


Figure 1.3: Cartoon representation of treadmilling. Actin subunits/monomers are shown as arrow heads with the indent representing the barbed end (hydrophobic cleft). The two sub filaments are shown using different shades of blue. An actin polypeptide bound to a fluorescently labelled actin-binding protein is shown in green with the fluorescently bound protein shown in yellow. Each row shows the composition of the filament moving through time. The fluorescently bound actin associates at the barbed end, its relative position appears to move along the filament as subunits are lost at the pointed end and added to the barbed end. This continues until the fluorophore dissociates at the pointed end.

The described conformational changes which accompany the transition between G and F-actin can greatly influence interactions with actin-binding proteins. ABPs generally prefer one of these two conformations resulting in the evolution of distinct monomeric and filamentous actin interactions. For example, gelsolin catalyses the severing of filaments which increases the relative concentration of barbed and pointed ends. This protein preferentially binds the F-actin conformation which helps direct its activity towards filaments (Burtnick *et al.*, 2004).

1.2.5 Control derived from treadmilling

Actin monomers are generally incorporated into growing filaments in an ATP state, undergo hydrolysis, and leave the filament in an ADP state as shown in **figure 1.3**. First theorised as the Wegner model in the 1970s, treadmilling can be observed by tracking the position of labelled actin-binding proteins on a filament (Wegner, 1976; Kuhn and Pollard, 2005). Due to the difference in growth rates between the filamentous ends, these labelled proteins appear to "move" through the filament from the barbed to the pointed end. This is visually reminiscent of a treadmill, hence the term.

Actin cytoskeletal networks are dynamic and are frequently re-organised to respond to stimuli and generate directed force. Actin filaments are essentially maintained through a balancing of association and dissociation rates due to treadmilling. This renders their lengths and integrity highly sensitive to external factors which may act upon these rates.

For example, capping proteins can be recruited to actin networks to preferentially bind the ends of established filaments. Barbed end capping proteins such as tensin impede the entry of new ATP-actin and thus promote depolymerisation whilst pointed end capping proteins such as Tropomodulins (TMs) decrease the rate of ADP-actin dissociation to reduce depolymerisation (Winder and Ayscough, 2005; Fowler and Dominguez, 2017).

1.2.6 Monomer sequestering

Actin polymerisation within the cytosol is favourable so long as there is a pool of available monomers. Theoretically, this monomeric pool would quickly deplete in an uncontrolled system undermining the dynamic nature of the actin cytoskeleton. Barbed and pointed end capping proteins can be used to manage the size of actin filaments as previously discussed. Additional machinery such as filament severing proteins, polymerisation enhancing proteins and depolymerisation factors are also employed to maintain robust control over filamentous networks (Winder and Ayscough, 2005). However, cells utilise an additional group of G-actin binding proteins to sequester a pool of monomers which can be released to the appropriate polymerisation sites.

G-actin binding proteins such as profilins, β-thymosins, and those containing a domain called Wiskott–Aldrich syndrome protein homology domain 2 (WH2) have a preference towards ATP bound actin whilst cofilins, twinfilins, and cyclase-associated proteins (CAPs) typically prefer ADP-bound G-actin (Xue and Robinson, 2013). This helps to separate the two monomeric pools to facilitate a more effect nucleation exchange of ADP-bound monomers prior being directed into a filament. These sequestering proteins also favour heavy interaction with the hydrophobic cleft which, as discussed previously, is required for strong longitudinal connections between actin polypeptides. Extensive masking of the cleft prevents unwanted interactions between monomers and provides cells with an additional method of polymerisation control (Pollard, 2016).

1.3 Nucleation

Actin filaments elongate readily in the presence of salt and available ATP-bound monomers (Pollard, 1986). However, the initial inter-monomer interactions that give rise to filaments (i.e., nucleation) are unfavourable and must rely on K_d affinity values estimated to be in the molar range (Sept and McCammon, 2001). This significantly reduces the occurrence of unwanted, spontaneous nucleation, especially when the majority of G-actin is actively sequestered by G-actin binding proteins. Consequently, the formation of new actin filaments must depend upon additional proteins which enhance nucleation (termed nucleators) and is highly dependent upon the concentration of available monomers. Nucleation works in concert with treadmilling to provide cells with greater control over their cytoskeletal composition. Treadmilling enables organisms to rapidly remodel actin networks, whilst nucleation offers cells a mechanism to tightly regulate the initial formation of these networks.

Restricting nucleation to defined locations through the spatiotemporal regulation of actin nucleators is vital to prevent unwanted polymerisation which can waste cellular energy, deplete monomeric actin reserves, and build unwanted cytoskeletal networks. This is well demonstrated by a variety of viruses including the Ebola and Marburg viruses which activate host Arp2/3 to nucleate actin filaments to propel themselves through the cytosol (Kloc *et al.*, 2022). Should nucleation not be tightly controlled, then similar force-driving networks may sporadically form throughout the cell.

1.3.1 Actin seeds

Whilst the initial interaction between actin monomers is unfavourable, the subsequent polymerisation proceeds readily. The minimum number of subunits required to reach this point of favourable monomer addition can be referred to as a “nucleus” and is typically defined as 3-4 monomers in length (Sept and McCammon, 2001). The term nucleus is often used interchangeably with the word “seed”. In agreement with Oda et al, this thesis will refer to seeds as the smallest oligomeric actin component required to achieve an elongation and depolymerisation rate consistent with a filament of any length (Oda, Aihara and Wakabayashi, 2016). Longitudinal nucleation will be described later in the thesis and applied to the computational modelling (**sections 1.3.4** and **6.6.4**). This nucleating scheme will require five actin subunits for subsequent polymerisation events to use binding rates consistent with a filament of any length – which fits the Oda, et al, 2016 definition of a seed. Therefore, to maintain consistent terminology throughout the thesis, a trimeric state will be used to describe a nucleus as elongation from this point is favourable. Meanwhile, seeds will be described as a pentameric state in the case of longitudinal nucleation (terminology described later in **section 1.3.4**). Nucleators function by either holding monomers in close spatial proximity to increase the rate of seed formation or by structurally mimicking a seed/nucleus.

1.3.2 Arp2/3

The Arp2/3 complex is a highly conserved eukaryotic nucleator (Machesky *et al.*, 1994; Welch, Iwamatsu and Mitchison, 1997). Its role and regulation has been extensively characterised using the bacterium *Listeria monocytogenes*. These bacteria can gain entry to the cytosol of human cells and move by triggering the nucleation of F-actin to generate force in a manner reminiscent of the Ebola and Marburg viruses (Cáceres, Abou-Ghali and Plastino, 2015; Kloc *et al.*, 2022). Presenting purified Arp2/3 to isolated *Listeria* was sufficient to reproduce this movement suggesting a close association between the complex and actin nucleation. This paved the way for *in vitro* studies, which extensively characterised activity of the complex (Welch, Iwamatsu and Mitchison, 1997; Mullins, Heuser and Pollard, 1998).

Arp2/3 functions to nucleate new actin filaments at a 70^{o} angle from the surface of an existing filament (termed mother filament), hence producing a branched actin network. Branched networks are frequently associated with force-driving functions such as clathrin-mediated endocytosis and lamellipodia formation (Chesarone and Goode, 2009).

As previously discussed, eukaryotic genomes contain many actin paralogues that maintain a high degree of structural similarity. Two of the Arp2/3 complexes' seven conserved subunits (Arp2 and Arp3) are key examples of such paralogues. Purified Arp2/3 revealed that these subunits are held in a "splayed" arrangement with an end-to-end orientation that shields the hydrophobic cleft of Arp3 (**figure 1.4**). This conformation presents Arp2/3 as an extremely weak nucleator (Goley and Welch, 2006).

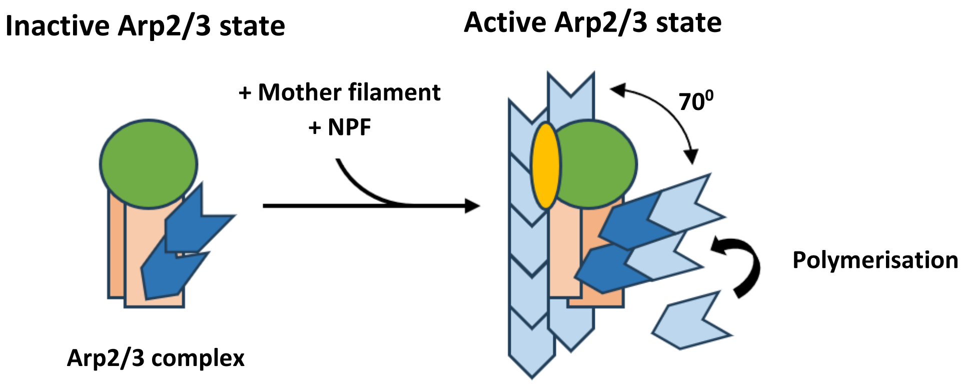


Figure 1.4: Arp2/3-mediated nucleation. Actin subunits/paralogues are shown as arrow heads with the barbed end (hydrophobic cleft). The actin paralogues (Arp2 and Arp3) are shown in dark blue while the two clamp subunits are shown as different shades of orange/brown. The rest of the Arp2/3 complex is shown in green. Following binding of a mother filament and NPF (light blue and yellow respectively), the clamp domains are rotated which re-orientates the Arp2 and Arp3 subunits such that they can elongate a new filament at a 70^0-degree angle from the mother filament.

Simultaneous binding of both a mother filament and an activating protein can reorientate the actin paralogues to expose both hydrophobic clefts. This structurally mimics the barbed end of a nucleus and is followed by the addition of an actin monomer (often supplied by the activator) to seed the filament. Arp2/3 activators are given the name Nucleation Promoting Factors (NPFs) and this group includes the Wiskott-Aldrich syndrome protein (WASp) family proteins (Goley and Welch, 2006). The *Listeria monocytogenes* protein capable of activating host Arp2/3, ActA, functions by mimicking this native interaction (Skoble, Portnoy and Welch, 2000). It should be noted that WISH/DIP/SPIN90 (WDS) family proteins can activate Arp2/3 in the absence of a mother filament by binding more closely to the complex's "clamp" subunits – proteins which play an important role in the structural repositioning of Arp2 and Arp3 (Shaaban, Chowdhury and Nolen, 2020).

1.3.3 Formins

Formins were the second class of actin nucleators to be identified following the characterisation of Arp2/3 (Sagot *et al.*, 2002). These homodimeric proteins can both enhance the elongation of pre-existing filaments and create new linear filaments through *de novo* nucleation. Rather than structurally mimic filament nuclei like Arp2/3, formins capture multiple G-actin monomers and closely position them in an orientation that facilitates nucleation (Weston, Coutts and La Thangue, 2012).

Formins are large multidomain proteins that can dimerise to form a ring with their FH2 domains. These domains are the minimum nucleating component and can bind either F-actin at the barbed end or monomeric G-actin. This occurs via two distinct pairs of binding sites which primarily interact with the hydrophobic, barbed end actin clefts (Tu *et al.*, 2012; Courtemanche, 2018). Structural studies reveal that this is accompanied by closure of the FH2 domains. This essentially forms a loop around two to three actin subunits which is maintained by a hydrophobic lasso-post interaction, the result of which is an increase to the overall complex affinity (Tu *et al.*, 2012; Thompson *et al.*, 2013). A new filament is nucleated when the substrate of this interaction is G-actin (**figure 1.5**).

Formins can employ their FH1 domains to enhance elongation. FH1 domains are highly variable in length between organisms, yet all share a high percentage of proline that can interact with profilin (Courtemanche, 2018).

Most G-actin is bound by sequestering proteins to prevent uncatalysed nucleation and to allow cytoskeletal components to distinguish between ATP-bound and ADP-bound actin which promotes efficient nucleotide exchange (explained in **section 1.2.6**). Profilin is the main sequesterer of ATP-actin and so recruitment via profilin helps to ensure that only polymerizable ATP-actin is directed towards the growing barbed end. Many formins possess additional actin-binding sites outside of the FH1-FH2 region, including WH2 domains, that may help provide additional monomers (Courtemanche, 2018).

Arp2/3 mediated nucleation is inhibited by profilin suggesting that the preference of formins towards profilin-actin may allow it to access an alternative pool of G-actin. This would prove beneficial for dividing cells where formins must compete with Arp2/3 for G-actin during the assembly of their contractile rings (Suarez *et al.*, 2015).

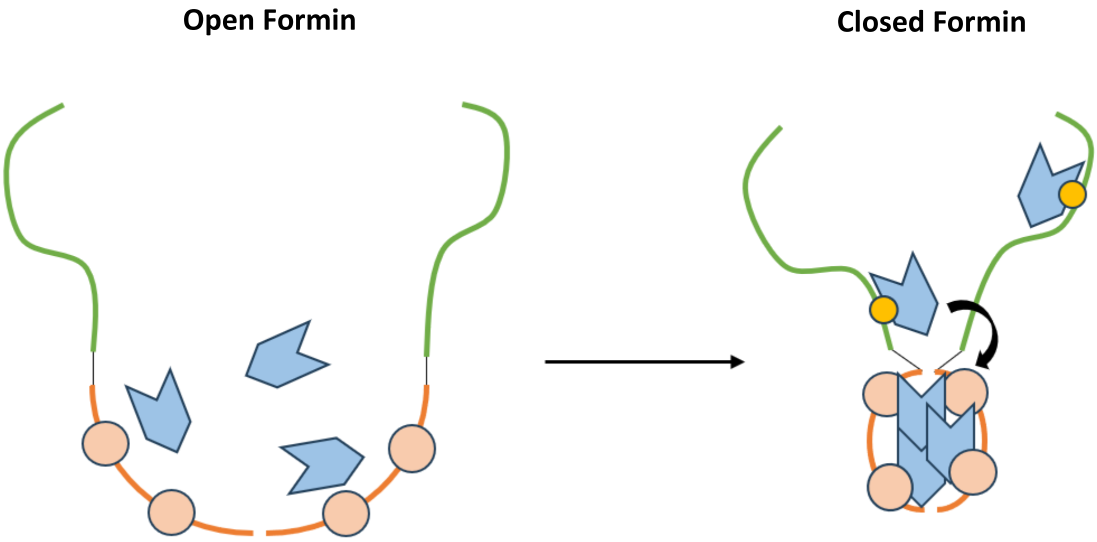


Figure 1.5: Formin-mediated nucleation. Actin subunits/monomers are shown as arrow heads with the barbed end (hydrophobic cleft). Profilin is shown as a yellow circle bound to actin. FH2 domains are shown in orange with the FH1 domain in green. Upon binding three actin monomers, formin dimers can close their constituent FH2 loop whilst contacting either two or three actin subunits. This constricted orientation encourages the FH2-bound actins to nucleate. Profilin-bound monomers can then be directed to the growing barbed end via the FH1 domains. This figure was modified from **figure 1** of Tu *et al.*, 2012.

1.3.4 Tandem binding sites

A third class of actin nucleators were identified in 2005 with the discovery of Spire-mediated nucleation (Quinlan *et al.*, 2005). Unlike the well-defined structural control employed by Arp2/3 and formins, tandem-monomer-binding nucleators (henceforth termed TMBNs) function by presenting a series of three to four G-actin-binding interfaces. These may each hold monomers in close spatial proximity such that they may orientate themselves into a nucleus to ultimately seed a *de novo* filament.

1.3.4.1 Tandem binding sites: WH2 domains

WH2 domains are actin-binding motifs roughly 17 residues in size that interact with the hydrophobic, barbed end cleft of actin (Chereau *et al.*, 2005). In contrast with the FH1 domain of formins which recruit actin via profilin, WH2 domains interact directly in a manner often antagonistic to the profilin-actin interaction (Bieling *et al.*, 2018). This allows WH2 domains to access an alternative pool of G-actin than elongating formins. Spire, JMY, and Cordon-bleu (Cobl) all possess a series of WH2 domains that have been shown to nucleate actin (Quinlan *et al.*, 2005; Ahuja *et al.*, 2007; Zuchero *et al.*, 2009).

Spire plays a critical role in oogenesis by nucleating the actin network that establishes the major body axes. Drosophila Spire (Spir) and the formin Cappuccino are known to develop this actin patch through a poorly understood synergy that requires physical interaction between the two nucleators (Quinlan, 2013). The C-terminal FYVE domain of Spire is suspected to fold back on its formin-binding N-terminal KIND domain to maintain a degree of autoinhibition. This inhibition may be released upon interaction with the membrane (Tittel *et al.*, 2015). Spire has long been theorised to oligomerise via association with formins, which are themselves dimerised. Supporting this, artificial dimerisation of the Spire tandem WH2 domain region (via gene fusion with a dimerisation domain) significantly increases the rate of nucleation (Namgoong *et al.*, 2011; Welz and Kerkhoff, 2023).

JMY is an ARP2/3 activator that is highly expressed within brain tissue and was first identified as a cofactor for p53/TP53. Later, the protein's nucleating ability was linked to cell migration, the inhibition of neuritogenesis in neurones (formation of neurites), and autophagy (Zuchero *et al.*, 2009; Firat-Karalar, Hsiue and Welch, 2011; Kast *et al.*, 2015). JMY contains a C-terminal WCA region along with two additional WH2 domains, a proline rich domain, coiled-coil domain, and a C-terminal LC3-interaction region. WCA regions are commonly associated with

NPFs and contain a monomer-binding WH2 domain and Arp2/3-activating central and acidic regions. Expression of a shorter, C-terminal construct covering just the WH2 and WCA domains was evidenced to suggest a degree of autoinhibition applied by the rest of the protein (Firat-Karalar, Hsiue and Welch, 2011). More recently, LC3 and STRAP have been shown to enhance and inhibit JMY respectively (Hu and Mullins, 2019).

Spire contains four WH2 domains whilst JMY contains three. Nevertheless, it has been hypothesised that these TMBNs both share a similar nucleating mechanism. The WH2 domains of each nucleator are spaced to orient their bound actin monomers end-to-end which encourages longitudinal contacts (Firat-Karalar and Welch, 2011). X-ray crystallography structures using Spire and a synthetic tandem of WASp WH2 domains (spaced with linker lengths roughly equal to those found in both Spire and JMY) showed that actin was arranged longitudinally which supports this mechanism (Ducka *et al.*, 2010; Rebowski *et al.*, 2010). Therefore, tandem WH2 nucleators such as Spire and JMY encourage the formation of a single actin sub filament strand rather than a true barbed-end seed which suggests recruitment of two to three additional actin subunits is required to form the base of the second sub filament strand (Sitar *et al.*, 2011). This thesis will refer to these proteins as “linear tandem nucleators” (**figure 1.6a**).

An important note is that JMY is also a nucleation promoting factor for Arp2/3. JMY is therefore proposed to nucleate the mother filament required for Arp2/3 branching using its tandem WH2 domains, whilst also activating the complex with its central and acidic regions (Zuchero *et al.*, 2009). Essentially, the WCA region typically associated with NPFs is extended by another two WH2 domains in JMY which allows for Arp2/3 activation in the absence of a pre-existing actin network.

Cordon-bleu (Cobl) nucleates actin via three WH2 domains which is similar to JMY. However, the TMBN differs in that the linker separating the C-terminal and central WH2 domains results in a greater separation than the WH2s of other tandem nucleators (Ahuja *et al.*, 2007; Firat-Karalar and Welch, 2011). This allows Cobl to position its three bound actin monomers in a cross-filament orientation which can be followed by immediate filament elongation (Ahuja *et* *al.*, 2007). This thesis will refer to TMBNs employing this mechanism as "cross-filament tandem nucleators" (**figure 1.6b**).

The difference between linear and cross-filament nucleators is reflected in their nucleation rates. *In vitro* assays using Spire require high concentrations of the tandem nucleator (0.5-1 µM) to match the rates produced using low-nanomolar concentrations (20 nM) of Arp2/3 (Quinlan *et al.*, 2005). Meanwhile, the activity of Cobl is roughly comparable to Arp2/3, with a maximum nucleation rate being reported at ~40 nM (Ahuja *et al.*, 2007).

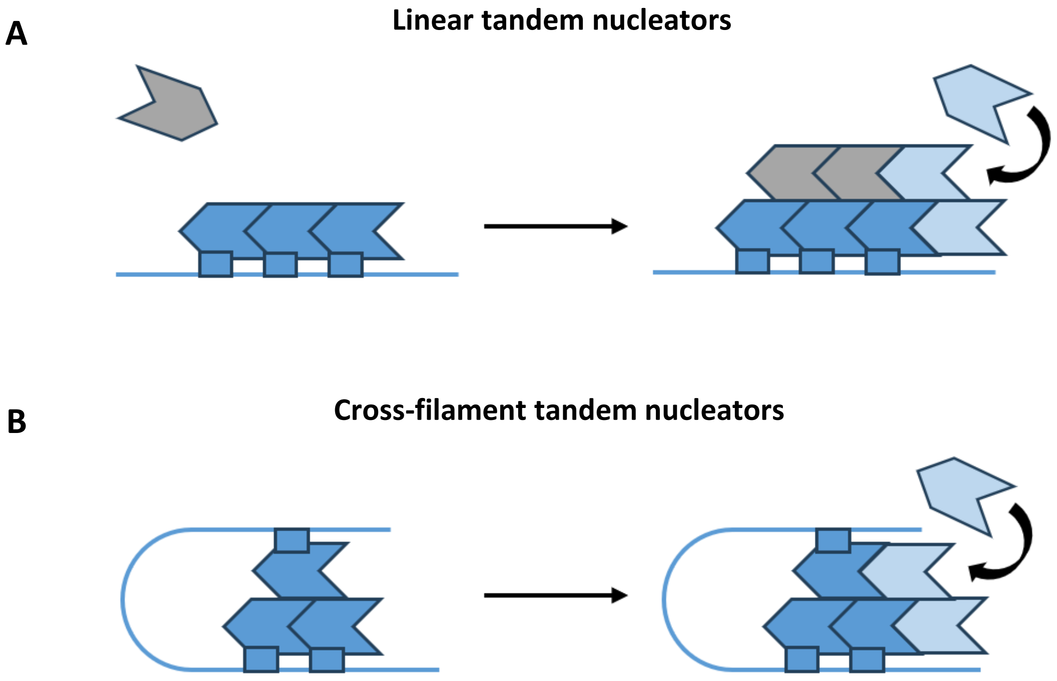


Figure 1.6: Comparing the linear and cross-filament tandem nucleators. Complexes on the left of the arrow represent early nucleation whilst complexes on the right of the arrow show the subsequent seed elongation. Actin subunits/monomers are shown as arrow heads with the indent representing the barbed end (hydrophobic cleft). Actin monomers bound to each nucleator (blue lines with boxes representing WH2 domains) are shown in dark blue while actin subunits that are recruited post seed formation are shown in light blue. **A**) Linear tandem nucleators require additional monomers to bind prior to completion of the seed (shown in grey) to form the base of the non-nucleator bound sub filament. This is associated with a slower rate of nucleation when compared to **B**) cross-filament nucleators.

1.3.4.2 Tandem binding sites: Leiomodin

Leiomodin (Lmod) is an actin nucleator functionally associated with muscle cells (Chereau *et al.*, 2008). It is hypothesised that Lmods may nucleate the actin cytoskeletal components required for muscle contraction (thin filaments) through tandem G-actin binding sites (Tolkatchev, Gregorio and Kostyukova, 2022). Unlike Spire, JMY, and Cobl which employ a series of repeated actin-binding motifs, Lmods nucleate actin via numerous, distinct actin-binding domains. These include (I) a central, alpha helical ABS1 domain which binds the tops of subdomains 2 and 4, (II) a large ABS2 domain which contacts the back of subdomains 1 and 2, (III) an uncharacterised proline-rich domain that may bind actin and (IV) a C-terminal WH2 domain that interacts with the barbed end cleft. Together, these four domains are hypothesised to "wrap around" three actin monomers via its multiple, distinct actin-binding surfaces (Fowler and Dominguez, 2017). In this manner, the nucleating mechanism of Lmods share a similarity to the formin FH2 domain (**Figure 1.5**) in that both form an enclosed "wrap" around three actin monomers through a variety of distinct actin-nucleator contacts to form a cross-filament nucleus.

A key observation for this thesis (due to its similarity to the later discussed Las17) is that the actin-binding interaction of the proline-rich domain is highly uncharacterised (Tolkatchev, Gregorio and Kostyukova, 2022). This region was first thought to recruit G-actin through the polyproline-binding protein profilin, as proline mutations in this region significantly reduced nucleation. However, it was later concluded that profilin inhibited Lmod suggesting a possible novel interaction (Chereau *et al.*, 2008).

1.3.4.3 Tandem binding sites: APC

The nucleation mechanism of adenomatous polyposis coli (APC) is poorly understood despite its vital function in regulating the cytoskeletal interplay between microtubules and actin during cell migration. Nevertheless, broad details have been developed following the discovery of its actin nucleating activity *in vitro*, and more recent discovery *in vivo* (Okada *et al.*, 2010; Juanes *et al.*, 2017).

APC can dimerise via multiple self-association domains which brings the C-terminal basic regions of each subunit together. These regions can each bind two G-actin monomers via actin nucleation sequences (ANS1 and ANS2) that can be brought together through the APC dimerisation to form a tetrameric seed (Juanes *et al.*, 2017). The APC basic region can also interact with microtubules and formins which play a key role in cooperating with, and directing, APC's function. The microtubule-binding protein EB1 can also bind this region to inhibit nucleation through direct occlusion of the G-actin binding sites (Juanes *et al.*, 2020).

1.3.4.4 Tandem binding sites: Bacterial nucleators

Pathogenic organisms are frequently observed to hijack well-conserved host machinery. This phenomenon can often be employed to improve our understanding of the hacked system as was the case for Arp2/3 and the bacterium *Listeria monocytogenes* (Welch, Iwamatsu and Mitchison, 1997). Likewise, *Vibrio parahaemolyticus* has helped our understanding of WH2 domain nucleators through the discovery that Spire may function as a dimer. This same study also expanded our knowledge of the bacterial actin nucleator VopL (Namgoong *et al.*, 2011).

VopL can bind G-actin via a series of three WH2 domains and can dimerise via a C-terminal self-associating domain. It has been proposed that Spire and VopL may share a similar nucleation scheme in binding a series of G-actin monomers longitudinally. VopL dimerisation is suggested to facilitate the formation of a hexameric seed through direct interactions between the growing protofilaments of each VopL subunit. This occurs in cooperation with the VCD domain which briefly contacts the pointed end of the seed (Burke *et al.*, 2017). The nucleator VopL also shares a strong sequence similarity with VopF and VopN suggesting that all three may nucleate actin using a conserved mechanism (Namgoong *et al.*, 2011).

1.3.5 Pyrene assays

The function of distinct nucleators can be explored through their effect on actin polymerisation. Attaching pyrene – an environmentally sensitive fluorophore – to actin allows for observation of this process. As pyrene-actin is incorporated into an actin filament, its local environment changes which induces a fluorescence change. The overall effect of this is that as more actin (and by extension pyrene-actin) enters an F-actin state, the emission profile of the solution changes. An example of pyrene assay data is given later in **figure 4.1** (Doolittle, Rosen and Padrick, 2013).

1.4 Clathrin-Mediated Endocytosis

The aim of this thesis is to better understand the early F-actin development at the endocytic patch. Following the broad overview of cytoskeletal systems, actin nucleation, and the protein factors which govern it, this chapter will now focus in on clathrin-mediated endocytosis and the role of the yeast WASp homologue, Las17.

All eukaryotic cells are enveloped by an outer lipid membrane. This not only functions as a barrier between the cytosolic and external environments but can also act as a platform for signalling systems. Clathrin-mediated endocytosis is used by eukaryotes to – among other things – allow extracellular nutrients to cross the lipid barrier and recycle or degrade membrane-associated signalling proteins (Feliciano and Di Pietro, 2012). CME can be simplified to a four-stage process (**figure 1.7**).

CME begins when early coat proteins recognise and bind the endocytic sequences from cargo proteins forming a flat ‘endocytic patch’ (Goode, Eskin and Wendland, 2015). Some of these proteins act as adaptors to bridge interactions between cargo and the coat protein clathrin (Howard *et al.*, 2002). While this first step may occur over a varied timescale (1-2 minutes in yeast), binding of the late coat proteins both commits the patch to endocytosis and marks the start of the more time-regimented steps. In budding yeast, late coat proteins include Sla1 and Las17. The third step of CME commences around 15-20 seconds after the arrival of Las17. This step is referred to as the 'invagination phase' and involves the centre of the endocytic patch invaginating (Goode, Eskin and Wendland, 2015). Invagination is coupled with a wave/burst of actin polymerisation which functions as a significant driving force for the membrane movement (Tyler, Allwood and Ayscough, 2016). The fourth and final stage of CME involves scission of the clathrin-coated, cargo-containing vesicle which can later fuse with endosomes for processing (Goode, Eskin and Wendland, 2015).

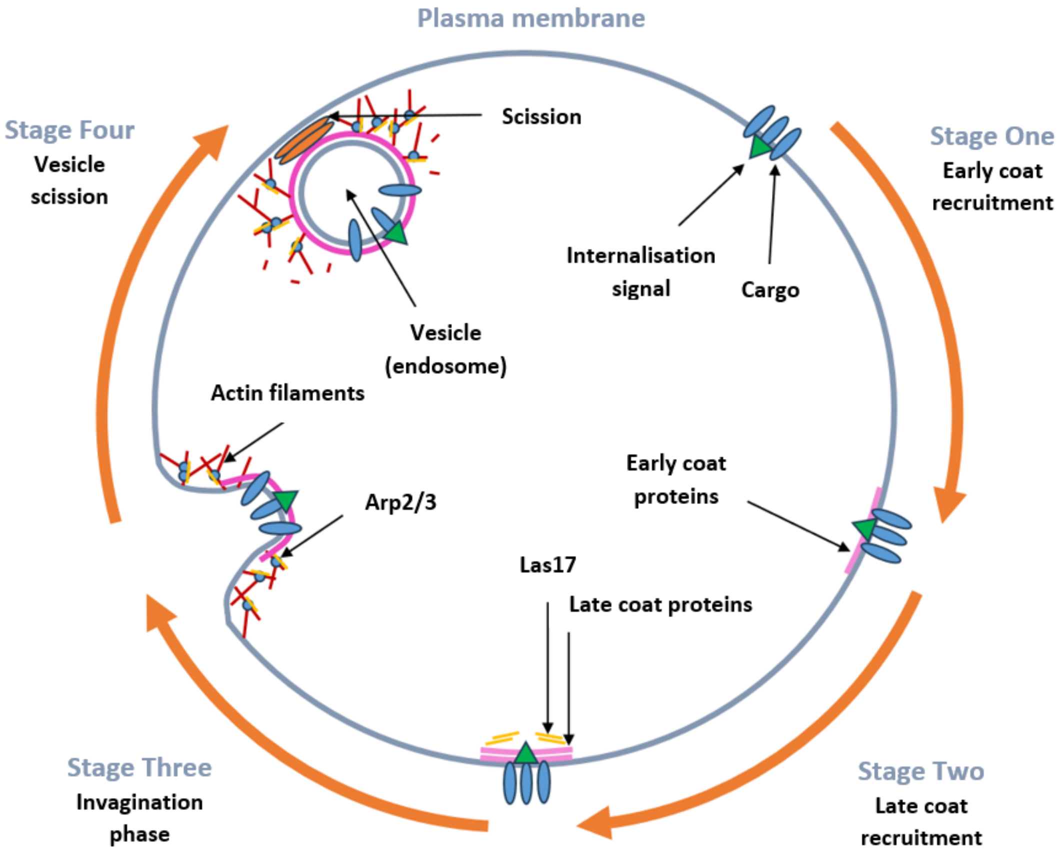


Figure 1.7: The four main stages of clathrin-mediated endocytosis. CME begins when early coat proteins recognise an internalisation sequence (stage one). Late coat proteins, including the NPF Las17, are recruited (in yeast) several minutes after signal expression (stage two) before Arp2/3 is recruited to drive actin polymerisation in a step often referred to as the invagination phase (stage 3). Finally, scission machinery can act upon the intrusion to capture the cargo proteins in a vesicle while proteins such as cofilin mediate disassembly of the F-actin patch produced during the invagination phase (stage 4) This figure was influenced by **figure 1** from (Goode, Eskin and Wendland, 2015).

Clathrin-mediated endocytosis is a highly conserved and critical process (Moseley and Goode, 2006). Mutating CME proteins associated with the regulation of actin polymerisation can induce severely deleterious phenotypes (Madania *et al.*, 1999), and thus achieving a greater understanding of these regulatory mechanisms will greatly benefit our understanding of the overall endocytic process and the development of actin networks in general.

1.4.1 The major drivers of invagination – clathrin and actin

A key component of CME is, as reflected in the name, clathrin. This is referred to as a coat-protein as it surrounds the internalised membrane to maintain structural coherency, define vesicle size, and sort the vesicle to the appropriate location. Clathrin is arranged as a tessellated polygonal scaffold made of hexameric oligomers connected to the membrane via a mosaic of adaptor proteins including AP2 and Sla1 (Taylor, Perrais and Merrifield, 2011; Tolsma, Cuevas and Di Pietro, 2018). Each individual oligomer contains three clathrin heavy chain subunits (which predominantly play a structural role) and three light chain subunits (which primarily play a regulatory role) (Djakbarova *et al.*, 2021). Because each hexameric clathrin unit becomes slightly curved following membrane recruitment, formation of the clathrin coat exerts forces upon the plasma membrane. These forces encourage membrane curvature, and thereby development of an endocytic invagination (Lacy *et al.*, 2018). However, this appears to be less critical of a factor in yeast where – unlike in mammalian cells – endocytosis can still occur in the absence of clathrin (Merrifield and Kaksonen, 2014). It is still unclear whether the transition of clathrin from a flat to curved hexamer is an immediate product of adapter-mediated recruitment, or whether this transition only occurs after a critical concentration of clathrin hexamers is localised (Djakbarova *et al.*, 2021). Other mechanisms such as the recruitment of BAR domains and (possibly) molecular crowding also support clathrin in encouraging membrane curvature (Lacy *et al.*, 2018).

Assays have also revealed the importance of actin polymerisation during the invagination phase – particularly in budding yeast. Actin polymerisation exerts forces that are used during CME to help overcome cellular turgor pressure, support clathrin (and other membranecurving mechanisms) in deforming the plasma membrane, and drive invagination. This was demonstrated in a study which mutated actin bundling related genes and found this reduced the likelihood for complete endocytic progression. Using sorbitol to reduce the effect of turgor pressure partially restored endocytic activity suggesting that the mutations reduced the magnitude of polymerisation-linked forces. The nucleator Arp2/3 functions as the main driver of polymerisation during the invagination step and thus plays an integral role during CME, particularly in yeast (Aghamohammadzadeh and Ayscough, 2009).

1.4.2 Properties of the endocytic actin network

Actin networks can be used to provide a variety of mechanical forces as described throughout **section 1**. The spatial organisation of nucleation and polymerisation/depolymerisation factors dictates whether a rapidly developed F-actin network results in membrane invagination (e.g., endocytosis) or protrusion (e.g., lamellipodia). This is because actin polymerisation directed towards a membrane exerts a pushing force at the site of contact which is energetically balanced by surrounding pulling forces. The filamentous network generated in yeast endocytosis is developed as a ring around the internalisation signal that pushes the membrane outwards to create a retrograde flow at the ring's centre, thus pulling part of the membrane inwards where it can be pinched off to form an internal vesicle (Carlsson, 2018).

A critical component of this mechanism is the spatiotemporal control maintained over CME-linked proteins to maintain the correct cytoskeletal shape. Therefore, understanding how endocytic nucleation occurs in the context of a dynamically changing patch will substantially aid our understanding of how invagination is correctly produced and maintained.

1.4.3 Identifying proteins involved with CME

Mutant screens have identified many CME-linked proteins, yet provide little insight into how these proteins function (Goode, Eskin and Wendland, 2015). Live-cell fluorescence microscopy has proven to be a popular method in identifying both when CME-linked proteins are recruited to the endocytic patch as well as their residence time (i.e., how long they stay at the patch) (Feliciano *et al.*, 2015; Allwood *et al.*, 2016). Understanding the spatiotemporal behaviour of these proteins is an important step in deciphering how their associated mechanisms work. CME involves a complex network of interactions and involves over 50 proteins with Arp2/3 being one of the most fundamental (Goode, Eskin and Wendland, 2015).

1.4.4 The role of Arp2/3 and the question of its mother filament

The burst in F-actin associated with the invagination phase directly follows the recruitment of Arp2/3 with the experimental addition of a competitive Arp2/3 inhibitor (CK-666) reducing endocytic activity in yeast (Feliciano and Di Pietro, 2012; Burke *et al.*, 2014). In conjunction with the fact that CME actin patches consist of a branched actin network, these observations demonstrate that Arp2/3 is the key driver of nucleation during the invagination phase.

Arp2/3 functions as a branching nucleator as explored in **section 1.3.2** which brings about an interesting question; how do mother filaments first arrive at the patch? Unlike mammalian cells, yeast does not possess an extensive cortical actin network close to its outer membrane and sometimes undergoes endocytosis in areas lacking an apparent F-actin meshwork. Furthermore, actin nucleation at the membrane of mammalian endosomes can occur in regions with little to no visible F-actin which renders the notion that mother filaments are solely supplied from a pre-existing cortical network unlikely (Tyler, Allwood and Ayscough, 2016).

One of the most potent Arp2/3 NPFs in budding yeast is Las17, homologue of the WASp protein family. This protein is also recruited to the endocytic patch prior to Arp2/3-mediated polymerisation suggesting that Las17 plays a critical role in the regulation of actin polymerisation during the invagination phase (Sun, Martin and Drubin, 2006; Goode, Eskin and Wendland, 2015).

It is now thought that Las17 may play a role generating these mother filaments as *in vitro* assays suggest that the NPF can nucleate actin to form seeds without Arp2/3 present (Urbanek *et al.*, 2013; Tyler, Allwood and Ayscough, 2016). Furthermore, truncating the Arp2/3-activating CA region of Las17 has only a minor impact on endocytosis (Galletta, Chuang and Cooper, 2008). Such a nucleating function may also explain why the time between Las17 arrival and the Arp2/3-driven F-actin burst varies much less than the previous CME stage (Feliciano and Di Pietro, 2012). Furthermore, the tandem WH2 nucleation shown by the neuritogenesis inhibitor JMY demonstrates a precedent for Arp2/3 NPFs also possessing *de novo* nucleating abilities (Zuchero *et al.*, 2009).

An additional theory derived from work in fission yeast (named the “sever, diffuse, and trigger” hypothesis) posits that cofilin may sever cortical actin filaments which can diffuse through the cytosol to be used as endocytic mother filaments. However, this study also noted the possibility that these severed filaments may act in conjunction with Las17-nucleated filaments to deliver a concentration of mother filaments sufficient for Arp2/3 endocytic activity (Chen and Pollard, 2013).

1.5 Las17 structure and function

Various assays have revealed several functional regions of Las17 (**figure 1.8**). At the N-terminus, a WH1 domain can bind lipid membranes and several CME-associated proteins such as Vrp1 (Goode, Eskin and Wendland, 2015; Allwood *et al.*, 2016). At the C-terminus, the “WH2 Central Acidic” (WCA) region confers the NPF activity of Las17. The central and acidic sub-regions can bind to, and activate, Arp2/3 whilst the WH2 domain binds G-actin (Urbanek *et al.*, 2013). The WH2 domain may function to scavenge G-actin and increase the local monomer concentration at the nucleation sites, thus increasing the rate of nucleation and polymerisation (Allwood *et al.*, 2016). Artificial dimerisation via expression of a WASp GST-WCA construct has been shown to increase polymerisation activity suggesting that dimerization may play a role in the function of WASp family proteins (Padrick *et al.*, 2008).

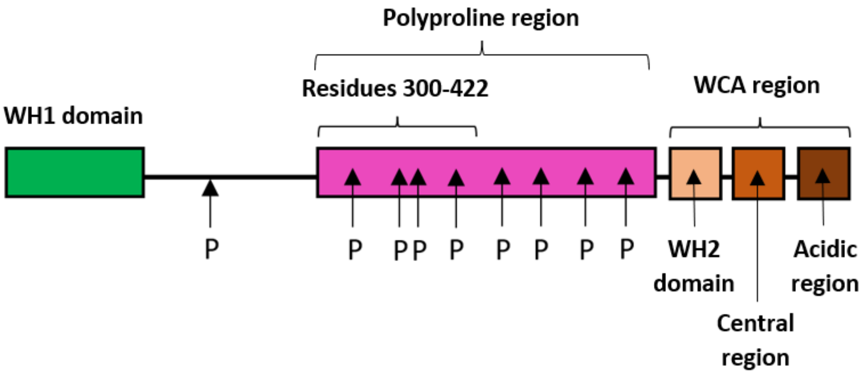


Figure 1.8: Structural regions of Las17. The WCA region is a defining feature of NPFs and includes an actin monomer binding WH2 domain and central + acidic Arp2/3 binding and activation region. The centre of Las17 is dominated by a large polyproline region that contains eight polyproline tracts (≥5 continuous prolines) and each are identified with the letter "P". The minimum nucleating construct of this region spans residues 300 to 422 and includes four tracts. Tracts two and four of the polyproline region (from N to C direction) are believed to be the two strongest actin-binding tracts thanks to their N-terminal double arginine. Las17 can bind membranes and autoinhibit its WCA region via the WH1 domain.

N-terminal to the WCA region is the polyproline (PP) region, which is predicted to lack tertiary structure. The region possesses eight polyproline tracts (stretches of five or more consecutive prolines) and thus numerous PxxP-motifs (Feliciano and Di Pietro, 2012; Urbanek *et al.*, 2013). Human WASp plays a largely unknown role in hematopoietic malignancies whilst mutations linked to the polyproline region are associated with immune deficiency (Jin *et al.*, 2004; Biber *et al.*, 2021). It was historically thought the sole function of the PP-region was to recruit possible interaction partners as PxxP motifs make up the core binding of SH3 domains and many CME-linked proteins possess such domains (i.e., Las17 functions as a scaffold). However, evidence suggests that Arp2/3-independent nucleation also occurs in this polyproline stretch suggesting that the region may also play an active role in endocytosis (Urbanek *et al.*, 2013; Allwood *et al.*, 2016).

1.5.1 Identifying the G-actin binding sites of Las17

Although the nucleating ability of Las17 has been demonstrated, questions remain about exactly how, and where, the initial binding events occur. Preliminary lab data from the Ayscough lab suggests the minimum nucleation construct of Las17 covers residues 300-422 of the polyproline region (E Allwood and K Ayscough, personal communication)(Urbanek *et al.*, 2013; Allwood *et al.*, 2016). Polyproline regions are known to recruit actin via profilin, however, profilin is not known to bind Las17 and its addition to *in vitro* assays does to affect the nucleating properties of Las17 (Urbanek *et al.*, 2013; Allwood *et al.*, 2016). Furthermore, the actin-binding region contains no currently characterised actin binding sites, suggesting a novel interaction mechanism must be at play. Yeast two-hybrid assays between actin and different peptide fragments from the PP-region reveal that residues 300 to 404 were capable of binding actin with the core binding of this interaction localised between residues 330 and 404 (Feliciano *et al.*, 2015). The region identified covers four polyproline tracts that this thesis will term PP1 to PP4. An additional polyproline tract is present N-terminal of residue 300. However, this region is separated from PP1 by a 134-residue linker, is not typically present during KA lab in vitro assays, and has not been investigated within any published study. In lieu of experimental data, this tract was excluded from the modelling.

A yeast two-hybrid study identified that mutations to actin aspartate and glutamate residues could disrupt binding suggesting the likely importance of basic residues in the Las17-actin interaction (Urbanek *et al.*, 2013). The core 300-404 region was compared against numerous WASp-family homologues revealing two highly conserved arginine pairs (R349, R350 and R382, R383) N-terminal to polyproline tracts PP3 and PP4 (Feliciano *et al.*, 2015). Mutating both pairs to alanine significantly reduces actin binding, Arp2/3-independent nucleation and elongation revealed that these residues play a crucial role in the binding mechanism (Allwood *et al.*, 2016). When this mutation was applied *in vivo*, endocytic progression was slowed (both before and after Arp2/3 recruitment) highlighting the importance of Arp2/3-independent activity during CME (Allwood *et al.*, 2016).

Removal of the arginine pairs did not completely abolish binding however, suggesting that other weak interactions, possibly mediated via the neighbouring polyproline tracts, also contribute to the binding (Allwood *et al.*, 2016). Supporting this, proline to alanine mutations at the C-terminal tracts greatly reduced the effect of Las17 on polymerisation.

Interestingly, the first polyproline tract in the 300-422 region (PP1) was also identified to possess two N-terminal arginines, albeit not contiguous. The residue sequence separating these three arginine-neighbouring tracts is roughly the same as the distance separating the tandem-nucleating WH2 domains of Spire. As previously discussed, leiomodin may also possesses an uncharacterised actin-binding site in its polyproline region (Tolkatchev, Gregorio and Kostyukova, 2022) suggesting that proline rich sequences may play a yet undiscovered role in direct actin binding.

While the nucleating half of the PP-region (300-422) possesses an affinity for G-actin and not F-actin, the polyproline tracts at the C-terminal half have a much higher F-actin affinity (Feliciano *et al.*, 2015; Allwood *et al.*, 2016). A possible nucleating mechanism involves G-actin being nucleated by the N-terminal tracts while the C-terminal tracts bind the newly nucleated filaments. It is hypothesised that this F-actin binding may both prevent the escape of mother filaments from the endocytic patch and/or allow for more efficient elongation (Allwood *et al.*, 2016).

1.5.2 Significance of proline-rich sequences

Proline is unique in that its sidechain forms a bond with its backbone amide, creating a cyclical shape. This backbone-interaction severely constrains one of the dihedral angles whilst also impacting the conformation of its N-terminal neighbour. Furthermore, the backbone interaction leaves the amide group lacking a proton for donating hydrogen bonds whilst its electron rich nature – caused by electron donation from the delta carbon – leaves the preceding carbonyl oxygen a good hydrogen bond acceptor. Proline can therefore act to interrupt secondary structure. Consequently, proline is frequently found separating secondary structural elements whilst proline-rich sequences often occupy a polyproline helix (Williamson, 1994). Supporting this, AlphaFold predicts that the polyproline region of Las17 may adopt an extended and polyproline helix rich conformation (Jumper *et al.*, 2021; Varadi *et al.*, 2022).

Polyproline helices can occupy either a type I (PPI) or type II (PPII) conformation depending upon whether their peptide bonds occupy a *cis* or *trans* conformation respectively. Out of these two, the PPII conformation is far more energetically favourable and is therefore frequently observed in protein structures – particularly collagen (Adzhubei, Sternberg and Makarov, 2013). Type II helices usually occupy an extended conformation that renders them easy targets for binding proteins. This is especially true for proteins which have tandem repeats of >4 consecutive proline residues such as Las17. The rigidity conferred from a high proline content results in a low entropy for unbound polyproline binding sites. This reduces the magnitude of the entropy drop upon binding, and thus the energetic penalty, allowing for rapid binding rates such as those observed for SH3 domains (Williamson, 1994).

While proline does not interact with membranes particularly strongly, the extended secondary structure conformation resulting from a high proline content can increase exposure of local hydrophobic residues. These in turn interact with membranes to reduce solvent exposure and thereby interact with the lipid bilayer (Top *et al.*, 2012). This may also serve as mechanism by which Las17 can position itself along the membrane, in addition to the directed binding of its WH1 domain.

These factors partially explain why many multi-ligand binding regions of membrane-associated proteins are proline rich, such as the central region of WASp family proteins, the estrogen receptor (ER) coactivator, PELP1 (De Luca *et al.*, 2017), and salivary proteins (Murray *et al.*, 1994). This is also true for non-scaffolding proteins such as the myosin light chain kinase in which its proline rich N-terminus extends away from the protein giving it the appearance of a tail “wagging the dog” (Williamson, 1994).

The extended conformation of polyproline sequences is not the only factor which makes them effective scaffolding regions. Proline frequently plays a role in the binding mechanism of proteins due to its unique properties. One of these properties is that the backbone restrictions help to correctly orientate binding residues. However, proline can often form part of the protein-protein interaction directly (Murray *et al.*, 1994). Aromatic rings from substrate binding sites can stack with proline via the hydrophobic effect and π stacking (Pandey *et al.*, 2014).

Aromatic-proline interactions constitute the core binding of SH3 domains – domains often used to localise proteins to the membrane. SH3 binding usually involves neighbouring basic residues as may be the case for the RRGPAPPPPP polyproline regions which constitute two of the 300-422 G-actin-binding sites. It is perfectly reasonable to assume that SH3 binding to these tracts would influence the ability of Las17 to nucleate filaments *de novo*.

1.5.3 The potential role of SH3 domains in regulating Las17-mediated nucleation

The WCA and WH1 regions of Las17 can interact to physically block Arp2/3 binding and thus confer a degree of autoinhibition, albeit with lower potency when compared to other members of the WASp family (Rodal *et al.*, 2003). However, studies first investigating Las17 autoinhibition were conducted prior to the discovery of Arp2/3-independent nucleation. This suggests an additional regulatory mechanism may be utilised by yeast to prevent unwanted nucleation.

The four N-terminal polyproline tracts which constitute the Las17 minimum nucleation component are all known to bind the SH3 domains of proteins at the endocytic patch (Tonikian *et al.*, 2009). Physical occlusion of the actin-binding sites may therefore be responsible for controlling the switch between activity and inactivity. The tandem SH3 protein Sla1 is thought to be a major negative regulator of Las17 in the cytosol and has been shown to compete with G-actin to bind the Las17 polyproline region (Feliciano and Di Pietro, 2012). Further understanding of how the Las17-SH3 interactome affects Las17-nucleation may provide insight into how the invagination phase of CME is regulated. This bears resemblance to the G-actin competitive inhibition applied to APC via EB1 which effectively acts as a binding tandem due to homodimerisation (Juanes *et al.*, 2020) (mentioned in **section 1.3.4.3**).

1.6 Structure and role of SH3 domains

SH3 domains function as modular protein-protein binding units that help to localise protein function spatially and temporally via the localisation of their SH3-bound substrate. This is most apparent in membranous signal transduction and cytoskeletal remodelling where SH3 domains serve to target the activity proteins (Kay, Williamson and Sudol, 2000). Extended proline-rich scaffolds such as the polyproline region are often activated at these sites to recruit multiple SH3-domain associated proteins.

Binding occurs via a core proline motif (PxxP) flanked by residues which confer specificity. Type II polyproline helices repeat every three residues which orientates the core prolines facing the same direction (Kay, Williamson and Sudol, 2000). Small amino acid substitutions to these residues can completely change the interaction target which has provided organisms with an invaluable method of functional targeting. This is aided by the modularity of the SH3 domain which is often a ~60 residue, self-contained globular unit (Panni, Dente and Cesareni, 2002). Substituting SH3 domains between proteins does not necessarily impact on intrinsic protein function (although this is not always the case), but rather redirects that function to the site of the substituted partner. However, new research with swapped/shuffled SH3 domains *in vivo* suggests that protein context plays an underexplored, yet important role in SH3 specificity (Dionne *et al.*, 2021).

SH3 domain interactions are also mediated via a salt bridge which occurs via a basic residue close to the core PxxP motif. This residue can be located either three residues N-terminal (+xxPxxP) or two residues C-terminal (PxxPx+) and facilitates the interaction alongside the surrounding amino acid groups which further finetune the specificity (Kay, Williamson and Sudol, 2000; Tonikian *et al.*, 2009). Molecular dynamic simulations indicate these basic domains serve as “guiding” interactions to help to correctly orientate the SH3 which is an observed feature of IDP substrates (Ahmad and Helms, 2009; Williamson, 2023). SH3s which recognise the former sequence are termed class I domains, those which recognise the latter sequence are termed class II while some proteins can recognise both. However, other SH3 domains fall outside of these classifications. For example, some such as Myo5 do not require a flanking charged residue or share the core motif of other SH3s (although the binding sequences of this group still contain a high proline content) (Tonikian *et al.*, 2009). Others such as p53 can recognise a PxxP core motif via the tertiary structure of the substrate rather than its primary sequence meaning that the core prolines may be spaced far apart in the peptide sequence (Kay, Williamson and Sudol, 2000).

The versatility of SH3s is highlighted by their abundance, with budding yeast employing 28 individual domains and humans having around 300 (Kärkkäinen *et al.*, 2006). Nine SH3 proteins are used during the recruitment of proteins to the endocytic patch. This includes 11 individual domains which constitute over a third of all yeast SH3s, demonstrating the major role these domains play in endocytosis (Hummel and Kaksonen, 2023).

1.6.1 Sla1

Sla1 arrives at the endocytic patch alongside Las17 whilst also stably associating in the cytosol to form a large complex containing multiple Sla1 and Las17 polypeptides. This oligomeric interaction is termed the SLAC complex and may function to keep Las17 in an inactive state via masking of the 300-422 region (Feliciano and Di Pietro, 2012).

The two N-terminal SH3 domains of Sla1 (SH3#1 and SH3#2) are located directly adjacent in sequence and are both capable of binding Las17 with a combined affinity of 56 nM (**figure 1.9**) (Feliciano and Di Pietro, 2012). It is assumed that Las17 binding is also supported by a third SH3 domain (SH3#3) which is separated by a much larger linker of ~200 residues, though this contributes less to Las17 inhibition than the SH3#1-2 region (Rodal *et al.*, 2003).

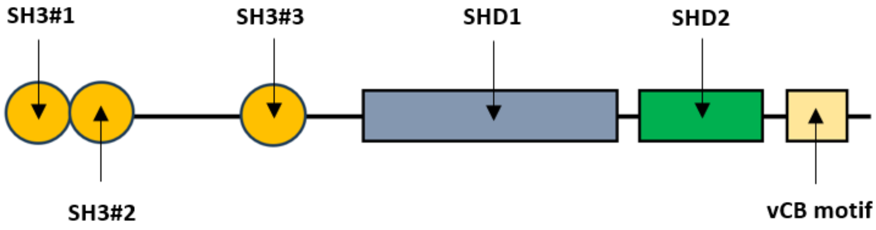


Figure 1.9: Structural regions of Sla1. The Las17-binding domains (SH3#1, SH3#2, and SH3#3) bind to the 300-422 region of Las17 and can physically occlude G-actin binding and thus inhibit nucleation. Their close spatial association effectively allows for tight binding. The SHD1, SHD2, and vCB motif are all protein binding regions, which help to recognise cargo and bind clathrin.

Sla1 plays a fundamental role in linking cargo recognition to clathrin recruitment during clathrin-mediated endocytosis, thus functioning as an adapter (Tolsma, Cuevas and Di Pietro, 2018). This is mediated via numerous protein-protein interaction domains located C-terminal of its SH3 domains, including an SHD1 domain, SHD2 domain, and a variant clathrin box (vCB) motif. Cargo can be recognised at either the SHD1 domain or by SH3#3, which bind the cargo sequence NPFX$_{(1,2)}$D and ubiquitin respectively (Howard *et al.*, 2002; Stamenova *et al.*, 2007).

Sla1 reduces the rate of actin polymerisation in the presence of Las17 and Arp2/3, while data from the Ayscough lab reveal that even low Sla1 concentrations can confer complete inhibition over Arp2/3 independent nucleation (Rodal *et al.*, 2003; J Palmer and E Allwood, personal communication). Furthermore, core Sla1 binding maps to the N-terminal, G-actin binding polyproline tracts. This interaction inhibits G-actin binding in a concentration-dependent manner indicative of competitive inhibition (Feliciano *et al.*, 2015). The above evidence leads to a model in which Sla1 association reduces the activity of Las17 by occluding the nucleation sites of the PP region from actin (**figure 1.10**).

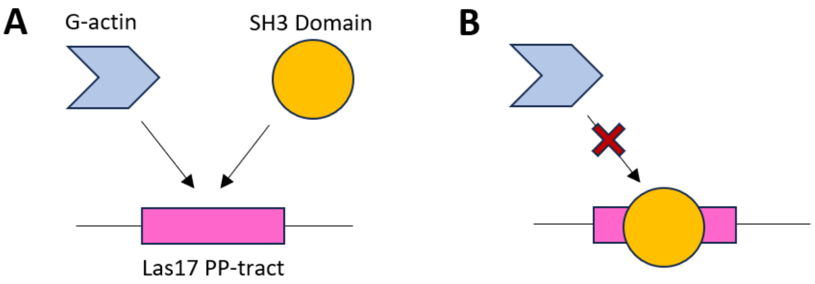


Figure 1.10: A visualisation of actin-SH3 competition. A) Both actin and SH3 domains can bind to Las17 polyproline tracts. The probability of both binding events are dependant upon the local concentrations and affinities. B) The binding is mutually exclusive as the binding of one agent (SH3 in the figure) masks access to the site for the other (actin the figure). Therefore, both actin and SH3s are in direct competition for Las17.

1.6.2 Ysc84

Another key endocytic protein which regulates Las17 function is Ysc84. This protein can bind Las17 via a single C-terminal SH3 domain and has been observed to affect actin filament dynamics using a membrane and actin binding C-terminal YAB domain. This domain has been shown to not bind pre-formed filaments suggesting a role in early filament formation (Robertson *et al.*, 2009) – possibly in supplying G-actin to the Las17 nucleation site. Investigating the actin binding ability of Ysc84 revealed a G-actin K_d of 0.85 µM while mutations aimed at reducing this affinity conferred a clear, deleterious effect to cells, highlighting the importance of this underexplored interaction (Urbanek *et al.*, 2015).

1.6.3 Bzz1

Bzz1 is a membrane-binding SH3 protein that is recruited to the endocytic patch later than Las17, Sla1, and Ysc84 with an arrival time shortly before the F-actin burst. Membrane binding (and dimerisation) occurs via an N-terminal F-BAR domain. Towards the N-terminus, Bzz1 can bind Las17 with two closely associated SH3 domains. The functions of these SH3s are not well characterised as the F-BAR domain may play a large role in Bzz1 recruitment – a role typically reserved for SH3 domains (Hummel and Kaksonen, 2023). Bzz1 is predicted to compete with Sla1 to bind Las17 suggesting that the tandem SH3 domains of Bzz1 may help to regulate Arp2/3-mediated endocytosis (Tonikian *et al.*, 2009).

1.6.4 The regulatory cloud hypothesis explored by this thesis

If the function of Las17 is to nucleate the mother filaments required by Arp2/3, then a glaring question remains – how could this mechanism be regulated? Competitive binding between Sla1-SH3s and actin has already been observed which may suggest that SH3 domains play a large role. Amazingly, Las17 is known to interact with all nine of the endocytic SH3 proteins demonstrating the pivotal role it plays in protein recruitment (Hummel and Kaksonen, 2023), as well as paving the possibility for an SH3 regulatory mechanism.

Sla1 arrives together with Las17 at the endocytic patch suggesting a role in Las17 regulation prior to its recruitment (**figure 1.11a**). Fluorescence microscopy aimed at analysing the dynamic composition of endocytic patches has suggested that Sla1 possesses a greater extent of inward mobility (i.e., the distance moved from the membrane plane during invagination) than Las17 which mostly remains at the base of invagination (Picco *et al.*, 2015). Conclusions drawn from this technique have proven contentious given variation between studies and the sites of tag-attachment (Goode, Eskin and Wendland, 2015). However, a more recent single-molecule localization microscopy study (a category of super resolution microscopy termed SMLM or STORM) supports this separation (Mund *et al.*, 2018). Sla1 is understood to associate with other patch proteins via its additional protein-binding domains which may help to localise Sla1 away from Las17. Indeed, mutating Sla1 to reduce clathrin-binding results in a delay between Arp2/3 and actin polymerisation (Tolsma, Cuevas and Di Pietro, 2018), possibly indicating increased Las17 inhibition through delayed Sla1 re-localisation.

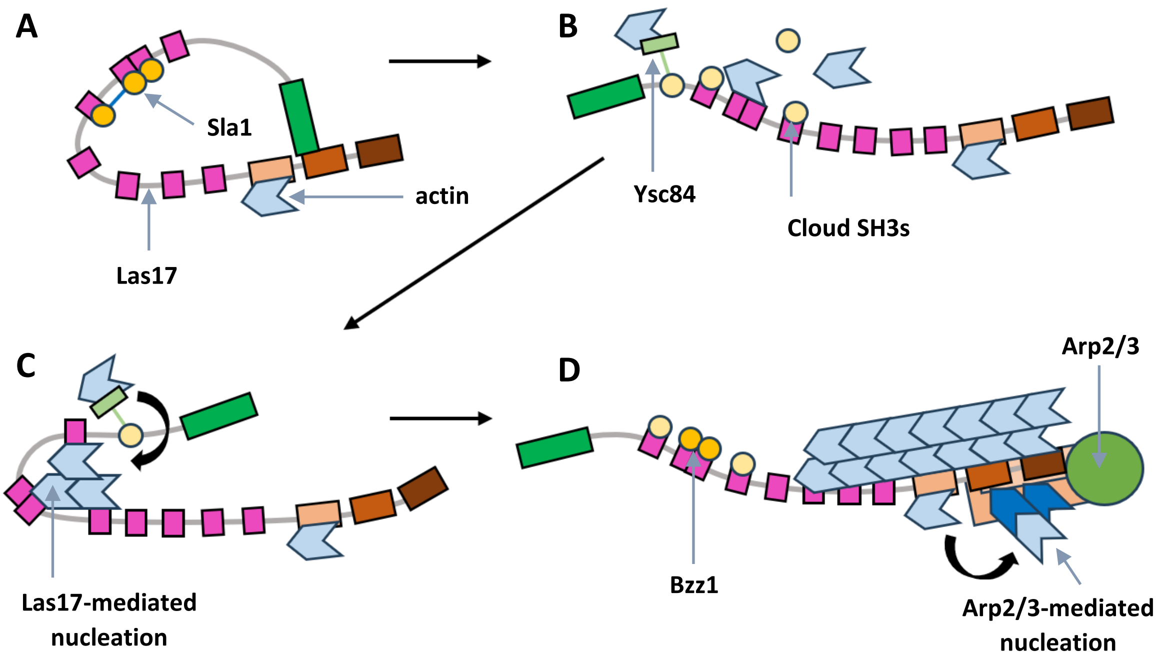


Figure 1.11: The regulatory cloud hypothesis. Actin subunits/monomers are shown as light blue arrow heads with the indent representing the barbed end (hydrophobic cleft). Las17 is represented using the same colour scheme as **figure 1.8** with actin-binding tracts from the polyproline region being shown as magenta squares. Tandem SH3 domains are shown as dark yellow circles, single SH3 domains are shown as tan circles, Ysc84 YAB domain as a light green rectangle and Arp2/3 is shown with the same colour scheme as **figure 4**. **A**) Las17 arrives at the endocytic patch in an inhibited state. This includes self-inhibition of the WCA region and Sla1 inhibition of Las17-mediated nucleation. **B**) Single domain SH3s from the endocytic SH3 cloud help to outcompete Sla1 which is localised more towards the centre of the patch using its cargo-binding domains. These cargo-binding interactions may help to spatially sequester the SH3 domains from Las17 and thereby facilitate the alleviation of Sla1-mediated inhibition. Removal of Sla1 allows actin to bind to the polyproline region with single SH3 domains possibly helping to control the activity level. The WH1 domain interacts with the plasma membrane releasing inhibition of the WCA. **C**) Actin is successfully nucleated by Las17 to form mother filaments. Ysc84 may enhance the subsequent polymerisation by supplying G-actin via its YAB domain. **D**) Inhibition is re-applied as more cloud SH3s are recruited including the tandem-binding Bzz1. F-actin binding, C-terminal Las17 tracts interact with the mother filament to prevent its dissociation from the patch and render it available to Arp2/3. Arp2/3 can then bind to the WCA region and mother filament to nucleate a branched network of actin filaments.

This thesis will investigate a possible regulatory hypothesis governed by the numerous SH3 domains dynamically present at the patch (**figure 1.11**). We can think of this as a "dynamic cloud of SH3s" which evolves as more proteins are recruited and re-localised. Competitive binding between SH3 domains may help to control the activity of Las17 culminating in nucleation while Ysc84 provides additional G-actin monomers (**figure 1.11b** and **figure 1.11c**). Nucleating activity may be progressively reduced as more SH3 domains are recruited to the patch until the tandem SH3s of Bzz1 (visually reminiscent of Sla1) reimpose the stronger, high avidity inhibition of Las17 nucleation (**figure 1.11d**). SMLM reveals that Las17 and Bzz1 share an almost identical spatial distribution in the patch (Mund *et al.*, 2018).

The Las17 binding site of Bzz1 is known to overlap with Sla1 which would result in binding competition. Therefore, it would follow that the competitive inhibition of actin imposed by Sla1 would be maintained regardless of which of these two proteins is bound. Bzz1 is described as enhancing Las17-Arp2/3 polymerisation which may contradict this (Sun, Martin and Drubin, 2006). However, data from the Ayscough lab reveals that Bzz1 SH3 domains inhibit Las17 nucleation *in vitro* (K Ayscough, personal communication). Furthermore, the F-BAR domains would effectively dimerise the WCA region of Las17 which is known to enhance Arp2/3 activity (Padrick *et al.*, 2008). A possible *in vivo* explanation is that Bzz1 inhibits Arp2/3-independent activity through competitive polyproline binding, whilst simultaneously promoting Arp2/3-dependent activity through promoting Las17-WCA dimerisation.

1.6.5 Fuzzy binding and the persistence of tandems

A cursory glance at the proteins mentioned thus far reveals a persistent theme – tandem binding. The core 300-422 nucleation region of Las17 contains four polyproline tracts while Sla1 possesses three Las17-binding SH3s, and Bzz1 possesses two. Why would a regulatory system evolve so many tandem binding domains as opposed to relying on fewer interactions?

One emerging view is that of "fuzzy binding" where it can sometimes prove advantageous for protein systems to utilise multiple weak interactions to facilitate an interaction (Williamson, 2023). This is because the relationship between the association rate (k_{on}) and dissociation rate (k_{off}) is well defined for a given affinity (K_{d}) as shown in **equation 1.1**.

$$K_d = \frac{k_{off}}{k_{on}} \tag{1.1}$$

A theoretical k_{off} rate calculated for high affinity interactions may be so low that dissociation occurs on a minute timeframe (Williamson, 2023). In the context of endocytosis, this may result in Sla1 dissociating from Las17 long after the mother filaments should ideally have been nucleated. This is particularly important for IDP substrates such as Las17 which are typically associated with higher k_{on} rates (Mollica *et al.*, 2016). For a strong, single interaction, these high rates would need be balanced by slow k_{off} rates (i.e., to satisfy **equation 1.1**).

Binding through multiple tandem interactions solves this issue by allowing the overall affinity to be constituted from multiple, weaker affinities and thus permit smaller individual k_{off} rates. The binding of one weak interaction would increase the effective concentration of its adjacent binding domain(s) which results in individually weak affinities combining to confer a much stronger overall affinity of the complex. This is called avidity (Williamson, 2023). While the avidity associated with tandem binding can allow for a much higher overall affinity in theory, the exact nature of how individual domains combine in effect would be heavily dependent upon their geometries. This makes the calculation of overall affinities from the constituent interactions challenging. One of our thesis aims was to acquire binding affinities for both tandem and individual SH3 domains.

Avidity also allows biological systems to employ a much more rapid complex dissociation (Williamson, 2023). For example, if one of the two binding interactions in a tandem substrate was abolished through phosphorylation, then the complex must rely solely on the remaining binding site. Each of the tandem binding sites is individually weak resulting in rapid dissociation. Multiple weak interactions also permit greater access to regulatory factors such as kinases and competitive binders as each binding site has a high rate of change between bound and unbound states. Theoretically, domains from the endocytic SH3 cloud could facilitate the dissociation of Sla1 from Las17 by competing with the tandem binding sites in much the same way.

Taken together, the persistence of multiple tandem binding sites may enhance the dynamic aspects of regulation – clearly important when considering a rapidly evolving endocytic patch – by promoting high avidity multi-binding, as opposed to high affinity single-binding interactions. This may allow Sla1 and Bzz1 to maintain robust inhibition of Las17-mediated nucleation without the need for excessively low dissociation rates which would otherwise extend the time required for switching between Las17 activity states. In a broader context, this may be how the SH3 cloud functions – by regulating actin access through the weak "fuzzy" interactions of many localised SH3s rather than a single, dedicated inhibition domain to preserve fast reactivity of the system.

1.7 Yeast as a model organism

Yeast has long been used as an effective model system for understanding human cellular biology. Yeast is one of the simplest eukaryotic organisms with a rapid growth rate and high number of paralogues (Botstein, Chervitz and Cherry, 1997). Many parities exist between these including the tandem SH3 domains of Nck bearing a striking resemblance to Sla1, and Las17 sharing a high degree of structural and functional similarity to its human WASp family paralogues (Chaki and Rivera, 2013).

Yeast two-hybrid assays between actin and the polyproline region of human WASp (300-430) reveal a positive interaction suggesting some functional conservation of the PP-region. Subsequent polymerisation assays failed to identify a nucleating effect (Urbanek *et al.*, 2013). However, studies have noted that this region of WASp can enhance polymerisation (Bieling *et al.*, 2018). Furthermore, a recent preprint article suggests that SCAR/WAVE (a member of the WASp family), was capable of driving *Dictyostelium* lamellipodia protrusions in the absence of Arp2/3. The group determined that the polyproline region of SCAR/WAVE was required for this phenomenon and theorised that the region may play an uncharacterised role in actin nucleation (Buracco *et al.*, 2022).

Ultimately, expanding our knowledge of Las17-mediated nucleation in yeast may aid the understanding of other WASp family proteins given our limited understanding of this novel mechanism. Additional characterisation may also help uncover new actin binding regions within the proteome and further our understanding of how cells utilise “fuzzy binding” – both of which are useful from a human biology perspective.

1.8 Taking a hybrid computational approach

Actin machinery is highly conserved and shared between multiple cytoskeletal mechanisms as explained throughout **section 1**. Shared functions make *in vivo* study challenging due to collateral effects that mutations may have on other, highly sensitive systems. This also complicates *in vitro* studies which may behave differently without protein context provided by non-expressed domains or protein-interaction partners (Dionne *et al.*, 2021). Endocytosis involves not only the cytoskeleton, but also membranes.

Membrane-binding proteins are difficult to purify and analyse *in vitro* as removal from this context can result in protein instability (Carpenter *et al.*, 2008). This has ramifications throughout the entire experimental process. IDPs (frequently proline rich) are also notoriously challenging to work with (Necci *et al.*, 2021). Las17 belongs to both groups and interacts with nine distinct SH3 membrane proteins rendering a purely experimental approach extremely challenging. A hybrid computational approach would allow the exploration of a wide range of scenarios and focus experimental work onto key areas. This thesis therefore uses this approach: computer modelling of the interactions, based on experimentally determined parameters and compared to experimental observations of endocytosis in yeast and *in vitro* assays.

1.8.1 History of modelling

Mathematical modelling has long aided scientists to rationalise observations and form experimental predictions to aid in development of a testable model. The interwoven nature of these approaches can be traced back more than one hundred years with a classic example being development of the Michaelis–Menten model (Johnson and Goody, 2011). With the advancement of technology, computational modelling has risen to aid in this area. It can now be seen everywhere from calculating early actin nucleation rates (Sept and McCammon, 2001), to solving protein structure (Jumper *et al.*, 2021). Modelling can range through a variety of forms. Molecular dynamic (MD) simulations are often grounded firmly in physics with actions being executed following force and energy calculations (Karplus and McCammon, 2002). MD simulations can provide valuable insight into specific protein-protein interactions such as the interactions of Arp2/3 (Zhang and Vavylonis, 2023). However, MD is computationally intensive, and is not feasible for the system size and timescales required here. On the other hand, pure ODE (ordinary differential equation) modelling is a useful tool for explaining data patterns (Torres and Santos, 2015). Ultimately, the type of modelling method used depends upon the experimental/"real-world" information available as well as the questions being asked.

1.8.2 Justification for Agent-based modelling

Agent based modelling (ABM) is a powerful tool used to explore a variety of biological questions. This method allows systems to be modelled through simulating numerous molecular 'agents' and defining how they interact with one another and the environment. These agents can represent anything from individual molecules to elements in a financial market (Azimi, Jamali and Mofrad, 2011).

A basic description of ABM can be given as follows: a number of discrete objects are defined that are allowed to interact via a set of pre-established rules across a series of discrete time steps. This differs from other approaches such as Markov Chain Monte Carlo (MCMC) simulations which are non-discrete (Speagle, 2019). Essentially, this approach aims to simulate a "virtual world" to represent a given system for the purpose of exploring said system. ABMs are stochastic non-inductive models that behave through a set of given rules (i.e., agents of type A will move towards the closest agent of type B). This means they are not reliant upon data patterns unlike many other (Azimi, Jamali and Mofrad, 2011; Torres and Santos, 2015). Rather, they are more like an "*in silico* playground" used to investigate the behaviour of hypothesised mechanisms based on a set of simple rules, which in turn provides useful insights into their veracity.

Space is often a key component of agent-based models owing to its origins in 2D grid systems. ABMs were quickly developed to incorporate 3-dimensional continuous (i.e., non-grid based) movement, allowing spatiotemporal behaviour to be represented within the system (An, 2008). Their modular nature allows agent types to be easily modified to represent a large variety of system elements (often with changes to variable values rather than changes to the agent structure) (Azimi, Jamali and Mofrad, 2011). Endocytosis uses a large range of proteins to confer its spatiotemporal control (Goode, Eskin and Wendland, 2015) lending itself perfectly to the application of agent-based modelling.

ABMs can be built without the need for extensive quantitative data and mathematical knowledge, so long as the rules imposed upon the system are logical. Intrinsic stochasticity and locally constrained agent rules combine to allow the manifestation of emergent phenomena. These are defined as system behaviours that could not have been reasonably inferred prior to executing the model and therefore can prove invaluable to experimentalists (Bonabeau, 2002). This is why the method remains a powerful tool in aiding the exploration of poorly understood systems. It all depends upon what questions the modeller wishes to extract from the method. ABM is best suited for asking qualitative questions regarding mechanistic veracity (i.e., will many weakly competing interactions accelerate the unbinding of a tandem domain) and to use the results (along with predictions of any emergent phenomena) to guide experiments (An *et al.*, 2009). It is for these reasons that agent-based modelling was chosen as the key method for this thesis.

1.8.3 Examples of relevant ABM computational models

Agent based models are highly applicable to membranous systems, such endocytosis and signal transduction, due to their inherent spatiotemporal nature. Examples of this application have been growing as its uses are becoming more apparent. ABM has been used in great effect during the study of vesicular trafficking (Birbaumer and Schweitzer, 2011; Klann, Koeppl and Reuss, 2012) and cytoskeletal organisation such as filament interactions (Das *et al.*, 2020; Sabirov and Spirov, 2020). These often take a more abstracted view of a cell or system and can provide enlightening insights into the spatial/structural organisation of these large structures. Agent-based modelling is also beginning to play a more prominent role in helping us understand actin patch dynamics. Advances in super resolution microscopy are driving the development of ever more detailed models which reveal just how related patch organisation and actin dynamics truly are (Mund *et al.*, 2018; Akamatsu *et al.*, 2020). However, existing ABMs encompassing endocytosis and the cytoskeleton are often too coarse-grained for direct application to this thesis and tend to focus on larger phenomena such as force generation or cytoskeletal structure.

Signalling pathways also have a history of applying ABMs and tend to focus more on individual protein-protein events. However, these models do not provide adequate solutions for multi-domain tandem binding and lack nucleation/polymerisation dynamics (Fullstone *et al.*, 2020). Therefore, we decided that developing an in-house agent-based model to fulfil this role may not only benefit our understanding of Las17 nucleation but may also be of interest to the broader modelling community.

1.8.4 The rise of GPU computing

The modular nature of agent-based modelling and its reliance on discrete objects lends the method well to parallel computing. Parallel computing allows multiple agents to undergo a discrete function simultaneously which significantly decreases computing time. GPU computing offers a powerful resource for executing parallelism as GPUs are constructed from thousands of relatively simple processor cores (at least compared to most commercial CPUs). Each core can be assigned a discrete threat of code (e.g., executing the movement function for agent X) which allows for a greater degree of parallelism when compared to historically dominant CPU computing (Lee *et al.*, 2010).

GPUs play a vital role in video games by allowing the calculations for each pixel to be simultaneously calculated during each frame. This is vital for achieving both high visual fidelity and smooth framerates – key parameters for satisfying both gamers and the esports scene. The gaming industry was valued at $250 billion for 2023 creating a massive economic driver for technological innovation. Comparing gameplay for a game released in 2010 with one released in 2020 shows the massive leap that has occurred in GPU computing over this last decade. With the video game market projected to reach $470 billion by 2026, this development is expected to continue (GlobalData, 2022).

Therefore, it comes as no surprise that growth in consumer-led GPU development is correlated with a rise in GPU-computing across the sciences (Carreño and Howes, 2018). The calculations which GPUs are specialised to perform are highly comparable to those employed in scientific parallel programming. GPU cores can discretely calculate individual agent functions the same way they can calculate individual pixels on a screen (Carreño and Howes, 2018). This GPU revolution in science will likely expand as the economic drivers persist, opening exciting avenues for computational modellers.

1.9 Aims and objectives

The hypothesis posed within this thesis is that Las17 nucleation is regulated by a network of numerous weak SH3 interactions (often avidity controlled) which provide robust inhibition over the Las17 polyproline region (**figure 1.10**). The developing cloud of single SH3 domains help to outcompete the tandem-binding Sla1 domains and later re-impose tandem inhibition following the recruitment of Bzz1. This gives a well-regulated activity window between Sla1 departure and Bzz1 arrival for mother seeds to be nucleated ahead of Arp2/3 arrival.

I aim to address the following:

- To construct a protein-level agent-based model that can adequately account for multi-domain proteins whilst also incorporating actin nucleation and polymerisation dynamics.
- Further experimentally characterise the Las17 nucleation and regulatory mechanism through obtaining key binding affinities
- Determine whether simulated polymerisation rates are broadly consistent with experimental measurements to identify elements that are missing from our Las17-mediated nucleation model.
- Answer the question – why so many tandem binding sites?
- Offer further insight into the relationship between Sla1 and Las17 using the AMB constructed within this thesis.
- Identify how the SH3 cloud interacts with Sla1 and use this to construct an updated model for nucleation and Las17 regulation with a focus on experiment collaboration.

2. Materials and Methods

2.1 Media and Buffer list

Buffer	*Composition*
2xTY	1.6% Tryptone 1% Yeast Extract 0.5% NaCl
DNA loading buffer	0.25% Bromophenol Blue 0.25% xylene Cyanol FF (Sigma) 30% Glycerol
High Salt PBS	1 x PBS 1% Tween 300 mM NaCl
High salt cleavage buffer	150 mM NaCl 50 mM Tris pH 7.0 1 mM EDTA 1 mM DTT
Low salt cleavage buffer	50 mM Tris pH 7.0 50 mM NaCl 1 mM EDTA 1 mM DTT
G-buffer	2 mM Tris pH 8.0 0.5 mM DTT (added <1 hour before use) 0.2 mM ATP (added <1 hour before use) 0.2 mM $CaCl_2$
His binding buffer	20 mM imidazole pH 7.4 500 mM NaCl 20 mM Na_2HPO_4
His wash buffers	40/60/80/100 mM imidazole pH 7.4 500 mM NaCl 20 mM Na_2HPO_4
His elution buffer	500 mM imidazole pH 7.4

	500 mM NaCl
	20 mM Na_2HPO_4
1x KME (10x stock used)	10mM Tris-HCl, pH 8.0
	50 mM KCL
	1 mM $MgCl_2$
	1 mM EGTA
MST buffer	50 mM Tris pH 7.4
	150 mM NaCl
	10 mM MgCl2
	0.05 % Tween-20
PBS	0.8% NaCl
	0.144% Na_2HPO_4 pH 7.4 (pH adjusted with HCl)
	0.02% KCl
Resolving Buffer	1.5 M Tris-HCl pH 8.8
	0.4% SDS
2x SDS PAGE gel loading buffer	50 mM Tris-HCl
	384 mM glycine
	0.2% SDS pH 8.6
SDS running buffer	192 mM glycine
	25 mM Tris
	1% SDS
Stacking Gel Buffer	0.5M Tris-HCl pH 6.8
	0.4% SDS
TAE buffer	40mM Tris
	20mM acetic acid
	1mM EDTA pH 8.3

2.2 Plasmid, peptides, and protein construct List

Expression plasmids used within this thesis:

Lab ID number (pKA)	Description	Origin/Reference
566	GST-Las17 (300-422)	Ayscough lab
928	Sla1-SH3#1 (3-68)	Ayscough lab
1247	Sla1-SH3#1-2 (5-131)	Ayscough lab
1280	His-Las17 (300-422)	Ayscough lab
748	Ysc4-SH3	Ayscough lab
947	Sla1-SH3#2 (69-132)	Cesareni lab
1339	Sla1-SH3#1-2 (5-131) W108A	Ayscough lab (generated as part of this thesis)
1189	GST-Las17 (300-422) RR(350,351)AA	Ayscough lab
1190	GST-Las17 (300-422) RR(389,390)AA	Ayscough lab
1191	GST-Las17 (300-422) RR(350,351)AA and RR(389,390)AA	Ayscough lab
1337	GST-Las17 (300-422) RR(3190,321)AA, RR(350,351)AA, and RR(389,390)AA	Ayscough lab

Bacterial strains used within this thesis:

E. coli **Strain**	**Origin/Reference**
Rosetta	KA lab
BL21	KA lab
DH5α	KA lab

2.3 Molecular biotechnology techniques

2.3.1 BL21 and DH5α *E. coli* transformation

BL21 cells are an *E. coli* strain engineered for protein expression whilst DH5α cells are an *E. coli* strain engineered for a high transformation efficiency. As a result, they were used in protein purification and plasmid purification respectively. The transformation protocol for both strains is identical. 1 µL of purified plasmid was pipetted into 20 µL of competent cells (of the appropriate strain) and left on ice for 30 minutes. The cells then underwent a heat shock treatment consisting of 30 seconds at 37^{o}C followed by 120 seconds on ice. Recovery was performed by mixing the cells with 300 µL of 2xYT media and incubating for 40-60 minutes in a 37^{o}C shaking incubator. Cells were spun at 3,000xg using a centrifuge, 200 µL of the supernatant was removed and the cell pellet resuspended. This suspension was then transferred onto a 2xYT agar plate containing the appropriate antibiotic for transformant selection. Ampicillin was the selection antibiotic for all expression constructs used throughout this thesis.

2.3.2 Rosetta *E. coli* transformation

Rosetta cells are an *E. coli* strain engineered for protein expression of eukaryotic peptides as they contain tRNAs capable of reading rare eukaryotic codons. Transformation of Rosetta cells follows an almost identical protocol to BL21 and DH5α (**section 2.3.1**) with the additional step of adding chloramphenicol to the agar plates along with the transformant selection antibiotic. This is because the tRNAs are located on a chloramphenicol-resistant plasmid and thus dual antibiotic selection helps identify colonies which contain both the eukaryotic codon tRNAs and expression construct.

2.3.3 DNA Electrophoresis

Samples of DNA were combined with DNA loading buffer in a 5:1 ratio. These sample mixtures were then loaded onto a 1% agarose gel (500 µL agarose, 2 µl of 10 mg/ml Ethidium Bromide, and 49.5 mL TAE buffer) and run at 80 V until the dye front begins to exit the gel. The resultant gel was transferred to a gel documentation system (Bio Rad) for imaging.

2.3.4 Point mutations using PCR

PCRs were performed using a BIOTAQ DNA Polymerase kit from Meridian Bioscience. In summary, this involved mixing 1 µL template DNA, 1 µL forward primer, 1 µL reverse primer, 0.5 µL 100 mM dNTP's, 1x PCR Reaction Buffer, 1 µM BioTaq Polymerase (5 U/µl), and 3 µL 50 mM $MgCl_2$ up to a volume of 50 µL with deionised water. The primers are designed to overlap with part of the target expression cassette and contain the desired mutation. Mixtures were placed in a thermal cycler from Biometra (T-Gradient Thermoblock), denatured for 60 seconds at 94° and run for 20-30 iterations with the following program: denature for 15 seconds at 94°C, primer anneal for 15 seconds at 55°C, extension at 72°C for 30 seconds per kilobase of DNA. Following PCR, samples of the resultant PCR product were cleaved using two restriction enzymes and visualised on a DNA gel using the method given in **section 2.3.1**. If appropriately sized bands were visible, the PCR product was purified as described in following section.

2.3.5 Plasmid purification following PCR

Competent DH5α cells were transformed with the PCR product produced in **section 2.3.4** using **method 2.3.1**. Five colonies were each transferred from the plate into 5 mL of 2xYT media containing the appropriate selection antibiotic (this was ampicillin for all constructs generated within this thesis) and grown overnight in a 37^{o}C shaking incubator. These five overnight cultures were then individually purified using the PureLink Quick Plasmid Miniprep from Invitrogen. Cells from the culture were lysed to release their DNA. This was followed by lysate precipitation and removal, transfer to the DNA-binding miniprep Spin Column, washing, and elution using deionised water. Plasmid concentration was determined using a nanodrop machine. Samples from each of the five purified plasmids were sent to **dnaseq.co.uk** for sequencing. The purification which best matched the target sequence was used to make a glycerol stock.

2.3.6 Glycerol stock

Glycerol stocks are DH5α cells transformed with a plasmid and stored in glycerol at -80^{0}C. This generates a backup of the plasmid that can be used should the purified plasmid stock be lost. Glycerol stocks were made by combining 500 µL transformed bacteria (grown overnight in a 37^{o}C shaking incubator) with 500 µL of sterilised 25% glycerol.

2.4 Protein production and purification

2.4.1 Production of an expression cell pellet

Transformations were undertaken using the appropriate expression plasmid. BL21 cells were used for most peptides (**section 2.3.1**), whilst Rosetta cells were used for Las17-encoding plasmids (**section 2.3.2**). Following incubation overnight at 37°C, transformant colonies were scraped up from agar plates and transferred into 2xYT media along with 1 mM of the appropriate antibiotic(s). These cells were grown for 7 hours in a 37°C orbital shaking incubator before their induction with 1 mM IPTG. The temperature of the shaking incubator was reduced to 17°C post-induction, and the bacterial flasks were incubated overnight. The resultant cultures were pelleted in the morning at 5,000g for 20 minutes.

2.4.2 Lysis of an expression cell pellet

Bacterial cell pellets produced using **method 2.3.1** were thoroughly resuspended in 15 mL PBS buffer using a Dounce homogeniser. 0.6 mL of 25x protease inhibitor cocktail (cOmplete™, EDTA-free Protease Inhibitor Cocktail from Roche), 25 µL leupeptin (10 mM), 10 µL 2.3 µM lysozyme (0.62 mM) and 15µL DTT (1 M) were all added to the resuspension. The cells were sonicated which consisted of 5 x 30 second blasts at 15 µm amplitude with each blast being separated by 30 seconds of no sonication to prevent overheating of the cells. Cellular debris was removed by centrifugation at 20,000xg for 30 minutes. The entire process was carried out on ice where possible while the centrifuge maintained a temperature of 4°C.

2.4.3 Buffer exchange

Buffer exchange involved moving a sample from its original buffer to a new buffer. It began by placing the sample onto a gel filtration column pre-equilibrated with the new buffer (column acquired from GE Healthcare). 1 mL of new buffer was added to the column after the sample had been absorbed and this was immediately followed by fraction collection. Fractions were collected in volumes of 400-600 µL and tested for protein by pipetting 2 µL of each fraction into 50 µL Bradford reagent. The fractions containing the greatest concentration of protein were combined as indicated by the strongest blue coloration of Bradford.

2.4.4 GST purification

A lysate supernatant of the appropriate GST-fusion peptide (produced using **method 2.4.2**) was combined with 300 µL of PBS-washed glutathione beads (GE Healthcare) and incubated in a 4^{o}C cold room for 1 hour on a roller. The beads were then pelleted via centrifugation at 1522xg for 1 minute and washed with 3 x 10 mL high salt PBS, 3 x 10 mL PBS, 2 x high salt cleavage buffer and 2x low salt cleavage buffer (buffer composition detailed in **section 2.1**). Each wash was carried out in three stages including: completely re-suspending the pellets in the appropriate wash buffer, pelleting the beads via centrifugation at 1522xg for 1 minute, and discarding the supernatant. The entire process was carried out on ice where possible except where the protein was placed in the cold room

Cleavage of protein from the beads was achieved by re-suspending the washed beads in 300 µL low salt cleavage buffer and adding either 10 µL precision protease for pGEX 6P-1 expressed protein or 10 µL thrombin for pGEX 4T expressed protein. The resultant solution was incubated for 8 hours in a 4^{o}C cold room on a spinner wheel. To elute non-cleaved GST-protein, the washed beads were instead resuspended in 20 mM reduced glutathione (pH 8.0) and incubated for 1 hour in a 4^{o}C cold room on a spinner wheel. After either cleavage or elution, the beads were pelleted at 1522g for 1 minute and the supernatant was transferred into a fresh eppendorf. This bead pellet was discarded, and the purified protein from the supernatant was exchanged into the desired buffer using **method 2.4.3**.

2.4.5 His-tag purification

A 5mL Luer lock syringe was attached to a 1 mL nickel column (Cytiva HisTrap HP Prepacked Columns from Fisher), pre-equilibrated with His binding buffer via drop-to-drop contact. A lysate supernatant of the appropriate His-fusion peptide (produced using **method 2.4.2**) was transferred into the syringe and pushed through the column. This was followed by 5 mL of His binding buffer. The column was then sequentially washed with 5 mL of 40 mM, 60 mM, 80 mM, and 100 mM imidazole His wash buffer. In cases where peptides were entirely lost during the washes, higher imidazole molarity washes were omitted. Therefore, Las17 constructs were only washed with 40 mM and 60 mM imidazole His wash buffers as the Ayscough group had previously noted substantial Las17 elution during the washes.

To elute the bound protein, 5 mL of His elution buffer was pushed through the column. Fractions of 500 µL were collected from the point of His elution buffer addition. These fractions were tested for protein by pipetting 2 µL of each fraction into 50 µL Bradford reagent. Fractions containing the greatest concentration of protein were combined as indicated by the strongest blue coloration of Bradford. The resultant purification product was exchanged into the desired buffer using **method 2.4.3**. The entire process was carried out on ice where possible and drop-to-drop contacts were used each time the syringe and nickel column were attached to prevent any air bubbles from entering the column.

2.4.6 Size exclusion chromatography (SEC)

A column gel filtration column (SuperdexTM 200 HR 10/30 from Pharmacia Biotech), pre-equilibrated in degassed PBS buffer, was used to separate the proteins. The protein to be subjected to SEC was injected into the loading loop. Degassed PBS was then run through the column and fraction sizes of 0.5 mL were immediately collected. The four fractions with the highest concentration of desired protein were combined and concentrated using a centrifugal concentrator (product number VS0602 from Sartorius Stedim Biotech). The peak fractions for all proteins except Las17 were identified using a UV detector which recorded the absorbance at 280 nm. Las17 does not contain any tryptophan or tyrosine residues so every second fraction was tested using a Coomassie gel until the peak volumes could be identified. A weak peak can be identified using the A280 reading, however gel verification was important as any impurities in the sample (e.g., Hsp70) may result in false identification of Las17.

2.4.7 Removal of chaperones

Chaperone removal followed the protocol detailed in (Rohman and Harrison-Lavoie, 2000). The volume of induced and non-induced cultures differed from the protocol (2 L for each). This was accommodated by changing the volumes used in later steps to maintain the proportions. Acetone was the method of choice for concentrating the non-induced lysate. An overview of the method involves incubating the lysate supernatant of the desired His-labelled protein (**section 2.2.2**) with ATP, $MgCl_2$, KCl, and DTT for 10 minutes at 37^oC. Denatured BSA was added, and the solution was incubated for a further 20 minute at 37^oC. The resultant product was then purified using the His-tag purification method shown in **section 2.4.5**. ATP, $MgCl_2$, KCl and DTT were added to all the wash buffers.

2.4.8 Making 15% polyacrylamide gels

Polyacrylamide gels were made using the mixtures described in (Sambrook, Fritsch and Maniatis, 1989 – tables 18.3 and 18.4). In summary, two gel mixtures were made. The first is resolving gel made up of 5 mL 30% acrylamide 0.8% bis-acrylamide (37.5:1), 3.75 mL Resolving Buffer, 100 µL 10% ammonium persulphate (APS), 5 µL TEMED, and 1.2 mL water. The second is stacking gel made up of 0.8 mL 30% acrylamide, 0.8% bis-acrylamide (37.5:1), 0.6 mL stacking buffer, 50 µL 10% ammonium persulphate (APS), 15 µL TEMED, and 3.5 mL water.

A front and back gel plate were thoroughly cleaned and clamped together. 4-4.5 mL of resolving gel was poured into gap between the plates followed by 0.5 mL of isopropanol to level out the surface and remove any bubbles. After the gel set, the isopropanol was removed, 1.5-2 mL of stacking gel was poured into the plates and a gel comb was inserted. The gels were wrapped in wet tissue paper and catering film to stop them from drying out, and stored in the fridge after the gel had fully set.

2.4.9 Protein separation by SDS PAGE

Protein samples were mixed with an equal volume of 2 x SDS PAGE gel loading buffer (50 mM Tris-HCl, 384 mM glycine, 0.2% SDS pH 8.6) and boiled at 100^{o}C for 5 minutes. SDS running buffer was added to the gel electrophoresis setup, boiled samples were loaded into the gel wells (see **section 2.4.8**) and 130 and 160 V were applied until dye front approached the end of the gel. The front and back plates of the gel were then removed and the polyacrylamide gel inside was transferred into a pot for staining.

2.4.10 Coomassie staining

The protein gel was first run using **method 2.4.9**. The finished gel was washed in Coomassie Brilliant Blue stain solution (0.2% Coomassie Brilliant Blue R250, 40% methanol, 10% acetic acid). The gel was left on a rocker at room temperature for 1-2 hours. Destaining was achieved through replacing the stain solution for de-stain solution (40% methanol, 10% acetic acid) and leaving the gel for 3 hours. The de-stain solution was replaced several times throughout this process. A gel documentation system (Bio Rad) was used for imaging.

2.4.11 Silver staining

The protein gel was first run using **method 2.4.9**. The finished gel was washed 3 x 20 minutes in fixing solution (40% methanol, 10% acetic acid). The gel was then washed for 20 minutes in silver mix (0.36% $AgNO_3$, 0.076% NaOH, 70 mM NH_4OH) and washed 2 x 10 minutes in dH_2O. The water was replaced with developer solution (0.02% formaldehyde, 0.005% citric acid) and the container/gel was shaken manually until protein bands were visible on the gel. Staining was halted by replacing the developer solution with stop solution (50% methanol, 10% glacial acetic acid). The plastic container and gel were maintained on a rocker and a gel documentation system (Bio Rad) was used for imaging.

2.4.12 SYPRO Ruby^{TM} fluorescence staining

The protein gel was first run using **method 2.4.9**. Staining was carried out according the “Basic Protocol” detailed in the manufacturer’s instructions (catalogue number S12000 from ThermoFisher Scientific). The basics of the technique involved fixing the gel in fixing solution

(40% methanol, 10% acetic acid) for 2 x 30 minutes, incubating the gel with the SYPRO RubyTM fluorescence stain overnight and washing the gel with 100 mL of wash solution (10% methanol, 7% acetic acid) for 30 minutes. The plastic container and gel were maintained on a rocker and a gel documentation system (Bio Rad) was used for imaging.

2.4.13 Turmeric staining

The protein gel was first run using **method 2.4.9**. Staining was carried out according to the method detailed in (Kurien, Dorri and Scofield, 2012). Turmeric was acquired from a local food store. The basics of the turmeric preparation involved boiling the turmeric for 10 minutes, centrifuging twice at 1,800 x g to pellet unwanted material (20 minutes) and transferring the supernatant into a fresh tube. The basics of the staining involved fixing the gel in fixing solution (40% methanol, 10% acetic acid) for 20 minutes, rinsing the gel twice with water and incubating the gel with the turmeric supernatant for 30 minutes.

2.4.14 InVision^{TM} His-tag In-gel staining

The protein gel was first run using **method 2.4.9**. Staining was carried out according to protocol detailed in the manufacturer's instructions (catalogue number LC6030 from ThermoFisher Scientific). The basics of the technique involved fixing the gel in fixing solution (40% methanol, 10% acetic acid) for 1 hour, washing the gel with water for 2 x 10 minutes, incubating the gel with the InVisionTM His-tag In-gel for 1 hour and washing the gel with 20 mM phosphate buffer, pH 7.8 for 2 x 10 minutes.

2.4.15 Zinc-Imidazole negative staining

The staining protocol was modified from (Castellanos-Serra and Hardy, 2006). The differences included using zinc sulphate (pH 6.5) in place of zinc chloride and bringing the pH of the imidazole to 7.4 before application to the gel. Gels were first run using **method 2.4.9**. The basics of the modified technique involved incubating the gel in SDS running buffer for 10 minutes, incubating the gel with imidazole mix (200 mM imidazole pH 7.4, 0.01% SDS), rinsing the gel with water twice, soaking the gel in 150 mM zinc sulphate (pH 6.5), shaking manually until the desired staining was achieved and replacing the zinc sulphate with water

2.5 Quantification of protein concentration

2.5.1 Quantification using SDS gels

Gels were loaded with a range of protein standard concentrations, run according to **method 2.4.9** and stained using the desired staining technique. Actin was typically used as the protein standard and the range of concentrations loaded always extended beyond the presumed sample concentration. Image Lab^{TM} software was used to estimate the concentration of the sample relative to the concentration standards.

2.5.2 Quantification using nanodrop

A NanoDrop^{TM} light Spectrophotometer (ThermoFisher Scientific) was used to measure the 280 nm absorbance of samples while a NanoDrop ND-1000 (ThermoFisher Scientific) was used to measure the 290 nm absorbance. Beer-Lambert's law was then used to estimate the concentration of protein (**equation 2.1**). *A* is the absorbance of the sample and given as an output by the nanodrop machine, ε is the extinction coefficient of the sample at the wavelength being measured, *c* is the concentration of protein, and *l* is the path length.

$$A = \varepsilon c l \tag{2.1}$$

The 280 nm extinction coefficient was estimated using the ProtParam tool (ExPASy.org). Actin was stored in an ATP-containing buffer. ATP absorbs at 280 nm and thus a wavelength of 290 nm was used. The 290 nm extinction coefficient used for actin was 0.63 mg^{-1} mL^{-1} cm^{-1} as was identified within the literature (Hatano *et al.*, 2018).

2.6 Biochemical assays

2.6.1 Pyrene assay

Pyrene assays were carried out in polystyrene wells (ProxiPlate^{TM} from PerkinElmer) and observed using a Cary Eclipse fluorimeter (emission 364 nm, slit 10 nm round; excitation 385 nm, slit 20 nm). A multichannel pipette was used to simultaneously transfer Las17 and KME (made up to 0.5x KME) to each assay well. The wells were mixed and checked for air bubbles before being placed into the fluorimeter. Each assay well (prior to the addition of Las17 and KME) contained 3 µM G-actin, 0.3 µM pyrene-actin, 0.2 mM ATP, 0.5 mM DTT, and G-buffer. Both actin and pyrene actin were obtained from other members of the Ayscough lab following the protocols detailed in (Spudich and Watt, 1971) and (Doolittle, Rosen and Padrick, 2013) respectively.

2.6.2 Lysine, cysteine, and SNAP-tag labelling for MST

Lysines were covalently linked to a 647 nm emission fluorophore using the Monolith RED-Maleimide 2nd Generation Lysine Protein Labelling Kit provided by NanoTemper Technologies (Cat.# MO-L011). Briefly, 10 µM of the prospectively labelled protein was mixed with 10-20 µM maleimide-NT647 dye, and the provided labelling buffer. The mixture was incubated in the dark for 30 minutes at room temperature.

Cysteine labelling followed the same protocol as lysine labelling except that the Monolith RED-Maleimide 2nd Generation Cysteine Protein Labelling Kit provided by NanoTemper (Cat.# MO-L014) was used as the labelling kit of choice.

SNAP-tag labelling followed the same protocol as lysine labelling except that the Monolith 2nd Generation Snap-Tag RED Protein Labelling Kit provided by NanoTemper (Cat.# MO-L019) was used as the labelling kit of choice. This kit uses benzylguanine (a binding target of SNAP-tags) as opposed to maleimide to covalently link the label and SNAP-tag.

Following labelling by the chosen kit, free dye was separated from the labelled protein using a buffer exchange column. The absorbance at 280 nm (protein) was compared against the absorbance at 647 nm (dye) to calculate labelling efficiency with a target efficiency value above 80%. When a low efficiency was observed, steps were taken which included increasing the initial concentration of the dye as per the manufacturer recommendations.

2.6.3 His-tag labelling for MST

His-tag labelling for MST used a reversibly-binding (as opposed to covalently-binding) dye with a red emission at 647 nm. This was achieved using a Monolith RED-tris-NTA His-Tag Labelling Kit provided by NanoTemper (Cat.# MO-L008). Briefly, 100 nM His-tagged protein was mixed with 50 nM dye and incubated in the dark at room temperature for 30 minutes. This was followed immediately by MST assays using the labelled His-tagged protein. Binding efficiency between the dye and His-tag was analysed using MST with free dye functioning as the target and the His-tagged protein as the ligand. An observed affinity above 10 nM was deemed sufficient.

2.6.4 Microscale thermophoresis

Microscale thermophoresis (MST) utilises the relationship between the rate of diffusion and shape of an object to determine a binding affinity. One protein is labelled with a fluorescent dye using one of the techniques described in **sections 2.4.2** to **2.4.5**. This protein is referred to as the “target” and is maintained at a 50 mM concentration across 16 eppendorfs. A protein suspected to interact with the target (termed the “ligand”) is titrated across the eppendorfs using serial 2-fold dilution. Solutions from each of the 16 eppendorfs are drawn into separate Monolith NT.115 Premium Treated Capillaries (NanoTemper Technologies). These capillaries are placed into a Monolith NT.115 Microscale Thermophoresis device (NanoTemper Technologies) and a binding curve is generated. All parameters were kept constant including buffer composition, 20% LED power, a 40% IR laser power, and a maintained temperature of 22^{o}C. Tween-20 was added to all buffers at a final concentration of 0.05% to increase sample homogeneity. Where indicated, other elements such as BSA or proteins which inhibit the interaction were added at a constant concentration to all eppendorfs prior to contact with the capillaries. The “K_d fit” model was executed in the provided software to produce a K_d value from the data. This fitting involves the value of Fnorm which was calculated from the fluorescence difference before and after heating.

2.6.5 SD-test

An SD-Test assess weather an interaction observed in MST is the result of a specific binding event, or a non-specific event such as aggregation. The protocol was taken from the website of NanoTemper (manufacturer of the MST device) (‘SD-Test - Definition and Relevance | Nanopedia’, 2023). This consisted of mixing each of the 16 loading samples with equal parts SD-mix (4% SDS and 40 mM DTT) before incubating them at 95^{o}C for 5 minutes. The denatured samples were then loaded into capillaries and run according to **method 2.6.4**.

2.6.6 Biolayer interferometry

Biolayer interferometry records the association and dissociation rate constants by measuring changes in light reflection due to the association and dissociation of an analyte. Samples were placed in the wells of a polystyrene plate (ProxiPlateTM from PerkinElmer) and placed in a 2-Channel System: Octet® R2 (from Sartorius). Either Octet NTA Biosensors or Octet GST Biosensors were used as the assay probes. The following protocol was used across 5-8 rows.

1. Baseline in buffer for 30 seconds
2. Sample loading for 30-120 seconds depending upon the protein*
3. Baseline in two separate buffer wells for 30 seconds each
4. Association phase for 60-180 seconds depending upon the protein
5. Dissociation phase for 60-180 seconds depending upon the protein
6. Probe regeneration using $NiSO_4$ and glycine (His-binding probes only)

**Before performing binding assays across a range of analyte concentrations, screening assays were performed to elucidate optimum conditions. During these screens, a constant analyte concentration was maintained whilst the concentration and loading time of the probe-binding protein was varied until an assay was observed where the loading curve was not saturating the probe whilst also providing adequate signal. This loading concentration/time was used for all future assays using this screened interaction.*

Analysis and visualisation were done using in GraphPad Prism 10.0.2 when no manual correction was required due to its output of a confidence interval. Where significant non-specific interactions were identified, the analysis was undertaken with excel (following data correction) using least-mean squared fitting and the solver function. Least-mean squared fitting used the following two formulas taken from the GraphPad Prism 10.0.2 "Association then dissociation" binding curve equations (**equations 2.2 and 2.3** respectively). The axis values are given by *x* and *y*, *Rmax* is the largest possible signal at 100% receptor binding, k_{on} is the association constant, k_{off} is the dissociation constant.

$$y = Rmax * \frac{[Analyte]}{[Analyte] + \ K_d}(1 - e^{(-1*[Analyte]k_{on}+k_{off}*x)}) \tag{2.2}$$

$$y = Rmax * \frac{[Analyte]}{[Analyte] + \ K_d}\left(1 - e^{(-1*[Analyte]k_{on}+k_{off}*x_0)}\right) e^{(-1*k_{off}*(x-x_0))} \tag{2.3}$$

Manual corrections were only applied when a linear increase in signal (system artefact – possibly due to non-specific interactions) was observed in the association phase and the accumulated increase was retained during the dissociation phase. This correction involved (1) calculating the linear gradient of the increase and (2) applying the inverse of this value across the data to remove the effect. Corrections were only applied to data where extensive screening (changing buffers, detergents, etc.) could not remove the observed artifact. Applying data correction to remove the linear increase before fitting has a precedent within literature (Müller-Esparza *et al.*, 2020).

3. Code Development: Model Alpha

The regulatory cloud hypothesis proposed within this thesis contains many endocytic proteins and binding interactions. These proteins, especially when considering their membrane-interacting nature, are difficult to express and analyse experimentally. A principal aim of this thesis was to build a computational model to assess the veracity of the cloud hypothesis and help direct experimental assays – which can be more time and resource intensive. Agent-Based Modelling (ABM) serves this function particularly well for reasons described in **chapter 1** – a spatiotemporal approach, the observation of emergent behaviours, and non-inductive nature (**section 1.8.2**). Two models were ultimately developed within this thesis with the second written in response to Model Alpha – the first computational model and the subject of this chapter. Due to the interlinked nature of code development, some sections of this chapter may prove more useful when viewed in the context of later sections.

3.1 Model Alpha design overview

The aim of the work presented in this chapter is to model the behaviour of Las17(300-422). This is modelled as an array of 5 polyproline binding sites. Each of these can bind to an SH3 domain, or alternatively to an actin monomer. The SH3 domains can be one of several individual proteins (e.g., Ysc84), or from a "cloud" of unspecified SH3 domains. They could also be part of a tandem pair of domains, e.g., in Sla1. Our limited understanding of Las17 nucleation mechanics made it difficult to state whether the polyproline region is a cross-filament, or linear filament nucleator (tandem nucleation mechanisms described in **section 1.3.4.1**). Therefore, we began with the simplest implementation which was a cross-filament nucleator (**figure 1.6**). This was implemented such that if three actin monomers are bound to Las17 at the same time, the actin monomers form an F-actin seed which can start to polymerise; alternatively, F-actin filaments can dissociate. Each of these binding and dissociation steps has a certain probability (calculated from their rates) that is accessed as a global variable array (*domain1_bind_prob*, *domain2_bind_prob*, *domain1_unbind_prob*, *domain2_unbind_prob*; explained later in **section 3.5.1**).

The model outlined above requires further detail to make it work properly. In Model Alpha, Las17 does not move. However, non-tandem domains (e.g., Ysc84 SH3s and G-actin) can and are represented with the SingleAgent data class. Tandem domains (e.g., Sla1) can also move, and any movements need to take all constituent domains with it. This was modelled by having separate agents, called TandemAgents, which represent such pairs. The interaction probabilities for each domain were located within a separate global variable array (e.g., *domain1_bind prob* and *domain2_bind_prob*) which enabled each domain in a tandem to possess their own rates. In addition to this, these agents need to know if they are bound at one end or both. Similarly, an F-actin trimer must behave differently from G-actin monomers, which was achieved by having a separate agent (a ScaffoldAgent) that re-calculates its barbed and pointed end positions following each movement step (translational followed by rotational). Las17 is even more complicated, because each site can be free or bound, with different affinities at each site and thus was represented by a unique agent (Las17Agents).

FLAME GPU uses a parallel graphics processing unit (GPU) for each function, meaning that all agents can move simultaneously within each iteration. This makes the overall calculation much faster, but means that at several points during an iteration, the program must carry out a set of checks to prevent multiple agents attempting an exclusive action: for example, to check that each binding site is not attempting to bind more than one protein. Las17 agents were fixed and arranged in a three-dimensional lattice to reduce model complexity. Fixing these agents substantially reduced the number of required parallel checks as it removed the possibility of a single actin or SH3 domain to simultaneously bind two Las17 agents as the separation of Las17s was greater than twice the radius of interaction (described later). This feature also reduced the need for managing complex coherency during movement and was particularly useful considering the parallel nature of the code used.

During each iteration, two agents are allowed to bind if they are close enough. Each interaction has a probability of binding and a probability of unbinding, meaning that there is no explicit definition of the dissociation constant, which is given simply by the ratio between the binding and unbinding rates from which these probabilities are ultimately derived. This is explained with more detail later in **section 3.7.1**. Agents are allowed to move a maximum distance per iteration defined by their estimated diffusion constant. One iteration also represents one time step (a unit of time applied to the simulation). This was given the value of 1 µs as this was sufficiently small enough to ensure no interaction probability exceeded a value of 1 whilst also being large enough to allow for sufficient stimulation running speeds.

Each iteration allows agents to move a random distance in continuous space defined by their diffusion coefficient and bounded by the spatial limits of the simulation space. Agents can then carry out their other operations such as binding, unbinding, polymerisation, and nucleation. Therefore, Model Alpha can be broken down into five phases (movement, polymerisation, Las17 binding, unbinding and actin nucleation.) as shown in **figure 3.1** and described in greater detail later this chapter.

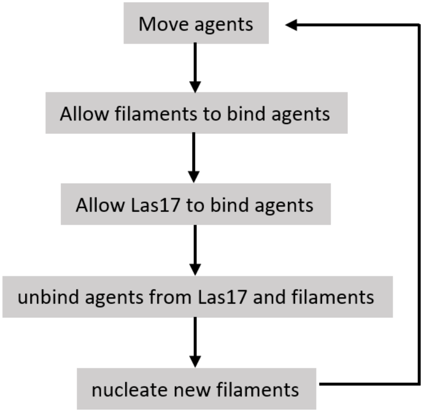

Move agents
Allow filaments to bind agents
Allow Las17 to bind agents
unbind agents from Las17 and filaments
nucleate new filaments

Figure 3.1. Simplified model flowchart. The functions used in the Model Alpha simulation can be grouped into five phases: movement, polymerisation, Las17 binding, unbinding and actin nucleation. These phases are shown in grey boxes whilst the flow of the code is given by the black arrows. A new iteration begins each time the code returns to the "*Move agents"* phase.

The percentage of actin agents occupying a filamentous state was measured during simulation run-time and used to derive a nucleation and polymerisation rate. Changing system variables such as binding affinities and the availability of SH3 domains allowed us to investigate how these factors influence Las17-dependent actin nucleation.

The possibility of Sla1 being outcompeted by cloud domains to increase activity was mathematically verified prior to building the ABM as shown in the **Chapter 9 appendix**.

3.1.1 Representing the Las17 binding regions

The 300-422 residue region that constitutes the Las17Agents included five binding sites termed bm1 to bm5. Bm1 represents an SH3-binding PxxP core motif between residues 314 to 317 whilst bm2-bm5 represent the four polyproline tracts – known to bind both actin and SH3 domains – of this peptide.

Bm3, which represents the second tract (PP2), can only bind SH3 domains and not actin agents. This is because PP2 is the only tract in the 300-422 region without an N-terminal arginine pair and is also spaced inconsistently. PP2 is directly N-terminal to the PP3 RRGPAPPPPP tract and viewing this region in PyMOL v4.6 reveals that this separation would be too small for both tracts to bind actin (**figure 3.2**). PP3 and PP4 also share a conserved RRGPAPPPPP motif that is hypothesised to be the strongest actin-binding site in the Las17 polyproline region. Furthermore, the separation between PP1 and PP3 is roughly consistent with the separation between PP3 and PP4 heavily suggesting PP3 to be involved with actin binding rather than PP2. This can easily be managed within the model by giving bm3 a vanishingly small affinity for actin.

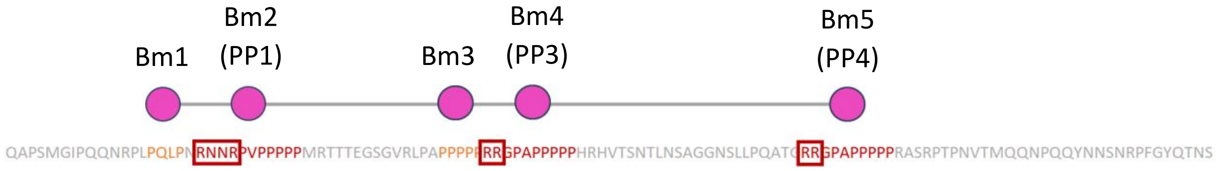

Bm2
(PP1)
Bm4
(PP3)
Bm5
(PP4)
Bm1
Bm3
QAPSMGIPQQNRPLPQLPNRNNRPVPPPPPMRTTTEGSGVRLPAPPPPRRGPAPPPPPHRHVTSNTLNSAGGNSLLPQATGRRGPAPPPPPRASRPTPNVTMQQNPQQYNNSNRPFGYQTNS

Figure 3.2. Structure of a Las17Agent. The structure of Las17 within the ABM is shown with magenta circles showing the position of binding motifs. Actin can bind via PP1, PP3, and PP4 which are given as bm2, bm4, and bm5. SH3 domains can bind any of the motifs provided they were prescribed an affinity within the initiation file. The 300-422 peptide sequence which this ABM interpretation represents is given below with red denoting the (assumed) actin-binding tracts/motifs, orange representing non-actin binding sequences which contain a PxxP motif, and red boxes which frame the arginine pairs that play a heavy role in actin interactions.

3.3 How runtime influenced the direction of code development.

Initial development of this project's computational model was undertaken using CPU-based non-parallelised python code. Three factors greatly affecting runtime were quickly identified. These included (1) using a dynamically typed computing language for the bulk of the simulation, (2) using sequential code which does not scale favourably to agent count, and (3) overcomplexity of the model. We consider each of these below.

Dynamically typed languages (e.g., python) execute the type-checking of variables at runtime meaning that these variables may possess a dynamic range in their function (Tratt, 2009). This versatility is particularly useful when developing code as a beginner, prototyping, or to speed up code development. However, this is also a major factor in reducing the speed at which python scripts are executed. On the other hand, statically typed languages (e.g., C) perform the type-checking during the compiling stage. While this reduces the versatility of the variables, it can also greatly improve performance (Tratt, 2009). Therefore, it was chosen to re-write the simulation into such a language.

Agent-based modelling requires the execution of functions across a number of agents represented as data class members (also termed “agent types”). Two methods for executing these functions include sequential and parallel execution. Sequential execution is defined as completing the function, or sections of the function, for each agent iteratively – running through a list of all the appropriate objects until completion. This method scales poorly with the number of agents in agent-based modelling, as the runtime for functions becomes roughly linear with the number of agents simulated. On the other hand, parallel execution allows all agents to undertake functions simultaneously, giving much better scaling (Richmond *et al.*, 2010). A downside to parallel code is that agents are unable to react to the decisions made by other agents during each execution layer. This complicates code development and can necessitate the need for checks which can often appear as non-intuitive functions. Nevertheless, it was decided that the anticipated performance increase was worth the added difficulties.

The final issue regarding runtime was the code itself, primarily edge-cases and over complexity. The former was a product of unfocussed program design due to the python model acting as a “first draft” in the code development pathway. Isolated issues which had the potential to cause bugs had to be addressed as additions to the code, thus reducing the simulation speed. However, this issue was not of major concern during this stage in development as several code revisions were planned. These revisions were focused on removing much of the model’s overcomplexity (a more significant issue) and simplifying the number of assumed variables. **Table 3.1** summarises the major features which were removed from the model along with justifications for their removal.

Table 3.1. Removed features and justifications

Removed component	**Original code implementation**	**Justification for removal**
Collisions between agents	Agents were prevented from physically occupying the same space by checking if they overlap at the end of the movement phase. If an overlap was detected, the movement would be rejected for the agents in question and their movements recalculated. This process would repeat until no overlaps in the system are detected. After this, the rest of the code was executed.	Agents were found to overlap with other agents infrequently (less than 1%), while the runtime cost for this feature was as much as 30%. Additionally, the binding radii are related to the probability of two points interacting at any instant during one timestep, and not merely to their positions at the time of overlap. The time of overlap is only one snapshot in the movement path predicted by the binding radius. Therefore, preventing agents from overlapping their physical radii was concluded as too computationally costly for the benefit.
Cooperativity between the SH3 domains of tandems	During the binding phase, tandem agents would check the bound status of their two SH3 domains. If one was bound, then the dissociation probability of the other would change in accordance with their cooperativity.	It is likely that some degree of allosteric cooperativity exists between the SH3 domains on Sla1 and Bzz1 considering their close spatial positioning. However, the magnitude of this is a complete unknown and thus inclusion of this feature was increasing the number of model assumptions unnecessarily. Instead, the cooperativity of tandem binding sites is solely an emergent behaviour linked to their close positions (i.e., avidity).
Orientation constraints on tandem agents	Movement constraints were placed on the rotational angles of tandem agents when only one SH3 domain was bound. This prevented the SH3 domains from moving in a way not permissible in the homology-modelled structures.	This feature was removed as it increased the complexity of the system in a biased way by preferentially targeting one agent type in a manner that may allow for unintentional overfitting. Furthermore, as mentioned when discussing “Collisions between agents”, the binding radii reflects the probability of binding in addition to its physical location, thus making this feature more redundant.

Table 3.1. Removed features and justifications (continued)

Removed component	**Original code implementation**	**Justification for removal**
Orientation constraints on tandem agents	Movement constraints were placed on the rotational angles of tandem agents when only one SH3 domain was bound. This prevented the SH3 domains from moving in a way not permissible in the homology-modelled structures.	This feature was removed as it increased the complexity of the system in a biased way by preferentially targeting one agent type in a manner that may allow for unintentional overfitting. Furthermore, as mentioned when discussing "Collisions between agents", the binding radii reflects the probability of binding in addition to its physical location, thus making this feature more redundant.

Table 3.1. Removed features and justifications. Many features were prototyped and removed as is typical during code development. However, this table gives justification for the removal of three previously well-integrated features that were present within early model builds. These features were removed to increase runtime and simplify the model.

Removing the three features mentioned in **table 3.1** increased the simulation run speed by between 30-50% and reduced the number of unknown parameters by three per multi-SH3 domain peptide. These changes were implemented during the code re-structuring that ultimately culminated in Model Alpha.

3.3.1 FLAME GPU 1 as a modelling environment

The two runtime issues of a dynamically typed computing language and sequential code were addressed by using the FLAME GPU 1 modelling environment. This environment was developed within the University of Sheffield and uses an XML framework to execute functions written in C (Richmond *et al.*, 2010).

The model is conceptualised in an XML document as a description of global variables, agents (data class objects), basic agent-specific function definitions, messages between functions and the order of function execution. Meanwhile, the specific function code and global function code is written in a C file. Both files are compiled into parallelised CUDA code which can be run on the GPU and neither utilised a code template (i.e., were written mostly from scratch). Executing the simulation requires an initiation file presented in an XML format that describes the number of agents and any specific variables. Program output includes updated versions of the XML initiation file given at time intervals (trajectory files) along with a csv file produced at the end of the simulation. This csv file is structured by the user and updated during a sequential CPU step executed between each iteration.

FLAME was written to be implemented on a GPU, which is the graphics hardware primarily developed for video display. GPUs carry out their operations in parallel and are specifically designed for rapid parallel calculations. They are therefore ideal for calculations of this sort. The GPU can handle a large number of parallel operations. A big advantage is that each iteration used typically thousands of Las17 agents, thousands of tandem agents, tens of thousands of single agents (which could be allocated to multiple different types), and a few hundred actin polymers. This effectively meant that each calculation simulated the interactions of thousands of independent Las17 molecules, and so provided internal statistical validation and avoided the need for multiple runs of the same calculation (provided the number of seeds generated were sufficient – e.g., greater than 50). This means that FLAME calculated a simulated one-second run for 1000 independent Las17 molecules, 1000 Sla1 molecules and 6000 actin molecules in only 6 hours. This is enough copies for a smooth and validated time course.

Agents can communicate by writing messages during the execution of a function (**figure 3.3a**). No more than one message can be written per agent and each message can include a list of variables. Other agents can then access this message list during the execution of a function, although only one message list can be accessed per function. This system allows for rapid and structured intercommunication between argents. However, it also serves as a limitation during code development as careful planning of messages is needed. For example, some messages may contain variables not intended for the agents which receive them but are instead passed along to later functions. Here, these message variables are referred to as "leapfrog variables" (**figure 3.3b**).

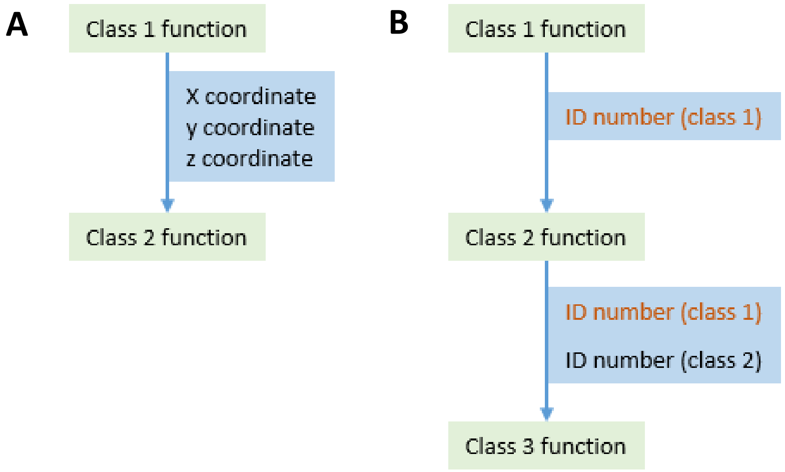


Figure 3.3. How messages function in FLAME GPU. Examples of messages are given. First, **A**) is an agent class (class 1) giving its three vector coordinates to a potential binding partner in agent class 2 using a message. Second, **B**) is an example of two agent classes (class 1 and class 2) giving their ID numbers to a third agent (class 3) for use in that function. Because each function can only read a single message list, class 1 first sends leapfrog variables (shown in orange text) to class 2 which in turn forwards those messages (along with their own IDs) to the third agent class.

Rigid structuring in the framework, such as communication through pre-defined messages, inability to call random numbers in global functions, and agent specificity to functions, can result in code redundancy (sections of identical, or highly derivative, code being used). Incidences of this were reduced where possible in accordance with good code development practice, however, they could not be avoided in some cases.

Ultimately, these minor issues related to environment rigidity were easily surmountable and minimised. The advantages of using FLAME included a significantly faster runtime (around 20-times when compared against the prototyped python model) and development within a structured environment consistent with other FLAME GPU models. The latter feature may aid others in understanding and adapting this thesis code, thereby increasing its application. The more recent version of FLAME (FLAME GPU 2) was not used as it was only made available during the later stages of code development. FLAME GPU 1 was used along with CUDA 10.0.

3.4 Model Alpha design specifics

Las17 agents are fixed in a three-dimensional lattice with tandem agents, single domain agents, and filament agents capable of moving in continuous space contained using periodic boundary conditions. The spatial separation between Las17 agents is defined within the initiation file and used to implement the concentration of Las17. The number of agents therefore dictate the sample size rather than the concentration. The concentration of other agents was determined by their ratio to Las17 agents.

Objects can interact via an interaction radius using a radius of capture system (Pogson *et al.*, 2006). This relates the binding radius and association rates into a single unified equation. The forward equation for a binding event between agents A and B to produce complex C can be shown as A + B = C with a rate of k_{on}. The change in [A] after a time step of Δt can therefore be represented with **equation 3.1**.

$$\Delta[A] = k_{on}[A][B]\Delta t \tag{3.1}$$

Because the binding of one agent A molecule to one agent B molecule to form complex C depletes the concentration of B by the same degree as A, the proportion of B molecules that bind A after a time step of Δt can be represented with **equation 3.2**.

$$\frac{\Delta[B]}{[B]} = \frac{\Delta[A]}{[B]} = k_{on}[A]\Delta t \tag{3.2}$$

The total volume inside which agent A interacts with agent B (V_i) over a simple time step can be given by the following equation for a given volume of V and assuming random distribution (**equation 3.3**). Meanwhile, the number of A molecules (n_A) at a given time can be given as **equation 3.4** (N_a representing Avogadro's constant) assuming concentration units of molar and volume units of cubic metres.

$$V_i = k_{on}[A]V\Delta t \tag{3.3}$$

$$n_A = 10^3[A]N_aV \tag{3.4}$$

These expressions can be combined to give the "interaction volume" (v) over a single timestep which would equate to V_i/n_A (**equation 3.5**).

$$v = \frac{k_{on}[A]V\Delta t}{10^3[A]N_aV} \tag{3.5}$$

All agents within Model Beta (and even the later discussed Model Beta) are spherical and thus the largest interaction radius possible can be given by **equation 3.6**. This is the key equation that defines the radius of capture first defined by (Pogson *et al.*, 2006).

$$r = \sqrt[3]{\frac{3k_{on}\Delta t}{4\pi 10^3 N_a}} \tag{3.6}$$

Model Alpha interactions can include tandem binding sites (e.g., the Sla1 binding a single Las17 with both of its SH3s) which differs from the agent-based NF-κB model presented in Pogson, et al, 2006. A direct implementation of this radius of capture method would results in a biasing issue in the cases of Sla1. Here, a small change in the association rate for one SH3 domain could result in complete binding abolition if the separation between Las17 binding sites is greater than interaction radius + the Sla1-SH3#1-2 linker. To address this, the radius of capture was kept constant, and a probability variable *P* was introduced to represent the difference in association rates between SH3s (**equation 3.7**).

$$P = \frac{3k_{on}\Delta t}{4\pi 10^3 Na\, r^3} \tag{3.7}$$

The binding probability *P* is therefore used by agents while they occupy a space within the radius of capture, *r*. The value of *r* chosen was 7.346 nm which was calculated from **equation 3.6** using a k_{on} value of 10^8 M^{-1} s^{-1} as this is around the diffusion limit for k_{on} and the largest rate achievable within Model Alpha (Schavemaker, Boersma and Poolman, 2018).

Binding probabilities for each possible pairwise interaction are calculated when generating the initiation file for a simulation run. These values are then placed into a global lookup table (*domain1_bind_prob* and *domain2_bind_prob*). The specifics for this are explained later in **section 3.7.1** when discussing the python script which generates the initiation file. The values used to generate the interaction radius always represented the largest possible value of k_{on} to prevent *P* values greater than 1.

The modelling environment requires two distinct files (as outlined in **section 3.3.1**). First is an XML file which defines the agent types, global variables, function conditions, messages, and the order of functions executed. The second file required is a lengthy C file which contains the code for each function referenced within the XML file. Functions are executed in parallel within a series of sequential "layers". **Figure 3.4** gives a flowchart of the simulation including agent functions and the pathways of messages. Later, a more detailed overview of each function and object data class will be given so this figure may prove a useful reference. Note that, due to a difference in terminology used by this model, actin filaments are sometimes referred to as scaffolds (e.g., ScaffoldAgent).

In addition to the model files, each simulation run requires a method for generating an initiation XML file (described later in **Section 3.7.1**). FLAME GPU derives its 'random' numbers from a seed which is generated during model compilation. Therefore, random numbers called within the model are pseudo-random. Generating a fresh invitation file every time the model is run redistributes the agent positions and thus introduces randomness between executions of the simulation.

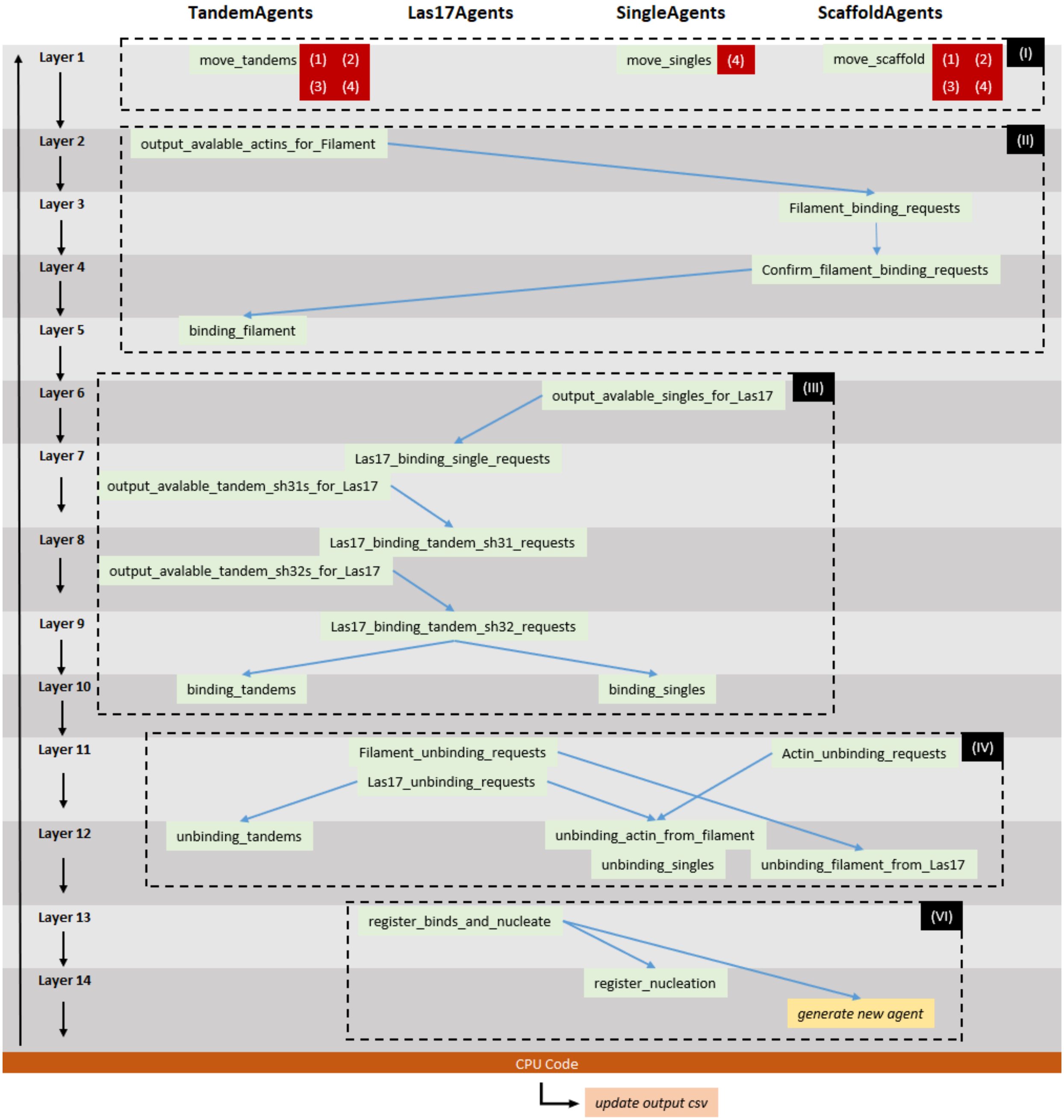


Figure 3.4. Detailed model flowchart. The flow of code during runtime is shown by the black line with a new iteration beginning each time Layer 1 is executed. Each layer is given on the left and functions belonging to that layer are denoted using green boxes in the appropriate agent class column (as shown along the top). All functions within a layer are executed simultaneously by all applicable agents. Blue arrows represent message lists with the function at the base of the arrow generating the list and the function(s) at the arrowhead reading the messages. Each function can only produce and accept a single message list. Red boxes show when global functions are accessed (1 = Tandem_x_count, 2 = Tandem_y_count, 3 = Tandem_z_count, and 4 = boundryCheck), with these functions being defined later in **section 3.6.1**. Functions can be broadly grouped into five phases as shown in **figure 3.1**. These include (I) movement, (II) polymerisation, (III) Las17 binding, (IV) unbinding, and (V) actin nucleation.

3.5 XML File

The XML file serves as the framework for the FLAME GPU model. It defines the structures of simulation elements along with what components run, in what order and with what conditions. This section (**section 3.5**) may also serve as a useful reference whist reading the subsequent section which walks through the C code and how it is broken down (**section 3.6**).

Model Alpha used four data classes to define its agents: SingleAgents, TandemAgents, ScaffoldAgents and Las17Agents. Each data class retains a set of agent variables which are updated during the simulation runtime. These agent types are defined throughout the next sub-sections in this thesis (**sections 3.5.1** – **3.5.6**).

3.5.1 Global Variables

There are 23 global variables defined within the XML file for Model Alpha. Global variables can only be changed during the CPU code at the end of each time step. They can be accessed – though not edited – by all agent functions where they are called. These variables are used to hold information that is specific to the simulation but not specific agents. This reduces the memory requirement during model construction as shared agent variables can be allocated as a global instead. Below is an overview of all global variables defined in the model (**table 3.2**).

Variable Name	Description	Functions referenced
TIME_COUNTER	Used to hold the current iteration number of the simulation timestep.	CPU code
PRINT_COUNTER	A counter variable that is increased by 1 every iteration and reset to 0 after an output is made to the csv file. This allows lines to be saved every **X** iterations thus saving memory.	CPU code
XMAX, YMAX, ZMAX	These define the maximum allowed axis value for their respective axis before a boundary condition will act upon the agent.	boundryCheck
XMIN, YMIN, ZMIN	These define the minimum allowed axis value for their respective axis before a boundary condition will act upon the agent.	boundryCheck
domain1_bind_probs	This array variable holds the binding probability between SH3 domains and Las17 tracts for each SingleAgent and the first SH3 domain of each TandemAgent.	Las17_binding_single _requests Las17_binding_tandem _sh31_requests
domain2_bind_probs	This array variable holds the probability of binding between SH3 domains and Las17 tracts for the second SH3 domain of each TandemAgent.	Las17_binding_tandem _sh32_requests
domain1_unbind_probs	This array variable holds the unbinding probability between SH3 domains and Las17 tracts for each SingleAgent and the first SH3 domain of each TandemAgent.	Las17_binding_single _requests Las17_binding_single _requests
domain2_unbind_probs	This array variable holds the probability of unbinding between SH3 domains and Las17 tracts for the second SH3 domain of each TandemAgent.	Las17_binding_tandem _sh32_requests
SH3_COLLISION_RADIUS	2x the radius of an SH3 domain and defines the minimum distance between the SH3 domains in a TandemAgent.	move_tandems
SLA1_SH3_MAX_SEPARATION	The maximum distance the first and second SH3 domains of Sla1 can be separated.	move_tandems
INTERACTION_RADIUS	The radii of interaction between domains. Once two domains are separated by a distance less than this value, they attempt to interact according to their binding probability.	Filament_binding _requests Las17_binding _single_requests Las17_binding_tandem _sh31_requests Las17_binding_tandem _sh32_requests

Table 3.2. Global variables of Model Alpha. A list of all the global variables used in Model Alpha, their descriptions, and which functions they refer to.

3.5.2 Agent data class: SingleAgent

SingleAgents represent the SH3 domains of endocytic protein that contain only one SH3 in their sequence. These include Cloud SH3s, Ysc84, and Rvs167, etc. **Table 3.3** shows the agent variables held by the SingleAgent data class.

Variable Name	Description	Default value
id	A unique identification number used to reference an individual agent.	Unique integer
type	The "type" variable identifies the protein that this agent represents. This allows the agent to access the correct binding rates.	0 if agent is actin, 5 if agent is cloud, 10 if agent is Ysc84
filament_subunit_number	This variable is only accessed by actin. It holds the position of the actin agent within a filament should it be incorporated in one.	-1
binding_state	A Boolean variable which shows if the agent is bound (1) or not bound (0) to Las17.	0
k	This variable is scaled based on the diffusion coefficient of the domain it represents and governs the rate of diffusion.	8.5 for SH3 4 for actin (explained later in **Section 3.7.1**)
x1, y1, z1	The coordinate variables for the domain's positional vector.	random

Table 3.3. The data class variables of SingleAgents. A list of all the agent-specific variables found within the SingleAgent data class, their descriptions, and default values given during creation of the initiation XML file.

3.5.3 Agent data class: TandemAgent

TandemAgents represent peptides of endocytic protein that contain two SH3 domains. These include Bzz1 and the first two SH3 domains (residues 3-132) of Sla1. **Table 3.4** shows the agent variables held by the TandemAgent data class. Theoretically, the vector positions of each domain could be calculated when called rather than stored as a variable. However, calculating them once during the movement phase and storing the variables was more efficient due to the domain vector positions being called multiple times throughout the code.

Variable Name	Description	Default value
id	A unique identification number used to reference an individual agent.	Unique integer
type	The “type” variable identifies the protein that this agent represents. This allows the agent to access the correct binding rates.	15 if agent is Sla1, 20 if agent is Bzz1
binding_state	How the agent is bound to Las17 with 0 being unbound, 1 having just the first SH3 bound, 2 having just the second SH3 bound and 3 having both domains bound.	0
k	This variable is scaled based on the diffusion coefficient of the domain it represents and governs the rate of diffusion.	3.2 for Sla1 2.7 for Bzz1
rotation_k	This variable scales the rate of rotation for a TandemAgent.	2
axis_phi, axis_theta	The two polar angles for the SH3-1 to SH3-2 axis.	random
centre_x, centre_y, centre_z	The vector coordinates for the centre of the agent.	random
x1, y1, z1	The vector coordinates for the first SH3 domain of the agent.	random
x2, y2, z2	The vector coordinates for the second SH3 domain of the agent.	random

Table 3.4. The data class variables of TandemAgents. A list of all the agent-specific variables found within the TandemAgent data class, their descriptions, and default values given during creating of the initiation XML file.

3.5.4 Agent data class: Las17Agent

The 300-422 region of Las17 is represented by the Las17Agent data class. This agent type holds many more variables than the other data classes as it regulates the entire multi-protein complex during Las17-SH3 and Las17-actin interactions. Another factor which necessitates a high number of variables is that an increasing number of binding sites results in an increasing number of variables due to each agent representing the entire peptide. This is a feature of

Model Alpha and is less pronounced in the second model presented later in this thesis. **Table 3.5** shows the agent variables held in the Las17Agent data class.

Table 3.5. The data class variables of Las17Agents

Variable Name(s)	Description	Default value
id	A unique identification number used to reference an individual agent.	Unique integer
bm1_partner_id, bm2_partner_id, bm3_partner_id, bm4_partner_id, bm5_partner_id	Variables holding the ID of agents bound to the five (bm1-bm5) binding sites on Las17.	-1
bm1_partner_domain, bm2_partner_domain, bm3_partner_domain, bm4_partner_domain, bm5_partner_domain	Domain numbers for bound agents (1 for SingleAgents or the first SH3 of TandemAgents and 2 for the second SH3 of TandemAgents). This is used to update the *binding_state* variable of TandemAgents.	-1
bm1_partner_unbindProb, bm2_partner_unbindProb, bm3_partner_unbindProb, bm4_partner_unbindProb, bm5_partner_unbindProb	The probability of a bound agent unbinding during each timestep. This variable is set during binding events to reduce the number of global variable calls.	0
bm1_partner_type, bm2_partner_type, bm3_partner_type, bm4_partner_type, bm5_partner_type	The type of the bound agents (should an agent be bound). These variables are used to detect when three actin agents are bound so that a filament can be nucleated.	-1
bm1_partner_id_request, bm2_partner_id_request, bm3_partner_id_request, bm4_partner_id_request, bm5_partner_id_request	Temporarily occupied variables that hold the ID variable of an agent attempting to bind a site on Las17. These can be overwritten during the binding function by closer candidates.	-1
bm1_partner_unbindProb_request, bm2_partner_unbindProb_request, bm3_partner_unbindProb_request, bm4_partner_unbindProb_request, Bm5_partner_unbindProb_request	Temporarily occupied variables that hold the unbinding probability of agents attempting to bind Las17 binding sites. This variable can be overwritten during the binding function by closer candidates.	-1
bm1_partner_type_request, bm2_partner_type_request, bm3_partner_type_request, bm4_partner_type_request, bm5_partner_type_request	Temporarily occupied variables that hold the “type” variable of agents attempting to bind Las17 binding sites. These variables can be overwritten during the binding function by closer candidates.	-1
bm1_partner_request_distance, bm2_partner_request_distance, bm3_partner_request_distance, bm4_partner_request_distance, bm5_partner_request_distance	Temporarily occupied variables holding the distances between each Las17 binding site and their closest binding candidate. This is used to determine which candidate is the closest and thus allowed to bind.	-1
PP1_actin_access, PP3_actin_access, PP4_actin_access	A Boolean variable which shows whether bm2 (PP1), bm4 (PP3) or bm5 (PP4) is available to bind G-actin (1) or not (0).	1

Table 3.5. The data class variables of Las17Agents (continued)

centre_x, centre_y, centre_z	The vector position for the centre of the agent.	random
bm1_x, bm2_x, bm3_x, bm4_x, bm5_x	The x variable for the positional vectors of each of the agent's respective binding site. Las17Agents cannot move in Model Alpha and are orientated parallel to the Y and Z plane.	centre_x *+ length from the centre of the 300-422 peptide*.
no_ACTIN	A Boolean reporter variable to show if the Las17Agent is (0) or is not (1) not bound to any actin agents.	0
one_ACTIN	A Boolean reporter variable to show if the Las17Agent is (1) or is not (0) not bound by exactly one actin agent.	0
two_ACTIN	A Boolean reporter variable to show if the Las17Agent is (1) or is not (0) not bound by exactly two actin agents.	0
NUCLEATION_EVENTS	The number of actin seeds nucleation by each agent is recorded using this variable.	0
bm1_SH3, bm2_SH3, bm3_SH3, bm4_SH3, bm5_SH3	Boolean reporter variables to show if binding sites either are (1) or are not (0) occupied by an SH3 domain.	0
bm1_actin, bm2_actin, bm3_actin, bm4_actin, bm5_actin	Boolean reporter variables to show if binding sites either are (1) or are not (0) occupied by an actin agent.	0
bm1_seed, bm2_seed, bm3_seed, bm4_seed, bm5_seed	Boolean reporter variables to show if binding sites either are (1) or are not (0) occupied by an actin filament.	0
seed_bound	A reporter Boolean showing whether this agent is (1) or is not (0) bound to a filament.	0
PP3_tandem_bound	A reporter Boolean variable showing whether bm4 (PP3) is (1) or is not (0) bound to a TandemAgent.	0
seed_id	The unique ID identifier for the filament bound to this Las17Agent.	0

Table 3.5. The data class variables of Las17Agents. A list of all the agent-specific variables found within the Las17Agent data class, their descriptions, and default values given during creation of the initiation XML file.

3.5.5 Agent data class: ScaffoldAgent

A ScaffoldAgent is dynamically built during the simulation runtime to represent an actin filament. Upon complete dissolution of the filament, the respective ScaffoldAgent is "killed" during the timestep when this occurred. **Table 3.6** shows the agent variables held by the ScaffoldAgent data class.

Table 3.6. The data class variables of ScaffoldAgents

Variable Name	Description	Default value
id	A unique identification number used to reference an individual agent.	Unique integer
binding_state	A Boolean variable to show if the agent is bound (1) or not bound (0) to Las17.	0
size	The number of actin subunits in the filament.	3
binding_prioritiser	A temporarily occupied variable used to identify which filament a monomer can bind should it be close enough to bind two or more different ScaffoldAgents.	-1
requested_id	A temporarily occupied variable used to hold the agent ID attempting to bind this agent prior to binding confirmation.	-1
barb_actin_no	The current subunit number at the barbed end position. This allows actin agents to identify their position within the filament.	2
point_actin_no	The current subunit number at the pointed end position. This allows actin agents to identify their position within the filament.	0
k	This variable is scaled based on the diffusion coefficient of the domain it represents and governs the rate of diffusion.	1.2
rotation_k	This variable scales the rate of rotation for a ScaffoldAgent.	2
axis_phi, axis_theta	The polar angles for the barb to point axis.	random
centre_x, centre_y, centre_z	The positional vector coordinates for the centre of the agent.	random
barb_x, barb_y, barb_z	The positional vector coordinates for the barbed end.	Random

Table 3.6. The data class variables of ScaffoldAgents (continued)

point_x, point_y, point_z	The positional vector coordinates for the pointed end.	random

Table 3.6. The data class variables of ScaffoldAgents. A list of all the agent-specific variables found within the ScaffoldAgent data class, their descriptions, and default value given during both creating of the initiation XML file and agent creation during nucleation.

3.5.6 Function order

The order of functions executed during each simulation iteration is detailed at the bottom of the XML file. This order is shown within **table 3.7** and may prove a useful lookup table during the later **section 3.6**. Due to the parallel nature of FLAME GPU, a function acts upon all agents once it is called and numerous functions can be called simultaneously. These rounds of parallel function executions are termed "function layers" and each layer is iterated through sequentially throughout the simulation.

Table 3.7. Function layers of Model Alpha

Layer number	Functions called	Brief overview of the layer's function
1	move_singles, move_tandems, move_scaffold	All agents are moved.
2	output_avalable_actins_for_Filament	Unbound, SingleAgents of type 0 give their positions.
3	Filament_binding_requests	ScaffoldAgents identify which agents they can bind.
4	Confirm_filament_binding_requests	ScaffoldAgents communicate between one another to sort any conflicts between binding requests.
5	binding_filament	SingleAgents permitted to bind ScaffoldAgents update their variables.
6	output_avalable_singles_for_Las17	Unbound SingleAgents give their positions

Table 3.7. Function layers of Model Alpha (continued)

7	Las17_binding_single_requests, output_avalable_tandem_sh31s_for_Las17	Las17Agents identify which SingleAgents they can bind. TandemAgents with their first SH3s unbound give their positions
8	Las17_binding_tandem_sh31_requests, output_avalable_tandem_sh32s_for_Las17	Las17Agents identify which first domains of TandemAgents they can bind. TandemAgents with their second SH3s unbound give their positions
9	Las17_binding_tandem_sh32_requests	Las17Agents identify which second domains of TandemAgents they can bind.
10	binding_singles, binding_tandems	SingleAgents and TandemAgents permitted to bind Las17Agents update their variables.
11	Actin_unbinding_requests, Las17_unbinding_requests, Filament_unbinding_requests	Las17Agents and ScaffoldAgents check whether they can unbind and bound agents this timestep. If so, they update their variables
12	unbinding_actin_from_filament, unbinding_singles, unbinding_tandems, unbinding_filament_from_Las17	SingleAgents, TandemAgents, and ScaffoldAgents permitted to unbind during layer 11 update their variables.
13	register_binds_and_nucleate	Reporter variables are set and Las17Agents can allow an actin filament to nucleate if it is bound by three actin agents simultaneously.
14	register_nucleation	If any actin filaments were nucleated this timestep, the monomers which make up the new seeds update their variables. This is to allow the resultant filament to move as a unit should it later unbind from Las17.

Table 3.7. Function layers of Model Alpha. A list of all the function layers detailed in the XML model file, the functions contained within, and the general purpose of the layer in context of the simulation.

3.6 Function file

As described in **section 3.5.6**, each iteration of the simulation is executed as a sequential series of function layers. Each layer contains one or more functions which are executed in parallel. The code for each function is written using C in the function file. Guard loops for these functions are defined in the XML file, however, to maintain reading coherency, these will be described alongside their respective functions below.

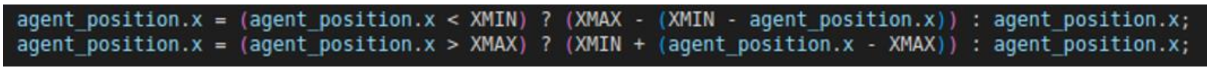
3.6.1 Global Functions

Lines of code were originally sometimes repeated multiple times within the function C file. Where possible, these lines were written into separate functions – called global functions – that can be called within the function layers. This method reduces the number of repeated lines which helps make code management more streamlined. These global functions are explained within this section.

<u>boundryCheck:</u>

This function accepts an agent's positional vector as an argument and checks whether it is located outside of the bounds set by the *XMAX*, *XMIN*, *YMAX*, *YMIN*, *ZMAX*, or *ZMIN* global variables (**section 3.5.1**). If any vector values exist outside the bounds set, they are changed to a value at the opposite axis of the simulation space (**code 3.1** for example code concerning the X axis. Equivalent code exists for the Y and Z axis).

Code 3.1



```
agent_position.x = (agent_position.x < XMIN) ? (XMAX - (XMIN - agent_position.x)) : agent_position.x;
agent_position.x = (agent_position.x > XMAX) ? (XMIN + (agent_position.x - XMAX)) : agent_position.x;
```

For example, if the coordinates of an agent are [x = 11, y = -5, z = 3] and the axis bounds are 10/-10, then the exceeded x coordinate will be set at *XMIN* (the opposite side of the axis) to give [x = -9, y = -5, z = 3].

This mechanism of maintaining a fixed simulation space is termed a "periodic boundary condition" as it simulates the area of a theoretically infinite solution with the assumption that "if a molecule leaves the space, another will enter it to keep the concentration stable".

<u>tandem_x_calc, tandem_y_calc, and tandem_z_calc:</u>

These three functions calculate a new coordinate position during movement based on the given arguments of distance (*r*), *theta* angle, *phi* angle, and an *original coordinate*. Movement in model Alpha is undertaken using polar coordinates. Therefore, the following 3 formulas are used to calculate a new coordinate (**equations 3.8-3.10**).

$$New\ x\ coordinate = r * \sin(theta)\cos(phi) + original\ x\ coordinate \tag{3.8}$$

$$New\ y\ coordinate = r * \sin(theta)\sin(phi) + original\ y\ coordinate \tag{3.9}$$

$$New\ z\ coordinate = r * \cos(theta) + original\ z\ coordinate \tag{3.10}$$

3.6.2 Layer 1: Agent Movement

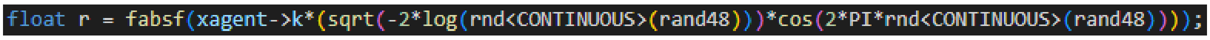
Agent's subject	**Function guard**	**>Message input** **Message output>**
Function: **move_singles**		
SingleAgent		
Function: **move_tandems**		
TandemAgent		
Function: **move_scaffold**		
ScaffoldAgent		

Las17Agents are fixed and arranged in a regular array in a 3-dimensional grid with each agent positioned equidistant from their neighbours to reduce model complexity and increase simulation run speeds. At the beginning of each timestep, all unbound SingleAgents, unbound

TandemAgents and unbound ScaffoldAgents execute similar functions to move via a Brownian motion random walk. This method was adapted in principle from the "Animated 3D random walk" webpage of Matplotlib (Matplotlib, 2012) with the addition polar coordinates and a scaling factor (*k*) to reproduce an experimentally consistent diffusion rate (explained later in **section 3.8**).

All unbound agents other than Las17 move through a three-stage process: direction, distance, and execution. First, a normally distributed random number is generated using the Box-Muller transformation and multiplied by the scaling coefficient *k* as shown below (**code 3.2**) (Box and Muller, 1958). This method uses continuous random numbers (between 0 and 1) for the transformation. The Box-Muller transformation is used as continuous random numbers are the only random number type that can be called using FLAME GPU 1. The scaling coefficient is set during the initiation phase and is derived from that object's estimated diffusion coefficient as described later in **section 3.7.1**.

Code 3.2



```
float r = fabsf(xagent->k*(sqrt(-2*log(rnd<CONTINUOUS>(rand48)))*cos(2*PI*rnd<CONTINUOUS>(rand48))));
```

Second, a random redirection is chosen using the spherical coordinate system (a derivation of the 2D polar coordinate system). Polar coordinates describe the point of an object in space relative to an origin using two angles (Φ phi and θ theta) and a distance. This method allows moving agents to explore a spherical space rather than a cuboid space as would be the result of simply adding randomised x, y, and z values. A new, post-movement vector position is generated using the random distance generated during the first stage, random values for each of the two polar angles, and the original vector coordinates of the agent. **Equations 12**, **13**, and **14** (given in **section 3.6.1**) are used to generate new x, y, and z coordinates respectively.

Finally, the movement is executed by replacing the original vector variable with the newly generated data structure. TandemAgents then rotate to determine the positions of their two SH3 domains while ScaffoldAgents rotate to determine the position of their barbed and pointed ends. These rotations follow the same method as movement. They use the agent's central movement vector as their base origin point and a random distance calculated using **equation 3.11**. Both agent types use the global tandem_x_calc, tandem_y_calc, and tandem_z_calc functions to reduce code redundancy. Upon movement completion, all agents execute the *boundryCheck* function to ensure that none exist outside the simulation space. This function applies the periodic boundary condition explained previously to vectors which do exist beyond the allowed space (i.e., wrap them around the other side of the simulation box).

$$distance = \frac{random\ number\ (0-1) * maximum\ possible\ separation}{2} \tag{3.11}$$

3.6.3 Layer 2: Actin positions

Agent's subject	Function guard	>Message input Message output>
Function: **output_avalable_actins_for_Filament**		
SingleAgent	Unbound and of type 0 (G-actin)	avalable_actin_location>

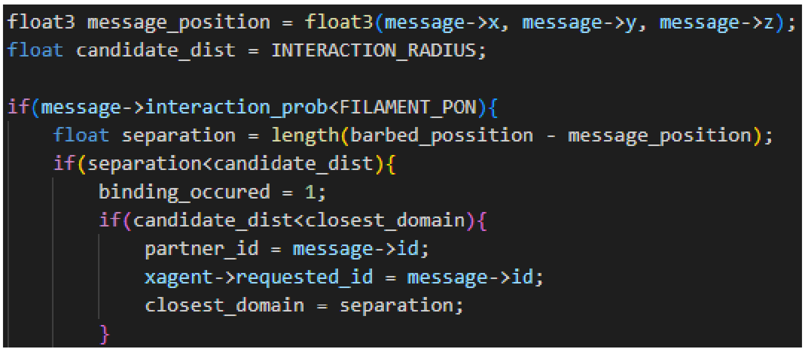
Each unbound actin SingleAgent outputs a message with its ID, a random number from 0-1, *x1*, *y1* and *z1*. In **layer 3**, these values will allow the agent to be identified by the barbed ends of filaments, test unbinding using the random number, and calculate distances using the given coordinates.

3.6.4 Layer 3: Identifying candidates for filament binding.

Agent's subject	**Function guard**	**>Message input** **Message output>**
Function: **Filament_binding_requests**		
ScaffoldAgent	Unbound and of type 0 (G-actin)	>avalable_actin_location priority_ticket>

ScaffoldAgents search the message list of **layer 2** and iterate through in search of an appropriate agent to bind (**code 3.3**). During each message iteration, the probability of binding is checked to verify that the unbound actin can bind the filament. This step is undertaken first to exclude cases where the agents are unable to bind, thereby avoiding the more computationally expensive step of distance calculation. Following this, the distance between the unbound actin and the barbed end of the filament is calculated using the FLAME GPU "length" function which takes the two coordinates as arguments. If the distance is less than the *INTERACTION_RADIUS* global variable, the two agents are in range to bind. In the case of multiple actin monomers binding the same filament simultaneously, only the closest actin candidate is permitted to bind. This is undertaken by recording each interaction partner in local variables that are overwritten should a closer interaction partner be identified. Filaments which successfully identify a potential binding partner output their ID, the ID of the partner, and the calculated distance as a message.

Code 3.3



```
float3 message_position = float3(message->x, message->y, message->z);
float candidate_dist = INTERACTION_RADIUS;

if(message->interaction_prob<FILAMENT_PON){
    float separation = length(barbed_possition - message_position);
    if(separation<candidate_dist){
        binding_occured = 1;
        if(candidate_dist<closest_domain){
            partner_id = message->id;
            xagent->requested_id = message->id;
            closest_domain = separation;
        }
```

3.6.5 Layer 4: Sorting binding requests

Agent's subject	Function guard	>Message input Message output>
Function: **Confirm_filament_binding_requests**		
ScaffoldAgent	Agents which identified a suitable binding partner during layer 3	>priority_ticket actin_ticket>

ScaffoldAgents which identified a suitable unbound actin agent to bind must first compare this prospective interaction with those of other ScaffoldAgents. This is because the parallel nature of FLAME GPU could, theoretically, result in the same actin agent binding two or more filaments simultaneously.

To identify which interactions should be prioritised, each ScaffoldAgent iterates through the messages of **layer 3** to identify any agents are both (1) attempting to bind the same actin agent and (2) closer to that agent than itself. If this dual condition is met, then binding for this pair will no longer occur. ScaffoldAgents which are successfully able to bind their prospective partners then update the appropriate variables. Binding involves the following variable updates:

- Increase the *size* variable by 1.
- Increase the *barb_actin_no* by 1.
- Recalculate *k* as *FILAMENT_BASE_K/size*.
- Recalculate *rotation_k* as *FILAMENT_ROTATE_BASE_K/size*.

Every ScaffoldAgent that successfully binds an actin agent then outputs a message with its ID, the ID of the actin agent, and *barb_actin_no*.

3.6.6 Layer 5: Updating newly bound actin agents.

Agent's subject	**Function guard**	**>Message input** **Message output>**
Function: **binding_filament**		
SingleAgent	Unbound agents of type 0 (actin)	>actin_ticket

All unbound actin SingleAgents iterate through the messages of **layer 4**. If their ID is identified, then the agent will update to a bound status. Binding involves the following variable updates:

- Changing *binding_state* to 1 as to indicate a bound status.
- Updating *filament_id* to the ID of the filament which this agent is now a part of.
- Update the *filament_subunit_number* to the *barb_actin_no* given in the message. This allows the agent to identify itself during a later unbinding event should one occur.

3.6.7 Layer 6: Single positions

Agent's subject	**Function guard**	**>Message input** **Message output>**
Function: **output_avalable_singles_for_Las17**		
SingleAgent	Unbound agents only	avalable_single_location>

Each unbound SingleAgent outputs a message with its id, type, a random number (0-1), *x1*, *y1* and *z1*.

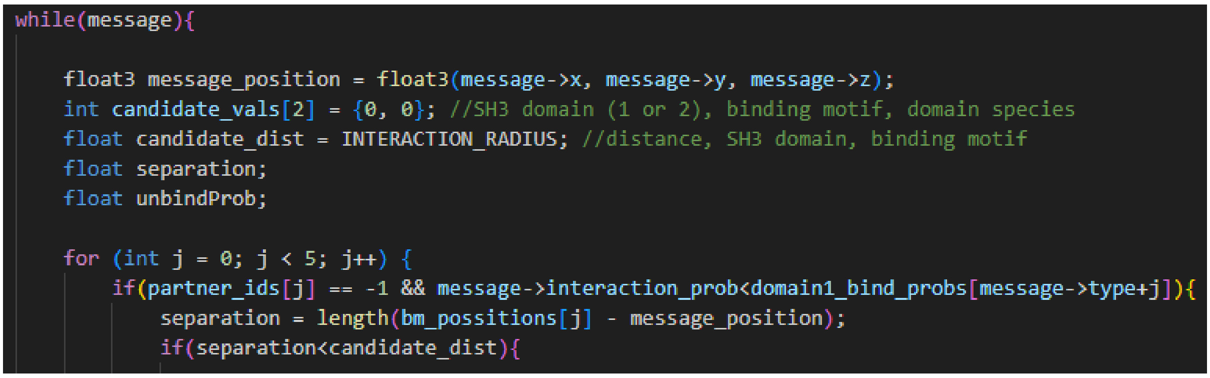
3.6.8 Layer 7: Identifying potential binding partners; Single domains.

Agent's subject	**Function guard**	**>Message input** **Message output>**
Function: **Las17_binding_single_requests**		
Las17Agent	Unbound agents only	>avalable_single_location
Function: **output_avalable_tandem_sh31s_for_Las17**		
TandemAgent	The first SH3 domain of the agent must be unbound	avalable_tandem_sh31_location>

Each Las17 will generate several five-element array variables and populate them with data from the five binding motifs (bm1-bm5). This allows the function to only require one set of instructions per binding site which can be iterated over, thus reducing code redundancy.

Each message from **layer 6** is read and analysed as shown in **code 3.4**. *Note that, for the purpose of being concise, **code 3.4** omits a set of binding criteria checks which could be used to implement motif-specific binding requirements.*

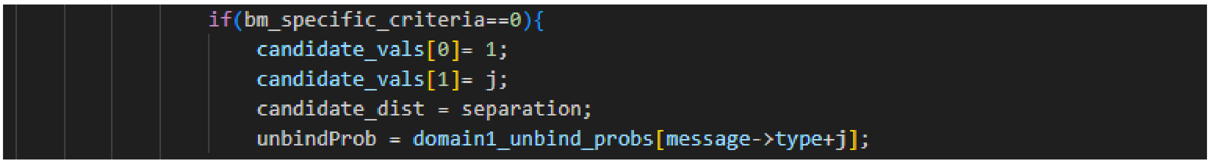
Code 3.4



```
while(message){

    float3 message_position = float3(message->x, message->y, message->z);
    int candidate_vals[2] = {0, 0}; //SH3 domain (1 or 2), binding motif, domain species
    float candidate_dist = INTERACTION_RADIUS; //distance, SH3 domain, binding motif
    float separation;
    float unbindProb;

    for (int j = 0; j < 5; j++) {
        if(partner_ids[j] == -1 && message->interaction_prob<domain1_bind_probs[message->type+j]){
            separation = length(bm_possitions[j] - message_position);
            if(separation<candidate_dist){
```



```
                if(bm_specific_criteria==0){
                    candidate_vals[0]= 1;
                    candidate_vals[1]= j;
                    candidate_dist = separation;
                    unbindProb = domain1_unbind_probs[message->type+j];
```

This code is derived from the previously discussed *Filament_binding_requests* function and identifies the closest unbound SingleAgent for each binding motif that has a separation less than the interaction radius. These are saved as temporary variables using the *agent request* agent variables. Unbind probabilities are recorded in the appropriate *unbindProb* variables for use in the unbinding functions of **layer 11**.

output_avalable_tandem_sh31s_for_Las17 is almost identical to the previously discussed *output_avalable_singles_for_Las17*. Each TandemAgent with an unbound first SH3 domain outputs a message with its id, type, a random number between zero and one, *x1*, *y1* and *z1*.

3.6.9 Layer 8: Identifying potential binding partner; SH3#1

Agent's subject	**Function guard**	**>Message input** **Message output>**
Function: **Las17_binding_tandem_sh31_requests**		
Las17Agent	Unbound agents only	>avalable_tandem_sh31_location
Function: **output_avalable_tandem_sh32s_for_Las17**		
TandemAgent	The second SH3 domain of the agent must be unbound	avalable_tandem_sh32_location>

Each Las17 agent then executes a function almost identical to the previously executed *Las17_binding_single_requests*. The **layer 7** message list is read, and the minimum distance required to record an interaction is taken from the temporary *request* variables saved at the end of *Las17_binding_single_requests*. This effectively acts as a continuation of the aforementioned function as to also analyse the first SH3 domains of TandemAgents.

The *output_avalable_tandem_sh32s_for_Las17* function is almost identical to previously discussed *output_avalable_singles_for_Las17*. Each TandemAgent with an unbound second

SH3 domain outputs a message with its id, type, a random number between zero and one, *x1*, *y1* and *z1*.

3.6.10 Layer 9: Identifying potential binding partners; SH3#2

Agent's subject	**Function guard**	**>Message input** **Message output>**
Function: **Las17_binding_tandem_sh32_requests**		
Las17Agent		>avalable_tandem_sh32_location binding_ticket>

Once again, Las17 agents execute code almost identical to the previously executed *Las17_binding_single_requests and Las17_binding_tandem_sh31_requests* functions. The **layer 8** message list is read, and the minimum distance required to record an interaction is taken from the temporary *request* variables saved at the end of the **layer 8** function, *Las17_binding_tandem_sh31_requests*. This effectively acts as a continuation of the previous two functions as to also analyse the second SH3 domains of TandemAgents.

A section of code is then executed which checks if any of the binding motifs have identified an unbound agent within the interaction radius. If so, these motifs proceed to bind the identified domains and output a message detailing the interaction.

3.6.11 Layer 10: Updating binding partners.

Agent's subject	**Function guard**	**>Message input** **Message output>**
Function: **Las17_binding_tandem_sh31_requests**		
SingleAgent	Agents must be unbound to Las17	>binding_ticket
Function: **output_avalable_tandem_sh32s_for_Las17**		
TandemAgent	Agent must have at least one unbound SH3 domain	>binding_ticket

Unbound SingleAgents check if they have been bound by a Las17 binding motif using the output from the **layer 9** messages. Agents that can identify themselves in the message list then proceed to update their binding variables to reflect the newly bound status.

Meanwhile, TandemAgents also use the **layer 9** message list to check if any domains have been bound by Las17. Agents that can identify themselves proceed to update their appropriate binding variables. This includes updating their *binding_state* variable to either 1 (just the first SH3 bound), 2 (just the second SH3 bound), or 3 (both SH3s bound).

3.6.12 Layer 11: Identifying unbinding interactions.

Agent’s subject	Function guard	>Message input Message output>
Function: **Actin_unbinding_requests**		
ScaffoldAgent	Agent must be unbound to Las17	freed_actins>
Function: **Las17_unbinding_requests**		
Las17Agent		>avalable_single_location
Function: **Filament_unbinding_requests**		
Las17Agent	Agent must be bound to a filament (identified through bm4_partner_id)	freed_seeds>

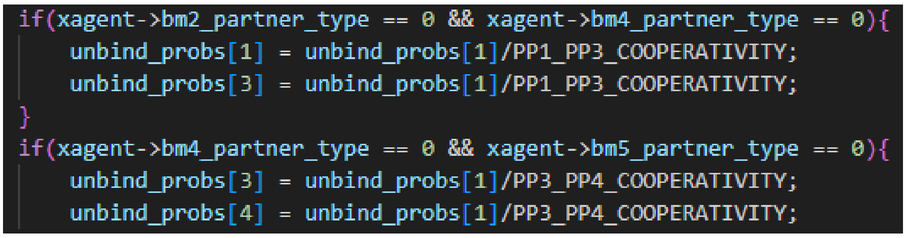
Actin filaments check whether an unbinding event will occur by generating a random number and testing whether it is smaller than the *FILAMENT_POFF* variable. Successful unbinding saves the value of *point_actin_no* into a temporary variable (*point_no*), reduces the filaments *size* variable by one, recalculates the k movement variables, and increases the *point_actin_no* variable by one. An “if” statement checks whether the filament size drops below three subunits. If this is detected, complete dissociation of the filament occurs and the ScaffoldAgent is “killed” (i.e., removed from the simulation).

The subunit number of any unbinding agents (via the temporary *point_no* variable) is then released as a message. If complete filament dissociation occurred, then the two remaining subunit numbers are also released.

Las17Agents check whether an unbinding event will occur this timestep by generating a random number and testing whether it is smaller than the appropriate *unbindProb* variable generated during the binding layers (**layers 7-9**). Successful unbinding releases the bound domain and writes the ID of the unbinding agent into a message. If the agent attempting to be unbound is an actin agent, then the cooperativity will be checked as shown in **Code 3.5**. Cooperativity functions within this model by decreasing the probability of unbinding.

Code 3.5

.



```
if(xagent->bm2_partner_type == 0 && xagent->bm4_partner_type == 0){
    unbind_probs[1] = unbind_probs[1]/PP1_PP3_COOPERATIVITY;
    unbind_probs[3] = unbind_probs[1]/PP1_PP3_COOPERATIVITY;
}
if(xagent->bm4_partner_type == 0 && xagent->bm5_partner_type == 0){
    unbind_probs[3] = unbind_probs[1]/PP3_PP4_COOPERATIVITY;
    unbind_probs[4] = unbind_probs[1]/PP3_PP4_COOPERATIVITY;
```

Las17Agents bound to an actin filament check a random number against the probability of filament unbinding (global *SEED_UNBIND_PROB* variable). If this number is lower than the unbinding probability, then an unbinding event occurs, and a message is written containing the ID of the actin filament.

3.6.13 Layer 12: Updating unbinding partners.

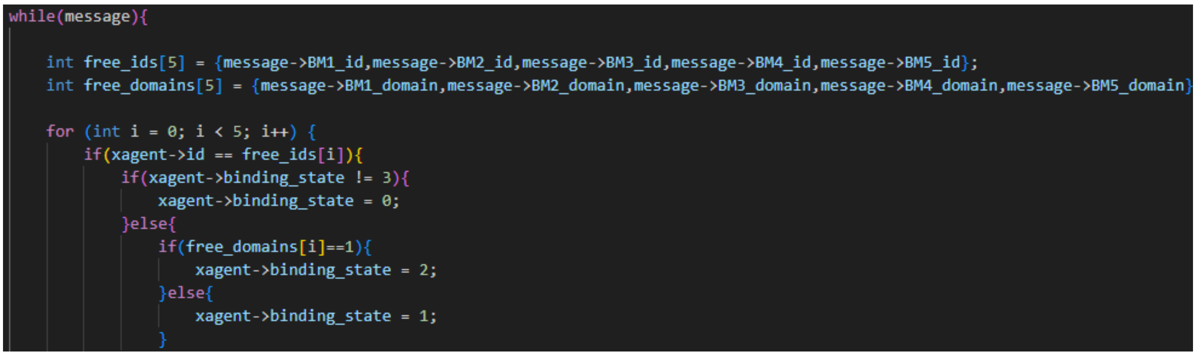
Agent's subject	**Function guard**	**>Message input** **Message output>**
Function: **unbinding_actin_from_filament**		
SingleAgent	Agent must be bound to a filament	>freed_actins
Function: **unbinding_singles**		
SingleAgent	Agent must be bound to a Las17Agent and not incorporated within a filament	>freed_domains
Function: **unbinding_tandems**		
TandemAgent	Agent must be unbound	>freed_domains
Function: **unbinding_filament_from_Las17**		
ScaffoldAgent	The agent must be bound	>freed_seeds

Actin agents that are located within a filament read the *freed_actins* message list of **layer 11**. This list contains the subunit numbers of all actin agents which unbound a filament this timestep and so are compared against the agent's *filament_subunit_number* variable. If a match is found, an unbinding event has occurred, and the agent becomes unbound.

Bound SingleAgents read the *freed_domains* message list of **layer 11**. If the ID of this agent and a message match, an unbinding event has occurred, and the agent becomes unbound.

Bound TandemAgents read the *freed_domains* message list of **layer 11**. This list contains the IDs of all agents which unbound from Las17 this timestep. If the ID of this agent and a message match, then an unbinding event has occurred. The exact effect of this unbinding is dependent upon the bound status that the TandemAgent previously occupied as shown in **Code 3.6**. In summary, the agent will adopt state 0, 1, or 2 depending upon the how the SH3 domains are bound.

Code 3.6



```
while(message){

    int free_ids[5] = {message->BM1_id,message->BM2_id,message->BM3_id,message->BM4_id,message->BM5_id};
    int free_domains[5] = {message->BM1_domain,message->BM2_domain,message->BM3_domain,message->BM4_domain,message->BM5_domain}

    for (int i = 0; i < 5; i++) {
        if(xagent->id == free_ids[i]){
            if(xagent->binding_state != 3){
                xagent->binding_state = 0;
            }else{
                if(free_domains[i]==1){
                    xagent->binding_state = 2;
                }else{
                    xagent->binding_state = 1;
                }
```

Bound ScaffoldAgents read the *freed_seeds* message list of **layer 11**. This list contains the IDs of all filament agents which unbound from Las17 this timestep. If the ID of this agent and a message match, then an unbinding event has occurred, and the agent becomes unbound.

3.6.14 Layer 13: Variable update and nucleation

Agent's subject	**Function guard**	**>Message input** **Message output>**
Function: **register_binds_and_nucleate**		
Las17Agent		nucleation_ticket>

Reporter variables used for result plotting are set. If three actin monomers are identified to be simultaneously bound to the same Las17 agent, then a nucleation event occurs. This generates a ScaffoldAgent to hold the three actin agents which formed the seed.

3.6.15 Layer 14: Updating nucleating partners.

Agent's subject	**Function guard**	**>Message input** **Message output>**
Function: **register_nucleation**		
SingleAgent	Only executed for bound agents of type 0 (actin)	>nucleation_ticket

Actin agents check if their IDs are in the *nucleation_ticket* message list. Their inclusion shows that the agent has been incorporated into a filament agent via nucleation. The appropriate variables are updated to reflect this including the *filament_subunit_number*, which is set according the Las17 binding motif this agent is bound to. Filaments are assumed to grow towards the rest of the Las17 polyproline sequence due to the presence of F-actin binding sites. Therefore, actin agents bound to PP1 are placed at the pointed end whilst actin agents bound at PP3 are placed that the barbed end.

3.6.16 CPU code

Non-parallelised code can be executed at the end of each iteration on the CPU. This is defined at the bottom of the C file and can generate an output csv file. The CPU code of this model writes various simulation statistics into the output csv including (but not limited to): the percentage of actin within a filamentous state, the percentage of each binding motif bound by an SH3 domain, and the percentage of PP3 sites bound by Sla1.

3.7 Python scripts

A python script was built to generate the XML initiation file required for FLAME GPU 1 simulations (XMLGenerator). This script was used to define global variables – including binding affinities – as well as the simulation agents and their concentrations. A second script was created using a standard python library (matplotlib) to graphically visualise the simulation while a third was written to display the csv output file. The latter two python scripts are not described further within this chapter for the reasons of brevity.

3.7.1 XMLGenerator walkthrough

The script accepts a single command-line argument which must be a cubed integer (*N*). This number defines the quantity of Las17 agents in the system. A continuous simulation space is then built as a 3-dimensional cube using **equation 3.12** with *axis_base* using units of nm and thereby dictating the concentration of Las17 peptides. The most frequent value used was 176.7 as this cubed is equal to the volume/molecule in a 300 nM solution – the standard concentration of Las17 used by the Ayscough lab *in vitro*.

$$axis\ length = \sqrt[3]{\mathrm{N}} * \ axis\_base \tag{3.12}$$

This is immediately followed by the number of each non-Las17 agents. These include CloudSH3s, Ysc84, Sla1, Bzz1, Actin seeds, and G-actin. The values chosen were calculated manually using the ratio of each agent vs Las17 agents to generate a ratio difference. This difference was then multiplied by the concentration of Las17 to give an agent concentration. Values used for Model Alpha were taken from standard *in vitro* assays used by the Ayscough lab (e.g., 300-900 nM Sla1, 1-3 µM cloud SH3s, and 5 µM actin) A G-actin concentration was also given separately as a variable for later calculations.

K_d values for each possible pairwise agent interaction are then defined using either experimental data, or predictions. Association rate constants (k_{on}) and dissociation rate constants (k_{off}) are calculated using **equations 3.13 and 3.14** respectfully. These values are given directly in cases where the k_{on} and k_{off} values of an interaction are known.

$$k_{on} = \mathrm{konBASE} \ \div \ (\frac{K_\mathrm{d}}{\mathrm{kdBASE}}) \tag{3.13}$$

$$k_{off} = \mathrm{koffBASE} * (\frac{K_\mathrm{d}}{\mathrm{kdBASE}}) \tag{3.14}$$

Equations 3.9 (see **section 3.4**) and **3.15** are used to convert the k_{on} and k_{off} values into probability functions (termed *konP* and *koffP* respectively). Calculated probabilities are then placed into arrays ready for generating the XML.

$$koffP = k_{\text{off}} * \text{timeStep} \tag{3.15}$$

The XML file is first initiated, global variables set and each agent defined. Agents are given a unique ID number, variable values specific to the agent type and random vector coordinates.

One of these agent-specific variables is the movement scaling coefficient, *k*. This variable is multiplied by a normalised random number during movement functions to calculate a movement distance (shown in **section 3.6.2**). The diffusion coefficient for molecules (*D*) can be used to predict the mean squared displacement ($<x^2>$) at a given time (*t*) as shown by the Einstein–Smoluchowski equation (**equation 3.16**) (Einstein, 1905).

$$< x^2 > = 6\text{Dt} \tag{3.16}$$

By varying *k*, the diffusion of agents within the simulation can be scaled such that it is consistent with the mean squared displacement predicted using the diffusion coefficient. Agent-specific *k* variables were identified and used when generating the agents. An example output during this scaling process is shown below (**figure 3.5**). An overlap of the diffusion coefficient derived curve and Model Alpha generated rate indicated a correct k value.

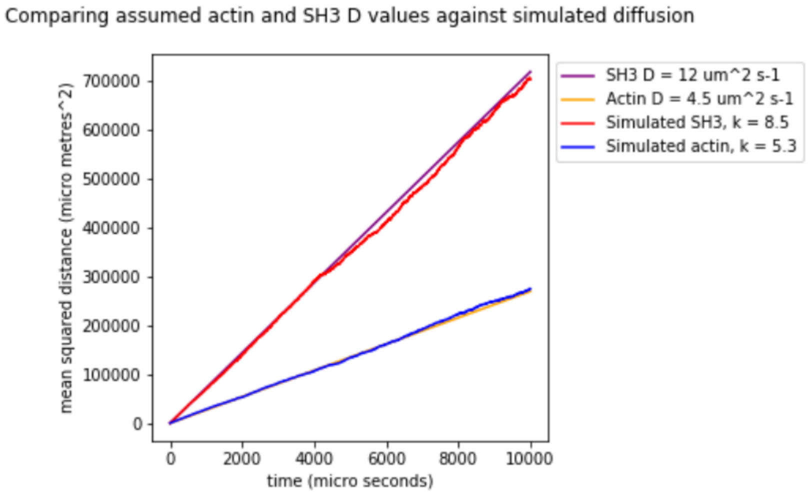


Figure 3.5. Determining values of the movement scaling factor, k. 1000 objects were simulated diffusing from a single point as the mean squared distance plotted over time. This linear relationship was scaled using the scaling factor k until the output matched the relationship expected using predicted diffusion coefficients. This figure shows k values of 8.5 and 5.3 suitably simulating the diffusion coefficients of a single SH3 domain and actin monomer respectively.

Diffusion coefficients were taken from the literature where possible and estimated when not. Actin was simulated with a diffusion coefficient of 4.5 µm^{2}s^{-1} as this value sits in between the experimentally determined values of 5.8 µm^{2}s^{-1} and 3.1 µm^{2}s^{-1} (McGrath *et al.*, 1998). SH3 domains have a diffusion coefficient of around 80 µm^{2}s^{-1} in water (Rothe *et al.*, 2016). D values for proteins less than 30 kDa are typically 4-20 times lower in the cytosol (Verkman, 2002), therefore the range of acceptable cytosolic diffusion coefficients were between 4 and 20 µm^{2}s^{-1}. A value of 12 µm^{2}s^{-1} was chosen as the diffusion coefficient for all single SH3 domains simulated.

The diffusion coefficients of all other agents were calculated using the Stokes–Einstein equation (**equation 3.17**) which states that diffusion coefficient of an object is directly related to the inverse of its radius (Miyamoto and Shimono, 2020). Here, k_B is the Boltzmann constant, T is the absolute temperature, r is the radius of the object and η_0 is the solvent viscosity. Considering that radius is approximately equal to the cube root of the molecular weight, this relationship was used to predict the diffusion coefficient of the remaining proteins (Kunitz, Anson and Northrop, 1934).

$$D = \frac{\mathrm{kBT}}{6\pi r \eta_0} \tag{3.17}$$

3.8 Code validation

Following model construction, a series of simulations were performed to validate that the model functions as intended. Four of these validations are shown below and investigate movement bias (**figure 3.6a**), the dissociation rate constant (**figure 3.6b**), the equilibrium dissociation constant (**figure 3.6c**), and the rate of filament polymerisation (**figure 3.6d**). These simulations revealed a high degree of consistency and thus validated functionality of the model. This was an important undertaking as it reaffirms that relationships between given variables and model results are sufficiently causal and not influenced by incorrect assumptions.

Visual observation of the simulation was periodically performed using a python script which generated a 2D movie (looking down through the Z axis) from the trajectory files. This was useful in providing a visual validation of phenomena such as the binding and unbinding processes. An example validation movie is given in the **supplementary materials** (movie_1).

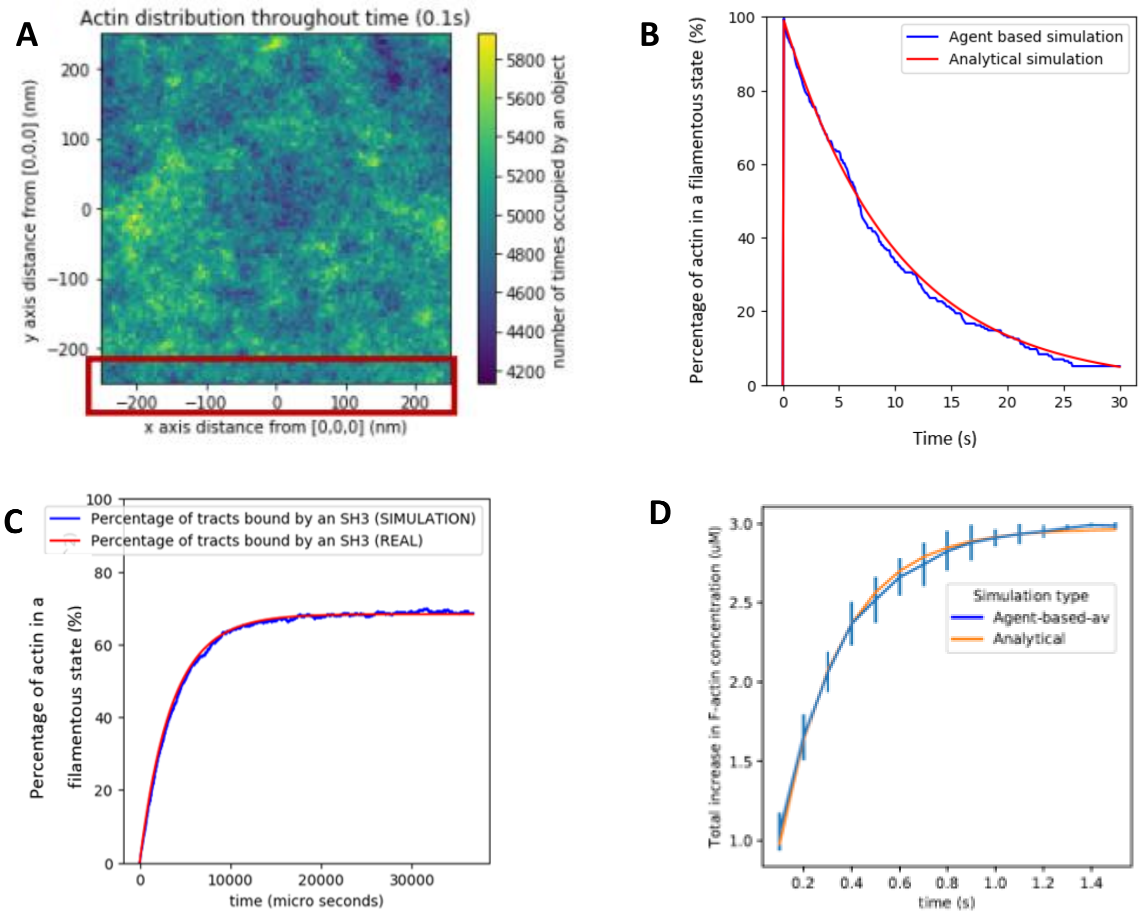


Figure 3.6. Validating model alpha. Several simulations were conducted to validate Model Alpha. **A**) A heat map produced using 5 x 5 nm bins, 100,000 agents and a time step of 1 µs. The plot increased the distribution counter of a bin by 1 for every agent located within at the end of each timestep. This demonstrated no obvious distribution bias with no population difference between the boundary and the rest of the simulation space (highlighted in red). **B**) A correct rate of unbinding (k_{off}) was observed in a simulation starting with 100% complex saturation and falling to zero (achieved by setting k_{on} as 0) demonstrating that the unbinding phase functions correctly (number of Las17 agents = 900). The figure compares the percentage of bound Las17 agents within Model Alpha (blue) against the percentage of bound Las17 agents predicted using a simple analytical model **C**) The same test was performed with the inclusion of a k_{on} value > 0. The resultant curve was highly consistent with the analytical prediction suggesting good functionality of the model (number of Las17 agents = 8,000). **D**) A 0.4 µM concentration of seeds and 3.0 µM concentration of G-actin was entered into the simulation at time 0. The percentage of actin agents occupying a filamentous state was plotted against time for both Model Alpha (blue “Agent-based-av” line) and an analytical model (orange line) derived from literature filament polymerisation rates (Pollard, 1986). The agent-based program reproduced the experimentally realistic behaviour accurately over 1.5 seconds (time stepping of 1 µs) across three simulation runs (error bars ranged by +/- the SEM) demonstrating functionality of the polymerisation phase.

3.8 Code derivations

Two major model derivations were written to function as specialised sub-models. First was the “Pyrene Simulator”. This model removed the simulations movement stages and instead used an analytical binding equation to bind agents to Las17. This decreased runtime significantly at the expense of fidelity. Pyrene Simulator was used to extrapolate simulated binding curves to allow for their completion. This was done by accepting the nucleation and polymerisation rates determined by the target simulation. This model was useful when comparing experimental and computational results due to the excessively long runtime required to generate polymerisation curves consistent with experimental pyrene traces.

The second specialised derivative was the MST Simulator model. This model removed the polymerisation component and instead plotted the percentage of Las17 (target) bound by objects (ligands) across six different ligand concentrations. The concentration of ligands was dynamically increased throughout the simulation by only interacting should their ID number be within a certain value range. This range was then extended after predefined time intervals. MST Simulator reproduces the output of Microscale Thermophoresis assays to help better explore their results.

4. Results Chapter: Model Alpha

4.1 Identifying Las17 purification issues

The first step in acquiring experimental results for the computational model was to purify key peptides of interest. Model Alpha outlined in **chapter 3** represents Las17 as the first four polyproline tracts. This is the minimum nucleating region and can be expressed as a peptide covering residues 300 to 422 (E Allwood and K Ayscough, personal communication). All the Las17 peptides expressed contained either a C-terminal His purification tag (termed 300-422-His) or an N-terminal GST purification tag (termed either GST-300-422 or simply 300-422 following cleavage of the tag). A larger Las17 peptide, covering all residues C-terminal of residue 300, was also expressed (termed 300-633 I555D). This construct covers eight polyproline tracts and the WCA region. To help direct investigation to the polyproline region, an I555D mutant was used which changed residue 555 in the WH2 domain to an aspartate to prevent actin binding via the WCA region (Tyler *et al.*, 2021). Three major hurdles had to be addressed when purifying Las17 including a co-purified protein, an inconsistency in Las17 gel staining and a loss of peptide during purification.

4.1.1 Presence of a 70 kilodalton contaminant

Las17 was frequently linked with a 70 kDa contaminant when purified from *E. coli* as routinely observed by the Ayscough lab. However, this 70 kDa band was not observed when purifying the protein within yeast (E Allwood, personal communication). Therefore, it is assumed that this interaction may not be relevant in an *in vivo* environment, but rather a consequence of expression within *E. coli*.

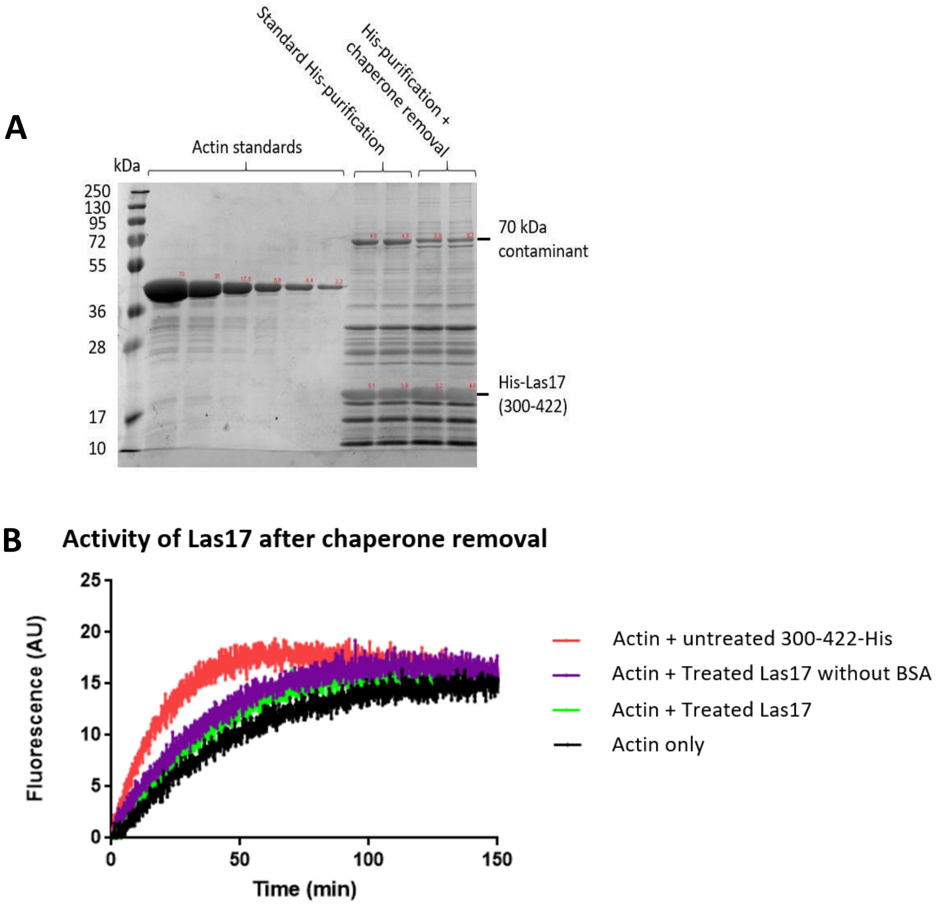


Figure 4.1: Chaperone removal. Investigating the efficacy of chaperone removal. **A**) Two identical lanes were run for His-Las17 (300-422-His) purified using the standard His-purification method and His-Las17 (300-422-His) purified using the His-purification protocol modified to remove chaperone (**chapter 2, section 2.4.7**). The ratio of 70 kDa contaminant to Las17 was reduced fivefold between the two methods. Note that the concentration of Las17 estimated by the gel was far lower than expected due to poor staining. **B**) His-Las17 (300-422) had a positive effect on actin polymerisation (red) relative to the actin only control (black) as measured by the pyrene assay. Removal of the chaperone reduced this activity gain substantially (purple). A control using the same conditions as the chaperone removal protocol without the addition of denatured BSA (green) gave a similar reduction in activity. This suggests that the change in activity relative to untreated Las17 (red) may be the result of time at 37°C.

The ~70 kDa contaminant was assumed to be a chaperone due to its size being consistent with Hsp70 – a common chaperone and actin paralogue. The concentration of Hsp70 was reduced five-fold relative to Las17 following the addition of ATP and urea-denatured protein (BSA) as detailed in the methods (**chapter 2**, **section 2.4.7**) (**figure 4.1a**). This change to the protocol did not increase the yield of purified Las17 following SEC and reduced the activity of

300-422 (**figure 4.1b**). A control which omitted the addition of denatured BSA, yet otherwise undertook the same protocol for chaperone removal, observed the same activity reduction suggesting that other factors such as 30 minutes at 37^{o}C were detrimental to the activity of Las17. The activity of Las17 in pyrene assays has always been variable and is sometimes even observed to produce a biphasic polymerisation curve (E Allwood, personal communication). This variability occurs regardless of the experimentalist and has never been explained leading to the conclusion that Las17 is an inherently unstable protein. Considering the high ratio of Las17 to contaminant (between 20:1 and 10:1 as estimated using gels), the lack of complete Hsp70 removal, and the reduction in control activity, future Las17 preparations did not involve removal of the chaperone.

4.1.2 Inconsistent staining of Las17 peptides

A problem encountered when purifying Las17 was a difficulty to reliably stain Las17 within a polyacrylamide gel. Individual staining methods for polyacrylamide gels are described in **chapter 2, sections 2.4.10 – 2.4.15**. Peptides solely covering residues 300-422 do not contain any tryptophan or tyrosine residues making A280 readings for this peptide unreliable – especially considering the frequent contaminant identified in **section 4.1.1** which likely absorbs considerably more strongly than Las17 at this wavelength. Consequently, measuring the relative intensities of peptide bands within gels proved to be a more reliable method.

Although Coomassie staining initially appeared effective, the staining was quickly lost from the Las17 band within the gel destaining steps relative to other protein bands. This was a frequent observation for Las17 purifications and is known within the Ayscough lab to affect multiple Las17 constructs, not only 300-422. Removing methanol from the destaining solution – a method employed by the Williamson lab when encountering similar difficulties with proline-rich salivary proteins – did not prevent the excessive destaining (**figure 4.2a**).

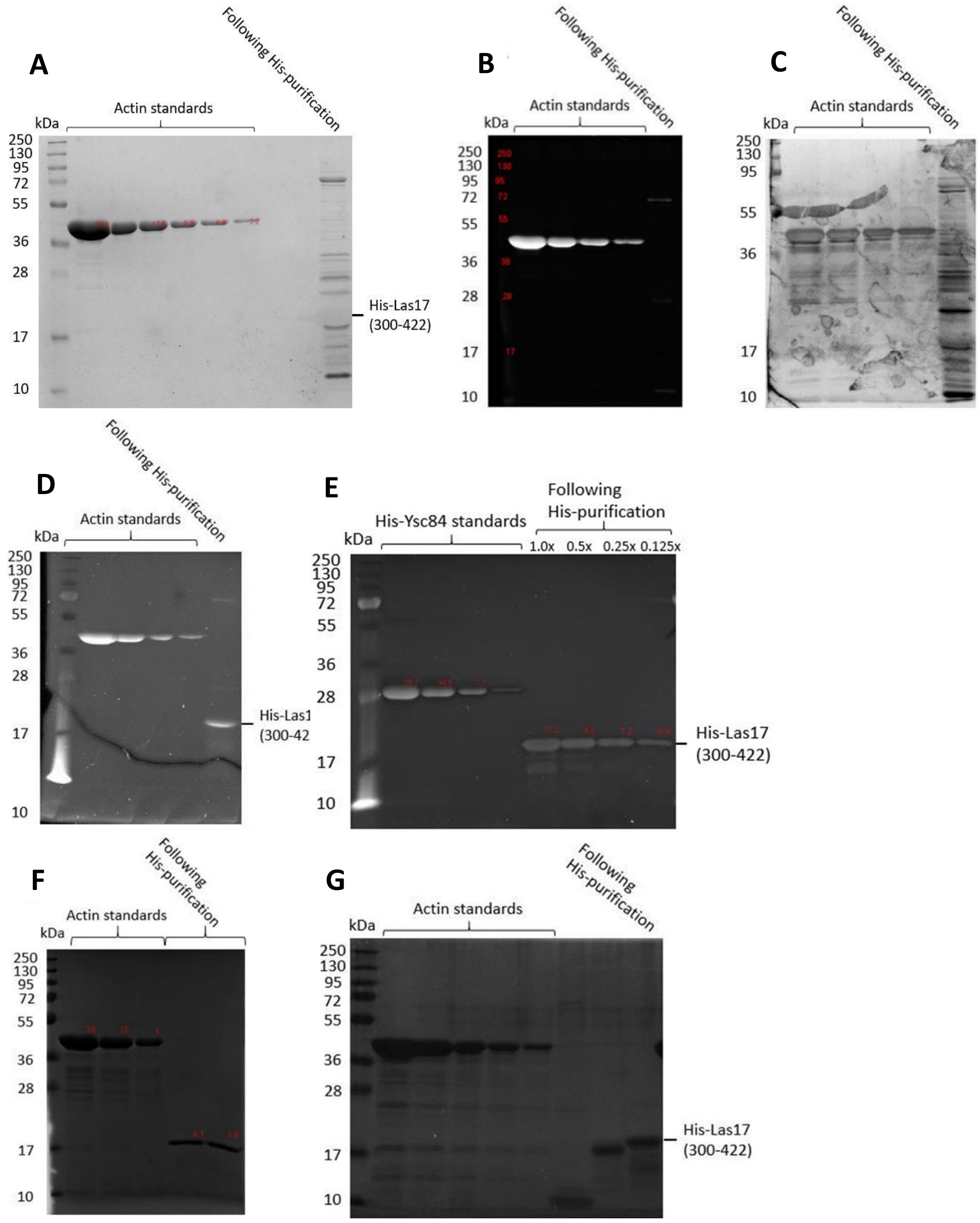


Figure 4.2: Las17 in-gel staining A) A Coomassie gel of nickel-affinity purified His-Las17 (300-422-His). The gel was destained for 2 hours, during which the Las17 band was almost entirely lost. Staining gels with **B**) SYPRO Ruby, **C**) silver stain, **D**) turmeric or **E**) in-gel His-stain did not give consistent concentration values due to poor staining. His-Ysc84 was used as the concentration standard when using the His-stain as this method binds His-tags to dye molecules and thus a His-tagged standard was required. Both **F**) lightly destained Coomassie (destained for 30 minutes) and **G**) zinc-imidazole negative stained gels gave similar concentration values of 10 and 11.8 µM respectively and showed limited loss in band signal.

Several alternative staining methods were investigated to identify a more reliable method of determining protein concentration. No Las17 band was observed when using SYPRO Ruby (**figure 4.2b**) or Silver Staining (**figure 4.2c**) gels whilst turmeric (**figure 4.2d**) and in-gel His-stain (**figure 4.2e**) both resulted in an unacceptably low degree of initial staining for the Las17 band. Ultimately, two methods were identified which gave the most reliable staining.

The first was zinc-imidazole negative staining which gave a reliably stable Las17 band. Second was using a modified Coomassie staining protocol that limited destaining to just 20 minutes. Out of these, negative staining was the most reliable method identified. However, the modified Coomassie protocol proved to be a more practical method when staining routine gels due to the increased ease-of-use.

4.1.3 Loss of the 300-422 peptide during purification

A frequent observation encountered throughout this thesis was the heavy aggregation of Las17 preparations. Purifications steps that employed column chromatography (e.g., size exclusion or buffer exchange methods) resulted in substantial loss of the peptide. The amount of Las17 lost within columns, along with the peptide's proclivity to aggregation, varied greatly from a small, yet noticeable, yield reduction to the loss of entire preparations. This made changing between buffers challenging.

The ProtParam tool predicts a small degree of A280 absorbance (extinction coefficient of 2980 M^{-1} cm^{-1}) despite 300-422 not containing any tryptophan or tyrosine residues. While this absorbance doesn't allow for an accurate calculation of concentration (due the variable presence of a ~70 kDa co-purified protein), it can be used in size exclusion chromatography to identify the point of Las17 elution. It was concluded that Las17 exits size exclusion columns in the 16.5 mL fraction at around 35 kDa (**figure 4.3**). This value is approximately double the size of the 300-422 peptide (~15 kDa) and suggests either heavy interaction been Las17 and the dextran-agarose column (SuperdexTM 200 HR 10/30 from Pharmacia Biotech) or self-interaction between peptides. A self-interaction is supported by the observation of Las17 aggregation following purification.

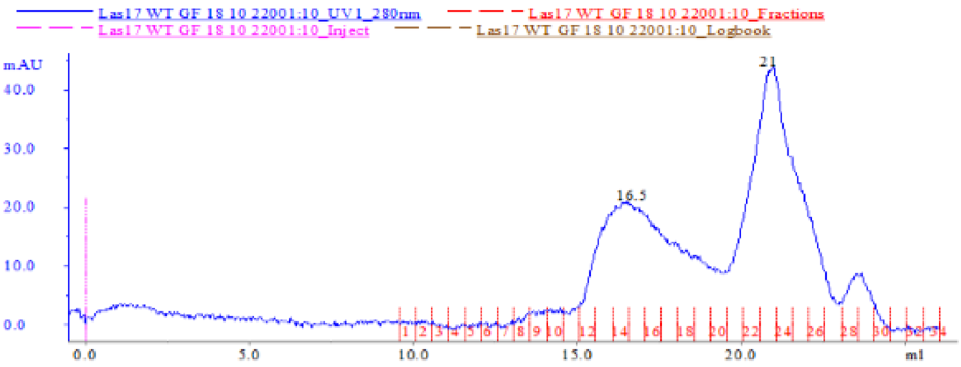


Figure 4.3: Size exclusion chromatography A280 profile. A280 readings from the fraction collector of a SuperdexTM 200 HR 10/30 size exclusion chromatography column. Purified Las17 (300-422-His) was injected into the column (buffer was degassed PBS) and the size exclusion protocol from **chapter 2** was followed. A peak around the 16.5 mL fraction coincides with eluted Las17 and suggests a size of around 35 kDa.

4.2 Determining interaction affinities

An important part of this thesis was to obtain the binding affinities for key interactions where those values are either unknown or underexplored. This helped to reduce the number of model assumptions. Two major techniques were employed when constructing Model Alpha – the name used to describe the first ABM constructed as part of this thesis. These were Microscale Thermophoresis and estimations using Model Alpha.

Sla1 and Las17 interact as part of the SLAC complex within the cytosol and colocalise at the endocytic patch around the same time (Feliciano and Di Pietro, 2012). Therefore, measuring the binding affinity between these two proteins was a key objective for this chapter. Sla1 interacts with Las17 using three SH3 domains. The N-terminal pair of domains are directly adjacent, with no linker separating them and have been shown in the literature to bind Las17 in a manner that competes with actin-binding. Therefore, the main aim was to determine individual affinities for the two N-terminal SH3 domains (termed Sla1-SH3#1 and Sla1-SH3#2) along with the affinity of the tandem SH3 construct (termed Sla1-SH3#1-2). The expectation was that that the peptide covering both domains in tandem would bind Las17 with a significantly higher affinity than the individually expressed domains would otherwise indicate. This is due to the spatial cooperativity (i.e., avidity) between the binding sites.

4.2.1 Microscale Thermophoresis

Microscale Thermophoresis (MST) was the first technique employed to determine binding affinities due to its low sample consumption – an important consideration considering the difficulty of purifying Las17 (along with several of the SH3 constructs). This technique involves titrating one ligand against a second, fluorescently labelled, protein of a fixed concentration. Each concentration combination is drawn into a separate capillary and an infrared laser (IR-laser) is applied to each in succession. As the IR-laser is applied, the fluorescently labelled protein diffuses either away from or, in some cases, towards the site of heating. This observation is called the Ludwig-Soret effect. The Soret coefficient (S_T) determines the measurable ratio of Fhot/Fcold, with Fhot being the fluorescence after application of the IR-laser and Fcold the fluorescence prior to the IR application. S_T is directly proportional to the diffusion coefficient (D) of the diffusing protein as shown by **equation 4.1**. Therefore, the degree of binding can be observed across the titration (via its effect on D) which produces an easily measurable binding curve (Duhr and Braun, 2006; Jerabek-Willemsen *et al.*, 2011).

$$S_T = \frac{D_T}{D} \tag{4.1}$$

4.2.1.1 Labelling for MST

The first stage of MST is to attach a fluorescent marker to one of the two proteins of interest. Las17 was chosen as the labelled partner for two reasons; (1) all the major binding interactions in the model included this protein as one of the interaction partners and (2) Las17 possesses multiple binding sites and so having this as the labelled protein made the subsequent analysis easier.

4.2.1.2 Labelling Las17

Numerous labelling methods were carried out with an overview given in **table 4.1** and a more detailed description provided by this section.

The first method used was to covalently link the fluorophore to a lysine on Las17. At the time, lysine labelling was the method of choice in the lab for actin-labelling so served as a good starting point for exploring MST. Success in labelling was determined via the degree of labelling (DOL). This is the percentage of molecules bound to the fluorophore after the labelling stage and should ideally be over 80%. Lower DOL values were acceptable should the signal provided within the subsequent MST assays be sufficient for low noise data.

A single lysine was present in the peptide linker connecting the His-tag and Las17-300-422 (expressed as a single Las17(300-422)-His construct). Unfortunately, <10% of peptides could be labelled using this technique during the labelling protocol. A degree of labelling around 0% was observed for the longer, WH2-domain inactivated, Las17(300-633)-His (I555D) peptide despite possessing 6 lysines in the C-terminal half of the peptide in addition to the His-tag linker lysine.

Las17 construct labelled	Labelling position	Labelling method	Labelling Success	Ligands tested against
300-422-His^1	C-terminus	Lysine	Unsuccessful (<10% labelling)	Sla1-SH3#1, Sla1-SH3#1-2^2
300-422-His^1	C-terminus	His-tag	Successful	Sla1-SH3#1, Sla1-SH3#1-2^2
300-633-His (I555D) ^{1}	C-terminus	Lysine	Unsuccessful (~0% labelling)	N/A
300-633-His (I555D^1) ^{1}	C-terminus	His-tag	Successful	Sla1-SH3#1, Sla1-SH3#1-2^2
300-422-His (Q300C) ^{1}	N-terminus	Cystine	Successful	Sla1-SH3#1, Sla1-SH3#1-2^2
300-422-His (Q300C) ^{1}	C-terminus	His-tag	Successful	Sla1-SH3#1, Sla1-SH3#1-2^2
SNAP-300-422-His^1	N-terminus	SNAP-tag	97% DOL	Sla1-SH3#1, Sla1-SH3#1-2^2
SNAP-300-422-His^1	C-terminus	His-tag	Successful	Sla1-SH3#1, Sla1-SH3#1-2^2

Table 4.1. MST-labelling methods used for Las17. All labelling methods were repeated in a range of buffers (PBS pH 7.0, PBS pH 7.4, PBS pH 8.0), with a range of Tween 20 (0-0.05%) and labelling times (1x, 1.5x and 2x the protocol suggestion). The most successful result of each screen is shown in the table. Las17 peptides are associated with a subscript value of 1 (^{1}). These include 300-422-His and 300-633-His (I555D) as described in **section 4.1**. Additionally, 300-422-His (Q300C) was a mutant of 300-422-His which replaced the first residue with a cysteine for ligation to Cysteine-binding dyes. SNAP-300-422-His is the 300-422-His sequence with an N-terminal, dye binding SNAP-tag. Sla1 peptides are associated with a subscript value of 2 (^{2}). These include the first SH3 domain (Sla1-SH3#1), second SH3 domain Sla1-SH3#2, and a tandem construct covering both SH3 domains (Sla1-SH3#1-2) All peptide constructs are described in **chapter 2** (**Section 2.2**).

Reversible binding of a fluorophore directly to the His-Tag was used as another labelling technique and was successfully verified via MST (see **methods**). Two additional labelling techniques were also carried out, each targeting the N-terminus of the protein. One of these was covalent cysteine-labelling to a Q300C mutant of His-300-422 which succeeded in the labelling stage as confirmed by a gel doc observation of fluorescence. The second technique was SNAP-labelling. SNAP-tags are large, globular labelling domains often used for *in vivo* labelling, although have also been used in MST (Majkut *et al.*, 2013), as they can covalently bind a fluorophore through their thiol group. This interaction is highly specific – hence its application to both *in-vivo* and *in-vitro* assays.

Producing MST binding curves proved to be extremely challenging with Las17 as the labelled target. Lysine, cysteine, and histidine-tag labelling techniques all failed to yield observable binding curves despite the trialling of numerous protocol modifications. However, a binding curve could be obtained using SNAP-labelled SNAP-Las17(300-422)-His which will be discussed in the next section.

4.2.1.3 SNAP-Labelled Binding

As mentioned at the start of **section 4.2**, gaining affinities between Las17 and Sla1 SH3 domains was the most important experimental value required for the modelling. Therefore, an affinity between Las17 and the first domain of Sla1 was the first interaction assayed. MST binding analysis between SNAP-labelled Las17 and Sla1-SH3#1 was undertaken using a fixed concentration of Las17 and a range of 16 Sla1 concentrations (15 µM to 0.0005 µM using a series of stepwise, 2-fold titrations). A single binding curve with a K_d of 0.17 ± 0.042 µM in G-buffer could be gained (**figure 4.4a**).

Unfortunately, this binding curve could not be repeated and possessed two undesirable effects (**figure 4.4b**). These included adsorption of the labelled partner to the capillary walls at lower ligand concentrations and a ligand-dependant fluorescence change. SD-testing suggested that this fluorescence change was binding-independent. These effects are both suggested to reduce the quality of the binding curve. Nevertheless, the changing in FNorm was sufficiently large to obtain a K_d. The calculated affinity was within the range of values typically observed for SH3 domains (Demers and Mittermaier, 2009).

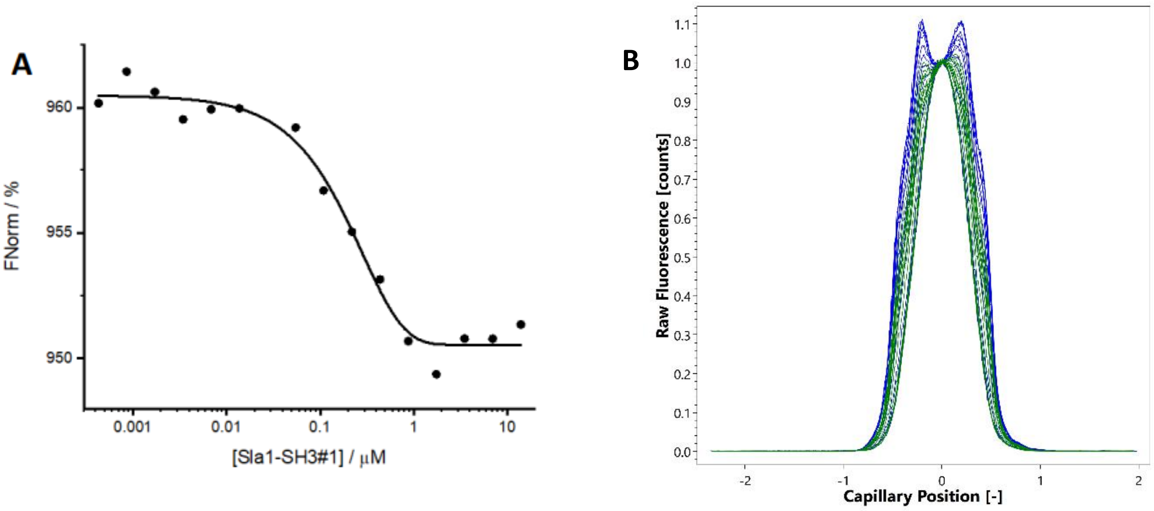


Figure 4.4: SNAP-labelled MST curve. FNorm in all graphs is calculated using the ratio Fhot/Fcold where Fhot is the fluorescence after heating and Fcold is the fluorescence before heating. **A**) A binding curve fit from a single assay between 25 nM SNAP-labelled SNAP-Las17-His (300-422) and unlabelled Sla1-SH3#1. **B**) Output files from the Monolith NT.115 reveal the issues faced when labelling SNAP-Las17 including adsorption of the labelled partner to the capillaries at lower ligand concentrations as well as a ligand-dependant fluorescence change. Capillary scans should overlap to produce a uniform curve. However, the dual peak observed here is indicative of adsorption to the capillary.

4.2.1.4 Labelling SH3 domains and actin

As a consequence of the difficulties associated with labelling Las17, it was decided to label the other protein partners for the interaction sets. The techniques explored included covalent lysine and cysteine labelling while the proteins labelled included Sla1-SH3#1, Sla1-SH3#2, Sla1-SH3#1-2, Sla1-SH3#1-3, and actin (**table 4.2** for a summary).

Construct labelled	**Labelling method**	**Labelling Success**	**Las17 peptides tested against**
Sla1-SH3#1	Lysine	Successful (65% DOL)	300-422-His, 300-633-His (I555D)
Sla1-SH3#2	Lysine	Successful (72% DOL)	300-422-His
Sla1-SH3#1-2	Lysine	Unsuccessful (~0% DOL)	N/A
Sla1-SH3#1-3	Lysine	Unsuccessful (~0% DOL)	N/A
Actin	Lysine	Successful (85% DOL)	300-422-His, 300-633-His (I555D)
Actin	Cysteine	Successful (76% DOL)	300-422-His, 300-633-His (I555D)

Table 4.2. MST-labelling methods used for non-Las17 peptides. All labelling methods were repeated in a range of buffers (PBS pH 7.0, PBS pH 7.4, PBS pH 8.0), with a range of Tween 20 (0-0.05%) and labelling times (1x, 1.5x and 2x the protocol suggestion). The most successful result of each screen is shown in the table.

The lysine labelling attempts of Sla1-SH3#1, SH3#2 and actin were all successful with a >65% degree of labelling for each. However, neither Sla1-SH3#1-2 nor Sla1-SH3#1-3 could be labelled despite each construct containing domains which could be successfully labelled when individually expressed.

In lieu of an experimentally determined Sla1 structure, SWISS-MODEL homology modelling was used to predict the locations of surface exposed lysines (Waterhouse *et al.*, 2018). Sla1-SH3#1 was modelled to the alpha chain of brain Spectrin where the three lysines in Sla1-SH3#1 were orientated towards a single face of the protein (**figure 4.5a**). Sla1-SH3#2 was modelled to the tyrosine-protein kinase Src where all four lysines in Sla1-SH3#2 were orientated towards a single face (**figure 4.5b**). Homology and AlphaFold modelling of the entire Sla1-SH3#1-2 peptide was undertaken to investigate how the lysines may be localised in the multi-domain constructs. Both modelling techniques suggest that the two domains, closely located in sequence, may directly interact with each other (**figure 4.5c and figure 4.5d**).

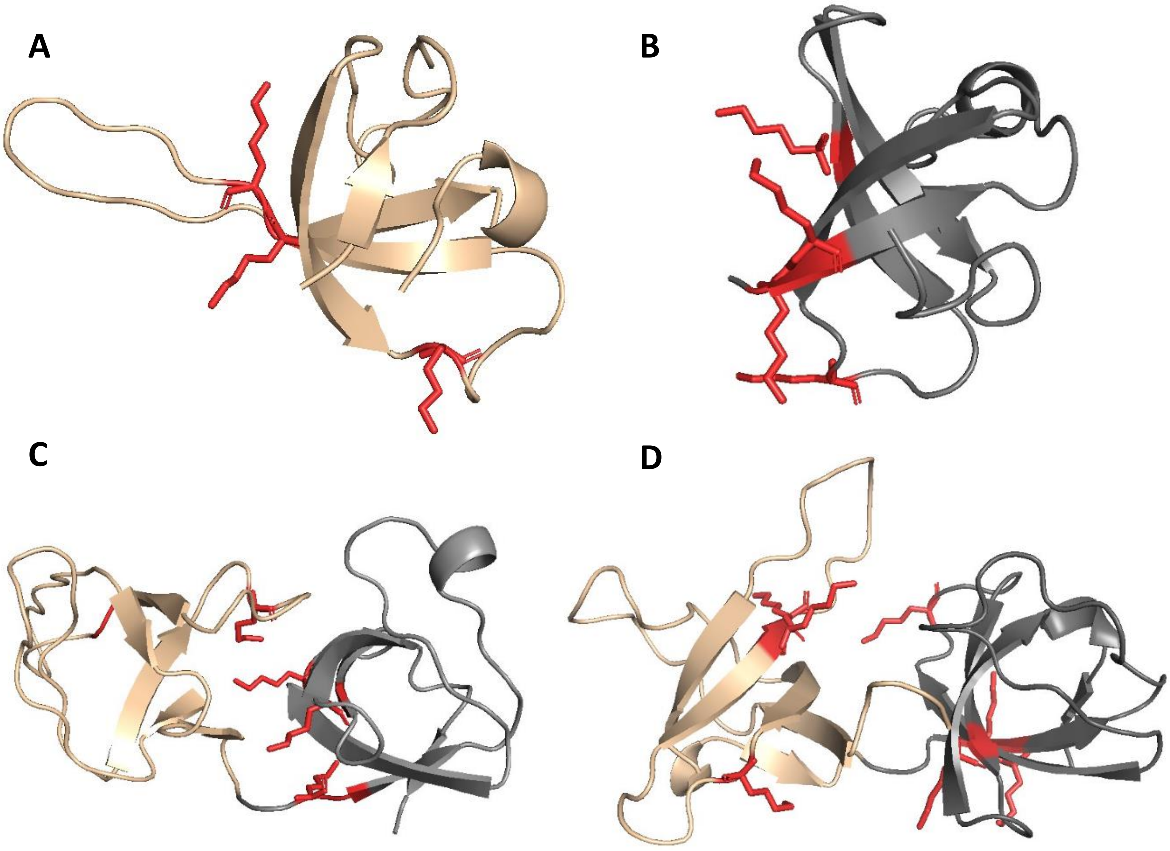


Figure 4.5. Structure predictions for Sla1-SH3 domains. Predicted structures for the SH3 domains of Sla1 are given with SH3#1 shown in tan, SH3#2 shown in grey, and lysine residues shown in red. All subfigures were made in PyMOL v4.6. **A**) SH3#1 modelled to the alpha chain of brain Spectrin (PDB: 2KR3). **B**) SH3#2 modelled to the SH3 domain of tyrosine-protein kinase Src (PDB: 2H8H). **C**) SH3#1-2 modelled to the YBL007Cp-like protein (PDB: B5VDX3) and **D**) the AlphaFold prediction of SH3#1-2 both suggest that many of the lysine residues are orientated towards their interaction interface which may shield them from labelling.

Homology modelling of Sla1-SH3#1-2 to tyrosine-protein kinase Src predicted that four of the seven lysines were oriented towards the interaction interface between the two SH3 domains (**figure 4.5c**). One of the three alternatively orientated lysines was also located close to this interface. The lysine locations are broadly consistent with AlphaFold modelling which predicts that four of the seven lysines are orientated towards the interaction interface of the SH3 domains (**figure 4.5d**) (Jumper *et al.*, 2021; Varadi *et al.*, 2022). As with the homology modelling, one of the three alternatively orientated lysines was also located close to this interface. Lysines would be expected to orientate themselves towards interfaces with complementary charges and this can be used to predict the shielded interfaces. However, Sla1-SH3#1-2 contains many Glutamate and Aspartate residues which are (generally) evenly distributed about the protein surface.

The interface-orientated lysines suggested in both the homology modelling and AlphaFold modelling involve residues from each domain. One hypothesis for the low labelling efficiency is that lysines amenable to labelling in individually expressed domains are not accessible in the Sla1-SH3#1-2 expression due to their location at the interdomain interface.

4.2.1.5 MST: Sla1

The most N-terminal pair of lysine-labelled, individually expressed, SH3 domains both produced reproducible binding curves. The folding of Sla1-SH3#1 was analysed via 1D NMR which confirmed a folded state (**figure 4.6**). Unfortunately, Sla1-SH3#2 could not be obtained at a high enough concentration and purity for NMR due to heavy precipitation and a large degree of breakdown. Spot array data from Sla1-SH3#2 shows low specificity which, taken together with the later conclusions of this chapter, suggest that this domain expression may not yield a well folded state.

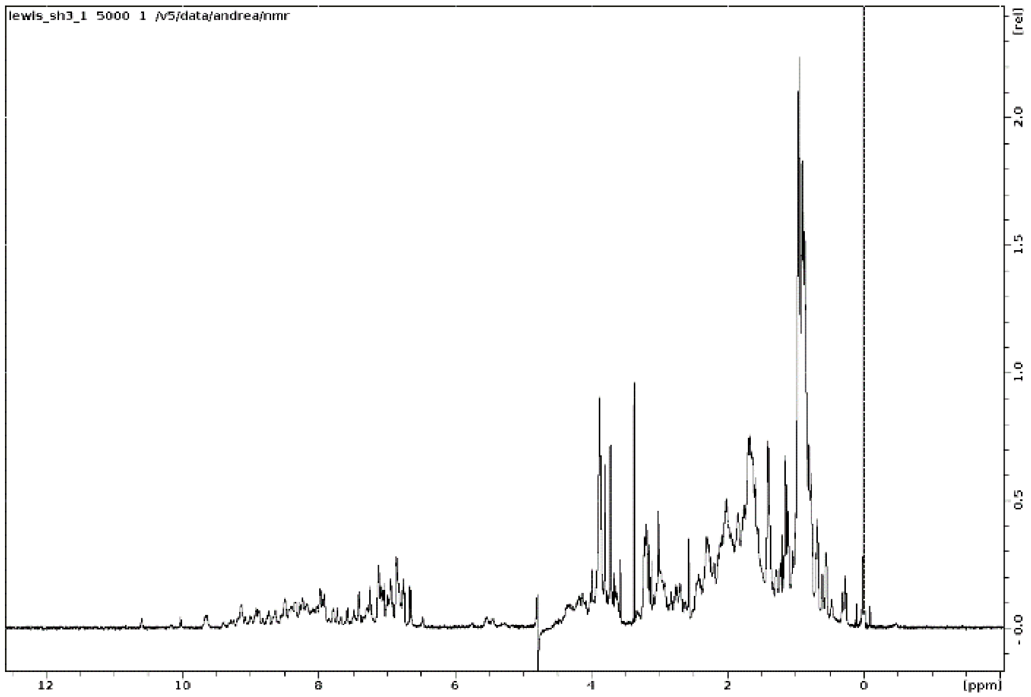


Figure 4.6. 1D NMR of the N-terminal Sla1 SH3 domain. The 1D NMR spectra for Sla1-SH3#1 shows sharp proton resonances for amide and methyl groups outside of random coil regions suggesting a well-folded state.

4.2.1.6 MST: Sla1-SH3#1

Continuing with the same rationale explained at the start of **section 3.1.3**, the affinity between Las17 and the first domain of Sla1 (Sla1-SH3#1) was determined using MST. A binding assay between Las17 and Lysine-labelled Sla1-SH3#1 was undertaken using a fixed concentration of Sla1 and a range of 16 Las17(300-422)-His concentrations (33 µM to 0.001 µM using a stepwise, 2-fold titration).

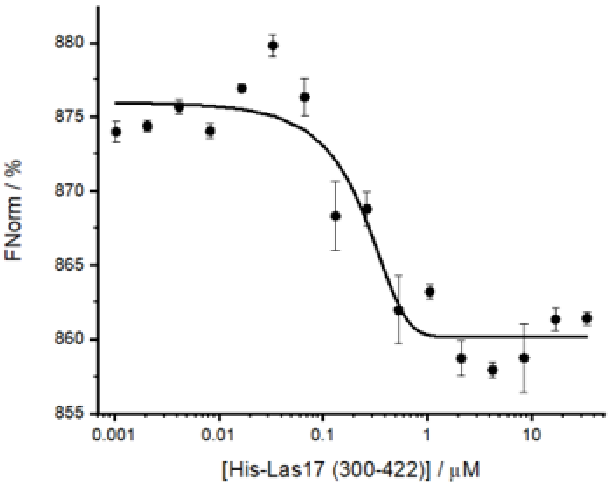


Figure 4.7: SH3#1 MST curve. FNorm in all graphs is calculated using the ratio Fhot/Fcold where Fhot is the fluorescence after heating and Fcold is the fluorescence before heating. The binding fit was made using 2 repeats between 25 nM Lys-labelled Sla1-SH3#1 and unlabelled Las17(300-422)-His.

The Sla1-SH3#1 construct gave similar binding profiles over two repeats (G-buffer) and a binding affinity of 0.25 ± 0.12 µM n=2 (**figure 4.7**). This value is consistent with the 0.17 µM measurement obtained using SNAP-labelled SNAP-300-422-His (**section 4.2.1.3**, **figure 4.4**). Visual inspection of the binding curve reveals that several of the data points deviated significantly from the sigmoidal fit (particularly between the concentrations of 0.008 µM and 0.064 µM). Although we were unable to obtain a third replicate, these data deviations from the fit may indicate the presence of multiple, competing binding events between Sla1-SH3#1 and the polyproline tracts.

4.2.1.7 MST: Sla1-SH3#2

The second domain of Sla1 (Sla1-SH3#2) was chosen as the next interaction of interest. MST binding analysis between Las17 and Lysine-labelled Sla1-SH3#2 using a fixed concentration of Sla1 and a range of 16 Las17(300-422)-His concentrations (33 µM to 0.001 µM using a stepwise, 2-fold titration) gave a reproducible binding curve in G-buffer with a binding affinity >12 µM n=3 (**figure 4.8**). The curve had not started to level off even at the highest ligand concentration used, meaning that the upper limit for the affinity is undetermined. To complete the curve would require a higher initial concentration of Las17(300-422)-His. However, Las17 has a high aggregation propensity and variable purification yield which prevented this approach. The poor binding attributed to the SH3#2 construct may be a result of poor folding as will be later explored in **chapter 6** (**section 6.11.1**).

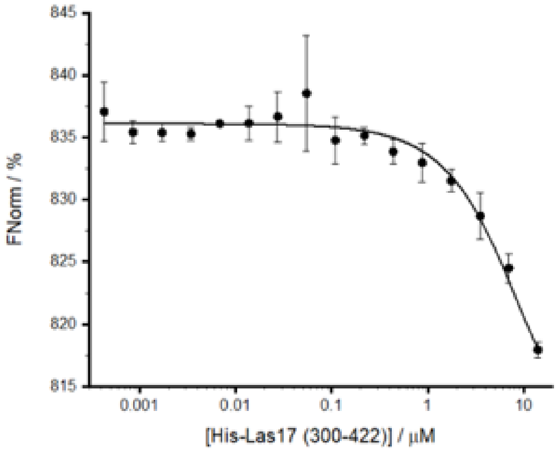


Figure 4.8: SH3#2 MST curve. FNorm in all graphs is calculated using the ratio Fhot/Fcold where Fhot is the fluorescence after heating and Fcold is the fluorescence before heating. The binding fit was made using 3 repeats between 25 nM Lys-labelled Sla1-SH3#2 and unlabelled Las17(300-422)-His.

4.2.2 Model-predicted binding affinities

An advantage of Agent-Based Modelling is that both known and unknown parameters can be fed into a model to allow complex behaviours to emerge without directed input. Changes to these behaviours can be observed by varying variable values. This was the principal method used to estimate the affinity between Las17 and Sla1-SH3#2. In the absence of a completed experimental binding curve for second SH3 domain, the model could be modified to output the percentage of Las17 bound at a variety of tandem Sla1 (Sla1-SH3#1-2) concentrations. By titrating the concentration of simulated tandems and varying an assumed affinity parameter for Sla1-SH3#2 whilst keeping the affinity of Sla1-SH3#1 constant, an affinity for the second domain can be predicted. The predicted Sla1-SH3#2 affinity is the value required to reproduce the affinity of the Sla1-SH3#1-2 tandem construct.

4.2.2.1 Using MST data to predict the affinity of Sla1-SH3#2

Interactions between Las17 and Sla1-SH31-2 were simulated using Model Alpha (Agent-based computational model outlined in **chapter 3**). The affinity between Las17 and Sla1-SH3#1 was set at 0.25 µM in accordance with the MST results in G-buffer described earlier in this chapter. The affinity for Sla1-SH3#2 was varied until the overall affinity between Las17 and the Sla1 tandem-SH3 peptide was consistent with the only identified data available at this point within the thesis (0.056 µM) (Feliciano and Di Pietro, 2012). This method predicted a Sla1-SH3#2 affinity of 1600 µM (**figure 4.9**) as this value, combined with a 0.25 µM affinity for Sla1-SH3#1, resulted in Sla1-SH3#1-2 binding Las17 at a K_d of 0.056 µM. This value broadly agreed with the conclusion drawn from MST experiments which suggested a very low affinity, as well as conclusions drawn from in-lab spot array data (internal data from the Ayscough lab).

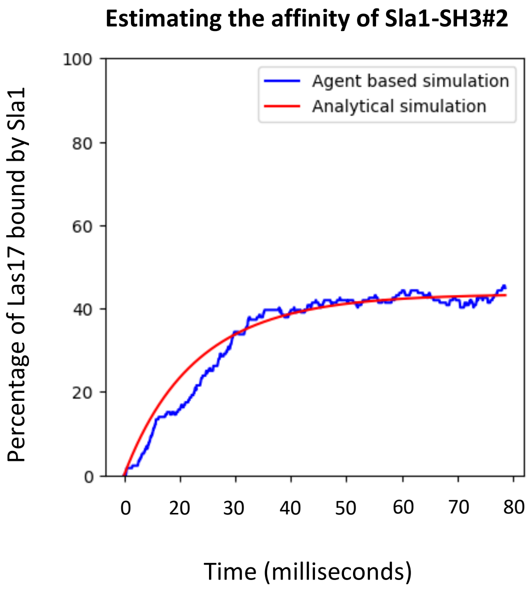


Figure 4.9: Using Model Alpha to predict the affinity of Sla1-SH3#2. The K_d for Sla1-SH3#1-2 (assumed at 56 nM) was simulated using a simple ODE model to produce an analytical binding curve (red). This was then compared to an output of Model Alpha where SH3#1 was fixed at 0.25 µM K_d and SH3#2 was varied (blue). The SH3#2 affinity required for these curves to produce a consistent binding curve (shown above) was 1600 µM.

4.2.3 Elucidating the binding positions of SH3 domains

The logical step following affinity analysis was to assess where the SH3 domains bind along the 300-422 sequence. Spot array data produced within the Ayscough lab was first used to link SH3 domains to regions of Las17. All core SH3-binding sites of the appropriate class were identified and compared against a large peptide screening study taken from the literature (Tonikian *et al.*, 2009). These comparisons were each given a score using rigid criteria chosen during the analysis. The scores attributed to each residue were then weighted by the importance of that residue position as determined by the screen results. Highly important positions were given a 2x modifier whilst important positions were given a 0.5x modifier. Combining the residue scores then generated an SH3 preference prediction for each site binding to Las17. These predictions were not used to compare SH3 domains, but rather to estimate which sites they are likely to occupy. The residue scoring criteria were as follows:

- The binding site residue matches the most frequently observed peptide residue at that position (+4 points).
- The binding site residue was identified at that peptide position, although it was not the most frequently observed (+2 points).
- The binding site residue was never observed at that peptide position, although one or more residues with a shared biophysical property were (e.g., hydrophobic, same charge type, etc.) (-2 points).
- The binding site residue was never observed at that peptide position, and neither were any similar residues with shared biophysical properties (-4 points).

Sla1-SH3#1-2 were expressed together within the peptide screen due to purification difficulties associated with the pair. SH3 binding positions were colour coordinated with their criteria scores given. Grey represents sites of the incorrect class, salmon represents sites which scored a negative value, and green represents sites with a positive value. The shade of green was based on the score value with higher scores giving a darker shade. Regions of binding identified within the spot array are highlighted in red (**figure 4.10**).

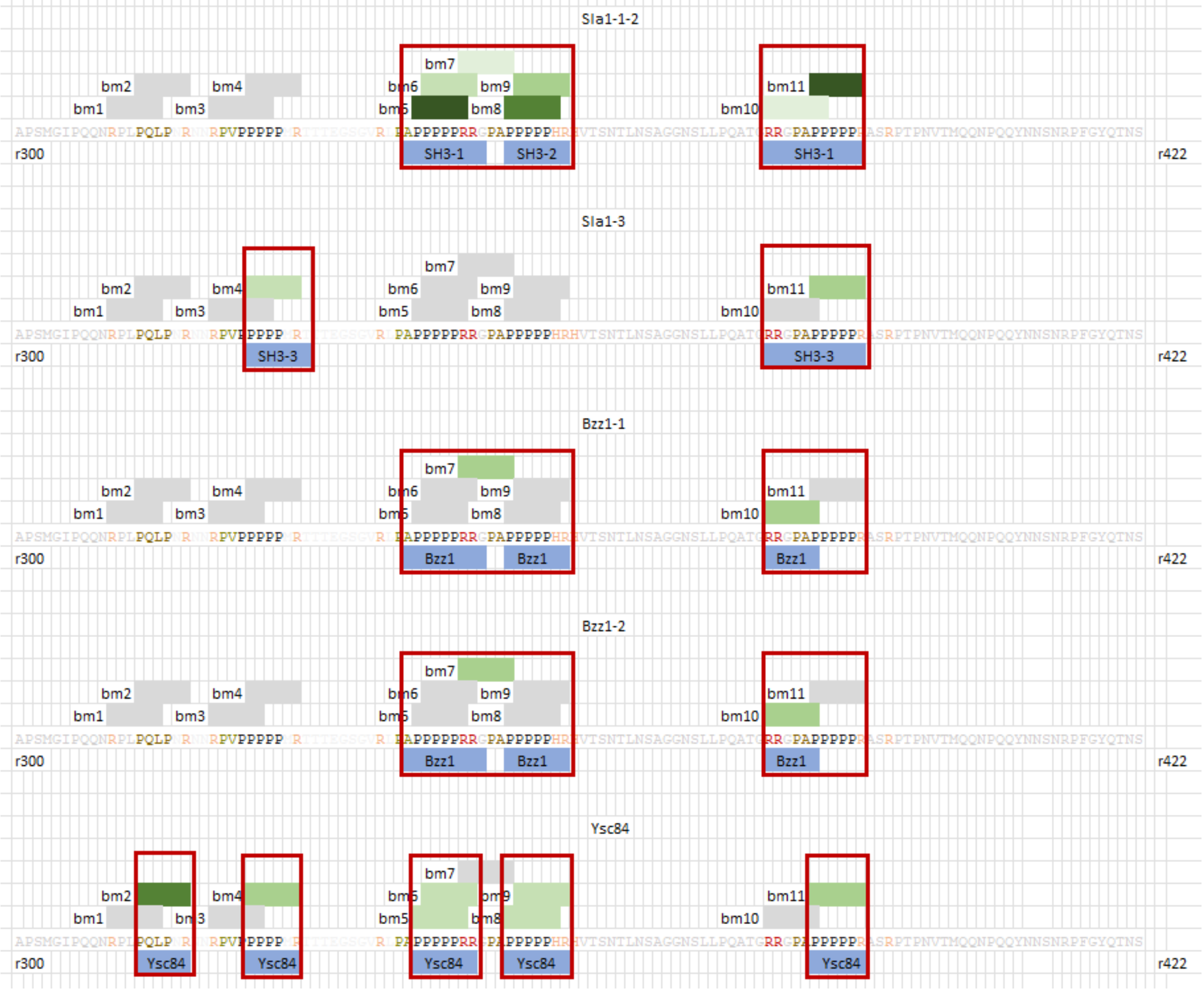


Figure 4.10: Predicting SH3 binding locations for the computational modelling. The specific binding motifs bound by SH3 domains from the endocytic patch was not well characterised. Therefore, the 300-422 residue sequence of Las17 was annotated with black representing polyproline tracts, gold representing core PxxP motifs which are located outside of the tracts, salmon for positive residues within three residues of a proline and red for the two pairs of arginines known to interact with actin. Each PxxP cite was then annotated above to establish an SH3 binding site. These sites were extending to an arginine if they fit either the class I (+xxPxxP) or class II (PxxPx+) definition. Ultimately, 11 SH3 binding motifs were identified (bm1-bm11). Each SH3 domain was mapped to the binding motifs indicated from spot array data (J Palmer, personal communication). This is shown under the appropriate sequence and surrounded with a red box. All motifs not expected to bind the SH3 domain (either due to being located outside of the red boxes, or being of the wrong classification) are shown in grey. Binding motifs suspected of binding the indicated SH3 domains are then coloured green with progressively darker shades indicating a higher criteria score (described before this figure).

4.2.3 Predicting the binding positions of actin

The arginine pairs at Las17 residue positions 349/350 and 382/383 have each been shown to significantly reduce actin nucleation following their mutagenesis to alanine. Proline to alanine mutations in the C-terminal tracts have also been shown to reduce the activity of Las17-mediated nucleation. Using this information, the 300-422 region was analysed to identify the likely actin-binding regions. It was found that three of the Las17 polyproline tracts in this region appear to share a similarity in sequence as each possesses a pair of arginines 3-4 residues N-terminal of their five prolines. These sequences (given from the first arginine to the final proline) include RNNRPVPPPPP, RRGPAPPPPP, and RRGPAPPPPP. **Figure 4.11** shows the positions of these tracts along the 300-422 sequence in red with a red box highlighting the two arginines associated with each tract.

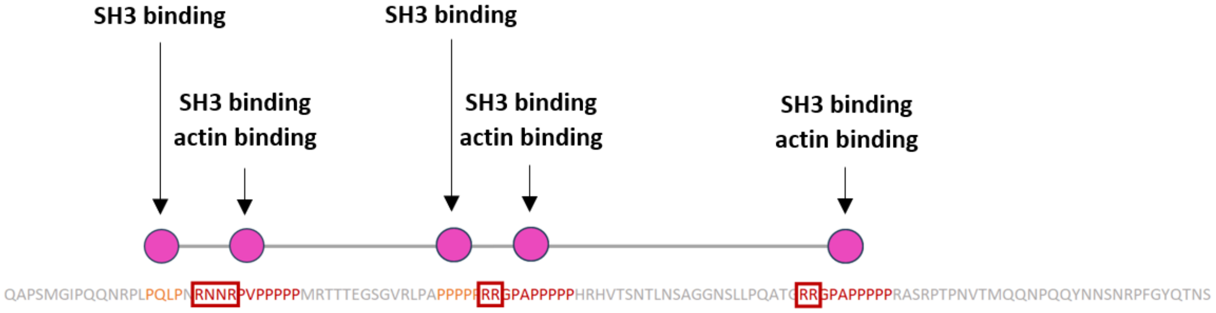

SH3 binding
SH3 binding
SH3 binding
actin binding
SH3 binding
actin binding
SH3 binding
actin binding
QAPSMGIPQQNRPLPQLPNRNNRPVPPPPPMRTTTEGSGVRLPAPPPPRRGPAPPPPPHRHVTSNTLNSAGGNSLLPQATGRRGPAPPPPPRASRPTPNVTMQQNPQQYNNSNRPFGYQTNS

Figure 4.11. Structure of a Las17Agent. The ABM structure of Las17 agents (top with magenta representing binding sites) is shown above the Las17 sequence with red indicating the predicted actin-binding sites, orange showing other PxxP locations, and red boxes framing the arginine believed to bind actin. To simplify the model, overlapping binding motifs identified in **figure 4.10** were coalesced to form five ABM binding motifs. These include bm1 (formally bm1-bm2), bm2 (formally bm3-4), bm3 (formally bm5-bm6), bm4 (formally bm7-bm9), and bm5 (formally bm10-bm11). Binding motifs which overlapped with actin-binding polyproline tracts were able to exclusively bind either an actin or SH3 agent. These included bm2 (overlaps with PP1), bm4 (overlaps with PP3), and bm5 (overlaps with PP4).

The second polyproline tract (shown in orange) possessed only a single N-terminal arginine and was located so close to the third tract that it was inconceivable that both sequences could simultaneously bind actin. Therefore, we believe the second tract does not significantly participate in binding the actin monomers required to seed a filament. This assumption was later tested and confirmed in **chapter 6**. For additional context, the only core SH3 binding PxxP motif identified outside of a polyproline tract (PQLP) was also coloured orange in the figure. Combining this analysis with the SH3 binding site predictions gives a sense of what agent types can bind to each of the five binding regions represented by the computational model. Binding of SH3 domains to either the third or fourth binding site was presumed to prevent actin association with the fourth site through masking of the arginine pair which plays a critical role in actin binding.

4.2.3 Discussion of binding affinities used for Model Alpha

Both agent-based models built within this thesis describe Las17 as a series of five binding motifs. These are henceforth termed bm1, bm2 (PP1), bm3 (PP2), bm4 (PP3), and bm5 (PP4). The binding affinities and locations determined within this chapter were combined to generate a set of model parameters for Model Alpha. These are given in **table 4.3**.

In summary, an affinity of 0.25 µM was chosen for Sla1-SH3#1 based on the MST results of **sections 4.2.1.3** and **4.2.1.6**. An affinity of 0.056 µM for the Sla1-SH3#1-2 construct was taken from the literature (Feliciano and Di Pietro, 2012) and the affinity of the second Sla1 SH3 domain (1600 µM) was calculated using an agent-based model prediction as described in **section 4.2.2.1**. The third SH3 domain of Sla1 (and by extension the 3-domain construct Sla1-SH3#1-3) was not included within this version of the model. The affinity for the highest scoring binding site of Ysc84 (on the site with the greatest saturation on the spot array) was given with a typical SH3 affinity found within the literature (0.25 µM) whilst the other sites were given using half this affinity to represent the apparent binding preference (Demers and Mittermaier, 2009). This was coincidently the same value determined for Sla1-SH3#1. The affinity for a typical weaker binding SH3 domain (1.0 µM) was used for the cloud domain interactions (Demers and Mittermaier, 2009). Actin binding at the bm4 and bm5 sites were given an affinity of 1 µM in lieu of experimental data whilst the assumed weaker bm2 site was given the affinity of 2 µM. These values are generally consistent with Cobl and other tandem actin-binding nucleators (Chereau *et al.*, 2005; Chen *et al.*, 2013).

<table>
<tr><th colspan="5">Peptide</th><th colspan="5">bm1 affinity</th><th colspan="5">bm2 affinity</th><th colspan="5">bm3 affinity</th><th colspan="5">bm4 affinity</th><th colspan="5">bm5 affinity</th></tr>
<tr><td colspan="5">Sla1-SH3#1</td><td colspan="5">-</td><td colspan="5">-</td><td colspan="5">0.25 µM (1)</td><td colspan="5">-</td><td colspan="5">0.25 µM (1)</td></tr>
<tr><td colspan="5">Sla1-SH3#2</td><td colspan="5">-</td><td colspan="5">-</td><td colspan="5">-</td><td colspan="5">1600 µM (2)</td><td colspan="5">-</td></tr>
<tr><td colspan="5">Cloud SH3</td><td colspan="5">-</td><td colspan="5">-</td><td colspan="5">0.5 µM</td><td colspan="5">0.5 µM</td><td colspan="5">-</td></tr>
<tr><td colspan="5">Ysc84</td><td colspan="5">0.25 µM</td><td colspan="5">0.50 µM</td><td colspan="5">0.50 µM</td><td colspan="5">0.50 µM</td><td colspan="5">0.50 µM</td></tr>
<tr><td colspan="5">Actin</td><td colspan="5">-</td><td colspan="5">2 µM</td><td colspan="5">-</td><td colspan="5">1 µM</td><td colspan="5">1 µM</td></tr>
<tr><th colspan="30">Baseline rates/ratios for calculations</th></tr>
<tr><th colspan="15">Base SH3 domain binding rates</th><th colspan="15">Base actin domain binding rates</th></tr>
<tr><td colspan="5">k_{on} (µM^{-1} s^{-1})</td><td colspan="5">k_{off} (s^{-1})</td><td colspan="5">K_{d} (µM)</td><td colspan="5">k_{on} (µM^{-1} s^{-1})</td><td colspan="5">k_{off} (s^{-1})</td><td colspan="5">K_{d} (µM)</td></tr>
<tr><td colspan="5">$0.98 * 10^{9}$</td><td colspan="5">63.7</td><td colspan="5">0.065</td><td colspan="5">$7 * 10^{6}$</td><td colspan="5">0.17</td><td colspan="5">0.024</td></tr>
<tr><th colspan="30">Other parameter values</th></tr>
<tr><th colspan="6">Binding radii</th><th colspan="6">Barbed end k_{on}</th><th colspan="6">Barbed end k_{off}</th><th colspan="6">Pointed end k_{on}</th><th colspan="6">Pointed end k_{off}</th></tr>
<tr><td colspan="6">7.346 nm</td><td colspan="6">11.6 M^{-1} s^{-1}</td><td colspan="6">1.4 s^{-1}</td><td colspan="6">1.3 M^{-1} s^{-1}</td><td colspan="6">0.8 s^{-1}</td></tr>
</table>

Table 4.3. Model Alpha default parameter values. Rates used for running Model Alpha. The justification for all parameters are given in main text for this section. Parameter values with a superscript of 1 were obtained using MST data collected as part of this thesis whilst values with a superscript of 2 were calculated using Model Alpha.

Association (k_{on}) and dissociation rates (k_{off}) were calculated using a reference interaction as described in **chapter 3**. The reference interactions were chosen by taking typically observed literature values and increasing the k_{on} (whilst decreasing the k_{off}) to fit the K_d to the values required (Demers and Mittermaier, 2009; Pang *et al.*, 2012). A low base K_d was used for the scaling such that the probability of binding interactions never exceeded a value of 1. This is because the radius was fixed and a k_{on} of too great a value would result in in a binding probability >1 for the fixed radius.

4.3 Computation Modelling – Model Alpha

The first application of Model Alpha following its construction and internal validation, was to compare simulated outputs to experimental data. Agent-based models are built using a bottom-up, individual-level approach which focuses on characterising emergent macro-level behaviours, often with a mix of qualitative and quantitative analysis. This in contrast to more mathematical techniques such as Ordinary Differential Equation modelling (ODE) (Azimi, Jamali and Mofrad, 2011; Torres and Santos, 2015). To quote George Box, "all models are wrong, some are useful" (Curchoe, 2020). Las17-mediated nucleation remains poorly characterised and so its modelling requires many assumptions and simplifications. Even the mechanics of Las17-actin binding remains almost entirely undiscovered. Nevertheless, our model can reveal emergent behaviour that would be associated with our hypothesis whilst identifying the components most responsible. These observations can lead to hypothesis refinement which in turn may help direct experimental design and thus prove "useful". Therefore, comparing the absolute quantitative values produced from this thesis' modelling with experimental data was of lower priority than comparing the parameter relationships and qualitative observations. However, it is still important to roughly align computational and experimental outputs to help identify any significant deviations. Larger discrepancies may suggest a significant mechanistic inaccuracy within the model which cannot be explained by inaccurate variable values.

Pyrene assays are frequently employed in the study of actin dynamics due to the relative fluorescence emitted by actin-pyrene indicating the degree of polymerisation within the assay. The percentage of F-actin accumulated within the simulation was recorded to allow for a computation output directly comparable to experimental studies. An initial analysis using the default parameters given in **table 2** revealed that Model Alpha nucleated actin with a rate significantly higher than Las17 (**figure 12a**). Interestingly, the observed rate was broadly within the range of Cobl – a known cross-filament nucleator (**figure 12b**). The actin-binding affinities used by Model Alpha were derived from the observed range for Cobl WH2s. This may explain why the two polymerisation curves appear similar whilst providing validation that our computation model was working as expected.

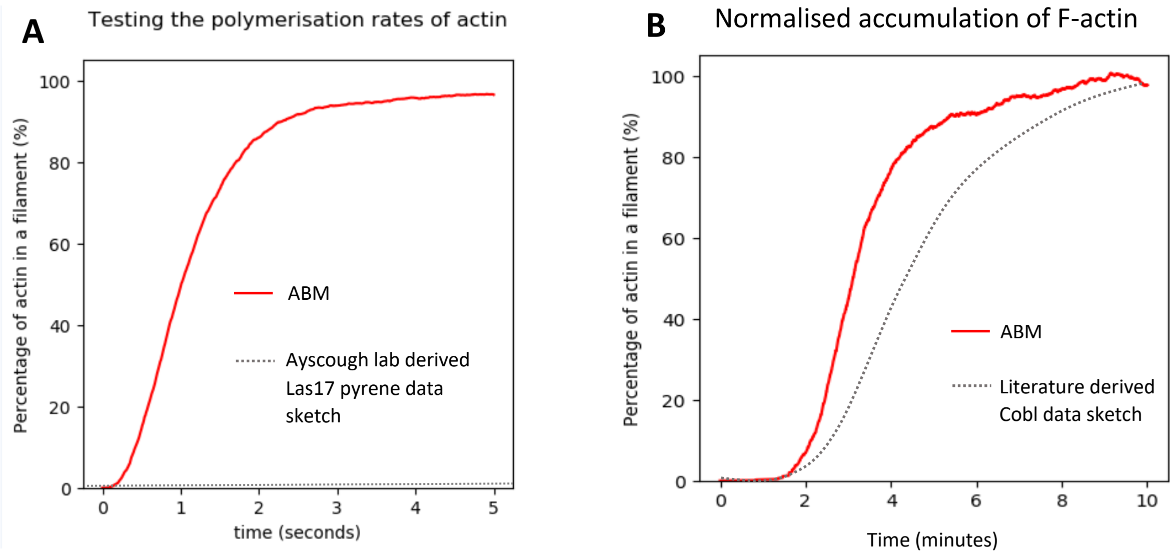


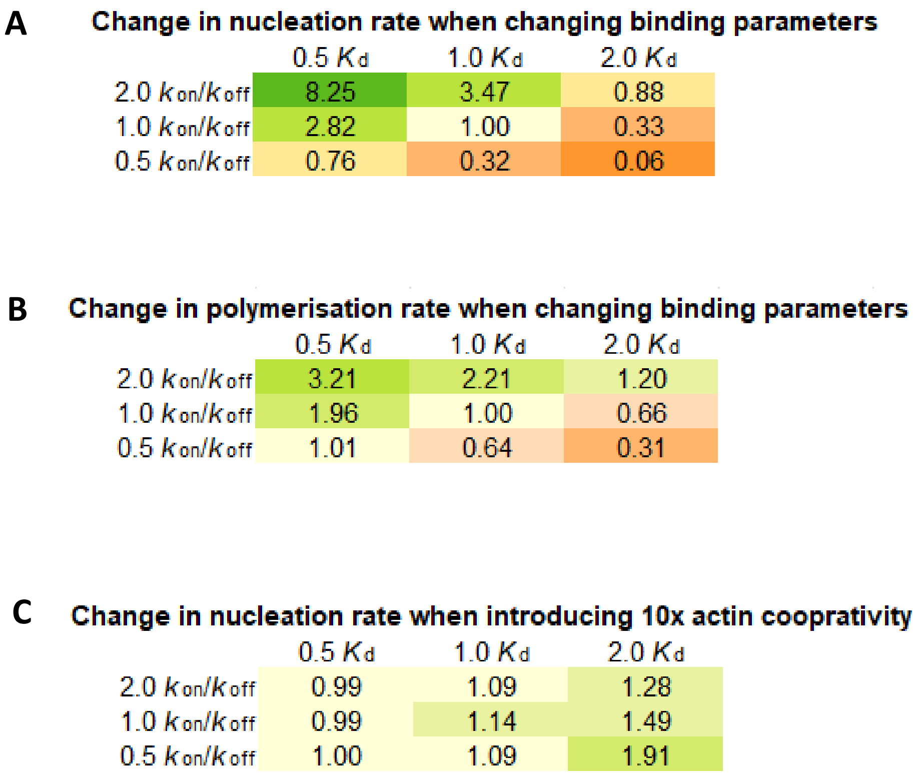
Figure 4.12: Comparing Model Alpha polymerisation curves with experimental data. Las17 and actin were simulated using Model Alpha to produce polymerisation curves for comparison to experimental pyrene data. Both comparisons show the Model Alpha ABM in red and a sketch closely following the experimentally determined values in a dotted grey. **A**) Experimental data for Las17 was taken after a discussion with E. Allwood of the Ayscough lab. The result shows that 300 nM Las17 is clearly many orders of magnitude more active than expected when nucleating 5 µM actin. **B**) Experimental data for Cobl was taken from **figure 4** of Ahuja *et al.*, 2007. The polymerisation curve for 40 nM las17 and 2 µM actin shows clear similarity to the polymerisation curve of the cross-filament nucleator Cobl.

4.4 Model Alpha parameter sensitivity analysis

The following stage of understanding Model Alpha was to assess its sensitivity to a variety of parameters. Developing an agent-based model of a vastly underexplored system comes with the understanding that many (if not most) of the assumed parameters will not be exactly true to the "real life" system. Therefore, it is prudent to understand how variation to our assumed parameter values affect the major phenomena we are observing. We expect two major outcomes from this study. First is to discern whether the major variation between simulation and experimental output can be removed through reasonable changes to the parameters, or if an update to our hypothesised nucleation mechanism appears more reasonable. Second is that a sensitivity analysis allows us to better understand how changes to variables affect our conclusions. Without robust experimental data, inaccuracies within our parameters are a given. Therefore, validating that our observations occur throughout a range of variable values provides greater confidence that our conclusions are not reliant on specific, less certain parameters. Put simply, the analysis aims to conclude how robust are the observed phenomena relative to changes in the variables.

4.4.1 Sensitivity analysis: Nucleation and polymerisation sensitivity

A reasonable place to start the sensitivity analysis was to explore how actin binding parameters affect nucleation and polymerisation rates. The analysed parameters included structural cooperativity of actin between the different Las17-binding sites, the magnitude of association and dissociation rates (whist keeping K_d constant), and the K_d values for polyproline tracts and actin. Each simulation was run with 1728 Las17 (0.3 µM) and 28800 actin (5 µM) agents for 3 seconds to ensure capture of both early nucleation and polymerisation rates. These concentrations are consistent with those used within the Ayscough Las17-mediated nucleation experiments. The initial rates of nucleation and polymerisation were calculated from the number of actin seeds and F-actin formed within the first second of simulation respectively. Both increasing the magnitude of rates and tract affinities were predicted to increase the observed rates. Results are shown in the figure below (**figure 4.13**). To make the data more digestible, the tabular results (each consisting of nine simulations) is given as the change in the rate from a control which is given in the figure legend and coloured based upon the magnitude of this change.



A Change in nucleation rate when changing binding parameters

	0.5 K_d	1.0 K_d	2.0 K_d
2.0 k_{on}/k_{off}	8.25	3.47	0.88
1.0 k_{on}/k_{off}	2.82	1.00	0.33
0.5 k_{on}/k_{off}	0.76	0.32	0.06

B Change in polymerisation rate when changing binding parameters

	0.5 K_d	1.0 K_d	2.0 K_d
2.0 k_{on}/k_{off}	3.21	2.21	1.20
1.0 k_{on}/k_{off}	1.96	1.00	0.66
0.5 k_{on}/k_{off}	1.01	0.64	0.31

C Change in nucleation rate when introducing 10x actin cooprativity

	0.5 K_d	1.0 K_d	2.0 K_d
2.0 k_{on}/k_{off}	0.99	1.09	1.28
1.0 k_{on}/k_{off}	0.99	1.14	1.49
0.5 k_{on}/k_{off}	1.00	1.09	1.91

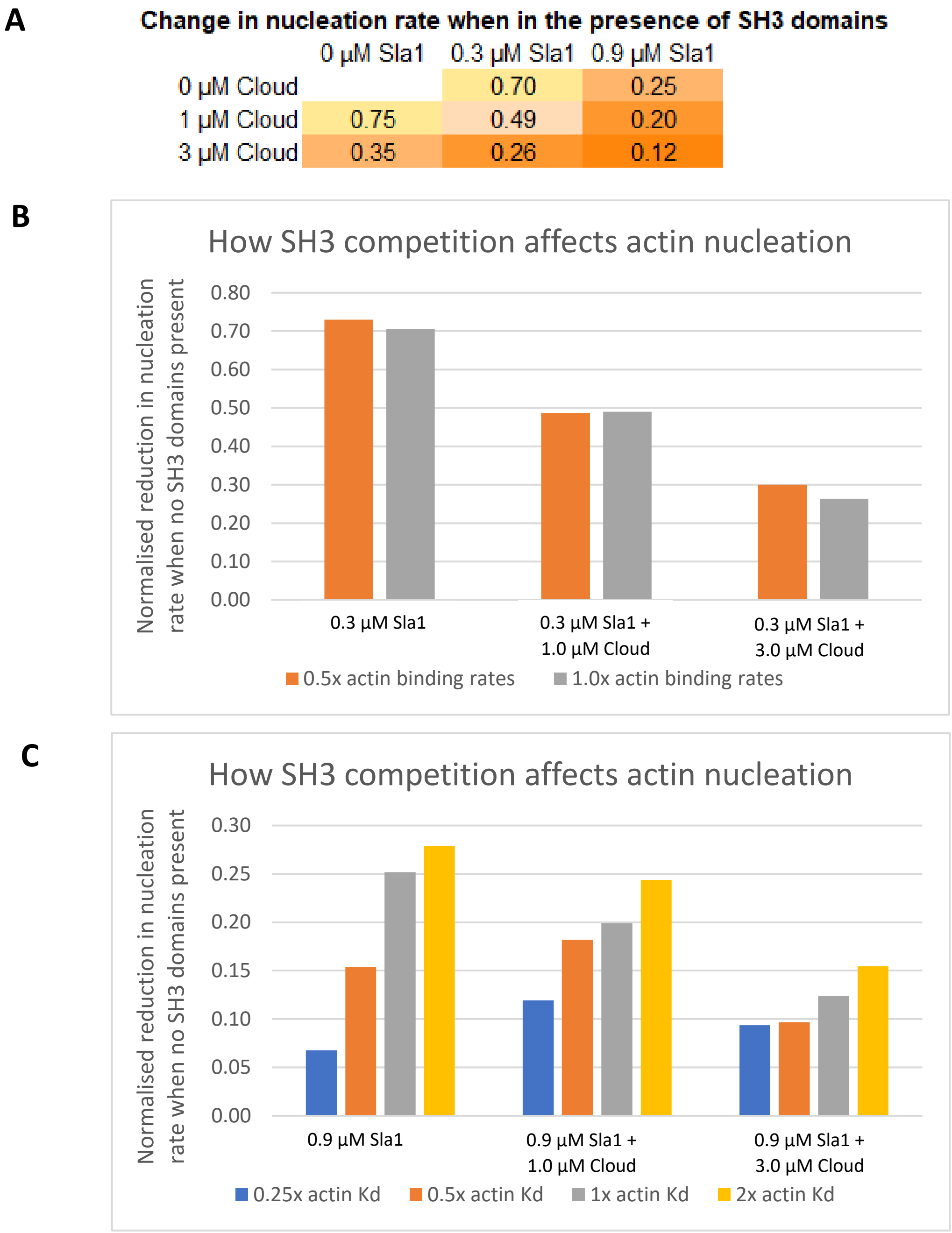
Figure 4.13: Actin rate sensitivity analysis. (**A-B**) The change in indicated rates (nucleation or polymerisation) after applying the indicated k_{on}/k_{off} magnitude modifier and/or K_d modifier. The rate produced when using the default values given in **Table 4.3** (i.e., the 1x modifiers) serves as the control rate. The displayed numbers are calculated as the difference in values (rate X/rate Y) when comparing the indicated rates against the control rate (i.e., a value of 0.70 means an observed rate 70% of the control). Results are coloured progressively more orange the larger the reduction in rate, progressively more green the larger the increase in rate and yellow when the difference is <10%. **C**) The change in rate following the introduction of 10x actin cooperativity (i.e., a 10x affinity increase when an actin agent is bound to an adjacent Las17 binding site). The same colour coordination used for figures **A-B** were applied here.

Increases to both actin-Las17 affinity and the magnitude of association and dissociation rates (given by the vertical columns and horizontal rows of **figure 4.13a** respectively) confers a positive and roughly additive relationship to the rate of nucleation. An analogous relationship can be observed with the rate of polymerisation, albeit with a more dampened effect (**figure 4.13b**).

Structural cooperativity parameters between the actin-binding sites were implemented into Model Alpha as described in **chapter 3**. A brief summary is as follows: Binding of actin at one site was modelled to reduce the actin-binding K_d of adjacent sites 10-fold. This value was arbitrary due to an absence of binding data. Surprisingly, the comparison revealed a much smaller effect on the nucleation rate than expected (**figure 4.13c**). Nevertheless, a positive effect was observed at lower K_d values suggesting structural cooperativity may play a larger functional role if Las17-nucleation relies on weaker binding affinities than modelled.

4.4.2 Sensitivity analysis: Sla1 inhibition

Sla1 functions as a potent inhibitor of Las17 in experimental assays. Internal data from the Ayscough lab suggest that 300 nM Sla1 is sufficient to completely inhibit 300 nM Las17 in an assay with 5 µM actin. This inhibition from Sla1 is a vital component of the Las17 regulatory model while its interplay with generic cloud SH3 domains is also suspected to play a role in Las17 regulation. Therefore, the sensitivity analysis was expanded to include both varying concentrations of the Sla1-SH3#1-2 peptide (0, 0.3, and 0.9 µM) along with varying concentrations of Cloud domains (0, 1, and 3 µM). Prior to their arrival at the endocytic patch, Las17 and Sla1 are known to interact together as part of the SLAC complex (Feliciano and Di Pietro, 2012). In addition to this, all non-actin proteins are mixed prior to the addition of actin in *in vitro* studies conducted by the Ayscough lab. All simulations undertaken with Sla1 were therefore started from an analogously pre-equilibrated state. This also simplified aspects of analysis such as being able to plot the reduction in Sla1 binding over time to better characterise SH3 competition.



Change in nucleation rate when in the presence of SH3 domains

	0 µM Sla1	0.3 µM Sla1	0.9 µM Sla1
0 µM Cloud		0.70	0.25
1 µM Cloud	0.75	0.49	0.20
3 µM Cloud	0.35	0.26	0.12

Figure 4.14: SH3 competition in model Alpha. **A**) The difference in values (result X/result Y) when comparing the indicated nucleation rates against the control rate produced when no SH3 domains are present (i.e., a value of 0.70 means an observed rate 70% of the control). Results are coloured progressively more orange the larger the reduction in rate. **B**) The change in rate when in the presence of SH3 domains relative to the no SH3 control that used the same actin rate as the result. **C**) The change in rate when in the presence of SH3 domains relative to the no SH3 control that used the same actin K_d modifier as the result.

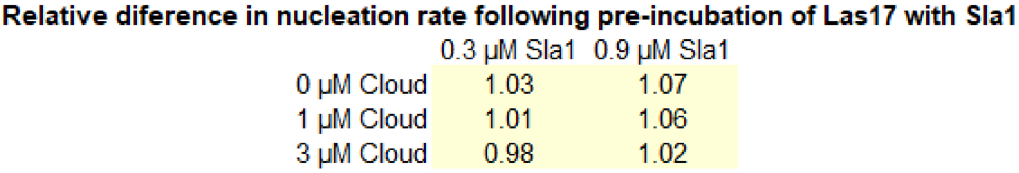
The results, as expected, reveal that both Sla1 and generalised cloud SH3 domains mostly confer an additive reduction to the rate of Las17 nucleation (**figure 4.14a**). Decreasing the magnitude of the association and dissociation rates 2-fold, whilst fixing the K_d, had very little effect on the outcome (**figure 4.14b**). This suggests that these parameters are less important when investigating how SH3 domains regulate Las17 (at least within the screened range) and so inaccuracies in their values are unlikely to change the major conclusions drawn from the model. Meanwhile, changing the K_d values between Las17 and G-actin conferred a greater variability to the degree of inhibition (**figure 4.14c**). A 2-fold reduction in K_d (from the baseline given in **table 4.3**) reduced the activity in the presence of 0.9 µM Sla1 from 0.25x to 0.15x their respective actin and Las17 only controls. The same negative concentration-dependant inhibition was mostly maintained across the K_d screen.

Part of the hypothesis proposed within **chapter 1** is that cloud SH3s may aid in relieving the inhibition of Sla1 through many weak, competing interactions to help overcome Sla1 sequestering of the Las17 polyproline tracts. This idea was confirmed as mathematically possible (**chapter 9 appendix**) when applied to a closed *in vitro* system with no method of removing Sla1 from the pool of Las17 binders. However, the mathematical modelling involved a significant simplification of the regulatory system and thus all computational analysis of Model Alpha was undertaken with this possible phenomenon in mind.

Interestingly, the addition of cloud SH3s can induce this theorised inhibition-relieving effect when simulating using low actin K_d values (**figure 4.14c**). For example, when running the simulation with 4-fold higher actin K_ds, adding 1 µM cloud SH3s to 0.9 µM Sla1 almost doubles the activity when compared to the Sla1-only simulation (0.12x vs 0.07x the actin and Las17-only control). It should be noted that the maximum gain in activity (following the addition of cloud SH3s to a Sla1-inhibited Las17 system) identified within this search fell far short of what was required to return Las17 activity to the level of Las17 alone (i.e., without any SH3s) (**figure 4.17d**).

4.4.3 Sensitivity analysis: The effect of pre-equilibrating Las17 and Sla1

As explained in the previous section, all ABM simulations involving Sla1 were pre-equilibrated prior to the addition of other proteins such as actin. This was performed by simulating Las17 and Sla1 in the absence of all other agents for a period of around 0.1 seconds. The sensitivity analysis from **figure 4.14a** was repeated with no pre-equilibration to help understand how this changes the nucleation rate.



Relative diference in nucleation rate following pre-incubation of Las17 with Sla1

	0.3 µM Sla1	0.9 µM Sla1
0 µM Cloud	1.03	1.07
1 µM Cloud	1.01	1.06
3 µM Cloud	0.98	1.02

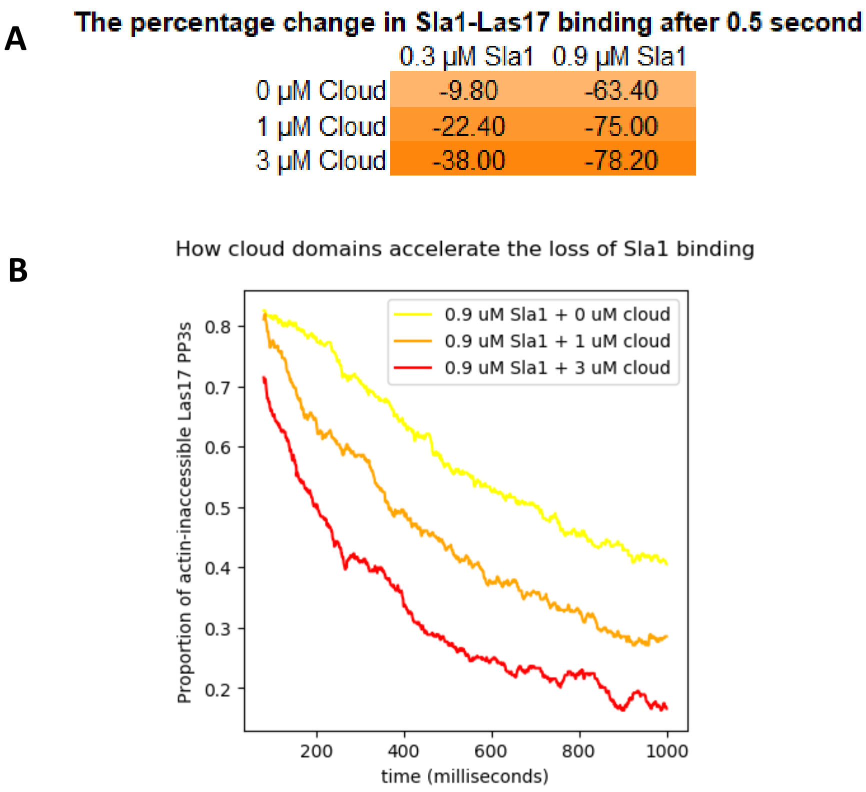
Figure 4.15: Investigating the pre-equilibration of Las17 with Sla1. The difference in values (result X/result Y) when comparing the nucleation rates without Sla1-Las17 equilibration against the nucleation rates with Sla1-Las17 equilibration. Comparisons that gave a <10% difference were coloured in yellow. As can be observed in the figure, no comparison exceeded this 10% threshold indicating no significant difference between the results.

The results of a pre-equilibrated system appear broadly consistent with the non-equilibrated simulations with no change in the resultant nucleation rates exceeding 10% (**figure 4.15**). However, it was identified that the addition of cloud SH3 domains did accelerate a loss of Sla1 from Las17 as described in the following section.

4.4.4 Cloud SH3s accelerate the departure of Sla1

A potential trigger for Las17 activity *in vivo* may be inferred from the different localisation patterns of Las17 and Sla1. Cargo interactions with Sla1 protein-binding domains may essentially “sequester” Sla1 SH3s by localising them at the centre of the endocytic patch. This is in contrast to Las17 which occupies a more peripheral location. Therefore, a useful parameter to analyse – especially when making *in vivo* inferences – would be the rate of Sla1 departure from Las17 via competition with the SH3 cloud.

A python script was built to run through the trajectory files of the SH3 domain sensitivity analysis to extract the occupancy of each Las17 binding site over time. This identified a marked depletion in the percentage of PP3 tracts (the major Sla1 binding region) occupied by Sla1 (**figure 4.16**). The rate of Sla1 loss was positively correlated with the concentration of cloud SH3 domains which demonstrates how their competition may help to accelerate the loss of Sla1 inhibition. For example, Sla1-only simulations experienced a loss of 50.5% after 1 second whilst the addition of 1 µM cloud SH3 domains increases this rate to 57.25%.



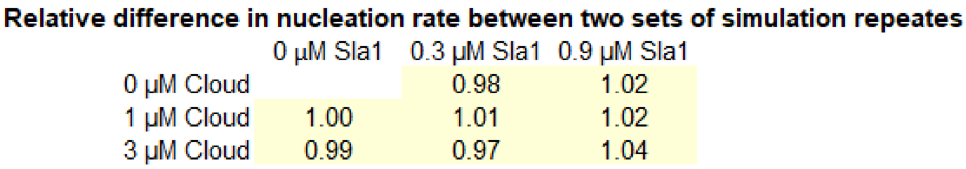
The percentage change in Sla1-Las17 binding after 0.5 second

	0.3 µM Sla1	0.9 µM Sla1
0 µM Cloud	-9.80	-63.40
1 µM Cloud	-22.40	-75.00
3 µM Cloud	-38.00	-78.20

Figure 4.16: The enhancement of Sla1 binding loss in Model Alpha. A) The percentage reduction in Sla1 binding after 0.5 seconds. The values were coloured progressively darker shades of orange the greater the loss of Sla1. Increasing rates of loss in the presence of SH3s suggests that cloud domains can enhance this departure. **B**) The percentage reduction in Sla1 binding over time for 0.9 µM Sla1 in the presence of no cloud domains (yellow), 1 µM cloud domains (orange), and 3 µM cloud domains (red).

4.4.5 Investigating the variability of modelling results between simulation runs

Running an agent-based model to capture several seconds of simulation time is extremely time consuming. Meanwhile, the parallel nature of the computational models built within this thesis resulted in simulation times scaling non-linearly to agent-count. As explained in **chapter 1**, it was decided to use a high agent count, rather than multiple low-agent count simulation replicates, to increase the confidence of our results. We believe this approach was warranted given the time constraints of a PhD. To assess the variability between simulations, the full actin-affinity sensitivity analysis described thus far was repeated using regenerated initiation XML files and number seeds. A comparison between the nucleation outputs is given in **figure 4.17**. All simulations were within 5% of their replicate pairs suggesting good confidence in conclusions derived from the model outputs. All conclusions made from the computation results generated as part of this thesis were drawn from a pattern of behaviour that was conserved across a sensitivity analysis. Furthermore, these conclusions typically relied upon significant deviations – often observing a >2-fold magnitude of change. These factors, combined with the high consistency between simulation replicates, provide greater confidence in the conclusions drawn.



Relative difference in nucleation rate between two sets of simulation repeates

	0 μM Sla1	0.3 μM Sla1	0.9 μM Sla1
0 μM Cloud		0.98	1.02
1 μM Cloud	1.00	1.01	1.02
3 μM Cloud	0.99	0.97	1.04

Figure 4.17: Assessing the reproducibility of Model Alpha results. The difference in values (result X/result Y) when comparing nucleation rates produced by the second sensitivity analysis against those produced by the first analysis. Comparisons that gave a <10% difference were coloured in yellow. As can be observed in the figure, no comparison exceeded this 10% threshold indicating no significant difference between the results of the two sensitivity analyses.

4.5 Chapter discussion

4.5.1 The experimental challenges of Las17

Las17 continues to provide obstacles to experimental analysis due to unpredictable and difficult to address behavioural challenges. This may be an example of the hurdles faced when studying a membrane-interacting protein. However, this is likely compounded by other factors such as a propensity to self-interact and a high predicted pI (12.35 for 300-422 as estimated using the ProtParam tool). Any one of these factors may explain the unpredictable behaviour attributed to this system, such as our inability to lysine label the peptide. Ultimately, these challenges highlight the benefit of computational insight. Using ABMs to better understand how proteins may interact, and compete for, Las17 can help to refine our regulatory hypothesis by testing the veracity of our assumed mechanisms whilst also providing input during experimental design.

4.5.2 Visualising binding competition of the Sla1 tandem

Our homology modelling suggests that the close spatial positioning of the SH3#1-2 tandem domains allows the region to function as a single binding unit. This allows the individual weak SH3 domains to combine greatly in effect through avidity. The Sla1-SH3#1-2 multi-domain unit can therefore be viewed as a ‘hinged seesaw’ rather than two independent domains (**figure 4.18**). In this model, there is often one domain bound to Las17 even whilst the other SH3 is unbound.

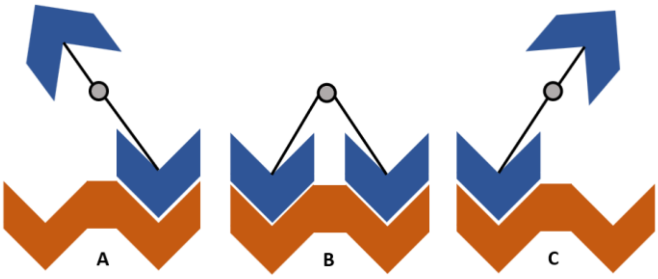


Figure 4.18: The hinged seesaw binding model of Sla1-SH3#1-2. The Sla1-SH3#1-2 tandem construct may be visualised as a single unit with two SH3 domains (shown in blue) connected via a hinge. These two domains (SH3#1 and SH3#2) can independently bind Las17 (orange). The binding of one SH3 results in a significant increase to the relative Las17 concentration for the other domain contributing to a high avidity interaction.

Model Alpha predicts how single SH3 domains are able to effectively outcompete Sla1. The hinged seesaw model helps to visualise how these weakly competing interactions reduce the tight overall binding between Sla1 and Las17. If a cloud SH3 binds to the unbound tract in either **state A** or **state C**, then transition to **state B** becomes impossible resulting in complex integrity being maintained through a single SH3-tract affinity. Considering the low affinity of these individual tract interactions (relative to the tandem), such binding competition would reduce the effect of avidity and thus drastically reduce the effective K_d. The weak binding interactions between individual Sla1 SH3s would rely upon faster k_{off} rates than the SH3#1-2 tandem and thus the presence of the SH3 cloud can accelerate the transition of Las17 from a Sla1-bound to a Sla1-unbound state. This is a good demonstration of the concept of "fuzzy binding" as described in **chapter 1** (Williamson, 2023).

4.5.3 Result confidence in light of time limitations

Comparing two replicates of a sensitivity analysis revealed a high degree of confidence between simulation runs as described in **section 4.4.5**. All the nucleation rates derived from the simulations fell within 5% of each other. Whilst this illustrates the power of extensive sampling through the use of high agent counts, it was still important to decide a set of fixed rules that must be followed when analysing data to reduce bias. To begin with, an exclusion criterion was imposed. This stated that all results must have a magnitude of difference greater than 10% of the compared simulation for it to be deemed a significant change to reduce the risk of false positives/negatives. Secondly, conclusions were only drawn if the observed phenomenon was consistent across two or more simulations whilst the effect confers a clear pattern across the local sensitivity analysis. All computational analysis performed within this thesis was undertaken using these two rules and results were heavily scrutinised.

4.5.4 Model Alpha results and the regulatory cloud hypothesis

The hypothesis proposed at the start of this thesis posits that cloud SH3s may aid in relieving the inhibition of Sla1 through many weak, competing interactions following arrival of the SLAC complex to the endocytic patch. This competition may help facilitate the rate of Sla1 unbinding from Las17 which leaves the polyproline tracts open to actin-binding, and subsequent nucleation. Cloud SH3s may help to “temper” the activity of Las17 through their own competition with actin. As the composition of this cloud changes – particularly with the recruitment of Ysc84 and Bzz1 - the effect of this competition increases until inhibition is reimposed.

The results of Model Alpha provide support for this hypothesis by showing that cloud SH3 domains accelerate the loss of Sla1 from Las17 whilst themselves providing a degree of inhibition. This effect of competing SH3s is even predicted to increase the nucleation rate (up to an observed 1.7-fold increase) of Sla1-inhibited Las17 *in vitro* even though Sla1 remains in the system and is free to rebind. The Ayscough lab has observed that Ysc84 is able to enhance the activity of Las17 and can even relieve inhibition of Sla1. The conclusions of Model Alpha presented within this chapter provides a potential mechanism for how competing SH3 domains may confer this effect.

4.5.5 Behavioural inconsistencies between modelling and experimental results

Although Model Alpha generally behaves as expected, a major deviation between modelled and experimental results was identified in the rate of F-actin polymerisation (**section 4.3**). The polymerisation rate was significantly higher in Model Alpha than for pyrene assays generated by the Ayscough lab. This may be explained by three possibilities: (1) incorrect parameter variables, (2) a difference in mechanism of actin binding that was unaccounted for in Model Alpha, or (3) an unaccounted mechanism converting the three bound Las17-bound actin monomers into a functional seed.

Another discrepancy between the results was that Sla1 was a less effective inhibitor than expected. Experimental assays from the Ayscough lab show that 300 nM Sla1 can maintain near-complete inhibition of Las17 nucleation. When simulating using analogous conditions in Model Alpha, the degree of Las17 activity was only observed to drop by ~20-80%. This range was dependent upon the K_d value used to model the actin-Las17 interactions which were assigned with low confidence. Nevertheless, this suggests that actin affinities are of key interest. Published data regarding the affinity between the Las17(330-422) region as a whole and actin (in G-buffer) support an affinity towards the upper end of the sensitivity analysis. Consequently, experiments were planned to better gauge the expected actin-Las17 K_d value.

4.5.6 Addressing the possibility of a more complex actin-binding mechanism

Another possible explanation for the extreme discrepancy between simulation and experimental pyrene curves may be provided by an unaccounted mechanism that governs the recruitment of actin monomers to Las17. Many proteins must multimerise to exhibit significant function. Therefore, one possible accounting for the difference in activity may be that Las17 must first self-interact (possibly to form a dimer). This would significantly reduce the simulated activity of Las17 whilst possibly leaving the nucleator open to greater inhibition via Sla1 should actin-binding prove positively cooperative.

4.5.6.1 The possible multimeric action of Las17

The identification of dimerisation as a possible unaccounted mechanism was not without an experimental grounding. Upon concluding the analysis of Model Alpha, past work conducted by the Ayscough lab was re-examined to search for signs of Las17 self-interaction. Several key observations were looked at in parallel to evidence that Las17 functions using self-multimerization (assumed hereafter to be dimerisation).

Internal lab data from the Ayscough lab using MST had previously investigated the stoichiometry of Las17 and actin. This analysis suggested that each actin monomer interacted with roughly four Las17 polyproline regions simultaneously which was originally considered unexpected. While it must be noted that the challenging nature of Las17 makes deriving an accurate concentration difficult (see **section 4.1.2**), this result highlights the possibility that actin-binding is not a simple 1:1 interaction. Supporting this, Las17 leaves dextran-agarose columns during SEC at the rough size of a dimer as previously discussed in **section 4.1.3**. Interactions between Las17 peptides may also explain the difficulties when attempting to label the peptide for MST as self-interactions may help to mask label access to the appropriate residue. Multimerization may also play a role in the extensively observed aggregation observed during purification.

In vivo evidence for this interaction can be taken from how Las17 and Sla1 interact in the cytosol. Las17 and Sla1 both arrive from the cytosol to the endocytic patch together as part of the SLAC complex (Feliciano and Di Pietro, 2012). This complex contains multiple copies of both Las17 and Sla1 proteins and thus Las17 arrives at the endocytic patch in extremely close spatial proximity with other Las17 molecules. Theoretically, this may allow for more effective multimerization.

Dimerisation of Las17 was not possible in Model Alpa due to the fixed nature of Las17 agents. To help explore the effects of this mechanism on Las17 regulation and to allow for the implementation of further simulation features, it was decided that a new computational model must be built using the lessons of Model Alpha to help expand our understanding of Las17 nucleation.

4.5.6.2 Addressing the possibility of a more complex nucleation mechanism

An unaccounted mechanism converting the three bound actin monomers into a functional seed may be evidenced by the nucleating mechanisms of other tandem G-actin binding nucleators. The WH2 domains of the linear tandem nucleators Spire and JMY orientate their three bound monomers in a linear arrangement. This is associated with a lower nucleation rate than cross-filament nucleators such as Cobl (Quinlan *et al.*, 2005) (**figure 1.6**). Actin filaments are constituted from two parallel sub filaments that interact laterally to form a twisted helix. Although the mechanistic process for how linearly arranged monomers are converted into actively polymerising seeds is ill understood, it has been proposed that Spire may recruit 2-3 additional monomers to interact laterally with Spire-bound monomers to form the base of the second protofilament. A similar mechanism was designed and integrated into Model Beta – the second model of this thesis described in **chapters 5 and 6**.

4.5.7 The need for a new agent-based model

As alluded to in the previous section, a new agent-based model (termed Model beta) is required to fully explore the mechanistic differences hypothesised as a consequence of this chapter's results. Model Alpha lacked both a generalised framework and las17 movement making the implementation of mechanism changes challenging. Analysing how these mechanisms affect SH3 competition within Model Beta may provide further insight into the regulatory model. In addition to this, more confirmatory affinity data was required to verify the binding mechanism of Sla1. Biolayer interferometry was therefore investigated as a supporting method to MST and will be discussed later in **chapter 6**.

5. Code Development: Model Beta

The major conclusion of **chapter 4** was that Las17-mediated actin nucleation is likely more complex than our first approximation. Two mechanisms we wished to explore were the possibility that Las17 may need to self-oligomerise to bind/nucleate actin, and that Las17 may function as a linear tandem nucleator similarly to Spire. Therefore, the program had to be modified to allow these new interactions. Model Alpha used a fixed lattice of Las17 agents to significantly simplify the model. This simplification also prevented oligomerisation. Therefore, an entirely new model, Model Beta, was built with Las17 movement in mind.

Permitting Las17 agents to move was an enormous undertaking due to how ingrained its static nature was. Major considerations included allowing five independent binding motifs to move whilst maintaining spatial coherency, executing more parallelism checks (e.g., to prevent multiple Las17 motifs attempting to bind the same agent simultaneously), and maintaining the movement and binding functionality of dynamically formed complexes (e.g., two Las17 connected via a Sla1 agent). The ultimate result of this code revision was an almost complete re-write. Other modifications were made to reflect the computational knowledge gained through the PhD. These additional modifications made Model Beta more modular in nature and capable of simulating many biological systems rather than being model specific. This chapter will explain the second program of the thesis using a similar descriptive method to **chapter 3**.

5.1 Model Beta conceptual changes

Although the code and model specifics differ in a major way between the versions, Model Beta represents the same biological system as Model Alpha and thus most features are preserved. This includes Las17 being represented by a series of five binding motifs (bm1-bm5), actin and SH3s competing to reversibly bind these domains, and three actin monomers nucleating to form an actin nucleus when simultaneously bound to the same Las17 agent. One difference here is that Model Beta sometimes required the recruitment of additional monomers to transition actin nuclei into a seed in accordance with the linear tandem nucleating mechanism presented in **chapter 1**. FLAME GPU 1 was still used along with the same file types (an XML model file and C function file).

Almost the entire code had to be overhauled as Model Beta allows Las17 to move freely and dimerise. The parallel nature of FLAME GPU required the program to carry out a set of checks to prevent multiple agents attempting an exclusive action (e.g., two agents occupying the same binding site). Model Alpha removed the need for most of these checks by fixing the position of Las17 as an extended conformation which prevented many such occurrences – principally during the movement phase. To ensure that this is done properly, the program defines a priority to each agent in a complex, and works through the priorities, disallowing any movements which violate structural parameters. This process is outlined in **section 5.3**. Additional checks were also implemented to maintain the priority system as explained later in **section 5.7**.

Another objective of the code revision was to generalise the agent types and functions such that new agents could be added, and others modified, with ease. All proteins are built using the same agent classes rather than utilising unique agents to describe proteins – as is the case within Model Alpha. Generalising agents removes the requirement for multi-domain proteins to possess their own data classes. This permits Sla1 agents to contain all three of their SH3 domains rather than focusing solely on the first two (i.e., using a TandemAgent in Model Alpha) and allows for the easy creation of new domains (e.g., the actin-binding domain of Ysc84). Observing how these new domains interplay with the mechanistic changes investigated using model beta may further develop our proposed regulatory hypothesis.

5.2 Model Beta design specifics

In this model, proteins can be envisioned as "beads on a string". All peptide constructs are described by two agent types: platforms and domains. PlatformAgents hold the position of up to five vectors (the number of vectors updated is dependent upon the peptide it represents). This data class can therefore be envisioned as a string in the "beads on a string" analogy as they hold the object's spatial position and execute its movement. Platform agents are critical in maintaining the spatial coherency of multi-domain peptides such that the domains do not move apart by a distance greater than their peptide linker. The term "complex" (e.g., within the phrase "binding complex") used throughout this chapter refers to two or more platforms which are bound together. Complex IDs are generated dynamically each time a complex is either formed or changes in composition. No limit is set on how many platforms can exist within a single complex as, theoretically, agents may 'daisy chain' together to form larger complexes and this potential spatial organisation should not be disallowed.

DomainAgents hold the interaction-specific variables and govern the binding and unbinding function phases. In the "beads on a string" analogy, domains represent the beads. Each domain is uniquely associated with one of the five vector positions of a PlatformAgent. These domains are referred to as the "child domains" belonging to the "parent" PlatformAgent. This data class updates its position once per timestep by looking up its parent platform from a list of messages. Every functional peptide motif is represented as a domain agent including actin monomers, individual SH3 domains and each of the five binding motifs of Las17.

The third and final agent type of Model Beta are the FilamentAgents. This data class functions similar to the ScaffoldAgent data class of Model Alpha by retaining a pointed and barbed end number, filament size, and end positional vectors to represent an actin filament. Actin domains can bind, and be incorporated into, FilamentAgents using code derived from Model Alpha.

5.2.1 Changes to actin nucleation

One interpretation of the higher than expected nucleation rate observed in Model Alpha is that the Las17-bound actin monomers may initially arrange themselves linearly in a similar manner to tandem WH2 nucleators such as Spire (Sitar *et al.*, 2011). The mechanistic specifics of this nucleating mechanism are poorly characterised and may involve a complex re-ordering of subunits. Therefore, integrating a similar method into this model required a simplified approach that would allow this phenomenon to be explored whist not contributing overcomplexity to the system – especially when mechanistic parallels are unknown.

In Model Beta, actin nuclei (formed by Las17 after it has bound 3 actins) can only transition into a seed once a fourth and fifth actin is bound. Here, it is assumed that a fourth actin monomer would associate with two of the nuclei subunits via lateral contacts to form a double stranded base to the growing filament. Rates for this reversible interaction were taken from the literature (Sept and McCammon, 2001; table 1 reaction F: k_{on} = 2.18 μM^{-1} s^{-1}, k_{off} = 1.30×10^{3} s^{-1}) and are consistent with the dominant hypothesis for Spire and Cobl (Sitar *et al.*, 2011). Model Beta assigns these values to the *SIDE_PON* and *SIDE_POFF* global variables. These values can be replaced in the initiation file by standard filament polymerisation rates to change Las17 to a cross-filament tandem nucleator – thereby allowing easy changing between the two nucleation mechanisms (**figure 1.6**). Filament agents are not classed as a seed until a fifth monomer binds to “lock in” the fourth subunit. Following this interaction, the oligomer is assumed to occupy the typical structure of a dual stranded, pentameric filament and thus transition to an actin seed is complete. Further gains and losses of actin monomers are processed as typical filament interactions (terminology of seed and nucleus described in **chapter 1, section 1.3.1**).

5.2.2 The inclusion of agent homodimerisation

All platforms in Agent Beta have the possibility to form a homodimer. Dimerisation is mediated by a sixth domain agent – termed "dimerisation domain" – which is unique to the other five possible agents which constitute protein functional motifs. The position of this agent is calculated separately to the other domains and is not retained by the parent platform as it simply occupies the platforms' central vector. Dimerisation domains can bind other dimerisation domains of the same protein type using the global array variables: *DIMER_BIND_RADII*, *DIMER_BIND_PROB*, and *DIMER_UNBIND_PROB*. The binding and unbinding variables can be changed if certain conditions are met (e.g., one of the parent platforms is actin-bound) to *COOP_DIMER_BIND_PROB* and *COOP_DIMER_UNBIND_PROB*. Dimerisation allows all child domains to access alternative binding/binding probabilities (all global variables are detailed later in **table 5.1**).

5.3 Changes to the model Beta movement phase – dynamic hierarchy

An overarching design principle had to be implemented to allow all agent types to move, interact, and change states in parallel whilst remaining in communication with one another. This linking principle is hierarchy and it had to be embedded in every part of the model's functionality. A rigidly enforced hierarchy was important to allow agents to freely form multi-peptide complexes of any size. For example, if a binding complex contains three agents, then these agents must move sequentially, rather than in parallel, to maintain spatial coherency of the structure (**figure 5.1**). Priority values are used to determine which code iteration the agent is permitted to move within (termed "priority blocks"). This is important when the program is run in a parallel environment. Binding information between two domains is stored in each of domain agents as their position within the priority hierarchy may change as other binding/unbinding events occur within their complex.

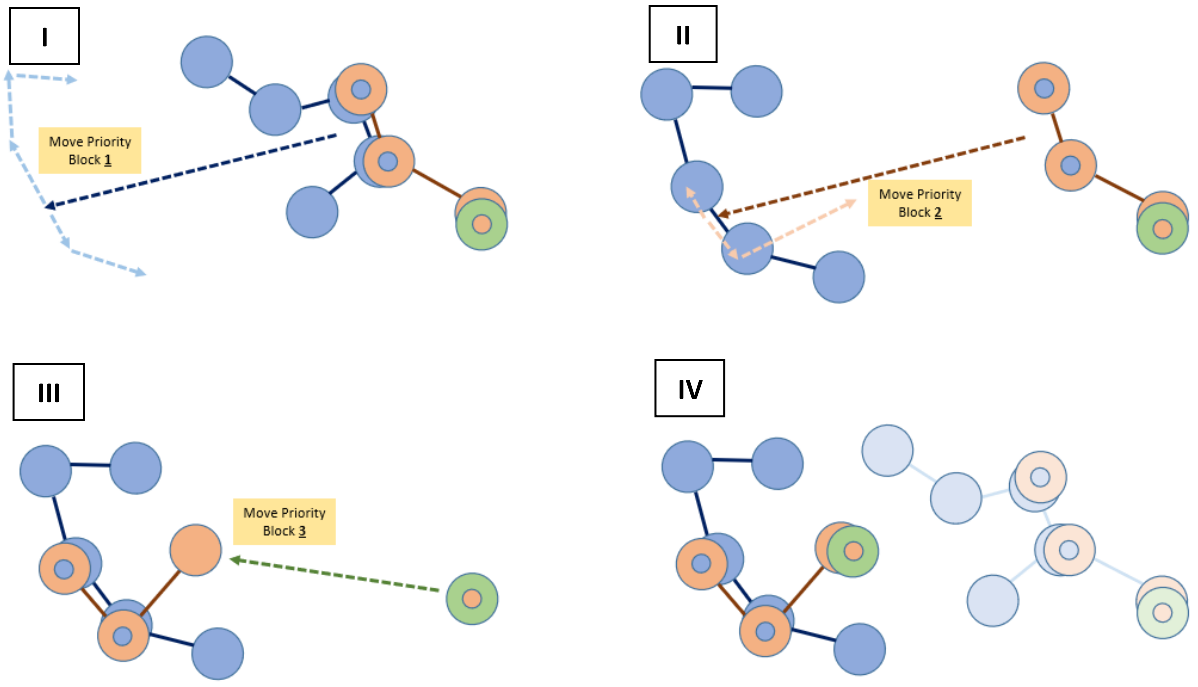


Figure 5.1. Simplified model flowchart. Agent complexes move iteratively via a priority system. This prevents bound agent domains (shown with a small inner circle matching the colour of their bound agent) moving in a manner independent of the rest of the complex. (I) Agents of priority block 1 first move via their central position (dark dashed arrow) and then re-calculate the position of any peptide domains using linker lengths with random orientations (lighter dashed arrows). (II-III) This then repeats for each ascending priority block with the initial movement (dark dashed arrow) being set by any domains bound to higher priority agents. (IV). This continues until all agents within the complex are moved to a new location (old location shown by faded image).

These movement hierarchies must also be dynamic to reflect the transient nature of such complexes. This is because both binding and unbinding interactions can change which agents are associated with a complex, and where in the hierarchy that agent is located (**figure 5.2**).

Of consideration is that some agents (e.g., Las17 and Sla1) contain multiple domains (referred to as a tandem arrangement) and each can form their own interactions. Also, breaking of one domain-domain interaction within a tandem may not cause the agent to dissociate from a complex as another domain from the same agent may still be bound. Consequently, priority values are re-determined following all binding and unbinding events with the agent that initiated the event being set as priority 1. Priority values increase with distance from the priority 1 agent such that a continuous chain of values is maintained (illustrated by the priority values changes shown in **figure 5.2**).

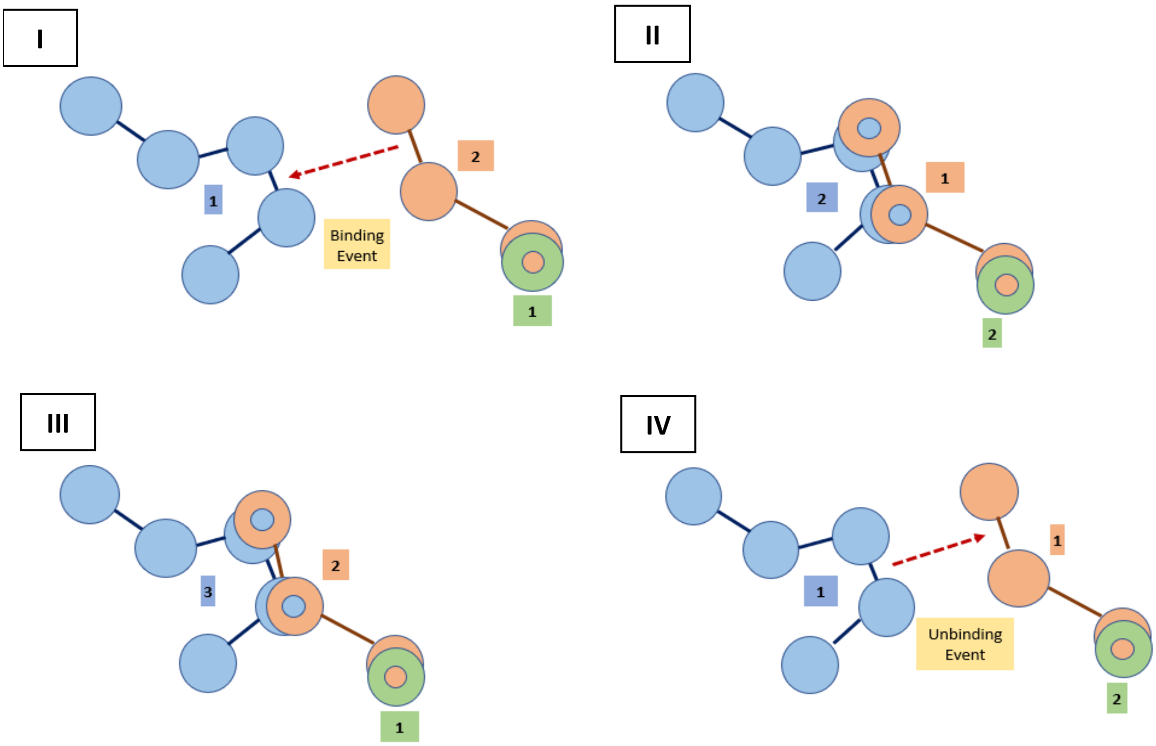


Figure 5.2. Simplified model flowchart. Agents binding and unbinding from one another change the makeup of complexes, which requires a recalculation of priority values. Three agents are shown within these examples: agent X in blue, agent Y in orange, and agent Z in green. Priority values are given in boxes the colour of their respective agents. Binding is shown using a small inner circle matching the colour of the bound partner. (I-II) An example of a binding interaction between agents X and Y (I) necessitating a change in the priority values (II). The method for determining these values was designed to reduce the number of priority blocks (hence the central agent became priority 1). (III-IV) An example of an unbinding interaction between agents X and Y (III) necessitating a change in the priority values (IV).

5.3.1 A distinction between simulation iteration and timestep

In FLAME GPU 1, agents can possess multiple “agent states” that act as guard loops for functions. Each function must define a currentState (the state agents must occupy to use the function) and a nextState (the state the agent moves into following the function). All three agent types featured in Model Beta can exist in one of two states – *resolved* and *unresolved*. Unresolved states can only access the movement phase while resolved states execute the remaining phases which uncouples the iteration and timestep. Functions were written to ensure that all agents switch states simultaneously with a change to the *resolved* state occurring once all agents have successfully moved and a return to the *unresolved* state occurring after the timestep has concluded.

As explained in **section 5.3**, Model Beta enforces the principle of hierarchy when moving agents. To overcome the drawback of parallel execution within functions, and to allow sequential agent movement, priority blocks were used. At the start of the timestep, the first priority block is moved within the *top priority movement phase* followed by the second priority block within the *lower priority movement phase*. If there are any agents with a priority value greater than two, then the current iteration ends and a new iteration begins. The term iteration used throughout this chapter follows that of FLAME GPU with each run through the function list by the program defining a function (even if many of those functions are not executed).

The next sequential priority block is then moved using the *lower priority movement phase* code. This continues until all agents have successfully moved position. Upon the achievement of this criterion, agents shift from an *unresolved* state to a *resolved* state wherein the rest of the code can be executed. This includes the binding and unbinding functionality. Upon completion of this code, the agents return to an *unresolved* state ready to begin a new timestep. The term timestep used throughout this chapter defines each period when agents occupy the *resolved* state as this is when agent interactions occur, and data is recorded. The following figure (**figure 5.3**) visually demonstrates this by showing the major code phases and path of programmatic execution.

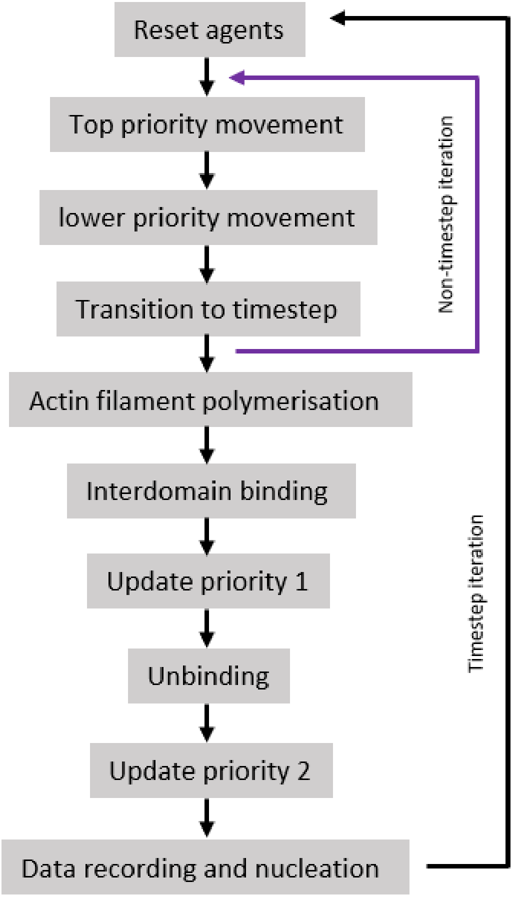


Figure 5.3. Simplified model flowchart. The functions used in the Model Beta simulation can be grouped into ten phases: reset agents, top priority movement, lower priority movement, transition to timestep, polymerisation, interdomain binding, first priority update, unbinding, second priority update, and actin nucleation. These phases are shown in grey boxes whilst the flow of the code is given by the black arrows. An iteration is defined as each time the code repeats a function. The purple arrow shows the only code that can be accessed within a non-timestep loop (*unresolved* state). All functions are accessed during a timestep iteration as agents transition from an *unresolved* state to a *resolved* state before transitioning back to an *unresolved* state at the beginning of the next iteration.

On average, the number of iterations exceeds the number of timesteps achieved by between two and three times due to this uncoupling of timestep and simulation iteration. When requesting n timesteps, the program must therefore be executed for >n*2 iterations to account for this. The exact number of iterations required depends upon the size that complexes reach during the simulation and so cannot be predicted exactly. Iterations are only repeated as a consequence of how priority blocks are moved and not other conflict. For example, binding conflicts are resolved by prioritising the interaction with the closest distance.

5.4 Detailed flowchart

Below is a detailed flowchart of the simulation including functions and message pathways (**figure 5.4**).

Figure 5.4. Detailed model flowchart

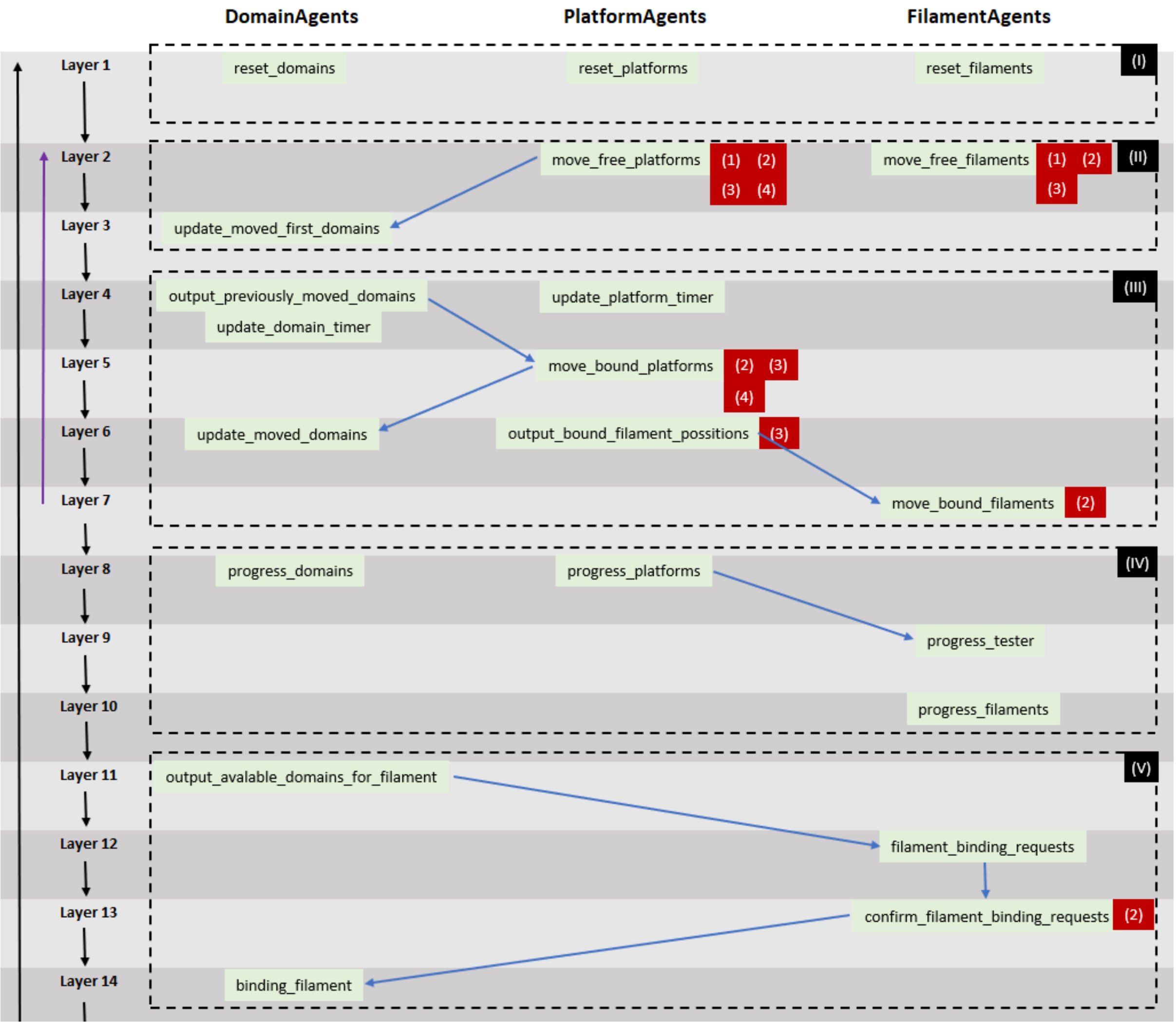
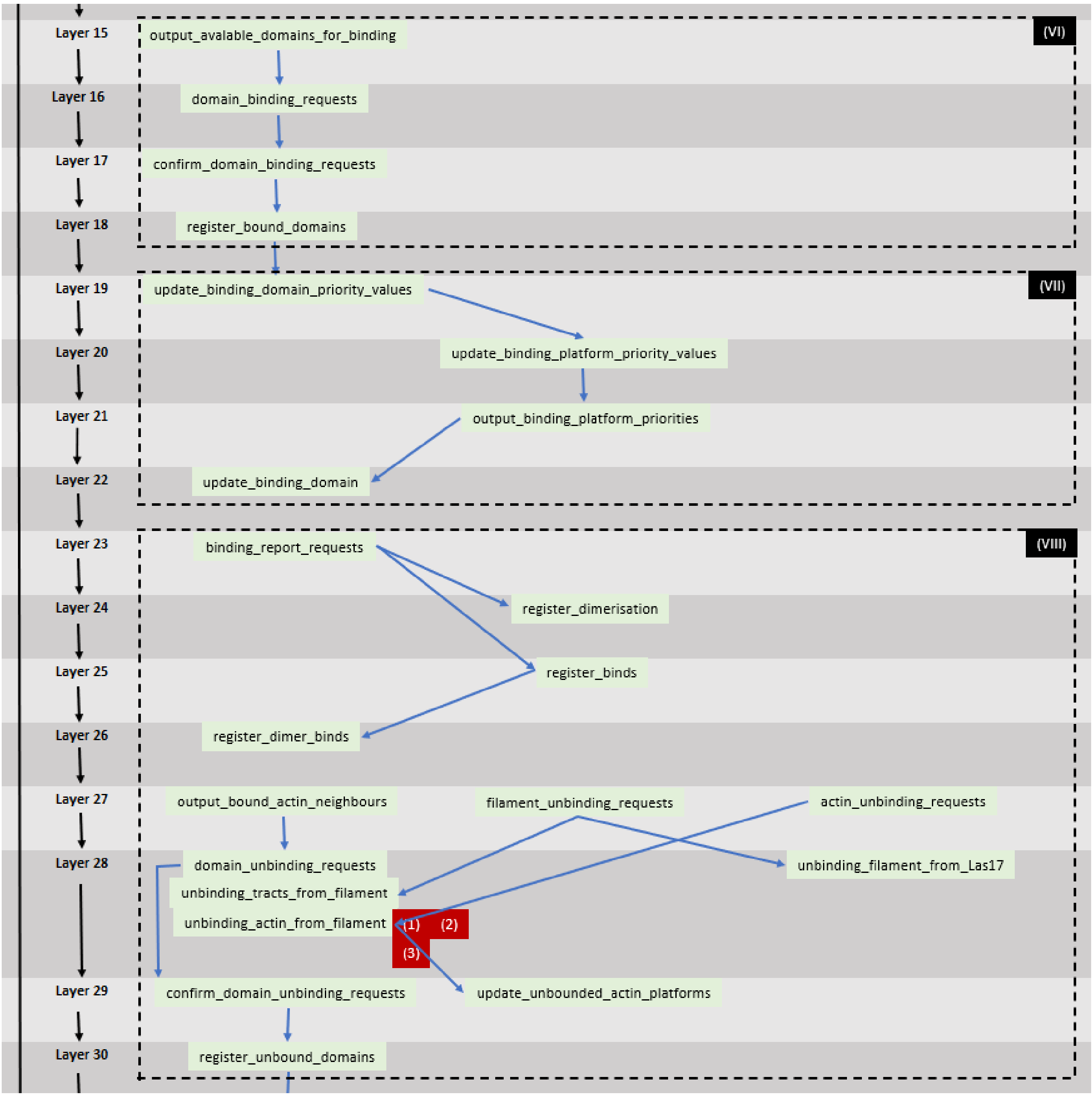


Figure 5.4. Detailed model flowchart (continued)


Layer 15
Layer 16
Layer 17
Layer 18
Layer 19
Layer 20
Layer 21
Layer 22
Layer 23
Layer 24
Layer 25
Layer 26
Layer 27
Layer 28
Layer 29
Layer 30
(VI)
output_avalable_domains_for_binding
domain_binding_requests
confirm_domain_binding_requests
register_bound_domains
(VII)
update_binding_domain_priority_values
update_binding_platform_priority_values
output_binding_platform_priorities
update_binding_domain
(VIII)
binding_report_requests
register_dimerisation
register_binds
register_dimer_binds
output_bound_actin_neighbours
filament_unbinding_requests
actin_unbinding_requests
domain_unbinding_requests
unbinding_filament_from_Las17
unbinding_tracts_from_filament
unbinding_actin_from_filament
(1)
(2)
(3)
confirm_domain_unbinding_requests
update_unbounded_actin_platforms
register_unbound_domains

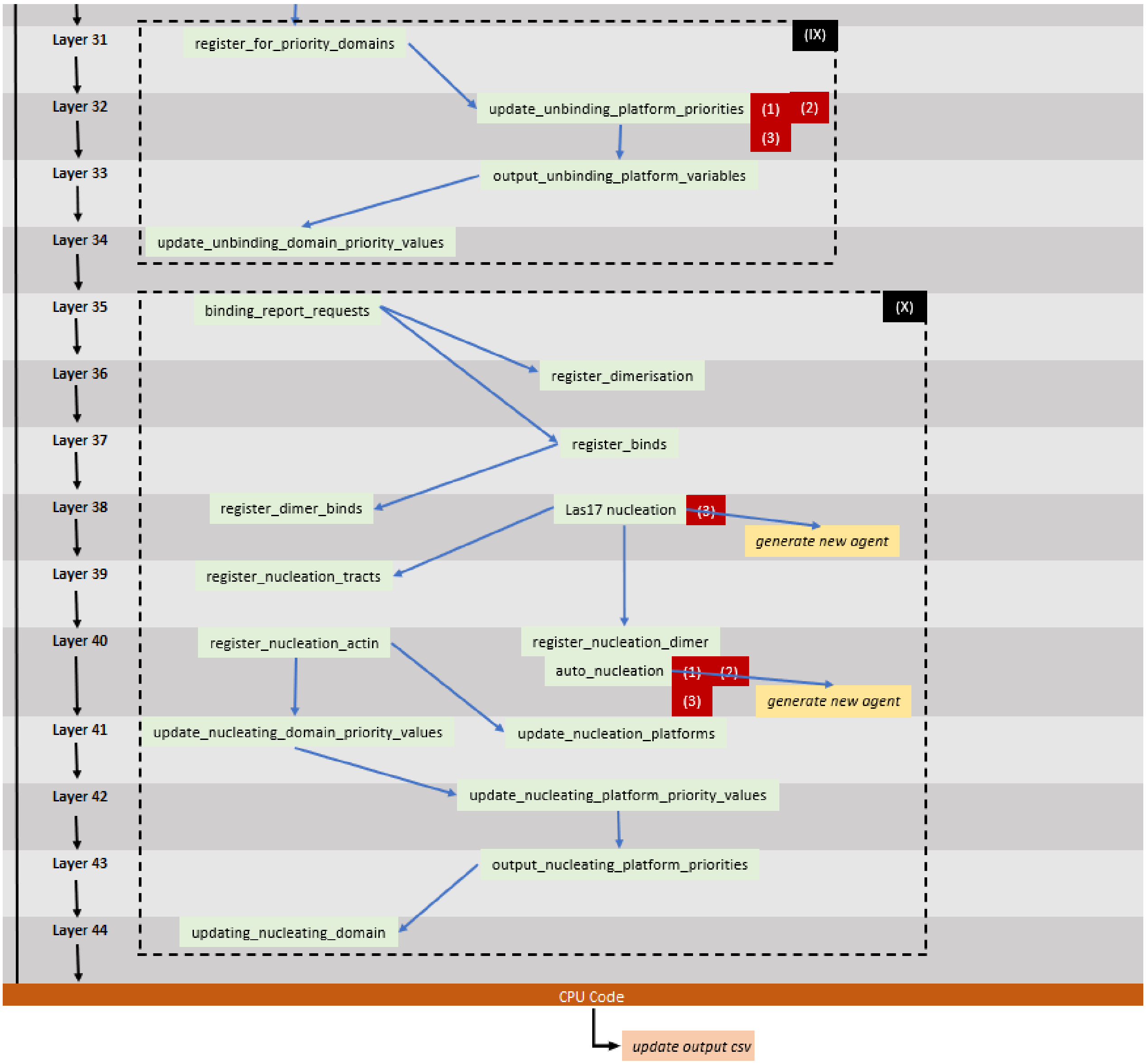

Layer 31
Layer 32
Layer 33
Layer 34
Layer 35
Layer 36
Layer 37
Layer 38
Layer 39
Layer 40
Layer 41
Layer 42
Layer 43
Layer 44
(IX)
register_for_priority_domains
update_unbinding_platform_priorities
(1)
(2)
(3)
output_unbinding_platform_variables
update_unbinding_domain_priority_values
(X)
binding_report_requests
register_dimerisation
register_binds
register_dimer_binds
Las17 nucleation
(3)
generate new agent
register_nucleation_tracts
register_nucleation_actin
register_nucleation_dimer
auto_nucleation
(1)
(2)
(3)
generate new agent
update_nucleating_domain_priority_values
update_nucleation_platforms
update_nucleating_platform_priority_values
output_nucleating_platform_priorities
updating_nucleating_domain
CPU Code
update output csv

Figure 5.4. Detailed model flowchart. The flow of code during runtime is shown by the black line with a new iteration beginning each time Layer 1 is executed. An iteration can also be considered a timestep if agents are able to transition into a resolved state during phase IV (*Transition to timestep*) and access function layers 11-44. Only layers covered by the purple arrow can be accessed during a non-timestep iteration. All layers (including the CPU code) are technically read during non-timestep iterations, however, functions not covered by the purple arrow cannot be executed due to being in the wrong state. Green boxes in the appropriate agent class column (as shown along the top) show the functions and all functions within a layer are executed simultaneously by all applicable agents. Blue arrows represent message lists with the function at the base of the arrow generating the list and the function(s) at the arrowhead reading the messages. Each function can only produce and accept a single message list. Red boxes show when global functions are accessed (1 = boundryCheck, 2 = movement_calc, 3 = random_angle, and 4 = random_distance). Functions can be broadly grouped into ten phases as first given in **figure 5.3** of this chapter. These include (I) reset agents, (II) top priority movement, (III) lower priority movement, (IV) transition to timestep, (V) polymerisation, (VI) interdomain binding, (VII) first priority update, (VIII) unbinding, (IX) second priority update, and (X) actin nucleation.

5.5 XML File

As with Model Alpha, this version also uses an XML file to serve as the framework due to both versions using the same FLAME GPU 1 environment. It defines the structures of simulation elements along with what components run, in what order and with what conditions. Model Beta uses three data classes to define its agents: PlatformAgent, DomainAgent, FilamentAgent. Each data class retains a set of agent variables which are updated during the simulation runtime. These agent types are defined throughout the next sections in this thesis.

5.5.1 Global Variables

There are 31 global variables defined within the XML file for Model Beta. Global variables can only be changed during the CPU code at the end of each time step. They can be accessed – though not edited – by all agent functions where they are called. These variables are used to hold information that is specific to the simulation but not specific agents. This reduces the memory requirement during model construction as shared agent variables can be allocated as a global instead. Below is an overview of all global variables defined in the model (**table 5.1**).

Table 5.1. Global variables of Model Beta

Variable Name	Description	Functions referenced
TIME_COUNTER	Used to hold the current iteration number of the simulation timestep.	CPU code
PRINT_COUNTER	A counter variable that is iterated by 1 every iteration and reset to 0 after an output is made to the csv file. This allows lines to be saved every **X** iterations thus saving memory.	CPU code

Table 5.1. Global variables of Model Beta (continued)

ITERATION	The current iteration following the last timestep conclusion. For example, in the model iteration directly following a timestep conclusion, this value will be 1.	CPU code
ITERATION_PLUS_ONE	This holds the value of “ITERATION + 1” so that the calculation does not have to be undertaken every time a function uses it. ITERATION_PLUS_ONE is referenced by agents (along with their priority value) which have not moved during the *lower priority movement phase* in this timestep to determine when the agent can move. ITERATION_PLUS_ONE is saved by agents into agent variables for lookup by function guard loops	update_platform_timer update_domain_timer
XMAX YMAX ZMAX	These define the maximum allowed axis value for their respective axis before a boundary condition will act upon the agent.	boundryCheck
XMIN YMIN ZMIN	These define the minimum allowed axis value for their respective axis before a boundary condition will act upon the agent.	boundryCheck
BIND_RADII	A 196-unit array holding the binding radii for each possible pairwise interaction.	domain_binding_requests confirm_domain_unbinding_requests
BIND_PROB	A 196-unit array holding the binding probability for each possible pairwise interaction	domain_binding_requests
UNBIND_PROB	A 196-unit array holding the unbinding probability for each possible pairwise interaction	domain_unbinding_requests
DIMER_BIND_RADII	A 6-unit array holding the binding radii for each possible homodimerisation.	domain_binding_requests
BIND_PROB_DIMERISED	A 196-unit array holding the binding probability for each possible pairwise interaction when the domain requesting the bind belongs to a homodimerized platform.	domain_binding_requests
UNBIND_PROB_DIMERISED	A 196-unit array holding the unbinding probability for each possible pairwise interaction when the domain requesting the unbind belongs to a homodimerized platform.	domain_unbinding_requests
DIMER_BIND_PROB	A 6-unit array holding the probability of homodimerisation for each agent type should two of the agents be separated by a distance less than the binding radii.	domain_binding_requests
COOP_DIMER_BIND_PROB	A 6-unit array holding the probability of homodimerisation when a cooperativity criterion is met. In the case of Las17, this is when the agent is bound by actin.	domain_binding_requests

Table 5.1. Global variables of Model Beta (continued)

DIMER_UNBIND_PROB	A 6-unit array holding the probability of each possible homodimer dissociating	Domain_unbinding_requests
COOP_DIMER_UNBIND_PROB	A 6-unit array holding the probability of each possible homodimer dissociating when a cooperativity criterion is met. In the case of Las17, this is when the agent is bound by actin.	domain_unbinding_requests
PP1_PP3_COOPERATIVITY	The unbinding probability between actin and the first or second actin-binding tracts are divided by this value if both the first and second acting-binding tracts are occupied by actin.	domain_unbinding_requests
PP3_PP4_COOPERATIVITY	The unbinding probability between actin and the second or third actin-binding tracts are divided by this value if both the second and third acting-binding tracts are occupied by actin.	domain_unbinding_requests
ACTIN_BARB_INTERACTION_RADIUS	The radius of interaction between actin agents and the barbed end of filament agents	filament_binding_requests unbinding_actin_ from_filament
ACTIN_POINT_INTERACTION_RADIUS	The radius of interaction between actin agents and the pointed end of filament agents	filament_binding_requests unbinding_actin_ from_filament
SEED_UNBIND_PROB	The probability of actin filaments unbinding the Las17 agent which nucleated it.	filament_unbinding_requests
ACTIN_DIAMETER	Filament length is governed by the number of actin subunits multiplied by ACTIN_DIAMETER.	move_free_filaments move_bound_filaments auto_nucleation
FILAMENT_BASE_K	A scaled variable used to define the movement distance for a filament during a movement function.	confirm_filament_ binding_requests actin_unbinding_requests register_binds_and_nucleate auto_nucleation
FILAMENT_ROTATE_BASE_K	A scaled variable used to define the rotational speed for a filament during a movement function.	confirm_filament_ binding_requests actin_unbinding_requests register_binds_and_nucleate auto_nucleation
BARB_POFF	The probability of an actin agent dissociating from the barbed end of a filament during each timestep.	actin_unbinding_requests

Table 5.1. Global variables of Model Beta (continued)

POINT_POFF	The probability of an actin agent dissociating from the pointed end of a filament during each timestep.	actin_unbinding_requests
SIDE_POFF	The probability of an actin agent dissociating from an actin nucleus four subunits in size during each timestep.	actin_unbinding_requests
BARB_PON	The probability of an actin agent binding to the barbed end of an actin filament.	filament_binding_requests
POINT_PON	The probability of an actin agent binding to the pointed end of an actin filament.	filament_binding_requests
SIDE_PON	The probability of an actin agent binding to an actin nucleus three subunits in size.	filament_binding_requests
INITIAL_ACTIN_CONC	The initial concentration of actin is used to determine the rate of salt-mediated actin nucleation during each timestep.	auto_nucleation
FREE_ACTIN_CONC	The current concentration of actin is used to determine the rate of salt-mediated actin nucleation during each timestep.	auto_nucleation
NUCLEATION_RATE	The nucleation rate is used to determine the rate of salt-mediated actin nucleation during each timestep via the following formula. NUCLEATION_RATE*(FREE_ACTIN_CONC /INITIAL_ACTIN_CONC)	auto_nucleation
FILAMENT_PITCH	This value is multiplied by the number of subunits contained within a filament to determine the length of that filament. The value is derived from the per subunit rise within actin filaments	move_free_filaments move_bound_filaments confirm_filament_ binding_requests
D1MAX D2MAX D3MAX D4MAX	A set of four, 10-unit arrays describing the minimum separations of each peptide. D1 holds the *domain1-domain2* distance, D2 holds the *domain2-domain3 distance*, etc.	move_free_platforms move_bound_platforms
D2_CENVEC2_MAX	A 10-unit array describing the maximum separation between domain 2 and the centre of each peptide sequence.	move_free_platforms
D2_CENVEC3_MAX	A 10-unit array describing the minimum separation between domain 3 and the centre of each peptide sequence.	move_free_platforms
D1MIN D2MIN D3MIN D4MIN	A set of four, 10-unit arrays describing the minimum separations of each peptide. D1 holds the *domain1-domain2* distance, D2 holds the *domain2-domain3 distance*, etc.	move_free_platforms move_bound_platforms
COMPLEX_BASE_K	When multiple proteins are bound together, this value (divided by the number of peptides in the complex) is used for the scaling movement variable.	move_free_platforms

Table 5.1. Global variables of Model Beta (continued)

MAX_BOUNDED_ATTEMPTS	This is the maximum number of attempts a PlatformAgent will make when moving a domain flanked by bound domains. If this value is exceeded, then a position equidistant from the flanking domains is chosen. This variable prevents excessive loop iterations.	execute_bounded_movement move_bound_platforms

Table 5.1. Global variables of Model Beta. A list of all the global variables used in Model Beta, their descriptions, and which functions they refer to (including global functions, agent functions, and the CPU function).

5.5.2 Data Class: DomainAgent

Each functional motif of a peptide is represented by a domain agent (e.g., each of the binding motifs within the 300-422 peptide, actin monomers, and SH3 domains). DomainAgents are linked with one of the five vectors of a specific PlatformAgent with the latter governing movement and DomainAgents governing interactions. **Table 5.2** shows the agent variables held in the DomainAgent data class.

Table 5.2. The data class variables of DomainAgents

Variable Name	Description	Default value
Id	A unique identification number used to reference an individual agent.	Unique integer
type	The “type” variable identifies the motif/domain that this agent represents. This allows the agent to access the correct binding rates. Up to 14 types can be set by the user with all possible pairwise interactions between types being set in the global arrays.	0-13

Table 5.2. The data class variables of DomainAgents (continued)

domain_index	An index to identify which of the five PlatformAgent vectors this agent is linked to (0=first domain, 1=second domain etc.)	0-4
platform_id	The unique identification number of the PlatformAgent this domain is linked to. This allows for the updating of vector variables.	Unique integer of a PlatformAgent
platform_priority	The priority value of the peptide this domain belongs to. Lower values move before higher values to maintain spatial coherency. Peptides in a complex are dynamically given a priority value.	1 if unbound and (possibly) higher if in a complex
platform_complex_size	The number of peptides contained within the complex this agent is associated with.	Equal to the number of peptides bound (directly or indirectly) to this peptide
platform_complex_id	The unique identification number of the complex this domain is a part of. The complex id is dynamically generated upon the formation of a new complex and allows agents that are part of the same complex to communicate with each other.	Unique integer of the binding complex this agent is associated with
platform_species	The “species” variable identifies the peptide that this agent is associated with. It consists of Las17 (0), actin (1), cloud SH3s (2), Ysc84 (3), Sla1 (4), and Bzz1 (5).	0-5
vector	This 3-unit vector object holds the position of the domain within the 3D simulation space. The value is obtained each timestep from the agent’s linked PlatformAgent.	random
awaiting_movement_domain	A Boolean variable used to distinguish agents that have moved this timestep (1) and those that have not (0). This variable is used to maintain synchronicity between agents so that some do not enter the timestep functions before others.	0
awaiting_confirmation	A Boolean variable used to signify if an agent attempted to bind another agent during the current timestep. This is used as a function guard so that subsequent functions in the binding phase are only accessed by the appropriate agents.	0
binding_state	A variable to show the current bound state of the agent with 0 being unbound, 1 being bound, 2 representing actin incorporated into a filament (F-actin), and 3 being a Las17 tract bound to a FilamentAgent.	0 if unbound and (possibly) higher if in a complex
dimerised	A variable to show the current homodimerisation state of the parent platform. This variable holds a value of -1 when not dimerised and the id number of the dimerised partner when in a dimer.	-1 if monomeric and higher if dimerised
bound_id	The *id* of the DomainAgent bound if there is an ongoing interaction.	-1 if unbound and higher if in a bound state

Table 5.2. The data class variables of DomainAgents (continued)

bound_type	The *type* of the DomainAgent bound if there is an ongoing interaction.	-1 if unbound and higher if in a bound state
bound_platform	The *platform_id* of the DomainAgent bound if there is an ongoing interaction.	-1 if unbound and higher if in a bound state
bound_filament_id	The *id* of the FilamentAgent bound if there is an ongoing interaction.	-1 if unbound and higher if in a bound state
bound_domain_index	The *domain_index* of the DomainAgent bound if there is an ongoing interaction.	-1 if unbound and higher if in a bound state
requested_id	A temporary variable to hold the *id* of the DomainAgent attempting to bind this agent prior to binding confirmation.	-1
requested_type	A temporary variable to hold the *type* of the DomainAgent attempting to bind this agent prior to binding confirmation.	-1
requested_platform	A temporary variable to hold the *platform_id* of the DomainAgent attempting to bind this agent prior to binding confirmation.	-1
requested_complex	A temporary variable to hold the *platform_complex_id* of the DomainAgent attempting to bind this agent prior to binding confirmation.	-1
requested_domain_index	A temporary variable to hold the *domain_index* of the DomainAgent attempting to bind this agent prior to binding confirmation.	-1
binding_dist	A temporary variable used to hold the distance between this agent and the DomainAgent attempting to bind it, prior to binding confirmation. This is used to prioritise between competing interactions for the same agent.	-1
filament_id	If the agent represents actin and is incorporated into a filament, this variable holds the unique identification number of the FilamentAgent which it is a part of.	-1 if free G-actin and higher if in an F-actin state
filament_subunit_number	This variable is only accessed by actin agents. If it is incorporated into a filament, this variable holds its position within said filament.	0 if free G-actin and different if in an F-actin state
probability_request	A random uniform number between 0 and 1 used to identify which agent in a prospective interaction will act as the dominant partner (i.e., having one agent "take the lead" to prevent A binding B if B is also trying to bind A).	-1
update_priority_domain	A reporter Boolean used to indicate when a complex needs to update its priority values.	0
domain_turn_counter	An agent variable used to hold the current value of the global ITERATION_PLUS_ONE variable for use in a guard loop.	2

Table 5.2. The data class variables of DomainAgents (continued)

bound_report_domain	During the *report and reset phase*, this agent holds the binding information of the DomainAgent using the equation "xagent->type*100) + (xagent->bound_type+1". For example, a value of 006 would indicate that this agent is a Las17 bm1 agent (+0*100) currently bound to an actin DomainAgent (+6). This variable is used by the CPU code to analyse global populations of specific interactions.	0
unbind_dimer_block	A Boolean that when equal to 1, prevents the homodimer from dissociating. This prevents dissociation when bound to an actin filament which is both kinetically extremely unlikely and presents a series of programmatic obstacles.	0
actin_within_dimer	A Boolean accessed by domains with homodimerized parent platforms. The value equals to equal to 1 when an actin agent is bound to the parent platforms' dimerised partner.	0 if no actin within its dimerised partner, otherwise 1
coopratiive_dimerisation_PP1	A Las17 variable to indicate if bm2 of parent platform is actin-bound (10), bound to a non-actin domain (1), or completely unbound (0).	0 if unbound, otherwise either 1 or 10
coopratiive_dimerisation_PP3	A Las17 variable to indicate if bm4 of parent platform is actin-bound (10), bound to a non-actin domain (1), or completely unbound (0).	0 if unbound, otherwise either 1 or 10
coopratiive_dimerisation_PP4	A Las17 variable to indicate if bm5 of parent platform is actin-bound (10), bound to a non-actin domain (1), or completely unbound (0).	0 if unbound, otherwise either 1 or 10
dimer_adjacent_bound_type	A variable accessed by domains with homodimerized parent platforms. Its value is the *type* variable of the domain bound to the equivalent binding motif of the parent platforms' dimerised partner.	0 if not dimerised, otherwise variable

Table 5.2. The data class variables of DomainAgents. A list of all the agent-specific variables found within the DomainAgent data class, their descriptions, and default values given during creation of the initiation XML file.

5.5.3 Data Class: PlatformAgent

Every peptide is represented by a PlatformAgent which can be associated with up to five DomainAgents (one linked with each vector as described in **table 5.3**). PlatformAgents govern the movement of peptides. **Table 5.3** shows the agent variables held in the PlatformAgent data class.

Table 5.3. The data class variables of PlatformAgents

Variable Name	Description	Default value
id	A unique identification number used to reference an individual agent.	Unique integer
species	The “species” variable identifies the peptide that this agent represents. It comprises Las17 (0), actin (1), cloud SH3s (2), Ysc84 (3), Sla1 (4), and Bzz1 (5).	0-5
complex_size	The number of proteins (PlatformAgents) within the same complex as this agent	1 if not in a complex
dimerised_partner_id	The ID variable of the platform homodimerised to this agent.	0 if non dimerised, otherwise higher
complex_id	The unique identification number of the complex this domain is a part of. The complex id is dynamically generated upon the formation of a new complex and allows agents that are part of the same complex to communicate with each other.	Unique integer
k	This variable is scaled based on the diffusion coefficient of the peptide it represents and governs the rate of diffusion.	Value depends upon the peptide size
vector1 vector2 vector3 vector4 vector5	Each of these five variables are fvec3 vector variables holding three coordinate values (x, y, and z). They retain the positions of the possible domains that can be associated with the platform.	Unique integer of the binding complex this agent is associated with
priority	The priority of this peptide. Lower values move before higher values to maintain spatial coherency. Peptides in a complex are dynamically given a priority value.	1 if unbound and (possibly) higher if in a complex
update_priority	A Boolean variable to indicate if a platform needs to update its priority due to a change in the complex it is associated with (e.g., a binding or unbinding event).	0

Table 5.3. The data class variables of PlatformAgents (continued)

awaiting_movement_platform	A Boolean variable used to distinguish agents that have moved this timestep (1) and those that have not (0). This variable is used to maintain synchronicity between agents so that some do not enter the timestep functions before others.	0
vec1_binding_partner vec2_binding_partner vec3_binding_partner vec4_binding_partner vec5_binding_partner	These variables hold the *type* values of any associated domains. Values are used in the *report and reset phase* for both reporting to the output file and nucleating bound actin monomers into actin filaments (if three actin monomers are bound to three adjacent domains simultaneously).	-1
vec1_dimer_partner vec2_dimer_partner vec3_dimer_partner vec4_dimer_partner vec5_dimer_partner	These variables hold the *type* values of any domains bound to the indicated position on its homodimerised partner.	-1
seed_id	The unique identification number of any filament bound to this platform (only used if this platform is Las17).	-1 if unbound to an actin filament and bound if higher
factin	A Boolean variable used by actin platforms to indicate if they have been incorporated into a filament (1) or are free (0).	0 if monomeric and 1 if located in a filament
filament_bound	The ID value of the actin filament bound to this platform if one is bound.	-1 if not bound to a filament, higher if so
NUCLEATION_EVENTS	A counter that increases by one every time this platform nucleates a new actin filament. This is used for data recording purposes.	0
NON_LAS17_NUCLEATION_EVENTS	A counter that increases by one every time an actin filament is nucleated by salt. One Las17 platform can nucleate three random actins if a random number is greater than the nucleation probability of a timestep.	0
platform_turn_counter	An agent variable used to hold the current value of the global ITERATION_PLUS_ONE variable for use in a guard loop.	2
bound_report_platform	During the *report and reset phase*, this agent holds the complex information of platforms. Values are calculated using the formula "(xagent->species*100) + (xagent->complex_size"). For example, a value of 401 would be given for a Sla1 protein (+4*100) that is unbound (+1) and thus not in a multi-platform complex.	0

Table 5.3. The data class variables of PlatformAgents. A list of all the agent-specific variables found within the PlatformAgent data class, their descriptions, and default values given during creation of the initiation XML file.

5.5.4 Data Class: FilamentAgent

A FilamentAgent is dynamically built during the simulation runtime to represent an actin filament. Upon complete dissolution of the filament, the respective FilamentAgent is “killed” during the timestep when this occurred. **Table 5.4** shows the agent variables held in the FilamentAgent data class.

Table 5.4. The data class variables of FilamentAgents

Variable Name	Description	Default value
id	A unique identification number used to reference an individual agent.	Unique integer
size	The number of actin subunits in the filament.	3
k	This variable is scaled based on the diffusion coefficient of the domain it represents and governs the rate of diffusion.	1.2
rotation_k	This variable scales the rate of rotation for a ScaffoldAgent.	2
axis_phi	Axis_phi is a polar coordinate for the barb to point axis.	random
axis_theta	Axis_theta is a polar coordinate for the barb to point axis.	random
filament_centre	An fvec3 vector variable holding three coordinate values (x, y, and z). *filament_centre* marks the centre of the filament is the point of movement and rotation for the overall filament.	random
barb	An fvec3 vector variable holding three coordinate values (x, y, and z). *barb* represents the position of the barbed end.	random
point	An fvec3 vector variable holding three coordinate values (x, y, and z). *point* represents the position of the pointed end.	random
awaiting_movement_filament	A Boolean variable used to distinguish agents that have moved this timestep (1) and those that have not (0). This variable is used to maintain synchronicity between agents so that some do not enter the timestep functions before others.	0

Table 5.4. The data class variables of FilamentAgents (continued)

execute_procession	A reporter Boolean that becomes true once all the DomainAgents and PlatformAgents are ready to enter the timestep. This variable is then used to activate *awaiting_movement_filament* and begin the timestep.	0
barb_actin_no	The current subunit number at the barbed end position. This allows actin agents to identify their position within the filament.	2 (if only a 3-subunit seed)
point_actin_no	The current subunit number at the pointed end position. This allows actin agents to identify their position within the filament.	0 (if only a 3-subunit seed)
barb_dist	A temporary variable holding the distance between the barbed end and its closest binding candidate. This is used to determine which candidate is the closest to avoid multiple filaments (or ends) attempting to bind the same agent simultaneously.	-1
point_dist	A temporary variable holding the distance between the pointed end and its closest binding candidate. This is used to determine which candidate is the closest to avoid multiple filaments (or ends) attempting to bind the same agent simultaneously.	-1
requested_barb_id	A temporary variable used to hold the agent ID attempting to bind this agent prior to binding confirmation.	-1
requested_point_id	A temporary variable used to hold the agent ID attempting to bind this agent prior to binding confirmation.	-1
awaiting_confirmation	A Boolean variable used to signify if an agent attempted to bind another agent during the current timestep. This is used as a function guard so that subsequent functions in the binding phase are only accessed by the appropriate agents.	0
elongation_allowed	A Boolean variable used to signify when the agent represents a nucleus (0) or seed (1).	0 (1 if the filament was either salt nucleated or already present at the start of a simulation)
binding_state	A Boolean variable which shows if the agent is bound (1) or not bound (0) to Las17.	0

Table 5.4. The data class variables of FilamentAgents. A list of all the agent-specific variables found within the FilamentAgent data class, their descriptions, and default values given during both creation of the initiation XML file and agent creation during nucleation.

5.5.5 Function order

The order of functions executed during each simulation iteration is detailed at the bottom of the XML file as detailed within **table 5.5**. Due to the parallel nature of FLAME GPU, a function acts upon all agents once it is called while numerous functions can be called simultaneously. These rounds of program calls are termed function layers, and each layer is iterated through sequentially (from 1 to 38) once per iteration of the simulation. Agents can also occupy either a resolved or unresolved state and these are given within the first column.

Table 5.5. Function layers of Model Beta

Layer number and states accessible (<X>)	Functions called	Brief overview of the layer's function
1 <resolved>	reset_domains, reset_platforms, reset_filaments	If the previous iteration was a timestep iteration, then agents return to the unresolved state.
2 <unresolved>	move_free_platforms move_free_filaments	Peptides in priority block 1 and unbound filaments move.
3 <unresolved>	update_moved_first_domains	DomainAgents belonging to priority block 1 platforms update their vectors.
4 <unresolved>	output_previously_moved_domains update_domain_timer update_platform_timer	Moved domains output their locations and agents update their timer variables (using the global *ITERATION_PLUS_ONE* variable) for the guard loop of layer 5.
5 <unresolved>	move_bound_platforms	Peptides in the priority block equal to the current value of *ITERATION_PLUS_ONE* move using the location outputs of layer 4 to constrain their movements.
6 <unresolved>	update_moved_domains output_bound_filament_possitions	DomainAgents belonging to platforms that moved in layer 5 update their vectors. All platforms which have moved this iteration and are bound to a filament output their location.
7 <unresolved>	move_bound_filaments	FilamentAgents which are bound to Las17 move using the outputs of layer 6.
8 <unresolved>	progress_platforms progress_domains	A global guard loop will shift all domains and platforms into the resolved state (ready for a timestep) if all agents have moved since the last timestep.

Table 5.5. Function layers of Model Beta (continued)

9 <resolved>	progress_tester	FilamentAgents search for a synchronisation message from layer 9. If found, they prepare for a timestep.
10 <unresolved>	progress_filaments	All FilamentAgents prepared for a timestep are moved into the resolved state (ready for a timestep).
11 <unresolved>	output_avalable_domains_for_filament	Unbound DomainAgents of type 5 (actin) output their locations.
12 <resolved>	filament_binding_requests	FilamentAgents identify which type 5 DomainAgents they can bind using the locations given in layer 12.
13 <resolved>	confirm_filament_binding_requests	FilamentAgents communicate between one another to sort any conflicts between binding requests such that the same G-actin is not simultaneously bound by two different filaments.
14 <resolved>	binding_filament	Unbound DomainAgents of type 5 (actin) check if they were bound in layer 14. If so, they update their variables accordingly.
15 <resolved>	output_avalable_domains_for_binding	All unbound DomainAgents output their locations.
16 <resolved>	domain_binding_requests	All unbound DomainAgents identify which other DomainAgents they can bind using the locations given in layer 16.
17 <resolved>	confirm_domain_binding_requests	DomainAgents communicate between one another to sort any conflicts between binding requests to prevent any agent from being simultaneously bound by two or more other agents.
18 <resolved>	register_bound_domains	All unbound DomainAgents check if they were bound in layer 18. If so, they update their variables accordingly.
19 <resolved>	update_binding_domain_priority_values	DomainAgents sharing a platform_complex_id with another agent that interacted in layers 19 and 20 output their binding data.
20 <resolved>	update_binding_platform_priority_values	Binding data from layer 20 is sorted through for each PlatformAgent. This is used to generate new priorities within complexes.
21 <resolved>	output_binding_platform_priorities	Platforms from layer 21 update their complex_size variable and output this along with their ID and new priority.
22 <resolved>	updating_binding_domain	DomainAgents belonging to platforms in layer 22 use the output to update their priority and complex size variables.
23 <resolved>	binding_report_requests	All DomainAgents output their binding data.
24 <resolved>	register_dimerisation	Platforms check if they are now dimerised using the layer 23 message list.
25 <resolved>	register_binds	Platforms update their binding reporter variables and output these as a message.

Table 5.5. Function layers of Model Beta (continued)

26 <resolved>	register_dimer_binds	Domains update their binding context variables using the layer 25 message list.
27 <resolved>	actin_unbinding_requests filament_unbinding_requests output_bound_actin_neighbours	PlatformAgents and FilamentAgents check whether they can unbind filaments and actin domains respectively. If so, they update their variables.
28 <resolved>	unbinding_actin_from_filament unbinding_tracts_from_filament unbinding_filament_from_Las17 domain_unbinding_requests	Actin DomainAgents unbound in layer 27 update their variables. DomainAgents check if their can unbind their interaction partners. Cooperativity is calculated using the *output_bound_actin_neighbours* message list from layer 27.
29 <resolved>	update_unbounded_actin_platforms unbinding_dimers_from_filament confirm_domain_unbinding_requests	Parent PlatformAgents of the actin DomainAgents updated in layer 28 update their variables. Domain agents which could unbind during layer 28 communicate between one another so that only one agent/complex can unbind. Successfully unbinding agents update their variables.
30 <resolved>	register_unbound_domains	All bound DomainAgents check if they were unbound in layer 29. If so, they update their variables accordingly.
31 <resolved>	register_for_priority_update	DomainAgents which unbound in layers 29 and 30 output their binding data.
32 <resolved>	update_unbinding_platform_priorities	Binding data from layer 31 is sorted through for each PlatformAgent. This is used to generate new priorities within complexes.
33 <resolved>	output_unbinding_platform_variables	Platforms from layer 32 update their complex_size variable and output this along with their ID and new priority.
34 <resolved>	update_unbinding_domain_priority_values	DomainAgents belonging to platforms in layer 33 use the output to update their priority and complex size variables.
35 <resolved>	binding_report_requests	All DomainAgents output their binding data (same function as layer 23).
36 <resolved>	register_dimerisation	Platforms check if they are now dimerised using the layer 35 message list (same function as layer 24).
37 <resolved>	register_binds	Platforms update their binding reporter variables and output these as a message (same function as layer 25).
38 <resolved>	register_dimer_binds Las17_nucleation	Domains update their binding context variables using the layer 37 message list (same function as layer 26). Las17 platforms sort through their binding data and if they identify an actin domain bound to PP1, PP3 and PP4 (from either themselves or their dimerised partner), they nucleate a new FilamentAgent.

Table 5.5. Function layers of Model Beta (continue)

39 <resolved>	register_nucleation_tracts	DomainAgents from Las17 platforms that nucleated in layer 38 update their variables accordingly.
40 <resolved>	register_nucleation_dimer register_nucleation_actin auto_nucleation	Las17 platforms which are dimerised to a platform that nucleated actin in layer 39 update their bound filament ID variable. Actin DomainAgents which nucleated in layer 38 update their variables accordingly. A single PlatformAgent nucleates a new Filament agent if permitted according to the rate of salt-mediated nucleation (mimicking salt nucleation in solution).
41 <resolved>	update_nucleation_platforms update_nucleating_domain_priority_values	Parent PlatformAgents of actin DomainAgents that nucleated in layer 38 update their variables accordingly. DomainAgents belonging to complexes which nucleated actin in layer 38 output their binding data.
42 <resolved>	update_nucleating_platform_priority_values	Binding data from layer 41 is sorted through for each PlatformAgent. This is used to generate new priorities within complexes.
43 <resolved>	output_nucleating_platform_priorities	Platforms from layer 42 update their complex_size variable and output this along with their ID and new priority.
44 <resolved>	updating_nucleating_domain	DomainAgents belonging to platforms in layer 43 use the output to update their priority and complex size variables.

Table 5.5. Function layers of Model Beta. A list of all the function layers detailed in the XML model file, the functions contained within and the general purpose of the layer in the context of the simulation.

5.6 Function file

The code for each function mentioned in **section 5.6.5** is written using C in the function file. Guard loops for these functions are defined in the XML file, however, to maintain reading coherency, these will be described alongside their respective functions below.

5.6.1 Global Functions

boundryCheck

This model employed global functions for the same reasons as discussed for Model Alpha. It also preserves the same boundary condition function, *boundryCheck*, from model Alpha (**chapter 3**). However, model beta can also be run in an *In Vivo* mode which replaces the upper and lower Z axis boundaries with elastic conditions to reproduce the effect of a membrane.

movement_calc

The *tandem_x_calc*, *tandem_y_calc*, and *tandem_z_calc* functions of Model Alpha were all combined to make the movement_calc function which accepts a movement distance, theta angle, phi angle, and origin vector and returns the new coordinates as a float3 vector object.

random_angle

The random_angle function multiplies a random uniform number argument between 0 and 1 with 2π to generate a random polar angle. This simple global function was implemented to help visually clean up the code making bug fixing and code modifications easier.

random_distance

Using the same logic as random_angle, this function helps to make code more concise when generating random distances. It accepts the minimum and maximum distance permitted along with a random uniform number between 0 and 1. A simple equation then calculates a random distance value between these bounds.

5.6.2 Layer 1: Timestep reset

Agent's subject	>**currentState** **nextState**>	**Function guard**	>**Message input** **Message output>**
Function: **reset_domains**			
DomainAgent	>Resolved Unresolved>		
Function: **reset_platforms**			
PlatformAgent	>Resolved Unresolved>		
Function: **reset_filaments**			
FilamentAgent	>Resolved Unresolved>		

Layer 1 is only executed following a successful timestep to move all resolved agents into the unresolved state ready for a new timestep preparation. These functions uncouple the next iteration from the next timestep as agents must occupy a resolved state to access timestep-specific functions. State reset occurs at the start of an iteration, rather than the end, so that the data recording CPU code is executed whilst agents are in the resolved state (indicative of a successful timestep).

5.6.3 Layer 2: Moving nondependent agents

Agent's subject	>**currentState** **nextState**>	**Function guard**	>**Message input** **Message output>**
Function: **move_free_platforms**			
PlatformAgent	>Unresolved Unresolved>	Must be in *priority block 1* and *awaiting_movement_platform* = 1	moved_domain_first>
Function: **move_free_filaments**			
FilamentAgent	>Unresolved Unresolved>	Must be unbound to a platform agent and *awaiting_movement_filament* = 1	

Platform agents and Filament agents move using code derived from the ScaffoldAgent movement function of Model Alpha. Platforms first move by generating random phi and theta angles and generating random distances between the five domain vectors that may be associated with a DomainAgent. These agents then use the *random_distance* global function by indexing into the D*(1-4)*MIN and D*(1-4)*MAX global variable arrays using their agent *species* variables. Platforms translationally move using the *centre* vector and then calculate the positions of their five domain vectors (using the random distances calculated earlier) moving outwards. For example, vector2 and vector3 calculate their movement away from the centre vector while vector1 and vector4 calculate their movement away from vector2 and vector3 respectively. Once successfully moved, *awaiting_movement_platform* is set to 0 to prevent repeated access to the function and a message is outputted containing domain locations, the ID of any homodimerised partner platform using the *dimerised_partner_id* variable, and the ID of any bound filament using the *filament_bound* variable (calculated at the end of timesteps). The latter two variables are outputted here as it is the most efficient function to disseminate the information to their child domains.

Using a similar method to the ScaffoldAgents of Model Alpha, FilamentAgents first translationally move using the *centre* vector. They then generate a random barb-point axis and execute a movement step (along this axis) for both the *barb* and *point* vectors with distances equal to half the length of the filament. Finally, *awaiting_movement_filament* is set to 0 to prevent repeated access to the function.

5.6.4 Layer 3: Updating nondependent domain agents

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **update_moved_first_domains**			
DomainAgent	>Unresolved Unresolved>	Must be in *priority block 1* and *awaiting_movement_domain* = 1	>moved_domain_first

Domain agents in *priority block 1* search for their linked PlatformAgent in the message output of layer 2 and update their vector location and dimerisation status (stored using the dimerised variable) from the message according to their *domain_index* variable. For dimerisation domains, if the filament ID number received has a value greater than -1 (i.e., the parent platform is bound to a filament) then *unbind_dimer_block* is set to 1 to prevent the protein dissociating from a dimerised partner. This is important as filaments are often bound by the domains belonging to both parent platforms of a Las17 dimer and navigating this would prove computationally challenging and resource intensive. Instead, Model Beta simplifies the system by only allowing Las17 dimers to dissociate following the dissociation of any bound filaments. Agents finish by updating *awaiting_movement_domain* to 0.

5.6.5 Layer 4: Preparation for dependant movement

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **output_previously_moved_domains**			
DomainAgent	>Unresolved Unresolved>	*awaiting_movement_domain* = 0	previously_moved>
Function: **update_domain_timer**			
DomainAgent	>Unresolved Unresolved>		
Function: **update_platform_timer**			
PlatformAgent	>Unresolved Unresolved>		

DomainAgents which have moved since the last timestep (*awaiting_movement_domain* = 0) output their location and binding data to help with the movement of lower priority blocks. DomainAgents and PlatformAgents update their agent turn counter variables using the *update_domain_timer* and *update_platform_timer* functions to a value equal to the global *ITERATION_PLUS_ONE* variable. This is for use in the guard loops of subsequent layers.

5.6.6 Layer 5: Moving dependent agents

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **move_bound_platforms**			
PlatformAgent	>Unresolved Unresolved>	*awaiting_movement_domain* = 1 and *priority* equals the *platform_turn_counter*	>previously_moved moved_domain>

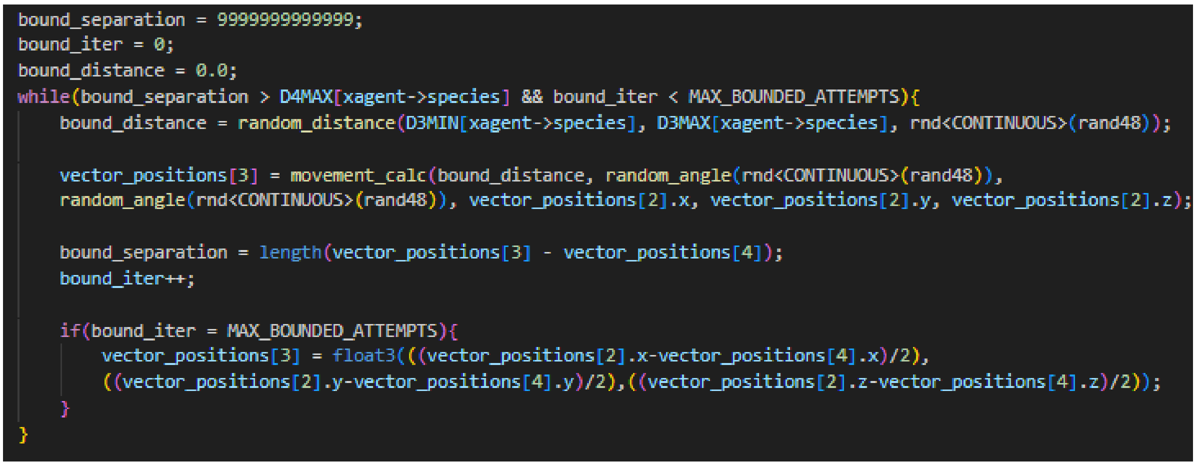
Layer 5 focuses on moving platforms with a priority greater than 1. It does this by moving agents with a priority equal to the *iteration number + 1*. This means that priority block 2 will first execute *move_bound_platforms* on the iteration directly following a successful timestep (iteration 1). The subsequent iterations will execute iteratively higher priority blocks (e.g., block 3, block 4 etc.) until all PlatformAgents have successfully moved and the timestep can begin.

This function is one of the longest and most complex functions of the model. It starts similar to *move_free_platforms* by generating random angles and distances. However, the function also begins by generating an array of five, float3 vector objects with all coordinates set at zero. Agents then run through all the message outputs from layer 4 to check if any moved domains (those which have successfully moved since the timestep) are bound to any of their domain locations. If so, the zeroed coordinate for that vector is set at the same value as the bound domain.

The function enters a complex series of nested "if" statements which check which domain vectors in the coordinate array are zeroed to determine how the movement of the agent is constrained. For example, if vector2 and vector4 are both bound to a higher priority agent, then the position of vector3 is physically constrained by its neighbours as movement may break spatial coherency.

In summary of the process, movements are performed using one neighbour as the movement origin. The distance between the new position and the second neighbour is measured to ensure that this distance is not greater than allowed. This occurs inside a "while loop" until either a position is found which satisfies all spatial constraints, or the number of attempts is greater than the *MAX_BOUNDED_ATTEMPTS* global variable. In the case of the latter, the position of the constrained domain is set equidistant from both neighbours to prevent excessive/infinite looping. The code below shows how this scenario is solved (**code 5.1**).

Code 5.1



```
bound_separation = 9999999999999;
bound_iter = 0;
bound_distance = 0.0;
while(bound_separation > D4MAX[xagent->species] && bound_iter < MAX_BOUNDED_ATTEMPTS){
    bound_distance = random_distance(D3MIN[xagent->species], D3MAX[xagent->species], rnd<CONTINUOUS>(rand48));

    vector_positions[3] = movement_calc(bound_distance, random_angle(rnd<CONTINUOUS>(rand48)),
    random_angle(rnd<CONTINUOUS>(rand48)), vector_positions[2].x, vector_positions[2].y, vector_positions[2].z);

    bound_separation = length(vector_positions[3] - vector_positions[4]);
    bound_iter++;

    if(bound_iter = MAX_BOUNDED_ATTEMPTS){
        vector_positions[3] = float3(((vector_positions[2].x-vector_positions[4].x)/2),
        ((vector_positions[2].y-vector_positions[4].y)/2),((vector_positions[2].z-vector_positions[4].z)/2));
    }
}
```

In the case of bound domains flanking several vectors (e.g., vector2 and vector5 bound, constraining both vector3 and vector4), an iterative approach is taken with multiple “while loops”. This code differs only slightly to what is explained above and is executed until the requirements for complete spatial coherence are achieved.

FLAME GPU 1 does not allow the calling of random numbers from within global functions. This unfortunate drawback resulted in a large degree of code repetition throughout this function. However, all possible binding scenarios were independently tested, and the function performs as expected for each of them. Following completion of the function a message is outputted containing domain locations, the ID of any homodimerised partner platform using the *dimerised_partner_id* variable, and the ID of any bound filament using the *filament_bound* variable (calculated at the end of timesteps). As with layer 2, the latter two variables are outputted here as it is the most efficient function for higher priority value platforms to disseminate the information to their child domains.

5.6.7 Layer 6: Updating dependent domain agents

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **update_moved_domains**			
DomainAgent	>Unresolved Unresolved>	*platform_priority* equals the *platform_turn_counter*	>moved_domain
Function: **output_bound_filament_possitions**			
PlatformAgent	>Unresolved Unresolved>	Las17 platform that must be bound to an actin filament	bound_filament_vectors>

DomainAgents belonging to platforms which moved in layer 5 update their vectors and dimerisation status (stored using the dimerised variable). Using the same code as described in *update_moved_first_domains* (layer 3), dimerisation domains also set the value of *unbind_dimer_block* to 1 should the filament ID received have a value greater than -1

PlatformAgents bound to actin filaments (in a manner only achievable by Las17 objects which nucleated said filament) output their vectors. This is because bound FilamentAgents always default to the vector position of their bound Las17 platform.

5.6.8 Layer 7: Updating bound filament locations

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **move_bound_filaments**			
FilamentAgent	>Unresolved Unresolved>	Agent must be bound to a nucleating platform	>bound_filament_vectors

The vector positions given in layer 6 are iterated through by bound actin filaments. If their id number matches, then the filament will move to the coordinates of its bound Las17 platform.

5.6.9 Layer 8: Timestep progression: Peptides

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **progress_platforms**			
PlatformAgent	>Unresolved Resolved>	**All** PlatformAgents must have moved since the last timestep (awaiting_movement_platform = 0)	record_ticket>
Function: **progress_domains**			
DomainAgent	>Unresolved Resolved>	**All** DomainAgents must have moved since the last timestep (awaiting_movement_domain = 0)	

FLAME GPU 1 allows the use of global guard loops that only allow the function to be executed if all agents of the same type satisfy the same condition. In the case of *progress_platforms*, all PlatformAgents must have *awaiting_movement_platform* value of "0" indicating that they have all completed the movement phase. The function releases a message containing the value of "1" before moving all agents into the resolved state ready for a new timestep. *progress_domains* is executed at the same time because, if all platforms have moved, all domains must also have moved (every domain is linked to a platform). This function moves DomainAgents into a resolved state.

5.6.10 Layer 9: Timestep progression: Filaments 1

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **progress_tester**			
FilamentAgent	>Unresolved Unresolved>	*platform_priority* equals the *platform_turn_counter*	>record_ticket

It is critical that all agent types enter the timestep phase together to maintain synchronicity. Therefore, *progress_tester* searches the layer 8 message list for values of "1" and, if so, sets the value of the FilamentAgent variable *execute_procession* to "1". Finding this value in the layer 8 output indicates that PlatformAgents have executed the progress_platforms function and have thus entered a resolved state.

5.6.11 Layer 10: Timestep progression: Filaments 2

Agent's subject	>currentState nextState>	Function guard	>Message input Message output>
Function: **progress_filaments**			
FilamentAgent	>Unresolved Resolved>	**All** FilamentAgents must an *execute_procession* value of "1"	

FilamentAgents enter a resolved state once all agent classes are synchronised ready to begin the timestep phase (via holding *execute_procession* values of "1").

5.6.12 Layer 11: Outputting actin monomers for filaments

Agent's subject	>currentState nextState>	Function guard	>Message input Message output>
Function: **output_avalable_domains_for_filament**			
DomainAgent	>Resolved Resolved>	Agents must be of type 5 (actin) and not part of a filament	avalable_actin_location>

Type 5 DomainAgents that are not incorporated into a filament (G-actin) which are either unbound (monomeric) or bound to a type 8 DomainAgent (bound to a Ysc84 YAB domain) are possible binding candidates for actin filaments. Therefore, in preparation for the polymerisation phase, these agents output their location as a message.

5.6.13 Layer 12: Actin polymerisation: requests

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **filament_binding_requests**			
FilamentAgent	>Resolved Resolved>		>avalable_actin_location priority_actin_ticket>

FilamentAgents search the layer 11 message list for G-actin DomainAgents close enough to bind. This uses code derived from Model Alpha (see **section 3.6.4** for in depth walkthrough) to allow for both barbed end and pointed end binding. In overview, this function iterates through all the messages and identifies the closest agents within the interaction radius (if there are any) to both the barbed and pointed end. Binding data is then saved in the temporary *requested* agent variables for sorting. If the FilamentAgent is still classified as a nucleus and not a seed (using the *elongation_allowed* agent variable), then binding of a fourth subunit will use the *SIDE_PON* variable to account for the possible linear tandem nucleating mechanism of Las17 (definition of nucleus and seed are given in **section 5.2.1**).

5.6.14 Layer 13: Actin polymerisation: confirmation

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **confirm_filament_binding_requests**			
FilamentAgent	>Resolved Resolved>	The agent must have found a prospective binding partner in layer 12	>priority_actin_ticket actin_ticket>

This function identifies which interactions are approved should any DomainAgents be the simultaneous target of multiple filaments. The code of *filament_binding_requests* is derived from Model Alpha (see **section 3.6.5** for in depth walkthrough). Interactions approved for binding then occur and the appropriate variables are updated. FilamentAgents that are defined as nuclei can update their *elongation_allowed* variable to redefine themselves as a seed should the size of the complex be five or greater.

5.6.15 Layer 14: Update polymerised monomers

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **binding_filament**			
DomainAgent	>Resolved Resolved>	Agents must be of type 5 (actin) and not part of a filament	>actin_ticket

All unbound actin DomainAgents iterate through the messages of layer 14. If their ID is identified, then the agent will update to a bound status and rewrite its variables accordingly.

5.6.16 Layer 15: Outputting unbound domains

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **output_avalable_domains_for_binding**			
DomainAgent	>Resolved Resolved>	Agents must be unbound	avalable_domain_location>

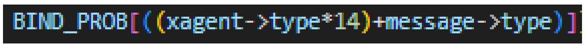
All unbound DomainAgents output their location and relevant dimerisation variables (e.g., cooprative_dimerisation_PP1, etc.) ready for the binding phase. They also send a random number within the message for use within the binding probability checks.

5.6.17 Layer 16: Domain binding: requests

Agent's subject	>currentState nextState>	Function guard	>Message input Message output>
Function: **domain_binding_requests**			
DomainAgent	>Resolved Resolved>	Agents must be unbound	>avalable_domain_location priority_domain_ticket>

Each unbound domain agent then assesses the binding potential for all other unbound domains. This code is derived from layer 12 (**section 5.7.13**) and likewise searches for DomainAgents which are within the interaction radius and with a random number (submitted in the layer 15 message list) less than the binding probability. Binding probabilities for specific interactions can be obtained from multiple possible global arrays depending upon the dimerisation status of their parent platform and whether the agent executing the function is a dimerisation domain. Non-dimerised, protein motif domains use the *BIND_PROB* global array using the following code (**code 5.2**).

Code 5.2



```
BIND_PROB[((xagent->type*14)+message->type)]
```

FLAME GPU 1 does not support 2D arrays. However, the *BIND_PROB* global variable (along with the other 196-unit arrays of Model Beta) reproduce the effect in a 1D array by combining both prospective interaction partners to get a unique index.

Dimerised protein motif domains use the *BIND_PROB_DIMERISED* global array which may contain alternative probabilities that reflect the dimerisation status (e.g., actin and Las17 interactions have a higher probability in the *BIND_PROB_DIMERISED* array than the *BIND_PROB* array). On the other hand, dimerisation domains obtain their binding probability from the DIMER_BIND_PROB global array using their *platform_species* variable. This array holds the probability of homodimerisation for each simulated protein when their binding radii overlap.

Model Beta allows for dimerisation cooperativity such that actin being bound at one tract may increase the dimerisation affinity assuming that the equivalent tract of the prospective dimerisation partner is free to interact with the actin. Theoretically, if the other tract is bound by another agent, then it would be unavailable to actin and thus unable to receive the cooperative benefit when dimerising. This process was regulated by making the dimerisation domains of Las17 sum their *cooprative_dimerisation* variables with those of the prospective binding partner. These summations are undertaken for each actin-binding polyproline tract individually (i.e. (*xagent->cooprative_dimerisation_PP1* + *message->cooprative_dimerisation_PP1*), (*xagent->cooprative_dimerisation_PP3* + *message->cooprative_dimerisation_PP3*), and (*xagent->cooprative_dimerisation_PP4* + *message->cooprative_dimerisation_PP4*)). Because *cooprative_dimerisation* variables have a value of 0 if unbound, 1 if SH3-bound and 10 if actin-bound, then if any summations equal to 10, then at least one polyproline is tract is actin bound whilst its equivalent tract on the other Las17 is unbound. In this case, the dimerisation domain searches the COOP_DIMER_BIND_PROB global array using its *platform_species* variable to obtain a binding probability.

5.6.18 Layer 17: Domain binding: confirmation

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
		Function: **confirm_domain_binding_requests**	
DomainAgent	>Resolved Resolved>	Agents must have identified a possible binding partner in layer 17	>priority_domain_ticket confirmed_domain_ticket>

This function is derived from layer 13 (**section 5.7.14**) and likewise checks to identify which interactions are approved should more than one DomainAgent attempt to bind the same target agent simultaneously. Like with actin polymerisation, the parallel nature of Model Beta requires checks to prevent multiple agents attempting an exclusive action.

An important consideration for using functions that employ parallel binding within the same agent class is that both agents may attempt to simultaneously bind each other. To avoid this and maintain unidirectional binding, agents can only attempt to bind domains which generated a smaller random number during layer 16 for calculating the binding probability (stored using the *probability_request* variable). To allow for efficient priority recalculation, only one DomainAgent per complex is allowed to bind during this layer.

5.6.19 Layer 18: Update bound domains

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **register_bound_domains**			
DomainAgent	>Resolved Resolved>	Agents must be unbound	>confirmed_domain_ticket binding_domain_update_priority>

All unbound DomainAgents iterate through the messages of layer 17. If their ID is identified, then the agent will update to a bound status and rewrite its variables accordingly. Following a binding event, the network of platform interactions and thus composition of a complex may change. Therefore, priority values must be redetermined and this begins by outputting messages containing the *complex_id* value for both interaction partners.

5.6.20 Layer 19: Binding priority update: interaction network

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **update_binding_domain_priority_values**			
DomainAgent	>Resolved Resolved>		>binding_domain_update_priority binding_platform_update_priority>

The first stage of redetermining priority values is formulating a comprehensive view of all interactions within a complex. Therefore, during *update_binding_domain_priority_values* all DomainAgents iterate through the messages of layer 18 and search for their *complex_id*. Domains which successfully match their variables with either of the two *complex_id* values set their *complex_id* to that of the dominant binding partner within the layer 17 interaction. They then output their binding information and both *complex_id* values from their input message as an outgoing message. DomainAgents which initiated the bind in layers 16/17 are given the "priority_prime" message value of 1.

5.6.21 Layer 20: Binding priority update: determining priority

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **update_binding_platform_priority_values**			
PlatformAgent	>Resolved Resolved>		>binding_platform_update_priority binding_complex_id_counter>

Redetermining priority values is one of the most complex and lengthy functions of this simulation. A flowchart detailing the steps of this function is shown in **figure 5.5**. It begins by iterating through the message list of layer 19. Entries with complex ID values matching the *complex_id* of the agent (either of the two given in the message) are identified. Functions in FLAME GPU 1 cannot read a message more than one time. Therefore, the platform ids and their bound PlatformAgents are saved for each matching message in a local variable list. This first message loop also initiates the *past_list* variable with the *platform_id* entry of the dominant binding partner within the layer 17 interaction involving this complex. If this ID matches the ID of the agent executing the function, then the function is ended, and *priority* value set to 1. Agents which never identified their complex_id within the message list also end the function at this point.

All agents continuing the function (via a local variable value of *update_required == 1*) are platforms associated with one of the binding partners. To identify their ultimate priority values, a “while loop” is executed which walks through the complex by identifying all platforms bound to agents within the *past_list*. These newly identified agents are saved in the *current_list*. At the end of the while loop iteration, *past_list* is cleared and repopulated by the contents of *current_list* (*current_list* is subsequently cleared). This platform-by-platform walking continues until the agent executing the function identifies its own ID number in the *current_list*. The number of “while loop” iterations required to reach this conclusion (+ 1) defines the priority value.

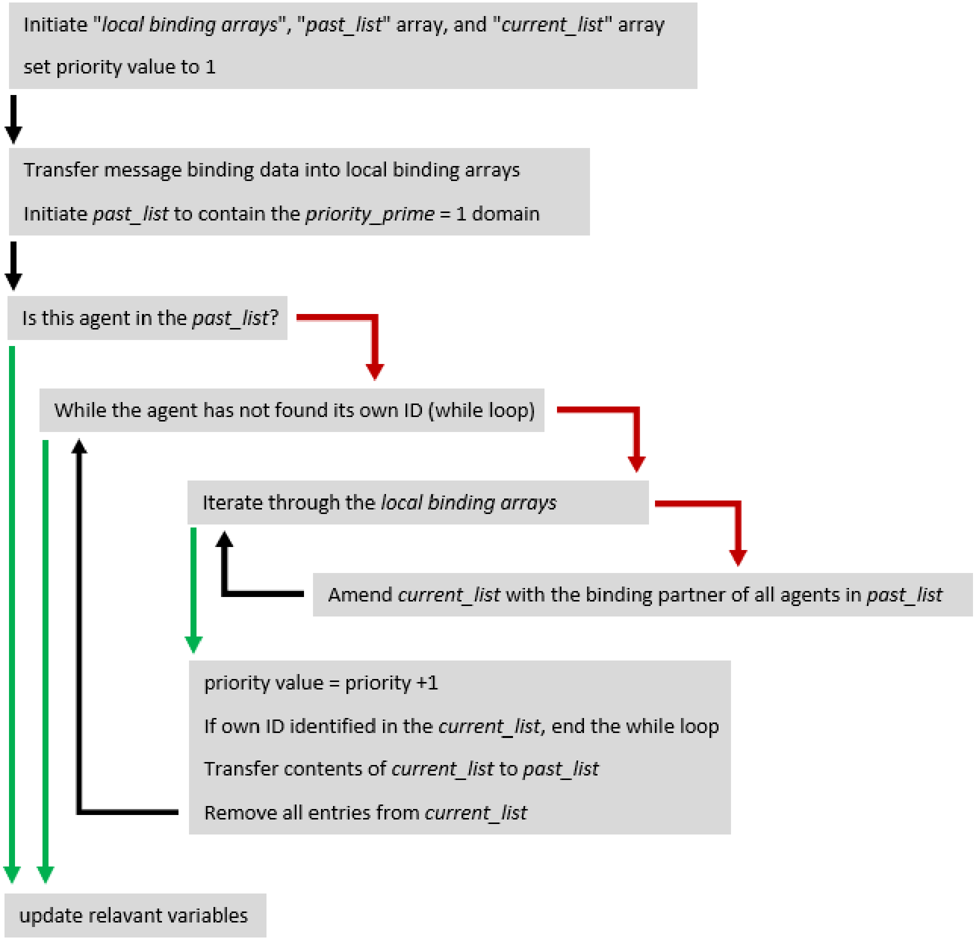

Initiate "local binding arrays", "past_list" array, and "current_list" array
set priority value to 1
Transfer message binding data into local binding arrays
Initiate past_list to contain the priority_prime = 1 domain
Is this agent in the past_list?
While the agent has not found its own ID (while loop)
Iterate through the local binding arrays
Amend current_list with the binding partner of all agents in past_list
priority value = priority +1
If own ID identified in the current_list, end the while loop
Transfer contents of current_list to past_list
Remove all entries from current_list
update relavant variables

Figure 5.5. Simplified flowchart of priority recalculation following the binding phase. Arrows show the direction of code with red paths being followed when a condition of a statement is not met, green when the condition is met, and black being followed regardless. Each agent essentially “walks” along the complex (starting from priority prime which is described in layer 19) until its ID number is identified. The agent priority is incremented every time the while loop is completed which results in the agent identifying its priority position relative to priority prime.

5.6.22 Layer 21: Binding priority update: disseminating priority values

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **output_binding_platform_priorities**			
PlatformAgent	>Resolved Resolved>	Agents must have updated their priority values in layer 20	>binding_complex_id_counter binding_final_update>

PlatformAgents which redetermined their priority values this timestep iterate through the output of layer 20 to count how many PlatformAgents are located within their complex. This value is then saved in the *complex_size* variable. Agents then output a message containing information regarding the complex, including size and priority.

5.6.23 Layer 22: Binding priority update: updating domains

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **updating_binding_domain**			
DomainAgent	>Resolved Resolved>	Agents must have output a message during layer 19	>binding_final_update

DomainAgents which belong to PlatformAgents that redetermined priority and size values iterate through the message list of layer 21 until they identify a message from their parent platform. *platform_complex_size* and *platform_priority* variables are updated accordingly.

5.6.24 Layer 23: Updating DomainAgent context 1: uploading data

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **binding_report_requests**			
DomainAgent	>Resolved Resolved>		upload_partners>

The context of DomainAgents, including the *cooprative_dimerisation* and *actin_within_dimer* variables may change following the binding stage and so must be updated. This begins by every DomainAgent outputting a message containing its binding data.

5.6.25 Layer 24: Updating DomainAgent context 1: reassessing dimerisation status

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **register_dimerisation**			
PlatformAgent	>Resolved Resolved>	Only proteins which have the possibility to form homodimers	>upload_partners

PlatformAgents must first identify whether they are now homodimerised before sorting through the binding data from layer 23. This is because the data processing requires input from both the agents' child domains and the child domains of any dimerised partner. Therefore, platforms must identify any newly dimerised partner platform prior to the sorting function. PlatformAgents read the messages from their child dimerisation domains and update their *dimerised_partner_id* to the *bound_platform* variable of the child.

5.6.26 Layer 25: Updating DomainAgent context 1: data processing

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **register_binds**			
PlatformAgent	>Resolved Resolved>		>upload_partners dimerisation_updator_list>

PlatformAgents run through the binding data messages from layer 23 having updated their dimerisation status via the previous layer. This allows platforms to extract the binding data of their child domains along with the child domains of any dimerised partner. This is then used to set the *vec_bidning_partner* and *vec_dimer_partner* reporter variables representing each of the five possible binding motifs that can be linked to the platform.

5.6.27 Layer 26: Updating DomainAgent context 1: informing the relevant domains

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **register_dimer_binds**			
DomainAgent	>Resolved Resolved>	Only proteins which have the possibility to form homodimers	>dimerisation_updator_list

DomainAgents read the output messages of layer 25. Variables which allow domains to better understand their environmental context (e.g., what is bound at their equivalent position by any dimerised platforms etc.) by updating the dimer_adjacent_bound_type, actin_within_dimer, and cooprative_dimerisation variables (for PP1, PP3, and PP4).

5.6.28 Layer 27: Unbinding: filament requests

Agent's subject	**>currentState nextState>**	**Function guard**	**>Message input Message output>**
Function: **actin_unbinding_requests**			
FilamentAgent	>Resolved Resolved>		freed_actins>
Function: **filament_unbinding_requests**			
PlatformAgent	>Resolved Resolved>	Platform must be bound to an actin filament which it nucleated	freed_seeds>
Function: **output_bound_actin_neighbours**			
DomainAgent	>Resolved Resolved>	Must be bound to a type 5 domain whilst itself being type 0-4 (Las17)	outputting_coopratvity>

The barbed and pointed end of FilamentAgents check whether they can release an actin agent. If so, a message is written which contains the released subunit numbers. Subunit unbinding probabilities for filaments defined as nuclei rather than seeds (*elongation allowed* = 0) take their subunit unbind probability from the *SIDE_POFF* global variable. If the number of subunits (size variable) drops below "3", then the function returns "death" which removes the agent from the simulation. The remaining subunits are then also released in the output message.

In *filament_unbinding_requests*, platforms bound to actin filaments (nucleating Las17s) compare a random number against the probability of unbinding. If the random number is lower, the filament's ID is given as a message and *seed_id* is set to "-1".

DomainAgents representing the Las17 motifs (types 0-4) which are also bound to actin agents release their binding details and *domain_index* value as a message for later use in calculating cooperativity.

5.6.29 Layer 28: Unbinding: domain requests and actin unbinding

Agent's subject	>currentState nextState>	Function guard	>Message input Message output>
Function: **unbinding_actin_from_filament**			
DomainAgent	>Resolved Resolved>	Must be incorporated into an actin filament (i.e., F-actin)	>freed_actins f_to_g_ticket>
Function: **unbinding_tracts_from_filament**			
DomainAgent	>Resolved Resolved>	Domain must be bound to an actin filament which it nucleated	>freed_seeds
Function: **unbinding_filament_from_Las17**			
FilamentAgent	>Resolved Resolved>		>freed_seeds
Function: **domain_unbinding_requests**			
DomainAgent	>Resolved Resolved>	Agent must be bound	>outputting_cooprativity freed_domains>

DomainAgents representing F-actin check the *freed_actins* message list of layer 27. Agents which identify themselves, unbind from the filament, and make a random movement with a distance equal to the interaction radius.

Filament-bound DomainAgents belonging to the nucleating tracts of Las17 (bm2, bm4, and bm5) read the *freed_actins* message list in search of themselves. If successfully located, the domains unbind the filament and update their variables accordingly.

FilamentAgents search the *freed_seeds* message list for themselves. If successfully located, filaments unbind from the nucleating platform and update their variables accordingly.

DomainAgents in *domain_unbinding_requests* check whether they can release their bound agent. This follows similar code to **section 5.7.24** with the addition of cooperativity. The "probability of unbinding" is taken from the appropriate 196-unit array using code analogous to that presented in code 2 (although the *message->type* is replaced with the local *bound_type* variable). Non-dimerised binding motif domains use the UNBIND_PROB array while dimerised binding motif domains use the UNBIND_PROB_DIMERISED. Dimerisation domains access the appropriate 6-unit global array using their platform_species variable as an index. Las17 homodimers which have bound an actin agent are assumed to dissociate with a lower rate because of cooperativity. Dimerisation domains which have an actin present within either the parent platform or the platform of the dimerised partner use the COOP_DIMER_UNBIND_PROB. Otherwise, DIMER_UNBIND_PROB is accessed to obtain the unbinding probability of the dimer.

The chosen unbinding probability is then divided by the appropriate cooperativity global variable if iteration through the layer 27 message list identifies an adjacent actin-binding motif bound to an actin agent. bm2 and bm4 bound use *PP1_PP3_COOPERATIVITY* whilst bm4 and bm5 bound use *PP3_PP4_COOPERATIVITY*. As opposed to **section 5.7.24**, agents which are capable of unbinding do not do so immediately. The reasoning for this is that, like during the binding phase, multiple simultaneous changes to complexes can disrupt the method of priority redetermination. Therefore, these agents only update a single variable (*probability_request*) to hold the random number value generated during the function. A message is output containing this number along with identifiable features of the agent.

5.6.30 Layer 29: Unbinding: confirmation

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **update_unbounded_actin_platforms**			
PlatformAgent	>Resolved Resolved>	Agents must be of species 1 (Las17)	>f_to_g_ticket
Function: **unbinding_dimers_from_filament**			
PlatformAgent	>Resolved Resolved>	The platform is bound to a filament	>freed_seeds
Function: **confirm_domain_unbinding_requests**			
DomainAgent	>Resolved Resolved>	The agent must have been in layer 28	>freed_domains confirmed_freed_domains>

The parent PlatformAgents of actin agents which were released from filaments during layer 28 identify themselves using the *f_to_g_ticket* layer 28 message list and update their vector locations accordingly.

PlatformAgents check the *freed_seeds* message list for references to their bound filament. If there is a match, the *filament_bound* variable is set to -1. This function exists to update the dimerisation partners of a Las17 agent which unbound their actin filament during layer 27.

DomainAgents which are successfully capable of unbinding this timestep iterate through the message list of layer 28. If no other agents within the same complex (and with a lower random number probability) are identified, then the agent will continue with the unbind, update the appropriate variables, and output a message. Agents which identified a lower probability random number within the same complex do not unbind. This ensures that more than one unbinding event never occurs within the timestep for the sample complex. The probability of unbinding (even for weak interactions) is sufficiently low that the chance of two unbinding events being simultaneously attempted within the same complex is extremely low.

5.6.31 Layer 30: Unbinding: domain unbinding

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **register_unbound_domains**			
DomainAgent	>Resolved Resolved>	Agent must be bound	>confirmed_freed_domains priority_registration>

Bound DomainAgents read the layer 29 unbind message list for references to themselves. Agents which succeed in identifying themselves, unbind and change their variables accordingly.

5.6.32 Layer 31: Unbinding priority update: interaction network

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **register_for_priority_update**			
DomainAgent	>Resolved Resolved>		>priority_registration platform_priority_update>

The unbinding phase then enters a series of priority determining functions similar to the binding phase. This begins with the *register_for_priority_update* function in which DomainAgents identify whether they are within the same complex as agents that have unbound this timestep. Agents which can identify themselves as one of the unbinding domains additionally identify themselves with a "partner number" ("1" for the dominant agent which initiated the unbind and "2" for the recipient of the unbind request in layer 30). Two new complex IDs are chosen using the ID of the dominant agent for *new_complexOne_id* and the ID of the recipient as *new_complexTwo_id*. Agents package this information, along with their binding data, as a message.

5.6.33 Layer 32: Unbinding priority update: determining priority

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **update_unbinding_platform_priorities**			
PlatformAgent	>Resolved Resolved>		>platform_priority_update complex_size_count >

This code is highly derivative of the function which redetermined priorities following binding (**section 5.7.21**). However, it contains the additional complexity of three distinct outcomes that cannot be pre-determined. The first is that the complex may not break into two as additional interlinking domains may be present. Furthermore, each platform involved has two potential complexes (and thus priority values) they may be located within. A visual demonstration of this is given in **figure 5.6**.

Like during the binding phase, dynamic lists are constructed that “walk” through the complex from an origin. However, two lists are developed during this function with one “walking out” from the dominant unbind partner whilst the second “walks out” from the unbind recipient. These two list constructions are also linked to their own priority counter. Three possible outcomes may result from a PlatformAgent executing this function.

First, is that both walking lists “meet up” and connect via the same platform. This suggests that additional interdomain interactions are maintaining the structural complex. During this case, priority values taken from the binding recipient walk are applied to all platforms within the complex.

Second, both lists continue until completion without “meeting up” and the platform identifies itself within the binding recipient list. Here, the agent accepts the binding priority value from the binding recipient walk and complex ID of *new_complexTwo_id* (layer 31 message).

Figure 5.6. Simplified flowchart of priority recalculation following the unbinding phase

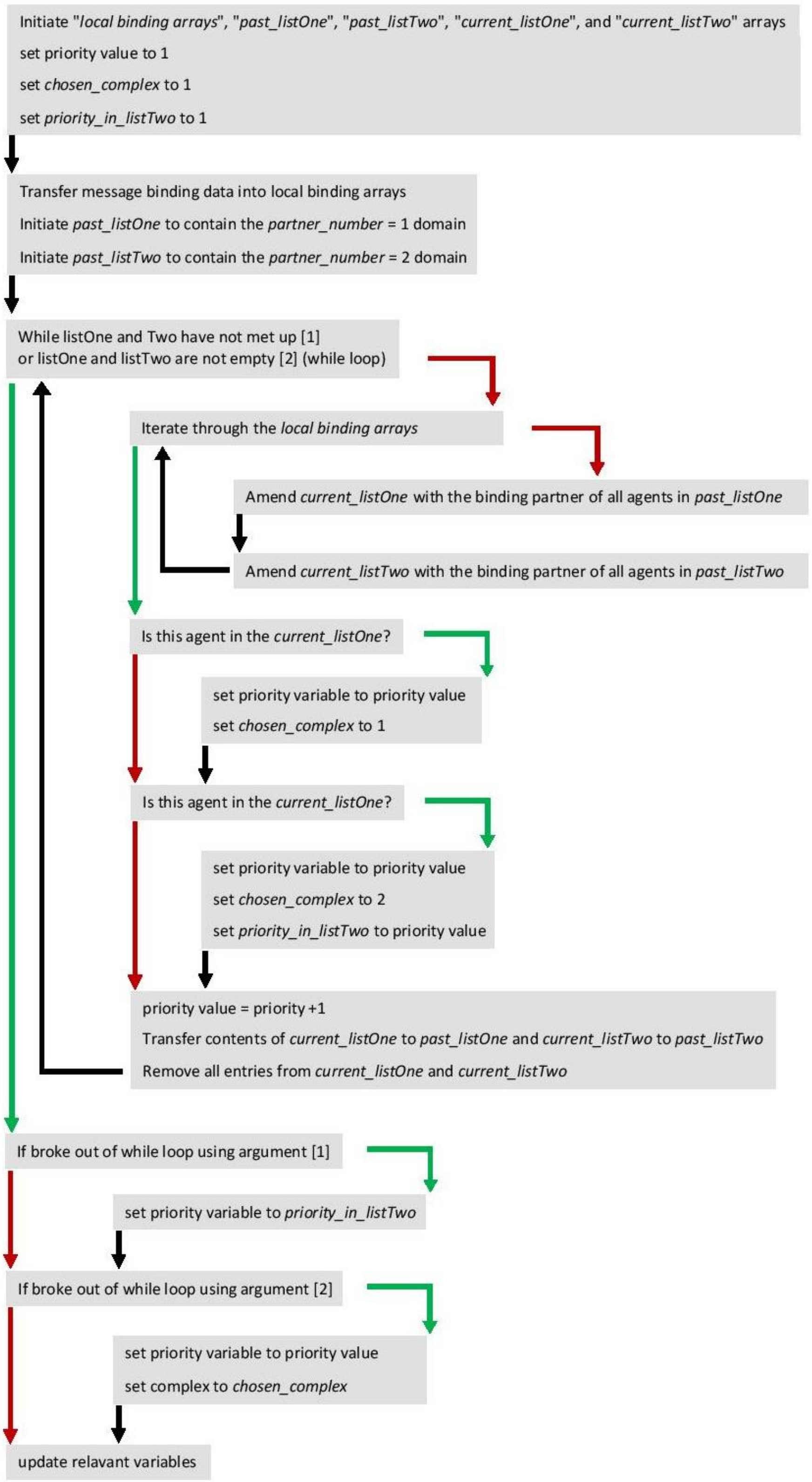

Initiate "*local binding arrays*", "*past_listOne*", "*past_listTwo*", "*current_listOne*", and "*current_listTwo*" arrays
set priority value to 1
set *chosen_complex* to 1
set *priority_in_listTwo* to 1
Transfer message binding data into local binding arrays
Initiate *past_listOne* to contain the *partner_number* = 1 domain
Initiate *past_listTwo* to contain the *partner_number* = 2 domain
While listOne and Two have not met up [1]
or listOne and listTwo are not empty [2] (while loop)
Iterate through the *local binding arrays*
Amend *current_listOne* with the binding partner of all agents in *past_listOne*
Amend *current_listTwo* with the binding partner of all agents in *past_listTwo*
Is this agent in the *current_listOne*?
set priority variable to priority value
set *chosen_complex* to 1
Is this agent in the *current_listOne*?
set priority variable to priority value
set *chosen_complex* to 2
set *priority_in_listTwo* to priority value
priority value = priority +1
Transfer contents of *current_listOne* to *past_listOne* and *current_listTwo* to *past_listTwo*
Remove all entries from *current_listOne* and *current_listTwo*
If broke out of while loop using argument [1]
set priority variable to *priority_in_listTwo*
If broke out of while loop using argument [2]
set priority variable to priority value
set complex to *chosen_complex*
update relavant variables

Figure 5.6. Simplified flowchart of priority recalculation following the unbinding phase. Arrows show the direction of code with red paths being followed when a condition of a statement is not met, green when the condition is met, and black being followed regardless. Each agent essentially "walks" along both possible complexes (one starting from each of the domains involved in the unbinding interaction) until either [1] the same platform is identified in both complex walks, or [2] both walks finish without sharing a platform. These complexes are termed "1" and "2" in the figure, although their values in the code are derived from the layer 31 message list. The case of [1] signifies that both lists have "met up" via another connection and thus all proteins associated with the unbinding event are still part of the same complex. Here, each platform takes its priority value from the second domain involved in the unbind interaction. The case of [2] signifies that both lists have exhausted all possible elements and yet never met up. Here, both the proteins that unbound now belong to different complexes. The agent takes the priority calculated in the while loop and the complex ID associated with the list that identified the agent.

Third, both lists continue until completion without "meeting up" and the platform identifies itself within the dominant binder list. Here, the agent accepts the binding priority value from the dominant binder walk and complex ID of *new_complexOne_id* (layer 31 message).

Regardless of what outcome was reached, a message is sent containing the complex_id value in preparation for a recalculation of the size variable.

5.6.34 Layer 33: Unbinding priority update: disseminating priority values

Agent's subject	>**currentState** **nextState**>	**Function guard**	>**Message input** **Message output>**
Function: **output_unbinding_platform_variables**			
PlatformAgent	>Resolved Resolved>	Agent identified its complex in layer 32	>complex_size_count platform_variable_update>

Agents which identified their *complex_id* at the start of layer 32 may be within a complex that is a different size than at the start of the unbinding phase. Therefore, agents update their complex_size variable with how many other PlatformAgents share the same complex_id as themselves. Platforms also output priority and size information as a message.

5.6.35 Layer 34: Unbinding priority update: updating domains

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **update_unbinding_domain_priority_values**			
DomainAgent	>Resolved Resolved>	Domain with a complex size greater than 1	>platform_variable_update

All DomainAgents contained within multi-platform complexes search for the IDs of their parent platform in the layer 33 message list. If found, they update their complex ID, priority, and complex size accordingly.

5.6.36 Layer 35: Updating DomainAgent context 2: uploading data

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **binding_report_requests**			
DomainAgent	>Resolved Resolved>		upload_partners>

The context of DomainAgents must be refreshed after any possible unbinding events. This uses the same function as described in layer 23.

5.6.37 Layer 36: Updating DomainAgent context 2: reassessing dimerisation status

Agent’s subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **register_dimerisation**			
PlatformAgent	>Resolved Resolved>	Only proteins which have the possibility to form homodimers	>upload_partners

PlatformAgents must identify whether they are now homodimerised before sorting through the binding data. This uses the same function as described in layer 24.

5.6.38 Layer 37: Updating DomainAgent context 2: data processing

Agent’s subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **register_binds**			
PlatformAgent	>Resolved Resolved>		>upload_partners dimerisation_updator_list>

PlatformAgents run through the binding data messages from layer 35 using the same function as described in layer 25.

5.6.39 Layer 38: Informing the relevant domains and Las17-mediated nucleation

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **register_dimer_binds**			
DomainAgent	>Resolved Resolved>	Only proteins which have the possibility to form homodimers	>dimerisation_updator_list
		Function: **Las17_nucleation**	
PlatformAgent	>Resolved Resolved>	Platform must be Las17 (*species* = 0)	nucleation_ticket>

DomainAgents read the output messages of layer 37 to update their environmental context variables. This uses the same function as described in layer 26.

If three actin agents are found to simultaneously bind the same Las17 monomer/dimer, then a new actin FilamentAgent is generated and the appropriate variables for a newly nucleating filament are set. A *nucleation_ticket* message is outputted containing the domain index of the actin-bound tracts and ID of the new filament. The three actin agents which constitute the nucleus must occupy a PP1, PP3, and PP4 position. However, they can be bound to either the Las17 agent executing the function, the homodimerisation partner of the Las17, or a combination of the two. To prevent both Las17 agents within a dimer simultaneously triggering a nucleation event, only the platform with the highest ID number can execute the function code. If both Las17 agents in a dimer have bound actin at the same tract, then the actin which is bound to the Las17 with the highest ID number is chosen to constitute the seed.

5.6.40 Layer 39: Register nucleation: Las17 domains

Agent's subject	>currentState nextState>	Function guard	>Message input Message output>
Function: **register_nucleation_tracts**			
DomainAgent	>Resolved Resolved>	Must be a species 0 platform (Las17)	>nucleation_ticket g_to_f_ticket>

Las17 DomainAgents search the layer 38 message list. If they successfully identify themselves, the agent(s) compare their domain_index variable(s) against the tract indexes of the message. If these values match, then the tract unbinds the monomeric actin agent and changes its binding_state variable to 3, indicating the domain is bound to an actin filament. A message is also output containing information for the actin platform including the leapfrog variable, *filament_id*.

5.6.41 Layer 40: Register nucleation: actin

Agent's subject	>currentState nextState>	Function guard	>Message input Message output>
Function: **register_nucleation_dimer**			
PlatformAgent	>Resolved Resolved>		>nucleation_ticket
Function: **register_nucleation_actin**			
DomainAgent	>Resolved Resolved>	Agents must be of type 5 (actin)	>g_to_f_ticket nucleated_platform_ticket>
Function: **auto_nucleation**			
PlatformAgent	>Resolved Resolved>	ID value of 1	

It is critical to update the homodimerisation partner of a Las17 agent which has nucleated Las17. Therefore, PlatformAgents search the layer 38 message list for dimerised partners. If identified, the filament_bound variable is updated accordingly.

Actin domain agents search the layer 39 message list. If they can identify themselves, the agent(s) will update their variables accordingly to reflect their incorporation into a newly nucleated actin seed. This includes using the leapfrog *filament_id* variable to link themselves to the correct filament. A message is then sent to update their parent platforms in the subsequent layer

The PlatformAgent possessing an ID value of "1" generates a random number variable and compares this against the probability of salt-mediated actin nucleation per timestep. If the random number is lower than this probability value, then a new FilamentAgent will be generated. This reflects the process of salt-mediated nucleation in solution whereby actin monomers can directly interact to form a nucleus. *auto_nucleation* functions to reproduce this process whilst avoiding the extreme rates associated with actin dimerisation.

5.6.42 Layer 41: Register nucleation: Las17 platforms

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **update_nucleation_platforms**			
PlatformAgent	>Resolved Resolved>	Must be a species 1 platform (actin)	>nucleated_platform_ticket
Function: **update_nucleating_domain_priority_values**			
DomainAgent	>Resolved Resolved>		>nucleation_ticket nucleating_platform_update_priority>

The parent platforms of actin DomainAgents which were incorporated into an actin filament during layer 40 identify themselves using the *nucleated_platform_ticket* message list and update their variable accordingly. This includes setting the *factin* variable to "1".

Nucleation changes the platform makeup of complexes involves. Therefore, priorities and complex sizes need to be redetermined. Domain agents executing the function *update_nucleating_domain_priority_values* search for complex IDs. If located, they output their binding information. Domains with platform ID numbers matching the Las17 agent which nucleated a filament are identified as "*priority_prime*".

5.6.43 Layer 42: Nucleation priority update: determining priority

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **update_nucleating_platform_priority_values**			
PlatformAgent	>Resolved Resolved>		>nucleating_platform_update_priority nucleating_complex_id_counter>

Platform agents redetermine their priority values using code derived from the binding phase (**section 5.7.21**). This time, the list walk begins from the parent platform of the "*priority_prime*" agent identified in layer 41 and a message is output containing the *complex_id*.

5.6.44 Layer 43: Nucleation priority update: disseminating priority

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **output_nucleating_platform_priorities**			
PlatformAgent	>Resolved Resolved>	Agents must have updated their priority values in layer 40	>nucleating_complex_id_counter nucleating_final_update>

Platforms which updated their priorities in layer 42 use the message input to count the number of PlatformAgents with matching *complex_id* values to recalculate the *complex_size* variable. They then output the priority and size for their child DomainAgents.

5.6.45 Layer 44: Nucleation priority update: updating domains

Agent's subject	**>currentState** **nextState>**	**Function guard**	**>Message input** **Message output>**
Function: **updating_nucleating_domain**			
DomainAgent	>Resolved Resolved>	Agents must belong to a complex which nucleated an actin filament this timestep	>nucleating_final_update

Agents search the output messages from layer 43 and update their priority and complex size variables to those given by their parent PlatformAgents.

5.7 Code Validation

The same code validations undertaken with Model Alpha were also undertaken with Model Beta. An equilibrium dissociation constant comparable to an analytical curve derived from the input was observed (**figure 5.7a**) along with a match between model-derived polymerisation rates and those from the literature (**figure 5.7b**). These results helped validate that the model was functioning as intended. A 2D movie visualisation of Model Beta is given in the **supplementary materials** (movie_2).

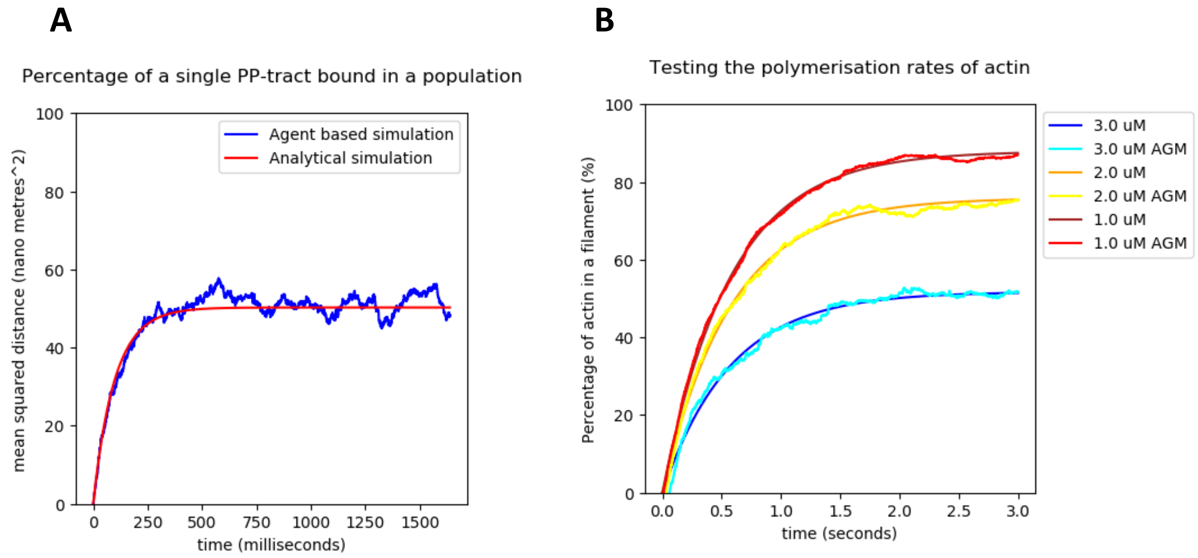


Figure 5.7. Validating model beta. **A**) The percentage of agent A (30 nM) bound to agent B (500 nM) over time with a K_d value of 500 nM. The Model Beta-derived curve was highly consistent with the analytical prediction suggesting good functionality of the model. **B**) A 0.4 µM concentration of seeds with 1.5, 3.0, and 4.5 µM concentration of G-actin were each simulated. The percentage of actin agents occupying a filamentous state was plotted against time (blue line represents Model Beta) and an analytical model (orange line) derived from literature filament polymerisation rates (Pollard, 1986). The agent-based program reproduced the experimentally realistic behaviour accurately over 1.5 seconds (time stepping of 1 µs) demonstrating functionality of the polymerisation phase.

5.8 Code derivations

The two code derivations of model Alpha were rebuilt using Model Beta. See **section 3.8** for a breakdown of these derivations.

6. Results: Model Beta

Model Alpha was only capable of simulating Las17 as a cross-filament nucleator and did not permit the movement of this protein. This was a reasonable first hypothesis due to our limited understanding of the interactions between actin and the polyproline tract. Implementing a cross-filament nucleating scheme was the computationally simplest mechanism to implement. However, the results of Model Alpha suggest that Las17 may utilise alternative and/or additional mechanisms to confer a lower rate. These two predictions included Las17 dimerisation and a linear nucleating mechanism. Testing either of these was not possible within the constraints of our previous model which necessitated a new approach. Model Beta was developed as a significantly improved ABM which followed a more generalised approach.

This was a significant undertaking due to the requirement of an almost complete rewrite of the code whilst also requiring greater complexity to overcome the hurdles imposed by a non-static Las17. Therefore, it was decided to gather additional experimental data to help further narrow the acceptable range of our model parameters. This included using Biolayer interferometry as a supporting method to MST as will be discussed later within this chapter.

6.1 Further characterising the actin nucleation scheme

Further characterising of the actin binding mechanism employed by the Las17 polyproline tracts was crucial in providing experimental evidence for a linear nucleation scheme. The novelty of Las17-mediated actin nucleation means that the existing literature was limited. However, several key studies (along with unpublished data from the Ayscough lab) helped to focus our exploration.

6.1.1 MST: Actin mutant affinities

Our pre-existing knowledge regarding the actin-binding sites of Las17 was based on three observations. First was that mutation to N-terminal arginine pairs reduced the rate of actin nucleation along with actin affinity. Second was that mutation of proline residues in the non-nucleating, C-terminal tracts reduced the rate of actin polymerisation. Thirdly, the first, third and fourth polyproline tracts in the nucleating component share sequence homology in the residues N-terminal of these prolines. This included the arginine pairs which (from PP1 to PP4) are as follows: RNNRPVPPPPP, RRGPAPPPPP, and RRGPAPPPPP.

A series of experiments was conducted using several Las17 mutation constructs. These assays were undertaken using Las17 preparations purified alongside one another whilst final concentrations were measured using the more robust methods identified in **chapter 4.1.2**. The consequence of this was that the quality of each preparation was highly consistent between the mutants – an important factor when considering the difficulties of Las17 purification. The actin binding affinities for each peptide were then determined via MST in short time succession using the same lysine-labelled actin preparation.

Plasmid identification	Peptide mutations
P566	300-422; No mutations
P1189	300-422; RR(349,350) to alanine
P1190	300-422; RR(382,383) to alanine
P1191	300-422; RR(349,350) and RR(382,383) to alanine
P1337	300-422; RR(319,322), RR(349,350), and RR(382,383) to alanine

Table 6.1: A list of Las17 300-422 peptides. The plasmid identifications given in the left column describe the peptides in the right column. Mutations of different arginine pairs are indicated with the mutated residue numbers delineated using the bracketed values.

All assays undertaken used a fixed concentration of 50 nM actin and a range of 16 Las17 concentrations which descended across the capillaries using a series of stepwise, 2-fold titrations. To increase the readability of this section, the different Las17 constructs analysed will be referenced according to their plasmid number used within the Ayscough lab. The different plasmid numbers are given in the table above (**table 6.1**)

The non-mutagenized 300-422 nucleating region of Las17 was expressed using plasmid P566 to a peak concentration of 12 µM. This gave a K_d value of 0.0904 ± 0.042 µM in G-buffer (**figure 6.1a**). Plasmid P1189 was then purified to 32 µM and subsequently analysed to determine the effect of removing the second arginine pair. This gave a reduced affinity of 0.191 ± 0.0074 µM in G-buffer (**figure 6.1b**). Plasmid P1190, a construct with the third arginine pair mutagenized, was purified to 17 µM and also gave a reduced affinity with a value of 0.119 ± 0.0062 µM in G-buffer (**figure 6.1c**). A construct containing both the P1189 and P1190 mutations (P1191) was purified to 37 µM to investigate how removal of both strong actin interaction sites would affect binding. These gave a further reduced affinity of 0.546 ± 0.047 µM in G-buffer (**figure 6.1d**). Finally, a construct with all arginine pairs mutagenized (P1337) was purified to 23 µM to identify how much the polyproline tracts contribute to the binding affinity. This gave a K_d value of 7.41 ± 0.82 µM in G-buffer, showing a marked reduction in affinity (**figure 6.1e**).

No issues were identified within any of the MST assays. This is in contrast to the MST results of **chapter 4** and thus highlights the difficulties associated with experimental characterisation of the Las17-SH3 binding system. The removal of arginine pairs progressively reduced the observed affinity whilst the construct containing all three pairs mutagenized gave markedly reduced binding (~7.4 uM) in comparison to the nanomolar affinities observed with one or more arginine pairs present. This both confirms our hypothesis that the 300-422 region binds actin via three distinct and novel binding motifs, whilst also suggesting that the polyproline tract contributes much less to the overall motif affinity than the arginine pairs. However, this result does not discount the potentially significant structural role that proline may play within the motif. The obtained affinities were also predictably comparable to one another which allowed us to continue our analysis to elucidate individual tract affinities.

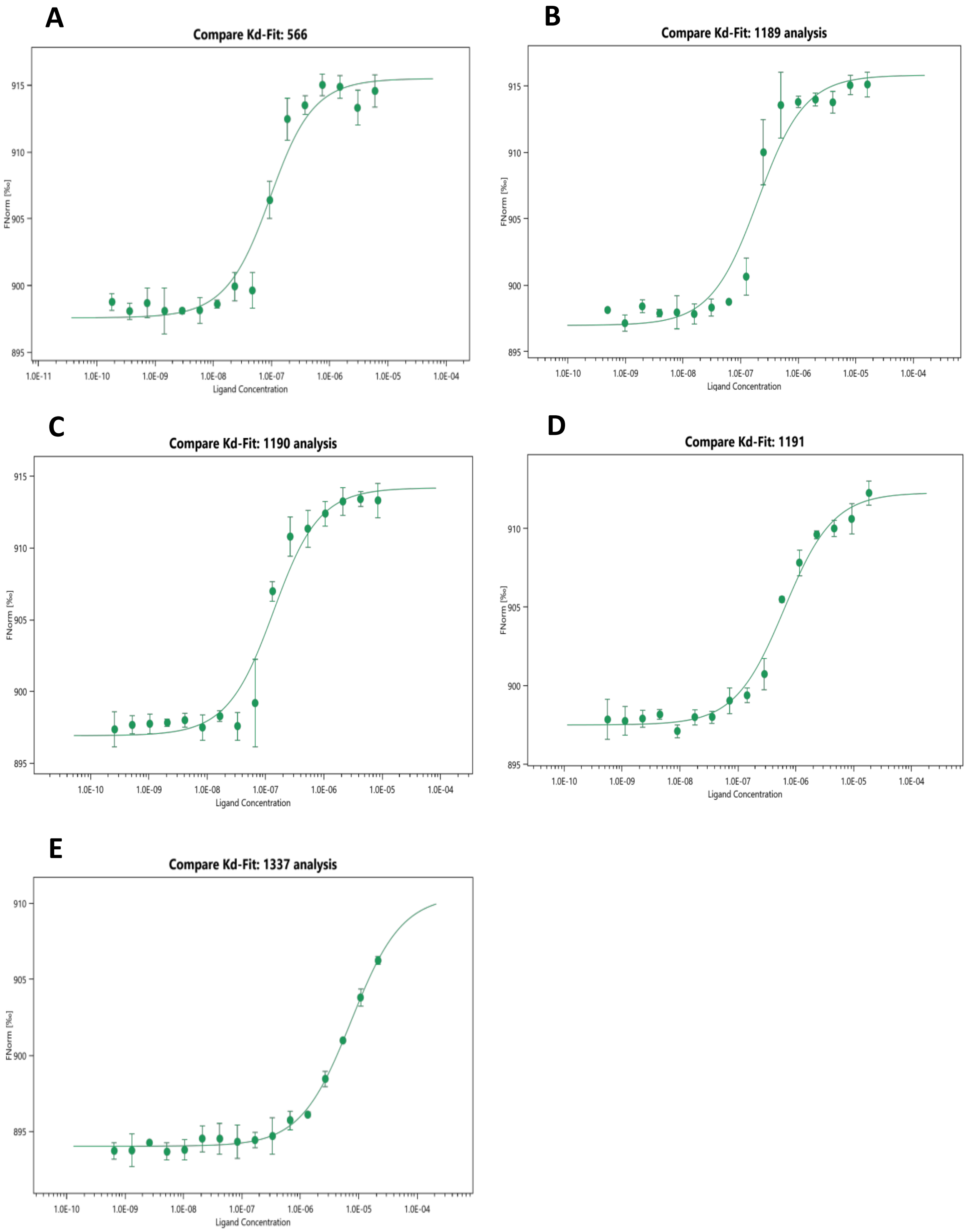


Figure 6.1: Las17 mutant peptide MST results. FNorm in all graphs is calculated using the ratio Fhot/Fcold where Fhot is the fluorescence after heating and Fcold is the fluorescence before heating. All binding affinities were fitted from three replicates (n=3) and used 50 nM labelled actin. **A**) A binding curve for 566 (WT 300-422). **B**) A binding curve for 1189 (300-422; RR349,350AA). **C**) A binding curve for 1190 (300-422; RR382,383AA). **D**) A binding curve for 1191 (300-422; RR349,350AA and RR382,383AA). **E**) A binding curve for 1337 (300-422; RR319,322AA, RR349,350AA and RR382,383AA).

6.1.2 Characterising the actin nucleating binding sites of Las17

A simple ODE model was built using a set of simple rate equations (**equations 6.1-6.2**). These rates were calculated for several seconds using a 1 µs timestep until an equilibrium was reached. All equations used a single actin concentration variable and three tract concentration variables (one for each binding site). This allowed each polyproline tract to be represented by a unique K_{d} value.

$$gain\ in\ complex = (k_{on} * [ligandOne] * [ligandTwo]) * timestep \tag{6.1}$$

$$loss\ of\ complex = (k_{off} * [complex]) * timestep \tag{6.2}$$

Comparison of the percentage of bound ligand predicted by the ODE model to the experimentally predicted degree of binding was undertaken using **equation 6.3** where P_t is the concentration of actin while L_t is the concentration of Las17 (**figure 6.2**). Equation 6.3 was taken from (Jarmoskaite *et al.*, 2020). Prediction errors were obtained by repeating the analysis described within this chapter to reproduce the error range of the experimental K_{d} error.

$$\begin{aligned} &Percentage\ P_t\ bound \\ &= 100 * \frac{\left((P_t + L_t + K_d) - \sqrt{(P_t + L_t + K_d)^2 - (4 * P_t * L_t)}\,\right)}{2 * P_t} \end{aligned} \tag{6.3}$$

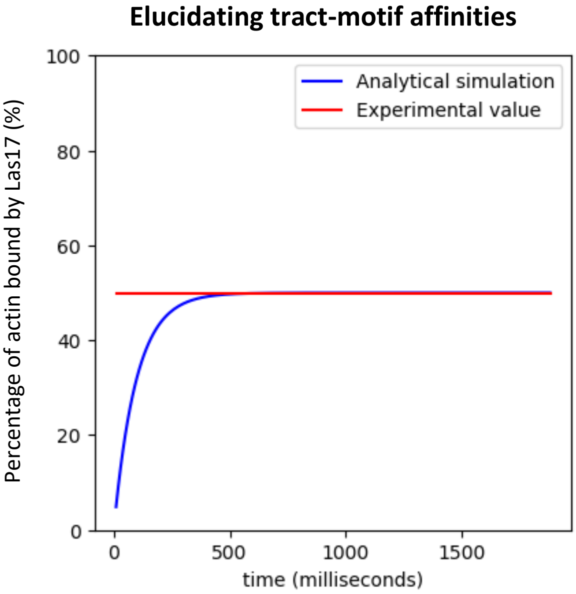


Figure 6.2: ODE fitting graph. Simulated binding curves were fitted against experimental data to elucidate individual tract affinities. To allow for this, the ODE simulated the accumulation of actin into complexes with three ligands available for complexing (PP1, PP3 and PP4) and all sharing the same pool of actin. The percentage of actin simulated by the ODE model is plotted here in blue whilst the experimentally predicted degree of binding (calculated using **equation 6.3**) is shown in red. The simulated binding affinities agreed with experimental data when the binding equilibrium matched (as shown above).

The first step was to determine the K_d of each binding site that can be attributed to the combined, direct effect of all residues excluding the arginines (i.e., P1337). This gave a predicted value of 22.2 ± 2.4 µM for each tract. Mutation to the C-terminal prolines has been observed to reduce Las17 polymerisation suggesting that the proline residues may contribute the most to this residual interaction. Alternatively, the tracts may play a predominantly structural role for the actin binding motif as frequently observed in other polyproline binding sites. Either way, the obtained affinity was very weak indicating that the arginine pairs contribute heavily to the affinity of this novel motif.

The K_d for PP1 was increased until an affinity matching the P1191 construct was reproduced whilst maintaining a tract affinity of 22.2 ± 2.4 µM for PP3 and PP4. This gave an affinity of 0.575 ± 0.045 µM. Following this, the affinity of PP3 was increased until a K_d matching the

P1190 construct was reproduced whilst maintaining a tract affinity of 0.575 ± 0.045 µM for PP1 and 22.2 ± 2.4 µM for PP4. An affinity of 0.177 ± 0.024 µM was determined for the motif. This process was repeated for PP4 to reproduce P1189 whilst maintaining an affinity of 0.575 ± 0.045 µM for PP1 and 22.2 ± 2.4 µM for PP3. Here, an affinity of 0.328 ± 0.04 µM was determined for the N-terminal nucleating tract.

Taken together, the above analysis concludes that PP3 binds actin with a roughly 2-fold higher affinity than PP1 whilst PP4 is the tightest binding, with a further 2-fold higher affinity than PP3 (i.e., quadruple that of the first tract). To validate this ODE-based analysis, these figures were implemented back into the model and compared against the MST results of P566. The affinity predicted was 0.0877 ± 0.013 µM which fell comfortably within the error range of our experimental measurement (0.0904 ± 0.042 µM). It must be noted that the errors attributed to calculating the protein concentration were not accounted for. Combining this caveat with the above shows good agreement with the experimental data.

The affinity predictions given were derived using G-buffer data due to the propensity of actin to polymerise in the presence of higher salt buffers. On the other hand, Model Beta primarily relies on PBS data as will be explained later this chapter. Conducting affinity assays in G-buffer usually gives affinity values between 10 and 20 times lower than those obtained in PBS. Fortunately, the binding strength of each tract relative to each other would likely be preserved given the almost identical sequence similarity of the binding motifs. This is the first time an affinity characterisation of the individual Las17 actin binding sites has been concluded which helped to provide a range for the K_d sensitivity analysis of Model Beta (**table 6.2**).

Binding region	G-buffer affinity (K_d)	Error estimation	Estimated PBS affinity range (K_d)
Tracts without N-terminal arginines	22.2 µM	10.9%	222 – 444 µM
PP1	0.575 µM	8.62%	5.75 – 11.5 µM
PP3	0.177 µM	13.5%	1.77 – 3.54 µM
PP4	0.328 µM	12.3%	3.28 – 6.56 µM

Table 6.2: A list of Las17 300-422 peptides. The plasmid identifications given in the left column describe the peptides in the right column. Mutations of different arginine pairs are indicated with the mutated residue numbers delineated using the bracketed values.

6.1.3 HADDOCK modelling of the actin-Las17 interaction

Almost no structural data exists for the interaction between Las17's polyproline region and actin. However, modelling peptide docking may prove insightful following the experimental verification of our three Las17 binding site model. High Ambiguity Driven protein-protein DOCKing (HADDOCK) is a computational method which aims to predict protein structures in the absence of complete structural data (Dominguez, Boelens and Bonvin, 2003; van Zundert *et al.*, 2016). HADDOCK is able to direct the sampled interaction space by allowing the user to specify known interaction residues that are suspected of playing a role in the binding – termed ambiguous interaction restraints (AIRs) within the program. This technique fits perfectly well with our current experimental data as, whilst we understand that binding motif arginines play a crucial role in the interaction, we also can also infer the important residues of actin using a published yeast-two-hybrid assay (Y2H) (Urbanek *et al.*, 2013; E Allwood, personal communication).

Figure 6.3a shows our predicted binding interactions given the residues indicated by the yeast two-hybrid paper along with other notable residues identified using biochemical knowledge. The AIRs given to HADDOCK were a combination of actin residues E316, D222, E224, and E226, along with the double arginines and prolines from Las17 PP3. It was predicted that two phenylalanines would likely interact with the polyproline tract via π-stacking as other residues identified in the mutant screen appeared to stabilise this region. Glu 276 was never assessed as part of the mutant screen. However, it was the best positioned residue local to Glu 316 that may allow for the coordinated binding of both arginines of the Las17 polyproline binding motif. Modelling was undertaken with the third actin-binding site in the 300-422 peptide as mutation to its arginines confers an obvious deleterious effect when nucleating (TG**RRGPAPPPPP**RA with bold residues constituting the core motif). The resultant scoring was determined from a variety of factors such as the energetics of the system and, like all docking scoring functions, has no correlation with affinity (Dominguez, Boelens and Bonvin, 2003; Kastritis and Bonvin, 2010).

Unfortunately, the highest scored calculated structures rarely involved consistent interaction with both of the arginines as one was often orientated away from the interaction interface regardless of which relevant residues were selected as AIRs (**figure 6.3b**). The AIRs given whilst modelling were frequently omitted from the interaction. Furthermore, the top scoring structures did not coalesce around a preferred binding scheme (see the top scoring structure **figure 6.3b** compared to the second top scoring structure **figure 6.3c**). This reduced our confidence in any one of the predictions.

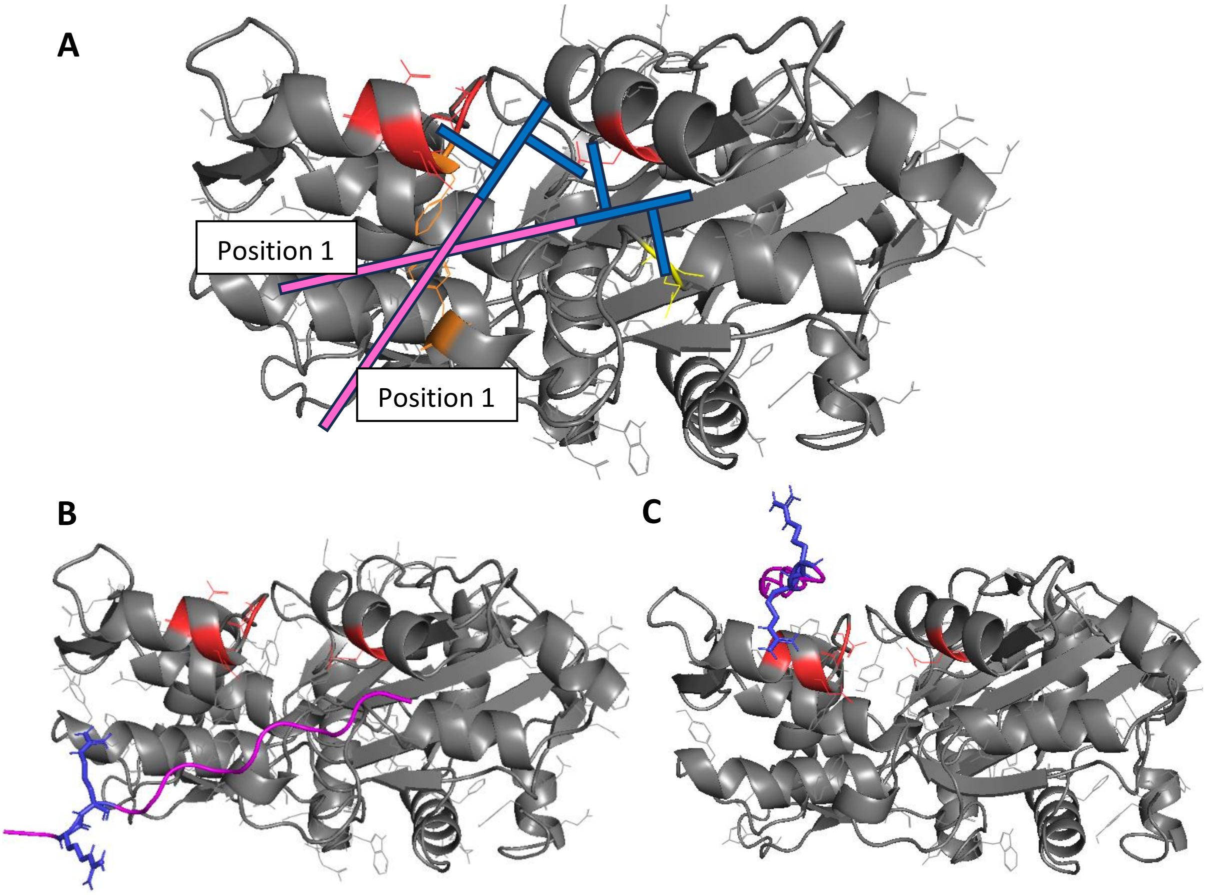


Figure 6.3: HADDOCK peptide docking of Las17 to actin. All figures were plotted using PyMOL v4.6 with G-actin shown in grey (PDB: 3HBT). Actin residues that are either negatively charged or aromatic are shown as wires as these are likely the groups that interact with Las17 tracts. **A**) Initial predictions for the binding of Las17 PP3 (two positions shown as simple boxes with the polyproline region in magenta and arginine in blue). Five negatively charged residues indicated by the Y2H are shown in red. Glu 276 is well positioned to bind one of the Las17 arginines and is thus shown in yellow. Phe 223 and Phe 266 are well positioned to bind the polyproline tract and are shown in orange. The first **B**) and second **C**) highest scoring structures are shown with clear AIR (here shown in red) violations and one of their arginine residues orientated away from actin such that they do not form any interaction with the monomer.

6.1.4 Flexible peptide docking of the actin-Las17 interaction

One interpretation of HADDOCK results is that the actin mutations which conferred a negative effect to actin binding may have acted allosterically. These mutagenized residues may play a structural, rather than a direct, role in the interaction. FlexPepDock and HPEPDOCK 2.0 are general flexible peptide docking softwares which do not require experimental data. As a consequence, the interaction samples the entirety of both protein surfaces and is thus independent of the yeast two-hybrid data. The 10 structural predictions with the highest scores predicted for of these methods each coalesced around the barbed end groove (**figure 6.4a** and **6.4b**).

The yeast two-hybrid study previously used to identify HADDOCK AIRs (**section 6.1.3**) was again analysed to identify other potential interaction residues (Urbanek *et al.*, 2013). All actin residue mutations which resulted in a clear relative reduction in colonies were taken as potential binding residues. Mapping these yeast two-hybrid mutations onto a structure of yeast G-actin reveals three mutation clusters (mutations shown in magenta and clusters circled in yellow numbered 1-3 in **figure 6.4c**). Our previous hypothesis was that the interaction site was likely within cluster 1 due to the associated mutations (act1-104 and act-111) abolishing all growth in the yeast two-hybrid analysis. However, both this cluster and cluster 2 flank a helix rich sequence that constitutes a large percentage of the barbed end groove surface (shown in orange). Cluster 1 contains residues which stabilise the top of this region. Meanwhile, Lysine 359 from cluster 2 stabilises the bottom via a hydrophobic interaction with a phenylalanine (shown in salmon) capped with an ionic bond to glutamate (shown in red). This provides a possible explanation as to why these clusters are associated with reduced Las17 binding as both would theoretically disrupt the barbed groove interface.

Mapping the highest scoring HEPEPDOCK structure (shown in green with the double arginines given in blue) onto the mutation-highlighted actin structure reveals a heavy interaction with the helical rich region (**figure 6.4d**). In contrast to the HADDOCK modelling presented in the previous section, all structures predicted from the flexible peptide modelling involve heavy interaction with the Las17-motif double arginines. In the prediction shown, the arginines “slot” into a hydrophobic groove created by phenylalanine 352 and methionine 355 (both shown in salmon) from within the helical region. Interestingly, the N-terminal of the Las17 arginines is also well positioned to form an ionic bond with glutamate 361 (from cluster 2) which was shown to negatively impact binding within the yeast two hybrid. This residue is also close to lysine 359 which was predicted to stabilise the base of the helical region in the previous paragraph.

Of particular note, this mode of binding parallels that of WH2 domains as both bind along the same groove (see **figure 6.4e** in comparison to **figure 6.4d**). This may indicate why the WH2 domains of Spire and the actin-binding motifs of Las17's polyproline region are separated by similar distances (**table 6.3**). Mapping the fourth WH2 domain of *Drosophila* Spire (shown in gold) to the mutation-highlighted actin also reveals a lysine residue (shown in blue) similarly positioned to the Las17 arginine pairs (although orientated in the opposite direction).

Figure 6.4: Flexible peptide docking of Las17 to actin

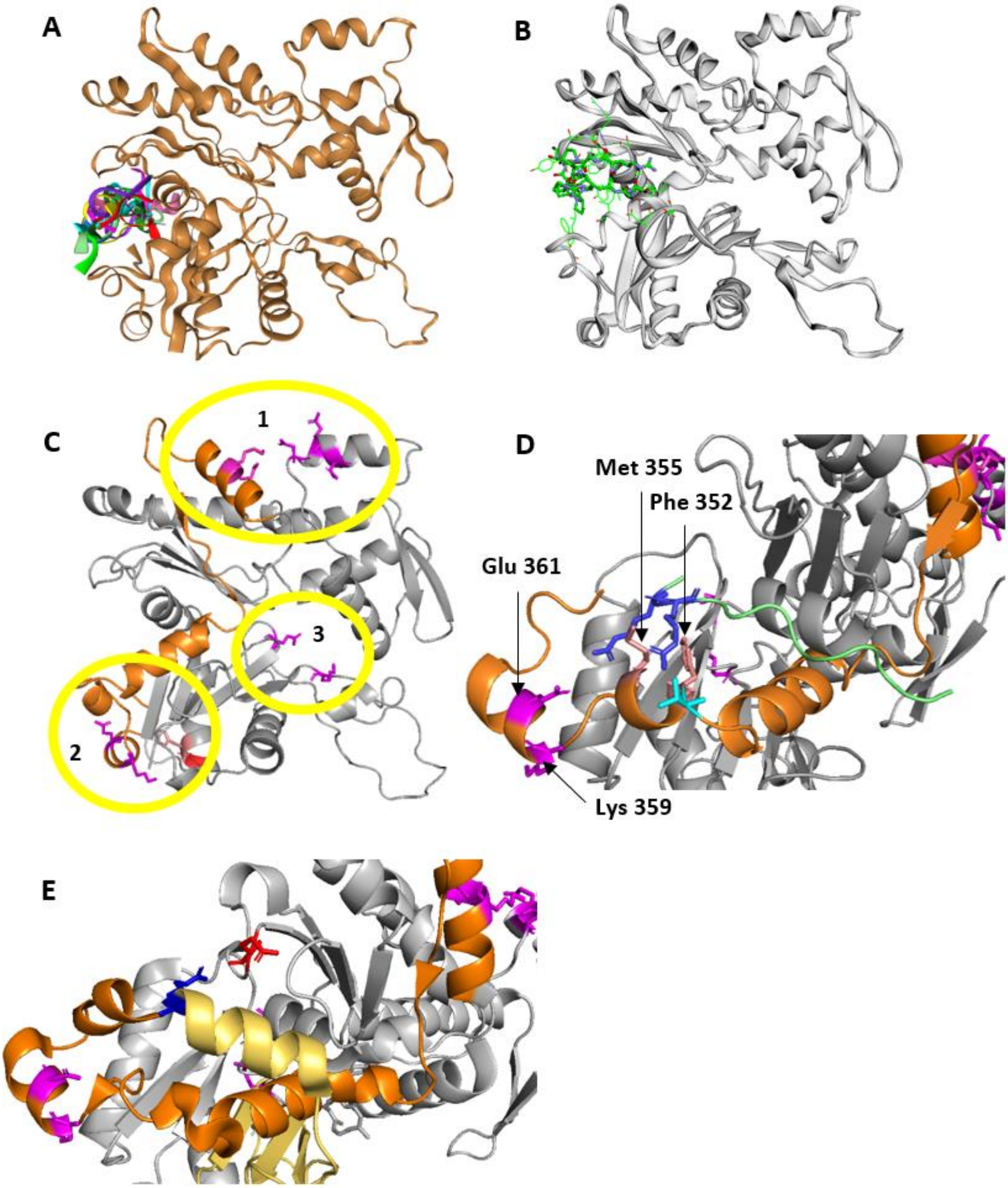


Figure 6.4: Flexible peptide docking of Las17 to actin. Las17 PP3 was docked to yeast actin (PDB: 3HBT) in an attempt to elucidate information for how the pair may bind. **A**) Docking PP3 to actin (shown in brown) using FlexPepDock and **B**) HPEPDOCK 2.0 (here, actin shown in grey) revealed a coalescence around the barbed end groove. **C**) Mapping the locations of residues deleterious to Las17 polyproline binding (magenta) revealed three clusters as ringed in yellow. Clusters 1 and 2 flank a helical rich region (orange) while a phenylalanine and glutamate predicted to interact with cluster 2 are shown in salmon and red respectively. **D**) A deeper analysis of the top scoring HPEPDOCK 2.0 structure reveals heavy interaction between Las17 (green with arginines shown in blue) and the helical rich region. The double arginines of this interaction are stabilised by hydrophobic residues (Phenylalanine 352 and Methionine 355 – both shown in salmon) and an ionic interaction between the N-terminal PP3 lysine (shown in blue) and Glutamate 361 (shown in red). **E**) This interaction strongly resembles the binding between the fourth WH2 domain of *Drosophila melanogaster* Spire (gold) and actin (PDB: 3MN7). Here, an interaction is highlighted between a WH2 lysine (blue) and glutamate from actin (red). All figures were generated within PyMOL v4.6.

Tandem nucleator	**Length between centre of binding sites 1 and 2 (residues)**	**Length between centre of binding sites 2 and 3 (residues)**	**Length between centre of binding sites 3 and 4 (residues)**
Las17 300-422 (Budding Yeast)	29.5 [(1)]	33.5 [(1)]	
Spire (Human, Spir1)	40 [(1)]	29.5 [(1)]	31.5 [(1)]
JMY (Human)	28 [(2)]	31 [(2)]	
Cobl (Human, cordon-bleu)	40 [(1)]	88 [(1)]	

Table 6.3: Residue spacing between the centre of actin-binding domains in tandem nucleators. The number of residues separating the centre of each actin-binding binding domain with the centre of the actin actin-binding motif indicated ([1]UniProt Consortium, 2023; [2]Zuchero *et al.*, 2009). This reveals that the nucleating tracts of Las17 are separated by roughly the same number of residues as the nucleotide WH2 domains of Spire and JMY. Also revealed is that the cross-filament nucleator Cobl employs a much larger separation (88 residues) than Las17 (29.5 and 33.5 residues) supporting the hypothesis that Las17 nucleates by incorporating actin monomers longitudinally rather than cross filament.

6.2 Gauging the dimerisation affinity of Las17

One of the mechanisms hypothesised from **chapter 4** was that Las17 may need to self-oligomerise to bind actin. To investigate this, a fixed concentration of SNAP-labelled Las17(300-633)-His I555D was assayed against a range of 16 Las17(300-422) concentrations (463 µM to 0.007 µM using a stepwise, 2-fold titration). The titrated peptide was GST purified with the tag removed.

A Las17-Las17 self-interaction was identified with a weak binding affinity of 38.4 µM over a single assay (n=1) (**figure 6.5a**). Only a single MST assay was performed due to the extremely large concentration of Las17 required to cover the binding curve. However, the result still indicates some degree of weak self-interaction. The binding assay was repeated following an SD-test of the samples. An SD-test (SDS denaturation test) is an assay which aims to assess the specificity of interactions. This involves mixing the MST solutions with SDS detergent before incubating at 95^{o}C for 5 minutes. An MST assay is performed using the SDS-denatured protein. Because SD-tests denature protein structure, a loss of binding signal would indicate that the interaction observed without denaturation was a specific binding event. On the other hand, nonspecific interactions may be indicated if the observed binding persists. No interaction was identified following the SD-test between Las17(300-633)-His I555D and Las17(300-422), demonstrating that the weakly observed self-interaction is a specific protein-protein interaction rather than a non-specific effect such as molecular crowding (**figure 6.5b**).

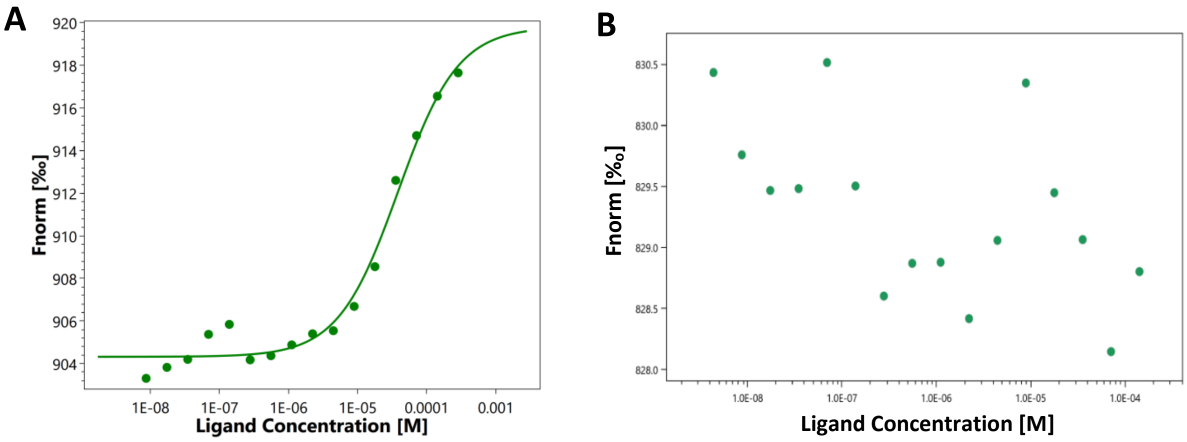


Figure 6.5: MST binding curve of Las17 self-interaction. FNorm in all graphs is calculated using the ratio Fhot/Fcold where Fhot is the fluorescence after heating and Fcold is the fluorescence before heating. **A**) The binding fit was made using a single repeat between 25 nM SNAP-labelled Las17 (300-633)-His I555D and unlabelled GST-cleaved Las17 (300-422). **B**) The assay was, then repeated following an SD-test to disrupt protein-protein interactions.

6.3 Biolayer Interferometry

Considering the labelling difficulties faced when using MST, it was decided that a second, label-free, approach may yield greater insight into the binding events. Biolayer interferometry (BLI) was chosen due to its ease of use, low sample (often nanomole) requirement and lack of a labelling step.

BLI functions by recording changes to light interference patterns that occur when an “analyte” binds to, and then dissociates from, a chosen “bait” ligand. Within BLI assays, the end of a small, disposable, tip coated in a binding substrate is used to reflect a beam of light. Binding of proteins to the probe surface increases its optical thickness which results in a wavelength shift. The degree of shift can then be compared to the reflection pattern in the absence of substrate binding.

The substrate-binding probes are moved between wells on an assay plate, and each well defines a distinct assay step. The following 6 steps are undertaken to observe a complete binding interaction: (**I**) initial baseline, (**II**) loading, (**III**) first loaded baseline, (**IV**) second loaded baseline, (**V**) association, and (**VI**) dissociation (adapted from Sultana and Lee, 2015) (**figure 6.6**). A titrated concentration of analyte is used to produce several binding profiles at different concentration points, thus providing a greater number of data to be drawn from during analysis. For this reason, a reasonably accurate K_d can be obtained from a single BLI experiment (although they are often performed in duplicate or triplicate for additional confidence) (Chen *et al.*, 2015; Weikai *et al.*, 2020).

First (step **I**), reflection through probes is baselined in the assay buffer before being moved into a well containing the bait ligand (step **II**) where binding to the probes induces wavelength shift. Two baseline steps in fresh buffer wells (steps **III** and **IV**) follow. These steps can also be used to assess the rate of ligand loss from the probe's binding surface through the rate of signal decay. Oversaturated probes often experience a greater rate of loss, which can reduce the reliability of results.

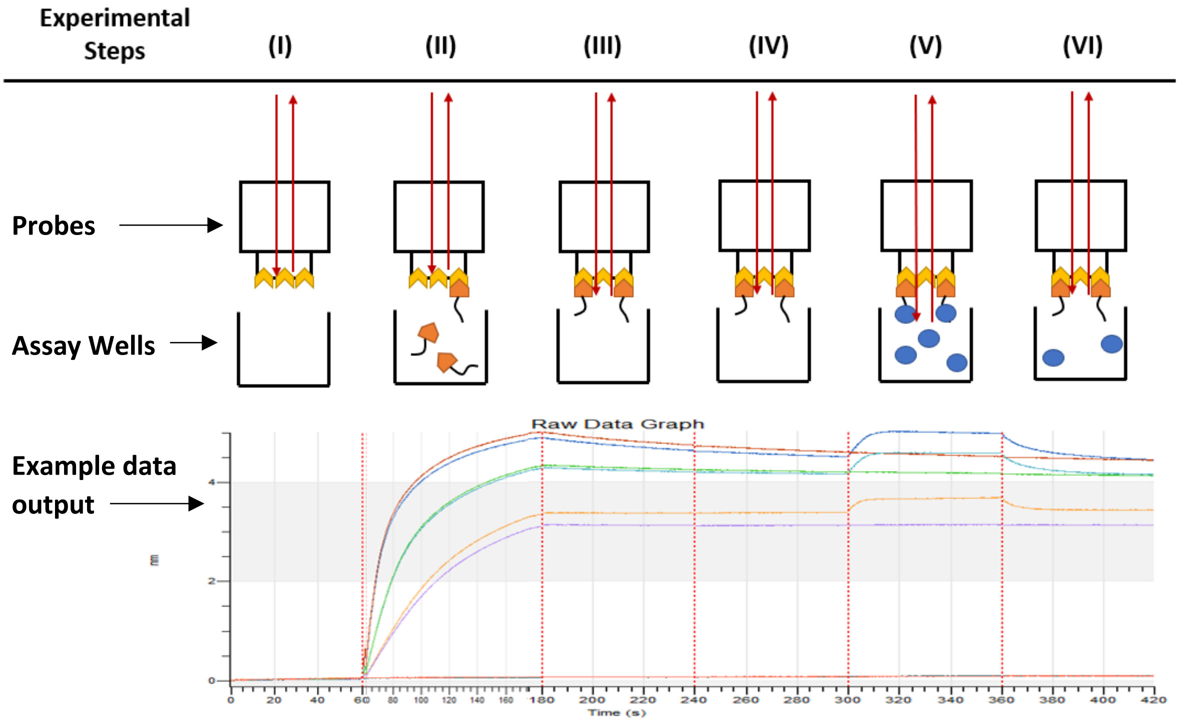


Figure 6.6: An illustration of the BLI experimental protocol. 6 diagrammatic views of the assay probes are given to illustrate the experimental steps of BLI. The red lines show the light path, yellow pockets represent binding sites on the probes (e.g., Ni-NTA or GST-antibodies), the orange shapes represent the binding ligand with orange blocks showing the probe-binding motif and the black showing the rest of the protein, and the blue spheres represent the analyte. An example data output is shown below with changes to the light interference being shown on the Y axis and time on the X axis. The graph is broken into the 6 experimental steps using dotted red lines.

The probes are then moved into wells containing the analyte of interest (step **V**). As the analyte binds to the ligand, the wavelength of the reflected light further changes causing an association curve until a binding equilibrium is reached. Finally, the probes are moved into wells containing assay buffer to begin the dissociation phase (**VI**). The reduction in analyte concentration disrupts the binding equilibrium and, consequently, almost all ligand-analyte interactions are unbinding events.

Dissociation causes the reflected wavelengths to return towards their original values producing a measurable curve. Both an association (k_{on}) and dissociation (k_{off}) rate constant can be measured from the curves produced in steps V and VI and these can be converted into an equilibrium dissociation constant (K_d) through **equation 6.4**:

$$K_d = \frac{k_{off}}{k_{on}} \tag{6.4}$$

6.3.1 Loading control assays

It is important to find the optimum loading concentration for the probe-binding ligand prior to undertaking any experiments. This optimum can be defined as a value high enough to give a strong amplitude of signal during the binding phase whilst not being so high as to oversaturate the binding probes. To find this value, an assay was performed for each ligand preparation that consisted of undertaking all 6 steps (**section 3.2**) at a variety of loading concentrations and fixed analyte concentration. An example of this assay is shown below (**figure 6.7**) wherein a loading concentration of 6.25 µg/mL was chosen.

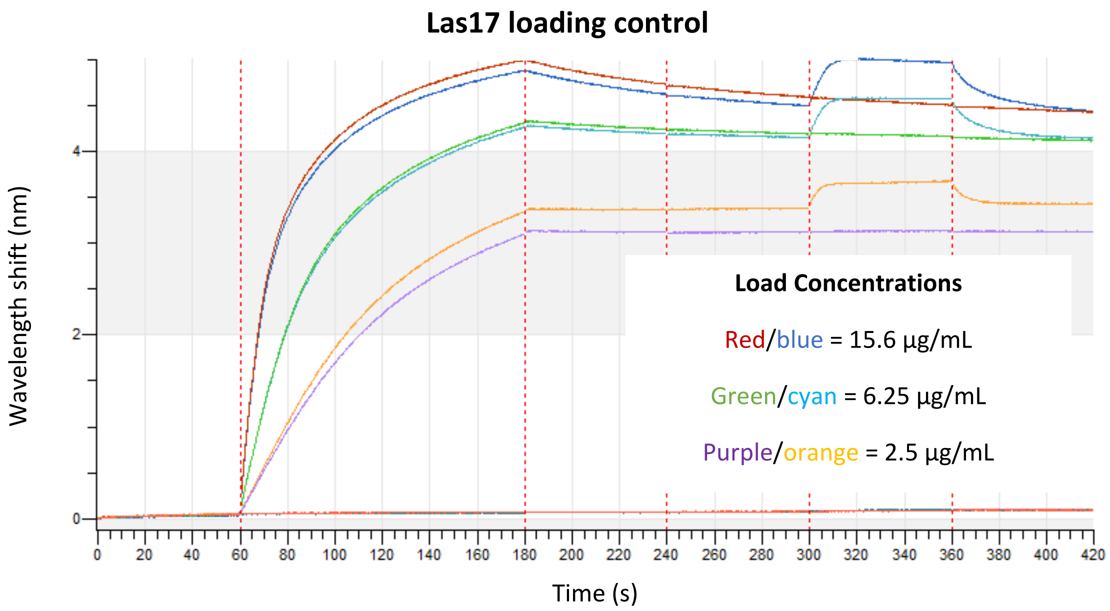


Figure 6.7: An example loading control assay. Three different loading concentrations of Las17(300-422)-His were used to bind Ni-NTA probes and a Sla1-SH3#1-2 analyte of fixed concentration. The probes at 6.25 µg/mL loading concentration gave the best results. The reasoning behind this was a good amplitude response for SH3-binding and not as much probe saturation as the 15.6 µg/mL probes. Oversaturation is indicated by a diminishing return on the signal gain along with a high rate of ligand loss from the probe.

6.3.2 BLI: Sla1

Sla1 can interact with Las17 via three SH3 domains as explained in **chapter 1** (Dionne *et al.*, 2021). Affinity assays for each individually expressed domain, along with the Sla1-SH3#1-2, and Sla1-SH3#1-3 tandem constructs were conducted.

6.3.2.1 BLI: Sla1-SH3#1

In continuation of the MST work and considering the importance of the Las17-Sla1 interaction for the modelling (explained in **section 3**), a binding affinity with the first SH3 domain of Sla1 (Sla1-SH3#1) was the first interaction assayed.

An affinity for the first SH3 domain of Sla1 and Las17(300-422)-His was obtained across two different sets of protein preparations. The reasoning for this was to provide validation of the protein quality during the buffer screening steps – the first to be done while using BLI as a method. Observing similar binding affinities between the different protein preparations would prove a degree of reproducibility and thus greater confidence in the results.

All replicates (four in total with three using separate protein preparations to the other replicate) using a fixed concentration of probe bound Las17 and a range of 4 Sla1 concentrations (1000 nM, 400 nM, 160 nM, and 64 nM) generated a K_d of 4.58 ± 0.62 µM, k_{on} of $4\text{x}10^5$ M^{-1} s^{-1} and k_{off} of 1.83 s^{-1} (n=3) for one preparation and a K_d of 10.4 ± 1.3 µM, k_{on} of $5.17\text{x}10^5$ M^{-1} s^{-1} and k_{off} of 5.36 s^{-1} for the other (n=1) (**figure 6.8a** and **figure 6.8b** respectively). The affinities were therefore consistent between preparations with a roughly 2-fold difference.

Probes analysed with a higher concentration of Sla1 yield considerable amounts of non-specific interactions while lower concentrations didn't produce a binding curve of sufficient amplitude and so were excluded from analysis. This prevented measuring of the rate through GraphPad Prism (our method of choice). Therefore, rate fitting was carried out using the data processing steps as detailed in **method 2.6.6**.

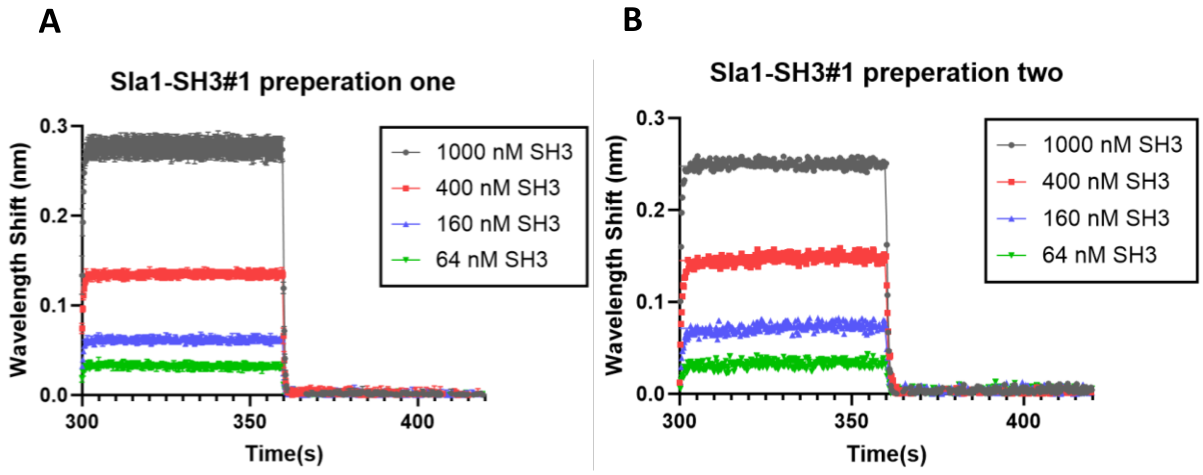


Figure 6.8: Singly expressed Sla1-SH3#1 BLI binding curves set 1. Binding of Sla1-SH3#1 to the 300-422 fragment of Las17 (7.5 µg/mL) across 4 Sla1 concentrations (1000-64 nM). Ni-NTA probes pre-incubated with Las17-His were placed in Sla1-containing buffer at t = 300 and moved into a buffer-only well at t = 360. **A**) and **B**) are identical assays using different protein preparations. Curve fitting was undertaken using the same data processing steps as detailed in **method 2.6.6** and plotted using in GraphPad Prism 10.0.2

While this affinity is ~20 times weaker than the 0.25 µM K_d determined in MST, the BLI assays were all carried out in PBS buffer which is of a much higher ionic strength than G-buffer. A K_d change over 100x can be observed when changing buffer and this effect would be particularly pronounced here due to G-buffer containing almost no salt (Welsh *et al.*, 2016).

The Sla1-SH3#1 construct was previously demonstrated to be in a folded state via 1D-NMR as previously described in **section 4.2.1.5**. However, protein context can often play a large role in SH3 substrate binding (Dionne *et al.*, 2021). To further validate our results for the first SH3 domain of Sla1, a W108A mutant was made using site directed mutagenesis from the Sla1-SH3#1-2 construct. W108A removes a critical tryptophan used in the polyproline binding loop of Sla1-SH3#2 and has been shown to disable SH3 activity (Tonikian *et al.*, 2009). Therefore, this would leave SH3#1 as the only active SH3 domain in the Sla1-SH3#1-2 construct. A similar binding affinity would demonstrate that multi-domain expression is not required for correct folding of the first Sla1 SH3 domain.

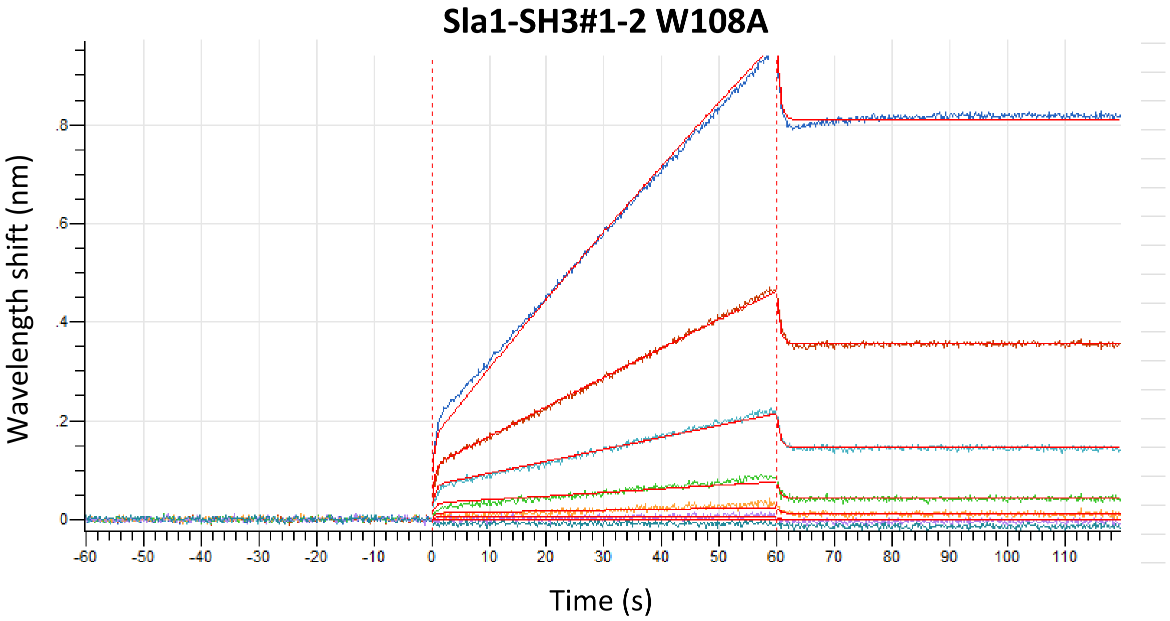


Figure 6.9: Tandem expressed Sla1-SH3#1-2 with SH3#2 inactivated (W108A) BLI binding curves. Binding of Sla1-SH3#1-2 W108A to the 300-422 fragment of Las17 (7.5 µg/mL) across 4 Sla1 concentrations (64-1000 nM). Ni-NTA probes pre-incubated with Las17-His were placed in Sla1-containing buffer at t = 0 and moved into a buffer-only well at t = 60. Curve fitting was undertaken within the Blitz analysis software (Data Analysis HT v12.0.2.59).

A new set of protein preparations were purified including the newly constructed Sla1-W108A mutant and fresh Las17(300-422)-His. An identical experimental design to the single expressed domains was used. This assay was performed in triplicate and yielded an equilibrium dissociation constant of 7.42 ± 0.9 µM, an association rate constant of 2.45x10^5 M^{-1} s^{-1} and dissociation rate constant of 1.82 s^{-1} (**figure 6.9**). This affinity lies in between the two affinities determined for the individually expressed Sla1-SH3#1 (10.4 µM and 4.56 µM) and demonstrates the reliability of using the singly expressed Sla1-SH3#1 constructs whilst also contributing to the pool of interaction data that was considered for computational modelling.

Taken together, these three sets of assays suggest that the first SH3-domain of Sla1 binds to Las17 relatively weakly with an affinity of around 7.5 µM in PBS. Considering the close spatial association of SH3#1 and SH3#2, this value seems more reasonable than the 0.25 µM value determined in MST using G-buffer. This is because a tight binding affinity must be balanced by an exceptionally weak SH3#2 affinity to reproduce the observed K_d of the SH3#1-2 tandem.

6.3.2.2 BLI: Sla1-SH3#2

Singly expressed Sla1-SH3#2 gave no binding data as no interaction between SH3#2 and Las17 (300-422) was observed in BLI. The creation of a Sla-SH3#1-2 tandem mutant that knocked out activity of the first SH3#1 domain (W41A which removes activity of SH3#1's binding loop) was attempted. Assuming that the neighbouring SH3 domain (Sla1-SH3#1) is required for correct Sla1-SH3#2 folding, this construct would allow the affinity of the second SH3 domain to be determined. Unfortunately, PCR of this mutant failed to yield any colonies following transformation into *E. coli* using both BL21 and Rosetta strains. Mutagenesis was attempted three times with varying polymerisation durations during each attempt. Transformation was subsequently undertaken three times for each PCR product and all successful colonies were sequenced by dnaseq.co.uk. Unfortunately, no colony could be identified with a successful W41A mutation. In lieu of experimental BLI data, inferring a possible affinity of the second SH3 domain using Agent-Based Modelling was chosen as an alternative method (described later in **section 6.3.4**).

6.3.2.3 BLI: Sla1-SH3#3

The next affinity chosen for experimental analysis was the third domain of Sla1. Obtaining the affinities for all three of Sla1's SH3 domains would allow for a more holistic view when describing the Sla1-Las17 interaction.

A triplicate of assays was performed that used a fixed concentration of probe-bound Las17 and a range of 4 Sla1 concentrations (1000 nM, 400 nM, 160 nM, and 64 nM). These assays gave an equilibrium dissociation constant of 16.71 µM, an association rate constant of 9.55×10^{4} M^{-1} s^{-1} and dissociation rate constant of 1.60 s^{-1} (**figure 6.10a**).

The SH3-only controls showed a high degree of non-specific interactions between the domain and the probes (**figure 6.10b**). Nevertheless, these non-specific interactions have a significantly lower amplitude than the binding interaction. They would also, in theory, be less prevalent in the data collection curves that use Las17-contacted probes due to a reduced availability of free probe surface. Furthermore, introducing a heterozygous ligand fit during data analysis did not improve fitting and increased the final chi squared value reached supporting the validity of the low-micromolar binding affinity given.

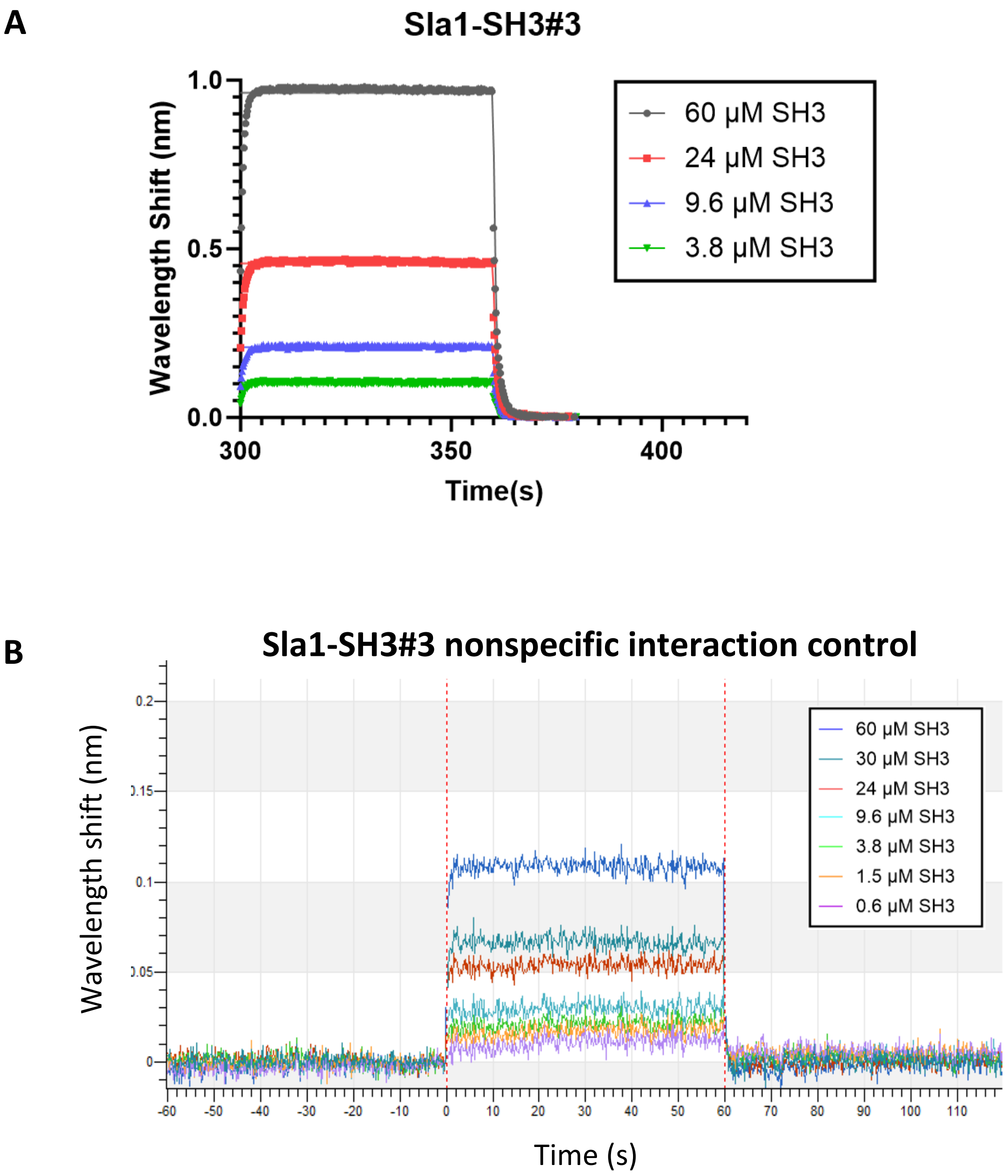


Figure 6.10: Singly expressed Sla1-SH3#3 BLI binding curves. Binding of Sla1-SH3#3 to the 300-422 fragment of Las17 (7.5 µg/mL) across Sla1 concentrations (60-3.8 µM). Curve fitting was undertaken using the "Association then dissociation" linear regression fit in GraphPad Prism 10.0.2. **A**) Ni-NTA probes pre-incubated with Las17-His were placed in Sla1-containing buffer at t = 300 and moved into a buffer-only well at t = 360. Curves were fitted using same data processing steps as detailed in **method 2.6.6** and plotted using in GraphPad Prism 10.0.2. **B**) The previous assay was repeated whilst omitting the loading of Las17 onto the probes to produce a no-Las17 control for assessing nonspecific probe-SH3 interactions. The concentration range was extended to 7 concentrations (60-0.6 µM).

The caveat of non-specific interactions to the 16.71 µM affinity may reduce the accuracy of the exact value given. However, the affinity is consistent with that of the first SH3 domain (~2-fold difference). Experimental literature provides support for a similar binding affinity between both the first and third SH3 domains of Sla1 and Las17. An *in-vivo* mutagenesis study revealed that swapping these two domains was less detrimental than anticipated (Dionne *et al.*, 2021). This may be explained if both domains bound with similar affinities.

6.3.2.4 BLI: Sla1-SH3#1-2

The next step following binding assays for each of the individual Sla1 SH3 domains was to measure the affinity of multi-domain constructs. These could then be used to validate the affinities of the first and third domains whilst helping predict the unknown affinity of the second domain. Furthermore, expression of larger constructs may help to ensure correct domain folding and thereby produce more accurate data. It was expected that Sla1-SH3#1-2 would bind the immobilised Las17 ligand with two modes. First would be the high affinity tandem interaction involving both domains. Second would be a lower affinity interaction between the unbound tracts and SH3#1. The latter would occur at higher analyte concentrations when the PP2 tracts would become saturated with SH3#2.

Sla1-SH3#1-2 was expressed and purified as a single, tandem-SH3 construct. A single assay was performed using a fixed concentration of probe-bound Las17 and a range of five Sla1 concentrations (250 nM, 100 nM, 40 nM, 16 nM, and 6.4 nM). As expected, complex binding occurred at higher concentrations. This can be accounted for using the 2:1 model available in the Blitz analysis software (Data Analysis HT v12.0.2.59) which gave a tandem affinity of 0.0877 ± 0.0036 µM, an association rate constant of 3.5x10^{5} M^{-1} s^{-1} and dissociation rate constant of 0.03 s^{-1} (**figure 6.11a**). Another method of obtaining an affinity was to fit the data using the lower concentration binding curves where this complex binding effect was least prevalent (40 nM, 16 nM, and 6.4 nM). This gave a tandem affinity of 0.0996 ± 0.0038 µM, an association rate constant of 5.5x10^{5} M^{-1} s^{-1} and dissociation rate constant of 0.054 s^{-1}

which is in good agreement with the 2:1 model fitting and was taken as the binding affinity of SH3#1-2 and Las17 (**figure 6.11b**).

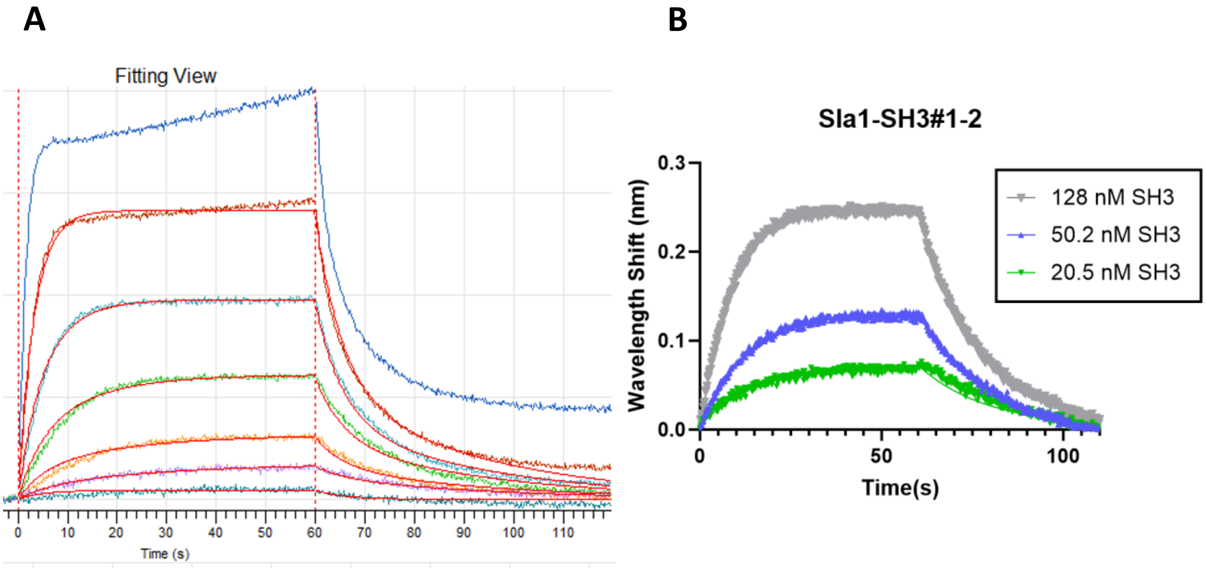


Figure 6.11: Sla1-SH3#1-2 tandem construct BLI binding curves. Binding of the Sla1-SH3#1-2 tandem SH3 construct to the 300-422 fragment of Las17 (7.5 µg/mL). Ni-NTA probes pre-incubated with Las17-His were placed in Sla1-containing buffer at t = 0 and moved into a buffer-only well at t = 60. Curve fitting was undertaken using **A**) the 2:1 binding model of the Blitz analysis software (Data Analysis HT v12.0.2.59) and **B**) an “Association then dissociation” linear regression fit in GraphPad Prism 10.0.2. The non-specific interactions observed with the single SH3 constructs were not significant for SH3#1-2 and thus, the processing steps as detailed in **method 2.6.6** were omitted.

Such a tight binding was to be expected considering the spatial proximity of SH3#1 and SH3#2 allowing for an extremely high avidity effect (Williamson, 2023). A K_d of 87.7 nM is also broadly consistent with the value of 56 ± 8 nM value determined by Feliciano and Di Pietro using a quantitative ligand-depletion approach in PBS (Feliciano and Di Pietro, 2012).

This consistency of the experimental result with another group between separate techniques also lends credence to the other affinities determined using the same experimental setup and data processing steps.

6.3.2.5 BLI: Sla1-SH3#1-3

Determining the affinity between the full length Sla1 SH3 construct (SH3#1-3) and Las17 would provide useful insight into the Sla1/Las17 binding scheme. Furthermore, it would help to provide additional confidence to the Sla1#1-2 and Sla1#3 affinities estimated in this thesis. A triplicate of assays was performed using a fixed concentration of probe-bound Las17 and a range of three Sla1 concentrations (80 nM, 32 nM, and 12.8 nM). This yielded an equilibrium dissociation constant of 0.0396 µM ± 0.0052 µM, an association rate constant of 3.9x10^{5} M^{-1} s^{-1} and dissociation rate constant of 0.016 s^{-1} (**figure 6.12**). Only lower concentrations of analyte were used to mitigate the effect of Las17 saturation as justified in section **6.3.2.4**.

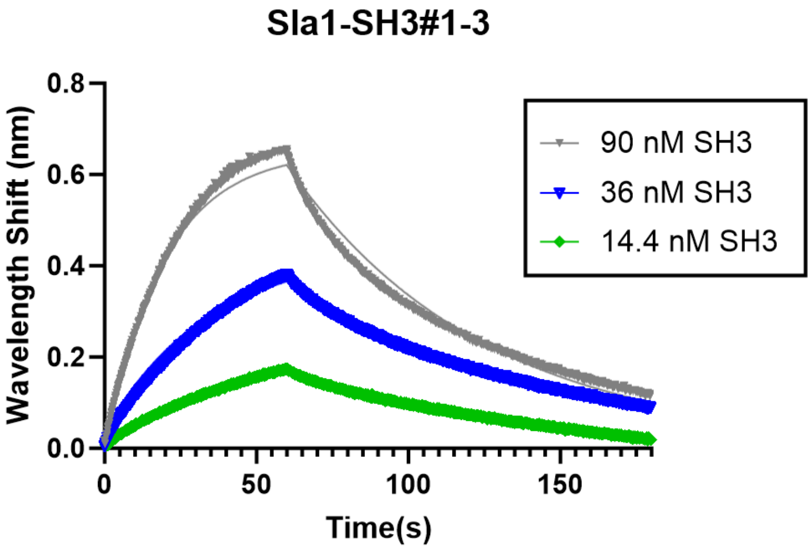


Figure 6.12: Sla1-SH3#1-3 tandem construct BLI binding curves. Binding of the Sla1-SH3#1-3 tandem SH3 construct to the 300-422 fragment of Las17 (7.5 µg/mL) across three Sla1 concentrations (64-10.2 nM). Ni-NTA probes pre-incubated with Las17-His were placed in Sla1-containing buffer at t = 0 and moved into a buffer-only well at t = 60. Curve fitting was undertaken using the "Association then dissociation" linear regression fit in GraphPad Prism 10.0.2.

This affinity is just over twice as tight as the Sla1-SH3#1-2 tandem. The weakly binding SH3#3 and long peptide linker to SH3#1-2 did not increasing the affinity as substantially as the jump from single SH3#1 to tandem SH3#1-2 – domains separated by a much smaller linker.

6.3.3 BLI: Ysc84-SH3

One of the key Las17-binding SH3 domains to arrive at the patch following the SLAC complex is Ysc84. This protein is recruited between the arrivals of Las17 and the Arp2/3 complex and thus may play a pivotal role in exposing the polyproline region to actin through Sla1-competition. Therefore, this affinity of the Ysc84 SH3 domain was determined through BLI and used as a proxy for the "cloud SH3" affinity parameters later used in the computational model.

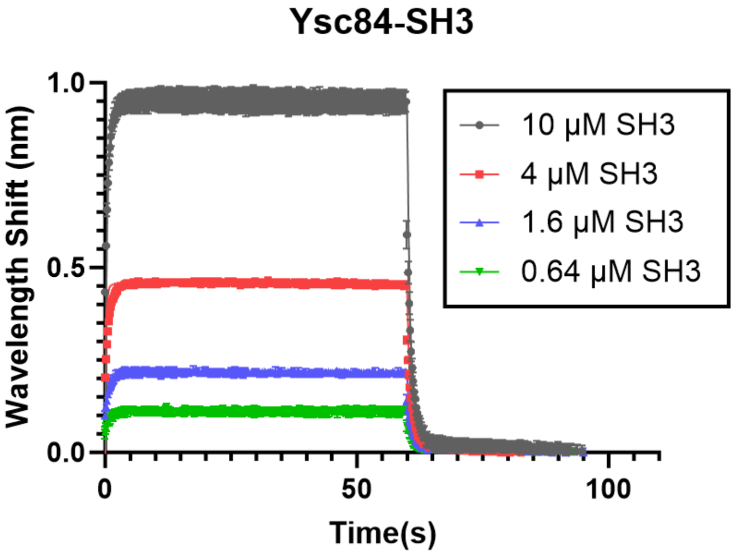


Figure 6.13: Ysc84-SH3 construct BLI binding curves. Binding of the Ysc84-SH3 construct to the 300-422 fragment of Las17 (7.5 μg/mL) across 4 Ysc84 concentrations (10-0.64 μM). Ni-NTA probes pre-incubated with Las17-His were placed in Ysc84-containing buffer at t = 0 and moved into a buffer-only well at t = 60. Curve fitting was undertaken using the "Association then dissociation" linear regression fit in GraphPad Prism 10.0.2.

A duplicate of assays was performed using a fixed concentration of probe-bound Las17 and a range of four Ysc84-SH3 concentrations (1000 nM, 400 nM, 160 nM, and 64 nM). This yielded an equilibrium dissociation constant of 2.18 μM ± 0.41 μM, an association rate constant of 7.29x10^5 M^{-1} s^{-1} and dissociation rate constant of 1.59 s^{-1} (**figure 6.13a**).

6.3.4 Model Inference: BLI

The affinity of Sla1-SH3#2 was re-calculated using the same method as described in **section 4.2.2.1**. Likewise, the affinity of Sla1-SH3#2 was varied and its effect on the tandem affinity observed through the modified output. The BLI affinities had significant advantages over the previously used values - these being a consistent buffer composition and the employment of the same experimental technique for both the SH3#1 and tandem affinity.

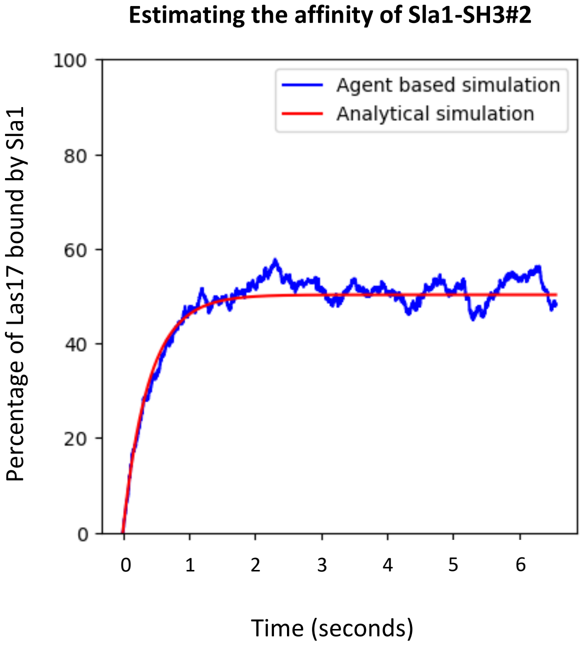


Figure 6.14: Using Model Beta to predict the affinity of Sla1-SH3#2. The k_{on} and k_{off} rates for Sla1-SH3#1-2 were simulated using a simple ODE model to produce an analytical binding curve (red). This was then compared to an output of Model Beta where SH3#1 was fixed at 7.5 µM K_d and SH3#2 was varied (blue). The SH3#2 affinity required for these curves to produce a consistent binding curve (shown above) was 15 µM.

A 7.5 µM SH3#1 affinity was fixed into the simulation and an overall binding affinity of 0.0877 µM for the tandem was achieved using an SH3#2 affinity of 20.0 µM (**figure 6.14**). A value of 7.5 µM for the first SH3 domain was taken as a rough average of all the BLI data.

Using the same technique as above, the affinity of the 3rd SH3 domain was estimated. An overall affinity of 0.040 µM for the full Sla1 SH3 peptide (SH3#1-3) was produced using the fixed affinity values of 0.0877 µM for SH3#1, 20 µM for SH3#2 and 30 µM for SH3#3 (**figure 6.15**). A predicted value of 30 µM for SH3#3 is around twice as weak as the experimentally determined affinity (16.5 µM). However, considering the assumed signal interference derived from non-specific binding, along with the high error value for the experimental measurement (± 9.62 µM) this value of 30 µM is within a consistent range – that being several times weaker than SH3#1. This modelling provided an important validation for the inference modelling used within this section through reaching similar values (low micromolar range).

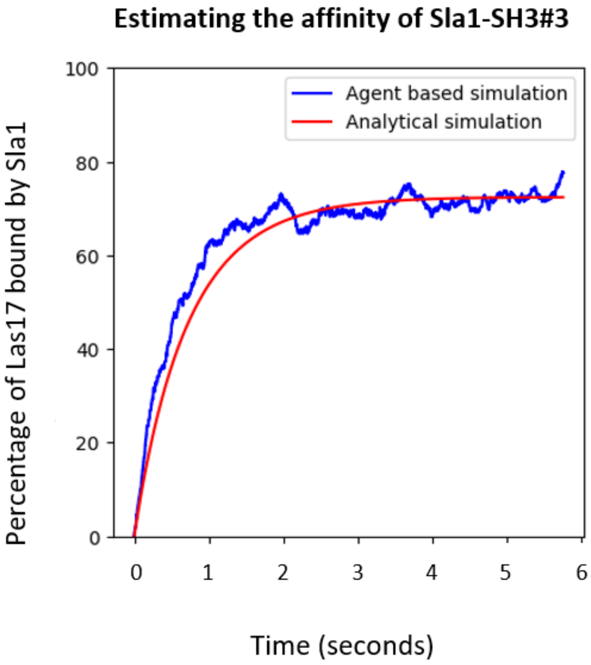


Figure 6.15: Using Model Beta to predict the affinity of Sla1-SH3#3. The k_{on} and k_{off} rates for Sla1-SH3#1-3 were simulated using a simple ODE model to produce an analytical binding curve (red). This was then compared to an output of Model Beta where SH3#1 was fixed at 7.5 µM K_d, SH3#2 was fixed at 15 µM K_d and SH3#3 was varied (blue). The SH3#3 affinity required for these curves to produce a consistent binding curve (shown above) was 30 µM.

6.4 Discussion of binding affinities used for Model Beta.

The second computational model shown within this thesis (Model Beta as described in **chapter 6**) had a greater availability of data during its creation as a result of the BLI results. These values were chosen as more suitable affinities for the model for two reasons. First was a consistency between the Sla1-SH3#1-2 affinity given by BLI in this thesis (0.0877 µM) and a quantitative ligand-depletion approach found within literature (0.050 µM) (**section 6.3.2.4**). The second reason was that BLI affinities were determined in PBS buffer as opposed to the G-buffer used in MST. G-buffer has an extremely low ionic strength whereas PBS is more analogous to the 0.5x KME G-buffer (sometimes referred to as F-buffer) used in the pyrene assay as well as the cytosol of yeast. Values used in Model Beta are given in **table 6.4**.

Up to 5 domains could be defined for each protein-type within Model Beta with no alterations to the code. Therefore, it was decided to simulate all three SH3 domains of Sla1 in this model. A Sla1-SH3#1 affinity of 7.5 µM was chosen as the average binding affinity calculated from BLI data. A 0.0877 µM affinity for Sla-SH3#1-2 and a 0.040 µM affinity for Sla-SH3#1-3 were likewise obtained from BLI data. These were then used to define an effective concentration for Sla1-SH3#2 (20 µM) and Sla1-SH3#3 (30 µM) via computational inference as described in section **6.3.4**. We decided to use the model-estimated affinity for SH3#3 as reproducing the overall tandem affinity which was concluded as more important in characterising Las17 behaviour. The model-estimated affinity is within a 2-fold range of the experimentally determined value and thus still remains generally consistent with experimental data.

Binding locations for these interactions were determined using our binding site scores (**figure 4.10**) and spot array data (E Allwood, personal communication). Both of these predictive tools predicted that Ysc84 binds the Las17-bm1 with a higher affinity than the other tracts. To reflect this, bm1 was given the BLI-determined value of 2.2 µM while bm2-5 were given affinities at half the maximally calculated binding strength (4.4 µM).

Our simulations with Model Beta began before the elucidation of individual actin-tract affinities described in **section 6.1.2**. Nonetheless, the K_d values chosen sat either within, or close to, their predicted range in PBS. The rate analysis used values from the lower end of these ranges (to be more consistent with our other simulation results) and are indicated when used. The affinities determined through Biolayer Interferometry are significantly lower than their MST-equivalent. However, this can be explained by the difference in buffer composition with the PBS used in BLI being more analogous to the simulated system than the G-buffer used throughout MST.

<table>
<tr><th colspan="5">Peptide</th><th colspan="5">bm1 affinity</th><th colspan="5">bm2 affinity</th><th colspan="5">bm3 affinity</th><th colspan="5">bm4 affinity</th><th colspan="5">bm5 affinity</th></tr>
<tr><td colspan="5">Sla1-SH3#1</td><td colspan="5">-</td><td colspan="5">-</td><td colspan="5">7.5 µM (1)</td><td colspan="5">-</td><td colspan="5">7.5 µM (1)</td></tr>
<tr><td colspan="5">Sla1-SH3#2</td><td colspan="5">-</td><td colspan="5">-</td><td colspan="5">-</td><td colspan="5">20 µM (2)</td><td colspan="5">-</td></tr>
<tr><td colspan="5">Sla1-SH3#3</td><td colspan="5">-</td><td colspan="5">30 µM (2)</td><td colspan="5">-</td><td colspan="5">-</td><td colspan="5">30 µM (2)</td></tr>
<tr><td colspan="5">Cloud SH3</td><td colspan="5">-</td><td colspan="5">-</td><td colspan="5">1.0 µM</td><td colspan="5">1.0 µM</td><td colspan="5">-</td></tr>
<tr><td colspan="5">Ysc84 SH3</td><td colspan="5">2.2 µM (1)</td><td colspan="5">4.4 µM (1)</td><td colspan="5">4.4 µM (1)</td><td colspan="5">4.4 µM (1)</td><td colspan="5">4.4 µM (1)</td></tr>
<tr><td colspan="5">Actin</td><td colspan="5">-</td><td colspan="5">4 µM (3)</td><td colspan="5">-</td><td colspan="5">2 µM (3)</td><td colspan="5">2 µM (3)</td></tr>
<tr><td colspan="5">Actin (where indicated)</td><td colspan="5">-</td><td colspan="5">8 µM (4)</td><td colspan="5">-</td><td colspan="5">2.5 µM (4)</td><td colspan="5">5 µM (4)</td></tr>
<tr><th colspan="30">Baseline rates/ratios for calculations</th></tr>
<tr><td colspan="15">SH3 domains</td><td colspan="15">Actin</td></tr>
<tr><td colspan="5">k_{on} (µM^{-1} s^{-1})</td><td colspan="5">k_{off} (s^{-1})</td><td colspan="5">K_{d} (µM)</td><td colspan="5">k_{on} (µM^{-1} s^{-1})</td><td colspan="5">k_{off} (s^{-1})</td><td colspan="5">K_{d} (µM)</td></tr>
<tr><td colspan="5">$0.98 * 10^9$</td><td colspan="5">63.7</td><td colspan="5">0.065</td><td colspan="5">$7 * 10^6$</td><td colspan="5">0.17</td><td colspan="5">0.024</td></tr>
<tr><th colspan="30">Other parameter values</th></tr>
<tr><th colspan="6">Binding radii</th><th colspan="6">Barbed end k_{on}</th><th colspan="6">Barbed end k_{off}</th><th colspan="6">Pointed end k_{on}</th><th colspan="6">Pointed end k_{off}</th></tr>
<tr><td colspan="6">7.346 nm</td><td colspan="6">11.6 M^{-1} s^{-1}</td><td colspan="6">1.4 s^{-1}</td><td colspan="6">1.3 M^{-1} s^{-1}</td><td colspan="6">0.8 s^{-1}</td></tr>
<tr><th colspan="12">Las17 dimerisation affinity when cooperatively binding actin</th><th colspan="6">Side actin k_{on}</th><th colspan="6">Side actin k_{off}</th><th colspan="6">Ysc83-YAB affinity for actin</th></tr>
<tr><td colspan="12">0.85 µM (5)</td><td colspan="6">2.18 M^{-1} s^{-1}</td><td colspan="6">$1.3 * 10^3$ s^{-1}</td><td colspan="6">0.15 µM</td></tr>
</table>

Table 6.4. Model Beta default parameter values. Parameters used for running Model Beta. The justification for all parameters are given either in main text for this section or **section 4.2.3**. Parameter values with a superscript of 1 were obtained using BLI data collected as part of this thesis, values with a superscript of 2 were calculated using Model Beta, values with a superscript of 3 fall close to the calculated MST affinities of section **Table 6.2**, values with a superscript of 4 fall within the centre of PBS affinity range estimated using the calculated MST affinities of section **Table 6.2**, and a superscript of 5 indicates the value was taken from the literature (Urbanek *et al.*, 2015).

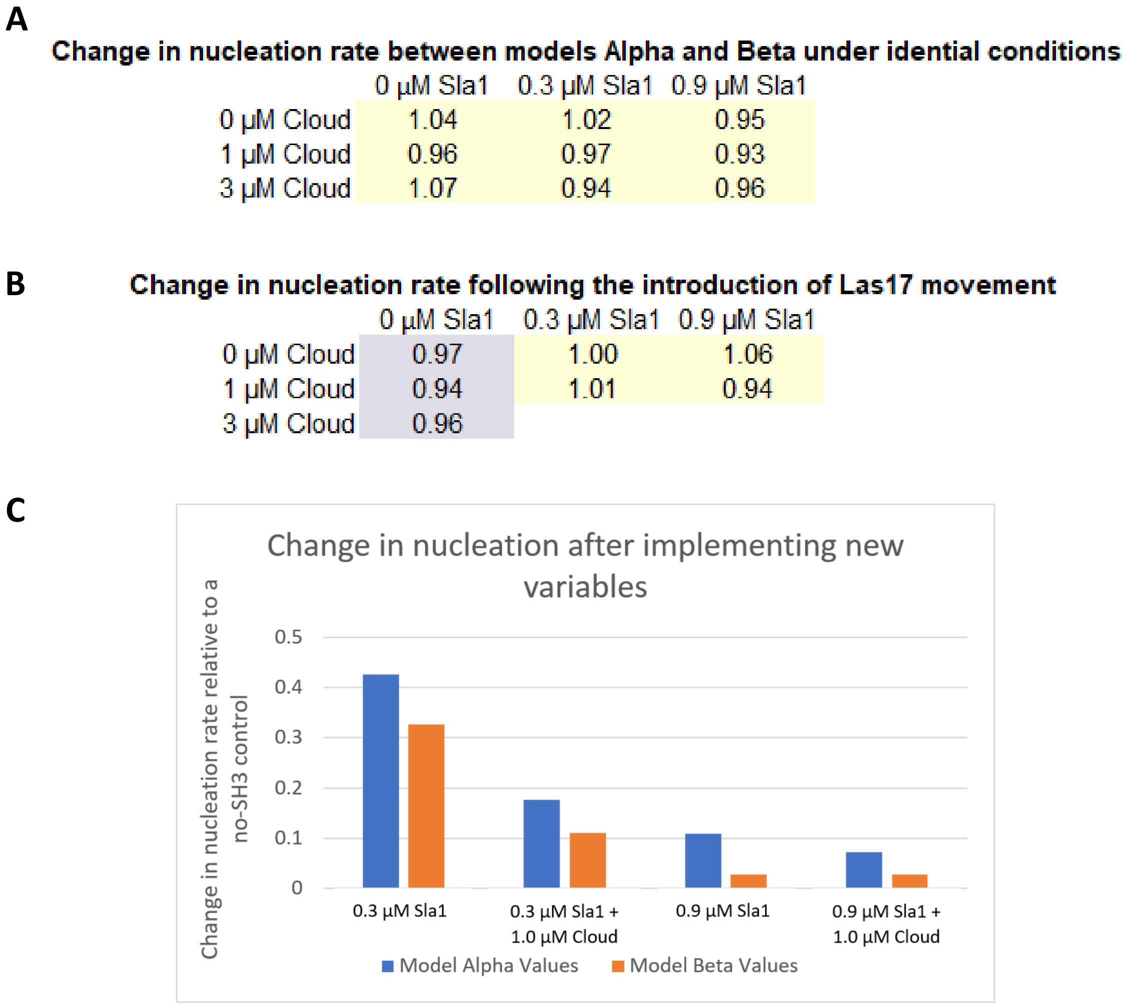
6.5 Validating Model Beta

Following the construction and internal validation of Model Beta, it was important to compare the model outputs with those of Model Alpha to assess the consistency between them. Each Las17 agent in Model Beta was arranged into a regular lattice using the initiation file and fixed in place by excluding all Las17 PlatformAgents from executing the movement phase. This reproduced the same spatial rules that were imposed on Las17 by Model Alpha. The actin binding rate and SH3 sensitivity analysis from **chapter 4** was then reproduced using the same parameter values (given in **table 6.4**).

The nucleation rates measured were consistent between the two models as all Model beta results fell within 10% of their Model Alpha counterparts (**figure 6.16a**). Because the code is almost completely new in Model Beta, this consistency provided a useful validation for the functionality of both ABMs along with a greater confidence in the comparison of their conclusions. The SH3 analysis produced using the default actin rates is the only comparison shown in **table 6.4** for brevity as this ±10% consistency trended across all other analyses.

A key element of Model Beta is that Las17 is capable of both lateral and rotational movement. The binding affinity calculation for Sla1-SH3#2 was recalculated as the new flexibility between binding sites conferred by the movement of Model Beta was predicted to increase Sla1-SH3#1 binding to PP4. This resulted in the K_d of Sla1-SH3#2 being increased from 1800 µM to 3300 µM. A binding affinity this weak would be an unreasonable estimate. Fortunately, Sla1-SH3#2 affinity calculations using data from the BLI assays yielded a more reasonable value of 15 µM (**section 6.3.4**).

A sensitivity analysis was performed to assess how inclusion of movement may affect our results beyond the known change to the previous Sla1-SH3#2 estimation. This was conducted by running unmodified Model Beta simulations with the default variables of Model Alpha. The exception to this was that a K_d of 3300 µM was used for Sla1-SH3#2 to remove any effects which may be derived from this known difference. This revealed that the movement itself did not significantly affect the simulation output as the calculated nucleation rates fell within 10% of their Model Alpha values (**figure 6.16b**).



A

Change in nucleation rate between models Alpha and Beta under idential conditions

	0 µM Sla1	0.3 µM Sla1	0.9 µM Sla1
0 µM Cloud	1.04	1.02	0.95
1 µM Cloud	0.96	0.97	0.93
3 µM Cloud	1.07	0.94	0.96

B

Change in nucleation rate following the introduction of Las17 movement

	0 µM Sla1	0.3 µM Sla1	0.9 µM Sla1
0 µM Cloud	0.97	1.00	1.06
1 µM Cloud	0.94	1.01	0.94
3 µM Cloud	0.96		

C

Change in nucleation after implementing new variables

Change in nucleation rate relative to a no-SH3 control
0.5
0.4
0.3
0.2
0.1
0
0.3 µM Sla1
0.3 µM Sla1 + 1.0 µM Cloud
0.9 µM Sla1
0.9 µM Sla1 + 1.0 µM Cloud
Model Alpha Values
Model Beta Values

Figure 6.16: Validating behavioural consistency between Model Beta and Model Alpha. (**A-B**) The difference in values (result X/result Y) when comparing two nucleation rate data sets. This includes (**A**) comparing results of Model Beta with those of Model Alpha and **B**) comparing a sensitivity analysis after the introduction of Las17 movement with an equivalent analysis produced without Las17 movement (both data sets were produced using Model Beta). Results are coloured yellow if they are within ±10% of their comparator value and grey if they are control comparisons. **C**) The change in nucleation rate when in the presence of SH3 domains relative to a control containing only actin and Las17 with no SH3 domains.

The final stage of analysing the changes between the two AMBs was to repeat the sensitivity analysis for a third time. This time, Las17 could move whilst the new parameters determined throughout this chapter were implemented – including representing all three SH3 domains of Sla1. Inclusion of the new model parameters increased the inhibition conferred by Sla1 as indicated by the reduced nucleation rates (**figure 6.16c**). This reduction was particularly pronounced at higher Sla1 concentrations which may be attributed to increased binding of Sla1-SH3#1-3 platforms across multiple Las17s. For example, the nucleation rate of Model Beta with 0.3 µM Sla1 was 76% of the rate produced by Model Alpha. Increasing the Sla1 concentration to 0.9 µM increased this discrepancy further with Model Beta observing a rate 26% lower than its Model Alpha counterpart.

For example, 0.3 µM Sla1-SH3#1-3 reduced the nucleation rate to 0.78-times the value observed with the same concentration of Sla1-SH3#1-2. Model Beta simulates all three Sla1 SH3 domains whilst Model Alpha only includes the SH3#1-2, hence this result was expected. Another marked difference is the drop in inhibition provided via cloud SH3 domains. This was also predicted considering the 2-fold increase of affinities.

6.6 Investigating the differences between simulated and experimental results

Las17 simulated using Model Alpha produced a nucleation rate many orders of magnitude higher than suggested by Las17 experimental assays. Three possible alternatives were considered to address this disparity. First was that parameter values chosen for the simulations may have been inaccurate, second was that Las17 may need to multimerise to bind actin at higher affinities, while third was the possibility of an alternative nucleation scheme. Each of these three possibilities were addressed using a series of Model Beta sensitivity analyses.

6.6.1 Exploring reasonable parameter adjustments

Model Beta was modified to implicitly model monomeric G-actin. This was done by allowing Las17 binding motifs to recruit actin using a probability function (rather than using agent location). Each binding event then generated an actin agent which was “killed” (i.e., removed from the system) after dissociation from Las17. Actin filaments were simulated implicitly using quadratic formulas whilst the timestep was increased to 1 ms. These modifications reduced the nuances of the system whilst permitting significantly faster running speeds. A local sensitivity analysis was then performed by systematically decreasing the magnitude of the association/dissociation rates until the model generated a polymerisation curve which overlapped with experimentally derived pyrene assays. The experimental assay was plotted using a simple ODE curve fitted to the initial (0-10 minutes) polymerisation rate of a pyrene assay conducted by a member of the Ayscough lab (E Allwood, personal communication). K_d values for actin to the polyproline motifs were fixed at the weakest value predicted for high salt buffer (**11.5** µM, **3.5** µM, and **6.5** µM for bm2, bm4, and bm5 respectively) to limit the analysis to parameters lacking experimental data.

Reproducing an experimentally consistent pyrene curve required a base actin-Las17 association rate of 0.002 uM^{-1} s^{-1} and dissociation rate of 10^{-9} s^{-1} (**figure 6.17**). These values are over 100,000,000 times smaller than typically observed rates (Pang *et al.*, 2012). Binding sites in exposed polyproline regions (such as the actin-binding motifs simulated here) are expected to bind using fast k_{on}/k_{off} rates due, in part, to their high rigidity which reduces the magnitude of the entropy drop which accompanies binding (Williamson, 2023). The uncharacteristically slow rates required to match experimental and ABL outputs are in direct contradiction to both our prediction and reasonability threshold. This suggests that the assigning of incorrect rate parameters cannot account for the extreme deviation between simulated and experimental results.

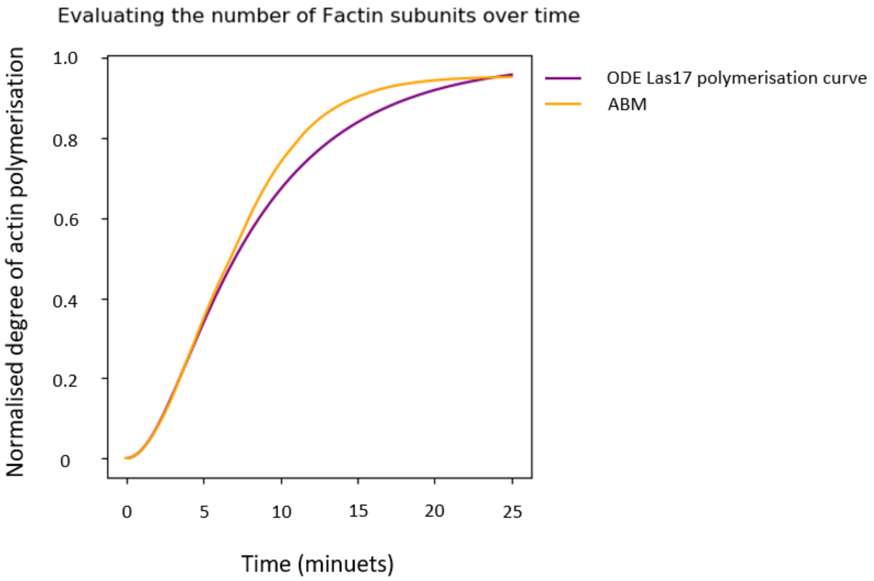
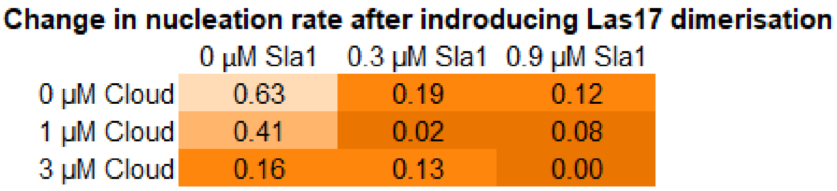


Figure 6.17: Fitting agent-based modelling to experimental data. 300 nM Las17 (with 5 µM actin) was simulated using Model Alpha to produce a simulated pyrene curve (normalised degree of actin occupying an F-actin state). The k_{on} and k_{off} between actin and Las17 were decreased until the ABM was generally consistent to the most active Las17 pyrene curve obtained by the Ayscough lab. This gave an indication as to how much these variables must be changed to reproduce experimentally consistent polymerisation rates.

6.6.2 Exploring the effects of Las17 multimerization

One of the unaccounted-for mechanisms highlighted through our work with Model Alpha was that multimerization of Las17 may play an important role during actin binding. This was supported through our retrospective analysis of Ayscough lab experimental data as explained in **section 4.5.6.1**. In lieu of robust experimental data, our multimerization model was implemented using the smallest set of elements (i.e., following the principle of parsimony). This included Las17 dimerising and binding actin with a weak affinity, Las17 dimerisation affinity increasing when both nucleators are bound to a shared actin agent, and actin affinity increasing upon Las17 dimerisation. This was the simplest case identified which could explain our experimental evidence whilst particularly fitting our observation that multiple Las17s coordinate actin during MST stoichiometry assays. A more detailed explanation can be found in **section 5.2.2**. Key parameter values included a 0.15 µM dimerisation affinity when mediated through actin cooperativity, no dimerisation in the absence of actin cooperativity, and a 5-fold reduction to the K_d of each actin-binding site when Las17 is in a monomeric state. We began investigating this mechanism with a sensitivity analysis.



Change in nucleation rate after indroducing Las17 dimerisation

	0 µM Sla1	0.3 µM Sla1	0.9 µM Sla1
0 µM Cloud	0.63	0.19	0.12
1 µM Cloud	0.41	0.02	0.08
3 µM Cloud	0.16	0.13	0.00

Figure 6.18: Investigating how dimerisation affects Las17 activity for cross-filament nucleation. The change in nucleation rates following introduction of the Las17 dimerisation mechanism to the cross-filament nucleation scheme. Results are coloured yellow if they are within ±10% of their comparator value and progressively more orange the greater the rate reduction. The presence of SH3 domains exacerbates the nucleation reduction conferred by dimerisation.

Dimerisation reduced the rate of nucleation to just over half (63%) the original no-dimerisation activity when no SH3 domains were present (**figure 6.18**). The rate reduction became progressively more pronounced in the presence of increasing SH3 domain concentrations. This culminated in no nucleation being observed when in the presence of 0.9 µM Sla1 and 3 µM cloud SH3s. Inhibition was scaled more aggressively with both Cloud SH3 and Sla1 concentrations as evidenced by falling values. Each box in the figure is calculated from **equation 6.5** and thus the smaller the number, the more negative the effect that dimerisation had upon the rate.

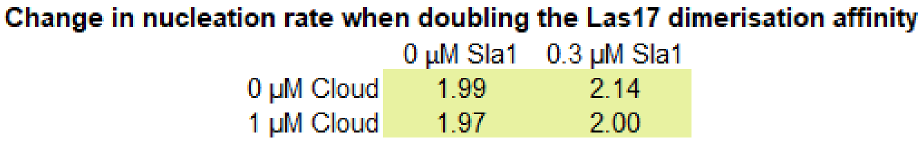
$$value\ in\ the\ figure = \frac{rate\ produced\ using\ dimerisation}{rate\ produced\ without\ dimerisation} \qquad (6.5)$$

A small degree of cloud-induced inhibition relief of Sla1 was identified when 0.3 µM Sla1 was combined with 3 µM cloud SH3s. This is consistent with the occasional closed-system inhibition relief identified in **chapter 4**.

6.6.3 Increasing dimer affinity

The parameter values chosen for Las17 dimerisation (0.150 µM) were estimated based upon the peptide concentrations known to nucleate actin filaments (0.150 – 3.00 µM). No experimental data was available to derive a sensible parameter range and thus a local sensitivity analysis had to be performed to better understand the relationship between parameter values and behavioural patterns.

Doubling the affinity from 0.15 µM to 0.075 µM approximately doubled all nucleation rates when compared against simulations run using the default affinity of 0.15 µM (**figure 6.19**). This suggests that changes to the dimerisation rate affect all simulations by the same degree regardless of SH3 composition. An equal parameter effect preserves the observed behavioural pattern regardless of value chosen which provides confidence that an inaccurate estimation would not significantly influence our model conclusions.



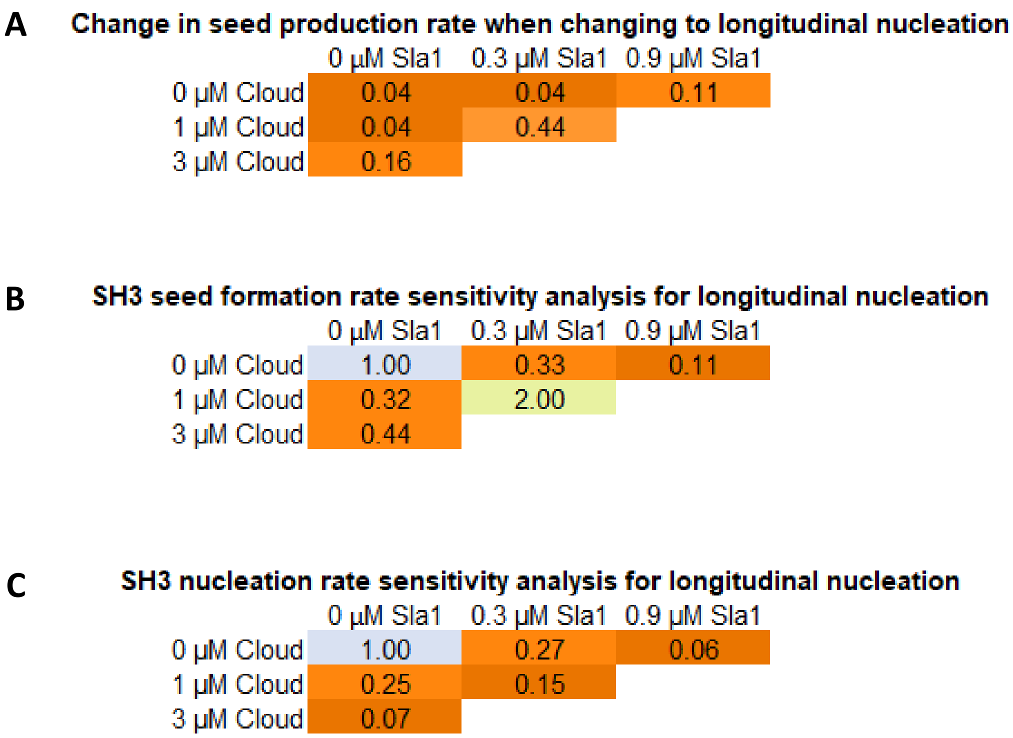
Change in nucleation rate when doubling the Las17 dimerisation affinity

	0 µM Sla1	0.3 µM Sla1
0 µM Cloud	1.99	2.14
1 µM Cloud	1.97	2.00

Figure 6.19: How doubling the dimerisation affinity affects the nucleation rate. The change in nucleation rates following a doubling of the Las17 dimerisation affinity. All simulations were undertaken using the Model Beta cross-filament nucleation scheme.

6.6.4 Comparing nucleating mechanisms

The third mechanism theorised was an alternative nucleation scheme. Model Beta simulations can be executed using either of two Las17 nucleation modes. These include a cross-filament nucleation scheme (shared with Model Alpha and the tandem nucleators Cobl and Leiomodin) or a longitudinal nucleation scheme (observed within other tandem nucleators including Spire and JMY) (**figure 1.6**). Selecting the chosen nucleation scheme could be done easily within the pre-initiation script. As described more thoroughly in **section 5.2.1**, the longitudinal nucleating mechanism posits that three actin monomers bound to Las17 must recruit an additional G-actin to seed the second sub filament. This recruited monomer interacts weakly with a K_d of only 600 μM as suggested by an analogous lateral interaction within a well referenced computational paper (Sept and McCammon, 2001; Reaction F, table 1). We refer to this recruitment as “second sub filament seeding”. The first lateral interaction requires the recruitment of a second G-actin to essentially “lock in” the weakly interacting subunit. All subsequent interactions (including interaction of the second G-actin) use rates typical of a polymerising actin filament. Therefore, Model Beta recorded a seed production rate which counted the number of fully seeded filaments (i.e., those that had undergone second sub filament seeding). It was theorised that adopting this mechanism may reduce the activity of simulated Las17 which would increase its parity to experimental pyrene-actin polymerisation curves.



A Change in seed production rate when changing to longitudinal nucleation

	0 µM Sla1	0.3 µM Sla1	0.9 µM Sla1
0 µM Cloud	0.04	0.04	0.11
1 µM Cloud	0.04	0.44	
3 µM Cloud	0.16		

B SH3 seed formation rate sensitivity analysis for longitudinal nucleation

	0 µM Sla1	0.3 µM Sla1	0.9 µM Sla1
0 µM Cloud	1.00	0.33	0.11
1 µM Cloud	0.32	2.00	
3 µM Cloud	0.44		

C SH3 nucleation rate sensitivity analysis for longitudinal nucleation

	0 µM Sla1	0.3 µM Sla1	0.9 µM Sla1
0 µM Cloud	1.00	0.27	0.06
1 µM Cloud	0.25	0.15	
3 µM Cloud	0.07		

Figure 6.20: Understanding the effects of longitudinal nucleation. The change in either seed formation rates (A-B) or nucleation rates (C) across a variety of sensitivity analyses. Results are coloured grey for controls and progressively more orange the greater the rate decrease. **A**) Comparing the rate of seed production when moving from a cross filament to a longitudinal nucleation scheme. **B**) Performing a seed production sensitivity analysis for the longitudinal scheme reveals that increasing the concentration of SH3 domains has diminishing returns on inhibition with 0.3 µM Sla1 and 1 µM Cloud SH3s even enhancing the rate. **C**) Replotting the sensitivity analysis for longitudinal nucleation to show the rate of nuclei, rather than seed, formation reveals that increasing the concentration of SH3 agents predictably decreases the nucleation rate – even when the seed production rate is enhanced.

As shown in **figure 6.20a**, the introduction of a longitudinal nucleation mechanism reduced the rate of filament formation by an even greater degree than we expected. The rate of actin seed formation in the absence of SH3 domains fell to just 4% the rate of cross-filament nucleation. However, the reduction in seed formation did not scale as aggressively with SH3 concentration when compared to cross-filament nucleation. For example, when 1 µM of cloud SH3s were present, switching to longitudinal nucleation reduced the rate of seed formation to 4% the cross-filament rate. Meanwhile, when 3 µM of cloud SH3s were present, switching to longitudinal nucleation only reduced the rate to 16% the cross-filament rate.

Showing only the longitudinal sensitivity analysis reveals that increasing the cloud SH3 concentration from 1 µM to 3 µM actually reduced inhibition of the seed formation rate from 32% the no-SH3 control value to 44% (**figure 6.20b**). When in the presence of 1 µM cloud SH3s and 0.3 µM Sla1, this even enhanced seed formation 2-fold. **Figure 6.20c** shows the nucleation rates (i.e., formation of a longitudinal trimer) to understand the mechanics behind this phenomenon. Here it was identified that increasing the SH3 concentration increases the degree of inhibition in a similar manner to cross-filament nucleation. This suggests that sequestering of G-actin by Las17 tracts reduces the rate of second subunit seeding. Binding of SH3 domains to Las17 can reduce this sequestering, increase the concentration of unbound G-actin, and thus promote the seeding of filaments. This phenomenon is therefore a consequence of modelling an *in vitro* system with limited actin agents and should not be relevant to an *in vivo* setting where the concentration of actin is generally stable. In an *in vivo* environment, the relationship between SH3 domain concentration and seed formation is predicted to be more similar to the nucleation rate analysis.

6.6.5 Combining both of the theorised mechanisms

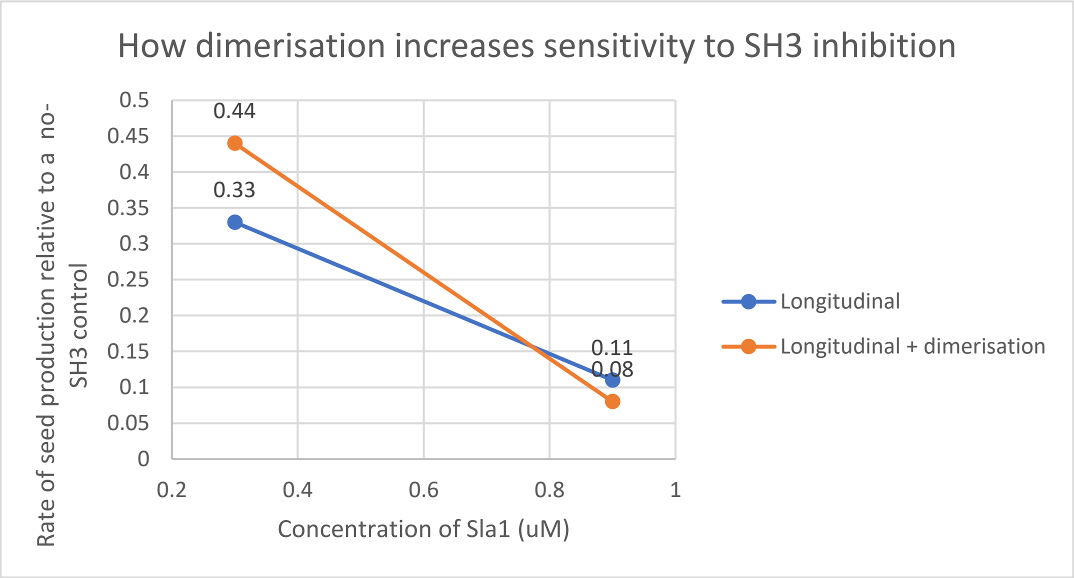
The sensitivity analyses performed using the newly explored Las17 mechanisms gave two similar, though distinct, behavioural profiles. Dimerisation reduced nucleation whilst significantly increasing the inhibitory effect of SH3 domains. Conversely, (in a closed *in vitro* system with finite actin agents) changing to a longitudinal nucleation scheme significantly reduces nucleation whilst reducing the inhibitory effect of SH3 domains. We suspect that Las17 likely employs a longitudinal nucleating scheme rather than cross-filament nucleation as will be discussed later (**section 6.11.3** and **section 7.1**). Therefore, an interesting question would be: How do these effects combine when both mechanisms are adopted?

A

SH3 seed formation rate sensitivity analysis for longitudinal nucleation with dimerisation

	0 µM Sla1	0.3 µM Sla1	0.9 µM Sla1
0 µM Cloud	1.00	0.44	0.08
1 µM Cloud	0.91	0.06	
3 µM Cloud	0.43		

B



C

SH3 nucleation rate sensitivity analysis for longitudinal nucleation with dimerisation

	0 µM Sla1	0.3 µM Sla1	0.9 µM Sla1
0 µM Cloud	1.00	0.06	0.01
1 µM Cloud	0.06	0.03	
3 µM Cloud	0.02		

Figure 6.21: Understanding the combined effects of longitudinal nucleation and dimerisation. A) A seed production sensitivity analysis for Las17 employing longitudinal nucleation and dimerisation mechanisms. Results are coloured grey for the control and progressively more orange the greater the rate decrease. **B**) Comparing how the reduction in nucleation as a function of SH3 domain concentration differs between simulations employing either longitudinal (blue) or longitudinal with dimerisation (orange) schemes. Simulations employing both schemes are more sensitive to SH3 domain concentrations as evidenced by the steeper decline in seed production rate when the Sla1 concentration was increased from 0.3 µM to 0.9 µM. **C**) Replotting the sensitivity analysis for longitudinal nucleation with dimerisation to show the rate of nuclei, rather than seed, formation reveals that SH3 agents have an even greater inhibition effect when compared against other schemes (**figure 6.18** and **figure 6.20c**).

Combining both dimerisation and linear nucleation results in an interesting combination of effects. The observed rate of filament seeding shows a reduced inhibition at low SH3 domain concentrations (**figure 6.21a**) when compared to the equivalent longitudinal nucleating analysis (**figure 6.21b**). However, the former showed a greater inhibition scaling with SH3 concentration – a relationship shared with the "cross-filament with dimerisation" analysis (**figure 6.21c**). For example, when 0.3 µM Sla1 was present, the longitudinal with dimerisation analysis gave a seed production rate of 44% the value observed in the no-SH3 control whilst the equivalent longitudinal without dimerisation simulation observed a more sizable activity drop to 33% of the control (**figure 6.21d**). When increasing the SH3 concentration to 0.9 µM Sla1, the rates fall to 8% and 11% of the control respectively which actually leaves the dimerisation system more inhibited. These rates illustrate the higher inhibition sensitivity to SH3 domains conferred due to the introduction of the dimerisation. This effect is most apparent when combining 0.3 µM Sla1 with 1 µM cloud SH3. This gives a 2-fold rate increase in the absence of dimerisation and an almost 20-fold rate decrease after the introduction of dimerisation.

6.6.6 Assessing how the different mechanisms affect experimental comparisons

It was observed in **chapter 4** (**section 4.3**) that emulating experimental pyrene curves with Model Alpha (cross-filament nucleation) produced a polymerisation rate many orders of magnitude greater than observed for Las17. It was this discrepancy that led to the predictions of longitudinal nucleation and dimerisation. Therefore, it would be interesting to assess how the introduction of these mechanisms changes the simulated emulation of pyrene curves.

Three simulations were completed using 300 nM Las17 and 5 µM actin in accordance with Ayscough lab experimental protocols. These employed either cross-filament nucleation, longitudinal nucleation, or longitudinal nucleation with dimerisation. All three simulations used actin-binding affinities from the centre of their PBS/Buffer predicted range calculated in **section 6.1.2**: 11.5 µM for PP1, 2.5 µM for PP3, and 4.5 µM for PP4. The F-actin binding affinity (seed K_d) was fixed at an estimated value of 5 µM to account for absence of the PP-tracts C-terminal to residue 422 which are theorised to stabilise the interaction between Las17 and F-actin (Allwood *et al.*, 2016). The value of 5 µM was chosen at it sits within the range of G-actin affinities and no experimental evidence was otherwise available. An experimental pyrene curve was produced using Las17(300-422)-His and actin at equivalent concentrations to the simulations, thereby allowing for experimental comparison.

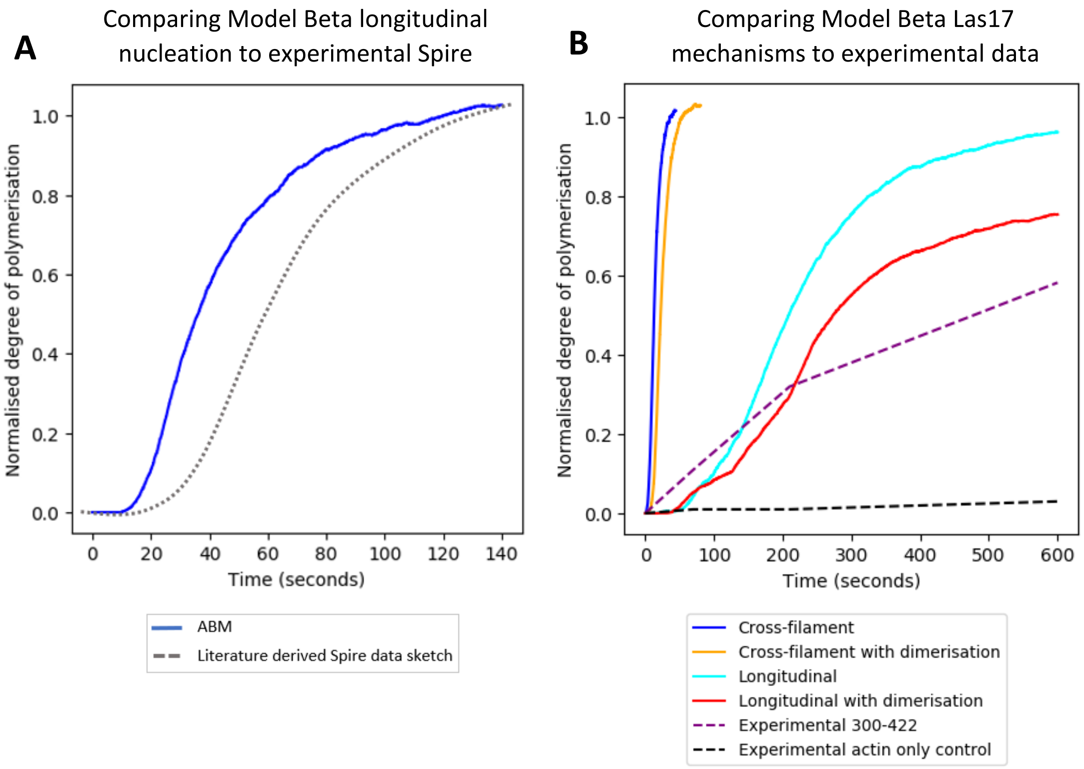


Figure 6.22: Comparing Model Beta polymerisation curves with experimental data. Las17 and actin were simulated using Model Beta and a variety of mechanisms to produce polymerisation curves for comparison to experimental pyrene data. **A**) 0.5 µM Las17 employing longitudinal nucleation was simulated with 4 µM actin and compared against a typical polymerisation curve from an analogous pyrene assay with Spire (obtained from **figure 1b** in Quinlan *et al.*, 2005). **B**) Simulating 0.3 µM Las17 with 5 µM actin and updated binding affinities (8 µM, 2.5 µM, and 5 µM) allowed for the comparison of nucleating mechanisms. Experimental Las17 data was taken from data provided by a member of the Ayscough lab (E Allwood, personal communication). The results revealed that mechanism schemes employing longitudinal nucleation show a much greater experimental consistency in early nucleation rates than those employing cross filament nucleation.

Changing the nucleation scheme from cross-filament to longitudinal had a profound negative effect on the rate of actin polymerisation which brought the simulated pyrene curve into much greater parity with experimental results (**figure 6.22a**). Introducing our dimerisation mechanism to the longitudinal nucleation scheme further reduced the simulated rate. This combination of both mechanisms agrees much more with the experimental data than the previous cross-filament nucleating scheme. Changing the Las17 concentration to 0.5 µM and actin to 4 µM and running the simulation using a longitudinal nucleation scheme reveals a broadly similar rate to the longitudinal nucleator Spire, increasing the confidence of our modelling (**figure 6.22b**).

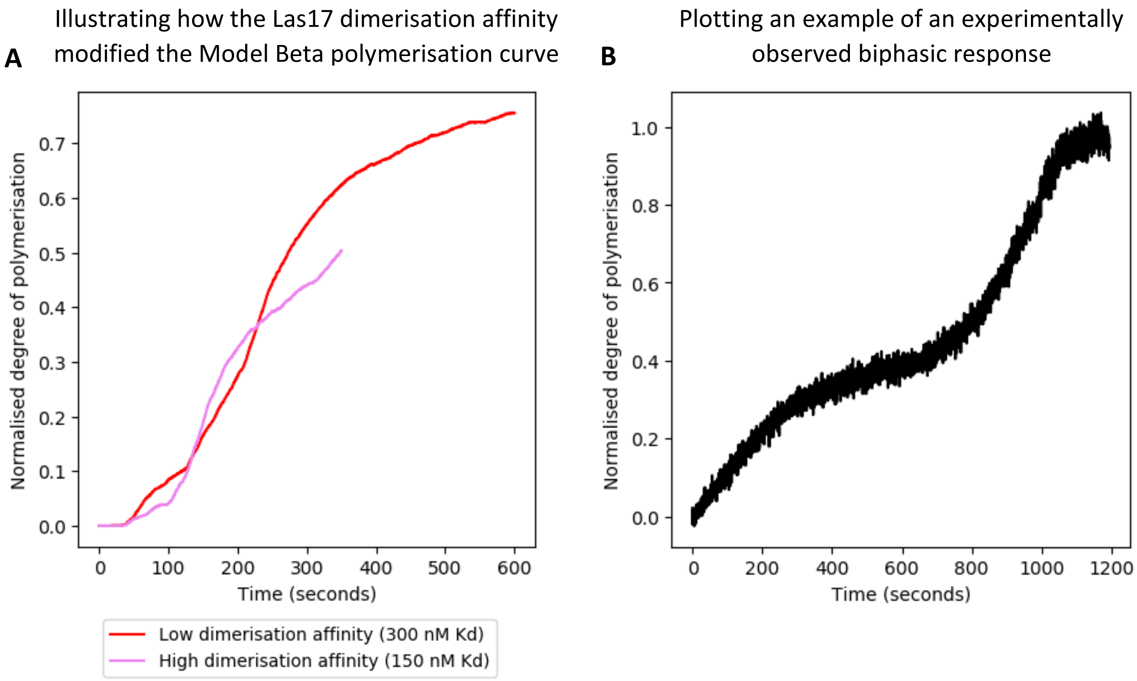


Figure 6.23: Characterising the biphasic nature of Las17 polymerisation curves. A) Simulated actin polymerisation curves reveal that combining longitudinal and dimerisation nucleating schemes introduces a biphasic characteristic. Increasing the affinity for Las17 dimerisation both enhances and delays the first phase (0 to ~120 seconds for a 300 nM affinity vs 0 to ~320 seconds for 150 nM affinity). The higher affinity curve (magenta) could not be completed past the early second phase due to computational time limits. However, the increasing rate clearly shows that this phase was entered at around 320 seconds. **B**) A pyrene curve using 300 nM Las17 and µM actin supplied by a member of the Ayscough lab produces a similarly biphasic profile (E Allwood, personal communication). This behaviour is highly variable between preparations of Las17.

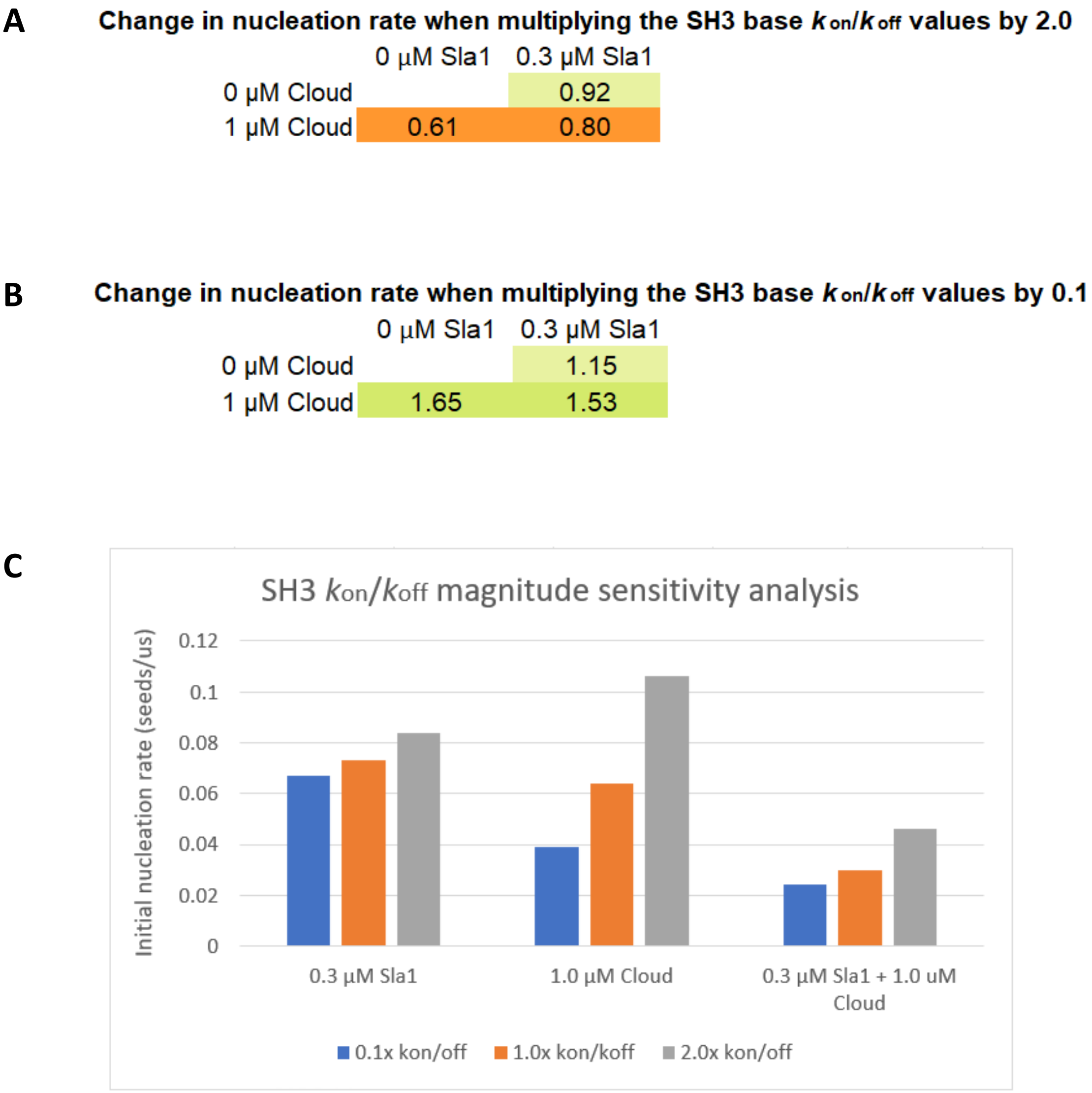
Whilst doubling the Las17 dimerisation affinity doubled the nucleation rate during cross-filament simulations (previously shown in **figure 6.19**), doubling the dimerisation affinity during longitudinal simulations had a surprising effect (**figure 6.23a**). The combination of Las17 dimerisation and longitudinal nucleation introduces a biphasic characteristic to the simulated polymerisation curve. Understanding actin sequestering can help rationalise this observation. At high concentrations of free G-actin, Las17 readily dimerises to bind actin and produce nuclei. Actin agents can be shared by two Las17 actin-binding sites due to the dimerisation mechanism and thus less G-actin is sequestered when compared to longitudinal nucleation with no dimerisation. Like dimerisation, the rate of second sub-filament seeding is also highly dependent upon the concentration of free G-actin. This results in dimerisation simulations producing a similar early polymerisation rate to longitudinal-only simulations (**figure 6.22b**). However, the probability of cooperative Las17 dimerisation through actin association decreases as the concentration of free actin drops. This reduces the rate of nucleation and thus slows the rate of F-actin accumulation. Meanwhile, a decrease in sequestering can also enhance filament elongation. These concentration dependent effects combine to produce a biphasic polymerisation curve in a manner challenging to predict without the aid of agent-based modelling. Most interestingly, a similar polymerisation profile is routinely observed during Las17 pyrene assays (**figure 6.23b**; **figure 6b** from Allwood *et al.*, 2016). Increasing the Las17 dimerisation affinity when both agents can cooperate to bind a simple actin results in a more pronounced, and delayed first phase.

6.7 Characterising the SH3 binding parameters

6.7.1 Effect of SH3 k_{on}/k_{off}

While BLI has given some insight into the k_{on}/k_{off} values of SH3 domain proteins, these values came with a degree of error. Meanwhile, the K_d values for cloud SH3s were estimated based upon typical SH3 binding affinities and thus come with particularly low confidence. We decided to investigate how changes to SH3 binding rates may affect Model Beta.

Changing the magnitude of k_{on}/k_{off} values for SH3 domains whilst keeping K_d constant shows that increasing these values also increases the inhibition associated with the SH3s (**figure 6.24a**). Likewise, decreasing the magnitude of these parameters decreases their associated inhibition (**figure 6.24b**). This effect was clearly more pronounced for cloud SH3 domains which relied on smaller individual domain K_d values (**figure 6.24c**). An effect conferred by the k_{on}/k_{off} magnitudes may suggest that actin monomers bound by Las17 have a high probability of being nucleated before their departure.



A

Change in nucleation rate when multiplying the SH3 base k_{on}/k_{off} values by 2.0

	0 µM Sla1	0.3 µM Sla1
0 µM Cloud		0.92
1 µM Cloud	0.61	0.80

B

Change in nucleation rate when multiplying the SH3 base k_{on}/k_{off} values by 0.1

	0 µM Sla1	0.3 µM Sla1
0 µM Cloud		1.15
1 µM Cloud	1.65	1.53

Figure 6.24: Investigating how SH3 k_{on}/k_{off} rates affects Las17 regulation. (**A**-**B**) Comparing the nucleation rate after multiplying the k_{on}/k_{off} values by **A**) 2x or **B**) 0.1x for every SH3 domain whilst keeping the K_d constant. Results are coloured yellow if they are within 10% of the comparator value, orange if the value decreases, and green if the value increases. **C**) Results from the first two subfigures were combined to illustrate how changing the magnitude of k_{on}/k_{off} affects tandem and single SH3s differently with a stronger correlation being observed for single domains (i.e., cloud SH3s).

6.8 Characterising the behaviour of Sla1

6.8.1 Why is Sla1 a tandem of SH3s?

One thing that becomes apparent when studying the interactome of the Las17 polyproline region is the prevalence of tandemly arranged binding domains. This leads to the reasonable question: Why don't these interactions rely on fewer, tighter binding domain and how does tandem binding affect the Las17 regulatory model hypothesised?

To investigate this, Sla1 agents were modified to become a single binding domain. The affinities of this domain were chosen such that the modified Sla1 (henceforth termed "single binding unit Sla1") occupied each of the Las17 tracts with the same proportion as tandem Sla1. This essentially tuned Sla1 into a single domain rather than a tandem to investigate how this changes the simulation behaviour. Tract affinities for this modelling are given in **figure 6.25a**. The investigative simulations were performed using cross-filament nucleation with 0.3 µM of the chosen Sla1 construct and either 0 µM cloud SH3, 1 µM cloud SH3, or 3 µM cloud SH3.

We found that Sla1 in the absence of cloud SH3s inhibits Las17 to ~30% the activity of the no-SH3 control regardless of the construct used (**figure 6.25a**). This was expected given that actin cannot bind the PP2/PP3 tracts whilst they are occupied by SH3 binding. Therefore, actin wholly competes with Sla1 via preventing association rather than through reducing the effective avidity. Increasing the concentration of cloud SH3s elucidates a significant difference between the two constructs. Tandem Sla1 was observed to be significantly more susceptible to cloud SH3 competition as reflected by nucleation rates being higher than their single binding unit counterparts. Plotting the reduction in PP3 availability due to Sla1 binding for each of the simulations clearly identifies the primary cause of this observation (**figure 6.25c**). Tandem binding leaves the Sla1 construct much more susceptible to cloud SH3 competition via avidity reduction when compared to the single binding unit.

A

Las17 binding motif	K_d for single binding unit Sla1 (μM)
Bm1	-
Bm2	10
Bm3	0.09
Bm4	0.08
Bm5	10

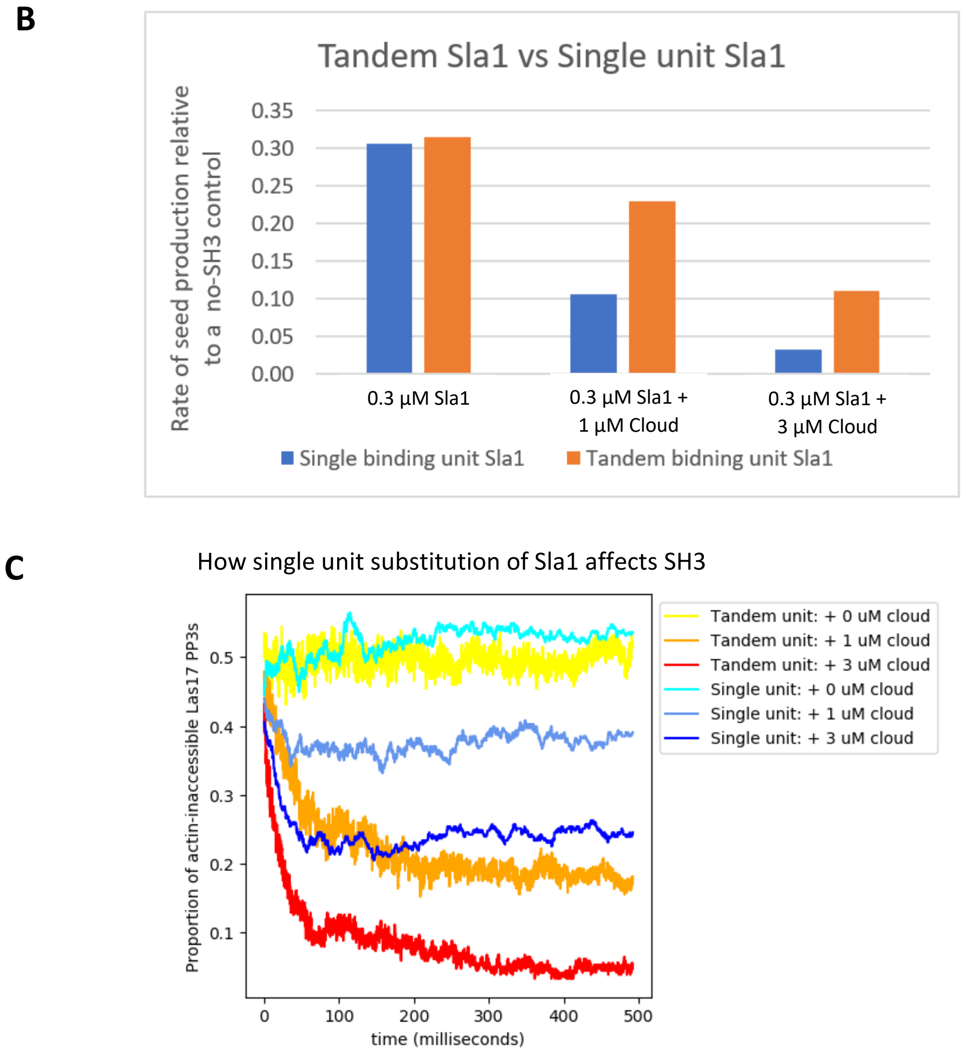


Figure 6.25: How tandem binding affects SH3 competition. A) The Las17 affinities required by the single unit Sla1 agent to reproduce the binding profile of tandem Sla1. **B**) Investigating how the rate of Sla1-inhibited actin seed production changes in the presence of an increasing concentration of cloud SH3s. Representing Sla1 as a tandem construct results in the inclusion of cloud domains conferring a reduced level inhibition as the negative effect of these binding events are partially countered by competition between Sla1 and cloud agents. **C**) Plotting the percentage of Las17 PP3 tracts bound by Sla1 reveals that the tandem construct (shades of yellow/red) is outcompeted by cloud domains more strongly than single construct Sla1 (shades of blue).

6.8.2 How sensitive is Sla1 to the linker distance?

The previous section revealed that the tandem nature of Sla1 plays an important role in the rapid Las17 departure in the presence of cloud SH3 domains. We followed this analysis by investigating how the length of the linker between Sla1-SH3#1 and SH3#2 domains also affects the behaviour of Sla1. A sensitivity analysis was conducted with the length of this linker increased 2-fold and 4-fold. It was predicted that the potency of Sla1 inhibition would decrease as the relative concentration of Las17 for unbound domains drops and thus the effect of avidity is reduced. However, the extent of this effect remained unknown.

Increasing the length of the linker distance decreased inhibition by Sla1 as expected (**figure 6.26a**). However, it also resulted in a greater degree of inhibition relief by the SH3 cloud which can be shown by plotting the degree of Sla1 binding over time (**figure 6.26b**). This reveals the importance of a short linker length in delivering robust control over Las17 activity.

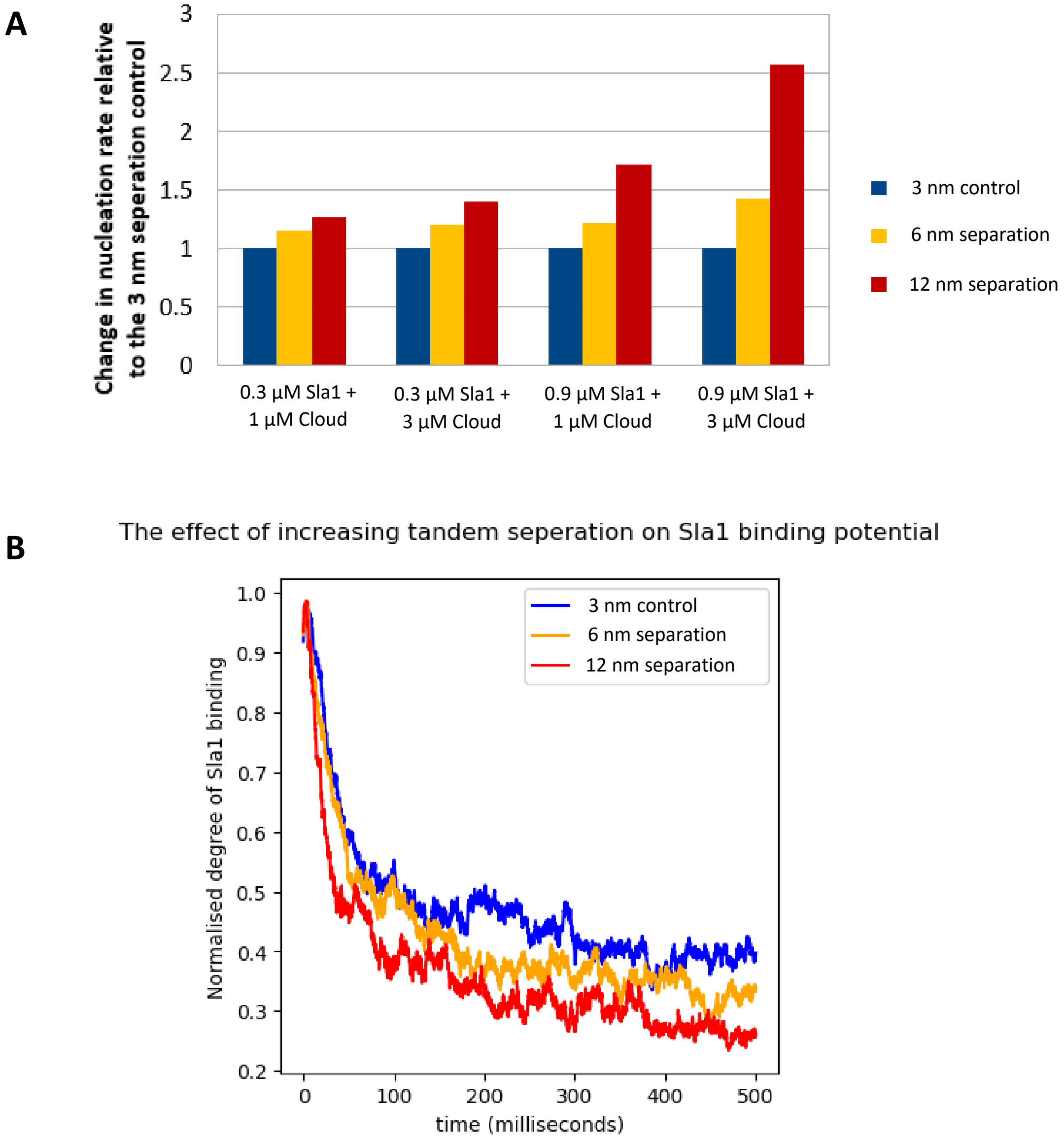


Figure 6.26: How size of the Sla1 tandem linker affects SH3 inhibition. A) How changing the maximal distance separating Sla1-SH3#1 and Sla1-SH3#2 affects the inhibition conferred by Sla1 relative to a 3 nm separation control. Increasing the distance reduces both inhibition of the tandem and increases its susceptibility to cloud SH3 competition. **B**) The degree of Sla1 binding (normalised to the highest binding percentage observed between the simulations) over time following exposure to the SH3 cloud. This value drops as Sla1 is outcompeted by the single SH3s and the rate of this departure is positively correlated to the length of the Sla1 linker separating the first and second SH3 domain.

6.8.3 Comparing directed and dynamic pre-equilibration between Las17 and Sla1

In vitro assays undertaken by the Ayscough lab typically combine Las17 and Sla1 prior to the addition of actin. An unintended consequence of this is that Las17 and Sla1 are effectively pre-equilibrated in the absence of actin. This state is emulated *in vivo* where Las17 and Sla1 arrive at the endocytic patch together as part of the SLAC complex (Feliciano and Di Pietro, 2012). Equilibrating simulations in an analogous way (described in **section 4.4.3**) not only reproduced this starting condition, but it also made it easier to elucidate many effects (e.g., the rate of Sla1 unbinding as a function of cloud domain concentration). Pre-equilibration could be undertaken through two different methods. First is through a directed approach. This involves calculating the proportion of Sla1 expected to be bound to Las17 at the given concentration within the pre-initiation script (using the equation for the SH3#1-3 K_d and the percentage bound formula of **equation 6.3**). Sla1 agents can then be pre-bound to Las17 (via the script) according to a binding probability prior to running the simulation. The second way of pre-equilibrating binding is through a dynamic approach. Here, Sla1 and Las17 are simulated alone for 0.5 seconds prior to the addition of actin. We decided to investigate the difference between these methods to identify any emergent phenomena which may have been overlooked. All simulations were undertaken using the cross-filament nucleation scheme, 343 Las17 agents (0.3 µM), 1029 Sla1 agents (0.9 µM), and run for 0.5 seconds of simulation time.

Surprisingly, we found that the dynamic approach of pre-equilibrating Las17 and Sla1 (produced 6 nuclei after 0.5 seconds) was almost three times as effective at conferring inhibition when compared to the directed approach (produced 17 nuclei after 0.5 seconds). The reasoning behind this can be inferred from the percentage of Las17 binding motifs bound by Sla1 at the end of the simulated second (**figure 6.27a**). This shows that the dynamically pre-equilibrated system results in tighter control over tract access. To better understand the phenomena responsible, the trajectory files produced after one second were scrutinised. Through this, we discovered that dynamically pre-equilibrating Sla1 and las17 allowed the formation of large Las17/Sla1 clusters with three of these supercomplexes containing over half of all Las17 agents (**figure 6.27b**).

Importantly, the simulation which was initiated following directed pre-equilibrium also accumulated large clusters by the end of the simulation (with the largest at 42 Las17s) suggesting that, as expected, these clusters slowly form regardless of pre-equilibration method. Nonetheless, employing either directed or dynamic pre-equilibration allows for simulations to begin with different levels of clustering, thus allowing for the analysis of the effect on Las17 activity.

A

Pre-equilibration method	**% Bm1 bound by Sla1**	**% Bm2 bound by Sla1**	**% Bm3 bound by Sla1**	**% Bm4 bound by Sla1**	**% Bm5 bound by Sla1**
Directed	-	55.69	61.51	67.63	53.06
Dynamic	-	61.22	69.38	71.42	63.27

B

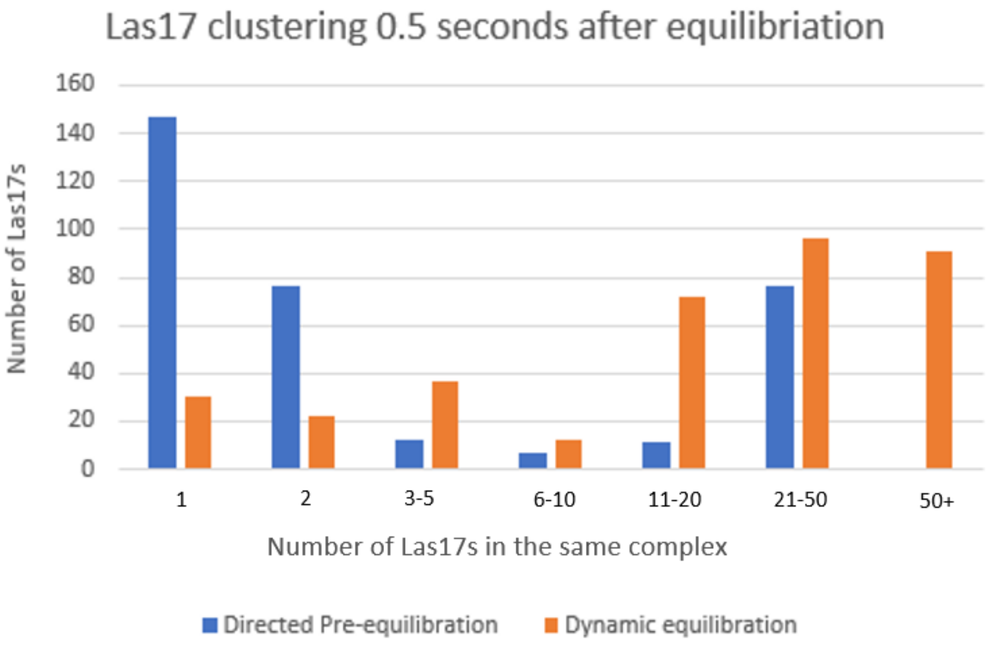


Figure 6.27: Investigating Sla1-mediated clustering of Las17. A) The percentage of each Las17 tract (Bm1-Bm5) bound by a Sla1 agent after one second. Allowing Sla1 and Las17 to dynamically equilibrate was observed to increase the percentage of Las17 bound by Sla1 at every tract. **B**) Plotting the number of Las17 agents in a complex with Z other Las17 agents (with Z being donated by the bin of the X axis). For example, following direct equilibration, 96 Las17 agents were within a complex that contained between 21 and 50 Las17s. The more the distribution favours higher bin values, the greater the degree of Sla1/Las17 clustering.

Repeating the pre-equilibration for a longer time crashed the simulation as the memory required to execute the priority recalculations exceeded the limit of the hardware. This suggests that a longer pre-equilibrating simulation may allow for the formation of even larger complexes.

All simulations undertaken following this section enforced a complex limit of 6 Las17 agents as this prevented large complexes (>>100 platforms) which cause instability to the simulation. Las17 clustering was identified in a later simulation that used 0.3 µM Sla1 due to encountering the same hardware limitation. However, the effect at this Sla1 concentration was not analysed in-depth due to time constraints. Nonetheless, this observation did make us aware that clustering occurs even with Sla1 concentrations as low as 0.3 µM. This may explain why the *in vivo* SLAC complex contains multiple copied of both Las17 and Sla1 (Feliciano and Di Pietro, 2012).

6.9 What is the regulatory function of Ysc84?

A cursory glance at the pattern of protein recruitment during CME highlights the important regulatory position held by Ysc84. It one of only of two SH3-containing proteins that is recruited to the endocytic patch in-between Las17 and Arp2/3 – the other being Bzz1 (Lu, Drubin and Sun, 2016). This leaves the protein very well placed to confer a dynamic regulatory effect on Las17-mediated nucleation. Ysc84 is also an actin binding protein capable of interacting with both G-actin (the preferred partner) and F-actin which may add a new and interesting dimension to Las17-mediated nucleation. A series of sensitivity analyses were performed across the Model Beta mechanisms. Each analysis was compared against a non-actin binding control which lacked the Ysc84 actin binding YAB domain. This control contained only the SH3 and thus helped distinguish between the effect of Ysc84's contribution to the SH3 cloud and the specific effect derived from its actin-binding function. Ysc84 binding is predicted most strongly at the bm1 site of Las17 (**section 4.2.3**). This PxxP core motif is separated from the actin-binding tracts and thus its occupancy with Ysc84 effectively adds another actin binding site to Las17 via the YAB domain. Consequently, the protein was predicted to enhance filament formation.

The YAB domain enhanced the formation of filaments relative to the SH3-only control across all longitudinal simulations (both with and without dimerisation) demonstrating the profound impact of the YAB domain within our model (**figure 6.28**). However, the magnitude of rate change was highly dependent on the nucleation scheme employed. Cross-filament nucleation showed no significant impact at 1.00 µM Ysc84 relative to its SH3-only control. However, inclusion of the YAB domain was inhibitory at high concentrations suggesting that G-actin sequestering by Ysc84 was negatively impacting nucleation **figure 6.28a**).

This observation differs significantly from the longitudinal nucleation analysis where 1.00 µM YAB-Ysc84 conferred a 12.6-fold increase to the seeding rate in the absence of Sla1 and a 26.6-fold enhancement in the presence of 0.3 µM Sla1 (**figure 6.28a**). This even resulted in the presence of Ysc84 increasing the rate of nucleation beyond the no-SH3 control and thereby changing Ysc84 from a Las17 inhibitor into a Las17 promotor (**figure 6.28b**). The introduction of dimerisation into the longitudinal nucleation scheme produced a similar effect. Here, 1.00 µM YAB-Ysc84 conferred a 12.2-fold enhancement in the absence of Sla1 and a 21-fold enhancement in the presence of 0.3 µM Sla1 (**figure 6.28a**). However, the higher inhibitory effect of SH3 domains within dimerisation simulations resulted in Ysc84 mostly functioning to restore seeding activity, rather than promoting it (**figure 6.28a**). At lower SH3 concentrations, the inclusion of Ysc84 still enhanced filament formation with 1.00 µM YAB-Ysc84 increasing the seeding rate 3.4-fold relative to the no-SH3 control.

These results indicate that the actin monomers supplied by the YAB domain have a greater positive impact on second sub filament seeding than nucleation. This is understandable as the first actin-binding interaction which seeds the second sub filament is dependent upon a significantly lower interaction affinity (600 µM) than actin binding directly to Las17 (1-2 µM) (Sept and McCammon, 2001). Increasing the concentration of YAB-Ysc84 from 1.0 µM to 3.0 µM reduces the seeding rate across all mechanisms (**figure 6.28a**). This is most pronounced in the longitudinal with dimerisation nucleation scheme. Here, the effect of Ysc84 changes from a promoter (339% seeding rate increase) to an inhibitor (17% seeding rate reduction).

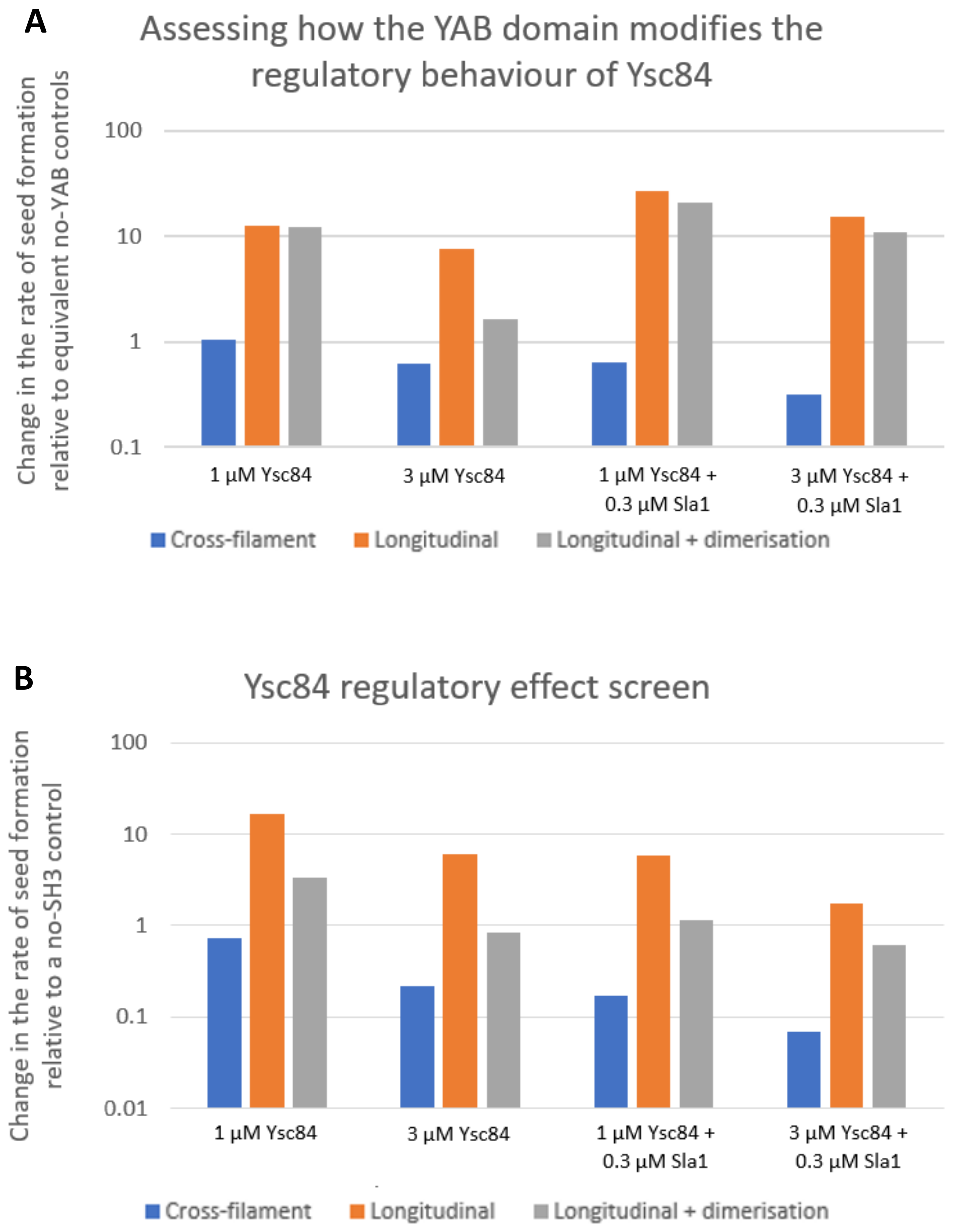


Figure 6.28: Predicting the function of Ysc84. Simulations were run at a variety of Ysc84 and Sla1 concentrations. **A**) The rate of seed production compared to the appropriate controls which were simulated using the same protein concentrations, but with Ysc84 lacking the YAB domain. All Las17 longitudinal simulations observed a rate increase conferred by the YAB domain. Inclusion of the YAB domain was mostly negative for cross-filament nucleation as any positive effects were offset by the reduction in nucleation as a result of G-actin sequestering by Ysc84. **B**) The simulations from the previous subfigure were plotted against a control lacking SH3 domains whilst employing the same Las17 nucleating mechanism. The presence of Ysc84 inhibited cross-filament nucleating Las17. However, both longitudinal with dimerisation and (to a greater extent) longitudinal without dimerisation mostly produced a rate increase when Ysc84 was within the system. For longitudinal with dimerisation, adding 1.0 µM Ysc84 to 0.3 µM Sla1 restored Las17 activity to the level of the no-SH3 control.

6.10 *In vivo* behavioural predictions

All the simulations analysed until this point had been undertaken using conditions that mimicked an *in vitro* context. The results generated from these may help direct *in vitro* experiments. Modifying the conditions and structure of the simulation space to better represent an *in vivo* context may provide insight into how our developing regulatory SH3 model may function in the cell and thereby provide experimental suggestions for *in vivo* work.

6.10.1 *In vivo* simulation modifications

The *in vitro* simulations assume a uniformly distributed, closed, static system with no Sla1 depletion. All four of these assumptions break down when looking through the lens of an *in vivo* context. Therefore, minor changes were implemented to better reflect this environment.

The periodic boundary condition imposed to represent a theoretically "infinite" solution was changed at the Z axis to incorporate the presence of a membrane. These boundaries were changed to represent an elastic surface which is a simplified approximation as polypeptide regions often form transient interactions with the lipid bilayer. However, to prevent the introduction of unknown variables and maintain model simplicity, we assume that over the course of a single time-step (1 μs), molecules approaching the membrane are deflected back towards the cytoplasm in an approximately elastic manner. The X and Y boundaries maintain their periodic boundary conditions to represent a theoretically "infinite" membrane plane.

Each PlatformAgent was granted an additional variable, termed *membrane_teather* which dictated how far that molecule could diffuse from the lower Z axis boundary (thereafter chosen as the membrane surface). PlatformAgents apply this condition to each domain to ensure that all the proteins simulated are localised correctly. This allowed proteins such as Bzz1 and Sla1 to be "tethered" to the membrane surface. Values for the *membrane_teather* of patch proteins were standardised at 80 nm to avoid any unforeseen effects that may derive from prescribing parameter values that are physiologically incorrect. A distance of 80 nm was taken as the typical distance of domains from their closest point of membrane tethering (~85 nm IDP for the SH3-YAB linker of Ysc84 and ~84 nm IDP for the FBAR-SH3#1 linker Bzz1) (Urbanek *et al.*, 2013; Zhao *et al.*, 2013). Las17 was fixed just above (10 nm) the membrane plane under the experimentally informed assumption that the polyproline region interacts with membranes (E Allwood, personal communication). This would significantly increase the local concentration of Las17 capture radii by around 1500 times.

The *in vitro* simulations reproduced the closed system employed in experimental assays whereby the concentration of actin depleted as filaments were polymerised. In an *in vivo* environment, endocytic patches make up only a small volume of the cell. This allows the G-actin concentration of the patch to remain approximately constant as new actin monomers can diffuse to the patch to 'replace' polymerised actin. To prevent actin sequestering within filaments, actin agents were moved to a random position in the simulation space following incorporation into a filament. The redistributed actin remained in an unbound state allowing it to continue diffusing and binding targets whilst the filament size was increased to represent a newly acquired subunit.

The patch region was fixed at a reasonable size (simplified as a square with an axis length of 200 nM). Endocytic patches in budding yeast have a size range between 50 and 300 nm in diameter (Berro and Pollard, 2014). A larger than average diameter was selected to allow for larger data sampling to improve our confidence in observed behaviours. Given that Model Beta represented the patch as a square, rather than a circle, the computational patch had an area ~2 times larger than patches at the centre of the observed size range in in yeast ($\pi 175^2$).

The concentration of free actin was fixed by preventing its sequestering into filaments as mentioned in the previous paragraph. However, the concentration of G-actin could only be kept constant by assuming that the loss of actin due to Las17 and Ysc84 binding was negligible. Patch associated proteins also accessed a separate X and Y periodic boundary conditions set to the aforementioned 200 nm diameter patch and centred on the membrane. This allowed the rest of the cubic simulation space surrounding the patch to be much greater in size (diameter of 617 nm) (**figure 6.29**). Because the number of agents generated by the initiation file was dependent upon the requested concentration of said agent, and because the concentration of an agent is dependent upon the volume/molecule, the number of actin agents was increased by allowing access to a larger simulation space. This allowed the number of actin agents to far exceed the number of actin-binding motifs/domains.

Model Beta patches were simulated dynamically as trajectory files could be amended with new agents to simulate the recruitment of new molecules over time. This involved taking the final trajectory file for one simulation (e.g., t = 0 to t = 10), copying new agents into the file (e.g., adding 200 Ysc84 proteins), and then using this modified trajectory as the initiation file for simulating the next phase of endocytosis (e.g., t = 10 to t = 15). Output csv files generated before and after the inclusion of new agents were combined to display a dynamic output. Sla1 could also be removed from the patch using a probability function. This function could remove unbound Sla1 from the simulation. Microscopy studies suggest that Sla1 occupies a separate phase of the endocytic patch than Las17 via four cargo-binding domains (Picco *et al.*, 2015). These interactions (e.g., the binding of large clathrin complexes) may further mask Sla1 SH3 domains from Las17. The “sequestering” of Sla1 agents can be reproduced by allowing for a stochastic depletion of Sla1.

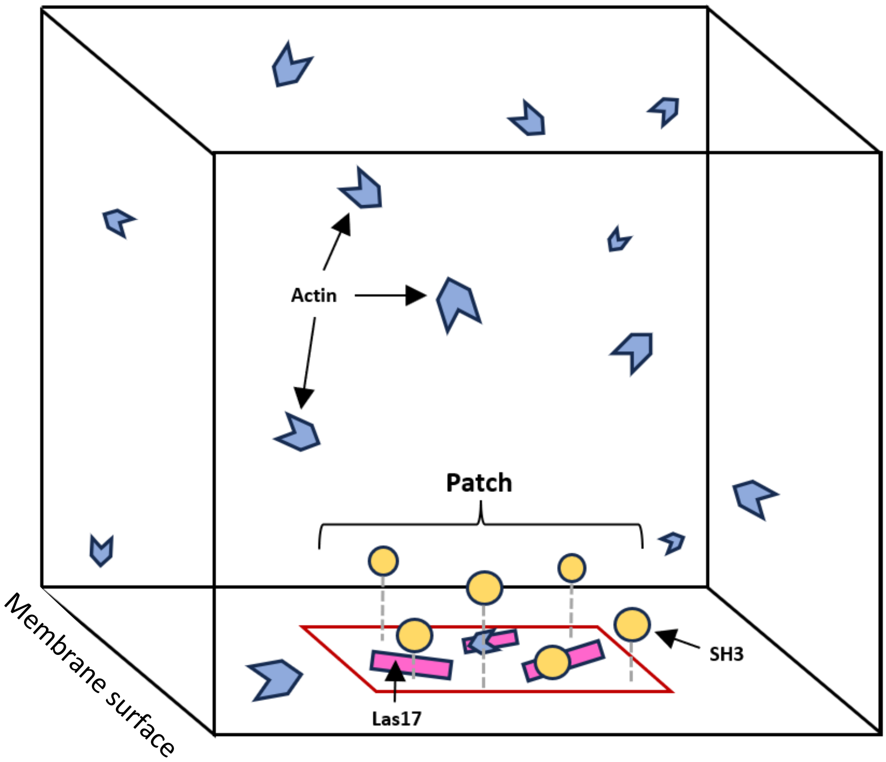


Figure 6.29: Illustrating the spatial organisation of Model Beta's *in vivo* modelling. Model Beta simulates the *in vivo* space as a cube with the lower Z representing the inner surface of the plasma membrane. The endocytic patch (shown as a red square) is smaller than the overall simulation space to allow for larger numbers of actin agents. This reduced the effect of excessive sequestering. All agents can either be non-patch associated (e.g., actin) which permits them to access the entire simulation space, or patch associated (e.g., Sla1) which constrains their coordinates to the patch. A membrane tether variable defines the maximal distance these agents can diffuse from the membrane surface. Las17 can also move around the patch region, however, the Z coordinate of the tracts are fixed just above the patch to account for assumed membrane interactions.

6.10.2 Outlining *in vivo* parameters

Endocytic patches change in composition throughout time as more proteins are recruited and lost from the region. Several studies have used high resolution fluorescence microscopy to track important endocytic proteins and provide both quantitative and qualitative insight into patch dynamics (Sun *et al.*, 2019). **Table 6.5** outlines the number of each molecule simulated within the *in vivo* simulations of Model Beta. As described in the previous section, the simulated actin patch was set towards the high end of the experimentally observed size range to allow for greater data sampling. Therefore, agent numbers at the higher end of the observed range were taken whilst the numbers for other proteins were taken such that their ratio to Las17 mirrored those seen within studies.

Protein	Peak amount observed experimentally	Amount in simulation
Actin	3 µM [(1)] [(2)]	3162 molecules (3 µM [(1)])
Las17	46-102 [(3)] [(4)]	125 molecules
Sla1	90-168 [(3)] [(4)]	250 molecules
Ysc84 (Lsb4)	Unknown, but assumed similar to other SH3s [(4)]	200 molecules
Bzz1	Unknown, but assumed similar to Sla1 [(3)] [(4)]	250 molecules
Cloud	Unknown as this agent represents background SH3 binding. Likely Lsb3 dominated.	100-200 molecules

Table 6.5: Estimating the number of patch molecules. Experimental studies used to base our computational estimates are given below and linked to tabular values with matching superscripts. Budding yeast possesses a few micromolar of cytosolic G-actin and values between 2 and 3 µM have been used in both agent-based and experimental studies to emulate an *in vivo* environment ([1]Wang *et al.*, 2016; [2]Schaller *et al.*, 2011). Las17 has been observed at between 46 and 102 molecules. Because the Model Beta patch had an area just under double the average value experimentally observed, 125 molecules were used for the *in vivo* modelling. This is just under 2x the average number of observed Las17s and only marginally above the upper observed value of 102 ([3]Picco *et al.*, 2015; [4]Sun *et al.*, 2019). These same studies both discovered that there are roughly two Sla1 proteins for every Las17 resulting in the selection of 250 Sla1 molecules (125*2) for the *in vivo* simulations. Bzz1 was implemented at 250 molecules. Both Sla1 and Bzz1 share a similar domain structure with regards to Las17 binding and were therefore simulated using the same agent count (shown later in **figure 6.3.2**). The number of Ysc84 agents is likewise unknown. Therefore, this protein was set at a protein count of 200 – just above those of other SH3s to account for the difference in patch size ([4]Sun *et al.*, 2019). Cloud SH3s represent all the unaccounted for SH3 interactions (including from the cytosol and early patch arrivals) and is thus not easy to estimate. Therefore, simulations were run using both 100 and 200 molecules.

Actin filaments dissociated from Las17 if they contain >5 subunits. The justification behind this was that the K_d of actin filaments to Las17 is an unknown and may reduce our ability to elucidate the relative effects of cloud proteins by itself influencing the nucleation rate. The objective of our ABM was not to provide accurate quantitative predictions, but rather qualitative predictions that can help direct experimental assay design and refine our regulatory hypothesis.

6.10.3 The effect of generalised cloud SH3 domains following SLAC arrival

The start of each simulation began with Las17 having arrived at the endocytic patch as part of the SLAC complex. This involves simulating Sla1 and Las17 for one second at the patch in the absence of actin. Following this pre-incubation, 3 µM actin was randomly distributed throughout the simulation space along with random spatial distribution of either 0, 100, or 200 cloud SH3 agents at the patch. The overall number of seeded filaments was recorded over 10 seconds.

All three simulations showed a slow nucleation rate and sensitivity to SH3 domains as a consequence of their close spatial proximity (**figure 6.30a**). This resulted in between 1 and 34 seeds being produced during the first 10 seconds of simulation across the tested mechanisms – much slower than in previous *in vitro* ABM assays. The presence of increasing cloud SH3 numbers progressively reduced the rate of filament production whilst accelerating the rate of Sla1 departure from Las17 (**figure 6.30b**). In a true *in vivo* environment, a delayed breakdown of the SLAC complex may delay dependant events such as the recruitment of Ysc84 to Las17 and linking cargo recognition to clathrin recruitment. Therefore, this shows that our regulatory cloud hypothesis must rely on a careful balance between the cloud-enhanced departure of Sla1 and competitive inhibition of the polyproline tracts. The results of cross-filament nucleation shown in **figure 6.30b** are consistent with the longitudinal nucleation scheme or longitudinal nucleation + dimerisation scheme. However, the slow rate of transition between nuclei and seed due to rate limiting second sub filament seeding resulting in almost no filaments being produced within the 10-second period simulated.

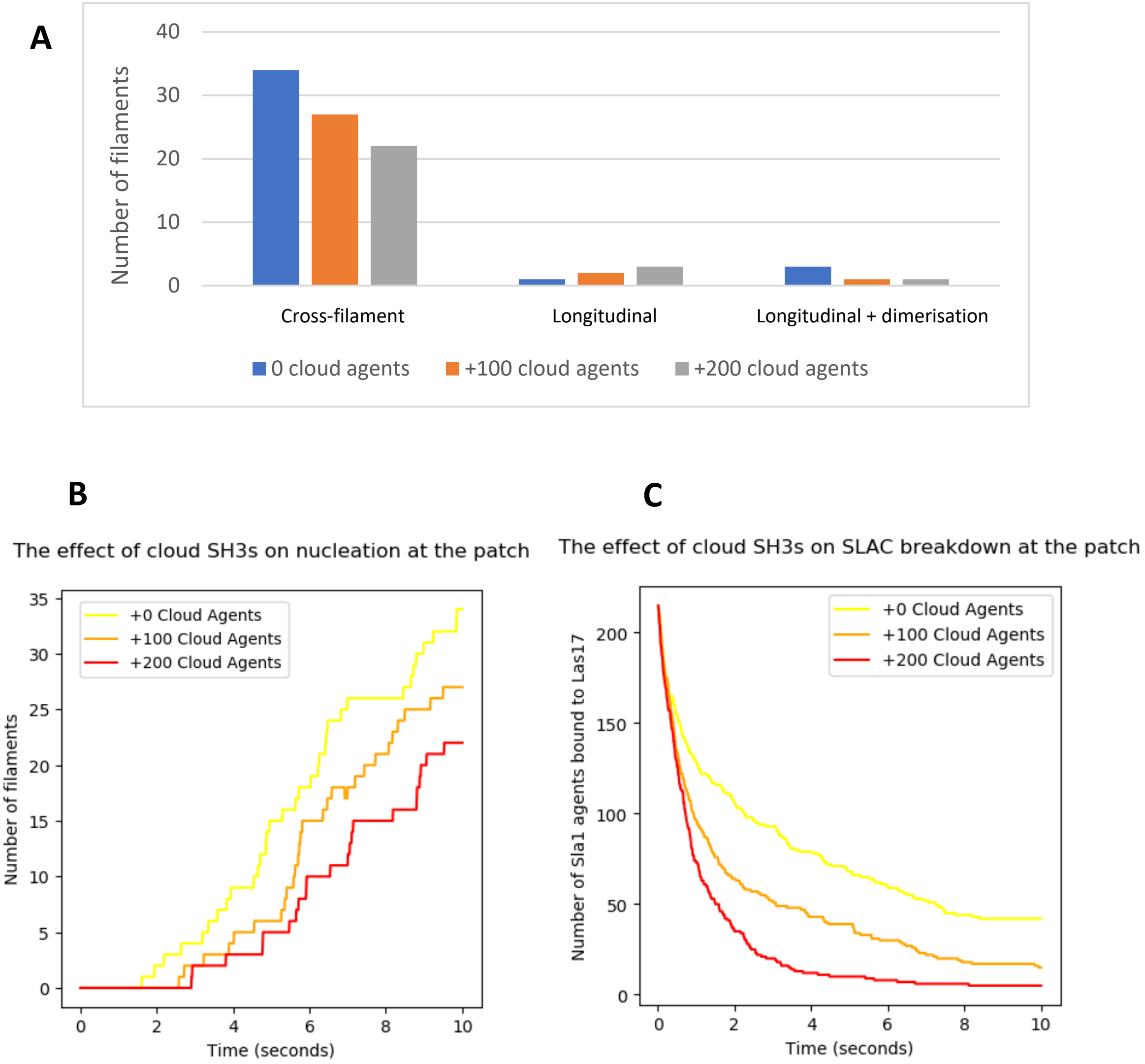


Figure 6.30: Impact of the generic SH3 cloud on the SLAC complex. All simulations were run for 10 seconds using the in vivo conditions outlined in **section 6.10.2**. Las17 and Sla1 were dynamically pre-equilibrated for one second prior to commencing the in vivo simulation (t = 0) to emulate arrival of the SLAC complex to the endocytic patch. **A**) The number of filaments produced over time for each of the Las17 nucleating mechanisms in combination with either 0, 100, or 200 cloud agents. Use of the longitudinal nucleation scheme results in almost no filament seeding occurring during the time simulated. Meanwhile, the cross-filament nucleating produced a much higher number of filaments. **B**) The nucleation of actin was adversely affected by the number of cloud agents. This figure shows the number of filaments produced over time in the presence of varying levels of cloud SH3s. **C**) However, the increasing cloud concentration greatly facilitated the loss of Sla1 from Las17 and thereby facilitating breakdown of the SLAC complex.

6.10.4 How does the recruitment of Ysc84 change patch behaviour?

The SH3 cloud changes in composition throughout endocytosis. The first major change following the arrival of Las17/Sla1 is the recruitment of Ysc84. After 10 seconds of simulation time, 200 Ysc84 agents were copied into the trajectory files closest to an accumulated simulation time of 10 seconds. This is generally consistent with the timescale observed *in vivo* (Lu, Drubin and Sun, 2016). The simulations were then run for another five seconds using the modified trajectory files as the initiation file. Recruitment of Ysc84 resulted in an increase in inhibition for the cross-filament model as the recruitment of G-actin from the YAB domain did not outweigh the negative effect of SH3 competition (**figure 6.31**). However, this effect differs substantially from the simulations employing a longitudinal nucleation scheme. Here, the introduction of Ysc84 induces a sharp burst in filament formation followed by a plateauing (particularly in the case of longitudinal + dimerisation) as the pool of nuclei awaiting second sub filament seeding were rapidly converted into filaments.

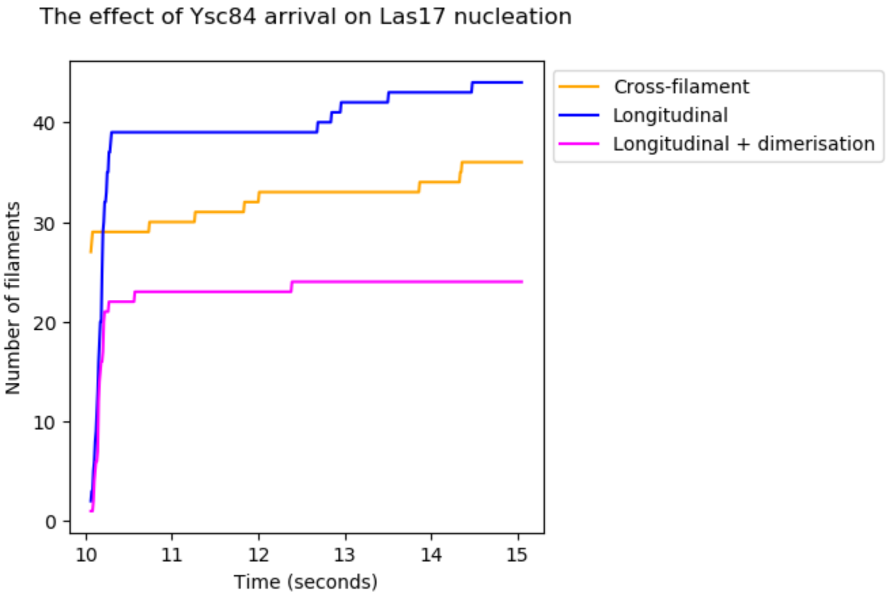


Figure 6.31: Impact of Ysc84 on the endocytic patch. The total number of filaments present at the endocytic patch over five seconds following the recruitment of Ysc84 at t = 10. All simulations presented were run with 100 cloud proteins. The rate of filament production fell ~33% for the cross-filament nucleation scheme with 27 filaments being produced within the first 10 seconds (see **figure 6.30a**) followed by 9 filaments over the five seconds following Ysc84 arrival. On the other hand, mechanisms employing longitudinal nucleation saw a sharp initial increase in the number of filaments followed by a plateauing.

6.10.5 How does the recruitment of Bzz1 change patch behaviour?

The final cloud SH3 to arrive at the patch before Arp2/3 is Bzz1 (Lu, Drubin and Sun, 2016). This protein can interact with Las17 via a close tandem of two SH3 domains highly reminiscent of Sla1-SH3#1-2 (**figure 6.32**). Bzz1 can also dimerise via a C-terminal F-BAR domain. Due to the apparent similarity of the tandem SH3s, a lack of Bzz1 affinity data, and a similar spot array profile to Sla1-SH3#1 (E Allwood, personal communication), we decided to represent both Bzz1 SH3 domains with the same affinity as Sla1-SH3#1. The dimerising F-BAR domain was given an affinity of 0.5 µM in lieu of experimental data as this was a reasonable value for a protein-protein interaction.

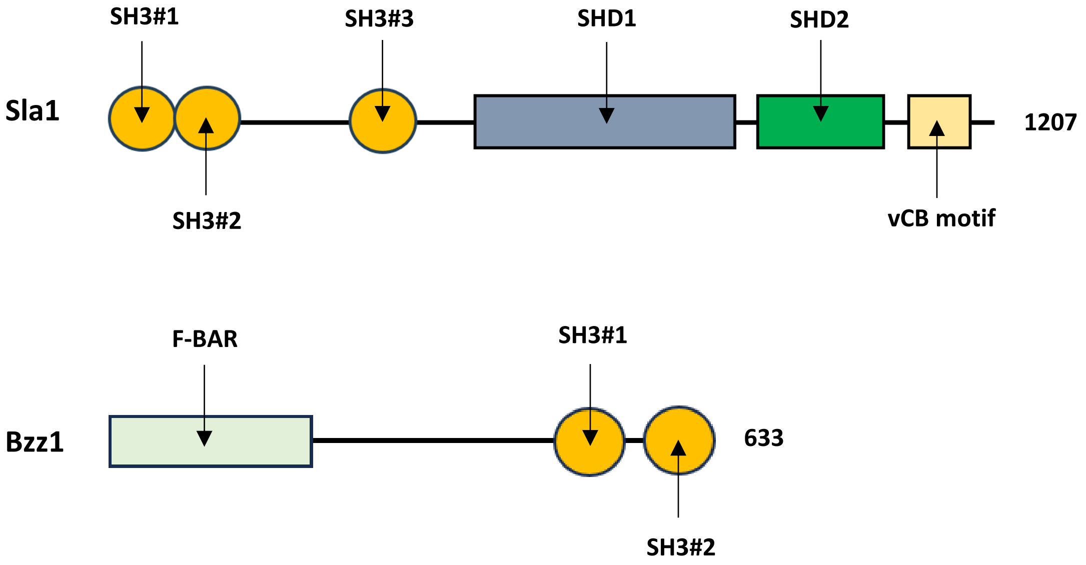


Figure 6.32: Illustrating the similarity in domain structure between Sla1 and Bzz1. Visualising the domain organisation in Bzz1 and Sla1 reveals a similarity in their Las17-binding regions with both domains possessing a close tandem of two SH3 domains (SH3#1 and SH3#2 in both proteins). The residue size of each protein is given to the right of the adjacent protein.

Trajectories files were modified to include 250 Bzz1 molecules using the same method described in the previous section. These modified files were simulated for a further five seconds. The arrival of Bzz1 coincides with increased inhibition of Las17-mediated nucleation across all mechanisms (**figure 6.33a**). The reason for this can be shown by plotting the percentage of actin-binding Las17 tracts occupied by actin over time (**figure 6.33b**). Before the arrival of Bzz1, 22.5% of all PP3 tracts and 36% of all PP4 tracts were actin bound. However, the recruitment of Bzz1 coincided with a significant inhibition over these tracts causing actin occupancy to fall to just 4-5% for both PP5 and PP4. This an almost complete shutdown of Las17’s two highest-affinity actin binding sites.

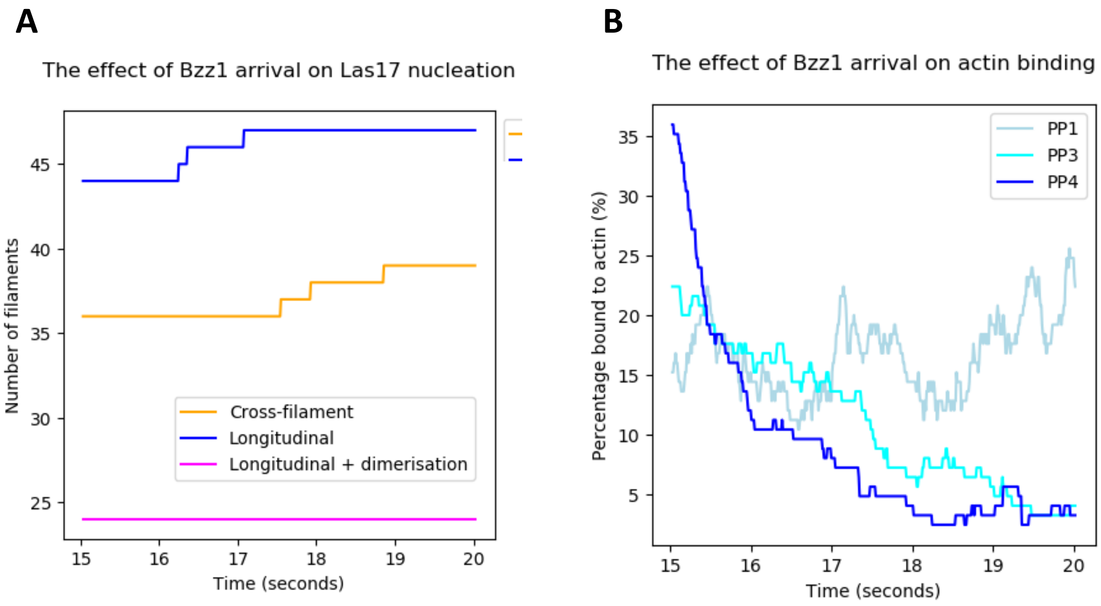


Figure 6.33: Impact of Bzz1 on the endocytic patch. The total number of filaments present at the endocytic patch over five seconds following the recruitment of Bzz1 at t = 15. All simulations presented were run with 100 cloud proteins. **A**) The recruitment of Bzz1 coincides with heavy reduction of filament production across all nucleating schemes. For example, only five new filaments were nucleated in the cross-filament simulation compared to nine filaments following Ysc84 recruitment (both simulated over 5 seconds. See **figure 6.31** for comparison). **B**) The percentage of Las17 tracts bound to actin over time following the recruitment of Bzz1. As can be clearly observed, this recruitment coincides with a significant decrease in the percentage of PP3 and PP4 tracts being bound to actin due to competition with the increasing number of SH3 domains.

6.10.6 Analysing how shifting composition to the cloud may regulate Las17 clustering

We previously showed how Sla1 facilitates the clustering of Las17 – even at high nanomolar concentrations (**section 6.8.2**). This clustering was even identified to have an impact on the degree of inhibition illustrating the importance of Las17 localisation. Therefore, we decided to investigate how the shifting SH3 cloud changes the propensity of Las17 to cluster. This study was undertaken through the analysis of the 0, 10, 15, and 20 second cross-filament nucleation trajectory files following the simulations described thus far and with exposure to 100 cloud SH3s (t = 10 s), five seconds after the arrival of Ysc84 (t = 15 s), and five seconds after the arrival of Bzz1 (t = 20 s).

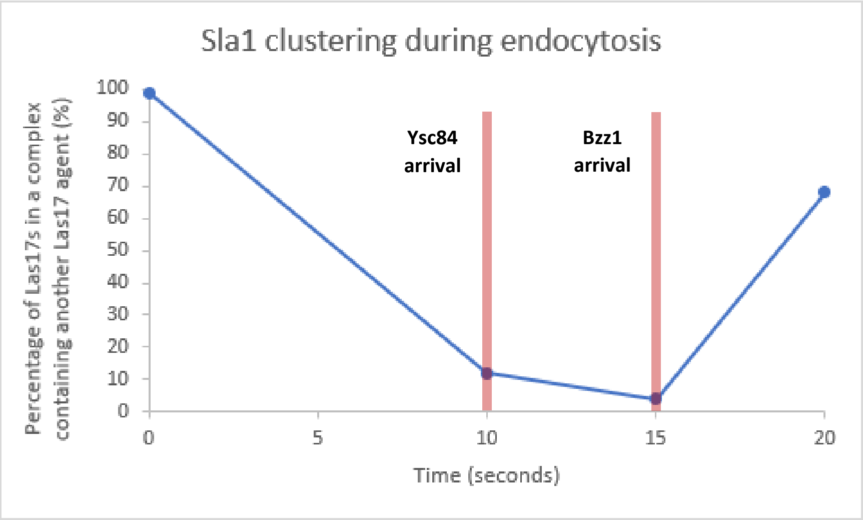


Figure 6.34: How shifting composition of the SH3 cloud affects Las17 clustering. The percentage of Las17s within a complex containing one or more other Las17 agents was collected from the 0, 10, 15, and 20 second trajectory files and plotted over time. The red lines denote when the simulation files were modified to include ysc84 and Bzz1. As shown, competition with cloud SH3s and removal of Sla1 results in a reduction of Las17 clustering as the tandem-binding Sla1 domains were responsible for this effect. Las17 clustering is almost entirely lost by 15 seconds. However, soon returns after the arrival of Bzz1 which can cluster Las17 using its own tandem domains. These domains have a larger separating linker than Sla1-SH3#1-2 so appear to act more like Sla1-SH3#3 with regards to the encouragement of Las17 clusters.

Las17 began the *in vivo* simulation in a highly clustered state with 98% of 300-422 platforms localised in a complex that contains another Las17 platform. The tendency for complexing dropped as Sla1 left the simulation (enhanced by cloud SH3s). Following five seconds contact with Ysc84, almost all Las17 agents were within a unique complex. Recruitment of Bzz1 shifted the localisation profile such that 68% of Las17s were clustered.

6.10.7 What role does the tandem nature of Sla1 play?

The simulations of **section 6.8.1** indicated that the tandem nature of Sla1 is important for a timely departure from Las17 and susceptibility to cloud SH3 domains. We investigated the extent of this effect in an *in vivo* context by replacing tandem Sla1-SH3#1-3 with a single-domain (interacting with the same overall binding profile and affinity). The specifics of this substitution are explained more thoroughly in **section 6.8.1**. Cross-filament nucleation was the mechanism of choice for this simulation as the higher rate of seed formation would provide an output with more observations, and thus confidence. The difference between the nucleation schemes is limited to the post-nucleation phase and thus not anticipated to play a role in SH3-actin competition.

As expected, simulating Sla1 as a single-domain binding unit resulted in the production of fewer actin filaments when compared to tandem Sla1 after the full 20 seconds of simulation time (**figure 6.35a**). This change in Sla1 representation is also accompanied by a reduced peak of seeding immediately following transition from the generic cloud activity phase to the Ysc84 activity phase. The reason behind this can be identified by plotting the degree of Sla1 binding over time (**figure 6.35b**). Showing consistency to our previous *in vitro* modelling results, the rate of Sla1 loss was significantly reduced when substituting tandem Sla1 for a single domain unit. In contrast to the tandem representation, the presence of SH3 cloud domains failed to accelerate the rate of Sla1 departure whilst still contributing to tract inhibition.

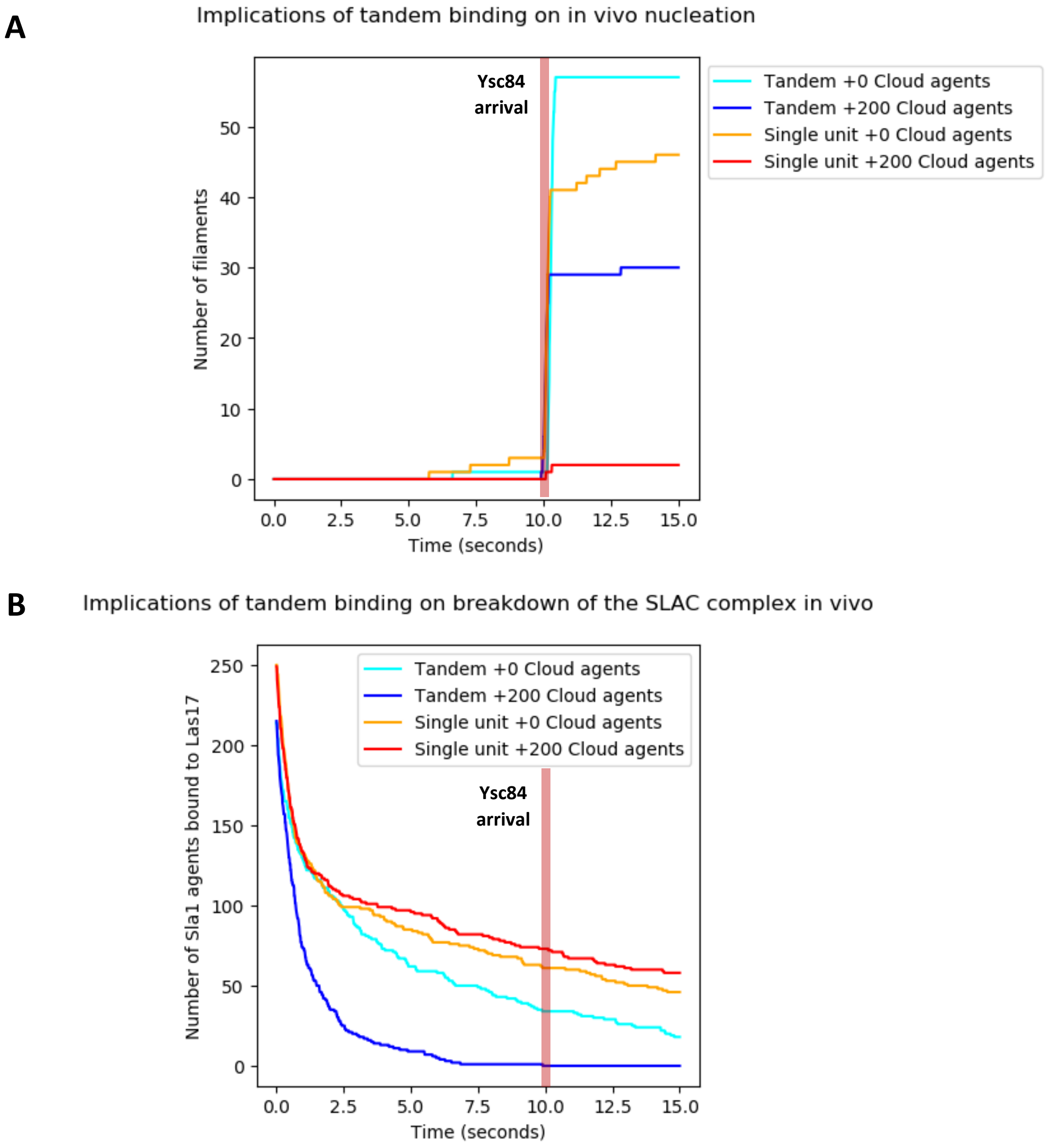


Figure 6.35: Impact of tandem binding on *in vivo* breakdown of the SLAC complex. The *in vivo* longitudinal nucleation simulations were repeated with tandem SH3#1-3 being substituted for single binding unit Sla1 (previously described in **section 6.8.1**). Output files from the 0-10 and 10-15 second simulations were combined to illustrate the change in Las17 activity over time. The red lines denote when the simulation files were modified to include Ysc84. **A**) Single binding unit substitution is associated with a marked reduction in filaments seeded during the Ysc84 recruitment phase. This is particularly apparent in the presence of 200 generic cloud agents where only two filaments were formed over the course of the single unit simulation compared with 30 during the equivalent tandem unit simulation. **B**) Plotting the number of Sla1 agents bound to Las17 over time supports the conclusions of **section 6.8.1** with tandem binding being required for fast breakdown of the SLAC complex. In contrast to tandem binding, adding 200 cloud domains to single unit Sla1 had no positive effect on the rate of Sla1 unbinding whilst still contributing to overall tract inhibition.

6.11 Chapter discussion

6.11.1 The poor experimental binding of Sla1-SH3#2

An accurate binding affinity for the middle SH3 domain of Sla1 could not be obtained through BLI whilst the MST binding curve (**chapter 4, section 4.2.1.7**) was so weak that a complete curve could not be achieved. A lack of sufficient binding for Sla1-SH3#2 could be explained by two possibilities. First, is that the second SH3 domain of Sla1 binds Las17 (300-422) with an affinity too low for accurate observation at the concentrations used. Attempts to concentrate the singly expressed domain post-purification resulted in heavy aggregation of the protein.

A second possibility is that Sla1-SH3#2 cannot fold correctly in the absence of its neighbouring SH3 domain (Sla1-SH3#1). These domains are directly adjacent and so it is reasonable to conclude that that the path to correct folding may require both domains to be present. Supporting evidence can be found with the interdomain interactions predicted using both Homology and AlphaFold modelling (**section 3.1.4**). Furthermore, Agent-Based Modelling suggests that, in order to reproduce the experimentally determined tandem affinity, this domain would have to bind Las17 with an affinity in-between SH3#1 and SH3#3 – both of which gave reproducible binding signals in BLI (**section 3.3.2**). Taken together, and in light of the aggregation prone nature of this domain, we believe that incorrect folding is the more likely explanation for our results.

6.11.2 Characterising the changes between Model Alpha and Model Beta

All three verification simulations conducted as part this chapter concluded that Model Beta functions as expected. Programmatic changes in how Model Beta conducts interactions relative to Model Alpha still resulted in consistent simulations between the pair when both models were executed using the same conditions and movement schemes. Meanwhile, inclusion of movement into Model Beta – excluding the anticipated effect on Sla1 – does not seem to significantly alter simulated observations. This suggests that the movement rate parameters for agents play only a minor role in behaviour. Finally, the inclusion of new parameter values derived from BLI change the rate outputs in a manner consistent with our predictions – those being an enhanced inhibition conferred by Sla1-SH3#1-3 relative to SH3#1-2. This further increased our confidence that Model Beta is behaving as expected.

6.11.3 Elucidating the nucleating mechanism of Las17

Cross-filament nucleation is a major method employed by non-mimetic nucleators (i.e., not Arp2/3) including formins, Leiomodin, and Cobl. However, the sensitivity analysis conducted using Model Alpha identified a markedly lower Las17 nucleation rate than observed experimentally. Fixing actin affinities at the lowest values predicted for PBS/F-buffer whilst decreasing the magnitude of k_{on}/k_{off} rates suggested that this difference cannot be explained by incorrect parameter values alone. This was because the binding rates required to gain a comparable rate were far too low to be reasonable. Another tandem-binding nucleator, Spire, nucleates filaments at a much lower rate than the cross filament nucleator Cobl (Quinlan *et al.*, 2005; Ahuja *et al.*, 2007). This is because Spire instead employs a linear nucleating mechanism whereby only a single sub-filament is stabilised. The difference between these nucleating mechanisms is shown visually in **figure 1.6**. Integrating this into our model decreased the nucleation rate significantly which greatly increased consistency to experimental data.

The results collected throughout **section 6.1** led to the conclusion that Las17 300-422 nucleates actin via three distinct binding sites covering polyproline tracts one, three, and four. The second tract may serve as a SH3-binding platform to confer additional inhibitory control over the tighter-binding middle tract – particularly through interaction with Sla1. The actin binding motifs of this region are largely uncharacterised and thus the use of ODE simulations with MST data marks the first time that individual binding affinities have been elucidated for each motif. This work, by extension, also confirmed the predicted number of binding sites in the region at three and, in conjunction with flexible peptide docking, allows for a predicted binding mechanism. Each actin binding site consists of five consecutive prolines preceded by two arginines 3-6 residues N-terminal of the polyproline tract (from PP1-PP4; RNNRPVPPPPP, RRGPAPPPPP, and RRGPAPPPPP). The highest affinity tract (PP3) positions these arginines 4/5 residues N-terminal – a positioning shared with the second strongest binding site (PP4). Removal of the arginine pairs mostly abolishes interaction with actin (38-125x increase in K_d). Interestingly, the residues separating the second arginine and first proline of the tract all appear to be small and hydrophobic (valine, proline, glycine, and alanine). Flexible peptide docking of PP3 suggests binding analogous to WH2 domains which supports a similar nucleating mechanism between Las17 and other tandem-WH2 nucleators.

The above characterisation of the Las17 300-422 binding sites provides insight into which nucleation scheme (cross-filament or longitudinal) is likely correct. If Las17 nucleates actin using WH2-like binding, then it may utilise the same longitudinal nucleating mechanism as Spire which spaces its WH2 domains with remarkably similar distances to the 300-422 actin-binding tracts (~30 residues). Our ABM supports this conclusion as representing Las17 as a longitudinal nucleator both increased experimental consistency and provided a satisfactory explanation as to why Ysc84 is recruited to the patch shortly following Las17's arrival. When employing a longitudinal nucleating scheme, the actin-binding function of Ysc84 significantly enhances the rate of second sub filament seeding. Recruiting Ysc84 in-between Las17 and Arp2/3 may promote the maturation of actin seeds following the departure of Sla1. This would also reduce the time separating the production of mother filaments and the arrival of the branching nucleator Arp2/3.

6.11.4 The effect of dimerisation

Dimerisation was another mechanism predicted as a result of Model Alpha and subsequently evidenced through the retrospective analysis of experimental data within the Ayscough lab. Introduction of dimerisation into the model addressed two of the experimental and simulation discrepancies: (1) A significantly higher than expected nucleation rate and (2) a lower-than-expected inhibition conferred by Sla1. This is achieved through a reduction in filament formation and an increased sensitivity to SH3 domain inhibition respectively. Allowing Las17 dimerisation within our *in vivo* model likewise increased the nucleator's sensitivity to the changing composition of the endocytic patch – a factor which appears evolutionarily beneficial.

6.11.5 Enhancing our understanding of Sla1 binding

The binding affinities obtained using BLI help confirm a long-standing hypothesis held throughout this thesis: That the individual SH3 domains of Sla1 interact with Las17 weakly (e.g., 7.5 µM for SH3#1), yet can combine in effect to deliver a much higher affinity (39 nM) due to the high avidity conferred as a result of their close spatial positioning. Extending the domain separation in Model Beta resulted in a reduction of Las17 inhibition (**section 6.8.2**). Obtaining affinities for both individual domains and the Sla1 tandem across the same conditions provided better consistency between results. This consistency likely allowed for the much more reasonable affinity revision for SH3#2 (15 µM rather than 1300 µM).

The strong 39 nM affinity calculated from BLI data was shown by Model Beta to maintain significant inhibition over Las17 nucleation – particularly whilst applying Las17 dimerisation. Meanwhile, the hinged seesaw model of binding mediated by the Sla1-SH3#1-2 region allows for a method of accelerated release through SH3 competition which works to reduce the avidity between Sla1 SH3 domains.

6.11.6 Role of the third Sla1 SH3

The Sla1 construct possessing all three SH3 domains (Sla1-SH3#1-3) binds Las17 with an affinity (0.040 µM) ~2-fold higher than the Sla1-SH3#1-2 tandem (0.0877 µM). This is a much smaller affinity jump than what is observed between Sla1-SH3#1-2 and its constituent domains which bind at micromolar affinities. The lower affinity change is likely a reflection of the third domain's relatively weak binding affinity (30 µM) and a significantly larger peptide linker connecting the two binding regions. A larger connecting linker should result in a much lower avidity effect as demonstrated by increasing the maximal length separating SH3#1 and SH3#2 in Model Beta (**section 6.8.2**). This modest increase in binding affinity is surprising, however, given the significantly greater inhibition conferred *in vitro* by the SH3#-1-3 construct relative to the SH3#1-2 expression (E Allwood, personal communication).

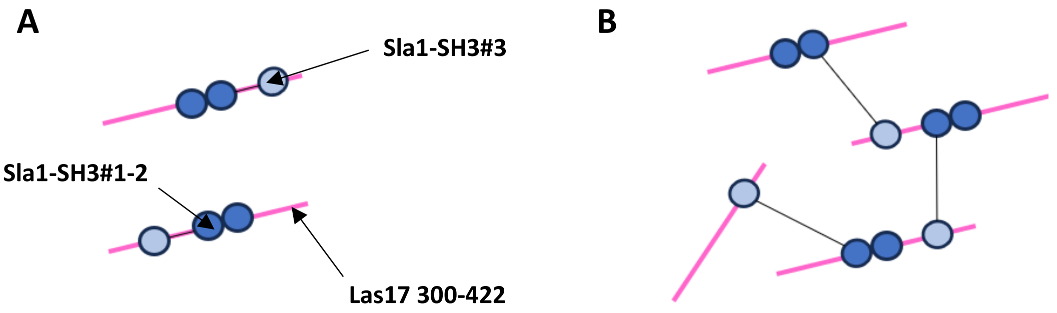


Figure 6.36: How the 3D organisation of Sla1 and Las17 differed from our expectation. A) It was anticipated that Las17 and Sla1 would generally form 1:1 interactions due to high avidity between the SH3 domains and the bound Las17. **B**) However, it was identified that Sla1-SH3#2 (light blue) often bound Las17s distinct from the Sla1-SH3#1-2 region (dark blue) resulting in a cluster of Las17s being connected through a series of Sla1 connections.

This apparent contradiction may be explained by our ABM analysis. Model Beta shows that even at high nanomolar concentrations, the third Sla1 SH3 domain often interacts with Las17 agents distinct to the SH3#1-2 tandem. Therefore, the SH3#1-3 tandem does not always bind using a simple 1:1 interaction (**figure 6.36a**) but may instead readily concatenate Las17 and

Sla1 (**figure 6.36b**). Supporting *in vivo* evidence for a higher-order interaction can be inferred from the SLAC complex which contains multiple copies of both proteins (Feliciano and Di Pietro, 2012). Two interpretations may be gleaned from this. First, Las17 would arrive at the endocytic patch in a pre-concentrated state. If Las17 activity relies upon one or more homo-multimerization events to nucleate efficiently then this may facilitate that process. Second, clustering these proteins increases the relative concentration of Sla1 relative to Las17 which may further increase the degree of inhibition.

By increasing the concentration of Las17 relative to Sla1, SH3#3 (in addition to its tract occluding binding) increases the magnitude of inhibition by a greater degree than its effect on Sla1 binding affinity would initially suggest. Analysing the trajectory files from one of the simulations described in **section 6.8.3** revealed that the majority of Las17 agents were located within just three multi-protein complexes after 0.5 seconds of simulation time (300 nM Las17 and 900 nM Sla1). This results in the formation of Las17-Sla1 "cluster zones" which are areas of concentrated protein (**figure 6.37**). Theoretically, these areas would have a higher concentration of unbound SH3 domains (shown red in the figure) which may increase the likelihood of recapture following unbinding, thereby enhancing inhibition and resistance to competitive binding interactions. This is an example of avidity. Preventing these clusters from forming during simulation pre-equilibration reduced the inhibitory efficacy of Sla1.

Identification of Las17-Sla1 clustering is an example of an emergent phenomenon in ABM and justifies our undirected approach to pre-equilibrate the two proteins. It must be noted that pre-equilibration is not required for clustering, however this can allow for an easier study of its effect. In conclusion, understanding Sla-SH3#3 interactions illustrates how the concept of "fuzzy binding" can extend beyond local interactions and involve higher-order spatial arrangements. The high avidity "hinged seesaw" binding of the first two Sla1 SH3 domains combines with the concatenating effect of the third SH3 to deliver robust inhibition over Las17-mediated nucleation.

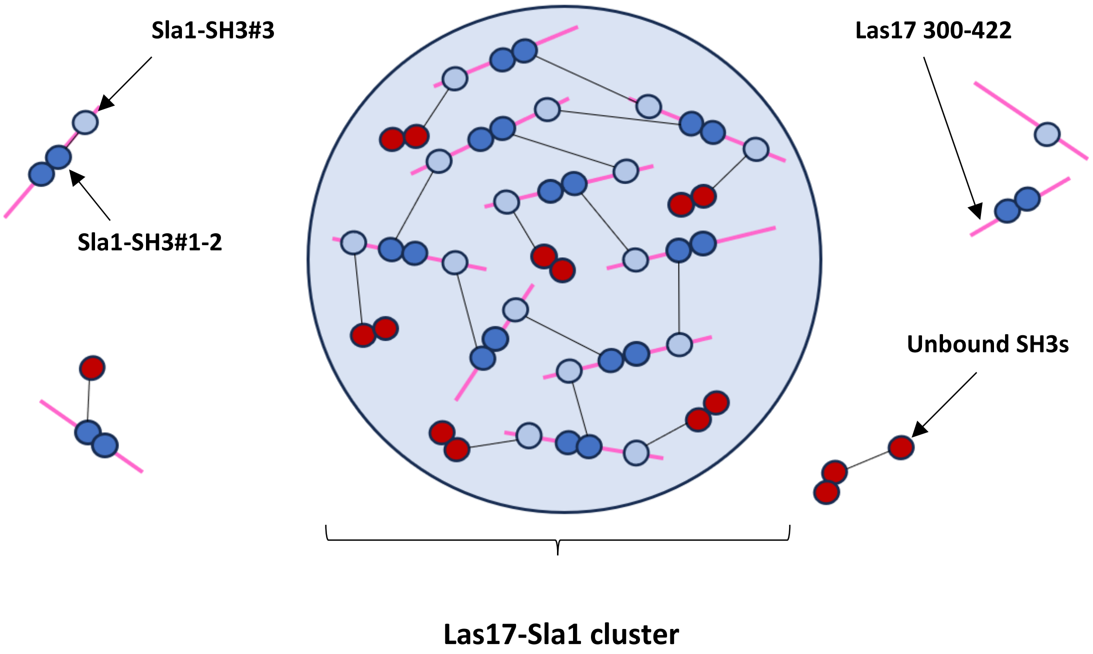


Figure 6.37: Regulatory clusters. The clustering of Sla1 and Las17 results in zones of concentrated protein (shown as a large light blue circle). Within this zone, the concentration of unbound Sla1 relative to Las17 is higher resulting in increased avidity between the proteins and thus inhibition. This can be illustrated by the concentration of unbound SH3 domains (shown in red) within the cluster being higher than the concentration of unbound domains outside the cluster. This would be exacerbated when there is more Sla1 than Las17 as spot array data suggests that both SH3#1 and SH3#3 may each bind two sites on Las17 thereby allowing for more Sla1 in cluster zones and thus a higher concentration of unbound domains (illustrated in the figure).

6.11.7 Understanding how our developing model may function *in vivo*

Each of the individual endocytic phases simulated as part of the *in vivo* analysis (SLAC arrival, exposure to the SH3 cloud, and development of that cloud through Ysc84 and Bzz1 recruitment) were combined to produce a “timecourse” for the production of mother filaments (**figure 6.38**). Combining the data this way reveals how the different mechanisms predicted through our ABM results may regulate Las17 function at the endocytic patch. Cross-filament nucleation results in a more even production of filaments. This is in contrast to longitudinal nucleation in which most filament production occurs as a “burst” directly following Ysc84 recruitment. This may prove evolutionarily advantageous as it would reduce the likelihood of mother-filaments either diffusing from the endocytic patch or partaking in unfavourable binding interactions. Constraining Las17-mediated nucleation to a narrow activity window would also limit the number of linearly nucleated filaments and thus encourage a greater degree of branching in the endocytic actin network. Arrival of Bzz1 further enhanced the competitive power of the SH3 cloud which further transitioned Las17 back into an inhibited state to close the activity window. Transition between the endocytic phases was particularly pronounced when using the Las17 dimerisation model demonstrating how oligomerisation could evolutionarily benefit the system.

SH3 competition was shown to enhance the departure rate of Sla1 across all *in vivo* simulations. Even though this enhanced removal of Sla1 did not confer an increase in activity, the rate of seed production was fairly stable at cloud SH3 numbers >100. A timely breakdown of this interaction would reduce unnecessary retention of Sla1 in a location non-conducive to its cargo binding function.

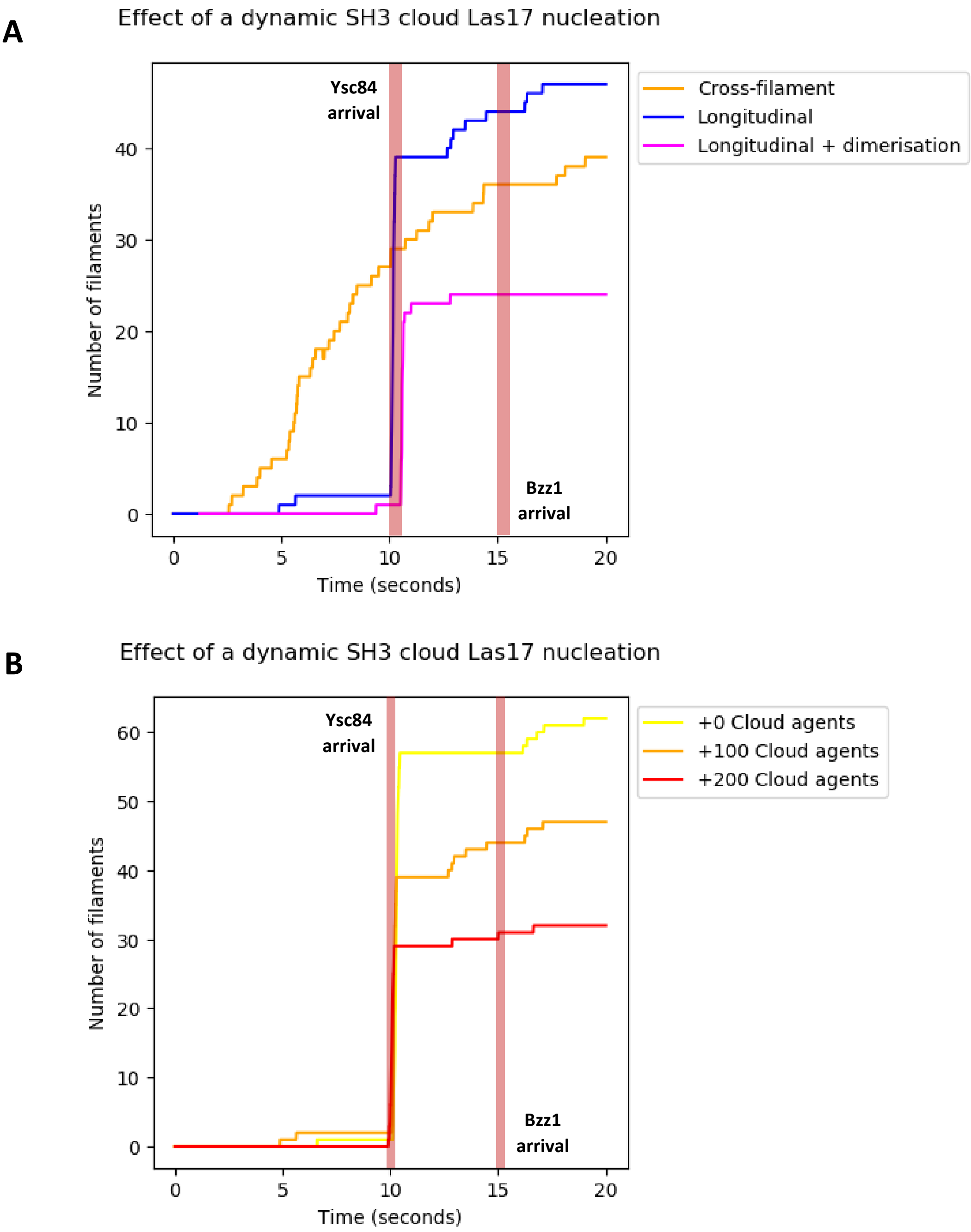


Figure 6.38: How Las17 activity is affected by the developing SH3 cloud. Output files from the 0-10, 10-15, and 15-20 second simulations were combined to illustrate the change in Las17 activity over time. The red lines denote when the simulation files were modified to include Ysc84 and Bzz1. **A**) The cross-filament model reveals that the rate of Las17-mediated nucleation of actin falls progressively following the introduction of Ysc84 and Bzz1. Meanwhile, Ysc84 recruitment during the longitudinal schemes coincides with a spike in actin filament production as the pool of longitudinal nuclei was rapidly seeded. The transition between activity phases was most pronounced when dimerisation was included. **B**) The plotting from the previous subfigure was repeated using an increasing number of Cloud SH3s which shows how cloud can regulate actin activity.

7. Final Discussion

7.1 Characterising the Las17 mechanisms predicted by ABM

Comparing our ABM results with experimental data has revealed potential mechanistic inaccuracies with our previous model of Las17-mediated actin nucleation as implemented in Model Alpha. First is that Las17 likely nucleates actin through stabilisation of a longitudinal trimer. Replacing the cross-filament nucleation scheme with longitudinal nucleation significantly increased parity with experimental observations. This prediction was subsequently supported via our characterisation of Las17 binding using 300-422 mutants in MST. We confirmed that G-actin binding in this region predominantly occurs at three sites (the first, third and fourth polyproline tracts). The centres of these three sites are spaced almost identically when compared to the WH2 domain spacing of the longitudinal nucleator Spire. In additional support of this, flexible peptide docking suggests that the Las17 PP4 tract may bind using a location and orientation overlapping the Spire WH2 domains. Taken together, we believe this is strong evidence that Las17 employs a linear nucleation mechanism similar to Spire.

The second mechanism predicted as a result of this modelling was Las17 oligomerisation. MST stoichiometry and size exclusion data were retrospectively analysed given the observed discrepancies in polymerisation rates. This experimental evidence suggests that multiple Las17 peptides may coordinate during actin binding to bind actin monomers. Introducing this scheme into our agent-based model increased the sensitivity of Las17 to regulatory SH3 interactions. This should be evolutionarily beneficial to cells and thus appears a reasonable mechanism, however further experimental work would be needed to deduce the nature of Las17 oligomerisation. An interesting note is that dimerisation and longitudinal nucleation interact in effect to produce biphasic actin polymerisation – a profile that has often been observed in Las17 pyrene assays. Experimental biphasic responses can vary in both their magnitude and profile, whilst changing the dimerisation affinity in Model Beta produces similar changes. Hsp70 may partially be responsible for this unpredictable behaviour. Variability in chaperone concentrations between Las17 samples could influence the propensity of Las17 to dimerise through its substrate binding function. This prediction could be experimentally tested by combining Las17 with varying concentrations of Hsp70.

7.2 Why would Las17 nucleate via a less efficient mechanism?

Our conclusion that Las17 is likely a longitudinal nucleator raises the interesting question: Why would evolution select for longitudinal nucleating mechanisms when cross-filament nucleation is significantly faster in seeding filaments? By discovering that Las17 likely fits into the class of longitudinal nucleators and analysing how our ABM predicts the effect of regulatory proteins, we propose two possible benefits of this arrangement.

F-actin networks have functionally important internal structure. For example, “curly” (the N-terminal region of IQGAP proteins) can bind to F-actin to encourage bending of membrane-associated filaments. Filament bending is an important feature of the contractile actomyosin ring network as the high persistence length of actin filaments likely necessitates the use of actin-binding proteins to maintain structural integrity during constriction of the ring (Palani *et al.*, 2021). The nucleators responsible for generating networks also have a profound effect on how the F-actin is structured. If cells require force-generating branched networks during endocytosis, then this can be achieved through the Arp2/3 nucleator. Different F-actin architectures are associated with different mechanical properties and preferences for binding proteins to best impart their intended effect. Therefore, an important consideration when promoting a new F-actin network is what nucleator(s) predominantly influence the architecture.

Three of the longitudinal nucleators identified thus far (Spire, JMY, and Las17) all function in concert with more efficient nucleating/elongating machinery. Spire synergises with the formin Cappuccino to form actin meshes during the establishment of oocyte polarity (Quinlan, 2013). Meanwhile, both JMY and Las17 interact with the Arp2/3 (Zuchero *et al.*, 2009; Urbanek *et al.*, 2013). In all three examples, the longitudinal nucleator is hypothesised to produce a limited, initial F-actin network that is greatly expanded upon by another nucleator. Therefore, the lower nucleation efficiency of longitudinal nucleators may allow them to seed new F-actin networks without significantly defining their architecture. Controlling the dominant architecture of a network would play an important role in the downstream recruitment of actin-binding proteins which frequently show an architectural preference. For example, ADF/cofilin shows a greater binding propensity for actin cables than branched networks (Gressin *et al.*, 2015). Las17 perfectly encapsulates this balance. The WASp homologue can nucleate linear filaments, while the endocytic actin-network requires a branched architecture. Here, the lower nucleation rate of Las17 allows for the production of linear mother filaments for Arp2/3 without directly defining the overall architecture – in this case, a branched network. This is in direct contrast to the cross-filament nucleator Cobl which functions as a dominant nucleator (possibly with aid from the homologue Cobl-like) at dendritic branch induction sites (Izadi *et al.*, 2021).

Another benefit for employing a longitudinal nucleation scheme is that the rate-limiting process of second sub filament seeding can be modulated through regulatory interactions. It has been suggested that dimerisation of Spire may promote nucleation. This may allow the two sub filaments nucleated by the Spire subunits to interact and seed a two stranded actin filament (Sitar *et al.*, 2011; Quinlan, 2013). This provides cells with an additional regulatory mechanism. Promoting the clustering of Spire may provide cells with a nucleation "trigger". Likewise, our ABM suggests that recruitment of the actin-binding protein Ysc84 may serve a similar role during endocytosis. Our *in vivo* modelling shows that, when applying a longitudinal nucleation scheme to Las17, the recruitment of this protein is associated with a "burst" of filament formation via a significant enhancement to the rate of second subunit seeding. This prediction is supported by preliminary data from the Ayscough lab which suggests that Ysc84 promotes the nucleation of Las17.

7.3 Regulatory cloud hypothesis

The results produced from this thesis – particularly those obtained from Model Beta – help to provide a deeper insight into how the SH3 domains that are present during endocytosis may help to regulate the production of mother filaments via Las17. The functions of the regulatory cloud can be broken down into four main stages.

7.3.1 Stage 1: The SLAC complex

First, Sla1 maintains robust inhibitory control over Las17 (**figure 7.1**). Model Beta and BLI both predict that the close spatial orientation of the first and second SH3 domain allows for a high avidity interaction that can confer tight control over the central nucleating tract. Our MST data shows that this tract has the highest affinity for actin in the 300-422 nucleating region. Therefore, robust binding over this tract would likely have the largest inhibitory impact on Las17.

Allowing Sla1 and Las17 to interact in our ABM in the absence of actin predicted that even nanomolar concentrations of both proteins encouraged clustering where the third SH3 domain would frequently bind distinct Las17 peptides other than the SH3#1-2 tandem. These clusters appeared to confer additional inhibition through colocalising the proteins. Colocalisation is expected to increase the relative concentration of each protein with respect to one another and thus allow for an increased occupancy of the Las17 tracts by Sla1 through avidity. This may explain why the cytosolic SLAC complex was identified to contain multiple copies of both Sla1/Las17 and why Sla1-SH3#1-3 confers significantly more inhibition than SH3#1-2 despite there only being a 2-fold difference in affinity. Ubiquitin – a common membrane signal for CME – interacts with Sla1-SH3#3 (Stamenova *et al.*, 2007). Binding competition between these proteins may play a role in triggering the Las17 regulatory switch by breaking apart colocalisation and encouraging Sla1 and Las17 to occupy different regions of the endocytic patch (following unbinding of the SH3#1-2 tandem). Additional masking of Sla1 via its cargo interactions may also help reinforce this separation.

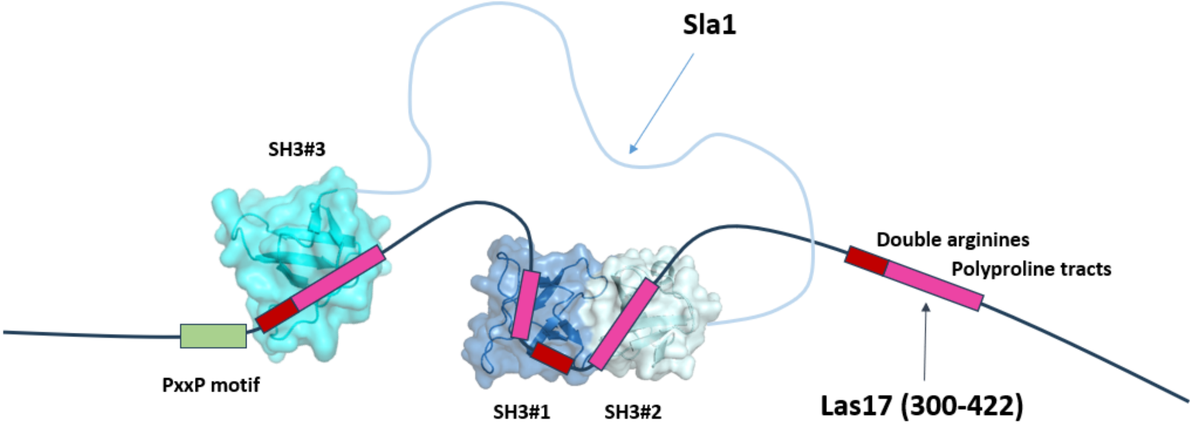


Figure 7.1: Visualising the SLAC complex. Each domain in Sla1 is illustrated using PDB files edited with PyMOL v4.6 (AlphaFold AF-Q4P3H6-F1 for SH3#1-2 and PDB: 1Z9Z for SH3#3). Las17 regions are denoted with coloured boxes. This includes magenta for PP-tracts, red for actin-binding double arginines, and green for the PxxP motif not affiliated with a polyproline tract. Sla1-SH3#1-2 bind polyproline tracts 2 and 3 to maintain a high avidity interaction with a strong K_d (**section 6.3.2.4**). Meanwhile, SH3#3 can bind to either the first or fourth tract. This can either be the same Las17 peptide as SH3#1-2 (shown above) or another Las17 which facilitates in Sla1/Las17 clustering (**section 6.11.6**). High avidity binding of the Las17 polyproline tracts can mask the binding of actin via physically occluding the overlapping G-actin binding motifs and thereby maintain robust inhibition of Las17's nucleating ability in the 300-422 region.

7.3.2 Stage 2: Initial exposure of Las17 to the SH3 cloud

Las17 interacts with over 11 endocytic SH3 domains via its polyproline tracts, with many of these interactions being localised to the 300-422 region (Hummel and Kaksonen, 2023; **section 4.2.3**). These result in numerous competing interactions between domains. We refer to this pool of binders as the "SH3 cloud". At the time of Las17/Sla1 arrival, the composition of this cloud would largely be dominated by Lsb3 which peaks in concentration at the patch prior to the arrival of the SLAC complex. However, the SH3 cloud will still contain a small concentration of other proteins which either arrived at the early stage of their recruitment profile or via general exposure to the cytosol.

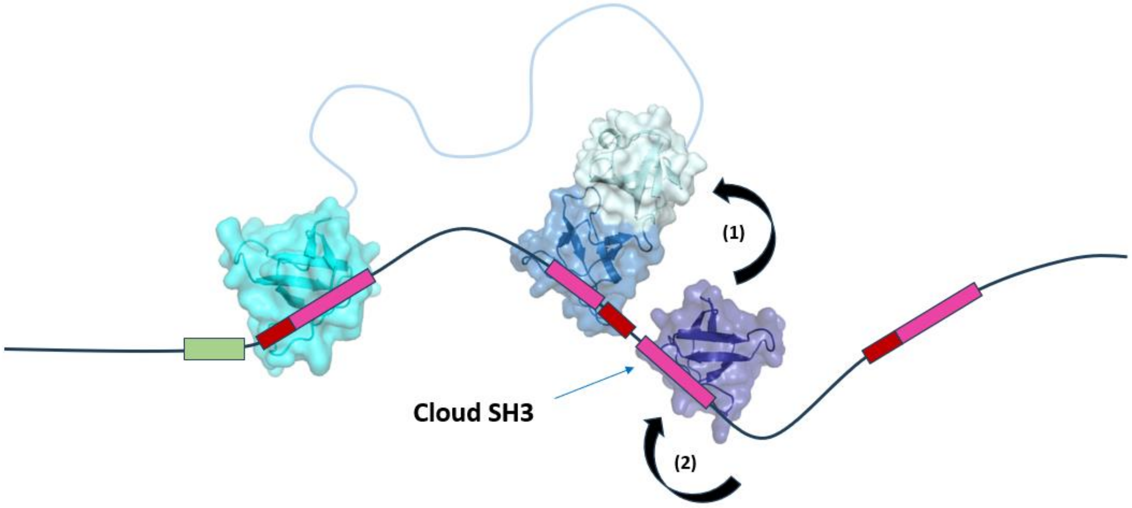


Figure 7.2: Visualising the SH3 competition. Each SH3 domain is illustrated using PDB files edited with PyMOL v4.6 (AlphaFold AF-Q4P3H6-F1 for Sla1-SH3#1-2, PDB: 1Z9Z for Sla1-SH3#3 and PDB: 2A08 for Ysc84 which, in this figure, represents the cloud SH3). Despite the high overall affinity of the SH3#1-2 tandem, the K_d for each individual domain is relatively weak resulting in periods of time where one of the tandem domains is unbound (shown with (1)) (**section 4.5.2**). This leaves the tandem interaction susceptible to competition from the SH3 cloud. Competing SH3 domains are able to bind one of the Las17 tracts during these transitory periods of domain unbinding (Shown with (2)). Consequently, the high avidity conferred by the close spatial positioning of the Sla1 SH3 domains is broken down and the tandem is forced to rely on a weak affinity interaction with a high k_{off}. This facilitates breakdown of the SLAC complex and thus hastens the separation of Sla1 and Las17 at the endocytic patch.

Evidence suggests that Sla1 occupies a distinct zone of the endocytic patch from Las17 with the former occupying a more central position as Sla1 cargo-binding interactions effectively define the centre of the patch. While the binding of ubiquitin to Sla1-SH3#3 may help to outcompete Sla1 from Las17 and aid in its spatial sequestering, a complete separation of the two proteins would still require breakdown of the tandem SH3#1-2 interaction with PP2/PP3. Both of our ABM models show that generalised cloud SH3 interactions can help to facilitate the breakdown of this high avidity interaction through competition with the individual SH3 domains. Both Sla1-SH3#1 and SH3#2 interact with Las17 with relatively weak K_ds of several micromolar and this often alternates between both SH3s bound, and only one bound (interaction (1) in **figure 7.2**) in accordance with the “hinged seesaw” visualisation given in **chapter 6**. The occupancy of a Sla1 binding region during a period of unbinding results in the tandem-Las17 interaction being maintained through a single, relatively weak interaction with a high k_{off} rate (interaction (2) in **figure 7.2**). This removes the effect of avidity and thus facilitates a more rapid departure of Sla1. Facilitating the breakdown of this interaction is expected to be important both for allowing Sla1 to localise to the inner cargo-binding region of the endocytic patch and for leaving the central actin-binding tract of Las17 300-422 open to actin binding (**figure 7.3**). SH3 competition is expected to play a large role in ensuring this timely release from Las17.

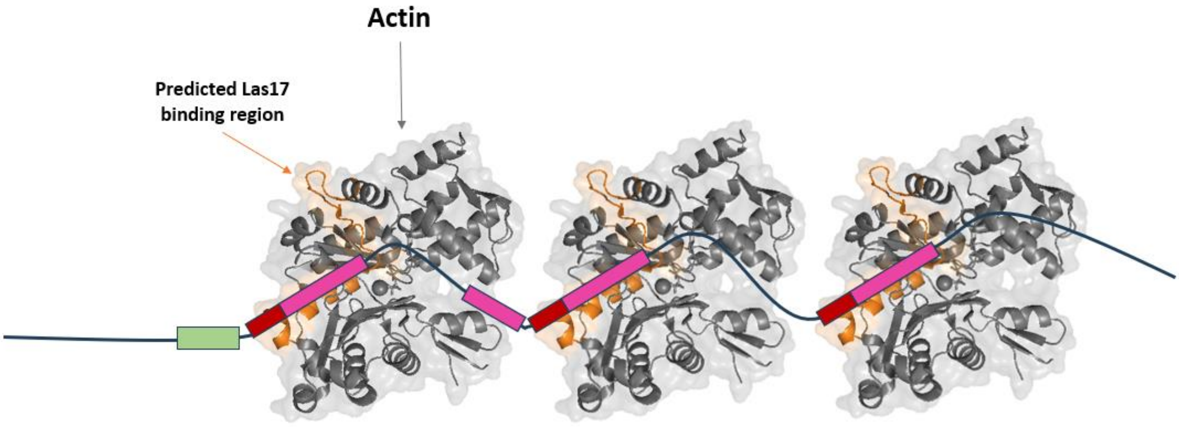


Figure 7.3: Visualising Las17 nucleation. Actin is illustrated using a PDB file of G-actin edited with PyMOL v4.6 (PDB: 3MN5). The helical rich region predicted to constitute part of the interaction region with Las17 is shown in orange (based on **section 6.1.4**). Cloud domains control the rate of nucleation via competition with actin following the departure of Sla1. However, single SH3 interactions do not confer as robust control as the tandem Sla1 SH3s allowing actin to bind the PP1, PP3, and PP4 to nucleate actin. Nucleation occurs via a longitudinal scheme wherein the actin subunits are arranged “end-to-end” to stabilise a single sub filament. However, the second sub filament must rely on a weak lateral interaction that significantly reduces the rate of filament maturation (seeding).

Ultimately, the promiscuous binding of Las17 tracts facilitates breakdown of the Las17-Sla1 inhibitory interaction through binding competition with the Sla1-SH3#1-2 tandem which reduces the gain in effective Sla1 affinity through avidity. Our work suggests that the reason why Las17 has so many SH3-binding interactions in the nucleating region is that this allows for timely release of Sla1 at the endocytic patch whilst also limiting and controlling the rate of actin nucleation. A slower rate of Sla1 dissociation would significantly impact downstream processes which may increase both the energetic costs and timescales for endocytic events.

7.3.3 Stage 3: The arrival of Ysc84

Ysc84 arrives at the endocytic patch in between the breakdown of the SLAC complex and the arrival of Arp2/3. The protein is likely recruited as a consequence of Las17 exposure following departure of Sla1, thus arrival of the protein depends upon a fast breakdown of the SLAC complex. Our SH3 binding location analysis suggests that the SH3 domain of this protein binds weakly to every polyproline tract in the 300-422 region, but preferentially binds to a PxxP motif N-terminal of the first nucleating tract. Interestingly, this tract is not predicted to bind any of the other SH3 domains analysed as part of this thesis leaving the site relatively uncontested for Ysc84 binding. This may reflect the unique role played by this protein. Unlike other SH3-proteins recruited to the endocytic patch during the timeframe analysed, Ysc84 confers a directly positive effect on Las17-mediated nucleation. Ysc84 was observed to supply G-actin to the 300-422 nucleating region via its actin-binding YAB domain throughout our simulations. This nucleation-promoting effect was confirmed using preliminary experimental data obtained by K. Ayscough.

If Las17 functions as a longitudinal, rather than a cross-filament, nucleator as predicted by this thesis then this enhancing effect becomes particularly pronounced. Here, the recruitment of Ysc84 effectively turns Las17 into a cross-filament nucleator by supplying G-actin to the nascent nuclei during second sub filament seeding (**figure 7.4**). This function is supported by the relatively large peptide linker separating the centre of PP1 with the YAB domain being over twice the distance separating the last two WH2 domains of the cross-filament nucleator Cobl. A distance this large would allow the YAB domain to position itself in an orientation suitable for seeding the second sub filament. *In vivo* simulations with Model Beta predict that the arrival of Ysc84 coincides with a "burst" in filament formation as many of the linear actin nuclei held by Las17 are seeded simultaneously with the arrival of Ysc84. This synchronisation could play a role in ensuring a limited time frame between filament formation and Arp2/3 arrival. However, it must be noted that this predicted burst would not be as pronounced as shown by our simulations as Ysc84 arrives at the endocytic patch over the course of several seconds. Nevertheless, a significant increase in filament formation would be expected during this recruitment process.

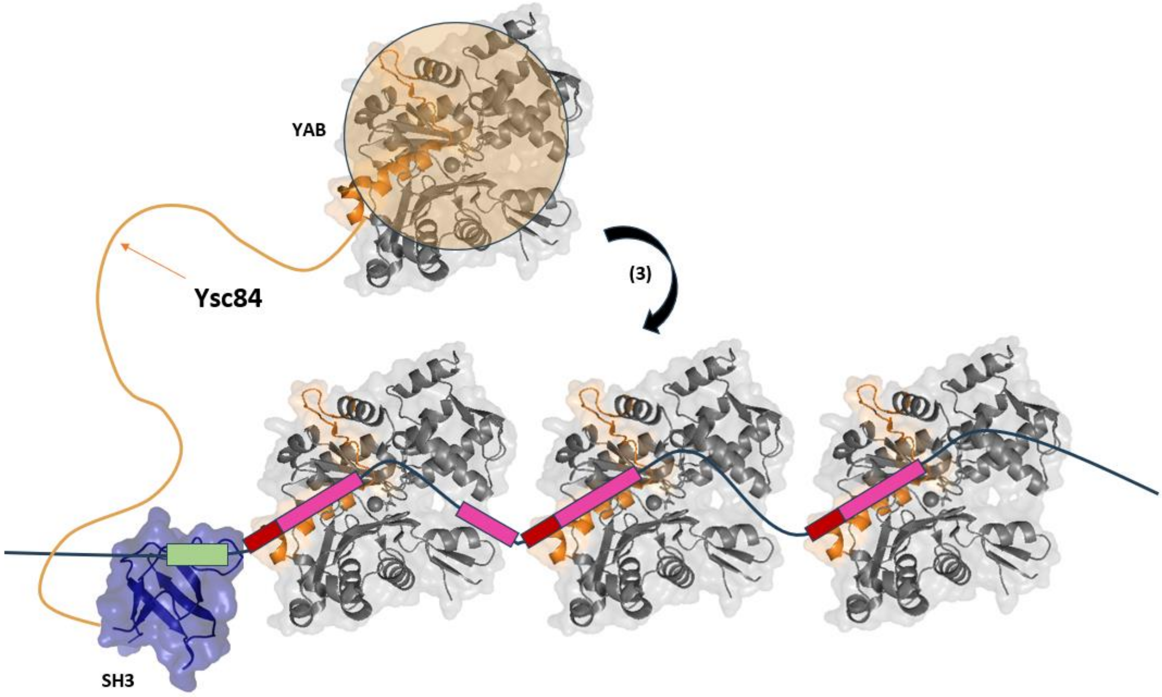


Figure 7.4: Visualising the role of Ysc84. Actin and the Ysc84 domains are illustrated using PDB files edited with PyMOL v4.6 (PDB: 3MN5 for G-actin and PDB: 2A08 for Ysc84-SH3). The YAB domain is shown as a transparent orange circle due to a lack of structural data regarding the YAB/actin interaction. The weak lateral interaction that must occur to seed the second actin sub filament can be enhanced by Ysc84 which arrives at the patch shortly after Las17 (Shown with (3)). The linker connecting the YAB domain to Las17 (via an SH3) functionally resembles the larger separation of actin-binding domains identified within the cross-filament nucleator Cobl (**section 6.9**).

7.3.4 Stage 4: Further changes in SH3 cloud composition

Bzz1, a tandem-SH3 protein which is predicted to bind Las17 through both the second and third actin-binding tracts, arrives at the endocytic patch immediately preceding Arp2/3. This is in turn followed by the recruitment of further SH3-domain proteins including Myo5 and Rvs167 (Lu, Drubin and Sun, 2016). Each of these increases the concentration of binding domains within the SH3 cloud which may serve as a “hand brake” on the production of actin mother filaments. This may help to limit the production of linear filaments such that the endocytic actin network has a greater degree of branching. Our *in vivo* simulations suggest that even a 2-fold increase to the SH3 cloud would decrease the nucleating activity of Las17 significantly (although the specifics of this cloud composition change would depend upon the binding locations of recruited proteins). Bzz1 serves as an especially potent inhibitor as shown by the reduction in access of actin to tracts 3 and 4. This is due in part to the same reason that makes the Sla1-SH3#1-2 tandem region effective: the high avidity between the spatially close SH3 domains (**figure 7.5**). Dimerisation of Bzz1 through its BAR domain may further increase the avidity effect through colocalising itself and Las17. Dimerisation of the Las17 WCA region is known to increase the activation of Arp2/3 and thus clustering of Las17 by Bzz1 may promote its NPF activity (Padrick *et al.*, 2008). Mother filaments nucleated within the 300-422 region may also remain bound to Las17 through both the F-actin-binding C-terminal polyproline tracts and Ysc84.

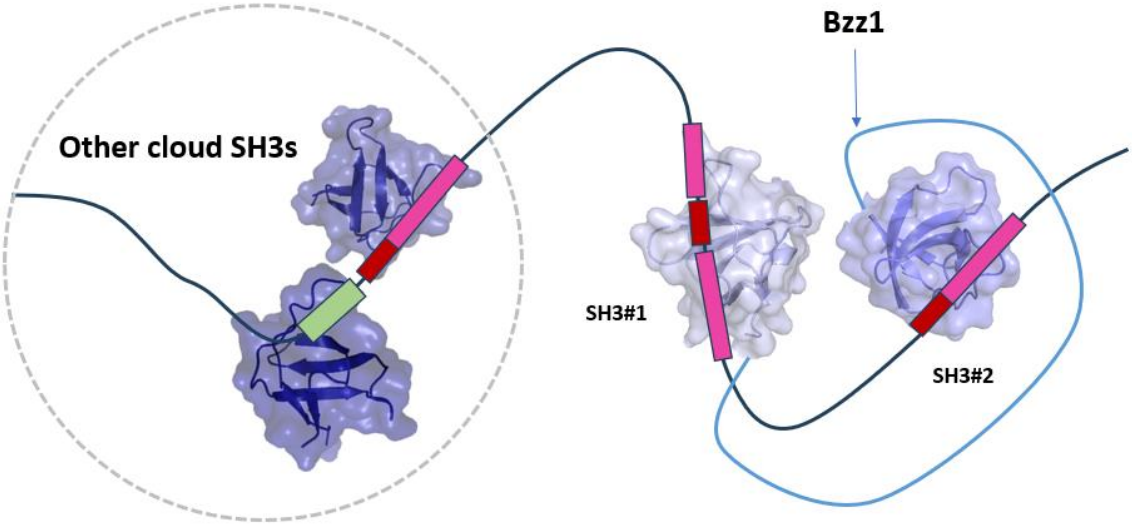


Figure 7.5: Visualising the role of further cloud recruitment. Each SH3 domain is illustrated using PDB files edited with PyMOL v4.6 (AlphaFold AF-A0A161HKK2-F1-model_v4 for Bzz1-SH3#1-2 and PDB: 2A08 for Ysc84 which, in this figure, represents the cloud SH3s). As endocytosis progresses, more SH3 domains arrive at the patch including Myo5, Rvs167, and Bzz1 (Lu, Drubin and Sun, 2016). These domains help to halt nucleation by "locking up" the actin binding tracts by physically masking them. Bzz1 is able to maintain a particularly strong inhibition via its tandem of SH3s that, reminiscent of Sla1, maintain robust binding through avidity.

7.4 Expanding our understanding of Fuzzy binding and actin networks

Substituting Sla1-SH3#1-3 in Model Beta for a single binding domain with a similar overall binding affinity reveals the benefits of a regulatory system that employs tandem binding sites. Making this substitution prevented the enhanced displacement of Sla1 via competition with the SH3 cloud which our ABM predicted to significantly increase residency (and thus inhibition) at the Las17 nucleating region following arrival to the endocytic patch. Using tandem binding gives cells an additional tool for regulating the activity of Las17. Competing interactions that reduce the effect of avidity allow the system to be more dynamic by enhancing the rate of transition between regulatory stages. This embodies the emerging concept of fuzzy binding and demonstrates the mechanisms that govern this principle (Williamson, 2023). Ultimately, the characterisation of these mechanisms helps to answer the question proposed at the start of this thesis: Why so many tandem binding sites?

Our conclusions here not only expand our understanding of the Las17 regulatory system but may also shed light upon other systems regulated through tandem binding events. Many other biological systems utilise tandem binding for high avidity interactions. For example, the yeast Ste20 kinase regulator Bem1 is another example of a protein which interacts through a tandem pair of SH3 domains (Winters and Pryciak, 2005). Furthermore, three proteins (Las17, Leiomodin, and SCAR/WAVE) may all bind actin directly through a polyproline tract (Chereau *et al.*, 2008; Buracco *et al.*, 2022). Nucleation via direct actin-binding polyproline regions is only recently being unearthed. These interactions have yet to be fully characterised with the first paper demonstrating actin nucleation via the WAVE polyproline binding region only being available in preprint. Considering the sequence overlap between polyproline actin binding sites and the core SH3-binding PxxP motif, similar regulatory models may still remain undiscovered.

As explained in **chapter 1**, the regulation of actin networks is highly complex with a high degree of interconnectivity and functional redundancy between processes. A good example being that the knockout of WAVE in *Dictyostelium* results in WASp relocating to the pseudopodia to assume much of the original WAVE functionality (Veltman *et al.*, 2012). Both Las17 and WAVE belong to the WASP-family of proteins which has been linked to both human hematopoietic malignancies and immune deficiencies (Jin *et al.*, 2004; Biber *et al.*, 2021). Therefore, a better characterisation of Las17 regulation may advance our understanding of these clinically relevant areas.

7.6 Future aims and directions

An important component of research is to consider its direction. Agent-based modelling can never perfectly replicate a biological system as models require assumptions and likely incorporate experimental errors. Instead, ABMs allow us to assess the veracity of our hypotheses and expand our understanding of phenomena which emerge from our current knowledge. These conclusions then – like the work presented within this thesis – suggest avenues for experimental research. Several of these future directions are outlined below:

Further characterisation of the Las17 polymerisation profile may help to characterise the predicted oligomeric function of the polyproline nucleating region. The biphasic nature of 300-422 pyrene curves was reproduced using our ABM dimerisation model wherein the affinity of cooperative dimerisation had an observable impact on the first polymerisation phase. This phase is highly variable experimentally and may be explained by the variable levels of Hsp70 present between protein preparations. Therefore, adding the chaperone to pyrene assays at varying concentrations may help to test this prediction. Further experimental assays aimed to validate and explore oligomerisation mechanisms would complement this potential work.

Our ABM analysis has shown that polymerisation assays can yield contradictory results to assays which observe nucleation – particularly when taking sequestration of G-actin into account. Further analysis of Las17 nucleation with methods that focus more on direct filament number measurements may serve this role. One such example is Total internal reflection fluorescence (TIRF) microscopy which was unfortunately dropped as a potential experimental aspect of this thesis due to the COVID pandemic.

Further investigation of Ysc84 and Bzz1 would help to better refine our *in vivo* model allowing for greater exploration of the evolving SH3 cloud. Model Beta predicts that both endocytic proteins may help to define an "activity window" for Las17 following arrival of the SLAC complex to the patch. A key aim of this avenue would be to better understand the actin binding behaviour of Ysc84. *In vivo* mutations would be a great way to further develop Model Beta whilst testing the predicted functions of different endocytic components.

8. Bibliography

Adzhubei, A. A., Sternberg, M. J. E. and Makarov, A. A. (2013) ‘Polyproline-II helix in proteins: structure and function’, *Journal of Molecular Biology*, 425(12), pp. 2100–2132. doi: 10.1016/j.jmb.2013.03.018.

Aghamohammadzadeh, S. and Ayscough, K. R. (2009) ‘Differential requirements for actin during yeast and mammalian endocytosis’, *Nature Cell Biology*, 11(8), pp. 1039–1042. doi: 10.1038/ncb1918.

Ahmad, M. and Helms, V. (2009) ‘How do proteins associate? A lesson from SH3 domain’, *Chemistry Central Journal*, 3(1), p. O22. doi: 10.1186/1752-153X-3-S1-O22.

Ahuja, R., Pinyol, R., Reichenbach, N., Custer, L., Klingensmith, J., Kessels, M. M. and Qualmann, B. (2007) ‘Cordon-bleu is an actin nucleation factor and controls neuronal morphology’, *Cell*, 131(2), pp. 337–350. doi: 10.1016/j.cell.2007.08.030.

Akamatsu, M., Vasan, R., Serwas, D., Ferrin, M. A., Rangamani, P. and Drubin, D. G. (2020) ‘Principles of self-organization and load adaptation by the actin cytoskeleton during clathrin-mediated endocytosis’, *eLife*, 9, p. e49840. doi: 10.7554/eLife.49840.

Allwood, E. G., Tyler, J. J., Urbanek, A. N., Smaczynska-de Rooij, I. I. and Ayscough, K. R. (2016) ‘Elucidating Key Motifs Required for Arp2/3-Dependent and Independent Actin Nucleation by Las17/WASP’, *PloS One*, 11(9), p. e0163177. doi: 10.1371/journal.pone.0163177.

Ambrose, A. J. and Chapman, E. (2021) ‘Function, Therapeutic Potential, and Inhibition of Hsp70 Chaperones’, *Journal of Medicinal Chemistry*, 64(11), pp. 7060–7082. doi: 10.1021/acs.jmedchem.0c02091.

An, G. (2008) ‘Introduction of an agent-based multi-scale modular architecture for dynamic knowledge representation of acute inflammation’, *Theoretical Biology & Medical Modelling*, 5, p. 11. doi: 10.1186/1742-4682-5-11.

An, G., Mi, Q., Dutta-Moscato, J. and Vodovotz, Y. (2009) ‘Agent-based models in translational systems biology’, *Wiley interdisciplinary reviews. Systems biology and medicine*, 1(2), pp. 159–171. doi: 10.1002/wsbm.45.

Azimi, M., Jamali, Y. and Mofrad, M. R. K. (2011) ‘Accounting for diffusion in agent based models of reaction-diffusion systems with application to cytoskeletal diffusion’, *PloS One*, 6(9), p. e25306. doi: 10.1371/journal.pone.0025306.

Barker, S. L., Lee, L., Pierce, B. D., Maldonado-Báez, L., Drubin, D. G. and Wendland, B. (2007) ‘Interaction of the endocytic scaffold protein Pan1 with the type I myosins contributes to the late stages of endocytosis’, *Molecular Biology of the Cell*, 18(8), pp. 2893–2903. doi: 10.1091/mbc.e07-05-0436.

Bernander, R., Lind, A. E. and Ettema, T. J. G. (2011) ‘An archaeal origin for the actin cytoskeleton: Implications for eukaryogenesis’, *Communicative & Integrative Biology*, 4(6), pp. 664–667. doi: 10.4161/cib.16974.

Berro, J. and Pollard, T. D. (2014) ‘Local and global analysis of endocytic patch dynamics in fission yeast using a new “temporal superresolution” realignment method’, *Molecular Biology of the Cell*, 25(22), pp. 3501–3514. doi: 10.1091/mbc.E13-01-0004.

Biber, G. *et al.* (2021) ‘Targeting the actin nucleation promoting factor WASp provides a therapeutic approach for hematopoietic malignancies’, *Nature Communications*, 12(1), p. 5581. doi: 10.1038/s41467-021-25842-7.

Bieling, P., Hansen, S. D., Akin, O., Li, T.-D., Hayden, C. C., Fletcher, D. A. and Mullins, R. D. (2018) ‘WH2 and proline-rich domains of WASP-family proteins collaborate to accelerate actin filament elongation’, *The EMBO journal*, 37(1), pp. 102–121. doi: 10.15252/embj.201797039.

Birbaumer, M. and Schweitzer, F. (2011) ‘Agent-Based Modeling of Intracellular Transport’, *The European Physical Journal B*, 82(3–4), pp. 245–255. doi: 10.1140/epjb/e2011-20283-x.

Bonabeau, E. (2002) ‘Agent-based modeling: methods and techniques for simulating human systems’, *Proceedings of the National Academy of Sciences of the United States of America*, 99 Suppl 3(Suppl 3), pp. 7280–7287. doi: 10.1073/pnas.082080899.

Bork, P., Sander, C. and Valencia, A. (1992) ‘An ATPase domain common to prokaryotic cell cycle proteins, sugar kinases, actin, and hsp70 heat shock proteins’, *Proceedings of the National Academy of Sciences of the United States of America*, 89(16), pp. 7290–7294. doi: 10.1073/pnas.89.16.7290.

Botstein, D., Chervitz, S. A. and Cherry, J. M. (1997) ‘Yeast as a Model Organism’, *Science (New York, N.Y.)*, 277(5330), pp. 1259–1260.

Box, G. E. P. and Muller, M. E. (1958) ‘A Note on the Generation of Random Normal Deviates’, *The Annals of Mathematical Statistics*, 29(2), pp. 610–611. doi: 10.1214/aoms/1177706645.

Buracco, S., Singh, S., Claydon, S., Paschke, P., Tweedy, L., Whitelaw, J., McGarry, L., Thomason, P. A. and Insall, R. H. (2022) ‘The Scar/WAVE complex drives normal actin protrusions without the Arp2/3 complex, but proline-rich domains are required’. bioRxiv, p. 2022.05.14.491902. doi: 10.1101/2022.05.14.491902.

Burke, T. A., Christensen, J. R., Barone, E., Suarez, C., Sirotkin, V. and Kovar, D. R. (2014) ‘Homeostatic actin cytoskeleton networks are regulated by assembly factor competition for monomers’, *Current biology: CB*, 24(5), pp. 579–585. doi: 10.1016/j.cub.2014.01.072.

Burke, T. A., Harker, A. J., Dominguez, R. and Kovar, D. R. (2017) ‘The bacterial virulence factors VopL and VopF nucleate actin from the pointed end’, *The Journal of Cell Biology*, 216(5), pp. 1267–1276. doi: 10.1083/jcb.201608104.

Burtnick, L. D., Urosev, D., Irobi, E., Narayan, K. and Robinson, R. C. (2004) ‘Structure of the N-terminal half of gelsolin bound to actin: roles in severing, apoptosis and FAF’, *The EMBO Journal*, 23(14), pp. 2713–2722. doi: 10.1038/sj.emboj.7600280.

Cáceres, R., Abou-Ghali, M. and Plastino, J. (2015) ‘Reconstituting the actin cytoskeleton at or near surfaces in vitro’, *Biochimica Et Biophysica Acta*, 1853(11 Pt B), pp. 3006–3014. doi: 10.1016/j.bbamcr.2015.07.021.

Carlsson, A. E. (2018) ‘Membrane Bending by Actin Polymerization’, *Current opinion in cell biology*, 50, pp. 1–7. doi: 10.1016/j.ceb.2017.11.007.

Carpenter, E. P., Beis, K., Cameron, A. D. and Iwata, S. (2008) ‘Overcoming the challenges of membrane protein crystallography’, *Current Opinion in Structural Biology*, 18(5), pp. 581–586. doi: 10.1016/j.sbi.2008.07.001.

Carreño, B. and Howes, L. (2018) ‘The rise of GPU computing in science’, *EMBL*, 25 April. Available at: https://www.embl.org/news/science/the-rise-gpu-computing-science/ (Accessed: 22 September 2023).

Castellanos-Serra, L. and Hardy, E. (2006) ‘Negative detection of biomolecules separated in polyacrylamide electrophoresis gels’, *Nature Protocols*, 1(3), pp. 1544–1551. doi: 10.1038/nprot.2006.233.

Chaki, S. P. and Rivera, G. M. (2013) ‘Integration of signaling and cytoskeletal remodeling by Nck in directional cell migration’, *Bioarchitecture*, 3(3), pp. 57–63. doi: 10.4161/bioa.25744.

Chandrasekaran, A., Upadhyaya, A. and Papoian, G. A. (2019) ‘Remarkable structural transformations of actin bundles are driven by their initial polarity, motor activity, crosslinking, and filament treadmilling’, *PLOS Computational Biology*, 15(7), p. e1007156. doi: 10.1371/journal.pcbi.1007156.

Chen, Q. and Pollard, T. D. (2013) ‘Actin filament severing by cofilin dismantles actin patches and produces mother filaments for new patches’, *Current biology: CB*, 23(13), pp. 1154–1162. doi: 10.1016/j.cub.2013.05.005.

Chen, X., Ni, F., Kondrashkina, E., Ma, J. and Wang, Q. (2015) ‘Mechanisms of leiomodin 2-mediated regulation of actin filament in muscle cells’, *Proceedings of the National Academy of Sciences of the United States of America*, 112(41), pp. 12687–12692. doi: 10.1073/pnas.1512464112.

Chen, X., Ni, F., Tian, X., Kondrashkina, E., Wang, Q. and Ma, J. (2013) ‘Structural basis of actin filament nucleation by tandem W domains’, *Cell Reports*, 3(6), pp. 1910–1920. doi: 10.1016/j.celrep.2013.04.028.

Chereau, D., Boczkowska, M., Skwarek-Maruszewska, A., Fujiwara, I., Hayes, D. B., Rebowski, G., Lappalainen, P., Pollard, T. D. and Dominguez, R. (2008) ‘Leiomodin is an Actin Filament Nucleator in Muscle Cells’, *Science (New York, N.Y.)*, 320(5873), pp. 239–243. doi: 10.1126/science.1155313.

Chereau, D., Kerff, F., Graceffa, P., Grabarek, Z., Langsetmo, K. and Dominguez, R. (2005) ‘Actin-bound structures of Wiskott-Aldrich syndrome protein (WASP)-homology domain 2 and the implications for filament assembly’, *Proceedings of the National Academy of Sciences of the United States of America*, 102(46), pp. 16644–16649. doi: 10.1073/pnas.0507021102.

Chesarone, M. A. and Goode, B. L. (2009) ‘Actin nucleation and elongation factors: mechanisms and interplay’, *Current Opinion in Cell Biology*, 21(1), pp. 28–37. doi: 10.1016/j.ceb.2008.12.001.

Colonne, P. M., Winchell, C. G. and Voth, D. E. (2016) ‘Hijacking Host Cell Highways: Manipulation of the Host Actin Cytoskeleton by Obligate Intracellular Bacterial Pathogens’, *Frontiers in Cellular and Infection Microbiology*, 6, p. 107. doi: 10.3389/fcimb.2016.00107.

Courtemanche, N. (2018) ‘Mechanisms of formin-mediated actin assembly and dynamics’, *Biophysical Reviews*, 10(6), pp. 1553–1569. doi: 10.1007/s12551-018-0468-6.

Curchoe, C. L. (2020) ‘All Models Are Wrong, but Some Are Useful’, *Journal of Assisted Reproduction and Genetics*, 37(10), pp. 2389–2391. doi: 10.1007/s10815-020-01895-3.

Das, A., Bhat, A., Sknepnek, R., Köster, D., Mayor, S. and Rao, M. (2020) 'Stratification relieves constraints from steric hindrance in the generation of compact actomyosin asters at the membrane cortex', *Science Advances*, 6(11), p. eaay6093. doi: 10.1126/sciadv.aay6093.

De Luca, A., Avena, P., Sirianni, R., Chimento, A., Fallo, F., Pilon, C., Casaburi, I. and Pezzi, V. (2017) 'Role of Scaffold Protein Proline-, Glutamic Acid-, and Leucine-Rich Protein 1 (PELP1) in the Modulation of Adrenocortical Cancer Cell Growth', *Cells*, 6(4), p. 42. doi: 10.3390/cells6040042.

Demers, J.-P. and Mittermaier, A. (2009) 'Binding mechanism of an SH3 domain studied by NMR and ITC', *Journal of the American Chemical Society*, 131(12), pp. 4355–4367. doi: 10.1021/ja808255d.

Dionne, U. *et al.* (2021) 'Protein context shapes the specificity of SH3 domain-mediated interactions in vivo', *Nature Communications*, 12(1), p. 1597. doi: 10.1038/s41467-021-21873-2.

Divakaruni, A. V., Baida, C., White, C. L. and Gober, J. W. (2007) 'The cell shape proteins MreB and MreC control cell morphogenesis by positioning cell wall synthetic complexes', *Molecular Microbiology*, 66(1), pp. 174–188. doi: 10.1111/j.1365-2958.2007.05910.x.

Djakbarova, U., Madraki, Y., Chan, E. T. and Kural, C. (2021) 'Dynamic interplay between cell membrane tension and clathrin-mediated endocytosis', *Biology of the cell*, 113(8), pp. 344–373. doi: 10.1111/boc.202000110.

Dominguez, C., Boelens, R. and Bonvin, A. M. J. J. (2003) 'HADDOCK: a protein-protein docking approach based on biochemical or biophysical information', *Journal of the American Chemical Society*, 125(7), pp. 1731–1737. doi: 10.1021/ja026939x.

Dominguez, R. (2019) 'Nucleotide-dependent conformational changes in the actin filament: Subtler than expected', *Proceedings of the National Academy of Sciences of the United States of America*, 116(10), pp. 3959–3961. doi: 10.1073/pnas.1900799116.

Dominguez, R. and Holmes, K. C. (2011) 'Actin Structure and Function', *Annual review of biophysics*, 40, pp. 169–186. doi: 10.1146/annurev-biophys-042910-155359.

Doolittle, L. K., Rosen, M. K. and Padrick, S. B. (2013) 'Measurement and Analysis of in vitro Actin Polymerization', *Methods in molecular biology (Clifton, N.J.)*, 1046, pp. 273–293. doi: 10.1007/978-1-62703-538-5_16.

Ducka, A. M., Joel, P., Popowicz, G. M., Trybus, K. M., Schleicher, M., Noegel, A. A., Huber, R., Holak, T. A. and Sitar, T. (2010) 'Structures of actin-bound Wiskott-Aldrich syndrome protein homology 2 (WH2) domains of Spire and the implication for filament nucleation', *Proceedings of the National Academy of Sciences of the United States of America*, 107(26), pp. 11757–11762. doi: 10.1073/pnas.1005347107.

Duhr, S. and Braun, D. (2006) 'Why molecules move along a temperature gradient', *Proceedings of the National Academy of Sciences of the United States of America*, 103(52), pp. 19678–19682. doi: 10.1073/pnas.0603873103.

Einstein, A. (1905) 'On the movement of small particles suspended in stationary liquids required by the molecular-kinetic theory of heat', *Annals of Physics*, 322(8), pp. 549–560.

Feliciano, D. and Di Pietro, S. M. (2012) 'SLAC, a complex between Sla1 and Las17, regulates actin polymerization during clathrin-mediated endocytosis', *Molecular Biology of the Cell*, 23(21), pp. 4256–4272. doi: 10.1091/mbc.E11-12-1022.

Feliciano, D., Tolsma, T. O., Farrell, K. B., Aradi, A. and Di Pietro, S. M. (2015) 'A Second Las17 Monomeric Actin-Binding Motif Functions in Arp2/3-Dependent Actin Polymerization During Endocytosis', *Traffic (Copenhagen, Denmark)*, 16(4), pp. 379–397. doi: 10.1111/tra.12259.

Firat-Karalar, E. N., Hsiue, P. P. and Welch, M. D. (2011) 'The actin nucleation factor JMY is a negative regulator of neuritogenesis', *Molecular Biology of the Cell*, 22(23), pp. 4563–4574. doi: 10.1091/mbc.E11-06-0585.

Firat-Karalar, E. N. and Welch, M. D. (2011) 'New mechanisms and functions of actin nucleation', *Current Opinion in Cell Biology*, 23(1), pp. 4–13. doi: 10.1016/j.ceb.2010.10.007.

Fowler, V. M. and Dominguez, R. (2017) 'Tropomodulins and Leiomodins: Actin Pointed End Caps and Nucleators in Muscles', *Biophysical Journal*, 112(9), pp. 1742–1760. doi: 10.1016/j.bpj.2017.03.034.

Fullstone, G., Guttà, C., Beyer, A. and Rehm, M. (2020) 'The FLAME-accelerated signalling tool (FaST) for facile parallelisation of flexible agent-based models of cell signalling', *NPJ systems biology and applications*, 6(1), p. 10. doi: 10.1038/s41540-020-0128-x.

Galletta, B. J., Chuang, D. Y. and Cooper, J. A. (2008) 'Distinct Roles for Arp2/3 Regulators in Actin Assembly and Endocytosis', *PLOS Biology*, 6(1), p. e1. doi: 10.1371/journal.pbio.0060001.

GlobalData (2022) 'Gaming Market Analysis by Region, Platform (Smartphone, Console, PC, Tablet, Handheld Controllers, Head-mounted Displays), Gamer Type and Segment Forecast to 2030', *Market Research Reports & Consulting | GlobalData UK Ltd.*, September. Available at: https://www.globaldata.com/store/report/gaming-market-analysis/ (Accessed: 23 September 2023).

Goley, E. D. and Welch, M. D. (2006) 'The ARP2/3 complex: an actin nucleator comes of age', *Nature Reviews. Molecular Cell Biology*, 7(10), pp. 713–726. doi: 10.1038/nrm2026.

Goode, B. L., Eskin, J. A. and Wendland, B. (2015) 'Actin and Endocytosis in Budding Yeast', *Genetics*, 199(2), pp. 315–358. doi: 10.1534/genetics.112.145540.

Gressin, L., Guillotin, A., Guérin, C., Blanchoin, L. and Michelot, A. (2015) 'Architecture dependence of actin filament network disassembly', *Current biology: CB*, 25(11), pp. 1437–1447. doi: 10.1016/j.cub.2015.04.011.

Hatano, T. *et al.* (2018) 'Rapid production of pure recombinant actin isoforms in Pichia pastoris', *Journal of Cell Science*, 131(8), p. jcs213827. doi: 10.1242/jcs.213827.

Holm A, null, Tejle, K., Magnusson, K. E., Descoteaux, A. and Rasmusson, B. (2001) 'Leishmania donovani lipophosphoglycan causes periphagosomal actin accumulation: correlation with impaired translocation of PKCalpha and defective phagosome maturation', *Cellular Microbiology*, 3(7), pp. 439–447. doi: 10.1046/j.1462-5822.2001.00127.x.

Howard, J. P., Hutton, J. L., Olson, J. M. and Payne, G. S. (2002) 'Sla1p serves as the targeting signal recognition factor for NPFX(1,2)D-mediated endocytosis', *The Journal of Cell Biology*, 157(2), pp. 315–326. doi: 10.1083/jcb.200110027.

Hu, X. and Mullins, R. D. (2019) 'LC3 and STRAP regulate actin filament assembly by JMY during autophagosome formation', *The Journal of Cell Biology*, 218(1), pp. 251–266. doi: 10.1083/jcb.201802157.

Hummel, D. R. and Kaksonen, M. (2023) ‘Spatio-temporal regulation of endocytic protein assembly by SH3 domains in yeast’, *Molecular Biology of the Cell*, 34(3), p. ar19. doi: 10.1091/mbc.E22-09-0406.

Isambert, H., Venier, P., Maggs, A. C., Fattoum, A., Kassab, R., Pantaloni, D. and Carlier, M. F. (1995) ‘Flexibility of actin filaments derived from thermal fluctuations. Effect of bound nucleotide, phalloidin, and muscle regulatory proteins’, *The Journal of Biological Chemistry*, 270(19), pp. 11437–11444. doi: 10.1074/jbc.270.19.11437.

Izadi, M., Seemann, E., Schlobinski, D., Schwintzer, L., Qualmann, B. and Kessels, M. M. (2021) ‘Functional interdependence of the actin nucleator Cobl and Cobl-like in dendritic arbor development’, *eLife*, 10, p. e67718. doi: 10.7554/eLife.67718.

Jarmoskaite, I., AlSadhan, I., Vaidyanathan, P. P. and Herschlag, D. (2020) ‘How to measure and evaluate binding affinities’, *eLife*, 9, p. e57264. doi: 10.7554/eLife.57264.

Jerabek-Willemsen, M., Wienken, C. J., Braun, D., Baaske, P. and Duhr, S. (2011) ‘Molecular Interaction Studies Using Microscale Thermophoresis’, *Assay and Drug Development Technologies*, 9(4), pp. 342–353. doi: 10.1089/adt.2011.0380.

Jin, Y. *et al.* (2004) ‘Mutations of the Wiskott-Aldrich Syndrome Protein (WASP): hotspots, effect on transcription, and translation and phenotype/genotype correlation’, *Blood*, 104(13), pp. 4010–4019. doi: 10.1182/blood-2003-05-1592.

Johnson, K. A. and Goody, R. S. (2011) ‘The Original Michaelis Constant: Translation of the 1913 Michaelis–Menten Paper’, *Biochemistry*, 50(39), pp. 8264–8269. doi: 10.1021/bi201284u.

Juanes, M. A., Bouguenina, H., Eskin, J. A., Jaiswal, R., Badache, A. and Goode, B. L. (2017) ‘Adenomatous polyposis coli nucleates actin assembly to drive cell migration and microtubule-induced focal adhesion turnover’, *The Journal of Cell Biology*, 216(9), pp. 2859–2875. doi: 10.1083/jcb.201702007.

Juanes, M. A., Fees, C. P., Hoeprich, G. J., Jaiswal, R. and Goode, B. L. (2020) ‘EB1 Directly Regulates APC-Mediated Actin Nucleation’, *Current biology: CB*, 30(23), pp. 4763-4772.e8. doi: 10.1016/j.cub.2020.08.094.

Jumper, J. *et al.* (2021) ‘Highly accurate protein structure prediction with AlphaFold’, *Nature*, 596(7873), pp. 583–589. doi: 10.1038/s41586-021-03819-2.

Kärkkäinen, S., Hiipakka, M., Wang, J.-H., Kleino, I., Vähä-Jaakkola, M., Renkema, G. H., Liss, M., Wagner, R. and Saksela, K. (2006) ‘Identification of preferred protein interactions by phage-display of the human Src homology-3 proteome’, *EMBO Reports*, 7(2), pp. 186–191. doi: 10.1038/sj.embor.7400596.

Karplus, M. and McCammon, J. A. (2002) ‘Molecular dynamics simulations of biomolecules’, *Nature Structural Biology*, 9(9), pp. 646–652. doi: 10.1038/nsb0902-646.

Kast, D. J., Zajac, A. L., Holzbaur, E. L. F., Ostap, E. M. and Dominguez, R. (2015) ‘WHAMM Directs the Arp2/3 Complex to the ER for Autophagosome Biogenesis through an Actin Comet Tail Mechanism’, *Current biology: CB*, 25(13), pp. 1791–1797. doi: 10.1016/j.cub.2015.05.042.

Kastritis, P. L. and Bonvin, A. M. J. J. (2010) ‘Are scoring functions in protein-protein docking ready to predict interactomes? Clues from a novel binding affinity benchmark’, *Journal of Proteome Research*, 9(5), pp. 2216–2225. doi: 10.1021/pr9009854.

Kay, B. K., Williamson, M. P. and Sudol, M. (2000) ‘The importance of being proline: the interaction of proline-rich motifs in signaling proteins with their cognate domains’, *FASEB journal: official publication of the Federation of American Societies for Experimental Biology*, 14(2), pp. 231–241.

King, J. S., Veltman, D. M., Georgiou, M., Baum, B. and Insall, R. H. (2010) ‘SCAR/WAVE is activated at mitosis and drives myosin-independent cytokinesis’, *Journal of Cell Science*, 123(13), pp. 2246–2255. doi: 10.1242/jcs.063735.

Klann, M., Koeppl, H. and Reuss, M. (2012) ‘Spatial modeling of vesicle transport and the cytoskeleton: the challenge of hitting the right road’, *PloS One*, 7(1), p. e29645. doi: 10.1371/journal.pone.0029645.

Kloc, M., Uosef, A., Wosik, J., Kubiak, J. Z. and Ghobrial, R. M. (2022) ‘Virus interactions with the actin cytoskeleton-what we know and do not know about SARS-CoV-2’, *Archives of Virology*, 167(3), pp. 737–749. doi: 10.1007/s00705-022-05366-1.

Kotila, T., Wioland, H., Selvaraj, M., Kogan, K., Antenucci, L., Jégou, A., Huiskonen, J. T., Romet-Lemonne, G. and Lappalainen, P. (2022) ‘Structural basis of rapid actin dynamics in the evolutionarily divergent Leishmania parasite’, *Nature Communications*, 13(1), p. 3442. doi: 10.1038/s41467-022-31068-y.

Kuhn, J. R. and Pollard, T. D. (2005) ‘Real-time measurements of actin filament polymerization by total internal reflection fluorescence microscopy’, *Biophysical Journal*, 88(2), pp. 1387–1402. doi: 10.1529/biophysj.104.047399.

Kunitz, M., Anson, M. L. and Northrop, J. H. (1934) ‘Molecular weight, molecular volume, and hydration of proteins in solution’, *The Journal of General Physiology*, 17(3), pp. 365–373. doi: 10.1085/jgp.17.3.365.

Kurien, B. T., Dorri, Y. and Scofield, R. H. (2012) ‘Spicy SDS-PAGE gels: curcumin/turmeric as an environment-friendly protein stain’, *Methods in Molecular Biology (Clifton, N.J.)*, 869, pp. 567–578. doi: 10.1007/978-1-61779-821-4_51.

Lacy, M. M., Ma, R., Ravindra, N. G. and Berro, J. (2018) ‘Molecular mechanisms of force production in clathrin-mediated endocytosis’, *FEBS letters*, 592(21), pp. 3586–3605. doi: 10.1002/1873-3468.13192.

Langford, G. M. (2002) ‘Myosin-V, a versatile motor for short-range vesicle transport’, *Traffic (Copenhagen, Denmark)*, 3(12), pp. 859–865. doi: 10.1034/j.1600-0854.2002.31202.x.

Lee, A., Yau, C., Giles, M. B., Doucet, A. and Holmes, C. C. (2010) ‘On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods’, *Journal of computational and graphical statistics*, 19(4), pp. 769–789.

Lu, R., Drubin, D. G. and Sun, Y. (2016) ‘Clathrin-mediated endocytosis in budding yeast at a glance’, *Journal of Cell Science*, 129(8), pp. 1531–1536. doi: 10.1242/jcs.182303.

Machesky, L. M., Atkinson, S. J., Ampe, C., Vandekerckhove, J. and Pollard, T. D. (1994) ‘Purification of a cortical complex containing two unconventional actins from Acanthamoeba by affinity chromatography on profilin-agarose’, *The Journal of Cell Biology*, 127(1), pp. 107–115. doi: 10.1083/jcb.127.1.107.

Madania, A., Dumoulin, P., Grava, S., Kitamoto, H., Schärer-Brodbeck, C., Soulard, A., Moreau, V. and Winsor, B. (1999) ‘The Saccharomyces cerevisiae Homologue of Human Wiskott–Aldrich Syndrome

Protein Las17p Interacts with the Arp2/3 Complex', *Molecular Biology of the Cell*, 10(10), pp. 3521–3538.

Majkut, P., Claußnitzer, I., Merk, H., Freund, C., Hackenberger, C. P. R. and Gerrits, M. (2013) 'Completion of proteomic data sets by Kd measurement using cell-free synthesis of site-specifically labeled proteins', *PloS One*, 8(12), p. e82352. doi: 10.1371/journal.pone.0082352.

Matplotlib (2012) *Animated 3D random walk — Matplotlib 3.8.0 documentation*. Available at: https://matplotlib.org/stable/gallery/animation/random_walk.html (Accessed: 25 September 2023).

McCullagh, M., Saunders, M. G. and Voth, G. A. (2014) 'Unraveling the Mystery of ATP Hydrolysis in Actin Filaments', *Journal of the American Chemical Society*, 136(37), pp. 13053–13058. doi: 10.1021/ja507169f.

McGrath, J. L., Hartwig, J. H., Tardy, Y. and Dewey, C. F. (1998) 'Measuring actin dynamics in endothelial cells', *Microscopy Research and Technique*, 43(5), pp. 385–394. doi: 10.1002/(SICI)1097-0029(19981201)43:5<385::AID-JEMT5>3.0.CO;2-Z.

Mermall, V., Post, P. L. and Mooseker, M. S. (1998) 'Unconventional myosins in cell movement, membrane traffic, and signal transduction', *Science (New York, N.Y.)*, 279(5350), pp. 527–533. doi: 10.1126/science.279.5350.527.

Merrifield, C. J. and Kaksonen, M. (2014) 'Endocytic Accessory Factors and Regulation of Clathrin-Mediated Endocytosis', *Cold Spring Harbor Perspectives in Biology*, 6(11), p. a016733. doi: 10.1101/cshperspect.a016733.

Miyamoto, S. and Shimono, K. (2020) 'Molecular Modeling to Estimate the Diffusion Coefficients of Drugs and Other Small Molecules', *Molecules*, 25(22), p. 5340. doi: 10.3390/molecules25225340.

Mollica, L., Bessa, L. M., Hanoulle, X., Jensen, M. R., Blackledge, M. and Schneider, R. (2016) 'Binding Mechanisms of Intrinsically Disordered Proteins: Theory, Simulation, and Experiment', *Frontiers in Molecular Biosciences*, 3, p. 52. doi: 10.3389/fmolb.2016.00052.

Moseley, J. B. and Goode, B. L. (2006) 'The Yeast Actin Cytoskeleton: from Cellular Function to Biochemical Mechanism', *Microbiology and Molecular Biology Reviews*, 70(3), pp. 605–645. doi: 10.1128/MMBR.00013-06.

Müller-Esparza, H., Osorio-Valeriano, M., Steube, N., Thanbichler, M. and Randau, L. (2020) 'Bio-Layer Interferometry Analysis of the Target Binding Activity of CRISPR-Cas Effector Complexes', *Frontiers in Molecular Biosciences*, 7, p. 98. doi: 10.3389/fmolb.2020.00098.

Mullins, R. D., Heuser, J. A. and Pollard, T. D. (1998) 'The interaction of Arp2/3 complex with actin: nucleation, high affinity pointed end capping, and formation of branching networks of filaments', *Proceedings of the National Academy of Sciences of the United States of America*, 95(11), pp. 6181–6186. doi: 10.1073/pnas.95.11.6181.

Mund, M. *et al.* (2018) 'Systematic Nanoscale Analysis of Endocytosis Links Efficient Vesicle Formation to Patterned Actin Nucleation', *Cell*, 174(4), pp. 884-896.e17. doi: 10.1016/j.cell.2018.06.032.

Murray, N. J., Williamson, M. P., Lilley, T. H. and Haslam, E. (1994) 'Study of the interaction between salivary proline-rich proteins and a polyphenol by 1H-NMR spectroscopy', *European Journal of Biochemistry*, 219(3), pp. 923–935. doi: 10.1111/j.1432-1033.1994.tb18574.x.

Namgoong, S., Boczkowska, M., Glista, M. J., Winkelman, J. D., Rebowski, G., Kovar, D. R. and Dominguez, R. (2011) 'Mechanism of actin filament nucleation by Vibrio VopL and implications for tandem-W domain nucleation', *Nature structural & molecular biology*, 18(9), pp. 1060–1067. doi: 10.1038/nsmb.2109.

Necci, M., Piovesan, D., CAID Predictors, DisProt Curators and Tosatto, S. C. E. (2021) 'Critical assessment of protein intrinsic disorder prediction', *Nature Methods*, 18(5), pp. 472–481. doi: 10.1038/s41592-021-01117-3.

Oda, T., Aihara, T. and Wakabayashi, K. (2016) 'Early nucleation events in the polymerization of actin, probed by time-resolved small-angle x-ray scattering', *Scientific Reports*, 6(1), p. 34539. doi: 10.1038/srep34539.

Oda, T., Iwasa, M., Aihara, T., Maéda, Y. and Narita, A. (2009) 'The nature of the globular- to fibrous-actin transition', *Nature*, 457(7228), pp. 441–445. doi: 10.1038/nature07685.

Okada, K., Bartolini, F., Deaconescu, A. M., Moseley, J. B., Dogic, Z., Grigorieff, N., Gundersen, G. G. and Goode, B. L. (2010) 'Adenomatous polyposis coli protein nucleates actin assembly and synergizes with the formin mDia1', *The Journal of Cell Biology*, 189(7), pp. 1087–1096. doi: 10.1083/jcb.201001016.

Padrick, S. B. *et al.* (2008) 'Hierarchical regulation of WASP/WAVE proteins', *Molecular Cell*, 32(3), pp. 426–438. doi: 10.1016/j.molcel.2008.10.012.

Palani, S., Ghosh, S., Ivorra-Molla, E., Clarke, S., Suchenko, A., Balasubramanian, M. K. and Köster, D. V. (2021) 'Calponin-homology domain mediated bending of membrane-associated actin filaments', *eLife*. Edited by P. Lappalainen, V. Malhotra, and P. Lappalainen, 10, p. e61078. doi: 10.7554/eLife.61078.

Pandey, A. K., Thomas, K. M., Forbes, C. R. and Zondlo, N. J. (2014) 'Tunable control of polyproline helix (PPII) structure via aromatic electronic effects: an electronic switch of polyproline helix', *Biochemistry*, 53(32), pp. 5307–5314. doi: 10.1021/bi500696k.

Pang, X., Zhou, K. H., Qin, S. and Zhou, H.-X. (2012) 'Prediction and dissection of widely-varying association rate constants of actin-binding proteins', *PLoS computational biology*, 8(10), p. e1002696. doi: 10.1371/journal.pcbi.1002696.

Panni, S., Dente, L. and Cesareni, G. (2002) 'In vitro evolution of recognition specificity mediated by SH3 domains reveals target recognition rules', *The Journal of Biological Chemistry*, 277(24), pp. 21666–21674. doi: 10.1074/jbc.M109788200.

Pegoraro, A. F., Janmey, P. and Weitz, D. A. (2017) 'Mechanical Properties of the Cytoskeleton and Cells', *Cold Spring Harbor Perspectives in Biology*, 9(11), p. a022038. doi: 10.1101/cshperspect.a022038.

Picco, A., Mund, M., Ries, J., Nédélec, F. and Kaksonen, M. (2015) 'Visualizing the functional architecture of the endocytic machinery', *eLife*, 4, p. e04535. doi: 10.7554/eLife.04535.

Pogson, M., Smallwood, R., Qwarnstrom, E. and Holcombe, M. (2006) 'Formal agent-based modelling of intracellular chemical interactions', *Bio Systems*, 85(1), pp. 37–45. doi: 10.1016/j.biosystems.2006.02.004.

Pollard, T. D. (1986) 'Rate constants for the reactions of ATP- and ADP-actin with the ends of actin filaments', *The Journal of Cell Biology*, 103(6), pp. 2747–2754.

Pollard, T. D. (2016) ‘Actin and Actin-Binding Proteins’, *Cold Spring Harbor Perspectives in Biology*, 8(8), p. a018226. doi: 10.1101/cshperspect.a018226.

Pollard, T. D. and Borisy, G. G. (2003) ‘Cellular motility driven by assembly and disassembly of actin filaments’, *Cell*, 112(4), pp. 453–465. doi: 10.1016/s0092-8674(03)00120-x.

Pollard, T. D. and Goldman, R. D. (2018) ‘Overview of the Cytoskeleton from an Evolutionary Perspective’, *Cold Spring Harbor Perspectives in Biology*, 10(7), p. a030288. doi: 10.1101/cshperspect.a030288.

Pollard, T. D. and O’Shaughnessy, B. (2019) ‘Molecular Mechanism of Cytokinesis’, *Annual review of biochemistry*, 88, pp. 661–689. doi: 10.1146/annurev-biochem-062917-012530.

Poupel, O., Boleti, H., Axisa, S., Couture-Tosi, E. and Tardieux, I. (2000) ‘Toxofilin, a Novel Actin-binding Protein from Toxoplasma gondii, Sequesters Actin Monomers and Caps Actin Filaments’, *Molecular Biology of the Cell*, 11(1), pp. 355–368.

Quinlan, M. E. (2013) ‘Direct interaction between two actin nucleators is required in Drosophila oogenesis’, *Development (Cambridge, England)*, 140(21), pp. 4417–4425. doi: 10.1242/dev.097337.

Quinlan, M. E., Heuser, J. E., Kerkhoff, E. and Dyche Mullins, R. (2005) ‘Drosophila Spire is an actin nucleation factor’, *Nature*, 433(7024), pp. 382–388. doi: 10.1038/nature03241.

Rebowski, G., Namgoong, S., Boczkowska, M., Leavis, P. C., Navaza, J. and Dominguez, R. (2010) ‘Structure of a longitudinal actin dimer assembled by tandem w domains: implications for actin filament nucleation’, *Journal of Molecular Biology*, 403(1), pp. 11–23. doi: 10.1016/j.jmb.2010.08.040.

Richmond, P., Walker, D., Coakley, S. and Romano, D. (2010) ‘High performance cellular level agent-based simulation with FLAME for the GPU’, *Briefings in Bioinformatics*, 11(3), pp. 334–347. doi: 10.1093/bib/bbp073.

Robertson, A. S., Allwood, E. G., Smith, A. P. C., Gardiner, F. C., Costa, R., Winder, S. J. and Ayscough, K. R. (2009) ‘The WASP homologue Las17 activates the novel actin-regulatory activity of Ysc84 to promote endocytosis in yeast’, *Molecular Biology of the Cell*, 20(6), pp. 1618–1628. doi: 10.1091/mbc.e08-09-0982.

Rodal, A. A., Manning, A. L., Goode, B. L. and Drubin, D. G. (2003) ‘Negative regulation of yeast WASp by two SH3 domain-containing proteins’, *Current biology: CB*, 13(12), pp. 1000–1008. doi: 10.1016/s0960-9822(03)00383-x.

Rohman, M. and Harrison-Lavoie, K. J. (2000) ‘Separation of copurifying GroEL from glutathione-S-transferase fusion proteins’, *Protein Expression and Purification*, 20(1), pp. 45–47. doi: 10.1006/prep.2000.1271.

Rothe, M., Gruber, T., Gröger, S., Balbach, J., Saalwächter, K. and Roos, M. (2016) ‘Transient binding accounts for apparent violation of the generalized Stokes-Einstein relation in crowded protein solutions’, *Physical chemistry chemical physics: PCCP*, 18(27), pp. 18006–18014. doi: 10.1039/c6cp01056c.

Sabirov, M. and Spirov, A. (2020) ‘3D agent-based modeling of some aspects of the interaction between microtubules and microfilaments in cell’, in *2020 Cognitive Sciences, Genomics and Bioinformatics (CSGB)*. *2020 Cognitive Sciences, Genomics and Bioinformatics (CSGB)*, pp. 1–5. doi: 10.1109/CSGB51356.2020.9214630.

Sagot, I., Rodal, A. A., Moseley, J., Goode, B. L. and Pellman, D. (2002) ‘An actin nucleation mechanism mediated by Bni1 and profilin’, *Nature Cell Biology*, 4(8), pp. 626–631. doi: 10.1038/ncb834.

Sambrook, J., Fritsch, E. F. and Maniatis, T. (1989) *Gel Electrophoresis of DNA. In: Sambrook, J., Fritsch, E.F. and Maniatis, T., Eds., Molecular Cloning: A Laboratory Manual, Chapter 6*. New York: Cold Spring Harbor Laboratory Press.

Schaller, V., Weber, C. A., Hammerich, B., Frey, E. and Bausch, A. R. (2011) ‘Frozen steady states in active systems’, *Proceedings of the National Academy of Sciences of the United States of America*, 108(48), pp. 19183–19188. doi: 10.1073/pnas.1107540108.

Schavemaker, P. E., Boersma, A. J. and Poolman, B. (2018) ‘How Important Is Protein Diffusion in Prokaryotes?’, *Frontiers in Molecular Biosciences*, 5, p. 93. doi: 10.3389/fmolb.2018.00093.

‘SD-Test - Definition and Relevance | Nanopedia’ (2023). Available at: https://nanotempertech.com/nanopedia/sd-test/ (Accessed: 23 September 2023).

Sept, D. and McCammon, J. A. (2001) ‘Thermodynamics and kinetics of actin filament nucleation.’, *Biophysical Journal*, 81(2), pp. 667–674.

Shaaban, M., Chowdhury, S. and Nolen, B. J. (2020) ‘Cryo-EM reveals the transition of Arp2/3 complex from inactive to nucleation-competent state’, *Nature Structural & Molecular Biology*, 27(11), pp. 1009–1016. doi: 10.1038/s41594-020-0481-x.

Sitar, T. *et al.* (2011) ‘Molecular architecture of the Spire-actin nucleus and its implication for actin filament assembly’, *Proceedings of the National Academy of Sciences of the United States of America*, 108(49), pp. 19575–19580. doi: 10.1073/pnas.1115465108.

Skoble, J., Portnoy, D. A. and Welch, M. D. (2000) ‘Three Regions within Acta Promote Arp2/3 Complex-Mediated Actin Nucleation and Listeria monocytogenes Motility’, *The Journal of Cell Biology*, 150(3), pp. 527–538.

Speagle, J. (2019) ‘A Conceptual Introduction to Markov Chain Monte Carlo Methods’, *arXiv: Other Statistics*. Available at: https://www.semanticscholar.org/paper/A-Conceptual-Introduction-to-Markov-Chain-Monte-Speagle/2e05525be9bb52e257205489104aaec1acd1a5fb (Accessed: 22 September 2023).

Spudich, J. A. and Watt, S. (1971) ‘The regulation of rabbit skeletal muscle contraction. I. Biochemical studies of the interaction of the tropomyosin-troponin complex with actin and the proteolytic fragments of myosin’, *The Journal of Biological Chemistry*, 246(15), pp. 4866–4871.

Squire, J. (2019) ‘Special Issue: The Actin-Myosin Interaction in Muscle: Background and Overview’, *International Journal of Molecular Sciences*, 20(22), p. 5715. doi: 10.3390/ijms20225715.

Stamenova, S. D., French, M. E., He, Y., Francis, S. A., Kramer, Z. B. and Hicke, L. (2007) ‘Ubiquitin binds to and regulates a subset of SH3 domains’, *Molecular cell*, 25(2), pp. 273–284. doi: 10.1016/j.molcel.2006.12.016.

Stuart, S. F., Leatherbarrow, R. J. and Willison, K. R. (2011) ‘A two-step mechanism for the folding of actin by the yeast cytosolic chaperonin’, *The Journal of Biological Chemistry*, 286(1), pp. 178–184. doi: 10.1074/jbc.M110.166256.

Suarez, C., Carroll, R. T., Burke, T. A., Christensen, J. R., Bestul, A. J., Sees, J. A., James, M. L., Sirotkin, V. and Kovar, D. R. (2015) 'Profilin regulates F-actin network homeostasis by favoring formin over Arp2/3 complex', *Developmental Cell*, 32(1), pp. 43–53. doi: 10.1016/j.devcel.2014.10.027.

Sultana, A. and Lee, J. E. (2015) 'Measuring protein-protein and protein-nucleic Acid interactions by biolayer interferometry', *Current Protocols in Protein Science*, 79, p. 19.25.1-19.25.26. doi: 10.1002/0471140864.ps1925s79.

Sun, Y., Martin, A. C. and Drubin, D. G. (2006) 'Endocytic internalization in budding yeast requires coordinated actin nucleation and myosin motor activity', *Developmental Cell*, 11(1), pp. 33–46. doi: 10.1016/j.devcel.2006.05.008.

Sun, Y., Schöneberg, J., Chen, X., Jiang, T., Kaplan, C., Xu, K., Pollard, T. D. and Drubin, D. G. (2019) 'Direct comparison of clathrin-mediated endocytosis in budding and fission yeast reveals conserved and evolvable features', *eLife*, 8, p. e50749. doi: 10.7554/eLife.50749.

Taylor, M. J., Perrais, D. and Merrifield, C. J. (2011) 'A high precision survey of the molecular dynamics of mammalian clathrin-mediated endocytosis', *PLoS biology*, 9(3), p. e1000604. doi: 10.1371/journal.pbio.1000604.

Thompson, M. E., Heimsath, E. G., Gauvin, T. J., Higgs, H. N. and Kull, F. J. (2013) 'Structure of the FMNL3 FH2/actin complex provides insight into formin-mediated actin nucleation and elongation', *Nature structural & molecular biology*, 20(1), pp. 111–118. doi: 10.1038/nsmb.2462.

Tittel, J., Welz, T., Czogalla, A., Dietrich, S., Samol-Wolf, A., Schulte, M., Schwille, P., Weidemann, T. and Kerkhoff, E. (2015) 'Membrane Targeting of the Spir·Formin Actin Nucleator Complex Requires a Sequential Handshake of Polar Interactions', *The Journal of Biological Chemistry*, 290(10), pp. 6428–6444. doi: 10.1074/jbc.M114.602672.

Tolkatchev, D., Gregorio, C. C. and Kostyukova, A. S. (2022) 'The role of leiomodin in actin dynamics: a new road or a secret gate', *The FEBS journal*, 289(20), pp. 6119–6131. doi: 10.1111/febs.16128.

Tolsma, T. O., Cuevas, L. M. and Di Pietro, S. M. (2018) 'The Sla1 adaptor-clathrin interaction regulates coat formation and progression of endocytosis', *Traffic (Copenhagen, Denmark)*, 19(6), pp. 446–462. doi: 10.1111/tra.12563.

Tonikian, R. *et al.* (2009) 'Bayesian modeling of the yeast SH3 domain interactome predicts spatiotemporal dynamics of endocytosis proteins', *PLoS biology*, 7(10), p. e1000218. doi: 10.1371/journal.pbio.1000218.

Top, D., Read, J. A., Dawe, S. J., Syvitski, R. T. and Duncan, R. (2012) 'Cell-cell membrane fusion induced by p15 fusion-associated small transmembrane (FAST) protein requires a novel fusion peptide motif containing a myristoylated polyproline type II helix', *The Journal of Biological Chemistry*, 287(5), pp. 3403–3414. doi: 10.1074/jbc.M111.305268.

Torres, N. V. and Santos, G. (2015) 'The (Mathematical) Modeling Process in Biosciences', *Frontiers in Genetics*, 6, p. 354. doi: 10.3389/fgene.2015.00354.

Tratt, L. (2009) 'Chapter 5 Dynamically Typed Languages', in *Advances in Computers*. Elsevier, pp. 149–184. doi: 10.1016/S0065-2458(09)01205-4.

Tu, D., Graziano, B. R., Park, E., Zheng, W., Li, Y., Goode, B. L. and Eck, M. J. (2012) 'Structure of the formin-interaction domain of the actin nucleation-promoting factor Bud6', *Proceedings of the*

National Academy of Sciences of the United States of America, 109(50), pp. E3424-3433. doi: 10.1073/pnas.1203035109.

Tyler, J. J., Allwood, E. G. and Ayscough, K. R. (2016) 'WASP family proteins, more than Arp2/3 activators', *Biochemical Society Transactions*, 44(5), pp. 1339–1345. doi: 10.1042/BST20160176.

Tyler, J. J., Smaczynska-de Rooij, I. I., Abugharsa, L., Palmer, J. S., Hancock, L. P., Allwood, E. G. and Ayscough, K. R. (2021) 'Phosphorylation of the WH2 domain in yeast Las17/WASP regulates G-actin binding and protein function during endocytosis', *Scientific Reports*, 11(1), p. 9718. doi: 10.1038/s41598-021-88826-z.

UniProt Consortium (2023) 'UniProt: the Universal Protein Knowledgebase in 2023', *Nucleic Acids Research*, 51(D1), pp. D523–D531. doi: 10.1093/nar/gkac1052.

Urbanek, A. N., Allwood, E. G., Smith, A. P., Booth, W. I. and Ayscough, K. R. (2015) 'Distinct Actin and Lipid Binding Sites in Ysc84 Are Required during Early Stages of Yeast Endocytosis', *PloS One*, 10(8), p. e0136732. doi: 10.1371/journal.pone.0136732.

Urbanek, A. N., Smith, A. P., Allwood, E. G., Booth, W. I. and Ayscough, K. R. (2013) 'A novel actin-binding motif in Las17/WASP nucleates actin filaments independently of Arp2/3', *Current biology: CB*, 23(3), pp. 196–203. doi: 10.1016/j.cub.2012.12.024.

Varadi, M. *et al.* (2022) 'AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models', *Nucleic Acids Research*, 50(D1), pp. D439–D444. doi: 10.1093/nar/gkab1061.

Veltman, D. M., King, J. S., Machesky, L. M. and Insall, R. H. (2012) 'SCAR knockouts in Dictyostelium: WASP assumes SCAR's position and upstream regulators in pseudopods', *Journal of Cell Biology*, 198(4), pp. 501–508. doi: 10.1083/jcb.201205058.

Verkman, A. S. (2002) 'Solute and macromolecule diffusion in cellular aqueous compartments', *Trends in Biochemical Sciences*, 27(1), pp. 27–33. doi: 10.1016/s0968-0004(01)02003-5.

Wang, X., Galletta, B. J., Cooper, J. A. and Carlsson, A. E. (2016) 'Actin-Regulator Feedback Interactions during Endocytosis', *Biophysical Journal*, 110(6), pp. 1430–1443. doi: 10.1016/j.bpj.2016.02.018.

Waterhouse, A. *et al.* (2018) 'SWISS-MODEL: homology modelling of protein structures and complexes', *Nucleic Acids Research*, 46(W1), pp. W296–W303. doi: 10.1093/nar/gky427.

Watson, H. (2015) 'Biological membranes', *Essays in Biochemistry*, 59, pp. 43–69. doi: 10.1042/bse0590043.

Wegner, A. (1976) 'Head to tail polymerization of actin', *Journal of Molecular Biology*, 108(1), pp. 139–150. doi: 10.1016/s0022-2836(76)80100-3.

Weikai, G. *et al.* (2020) 'Synthesis and Biological Evaluation of B-Cell Lymphoma 6 Inhibitors of N-Phenyl-4-pyrimidinamine Derivatives Bearing Potent Activities against Tumor Growth', *Journal of Medicinal Chemistry*, 63(2), pp. 676–695. doi: 10.1021/acs.jmedchem.9b01618.

Welch, M. D., Iwamatsu, A. and Mitchison, T. J. (1997) 'Actin polymerization is induced by Arp 2/3 protein complex at the surface of Listeria monocytogenes', *Nature*, 385(6613), pp. 265–269. doi: 10.1038/385265a0.

Welsh, K. *et al.* (2016) ‘Characterization of Potent SMAC Mimetics that Sensitize Cancer Cells to TNF Family-Induced Apoptosis’, *PLoS ONE*, 11(9), p. e0161952. doi: 10.1371/journal.pone.0161952.

Welz, T. and Kerkhoff, E. (2023) ‘The role of SPIRE actin nucleators in cellular transport processes’, *Journal of Cell Science*, 136(6), p. jcs260743. doi: 10.1242/jcs.260743.

Weston, L., Coutts, A. S. and La Thangue, N. B. (2012) ‘Actin nucleators in the nucleus: an emerging theme’, *Journal of Cell Science*, 125(15), pp. 3519–3527. doi: 10.1242/jcs.099523.

Williamson, M. P. (1994) ‘The structure and function of proline-rich regions in proteins’, *The Biochemical Journal*, 297 (Pt 2)(Pt 2), pp. 249–260. doi: 10.1042/bj2970249.

Williamson, M. P. (2023) ‘Protein Binding: A Fuzzy Concept’, *Life*, 13(4), p. 855. doi: 10.3390/life13040855.

Winder, S. J. and Ayscough, K. R. (2005) ‘Actin-binding proteins’, *Journal of Cell Science*, 118(Pt 4), pp. 651–654. doi: 10.1242/jcs.01670.

Winters, M. J. and Pryciak, P. M. (2005) ‘Interaction with the SH3 Domain Protein Bem1 Regulates Signaling by the Saccharomyces cerevisiae p21-Activated Kinase Ste20’, *Molecular and Cellular Biology*, 25(6), pp. 2177–2190. doi: 10.1128/MCB.25.6.2177-2190.2005.

Xue, B. and Robinson, R. C. (2013) ‘Guardians of the actin monomer’, *European Journal of Cell Biology*, 92(10–11), pp. 316–332. doi: 10.1016/j.ejcb.2013.10.012.

Zhang, S. and Vavylonis, D. (2023) ‘Steps of actin filament branch formation by Arp2/3 complex investigated with coarse-grained molecular dynamics’, *Frontiers in Cell and Developmental Biology*, 11, p. 1071977. doi: 10.3389/fcell.2023.1071977.

Zhao, H., Michelot, A., Koskela, E. V., Tkach, V., Stamou, D., Drubin, D. G. and Lappalainen, P. (2013) ‘Membrane-sculpting BAR domains generate stable lipid microdomains’, *Cell Reports*, 4(6), pp. 1213–1223. doi: 10.1016/j.celrep.2013.08.024.

Zuchero, J. B., Coutts, A. S., Quinlan, M. E., Thangue, N. B. L. and Mullins, R. D. (2009) ‘p53-cofactor JMY is a multifunctional actin nucleation factor’, *Nature Cell Biology*, 11(4), pp. 451–459. doi: 10.1038/ncb1852.

van Zundert, G. C. P. *et al.* (2016) ‘The HADDOCK2.2 Web Server: User-Friendly Integrative Modeling of Biomolecular Complexes’, *Journal of Molecular Biology*, 428(4), pp. 720–725. doi: 10.1016/j.jmb.2015.09.014.

9. Appendix: Mathematical validation

An important step following the outlining of the model was to investigate the system mathematically. This helped verify that the phenomena I aimed to observe were (at least theoretically) possible. Two cases were considered for model application – “ligand depletion” and “ligand constant”. The case of “ligand depletion” states that the multi-SH3 domain (Sla1) can be removed from the pool of available agents following dissociation from Las17 through interaction with a sequestering agent (e.g., cargo, ubiquitin, etc.). Cloud SH3 domains may help facilitate this process through competitive binding. This system did not require mathematical validation as removal of Sla1 would obviously increase Las17 activity and thus can easily be represented within Model Alpha.

However, the “ligand constant” case was less obvious and required mathematical exploration to assess its possibility. A ligand constant system is analogous to a pyrene assay in which the inhibition relief of Sla1 by Ysc84 has been observed within the Ayscough lab. In this system, competition between cloud domains and Sla1 can promote nucleation through SH3-SH3 competition reducing the percentage of bound Sla1. In the ligand constant case, Sla1 is not removed from the system via sequestration and remains able to re-bind Las17. Therefore, this mechanism was mathematically assessed prior to building Model Alpha to assess whether cloud competition in a ligand constant state can theoretically reduce Sla1 inhibition.

Exploration of the ligand constant case began by outlining the interactions from the perspective of cloud domains binding the third actin-binding Las17 polyproline tract (PP4). Below shows this visually and assumes independent binding between the sites and one of the Sla1 SH3 domains binding at a lower affinity than the first (**figure 1A**). Note that the Las17 binding sites can bind both actin and cloud SH3s. Changing the binding partners (i.e., clouds binding PP3 rather than PP4) essentially creates a mirror image of what is being shown in **figure 1A** whilst preserving the same ratios of rows in the figure and therefore can be omitted for mathematical simplicity. All mathematical analysis in this section was completed with the aid of Dr Jeremy Craven.

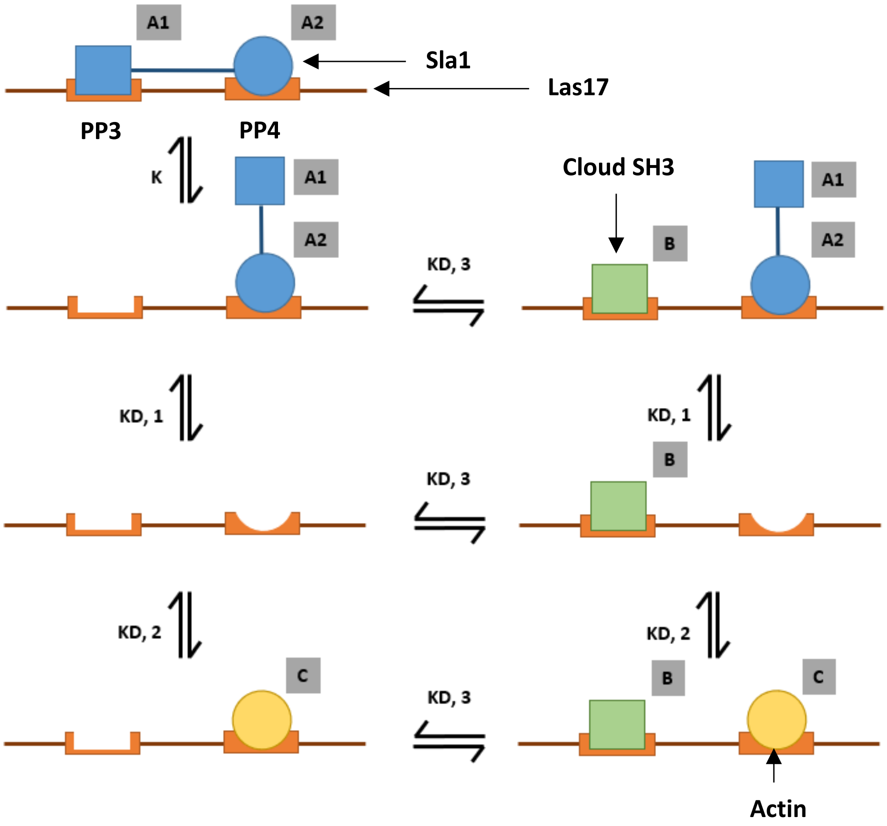


Figure 1A: Visualising the competing interactions between actin, Sla1, and cloud SH3s with respect to actin binding PP4. Here, the orange represents the major Sla1-binding region of Las17 covering the PP3-PP4 region (polyproline tracts shown as orange U-shapes). Regarding the Las17 ligands: *A1* represents the second SH3 domain of Sla1, *A2* represents the first SH3 domain of Sla1, *B* represents a cloud SH3 domain binding PP3 (e.g. Ysc84), whilst *C* represents actin. The K_d (or k) for each interaction is shown along with a number signifying that value (i.e., K_D 1 is the K_D value of a cloud SH3 binding to Las17 PP3, etc.).

Concentrations for each of the states can be represented mathematically using basic rate and dissociation constant equations as shown below (**figure 1B**). This allows us to combine and rearrange the expressions to investigate how certain components (i.e., the percentage of actin-bound Las17) changes in the presence, or absence, of cloud SH3 domains (element *B*).

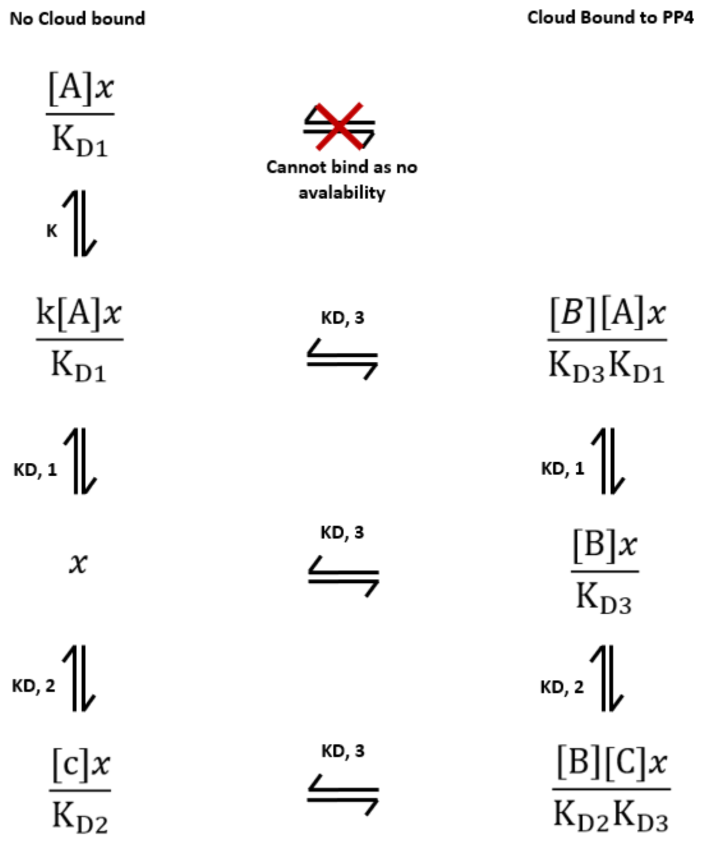

No Cloud bound
Cloud Bound to PP4
$\frac{[\mathrm{A}]x}{\mathrm{K_{D1}}}$
Cannot bind as no avalability
K
$\frac{\mathrm{k}[\mathrm{A}]x}{\mathrm{K_{D1}}}$
KD, 3
$\frac{[B][\mathrm{A}]x}{\mathrm{K_{D3}K_{D1}}}$
KD, 1
KD, 1
x
KD, 3
$\frac{[\mathrm{B}]x}{\mathrm{K_{D3}}}$
KD, 2
KD, 2
$\frac{[\mathrm{c}]x}{\mathrm{K_{D2}}}$
KD, 3
$\frac{[\mathrm{B}][\mathrm{C}]x}{\mathrm{K_{D2}K_{D3}}}$

Figure 1B: The competing interactions shown mathematically between actin, Sla1, and cloud SH3s with respect to actin binding PP4. Here, each visual ligand-binding state from **Figure 3.3** is replaced with an expression derived using basic rate and dissociation constant equations.

The above cloud-bound column can be simplified to investigate the effect of agent *B* (the cloud) in **figure 1A** using **equation 1A**. Combining all the equations above then gives the value for R_{TOT} (total Las17) as shown by the respective expression (**equation 1B**).

$$\beta = \frac{1 + [\mathrm{B}]}{K_{\mathrm{D3}}} \tag{1A}$$

$$R_{TOT} = \beta x + \frac{(\beta + k)[\mathrm{A}]x}{K_{\mathrm{D1}}} + \beta \frac{[c]x}{K_{\mathrm{D2}}} \tag{1B}$$

Rearranging the *R_{TOT}* formula to show the concentration of actin-bound Las17 (first solving for *x*) can be done in the following expression (**equation 1C**). This formula shows us how much Las17 would be actin-bound.

$$\beta \frac{[c]x}{K_{\mathrm{D2}}} = \frac{R_{TOT}[\mathrm{c}]}{K_{\mathrm{D2}}(1 + \frac{(\beta + k)[A]}{\beta K_{\mathrm{D1}}} + \frac{[c]}{\mathrm{K}_{\mathrm{D2}}})} \tag{1C}$$

To investigate how absence of the SH3 cloud changes this expression, we removed the PP3-binding cloud domain (*B* – within the β expression) from **equation 1C** which changes the expression to the following (**equation 1D**). This shows us how much Las17 would be actin bound if no cloud SH3s were present in the system.

$$\beta \frac{[c]x}{K_{\mathrm{D2}}} = \frac{R_{TOT}[\mathrm{c}]}{K_{\mathrm{D2}}(1 + \frac{(1 + k)[A]}{1K_{\mathrm{D1}}} + \frac{[c]}{K_{\mathrm{D2}}})} \tag{1D}$$

This shows that the concentration of Las17 with actin at the PP4 site can increase when originally sequestered by Sla1 and exposed cloud SH3s, provided that K>0 and β>1, as the expression yields a lower percentage of actin-bound Las17 when comparing **equation 1C** and **1D**. This conclusion would hold when mirrored for SH3s (*B*) binding PP4 and actin (*c*) binding PP3 as the ratio between rows 2-4 in **figure 1B** would remain the same.

This mathematical analysis showed that adding more cloud domains can increase the overall percentage of actin bound to Las17 (particularly when considering cooperativity between bound actin monomers) through reducing the degree of Las17 sequestered in row 1 (relieving Sla1 inhibition) and therefore leaving more to be redistributed into rows 2-4. It should be noted that this mathematical expression was a significant simplification of the model. These equations reveal that cloud SH3s can theoretically reduce Sla1 inhibition in a ligand constant state. However, this was not guaranteed and only accounted for part of the regulatory cloud hypothesis. Our agent-based model allowed us to further test this hypothesis and expand our understanding of the mechanism.