A Symmetry Approach to Virus Architecture

Jessica Paige Wardman

PhD

University of York

Biology

July 2012

Abstract

The structure and symmetry of viruses has been the subject of study since Crick and Watson in 1956, and there have been several complementary theories describing different aspects of the geometry of these complicated entities. Included here is a unified theory that relates the structure and sizes of the different viral components, from the capsomeres to the packaging of the genomic material, providing, through a set of structural constraints on viral structures, a new classification scheme for viral structures. Moreover, aspects of this theory also apply to fullerene structures in chemistry, showing that this symmetry principle is deeper than just biological in nature.

Contents

List of Figures

List of Tables

Acknowledgements

There really are too many people to thank here, so in no particular order, I would like to mention my supervisor, Professor Reidun Twarock, who is a great driving force for research and science; Dr Tom Keef, a good friend, a very patient explainer and, in many respects, a mentor; David Salthouse, good friend and colleague, who has provided a sounding board for me to chatter at while I refine my ideas and Tom Horrocks, Carl Miller and Frances Lee, for sitting patiently while I fumble through the explanations of how viruses look roughly like footballs and why this is important and amazing.

Lastly, but the very opposite of leastly: my fiancée Dr Adele Taylor, for supporting me in the bad times, making sure I look after myself in the good times, and for the endless patience while I chatter, muse, dream and endlessly type; and my wonderful son, Myrddin, aged $8\frac{1}{2}$, who is only too able to send me off down a different track by pointing at something and asking "What's that?".

Author's Declaration

The *Introduction* is purely a run-down of previous literature; *Constructing Templates for Virus Architectures* is a revamp of work carried out previously by Keef and Twarock in [42] and [43], although this work takes a slightly more unified linear algebraic approach instead of a group theoretic one; the remaining four chapters are all original work, although parts of *The Best-fit Algorithm*, *Applications to Viruses* and *Conclusions* are presented in [44] (being that certain of the point arrays match well to Pariacoto Virus and Bacteriophage MS2) and [46], where the algorithm is discussed, along with its application to Pariacoto Virus, Bacteriophage MS2 and Simian Virus 40.

Chapter 1

Introduction

1.1 Virus Sizes and Shapes

Viruses are a marvel of biology; at the opposite end of the biological scale spectrum to humans and other animals, they are still highly complex entities. Described as the "most abundant biological entities on the planet" [7] they exist at a large variety of scales, from the tiny Satellite Tobacco Mosaic Virus [57] at 88Å radius (Figure 1.2), through Chilo Iridescent Virus at 925Å radius [118] (Figure 1.3) up to the (as of writing) largest virus known, Mimivirus [49][117] at a radius of approximately 3750Å (Figure 1.4). Figure 1.1 shows a number of viruses to scale with one another.

But not only do viruses exist at a large variety of scales, they come in several classes: *symmetric* (*icosahedral*), like the three viruses already mentioned; *helical*, such as Tobacco Mosaic Virus [73][89] (Figure 1.5); *enveloped*, such as HIV [75][80] (Figure 1.6); and *complex*, like Bacteriophage T4 [61] (Figure 1.7).

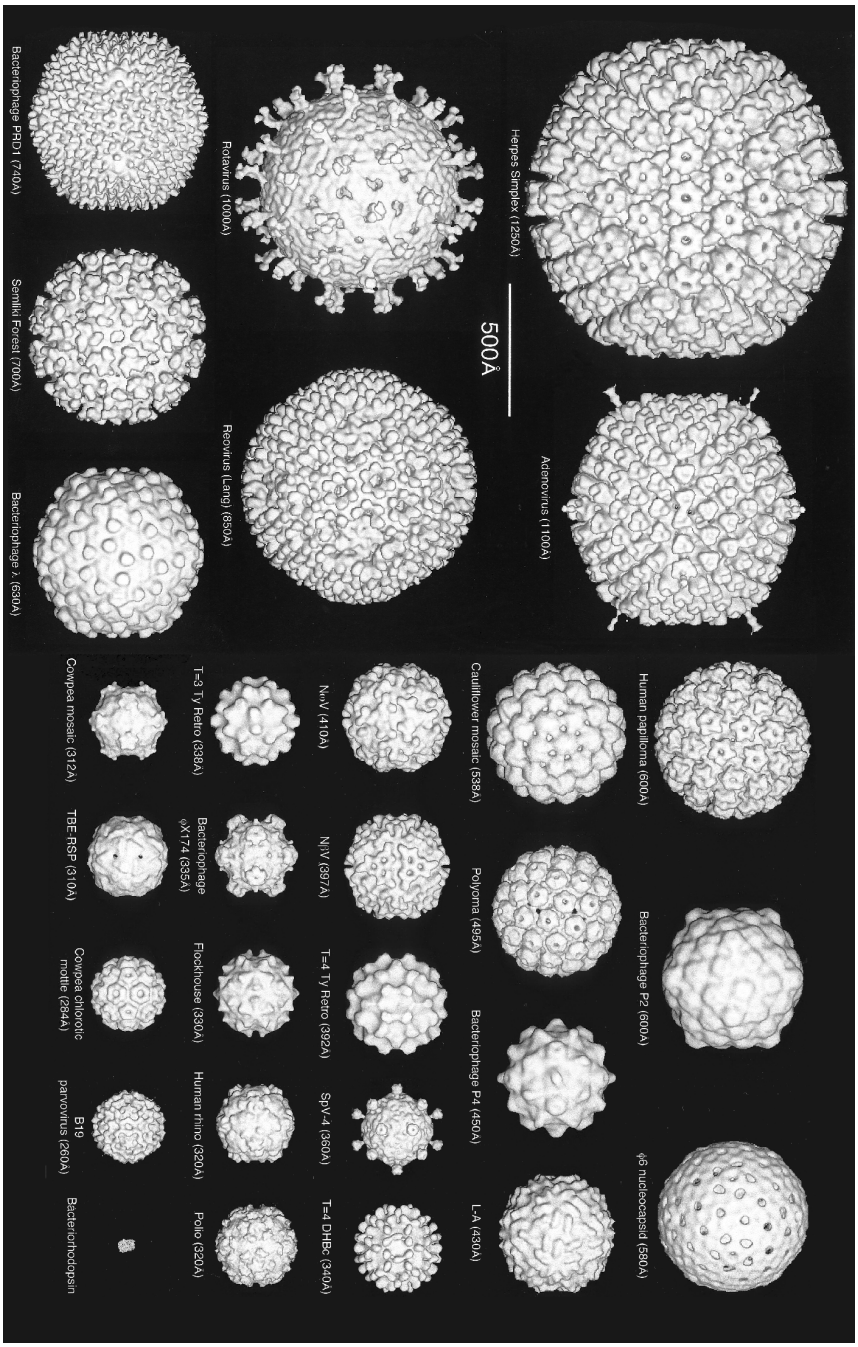


Figure 1.1: The ubiquity of icosahedral symmetry in viruses [2].

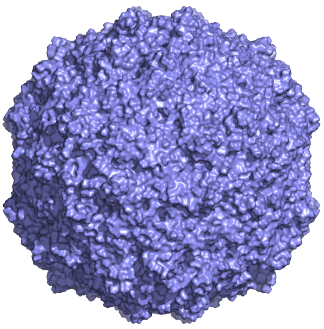
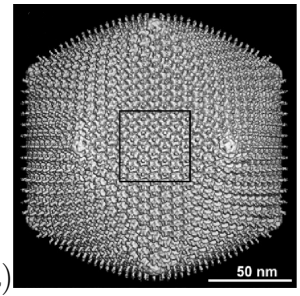


Figure 1.2: Satellite Tobacco Mosaic Virus [57] rendered with surface representation in PyMol [92] viewed down a 2-fold axis.

(a) 

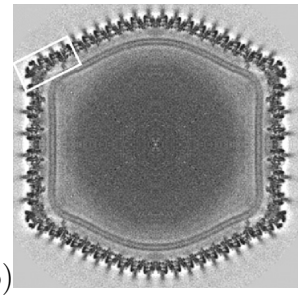
(b) 

Figure 1.3: Chilo Iridescent Virus electron density viewed down a 2-fold axis in entirety (a) and a central section (b) [118].

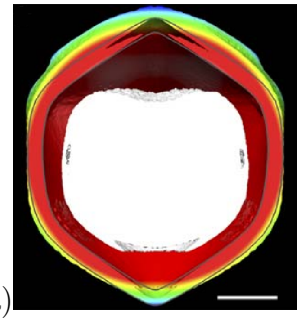
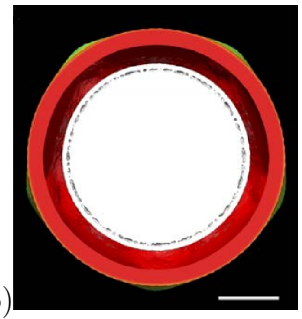
(a) 
(b) 

Figure 1.4: Sections of Mimivirus showing its size (scale bar is 1000Å). Mimivirus mostly exhibits icosahedral symmetry, although there is a unique 5-fold vertex similar to that in Bacteriophage MS2. (a) The slice of Mimivirus down a 2-fold axis perpendicular to the unique 5-fold vertex, with a grey icosahedron superimposed. (b) The view along the 5-fold axis opposite the unique 5-fold vertex [49, 117].

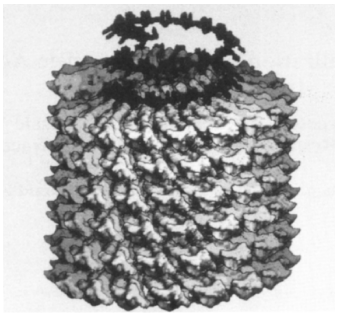
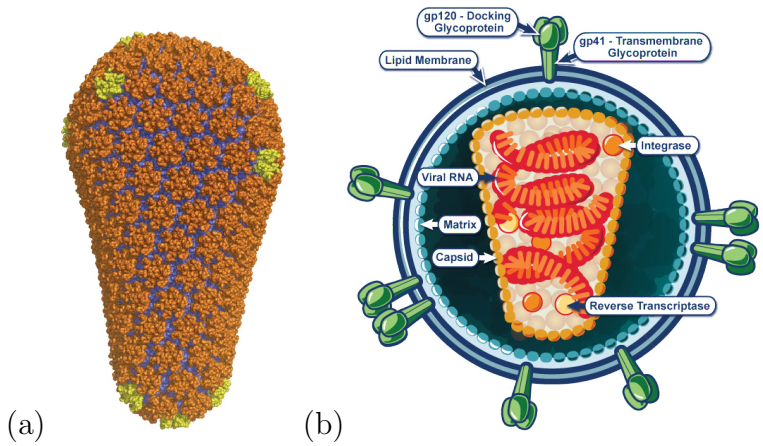


Figure 1.5: Computer representation of TMV with two turns of RNA protruding [73].



(a) (b)

Figure 1.6: (a) HIV capsid with hexamers (orange), pentamers (yellow) and dimers (blue) [80]. (b) A schematic diagram of the HIV capsid and its envelope [75].

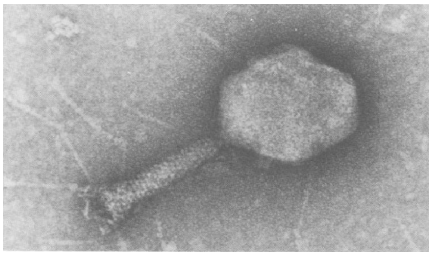


Figure 1.7: Electron micrograph of negatively stained Bacteriophage T4 [47].

We can see in these figures that these viruses display high degrees of symmetry: TMV repeats one type of coat protein to build the shell around its genome; HIV has three types of protein complexes in the capsid, making clusters of 6, 5 and 2 proteins (known as hexamers, pentamers and dimers respectively), and the head of Bacteriophage T4 is highly symmetric (indeed, as is the tail, albeit with a lesser level of symmetry). The reason that viruses have so much symmetry is the *principle of genetic economy* [11]. If every protein in an icosahedrally-symmetric viral capsid had to be coded for individually, the genomes would have to be 60 times longer at least. Using symmetry, viruses can code for fewer proteins but build a capsid the same size; icosahedral symmetry is particularly good for this as it is the largest finite group in three dimensions [108] and therefore has more repeats of the *asymmetric unit* (also called the *fundamental domain* or *unit cell*) in the capsid. This idea, that viruses repeat small units over their capsid, was first proposed by Crick and Watson [11] and examined by its application to a large number of specific viruses by Horne [33], but then refined into the *theory of quasi-equivalence* by Caspar and Klug [9] and Coxeter [10].

1.2 Previous Frameworks

1.2.1 Quasi-Equivalence

The fundamental idea of quasi-equivalence is that each protein is in approximately the same local environment as any other; something especially important when different conformers[1] of the same protein fill all of the different positions on the capsid. This is achieved by an idea akin to the unfolding of a cube to form a section of a square lattice. In the same way, an icosahedron can be unfolded into a section of a triangular lattice (see Figure 1.8). Quasi-equivalence demands that the individual subunits, originally expected to be minor variations of the protein structure, lie in positions that can be described by sub-triangulating the faces of the icosahedron.

This is achieved by placing the unfolded icosahedral net on a smaller-scale triangular lattice, the only constraint being that the vertices of the net match the vertices of the lattice. This requirement ensures that when the net is refolded into an icosahedron, the edges of the triangles match up correctly (for an example of a $T = 4$ triangulation of an icosahedron with corresponding net, see Figure 1.9). Examples of how a single face of the net can be superimposed on a triangular lattice are shown in Figure 1.10.

Inequivalent superpositions, that is, superpositions corresponding to different viral configurations, are described via their T-number, although, as we shall see, this is not completely sufficient. The T-number

[1] A particular way of folding a polypeptide chain is called a *conformer* of that protein.

of a triangulation is defined by how the triangles in it correspond to the underlying triangular lattice (as in Figure 1.10); the coordinates (a, b) of one side of one triangle are found in terms of the sides of the triangles in the lattice (see Figure 1.11(a) for $T = 3$), and then T is calculated from

$$T = a^2 + ab + b^2. \tag{1.1}$$

As noted by Goldberg [24], though, the T-number is not sufficient to determine the capsid structure completely; for example, $7^2 + 7 \times 0 + 0^2 = 49 = 5^2 + 5 \times 3 + 3^2$, and so viruses with T-number 49 come in two different forms. Moreover, viruses with $a = m$ and $b = n$ are *not* the same as $a = n$ and $b = m$; they are mirror images of one another. In some cases, even, viruses (notably Hepatitis B) come in two sizes of capsid (in the case of Hepatitis B, these are $T = 3$ and $T = 4$ [12, 109]).

An example of how proteins can be distributed with respect to the symmetry axes of the icosahedron is shown in Figure 1.12, where representatives of the three conformers of the capsid proteins are shown along with the nearby symmetry axes.

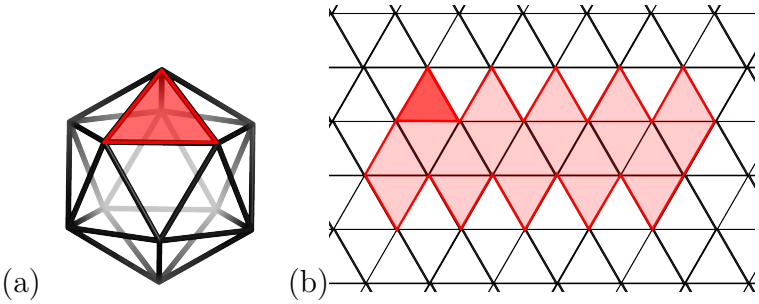


Figure 1.8: An icosahedron (a) and the same icosahedron cut along its edges and laid flat on a triangular lattice (b).

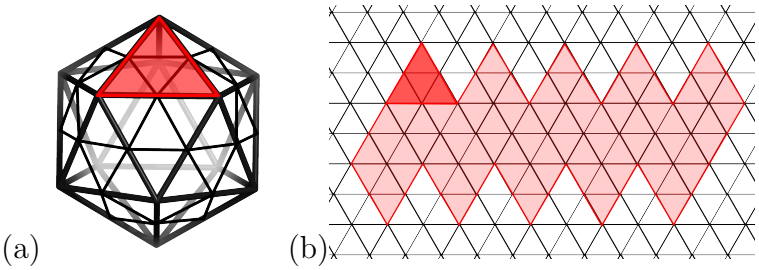


Figure 1.9: An icosahedron triangulated with $T = 4$ (a) and the same icosahedron cut along its edges and laid flat on a triangular lattice (b).

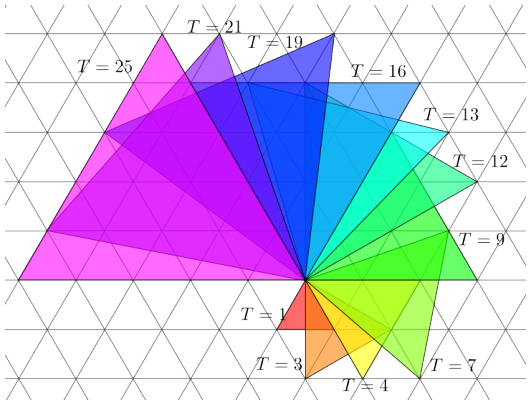


Figure 1.10: A diagram of the triangles on a triangular lattice corresponding to the allowable triangulation numbers between 1 and 25.

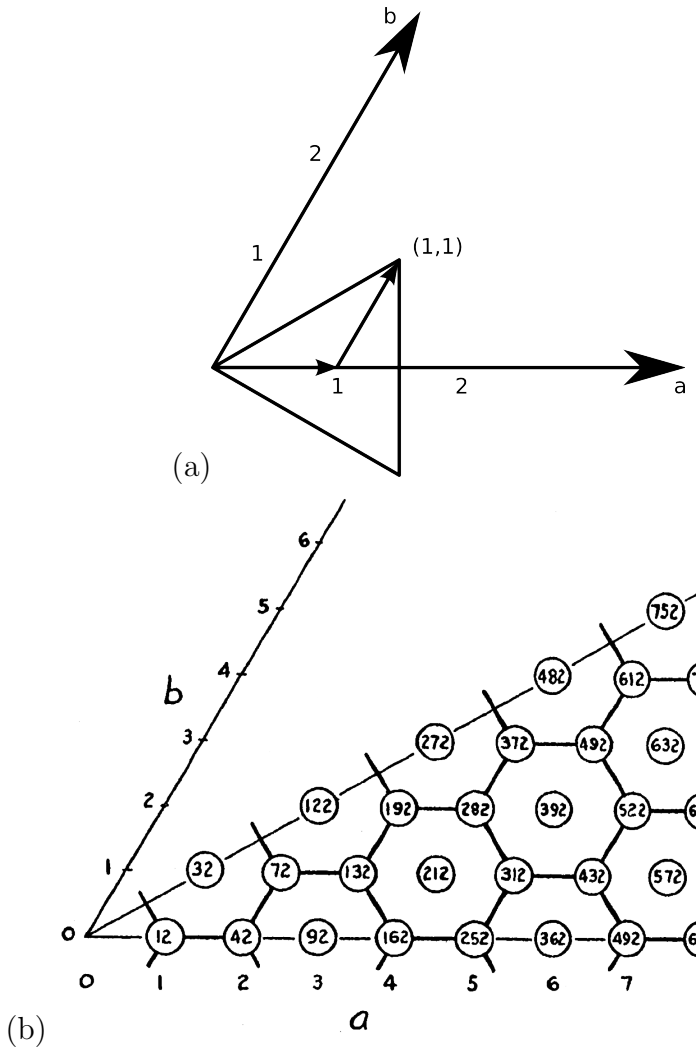


Figure 1.11: (a) A $T = 3$ triangulation is found when $a = b = 1$. (b) A graph showing the number of pentameric and hexameric patches corresponding to the possible T-numbers [24] showing the $T = 3$ triangulation has 32 total patches — 12 pentagonal and 20 hexagonal.

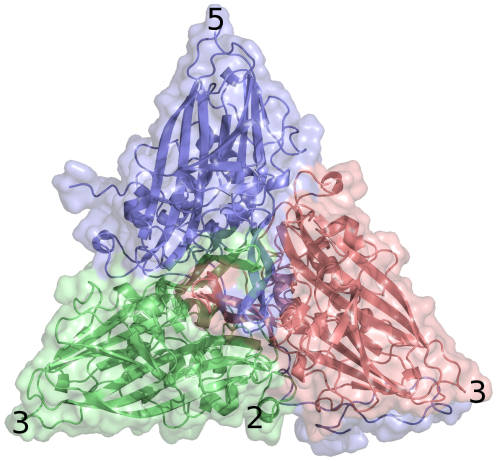


Figure 1.12: A trimer of Pariacoto Virus with the global symmetry axes marked showing the local symmetry axis in the middle of the trimer.

Symmetry is not the only selective pressure on virus capsids. Mannige and Brooks [69] showed how the shapes of the main body of the proteins in the capsid are restricted to trapezoidal shape, at least in eight out of twelve viral families studied when only different conformers of the same protein make up the capsid. Mannige and Brooks [71] then also showed that viruses tended to prefer lower *hexamer complexity* — that is, viruses prefer to have a lower number of distinct types of hexamers. This quality of hexamer complexity is dependent only on a and b from equation (1.1).

Extreme cases of icosahedral viruses having a low number of distinct types of hexamers are those viruses formed of more than 12 pentamers such as Polyoma Virus [82] which has 72 pentameric structures rather than the 12 pentamers and 60 hexamers predicted by quasi-equivalence. This was analysed, along with other similar cases, by Rossmann [87] with the explanation that protein structures were inherently flexible, and as such in certain viruses were forced into pentameric locations rather than the hexameric positions expected by quasi-equivalence. A theory that predicts how this arrangement can happen is *viral tiling theory* [106], and such a virus is examined in Chapter 4, Section 4.3.2.

1.2.2 Viral Tiling Theory

Quasi-equivalence, as we have seen, is based on the idea of tiling icosahedral objects with equilateral triangles, with the number of triangles on each face of the icosahedron giving the T-number of that tiling (recall Figure 1.10). Viral tiling theory keeps the idea of a tesselation of the icosahedron, but relaxes the assumption that the tiles must be equilateral triangles, allowing any shape of tile, although kites, darts and rhombs (taking inspiration from Penrose tilings [78]) seem to be sufficient. Furthermore, the icosahedral symmetry remains, and the principle of quasi-equivalence that ensured that identical capsomeres were placed in structurally similar locations is extended to the *generalized principle of quasi-equivalence* [106] stating that "On any given tile protein subunits are located only at corners subtending the same angle". This generalises quasi-equivalence, which automatically satisfies this broader concept, as all proteins lie within equilateral triangles. Moreover, the tiles indicate the bonds between the capsomeres of the virus whether within a tile or across an edge. For the most part, those bonds within tiles are inter-capsomere and those across edges are intra-capsomere (a minor exception is Polio in Figure 1.13(c) where most intra-capsomere bonds are across edges except some between B and C chain proteins within the kite).

Some examples of this theory are shown in Figure 1.13, where three $T = 3$ viruses are shown along with their (different) tilings from viral tiling theory. Pariacoto Virus (Figure 1.13(a)) does follow quasi-equivalence fully — the tiles necessary are the expected triangles, and

each A chain protein around a 5-fold axis links to two different hexamers; Bacteriophage MS2 (Figure 1.13(b)) is also a $T = 3$ virus, but forms dimers that are best described by rhombs — one that contains A and B chain proteins and one that forms CC dimers; finally, Poliovirus (Figure 1.13(b)) is tiled by kites, and the A chain proteins in the pentamers each link to two proteins from the same hexamer.

Moreover, while quasi-equivalence predicts the correct layout of capsomeres for Simian Virus 40 (SV40), it incorrectly predicts 12 pentamers and 60 hexamers; SV40 has, instead, 72 pentamers, as can be seen in Figure 1.14(a). Viral tiling theory can account for this, tiling the capsid with kites (around the 5-fold axes) and rhombs (elsewhere) to give the observed 72 pentamers. Moreover, as the connections within the tiles illustrate, this approach correctly models the location of the bonds between the pentamers — much as happened with the $T = 3$ viruses of Figure 1.13.

However, viral tiling theory, albeit being more complete than quasi-equivalence, is still a surface theory and does not take into account any radial information.

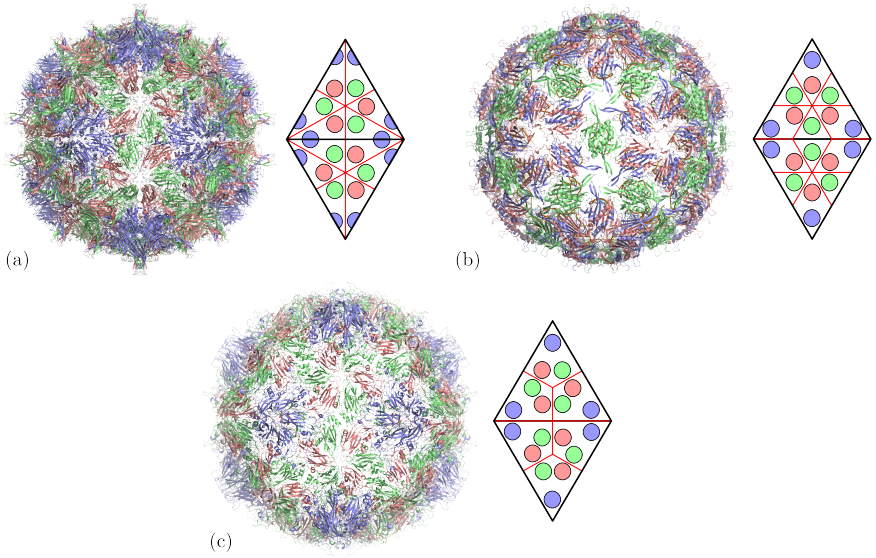


Figure 1.13: (a) Pariacoto Virus, (b) Bacteriophage MS2 and (c) Polio Virus are all $T = 3$ viruses; Pariacoto Virus is tiled with triangles fitting its trimers, in accordance with quasi-equivalence, MS2 is tiled with rhombs, as best describe its bonding pattern of dimers, and Polio is tiled with kites.

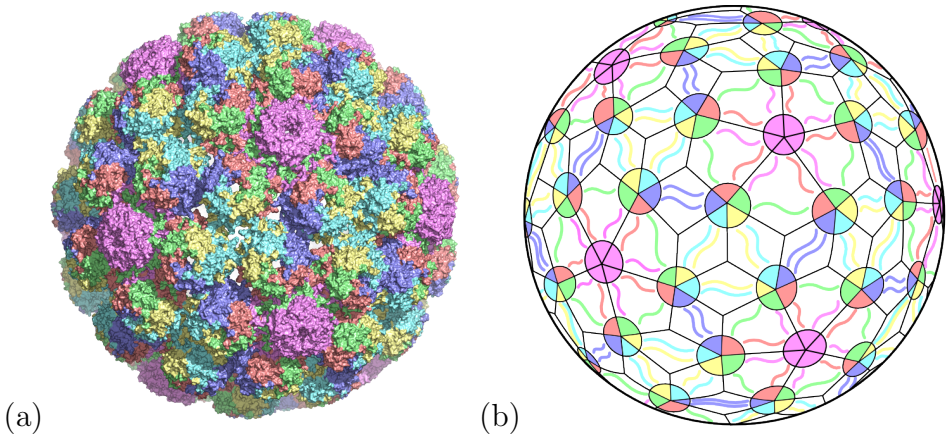


Figure 1.14: (a) The surface of Simian Virus 40 and (b) the tiling using kites and rhombs appropriate to the bonding pattern, explaining the two different pentameric clusters.

1.2.3 Packing Lattices

Janner [36, 37, 38, 39, 40] has made a start on a 3D theory, where he embeds the viral capsid into a *packing lattice*. Encasing polyhedra for the viruses are found in terms of the lattice points by visual inspection, and then these encasing polyhedra are subdivided so as to give proposed boundaries to the viral components. Such a subdivision of an encasing polyhedron for Pariacoto Virus focusing on the A chain protein is shown in Figure 1.15.

It is not clear how best to embed a virus within these lattices, given no three-dimensional lattice has icosahedral symmetry, nor which scaling such a lattice should be at. Here, we develop a three-dimensional approach based on quasi-lattices [45, 94] to continue Keef and Twarock's work with point arrays which encode optimal ways of how icosahedral symmetry may be realised at different radial levels simultaneously [42, 43]. These point arrays will be examined more closely in Chapter 2.

1.3 Available Data on Virus Structures

Useful tools of recent times for the study of viruses are *cryo-EM* and *X-ray crystallography*. Cryo-EM along with associated image processing techniques maps the electron density of a virusand the files containing the results can be downloaded from (for example) the EM DataBank [59] and viewed with software such as Chimera [79]. X-ray crystallography (first used on Tomato Bushy Stunt Virus (TBSV) [85]) interrogates the virus structure at a resolution typically around

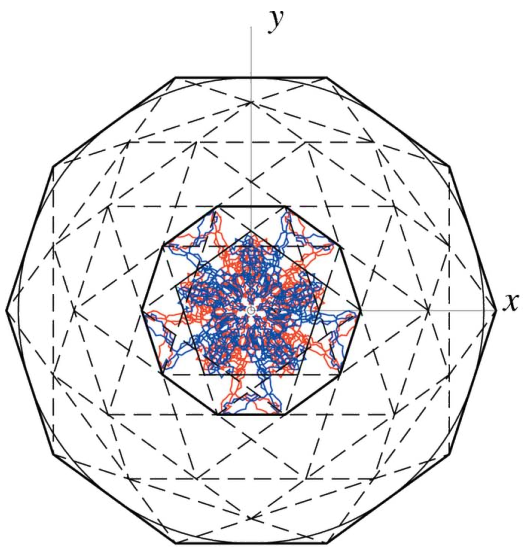


Figure 1.15: The encasing form of Pariacoto Virus viewed down the 5-fold axis, showing two pentamers of A chain protein, with a subdivision of the external decamer [39].

3Å and results, once a suitable model is constructed from the density obtained, in a `pdb`-file (as can be downloaded from VIPER [83] or the Protein DataBank [5]), listing the coordinates of the detected non-hydrogen atoms. Such a `pdb`-file can be displayed with one of several viewers (such as Chimera [23, 79] and PyMol [92]), and this level of detail allows for normal mode analysis — probing the dynamic properties of viruses — to be carried out in a fully atomistic way (eg [15]), as opposed to a more coarse-grained model (eg [77]).

As we shall see with carbon cages in Chapter 5, objects with icosahedral symmetry can undergo buckling transitions, and these are based on their *Foppl von Karman* number [64], defined as

$$\gamma = \frac{YR^2}{\kappa},$$

where R is the radius of the object under consideration, Y is the 2D Young's modulus (stiffness) and κ is the bending modulus. Mannige *et al.* have used this to show that such a transition is impossible for $T < 7$. There are precisely two types of capsid structure for $T = 7$, although the transition requires some small energy input. However, for $T > 7$ the situation is variable — a $T = 9$ virus, for example, may not be able to buckle at all, due to the layout of its proteins [70].

The investigation of viral capsids via these non-standard mathematical and biophysical means proceeds on many fronts: for example Zandi *et al.* [121] (following Goldberg [25]) demonstrated through simulation of capsomeres self-arranging around a sphere how certain numbers were favoured, recovering the particular T-numbers Caspar, Klug and Goldberg determined. Figure 1.16 shows the overall energy $\varepsilon(N)$ of a capsid of N capsomeres including one energy term taking into account interconversion between pentamers and hexamers, and another term for the relative movement of the capsomeres. They demonstrate that the local minima of $\varepsilon(N)$ occur at those configurations for which the organisation of the capsomeres corresponds to T-numbers.

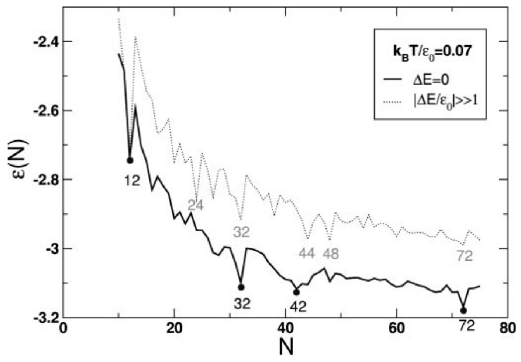


Figure 1.16: Internal Energy per Capsomer $\varepsilon(N)$ is a pronounced local minimum when N (number of capsomers) is 12, 32, 42 and 72, corresponding to $T = 1$, 3, 4 and 7 [121].

1.4 Viral Genomes

We have seen that viral capsids exhibit symmetry and high degrees of order in their organisation. Interestingly, the viral genomes within the capsids also show evidence of order [59, 93, 98, 104].

While higher life forms achieve complexity through many agents acting in concert to a common goal, the virus is a remarkable example of how one component can achieve a complex variety of functions. In particular, viral genomes must code for the correct proteins to form the protective capsid of the required T-number to surround a genome of that length [122], be of a structure amenable to packaging by those proteins [16, 28, 41, 102], be able to fold sufficiently well to fit in the limited space available [119], aid (in some cases) with conformal switching of the coat proteins to enable them to take up the appropriate quasi-equivalent structure [18, 113]; in some cases act as a scaffold for coat proteins to assemble around [4] as well as the primary function

of genomic material: allowing infection and reproduction. There is also evidence that stem-loop patterns are responsible for co-operative effects in virus capsid assembly in (ss)RNA viruses [14].

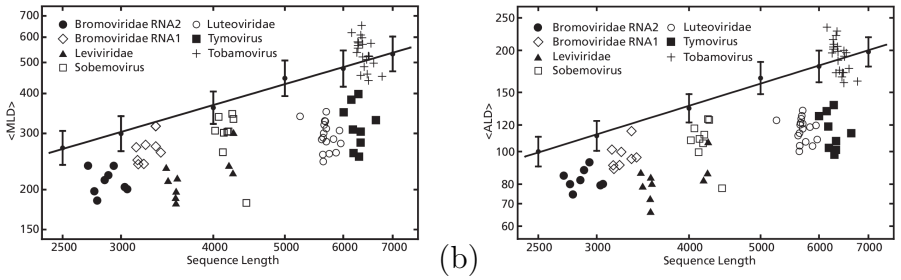


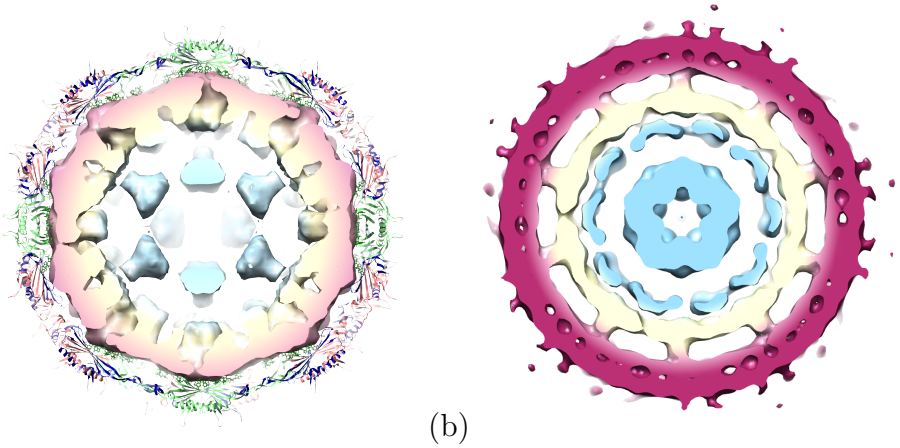
Figure 1.17: Log-log plot of maximum (a) and average (b) ladder distances across viral ssRNA (see legends) and random ssRNA [119].

Viral genomic material is especially good at folding into a smaller volume when compared to other (ss)RNAs [119]: Figure 1.17 shows that the *Ladder Distance* between two bases, LD_{ij} — the number of base pairs crossed when traversing the most direct path in the folded RNA that connects bases i and j counting only double-stranded sections — is lower for viruses (see the legends) than for random ssRNA (the line with error bars). That is, RNA from icosahedral viruses folds more efficiently than random ssRNA; Tobamovirus is rod-shaped, and therefore under no pressure to fold efficiently, hence the RNA folds no more or less efficiently than the baseline random RNA. It seems reasonable to assume that the same pressure would exist on viral DNA as well.

Moreover, in order to fit genomic material in such a confined space, even with its propensity to fold compactly, some viruses have to package their genomes under enormous pressures (up to 50 atmospheres!)

[22] which requires efficient packaging [48, 115], even if this packaging can lead to knots in the genomic material [66, 67, 88].

In addition to the patterns in the RNA sequence, there is ample evidence for structured features in the geometric organisation of the genomes as seen in cryo-EM images of Bacteriophage MS2 and Hepatitis B (see Figure 1.18); such structure is not evident in the `pdb`-file as this volume of the virus was not modelled.



(a) (b)

Figure 1.18: Cryo-EM structures of (a) Bacteriophage MS2 [59, 104] (viewed down a 2-fold axis) and (b) Hepatitis B [59, 93] (viewed down a 5-fold axis) show evidence of icosahedrally structured genomic cages.

1.5 Thesis Structure

Chapter 2 examines the point arrays encoding the quasi-lattice information, their construction and some immediate consequences.

The algorithm presented in Chapter 3 interrogates the available data for virus capsid proteins and calculates which point array is the most suitable description of the biology, and we show how the point arrays so chosen provide a prediction determining some of the structural constraints on virus architecture. Results of this applied to a selection of 11 viruses is presented in Chapter 4.

This thesis, while primarily concerned with viruses and their structure, also looks at how the mathematics used here to understand their architectures can be applied to fullerene structures; an introduction to fullerenes is given at the beginning of Chapter 5, which details the application of this method to these structures.

Lastly, the conclusions drawn from this work are given in Chapter 6, and more detailed information about the various point arrays used are in Appendix A.

This thesis is accompanied by a CD that contains the software developed in this thesis.

Chapter 2

Constructing Templates for Virus Architectures

We have seen in Chapter 1 that a large number of viruses of greatly differing sizes (with radii from 88Å to around 3750Å) exhibit icosahedral symmetry at a number of different radial levels. Previous theories have, as discussed, described the capsid layouts of icosahedral viruses to a greater or lesser extent, but none have yet incorporated radial information. Zandi *et al.* showed that the capsid layouts corresponding to T-numbers also map to local minima in energy functions [121] (as shown in Figure 1.16), so it is not inconceivable that viruses also make use of symmetry in its extended form to exploit minima in more general free energy landscapes.

Therefore, we need a mathematical tool to predict how different radial levels of viruses are organised. This can be achieved with an (affine) extension of the symmetry group, as we show in the next section.

2.1 The Basics

The problem of finding affine extensions of the symmetry group is related, from a mathematical point of view, to the construction of lattices (that is, infinite periodic structures). The idea, in a nutshell, is to use a base shape that encapsulates the underlying symmetry required (for example, using a hexagon for 6-fold rotational symmetry) and move it in space in a specific coordinated way (this is what the extended symmetry group encodes) to obtain a lattice (or lattice-like, for non-crystallographic symmetries) arrangement. For the hexagon mentioned, we obtain a lattice. For a pentagon, a lattice is not possible (See Figure 2.1), due to the *crystallographic restriction* [94, 91], which says that the point groups of 2-dimensional lattices must be of the order 2, 3, 4 or 6; the pentagon has order 5. However a similar construction is possible; that is, one that has long-range order but no periodicity.

In analogy to this 2-dimensional example, we start with different instantiations of icosahedral symmetry in three dimensions. The different possible types of tilings accommodating the affine extensions of the icosahedral group correspond to projections of 6-dimensional lattices, just as the Penrose tiling can be obtained via projection from a 5-dimensional lattice [94]. Since there are three Bravais lattices with icosahedral symmetry in six dimensions [63, 6] (the simple cubic, body-centred cubic and face-centred cubic lattices), we start with three related basic shapes: the icosahedron, the dodecahedron and the icosidodecahedron. These have vertices on (respectively) the 5-,

3- and 2-fold axes of symmetry (see Tables A.1, A.2 and A.3 — page 168), and correspond to a projection of the bases of the simple cubic, body-centred cubic and face-centred cubic lattices in six dimensions [35], respectively. Clearly, applying the 60 elements of the icosahedral rotation group to these maps the structures onto themselves (that is, the structures are invariant under icosahedral symmetry). We therefore extend icosahedral symmetry by allowing a single translation to be added. This translation is restricted to being along one of the 5-, 3- and 2-fold symmetry axes, and only certain lengths are possible as otherwise the resulting set of points would be trivial. That is, it is chosen so that there are fewer points in the extended point array than would be expected for a random translation, i.e. some of the points generated by the translation coincide at the same point in space (mathematically, the new group has non-trivial relations). These are the *allowable* translations. Figure 2.2 illustrates this.

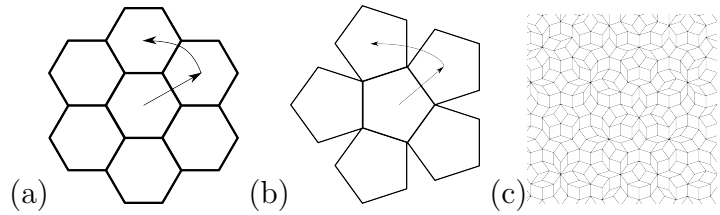


Figure 2.1: When translated, hexagons can form a periodic construction (a) but pentagons cannot, despite forming long-range order (b) in the same way as Penrose Tilings (c).

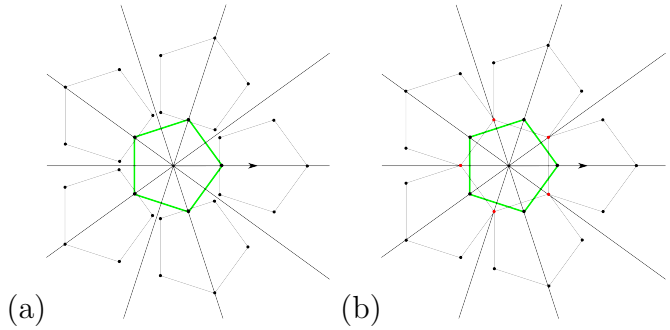


Figure 2.2: A pentagon translated (a) trivially and (b) non-trivially. Note that there are 30 points in (a) but only 25 in (b) as pairs of points coincide on the symmetry axes (marked in red).

2.2 A Reformulation of the Problem

Every point in an array can be expressed as $\mathbf{p}_i + \lambda\mathbf{t}_j$, where $\mathbf{p}_i$ is a vertex of the base shape, $\mathbf{t}_j$ is the translation vector and $\lambda \in \mathbb{R}$ is a multiplier measuring the length of the translation. What is important in this formulation, though, is the relative scaling of the base shape (given by $|\mathbf{p}_i|$) to the translation ($|\lambda\mathbf{t}_j|$). There are therefore two options: fix the base shape and scale the translation, which has been done previously in [42] and [43]; or to fix the translation and scale the base shape. These options are illustrated in Figure 2.3. This different viewpoint makes it possible to be more systematic in calculating all the allowable pairs of $\mathbf{t}$ and λ for each base shape. In Figure 2.4 each translated icosahedron is given by 12 points expressed as $\mathbf{p}_i/\lambda + \mathbf{t}_j$ (the origin is the cyan point in the middle of the image).

In that figure are four sets of icosahedra viewed down a 2-fold axis, translated by the same amount (along 5-fold axes), but each scaled differently. In essence, as λ decreases, the icosahedra "grow" from the four centres (which, in this case, would be four of the 12 vertices of an unscaled un-translated icosahedron), and so the only places their points can possibly intersect are along the planes bisecting the line connecting two of the adjacent centres — these planes being denoted by the dotted lines — and the three translations (blue, cyan and chartreuse) that have points that hit these planes.

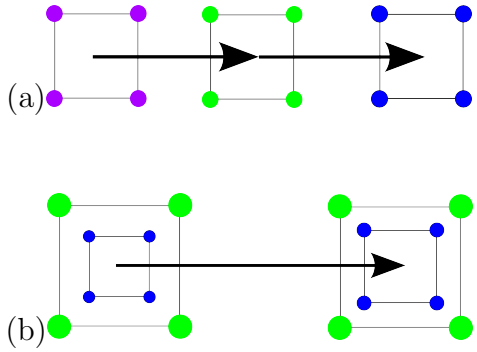


Figure 2.3: For the geometry of the construction, only the relative size of the the base shape versus the translation length is important; if both are scaled simultaneously, the same result is obtained. The shorter translation in (a) is the same as the large squares in (b), while the longer translation corresponds to the small squares.

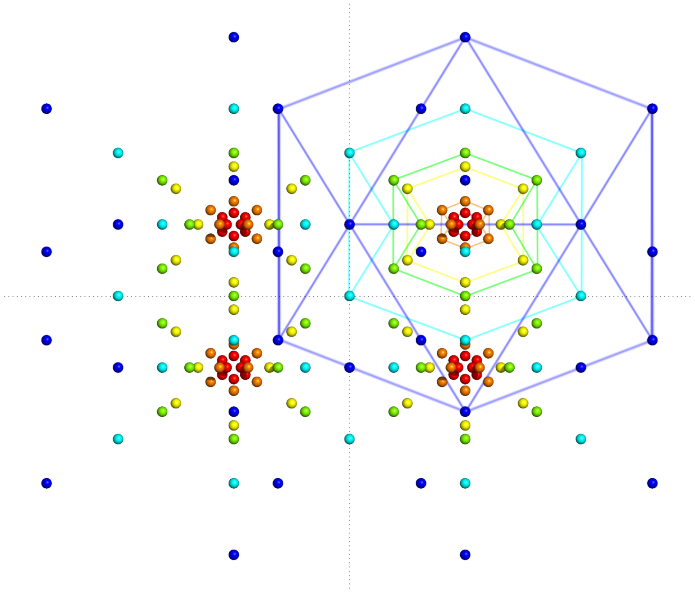


Figure 2.4: Icosahedra scaled by $1/\lambda$ for $\lambda = 10$ (red), $\lambda = 5$ (orange), $\lambda = 2$ (yellow), $\lambda = \tau$ (chartreuse), $\lambda = 1$ (cyan) and $\lambda = \tau - 1$ (blue) viewed down a 2-fold axis and how they fit together when translated along 5-fold axes.

Figure 2.5 shows a few of the symmetry planes and axes laid out diagrammatically, showing the relationships between them. Note in particular that only two of the three types of planes need to be checked — the plane between a 5-fold and a 3-fold axis is exactly that between a 2-fold and a 3-fold.

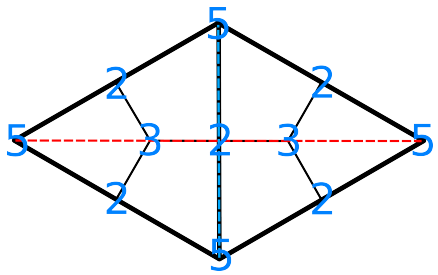


Figure 2.5: Only two symmetry planes need to be checked for coinciding points, appearing here as lines because of the projection. The red plane (the horizontal line) checks intersections between two adjacent 5-fold axes and two adjacent 2-fold axes and the blue plane (the vertical line) checks intersections between two adjacent 3-fold axes.

2.3 Finding Allowable Translations

To calculate the intersection of the vertices of the translated base shape and a symmetry plane, the equations of the two compenents are needed. The equation of a plane is $(\mathbf{n} - \mathbf{n}_0) \cdot \mathbf{x} = \mathbf{0}$ (where $\mathbf{n}$ is the normal to the plane and $\mathbf{n}_0$ is any point on that plane), and that of a line is $\mathbf{x} = \mathbf{x}_0 + \lambda\mathbf{t}$ where $\mathbf{t}$ is the direction of the line and $\mathbf{x}_0$ is any point on that line. In both cases, $\mathbf{x}$ is a general point on the plane and the line respectively. For these to intersect, $\mathbf{x}$ must satisfy both of these equations. In this case, $\mathbf{n}_0 = \mathbf{0}$ (the plane spanned by the symmetry axes goes through the origin), $\mathbf{n} = \mathbf{a} \times \mathbf{b}$ (the normal

vector is the cross product of two vectors spanning the plane), $\mathbf{x}_0 = \mathbf{p}_i$ (a general point on the line is one from the base point) and $\mathbf{t} = \mathbf{t}_j$ (the line is in the translation direction). This gives

$$(\mathbf{a} \times \mathbf{b}) \cdot (\mathbf{p}_i + \lambda \mathbf{t}_j) = \mathbf{0}$$

which rearranges to

$$\lambda = \frac{-\mathbf{p}_i \cdot (\mathbf{a} \times \mathbf{b})}{\mathbf{t}_j \cdot (\mathbf{a} \times \mathbf{b})}, \tag{2.1}$$

where $\mathbf{p}_i$ is the base point being translated, $\mathbf{t}_j$ is the translation vector, and $\mathbf{a}$ and $\mathbf{b}$ are the axes of symmetry determing the plane of intersection. This is the multiplier in $\mathbf{p}_i + \lambda \mathbf{t}_j$ and hence determines the length of the translation with respect to the size of the base shape. This process is carried out for each point $\mathbf{p}_i$ in the base shape (i.e. either icosahedron, dodecahedron or icosidodecahedron — see points listed in Tables A.1, A.2 and A.3 (page 168)) and one translation vector $\mathbf{t}_j$ from each of the icosahedron, dodecahedron or icosidodecahedron (as symmetry ensures any more is unnecessary). The allowable symmetry planes consist of all planes containing "adjacent" 2- and 3- fold axes or "adjacent" 2- and 5-fold axes as discussed earlier and shown in Figure 2.5. Adjacency is required to only check intersections on the edges of the kite — other planes between symmetry axes are not symmetry planes of the icosahedron.

A program implementing this algorithm (in Symbolic Python [100, 114]) is available as

`standard-symplane_normal_translations-sympy.py`.

	Two fold	Three fold	Five fold
Icosidodecahedron	11 (5,11)	8 (5,8)	6 (6,6)
Dodecahedron	8 (5,8)	5 (3,4)	4 (4,4)
Icosahedron	6 (6,6)	4 (4,4)	3 (3,3)

Table 2.1: The distribution of point arrays for the three base shapes by this method and, in brackets, by [42] and [43] respectively.

The allowable point arrays determined with this method for each of the three base shapes are given in Tables A.6, A.7 and A.8 in Appendix A (page 172). Observe that as the number of points in the base shape increases, the number of allowable point arrays with that base shape also increases, from 13 to 17 to 25. This set of 55 point arrays is referred to as the *library* of *pure* point arrays.

Note that this search over all points in the base shape is slightly inefficient, as not all of them need to be checked — they naturally form equivalence classes based on their orientation with the translation vector. Computing these equivalence classes to restrict the search space does not decrease the overall computing effort, though, as it would take longer to compute the classes than would be saved by knowing them. It is, however, interesting to note that this provides an explanation for the numbers of allowable point arrays and why they are not simply divisors or multiples of 12, 20 and 30, as, for example, translating an icosahedron along a 5-fold axis breaks the 12 points into equivalence classes with 1, 5, 5, and 1 members.

Table 2.1 lists the numbers of ways each base shape can be translated meaningfully along a symmetry axis, and in brackets are the corresponding figures from [42] and [43]. It can be seen that this approach is more exhaustive, because it also considers those points that

have multiplicity due to their locations on symmetry planes rather than just symmetry axes — those missed in [42] (that is, those that are formed when the base shape meets a symmetry plane rather than a symmetry axis) are those starred in the tables in the Appendix; the one unfortunately missed by [43] is double-starred. The software used to compute these 55 point arrays is available as

`standard_normal_cloud_generation.py` on the attached CD.

It can be seen that Table 2.1 is symmetric along the top-left to bottom-right diagonal — that there are, for example, as many point arrays found by translating an icosahedron along a 3-fold axis as there are by translating a dodecahedron along a 5-fold axis. The reasons for this will be explored in Section 2.4.

2.4 Direct Consequences

Keef and Twarock calculate their point arrays have 26 distinct outsides (defined there as the *outer layer* which is "those points with the largest distance from the centre" [43]), the reduction in number coming from the fact that two combinations of translations can have results that are identical, given scaling. That is:

$$\mathbf{a} + c\mathbf{b} = l(\mathbf{b} + d\mathbf{a}) \tag{2.2}$$

where $\mathbf{a}, \mathbf{b}$ are vectors pointing to vertices of the three base shapes and $c, d, l \in \mathbb{R}$. This is made a little clearer in Figure 2.6, which shows the general idea (in this example, $d = l = 2$ and $c = 1/2$).

Moreover, it can clearly be seen that c and d must be reciprocals

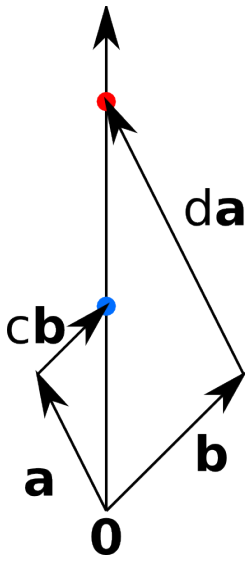


Figure 2.6: The red and blue dots correspond to two related combinations of translation and base shape and generate the same polyhedral shape under the action of the symmetry group.

of one another ($|d\mathbf{a}|/|\mathbf{a}| = |\mathbf{b}|/|c\mathbf{b}|$, so $cd = 1$), and that l is equal to one of a and b. This provides an explanation for the symmetric property of Table 2.1 and, moreover, makes precise which point arrays are paired with which, and hence are identical (excluding the different base shapes). Further to the 18 pairs of arrays generated in this fashion, there are 19 others. However, those 19 arrays that exist on the diagonal also match up in this way (that is, where an icosahedron is translated along a 5-fold, etc.) to produce 8 more pairs (1 from the icosahedron along a 5-fold, 2 from the dodecahedron along a 3-fold and 5 from the icosidodecahedron along a 2-fold). In fact, every array is paired with another, except for 12, 24 and 36, which are in some sense paired with themselves (having a translation length of 1); this produces a set of arrays that have 29 distinct exteriors. Arrays

No.	Start	Axis	Amount	No.	Start	Axis	Amount
1	Icos	2	$-1+\tau$	55	IDD	5	τ
2	Icos	2	$4-2\tau$	54	IDD	5	$1/2+\tau/2$
3	Icos	2	1	53	IDD	5	1
4	Icos	2	$-2+2\tau$	52	IDD	5	$\tau/2$
5	Icos	2	2	51	IDD	5	$1/2$
6	Icos	2	2τ	50	IDD	5	$-1/2+\tau/2$
7	Icos	3	$-1+\tau$	30	Dodec	5	τ
8	Icos	3	1	29	Dodec	5	1
9	Icos	3	τ	28	Dodec	5	$-1+\tau$
10	Icos	3	$1+\tau$	27	Dodec	5	$2-\tau$
11	Icos	5	$-1+\tau$	13	Icos	5	τ
14	Dodec	2	$2-\tau$	49	IDD	3	$1+\tau$
15	Dodec	2	$-6+4\tau$	48	IDD	3	$1/2+\tau$
16	Dodec	2	$-1+\tau$	47	IDD	3	τ
17	Dodec	2	$4-2\tau$	46	IDD	3	$1/2+\tau/2$
18	Dodec	2	1	45	IDD	3	1
19	Dodec	2	$-2+2\tau$	44	IDD	3	$\tau/2$
20	Dodec	2	2	43	IDD	3	$1/2$
21	Dodec	2	2τ	42	IDD	3	$-1/2+\tau/2$
22	Dodec	3	$2-\tau$	26	Dodec	3	$1+\tau$
23	Dodec	3	$-1+\tau$	25	Dodec	3	τ
31	IDD	2	$-1/2+\tau/2$	41	IDD	2	2τ
32	IDD	2	$2-\tau$	40	IDD	2	$1+\tau$
33	IDD	2	$1/2$	39	IDD	2	2
34	IDD	2	$-1+\tau$	38	IDD	2	τ
35	IDD	2	$\tau/2$	37	IDD	2	$1/2+\tau/2$

Table 2.2: Each point array in the left hand column has an identical exterior to the corresponding point array in the right hand column.

identical in this regard are indicated in Table 2.2.

Additionally, when combining point arrays as in Section 2.5.1, we can see that certain point arrays will occur as the outer point array more than others, which further constrains the possibilities. This is understandable, because the smaller the translation multiplier, the more likely that array is to be on the outside of a combination. Furthermore, not only does the translation length indicate how likely a

given array is to be an exterior (for example, array 31 has a translation length of only $-1/2 + \tau/2 = 0.309$ and is the exterior in 25 combination arrays) but how 'overlapping' those arrays are. That is, how large the 3-dimensional annulus containing points from both point arrays is; it does not mean that points have to coincide. For example, array 31 can be combined with array 6 (they are both translated along a 2-fold axis), but the outermost points of the re-scaled array 6 lie closer to the origin than the innermost points of array 31. In fact, they do not overlap at all. Meanwhile, arrays 6 and 41 can be combined, but have the same translation multiplier and hence overlap almost completely. A graph showing the point arrays and their radii (after scaling for combining) is shown in Figure 2.7. This phenomenon could well have implications for the interpretation of these point arrays for viruses. Notably, if a virus capsid only overlaps with the outer point array, no information is gained about the interior of the virus (such as genome organisation), as many point arrays may fit. Examples of this are analysed in Section 4.4.

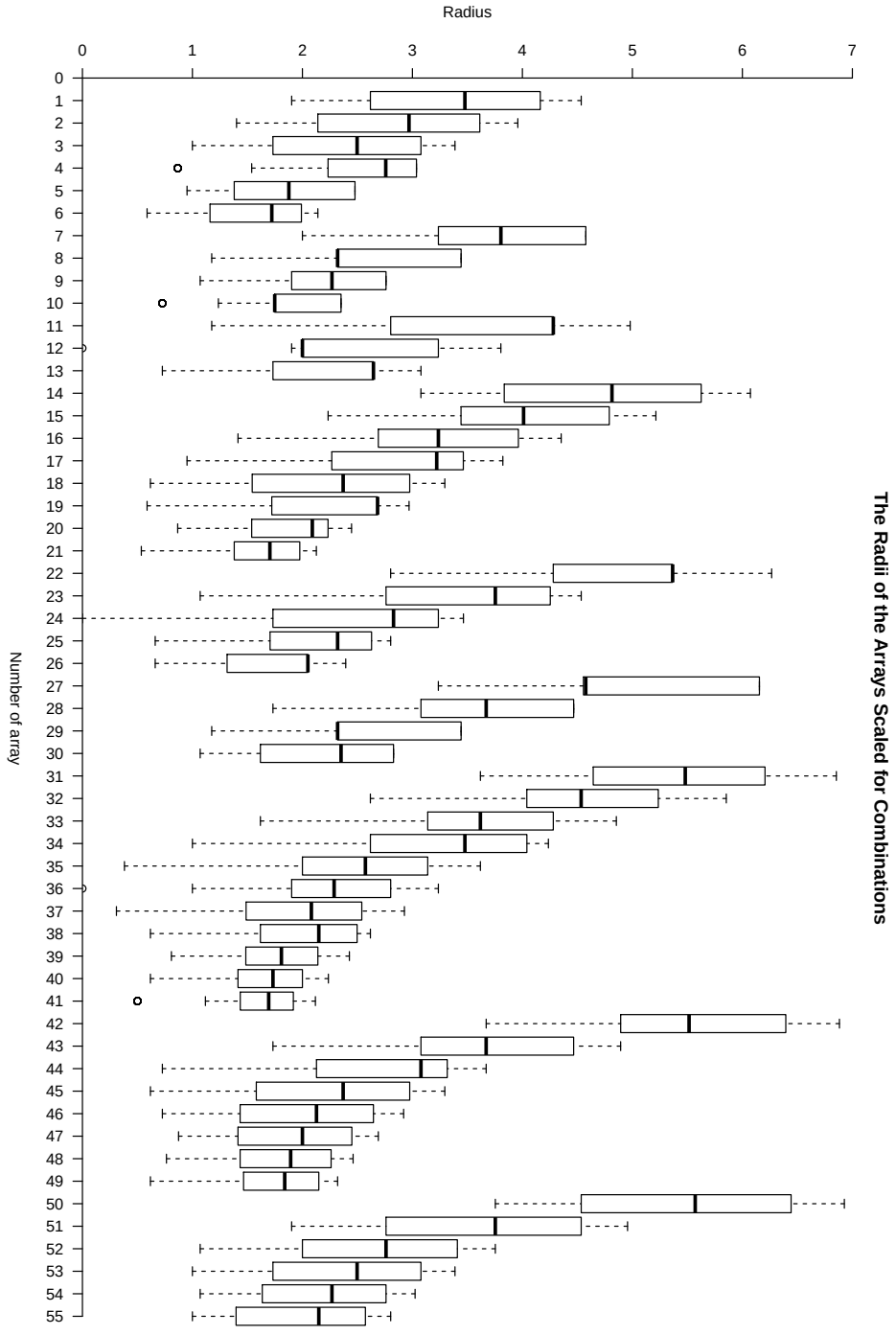


Figure 2.7: The radii of the scaled point arrays showing that some pairs of compatible point arrays do not overlap at all.

2.5 Denser Point-Arrays

The affine extensions of point arrays have a degree of freedom that corresponds to how often the translation operator acts (so far, we have only discussed it acting once). If a given translation is repeated and the resulting point arrays scaled to the same radius, the higher iteration arrays will be denser (see Figure 2.8). Given the size of the protein container with respect to its interior radius, different cut-offs may be appropriate. Furthermore, the higher the iteration, the more faceted the point arrays become; that is, the more their exterior approaches the shape of the polyhedron corresponding to the translation vector (that is, several 5-fold translations will result in something tending to an icosahedron). For smaller viruses (generally up to $T = 4$, but occasionally up to $T = 7$) the first iteration is sufficient, but it can be useful to combine compatible arrays, as explained in the next section.

2.5.1 The Combination Point-Arrays

The first method to create denser point arrays is to combine compatible arrays as described in [42]: two arrays are compatible if they are translated along the same axis of symmetry, and we scale each array so that the translation multiplier (λ in 2.1) is the same. This gives us a total of 569 *combination* point arrays, each of which (with 55 exceptions) have approximately double the number of points in them of members of the 55 (the exceptions being the 55 combinations when a point array is combined with itself). The combinations are numbered from 1 to 1083 by combining the compatible (in the sense of having the

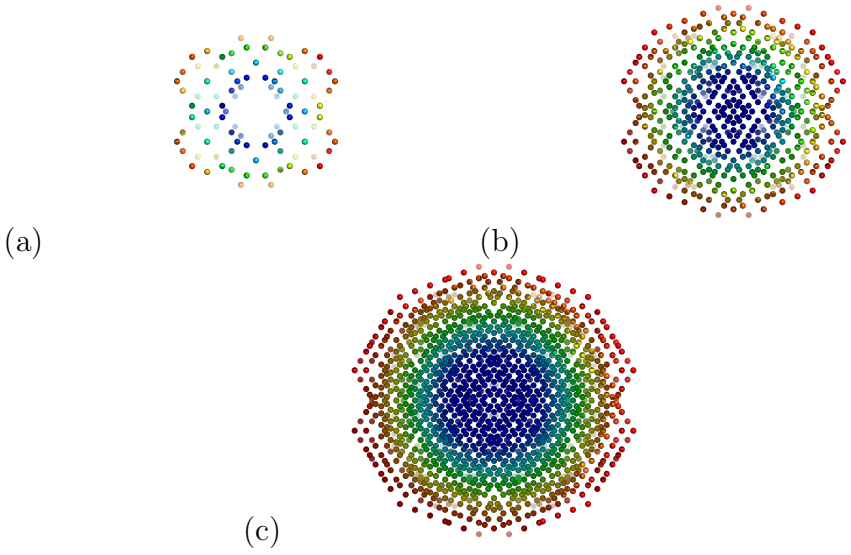


Figure 2.8: A translation applied (a) once (b) twice and (c) three times, showing how the point arrays become denser with higher iterations.

same translation direction) point arrays systemically: combination 1 is point array 1 combined with itself; combination 2 is point array 1 combined with 2; combination 3 is 1 with 3 etc. up to combination 1083 being point array 55 with itself. However, note that point array 1 combined with 2 is exactly the same as point array 2 combined with 1, and so the 569 combinations carry labels from 1 to 1083. However, this number system had procedural advantages, and we choose to keep this notation to be consistent with previous publications.

For example, in Tables A.6, A.7 and A.8 (see page 172), point array 1 can be combined with point arrays 1–6, 14–21 and 31–41, although a point array combined with itself yields no further information. To combine two compatible point arrays, each are scaled so that the translation has a multiplier (recall λ) of 1 (scaling the base shape

commensurately). For example, to combine point array 1 with point array 20, point array 1 (which has a multiplier of $\tau - 1$) is scaled by a factor of $(\tau - 1)^{-1}$ and point array 2 (with a multiplier of 1/2) is scaled by a factor of 2. This process leaves both point arrays scaled so that the translation multiplier of each is 1.

2.5.2 Second Iteration Arrays

The 55 point arrays originally generated can be viewed as the orbit of a single point under the action of the icosahedral group with a translation vector added, but allowed to act at most once. A natural extension, then, is to allow this translation vector to act at most twice. In essence, we repeat the copy-and-translate process, but instead of our base shape being either the icosahedron, the dodecahedron or the icosidodecahedron, our base shape is one of the point arrays listed in Tables A.6, A.7 and A.8, and the translation used is the same as for the base array.

This procedure creates point arrays with considerably more points in them than the original 55 arrays, and even the combination point arrays mentioned previously (the original 55 have a mean number of points of 406, the combination arrays of 820, but the second iteration point arrays have a mean number of points of 3,116).

There is no mathematical reason why this process cannot be repeated, except that the number of points in the array (and hence the number of constraints it imposes on the matched virus) grows exponentially (as the base shape is being copied a minimum of 12 more times). As well as becoming more numerous, points in second (and

further) iteration point arrays become radically more proximate (to the extent of filling all of space as the number of translations tends to infinity), which is what allows higher iteration point arrays to better match higher T-number viruses.

2.6 Calculating the Exteriors of the Point-Arrays

For later purposes we will require the outermost points of the arrays separately. The procedure for calculating these is straightforward. Using R [81] and the R Geometry package [27], we can calculate the points on the convex hull of a point array. Difficulties arise due to rounding errors and so not all of the points on the convex hull are found. It can occur that a point lies a fraction within the convex hull (most commonly a problem with points that lie in the middle of faces of the convex hull, see Figure 2.9), and so this procedure will not detect them. The function used (`convhulln`) provides a triangulation of the convex hull whose vertices are precisely a subset of the points required. The triangulation provided is arbitrary, but irrelevant; the polyhedron described is always the same and the triangulations are equivalent from a procedural point of view.

Instead, once the triangles forming the convex hull have been found, the distance from each point in the array to this polyhedron can be computed by applying Eberly's method [17, 19] to each triangle forming part of the convex hull. Then those points that are sufficiently

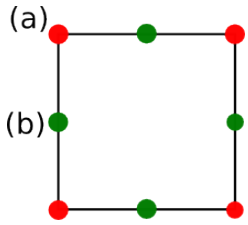


Figure 2.9: The red points (a) are easily picked up as being on the convex hull while the green points (b) may or may not be correctly found due to rounding errors.

close to the polyhedron can be taken as the ones that "should" be part of the exterior. That is, the distance from a point $\mathbf{x}$ to the surface of polyhedron A is:

$$d(A, \mathbf{x}) = \min_{\mathbf{y} \in A} |\mathbf{x} - \mathbf{y}|$$

where A includes the faces, edges and vertices of the polyhedron (or, more precisely, the triangulation of the polyhedron). We take as being on the hull all $\mathbf{x}$ in the point array such that $d(A, \mathbf{x}) \leq \varepsilon$ for some (small) $\epsilon \in \mathbb{R}$.

This procedure is slightly more involved than just using the points found by `convhulln`, as there is not (currently) software that can compute convex hulls symbolically[1]. Instead, the symbolic representations of the point array are evaluated. This new version is used to compute the convex hull, and then each symbolic point is tested against those known to be on the hull. This is unfortunately not a particularly fast procedure, but only needs to be carried out once and the result can be reused. The results can be found on the attached CD.

[1]That is, using algebraic expressions rather than floating point numbers. This is a method of avoiding rounding errors in computations.

Chapter 3

The Best-fit Algorithm

Now that the point arrays of the library have been created, their implications for virus architecture are investigated. This is achieved essentially by superimposing the point arrays on the virus in question via the scaling approach detailed in Section 3.2.2 and applying a scoring function that probes the point arrays' fit to topographical features and proximity to capsid proteins. The process can be sped up somewhat by only considering a fraction of the whole structure due to symmetry.

3.1 Reduction to the Asymmetric Unit

The virus and the point arrays both satisfy icosahedral symmetry and so only the asymmetric unit of both the virus and the point arrays need be considered. This need only be done once on the point arrays, and then these asymmetric units will be the input into the algorithm, along with the input from the `pdb`-file, which is restricted to those

atoms whose positions lie in the asymmetric unit as a pre-processing step.

The vertices of a typical asymmetric unit intersecting the icosahedron's surface has vertices given by

$$\begin{aligned} v1 &= (0, \tau, 1) && \textit{(the 5-fold)} \\ v2 &= \frac{1}{2}(1, \tau, 1+\tau) && \textit{(a 2-fold)} \\ v3 &= \frac{1}{2}(-1, \tau, 1+\tau) && \textit{(the other 2-fold)} \\ v4 &= \frac{1}{3}(0, \tau, 1+2\tau) && \textit{(the 3-fold)} \end{aligned} \tag{3.1}$$

where $\tau = \left(1+\sqrt{5}\right)/2$ (see Figure 3.1).

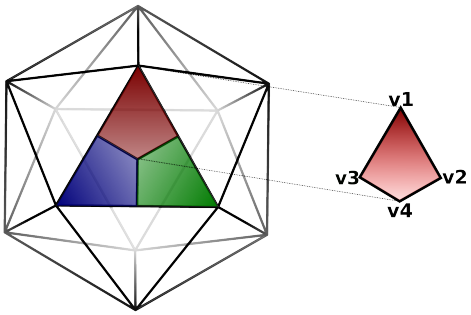


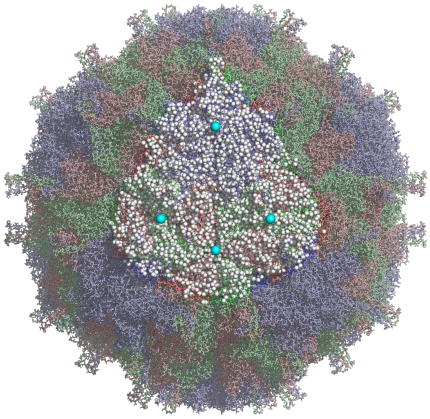
Figure 3.1: Three asymmetric units of the icosahedron shaded in red, green and blue displaying how they meet at a 3-fold axis, and a close-up view of one cell annotated with the vertices from (3.1).

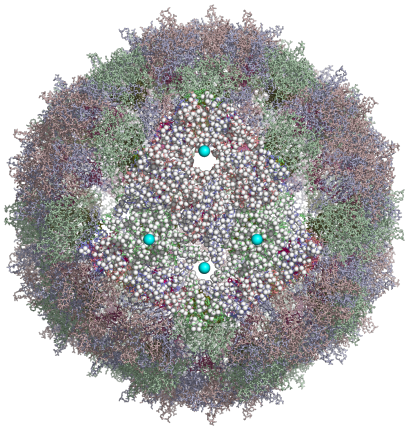
3.1.1 Structural Data Reduction

A virus is a complicated object and contains many atoms (around 510,000 for Pariacoto Virus ($T = 3$) and approximately 2,943,660 for the full Bluetongue capsid ($T = 13$) [29]) but is highly symmetric (recall there are 60 asymmetric units). The amount of data to be processed is therefore reduced by projecting the position of each atom to the origin and calculating whether it passes through a kite one and a half times the linear size of that given in (3.1). The simplified version of Pariacoto Virus has merely 39,710 atoms (a reduction of 92.3%) and Bluetongue would be reduced to 240,733 (a reduction of 91.8%).

This process does not need to be particularly precise as long as the resulting set of atoms fully contain the asymmetric unit plus information on nearby sections of the surrounding subunits, as a matching carried out with the entirety of the virus would produce the same end result due to symmetry. However, the fewer atoms in the section of virus that are checked, the faster the process will be. On the other hand, information beyond the fundamental domain must be kept as the relative locations of points on the boundary of this to surrounding protein is important. Figure 3.2 shows the chosen selection against the whole virus in two cases.

(a)





(b)

Figure 3.2: The vertices of the kite (cyan) and the C_α atoms inside the cone defined by the expanded kite (white) showing the reduction in number of atoms caused by this simplification for (a) Pariacoto Virus and (b) Bacteriophage MS2.

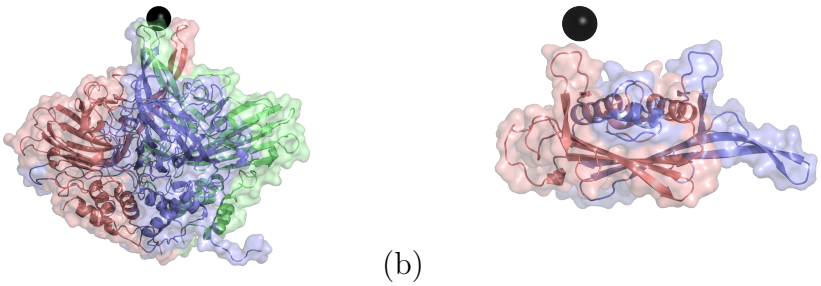
3.1.2 Model Data Reduction

In the same way as with the structural data, a point in a point array is in the asymmetric unit if, when projected down to the origin, it passes through the kite formed by the four vertices given in (3.1). Care must be taken, though, with points that project through the very edge of the kite, as a lack of infinite numerical precision can cause some points to be rejected when they should be registered as located within the asymmetric unit. To cover this, given the point arrays are not infinitely dense, the kite can be enlarged very slightly to ensure all the appropriate points are captured. It is essential that points are not doubly-counted, but this can easily be checked by calculating how many points the reduced asymmetric unit would correspond to with full symmetry applied and comparing this to the number of points in the full point array.

The point arrays end up being reduced by an amount somewhere between 94.2% and 98.2%, with half of the arrays being reduced by between 97.6% and 98.0%.

3.2 Identification of the Best-fit Point-Array

The point arrays identified by Dr Keef via visual inspection previously matched outermost features of the viruses well (see Figure 3.3), and therefore we have created an algorithm that mimics his procedure.



(a) (b)

Figure 3.3: Points matching outermost features of (a) Pariacoto Virus and (b) Bacteriophage MS2.

3.2.1 Defining the Outermost Viral Features

We have already defined the outside of a point array in Chapter 2, Section 2.6. Here we develop a similar procedure for the structural data representing the virus. To define the outermost features of a virus capsid, we determine the C_α atoms that lie above 95% of the maximum radius on which an atom occurs, cluster these hierarchically by distance using the `hclust` function in R, and then take the mean of each cluster. This target point is then raised to the same radius as the top of the radially most distant C_α atom — that is, the radius of the atomic position given by the `pdb`-file plus the van der Waal radius of that atom.

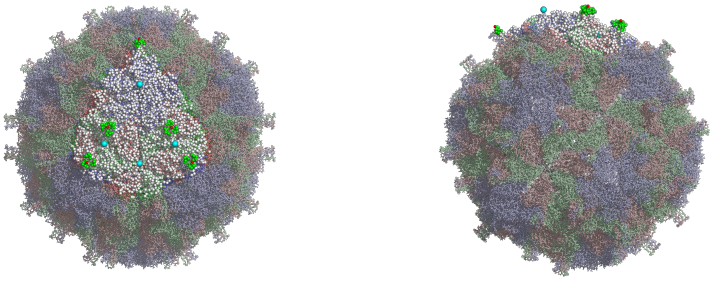


Figure 3.4: The outermost C_α atoms (green) in the asymmetric unit (white) of Pariacoto Virus and their associated scaled mean (red).

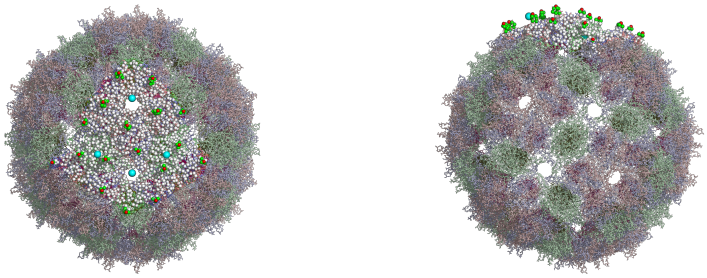


Figure 3.5: The outermost C_α atoms (green) in the asymmetric unit (white) of Bacteriophage MS2 and their associated scaled mean (red).

The C_α atoms in the asymmetric units that are picked out are shown in green in Figures 3.4 and 3.5, whereas the target points are indicated in red. Close-ups of these are in Figures 3.6 and 3.7. As we discuss later, this will be a target point in our procedure for matching the outermost array points.

3.2.2 Scaling and Scoring

For each pair of target point $\mathbf{p}_t$ (as discussed in Section 3.2.1) and outer array point $\mathbf{p}_a$ (i.e. one on the convex hull), the minimal distance $d_{\min}$ that can be achieved between them by a collective rescaling of all array

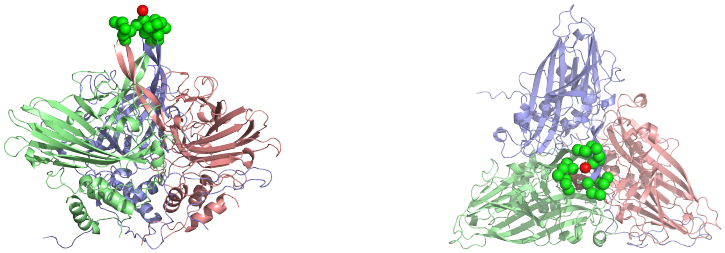


Figure 3.6: A trimer of Pariacoto Virus with outermost C_α atoms (green) and target point (red).

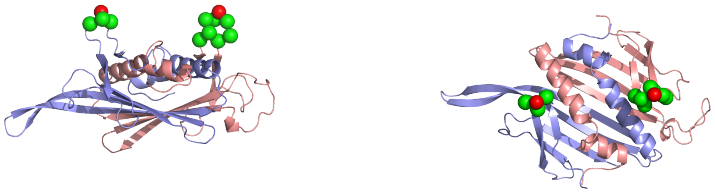


Figure 3.7: A dimer of Bacteriophage MS2 with outermost C_α atoms (green) and target point (red).

points is

$$d_{\min}(\mathbf{p}_t, \mathbf{p}_a) = \sqrt{|\mathbf{p}_t|^2 - \left|\frac{\mathbf{p}_t \cdot \mathbf{p}_a}{|\mathbf{p_a}|^2}\mathbf{p}_a\right|^2} \tag{3.2}$$

(a pictorial representation of the gauge point being scaled to the target point is given in Figure 3.8). The shortest distance over all such pairs $(\mathbf{p}_t, \mathbf{p}_a)$ determines the scaling of the whole array; that is, the array is scaled so that $\mathbf{p}_t$ and $\mathbf{p}_a$ are at this distance from one another. The choice of $\mathbf{p}_a$ that realises this distance is referred to as the *gauge point*, $\mathbf{p}_g$. The choice of array point is restricted to those on the convex hull to ensure that the point array and virus are initially scaled to approximately the same size.

This minimal distance is part of the score that will be assigned to

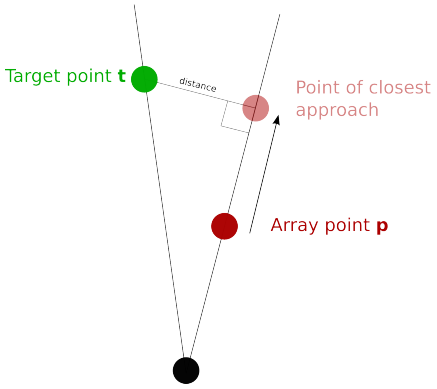


Figure 3.8: The array point slides to the point of closest approach to minimise the distance to the target point.

the array (referred to as S_1 in equation 3.5 (page 70)). We therefore have

$$S_1 = d_{\min}(\mathbf{p}_t, \mathbf{p}_g) = \min_{(\mathbf{p}_t, \mathbf{p}_a)} d_{\min}(\mathbf{p}_t, \mathbf{p}_a). \tag{3.3}$$

Figure 3.9 shows such a gauge point having been scaled to match a target point according to this procedure. It can now be seen why the raising of the target point $\mathbf{p}_t$ to the radius of the top of the highest C_α atom (as mentioned in Section 3.2.1) is required; it ensures that the gauge point $\mathbf{p}_g$ will not be placed inside material[1]. The reason for this is that the points of the arrays are intended to be boundary conditions for the proteins, and so must be on or near the boundaries of those proteins, not inside them.

Once the array under consideration has been scaled to as described above, it is scored. First we introduce some terminology: *vdw* is the

[1]Note that this could be checked with [68] via MSMS [90] (also available in PyMOL [123]) if it could be made stable for tiny probes. MSMS calculates the *Solvent Accessible Surface* (SAS) and *Solvent Excluded Surface* (SES) analytically for a given molecule, but is only stable for atomic size probes, which are considerably larger than the infinitesimal points we use.

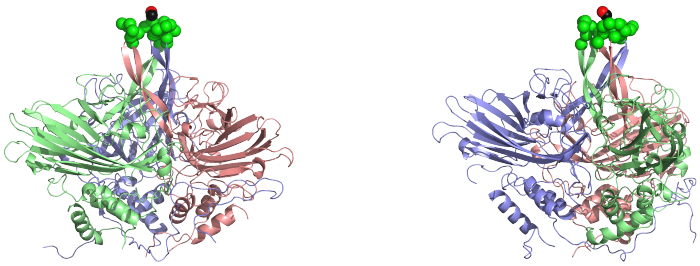


Figure 3.9: A trimer of Pariacoto Virus with outermost C_α atoms (green), target point (red) and gauge point (black).

van der Waal's radius of each atom and is taken at 1.9Å [8]; *threshold* is how close an array point has to be to the capsid of the virus before it is counted as representative of the capsid (set at 4Å); *inner threshold* is how close an array point has to be to proteins if it is to count multiple times (set at 2Å); *virus radius* is the maximum radius an atom occurs, plus *vdw*; *inner radius* is the minimum radius an atom occurs, minus *vdw* and *middle radius* is the mean of *virus radius* and *inner radius*.

We now compare the point arrays with data from `pdb` files. These data contain the coordinates of all the detected non-hydrogen atoms in each protein of the virus capsid (as mentioned in Section 1.3) organised according to chains.

For each point in an array, the distance to the nearest atom of each chain is computed. This produces a table of data similar to Table 3.1[2], where the double line indicates the middle radius of the virus. The meanings of the columns are as follows: n gives the number of points of the array at each radius r; A, B and C[3] give the distances from a

[2]The data given is actually for Pariacoto Virus, but is representative of the steps taken in general.

[3]Pariacoto only has three chains. For larger viruses with more chains, more letters are used.

point at radius r to the nearest atom in the protein chain indicated in the column header.

The entries 7777, 8888 and 9999 are placeholders, indicating that the algorithm has not calculated a distance because it will not be required in the scoring steps — reasons are given in the various stages of scoring listed below. This enables the algorithm to work considerably faster without compromising accuracy. The three different numbers indicate three different regions: 9999 indicates points more than *threshold* (4Å) below *inner radius* (that is, rows 1, 2 and 3 in Table 3.1) — these will never be able to be representative of the capsid and will only match genomic material; 7777 represents points between *inner radius* minus *threshold* and *middle radius* (i.e. rows 5, 6, 7 and 8) — these may match capsid or genomic material depending on their position; and 8888 is the placeholder for points above *middle radius* (rows 9, 10 and 11), which should match to protein.

		n	A	B	C	r
1	Inner	20	9999	9999	9999	43
2		20	9999	9999	9999	70
3		12	9999	9999	9999	77
4	Middle	60	7777	7777	7777	92
5		60	7777	7777	7777	94
6		60	7777	0.017	0.107	112
7		20	7777	7777	7777	114
8		30	7777	7777	0.471	131
9	Outer	60	0.054	1.749	8888	140
10		60	8888	11.477	14.193	162
11		60	8888	3.061	2.927	174

Table 3.1: Sample RMSD output.

		n	A	B	C	r	x	SS
1	Inner	20	9999	9999	9999	43	0	-
2		20	9999	9999	9999	70	0	-
3		12	9999	9999	9999	77	0	-
4	Middle	60	7777	7777	7777	92	0	-
5		60	7777	7777	7777	94	0	-
6		60	7777	**0.017**	**0.107**	112	2	0.012
7		20	7777	7777	7777	114	0	-
8		30	7777	7777	**0.471**	131	1	0.222
9	Outer	60	**0.054**	**1.749**	8888	140	2	3.062
10		60	8888	**11.477**	14.193	162	1	131.722
11		60	8888	3.061	**2.927**	174	1	8.567

Table 3.2: Annotated sample RMSD output where the bolded entries denote scores that indicate the corresponding point of the array is representative of protein and hence will contribute to the scoring of the point array.

An analysis of these tables is then conducted by scoring the array according to the following procedure:

1. Flag each entry (entries shown in bold in Table 3.2 are those flagged) in the A, B and C columns less than *inner threshold* (2Å). This is so that points that are very close to multiple proteins score to each of those proteins.

2. For each row corresponding to a radius greater than the *middle radius* (that is, rows 9–11), flag the minimum entry (of columns A, B and C). This part ensures that floating points far from the capsid are penalised (by forcing them to count), but does not consider points that are predictive and occur in the space within the capsid for which no data is available in the `pdb`-file, making sure they do not penalise the array.

3. For each row corresponding to a radius less than the *middle radius* (rows 1–8, but in practice, only rows 4–8 need to be checked as rows 1–3 are guaranteed to be too far from capsid material to be representative), flag the minimum entry (of columns A, B and C) if it is less than *threshold* (4Å). This ensures points that are within the capsid and match material well contribute to the score of the array[4].

4. For each row, count how many entries are flagged (this gives column x in Table 3.2) and measures how many protein chains

[4]Note that flags from steps 1, 2 and 3 can lead to more than one flag per entry. For example, 0.054 at radius 140 is flagged twice because it meets criteria 1 and 2. This is irrelevant because it only matters whether an entry is flagged at all or not.

the point under consideration matches to.

5. For each row with at least one flagged entry, sum the squares of the flagged entries (which gives column SS in Table 3.2).

6. Then, letting i run over each row and denoting as n_i, x_i and SS_i the entries in the ith row, calculate

$$S_2 = \sqrt{\frac{\sum_i (n_i \times \mathrm{SS}_i)}{\sum_i (x_i \times n_i)}}. \tag{3.4}$$

This calculates a value akin to the RMSD of the bolded entries to the surfaces of the proteins, adjusted to compensate for the fact that there are different numbers of points per radial level. In effect, it measures how well those points near to proteins represent the surfaces of those proteins, where $S_2 = 0$ would imply those points all lie precisely on protein boundaries.

The two scores S_1 and S_2 (calculated in (3.3) and (3.4) respectively) need to be considered simultaneously to arrive at a combined score. For this, we consider these scores as coordinates of a point in a plane as illustrated in Figure 3.10. The red point scores well on S_2 (it fits well to the capsid), but not very well on S_1 (matching a tower well), whereas the blue point fits to a tower well, but not to the capsid. However, the green point scores less well than the blue on S_2 and less well than the red on S_1, but scores adequately on both. This, then, is the scoring point for the array to be kept, and we use the score given by

$$S = \sqrt{S_1^2 + S_2^2} \tag{3.5}$$

which is essentially the Euclidean distance of the scoring point from the ideal situation represented by 0 for both S_1 (matching the target point exactly) and S_2 (all points lying exactly on protein surfaces). Both scores are measured in Angstroms and are therefore directly comparable — not even rescaling is necessary, as they both occur over similar ranges.

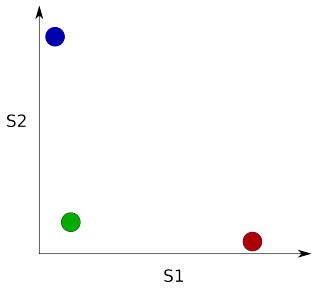


Figure 3.10: Three sample sets of scores.

This combination ensures that the best-fit array matches both the outermost characteristic features (such as towers) of the virus and the capsid well. However, the `pdb`-file is given only up to a certain resolution, and so checking that the best-fit point array does not apply purely because of these margins is necessary (that is, we need a robustness check for our procedure). We therefore test each array not just at the scaling given by the minimum distance from equation (3.2) (page 63), but also by the scalings found by moving the target point up and down by increments of 0.1Å in a range of ±5Å. The amount that the target point is moved is referred to as the *shunt*. Arrays that

only occur over a small (usually 0.2Å) range are viewed as artifacts of the data and not representative.

3.3 Analysing the Data

The algorithm outlined in Section 3.2 provides a large amount of information on how each point array matches to a given virus at each of many scalings. These data then need to be interpreted to remove any anomalies and produce the candidate(s) for the best-fit array.

The output of the algorithm produces data such as that given in Table 3.3 (except the last column, which is added after some processing, as explained later). The column headings here are truncated for reasons of space. The headings, where different from Chapter 4, are given in *emphasis*. For reference, the columns are as follows:

No. *(Combo Number)* — the number of the combination point array;

Out *(Outside)* — the point array (of the 55 basic arrays) that lies on the outside of the combination;

In *(Inside)* — the point array that lies on the inside of the combination[5];

RMSD — the quasi-RMSD score (S_2 from equation 3.4);

Tower Dist. (*Dist. to Tower)* — the distance from the gauge point to a tower midpoint (S_1 from equation (3.3));

[5]In cases such as combination point array 70 that consists of arrays 3 and 36 where the translation multiplier λ of each is 1, both arrays can be considered on the outside. In these cases, the arrays are presented in numerical order.

Final Score — the final overall score (S from equation (3.5))

R *(Combo Radius)* — the maximum radius on which an array point lies;

Shunt the radial adjustment of the target point which gives rise to the best (lowest) score for the point array under consideration;

NH *(Number of Hits)* — the number of points corresponding to capsid material;

Prev *(Prevalence)* — the number of scalings for which the point array listed occurs in the tested range[6].

No.	Out	In	RMSD	Tower Dist.	Final Score	R	Shunt	NH	Prev
563	50	28	4.698	0.892	4.782	173.9	-1.6	5	24

Table 3.3: Sample results showing which array this row is for, the outside and inside components of that combination, S_2, S_1 and S, the radius of the array, which shunt was used, the number of matches against protein and how many shunts this array is relevant at.

Two filters are applied to the data to remove anomalies or point arrays that do not sufficiently fit the capsid:

1. *Remove any rows that have* $NH \leq 3$. These are deemed not to be sufficiently representative of the capsid.

2. *Remove any rows that have Final Score* > 1000. As part of the scoring algorithm, any array that has a point with a negative distance to a protein (that is, a point that is located *inside* the van

[6]Table headings are given in abbreviated form in Table 3.3.

der Waals radius of an atom) has a penalty term of 1000 applied to its score[7]. We are attempting to find geometric constraints on material boundaries for the virus, so points lying within capsid material are not appropriate.

Once the list of scores has been pared down in this way, the best scaling for each array is considered. A final statistic is calculated, which is the *Prevalence* of that array in the reduced list — that is, the number of scalings for which the given array occurs in the reduced list. Ideally, if an array has a *Prevalence* that is low, it could be rejected, considering the caveat that the *Prevalence* will be artificially low at the extreme ends of the shunt range; if the best fit of a point array occurs at the end of the search range for the shunts, the range must be increased. The remaining data is then sorted by *Score* (the overall score as given by equation (3.5)) with the point array with the lowest score being rated the best-fit point array.

[7]The exact figure is irrelevant; the key is that the array is removed from consideration.

Chapter 4

Applications to Viruses

This chapter contains the results of applying the algorithm developed in Chapter 3 to a selection of `pdb`-files downloaded from the VIPER website [83].

Each virus (unless otherwise specified) was tested against the first iteration combination point arrays described in Chapter 2, Section 2.5.1, i.e. the library of point arrays, at a range of shunts between -5Åand 5Å, with a step size of 0.1Å. The results for each virus are given in a table with the structure of Table 3.3.

Several sections are presented, each focussing on a different aspect of virus structure.

4.1 Genomic Cages

In addition to information on the atomic positions of the capsid proteins, for a number of viruses there is also data on the organisation of the genomic material, and this can be used to validate predictions of the algorithm. Here, three viruses for which X-ray, cryo-EM or neutron scattering data reveal ordered features in their genome organisation are studied, as well as one (Bacteriophage GA) that is evolutionarily related to such a virus (Bacteriophage MS2).

4.1.1 Pariacoto Virus

The first test case is Pariacoto Virus, which is a $T = 3$ virus with single-stranded RNA that infects the Southern Army worm found in Peru. Its structure was resolved to 3.0Å [98], and is available from VIPER (PDB-ID `1f8v`). The `pdb`-file includes 88% of the protein capsid which exhibits prominent towers on local 3-fold symmetry axes (Figure 3.9 (page 65) shows the tower from the side, indicating its height, and Figure 1.12 in the Introduction on page 25 shows it from above, demonstrating the local 3-fold location). It also, importantly, includes a modelled dodecahedral cage of RNA (containing approximately 35% of the viral genome) within the capsid [13].

The R and S strands of 'protein' forming the RNA were removed from the `pdb`-file before the algorithm was applied leaving just the A, B and C chains of actual protein; that is, RNA was not taken into account when determining the best-fit point array. The results are given in Table 4.1. They clearly show that the best outer point array

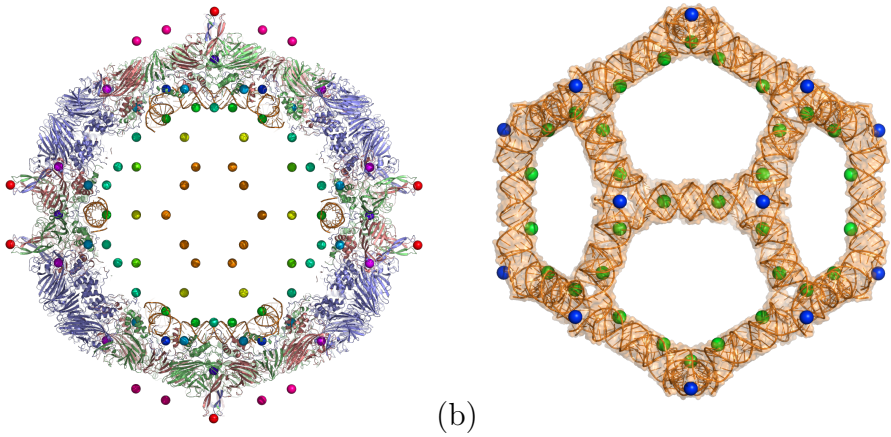
is 50; interestingly, the first 13 results have 50 as their outside, and the 14th result is array 131, which is array 6 combined with itself (note, array 6 has the same exterior as array 50). The first array that has a different exterior is array 228 in 15th place, which has a final score of 7.506 compared to the score of the best-fit point array which is 4.782.

Figure 4.2 shows the best-fit point array overlaid on not only the crystal structure of the capsid proteins, but also on the RNA cage that was removed prior to running the algorithm. The red, magenta, purple and light blue points map around the capsid protein, marking out the height of the tower, the midpoints of trimer-trimer interactions as well as the lowest extent of the capsid material. There are also some "floating" points (pink), shown in Figure 4.1(a); these count badly against the score of the point array, but this negative effect is clearly countered by the good fit of the rest of the points, as evidenced by the fact that this point array achieves the lowest score.

What is astonishing is the fit that this point array has to the RNA cage. Figure 4.1(b) shows the (dark) blue and green points marking out a dodecahedral cage that traces the RNA double-helix (A-duplex RNA), with the blue points located at the 3-junctions and the green points fitting snugly into the minor grooves.

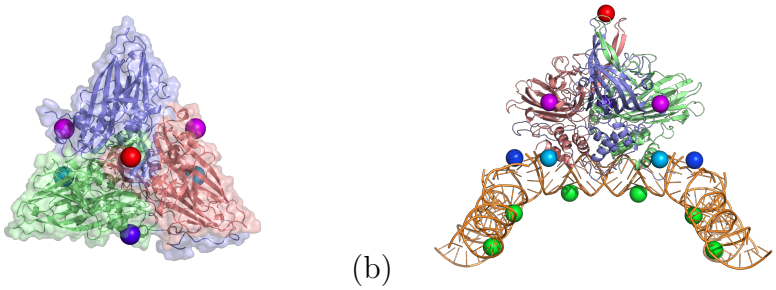
Combo Number	Outside	Inside	RMSD	Dist. to Tower	Final Score	Combo Radius	Shunt	Number of Hits	Prevalence
563	50	28	4.698	0.892	4.782	173.894	-1.6	5	24
1014	50	51	4.707	0.892	4.79	173.894	-1.6	5	15
226	50	11	4.768	0.892	4.85	173.894	-1.6	5	41
550	50	27	5.246	0.892	5.321	173.894	-1.6	6	24
239	50	12	5.646	0.892	5.716	173.894	-1.6	4	45
252	50	13	5.646	0.892	5.716	173.894	-1.6	4	45
576	50	29	5.646	0.892	5.716	173.894	-1.6	4	45
589	50	30	5.646	0.892	5.716	173.894	-1.6	4	45
1013	50	50	5.646	0.892	5.716	173.894	-1.6	4	45

Table 4.1: The ten lowest scoring combination point arrays for Pariacoto.



(a) (b)

Figure 4.1: The best-fit point array overlaid on (a) the crystal structure of the capsid material and RNA cage of Pariacoto Virus and (b) just the RNA.



(a) (b)

Figure 4.2: The best-fit point array overlaid on the crystal structure of the capsid material of a trimer of Pariacoto Virus from (a) the top and (b) from the side with the associated RNA fragment.

However, there is further information to be gleaned from the points that the best-fit point array 563 produces near the capsid centre: its points at radial level 92Å correspond to the minor grooves of the RNA, and are 31.1Å apart; its points closest to the center of the capsid (radial level 43Å) are also 31Å apart. Given that not all of the RNA is accounted for in the outer dodecahedral cage, there must be more in the capsid, and this point array suggests that there could be an extra cage closer to the middle. It is an area where the cryo-EM data[1], albeit at a much weaker signal, shows a further ring structure, as shown in Figure 4.3.

Moreover, while the orange points are 31Å apart, the yellow points are 50.2Å apart (one turn of RNA plus $\sim$ 19Å). This is a similar pattern to the dark blue points (which mark the corners of the dodecahedral cage) being 81.2Å apart (two turns of RNA plus $\sim$ 19Å), suggesting that the yellow points are in a similar position relative to the orange points as the dark blue points are to the green points in Figure 4.1. This could imply that, in analogy, they are marking a cage of RNA with one turn of RNA per side, rather than the two turns in the larger RNA cage at larger radius that accounts for the 35% of RNA seen in the crystal structure. If further data for this region were to become available, potentially without the icosahedral averaging commonly used, this prediction could be tested.

Having investigated the best-fit point array, the next two arrays are analysed to see if they contain any further information. (After these two further arrays the final score jumps significantly — from

[1]Reconstruction kindly provided by Jack Johnson

4.850 to 5.321.) 1014 contains only minimal further information, but 226 includes extra points of interest. See Figure 4.4 for a comparison of the point arrays on the data. Note that the extra points in 226 'bracket' the RNA in a way that 563 does not (see Figure 4.5).

Recalling the other 3D approach, that is, that by Janner mentioned in Section 1.2.3, the results here are compared with those available in [39]. Figure 4.6(a) shows the encasing form of Janner with two sets of inscribed inverted pentagons (i.e. two inverted pentagons inscribed within two inverted pentagons) and a further inscribed decamer used to provide the scaling from the encasing form to the protein pentamer and Figure 4.6(b) shows the same selection of proteins with the gauge points (magenta) and those points marking the three-junctions of the RNA in comparison; the latter figure uses two complementary pentagrams inscribed within the encasing form of the gauge points to provide the scaling necessary. It can be seen that the algorithm and point arrays presented here are compatible with and extend Janner's method while being simpler in execution, due to, in part, the more natural scaling available when using a finite array as opposed to an infinite lattice.

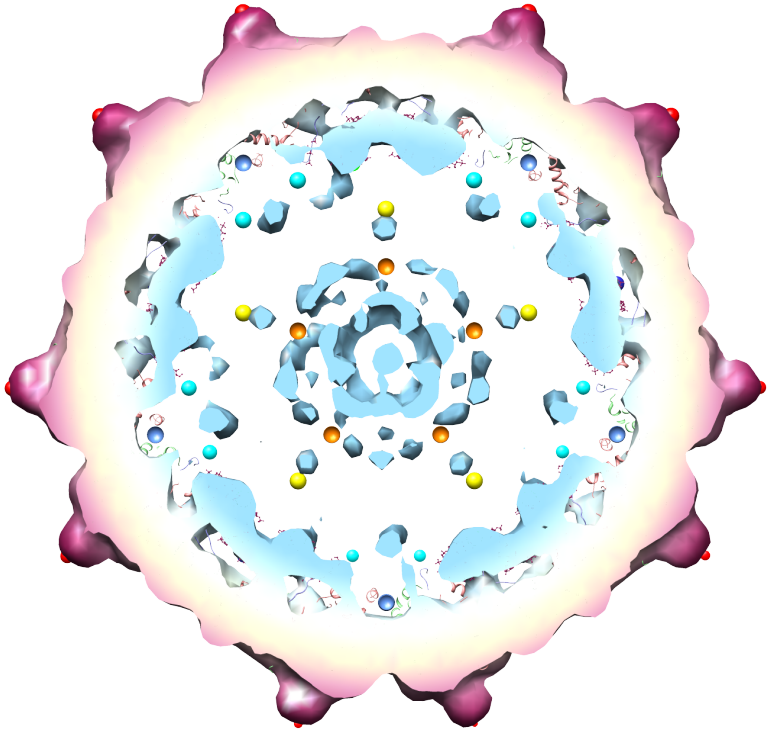


Figure 4.3: The best-fit point array overlaid on the crystal structure [98] of the capsid material and RNA cage as well as cryo-EM data[3] of Pariacoto Virus. Density belonging to the capsid is shown in purple and cream, the layer of RNA adjacent to the interior surface of the capsid is in light blue, and there is evidence for a predicted layer of RNA (also shown in light blue) between the orange and yellow points near the centre.

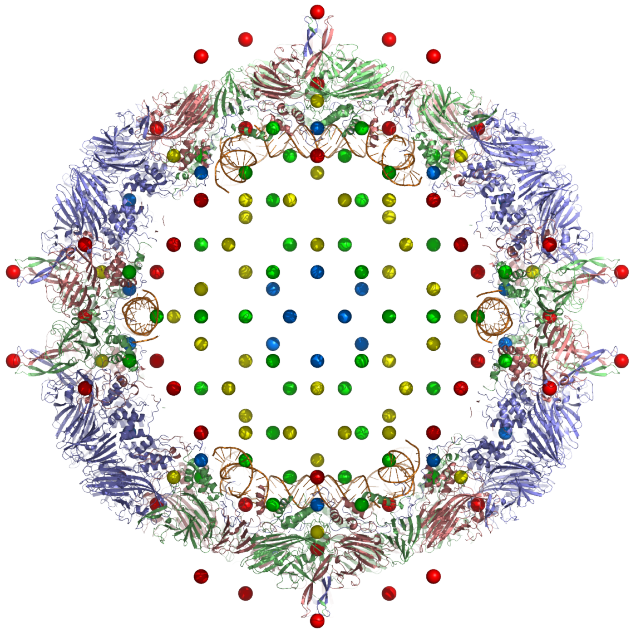
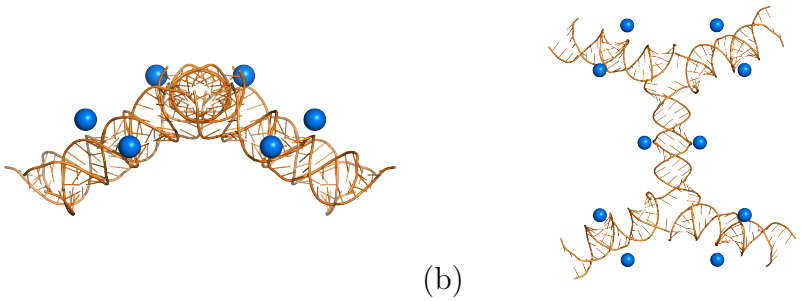


Figure 4.4: A slab through Pariacoto Virus with the best-fit point array (563 — green), and the two next-best arrays (1014 shown in yellow, and 226 in blue) and the points shared by all three point arrays (that is, outer array 50 — red) superimposed.



(a) (b)

Figure 4.5: Points in array 226 (blue) mark additional information about the RNA cage — (a) side view and (b) top view.

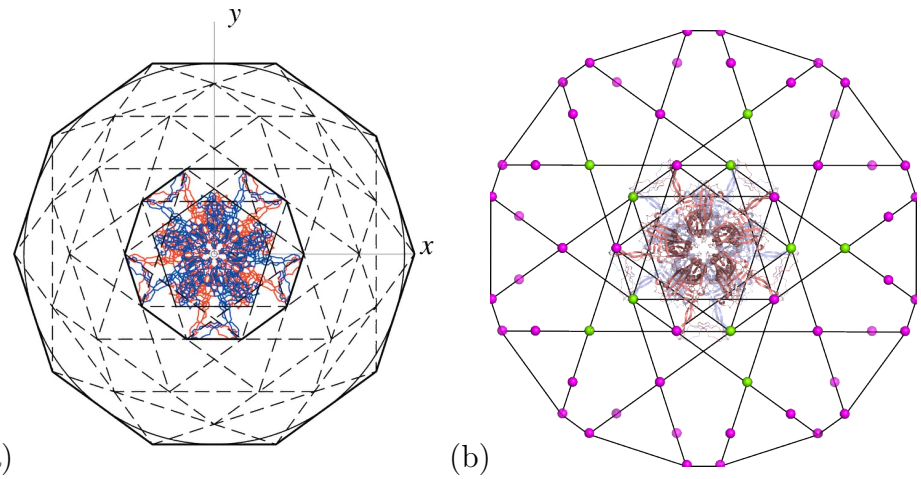


Figure 4.6: A view down the 5-fold axis of a selection from the A chain proteins of Pariacoto Virus from (a) Janner [39] and (b) the algorithm presented here showing the encasing form and the inscribed shapes delimiting the protein bulk and outermost features.

Capsid to RNA prediction

With Pariacoto virus `pdb` having a modelled cage of RNA, it is possible to take the results from the previous section and see how they match to the genomic structure in a more structured way. In particular, each point array that successfully models the capsid (by which is meant "does not place a point within an atom" — that is, has not incurred the penalty of 1000) can be scored against the RNA while keeping the scaling that best represents the capsid. The process is very similar to the previous scoring, but does not force any points to score; those that are within 4Å of the RNA are scored, while those that are further away are not.

Figure 4.7 is a graph showing the original score to the capsid proteins compared to the new score comparing the array to the genome. Unfortunately, as can be seen, there is no correlation between these scores. However, of the 196 combination point arrays that fit to the capsid in some way, fully 50 do not have any points anywhere near the genomic material (these are not shown on the graph), and 28 place points within an atom of it. Of the remaining 118, point array 563 (the best fit to the capsid — and hence at the far left of Figure 4.7) lies at number 19. The old and new scores of the arrays best fitting the genome (up to and including number 563) is shown in Table 4.2.

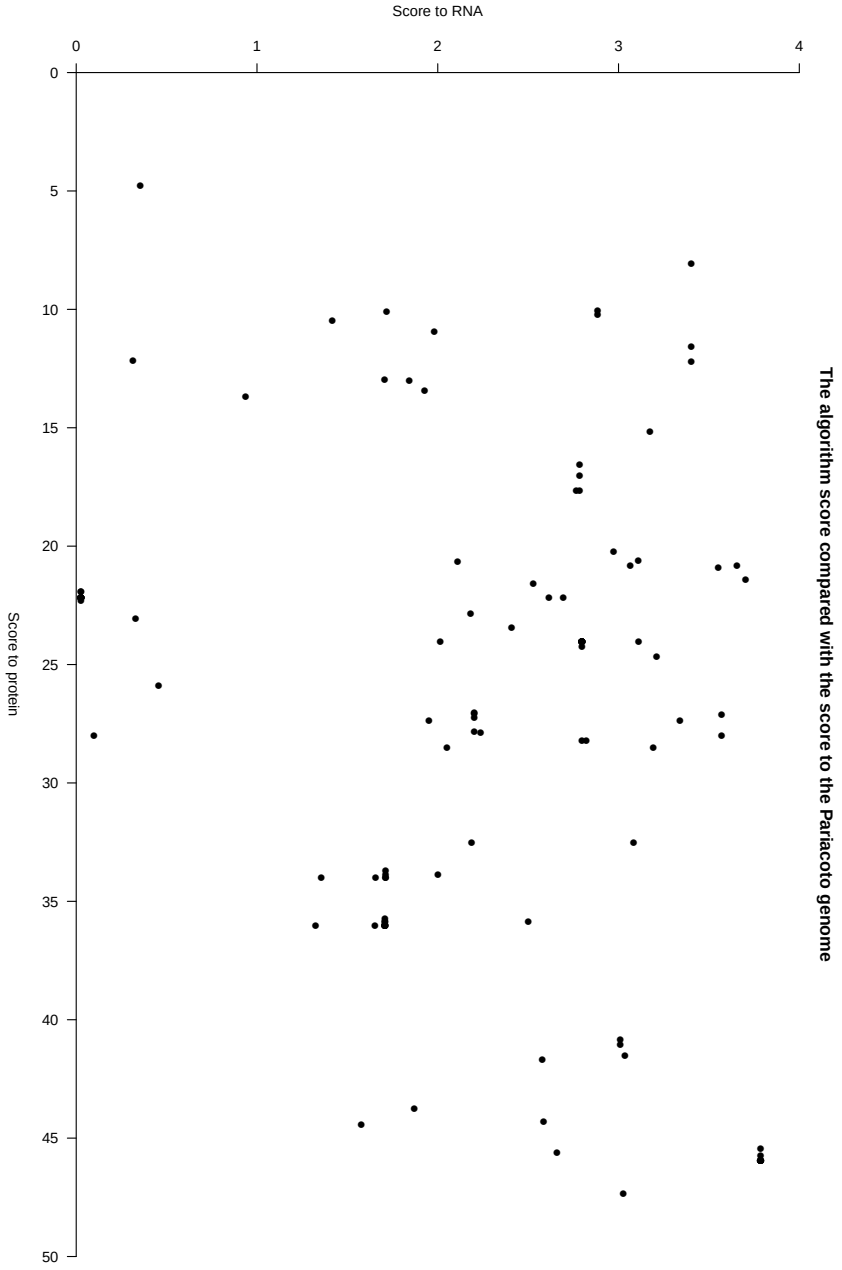


Figure 4.7: A scatter graph showing the comparison each point array's score to the capsid informs their score to the genomic cage. As can be seen, there is no correlation. Point array 563 has the lowest score when matching to protein (and is displayed in the bottom left of the graph) but has only the 19th best score to the genome.

Array	Score to Capsid	Score to Genome
40	21.915	0.0267
65	21.935	0.0267
90, 115, 140, 397, 422, 447, 609, 615, 616, 617, 618, 619	22.157	0.0267
869	22.283	0.0273
499	28.001	0.0993
345	12.18	0.3142
86	23.07	0.3295
563	4.782	0.3540

Table 4.2: The nineteen lowest-scoring point arrays to the Pariacoto genome.

None of these matches place a point directly within an atom (else they would have incurred the scoring penalty and not receive such a good score), but that does not necessarily make them good matches. Figure 4.8, in particular, shows the point responsible for the score of 0.0267 for the top 14 matches (all of which match in precisely the same manner and are displayed in Figure 4.9(a) and (b)) lies within the protein material and, were it subjected to such a stability check, would no doubt fail. The previous scoring used the concept of moving the target point up and down to establish a range over which an array would receive a valid score (the Prevalence), while this method, making use of the "best" scaling for each array, only inspects one such scaling. The point giving array 869 such a good score (displayed in Figure 4.9(c) and (d)) is virtually identically placed.

Point arrays 499, 345 and 86, though, are in very reasonable positions compared to the genome atoms and would pass any stability analysis. 345, in particular, has points in virtually identical locations

to 563, demonstrating that it alone of the other point arrays matches the turn length of the RNA.

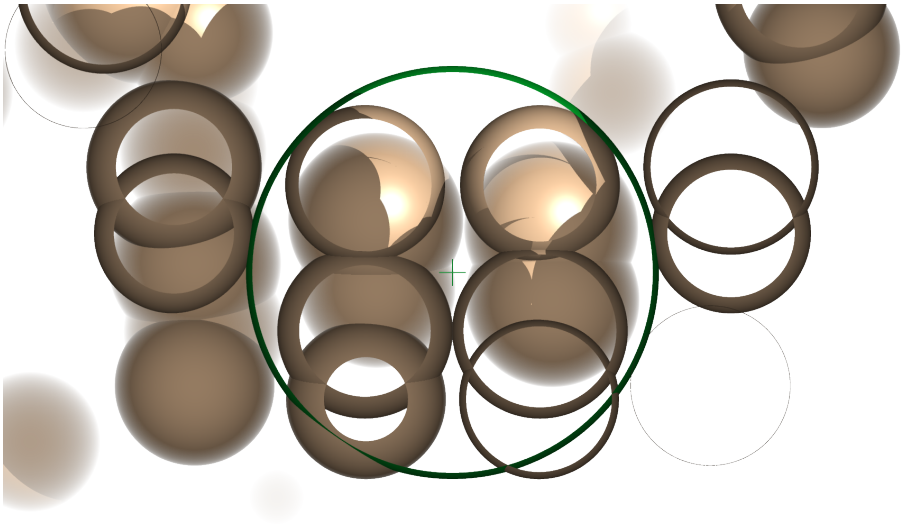


Figure 4.8: A extreme close-up view of the point from point array 40 that matches the Pariacoto genome (the large circle denotes a 4Å radius — the cross in the middle marks the precise middle), showing how it lies between a number of atoms (the beige spheres mark the modelled 1.9Å van der Waals radius) and would fail a stability test.

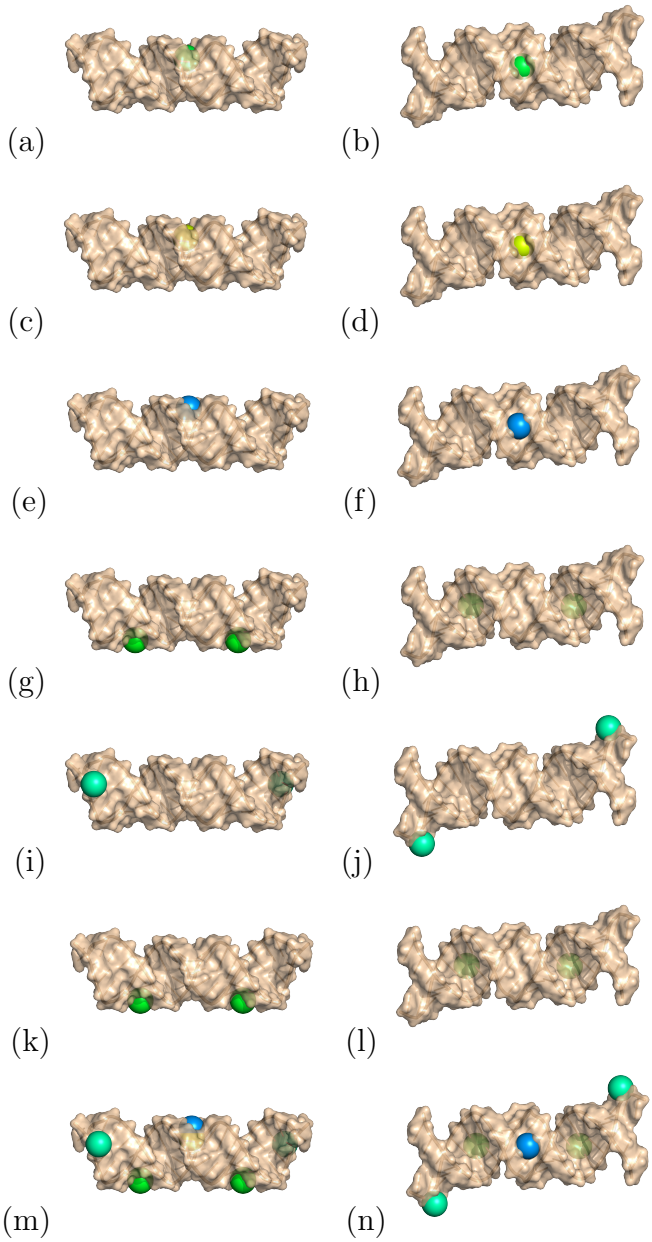


Figure 4.9: A view from the side (left) and top (right) of (a) and (b) point array 40 (same as arrays 65, 90, 115, 140, 397, 422, 447, 609, 615, 616, 617, 618, and 619), (c) and (d) point array 869, (e) and (f) point array 499, (g) and (h) point array 345, (i) and (j) point array 86, (k) and (l) point array 563 and (m) and (n) all the arrays.

4.1.2 Bacteriophage MS2

Bacteriophage MS2 is also a $T = 3$ virus, but it infects *Escherichia coli*. The crystal structure is deposited as PDB-ID `1zdh`. It has been resolved to 2.7Å [26, 112], and this atomic model of recombinant phage includes the RNA stem-loops that make contact with the capsid protein [113]. MS2 is composed of dimers, rather than trimers as in Pariacoto Virus. There are 60 AB dimers and 30 CC dimers, the CC dimers being symmetric and the AB dimers having one of their FG-loops folded to allow the dimer to avoid steric clashes when arranging around the 5-fold axes, as shown in Figure 4.10.

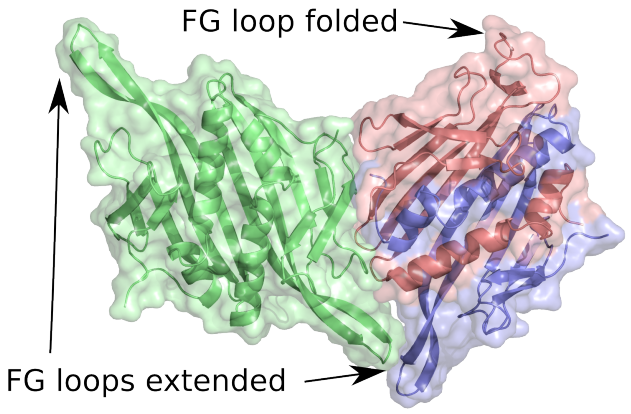


Figure 4.10: A (left) CC and (right) AB dimer of MS2 displaying their FG loops — the loops of the CC dimer (green) are both extended as is that of the A chain protein (blue), whereas that of the B chain protein (red) is folded to allow their arrangement around the 5-fold axes.

Here, point array 7 is the best-fit outer array shared by the two best-scored point arrays (note that 21 and 42 have identical exteriors). 152 is the best-fit point array, with 163 also analysed to see how far it differs.

Figure 4.11 shows the best-fit point array superimposed on the crystal structure of MS2, coloured by radial level, with close-ups of the AB dimers and CC dimers in Figures 4.13(a) and 4.14(a), respectively. Figure 4.12 shows the best-fit point array (152 — green) together with the next-best point array (163 — blue), and the common points in red. It shows that both arrays have points at similar radial levels in the interior of the capsid, and this implies the same prediction for the radial distribution of RNA inside the capsid. As well, not only do the arrays mark the AB-loop of the B conformer (as expected given the matching method used), but indicate where each stem-loop of RNA attaches to the dimer. Note that the RNA was not present in the `pdb`-file while the algorithm was processing it. Indeed, the array coming second has more points near the RNA underneath the CC dimers than the best-fit array, but is no different under the AB dimers.

Cryo-EM data (EMDB-IDs 1431, 1432 and 1433) [59, 104] are available, and in Figure 4.15 we can see how well every point of the best-fit point array matches the experimentally determined RNA structure within the capsid. From Figure 4.12 there is little reason to assume that array 163 would clash with the cryo-EM data, but the points in the middle of the capsid are too close to the centre to reasonably match material, corroborating that our algorithm has selected the correct model.

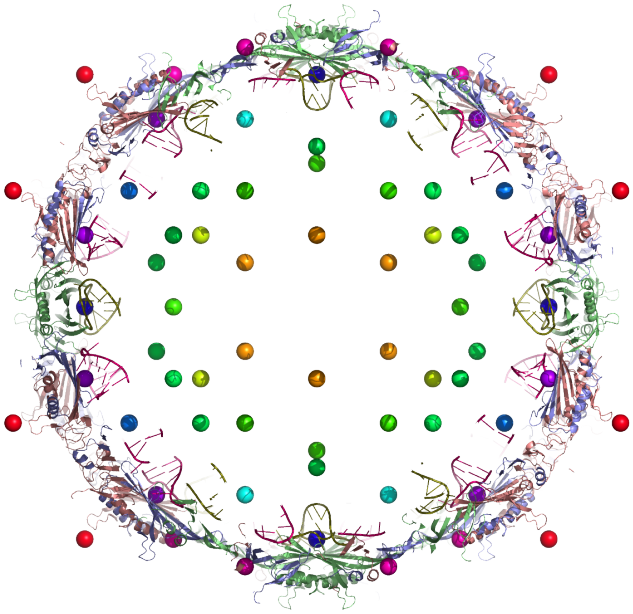


Figure 4.11: The best-fit point array (152) overlaid on the crystal structure [112] of Bacteriophage MS2.

Combo Number	Outside	Inside	RMSD	Dist. to Tower	Final Score	Combo Radius	Shunt	Number of Hits	Prevalence
152	7	8	2.53	5.826	6.351	148.797	4.4	4	7
163	7	45	3.182	5.826	6.638	148.797	4.4	4	7
446	21	21	2.263	6.532	6.913	141.486	2.3	5	16
160	42	7	2.276	6.532	6.917	141.486	2.3	6	16
177	42	8	2.276	6.532	6.917	141.486	2.3	6	16
194	42	9	2.276	6.532	6.917	141.486	2.3	6	16
211	42	10	2.276	6.532	6.917	141.486	2.3	6	16
484	42	23	2.276	6.532	6.917	141.486	2.3	6	16
501	42	24	2.276	6.532	6.917	141.486	2.3	6	16
518	42	25	2.276	6.532	6.917	141.486	2.3	6	16

Table 4.3: The ten lowest scoring combination point arrays for Bacteriophage MS2.

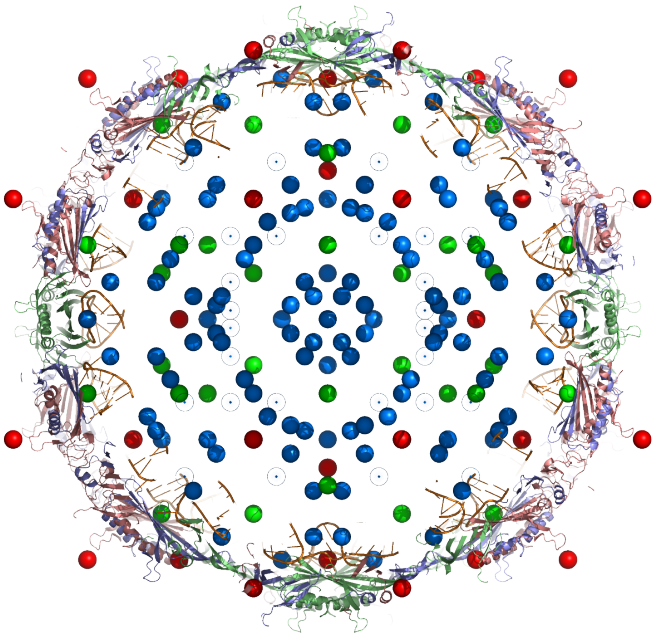
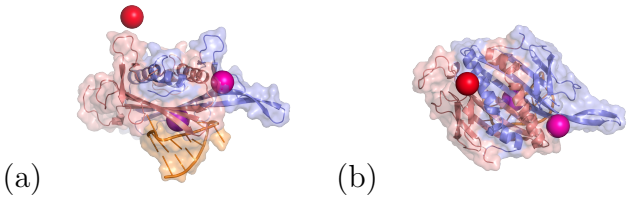
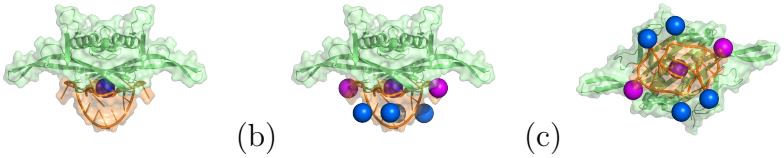


Figure 4.12: The best-fit point array (152 — green) with the next-best point array (163 — blue) with the common points (red) overlaid on the crystal structure of Bacteriophage MS2. As can be seen, they both match to the B chain towers (red protein), as well as the upper surface of the A chain protein (blue) and lower surface of the CC dimers (green). Morever, the locations of interest are close for both point arrays.



(a) (b)

Figure 4.13: The two best-fit point arrays matching to an AB dimer (A chain is blue, B chain is red) of MS2 from the (a) side and (b) top; they match the AB dimer identically.



(a) (b) (c)

Figure 4.14: (a) The best-fit point array (152) matching to a CC dimer of MS2 from the side; (b) The next best-fitting point array (163) matching to a CC dimer of MS2 from the side; (c) The next best-fitting point array (163) matching to a CC dimer of MS2 from underneath.

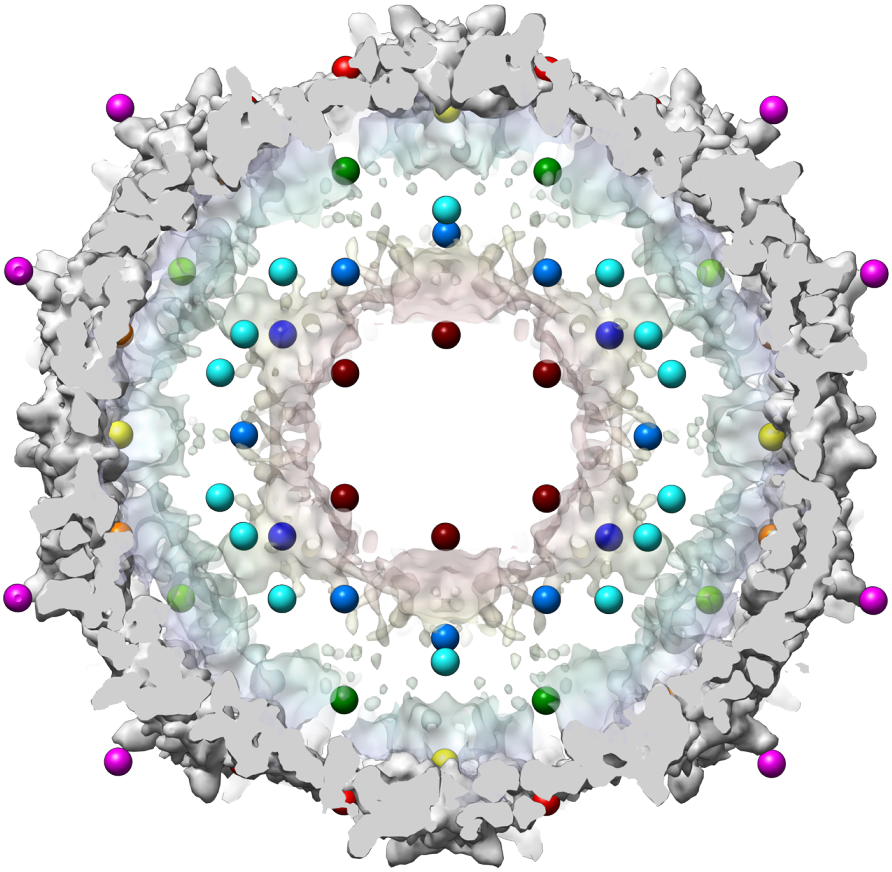


Figure 4.15: The best-fit point array overlaid on the cryo-EM structure [59, 104] of the genomic material (showing the double shell structure) and crystal structure [112] of the capsid of Bacteriophage MS2.

4.1.3 Bacteriophage GA

Bacteriophage GA has been solved to 3.4Å resolution [99] and is available from VIPER with PDB-ID `1gav`. It is very similar in structure to Bacteriophage MS2 despite relatively low sequence similarity, and its structure was solved by using the structure of MS2 as a template. Not particularly surprisingly, then, point array 152 is once again the best-fitting point array as shown in Table 4.4.

The best-fit point array and the common points for the other 9 arrays (being array 42) are shown superimposed together on the crystal structure in Figure 4.16. As can be seen in both that and Figures 4.17 and 4.18, they match the exterior of Bacteriophage GA differently (arrays 7 and 42 are not related by the method of Section 2.4), but yet both mark the potential contact point of the RNA to the CC dimer — the cyan point in Figure 4.18 — assuming Bacteriophage GA is similar to Bacteriophage MS2 in this respect.

It is remarkable that whilst the ensembles of runners-up for Bacteriophages MS2 and GA are different despite large overlaps (there are 7 point arrays in common) the best-fit point array remains constant. This implies that the algorithm may well be picking out the same best-fit array for evolutionarily related viruses, encapsulating the essential features of their geometries.

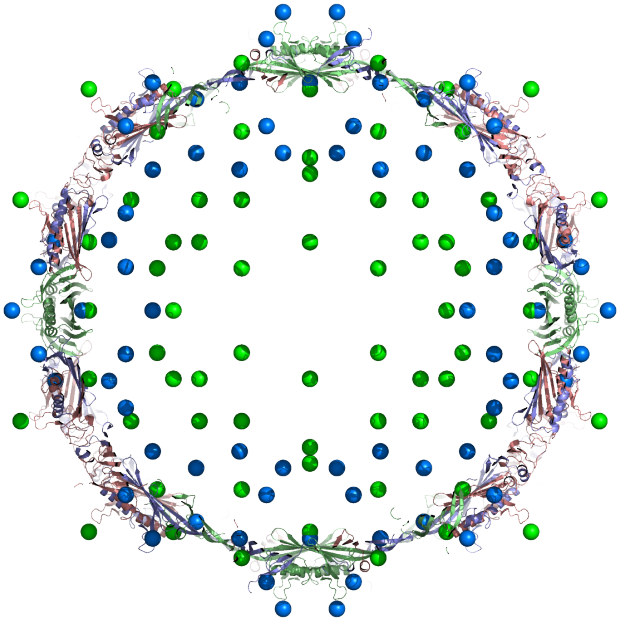
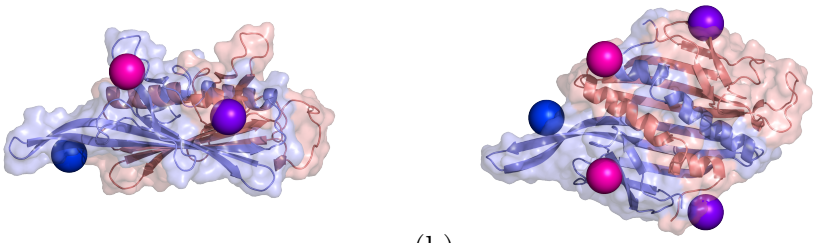


Figure 4.16: The best-fit point array (152 — green) and the common points for the next best 9 point arrays (879 — blue).

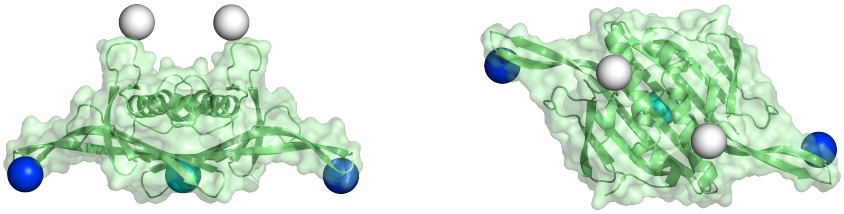


(a) (b)

Figure 4.17: An AB dimer of Bacteriophage GA with the points common to the 9 runner-up point arrays superimposed from (a) the side and (b) the top.

Combo Number	Outside	Inside	RMSD	Dist. to Tower	Final Score	Combo Radius	Shunt	Number of Hits	Prevalence
152	7	8	2.167	4.236	4.758	149.068	4.1	4	10
160	42	7	2.654	4.809	5.493	142.705	2.8	6	23
177	42	8	2.654	4.809	5.493	142.705	2.8	6	23
194	42	9	2.654	4.809	5.493	142.705	2.8	6	23
211	42	10	2.654	4.809	5.493	142.705	2.8	6	23
484	42	23	2.654	4.809	5.493	142.705	2.8	6	23
501	42	24	2.654	4.809	5.493	142.705	2.8	6	23
518	42	25	2.654	4.809	5.493	142.705	2.8	6	23
535	42	26	2.654	4.809	5.493	142.705	2.8	6	23
879	42	42	2.654	4.809	5.493	142.705	2.8	6	23

Table 4.4: The ten lowest scoring combination point arrays for Bacteriophage GA.



(a) (b)

Figure 4.18: A CC dimer of Bacteriophage GA with the points common to the 9 runner-up point arrays superimposed from (a) the side and (b) the top.

4.1.4 Tomato Bushy Stunt Virus

The structure of Tomato Bushy Stunt Virus (TBSV) has been solved to 2.9Å resolution and is deposited with PDB-ID `2tbv` [32]. It has a $T = 3$ structure made up of 180 copies of the same protein, although 25% of the protein structure is unknown [103]. One of the terminal arms of each of the capsomeres is not distributed icosahedrally, and so is averaged away. According to the literature the bulk of the unseen protein lies in a second, internal, shell, and "most of the RNA is sandwiched between the two protein shells." [103]. Moreover, the two shells are connected only at the 3-fold axes.

Table 4.5 shows a single best-fit point array, although the best-fit exterior array (29) occurs in 3 of the 10 lowest scoring arrays, and 5 of the others are the related point array 8. Out of interest, then, we present the best-fit point array with the next three arrays in Figure 4.19.

As can be seen, which is not unexpected, they are very similar, but each yields different information about the AB and CC dimers, as shown in close-up views in Figures 4.20 and 4.21. The bottom row in each figure shows all four point arrays superimposed on the dimers simultaneously to illustrate the different areas highlighted by the different point arrays.

Moreover, [103] offers some data on the interior structure of TBSV (see Figure 4.22) which shows that there is structured genomic material inside the capsid. Figure 4.23 shows the same four point arrays displayed previously, with the X-ray data, overlaid on this neutron-

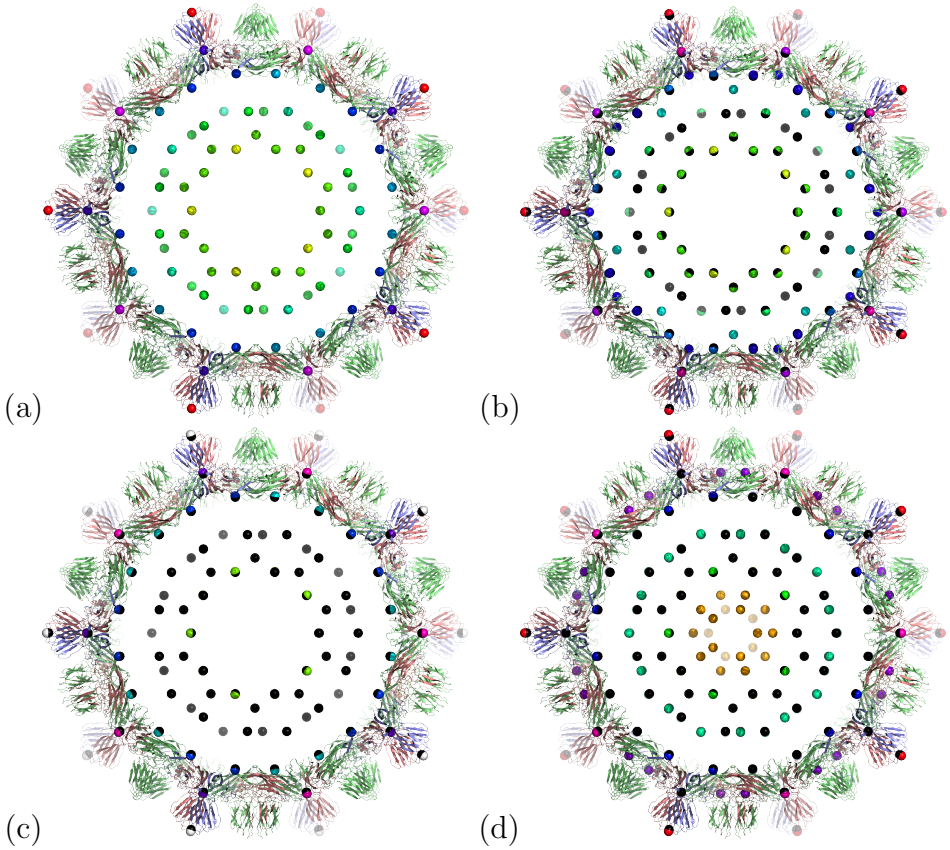


Figure 4.19: Tomato Bushy Stunt Virus with (a) the best-fit point array (580) coloured by radial level; (b) point array 575 coloured by radial level with point array 580 shown in black; (c) point array 170 coloured by radial level with point array 580 shown in black and (d) point array 175 coloured by radial level with point array 580 shown in black.

scattering density plot (aligned in the same way), and Figure 4.24 shows all four superimposed simultaneously. As can be seen, the closeness of the fit is remarkable, in all cases — the only discrepancy is between the X-ray data and the neutron-scattering data on each side, where some of the A and B proteins do not seem to be picked up by the neutron-scattering; the point arrays match very well.

Combo Number	Outside	Inside	RMSD	Dist. to Tower	Final Score	Combo Radius	Shunt	Number of Hits	Prevalence
580	29	54	1.921	5.922	6.226	174.139	-3.5	4	40
575	29	30	2.305	5.97	6.399	175.538	-2.1	4	43
170	8	9	2.786	5.973	6.591	175.638	-2	4	22
175	8	25	2.839	5.973	6.613	175.638	-2	4	36
182	8	47	2.73	6.133	6.713	180.335	2.7	7	5
171	8	10	3.27	6.004	6.836	176.537	-1.1	4	52
176	8	26	3.348	5.973	6.847	175.638	-2	4	52
94	35	4	1.781	6.86	7.087	167.715	-5	10	7
579	29	53	3.998	5.871	7.103	172.64	-5	5	14
769	37	41	1.885	6.86	7.114	167.715	-5	12	7

Table 4.5: The ten lowest scoring combination point arrays for Tomato Bushy Stunt Virus.

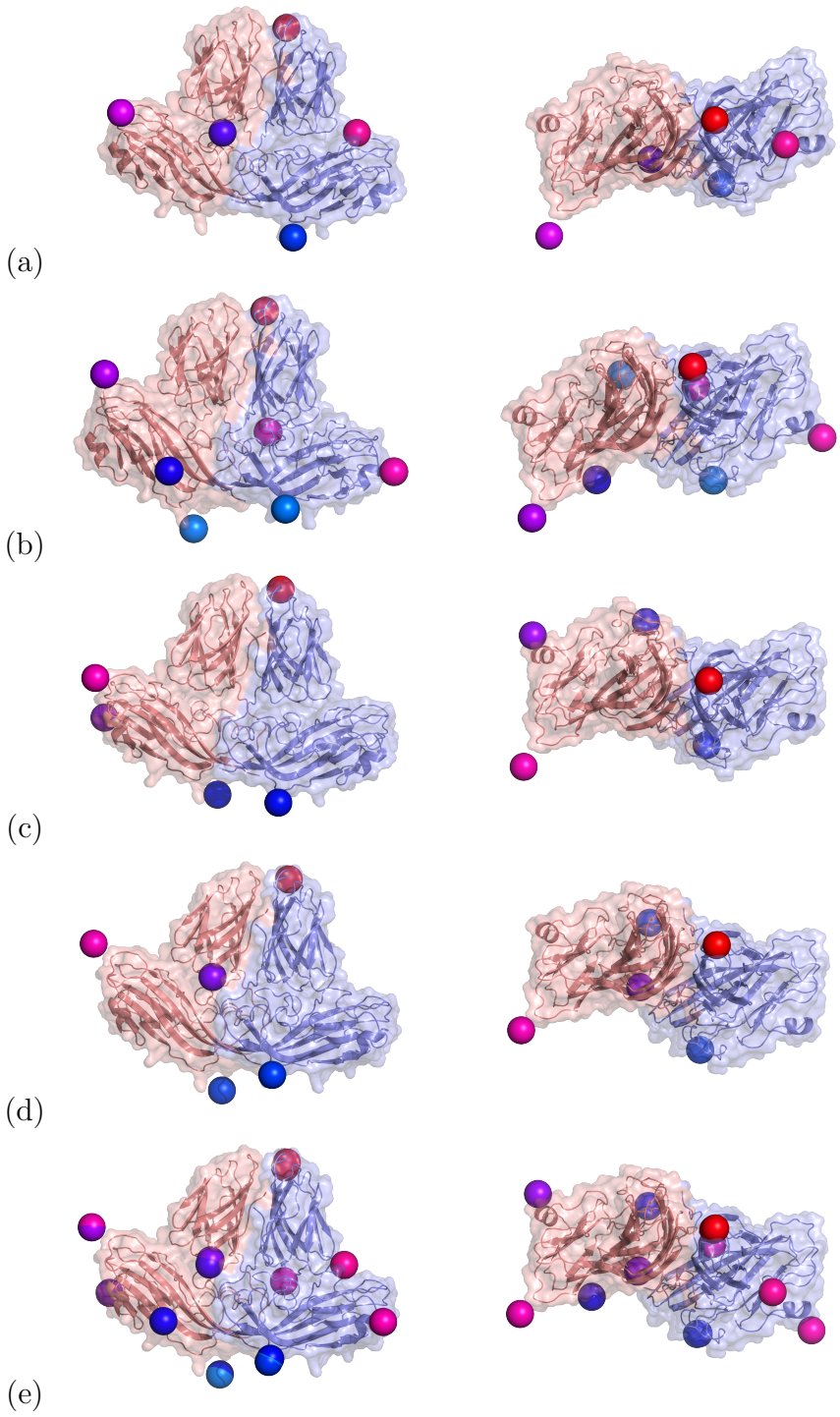


Figure 4.20: An AB dimer of Tomato Bushy Stunt Virus, in a side view (left column) and top view (right column) displaying point arrays (a) 580, (b) 575, (c) 175, (d) 170 and (e) all four combined, in rows from top to bottom, respectively.

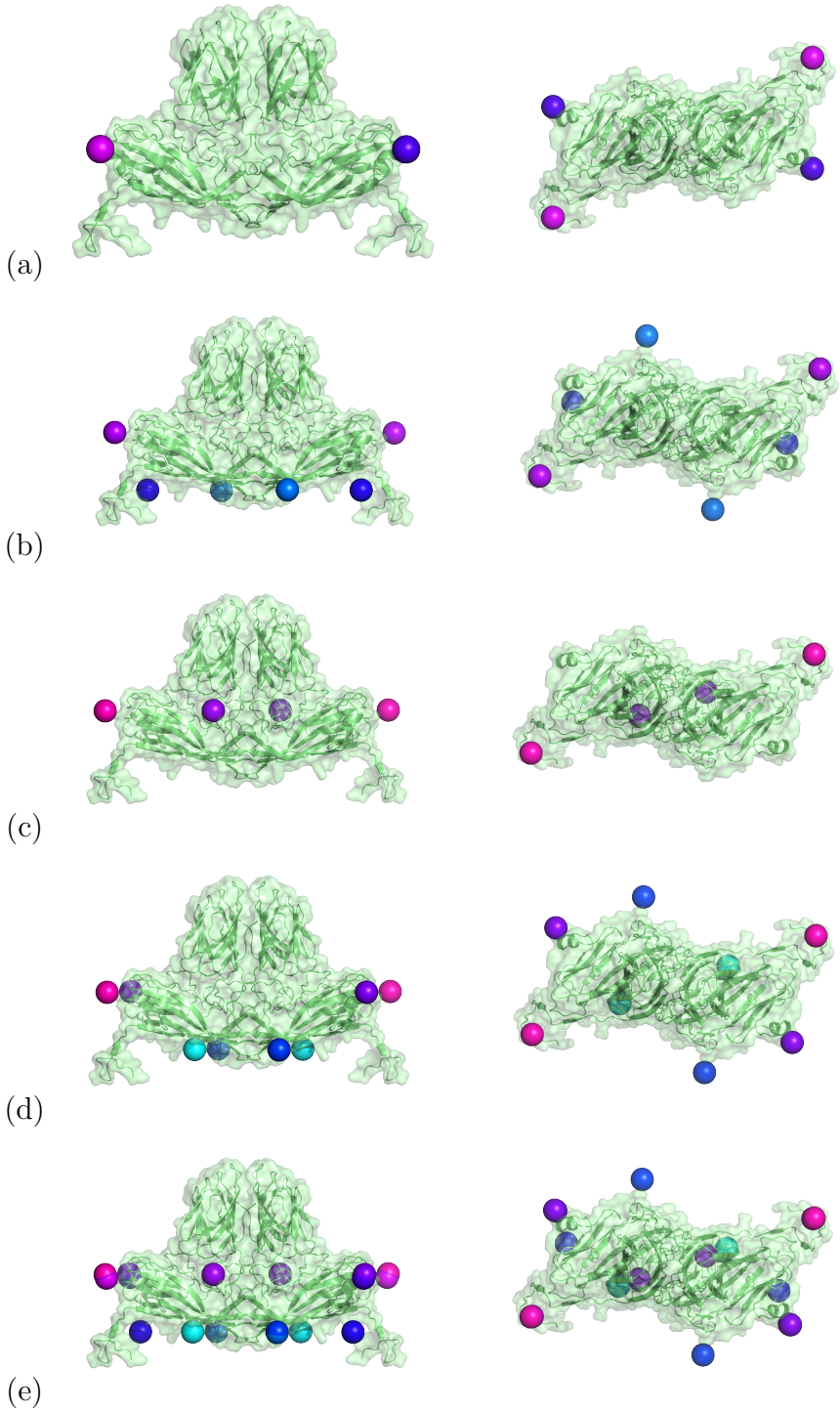


Figure 4.21: A CC dimer of Tomato Bushy Stunt Virus, in a side view (left column) and top view (right column) displaying point arrays (a) 580, (b) 575, (c) 175, (d) 170 and (e) all four combined, in rows from top to bottom, respectively.

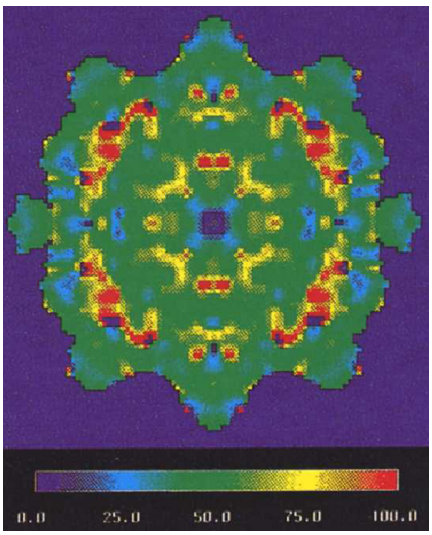


Figure 4.22: A 2-fold view through the neutron-scattering density results for TBSV [103]. According to the literature, areas below 55% correspond to protein, those between 55% and 70% to RNA, and other areas to solvent.

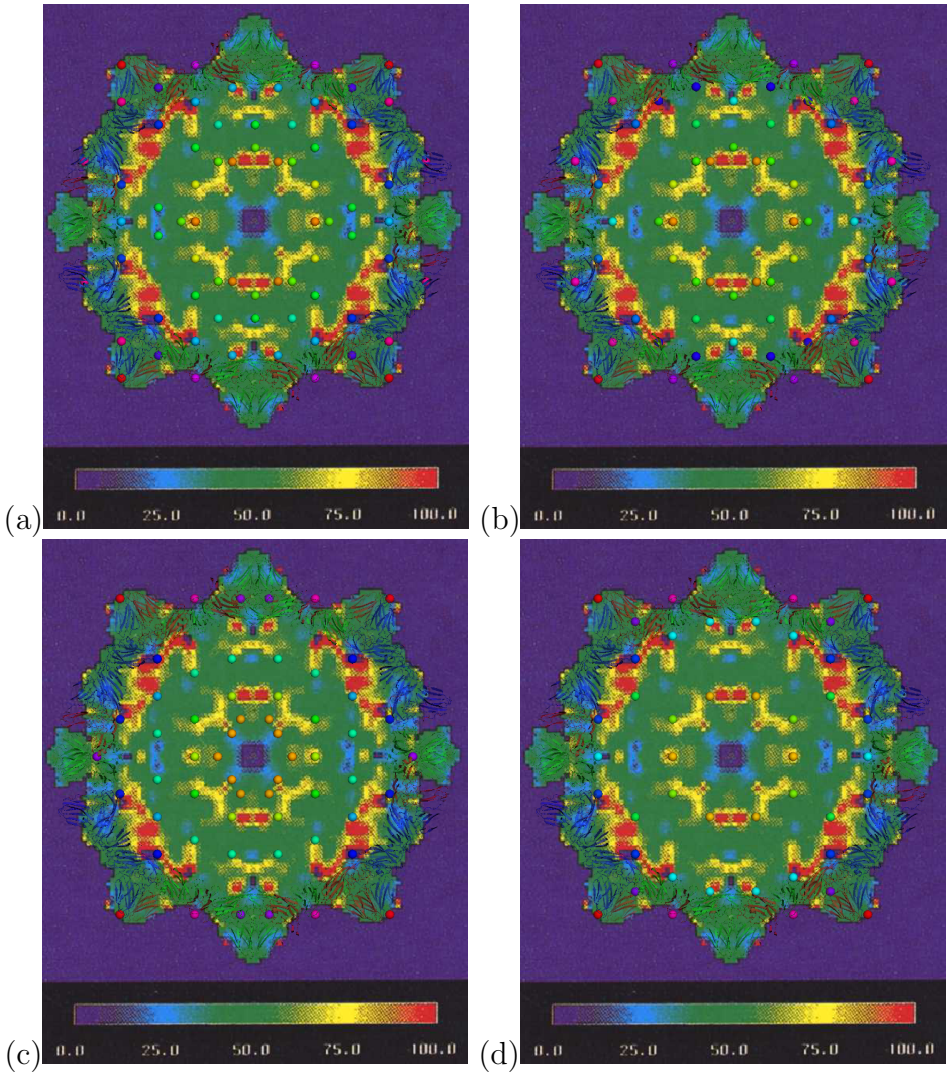


Figure 4.23: A 2-fold view through the neutron-scattering density results for TBSV [103] superimposed with point arrays (a) 580 (b) 575 (c) 175 and (d) 170. According to the literature, areas below 55% correspond to protein, those between 55% and 70% to RNA, and other areas to solvent.

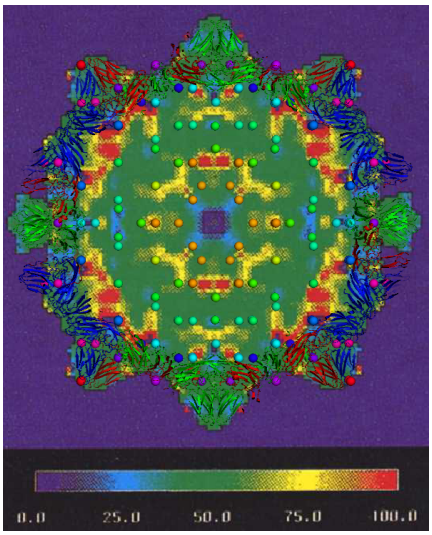


Figure 4.24: A 2-fold view through the neutron-scattering density results for TBSV [103] with all four point arrays (580, 575, 175 and 170) superimposed simultaneously. According to the literature, areas below 55% correspond to protein, those between 55% and 70% to RNA, and other areas to solvent.

4.2 Swelling Transformations

The virus discussed in this section, Cowpea Chlorotic Mottle Virus, undergoes a swelling transformation which is a putative intermediate in the infection process. In this section, we examine the start and end states of the transformation as this can be used as input to examine the mathematical transitions between the point arrays [35]. The number of combinations of start and end states is large, so it is the hope that this work can provide information on where effort may be most fruitfully spent.

This particular virus was picked for this project as it was of a suitable size for the point arrays to work best on, and had suitably well-defined start and end states. There is no direct link in this work between the start and end states and their associated point arrays; it is hoped that the transition can give insights into the swelling of the capsid.

4.2.1 Cowpea Chlorotic Mottle Virus

The structure of Cowpea Chlorotic Mottle Virus (CCMV) has been solved to 3.2Å resolution and has been deposited with PDB-ID `1cwp` [95]. It is a $T = 3$ virus with a capsid formed of 180 chemically identical proteins covering the expected three quasi-equivalent locations.

Table 4.6 illustrates a situation that demands care — four of the lowest-scoring point arrays occur at a shunt of -5Å. This can artificially lower their prevalence due to the range of acceptable scalings butting up against the usual cutoff of -5Å. We therefore re-ran the algorithm

using a wider search range. This gives the results shown in Table 4.7. Point arrays 137 and 919 are removed from contention because of their low prevalences (2 — which are no longer artificially lowered because of the increased search range) and we see that 917 is the best-fit point array. Note that this point array is identical to the best-fit point arrays given in [35] as regards its overlap with capsid material. However, 917 exhibits one additional match to capsid protein, which is in an excellent position between the B and C chains and is shown in Figure 4.25. The data communicated to Indelicato *et al.* was preliminary data communicated privately before the algorithm (and in particular the first filter of Section 3.3) was complete.

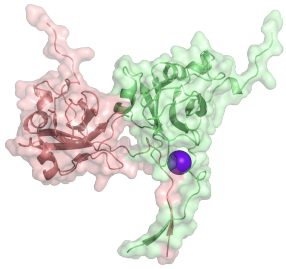
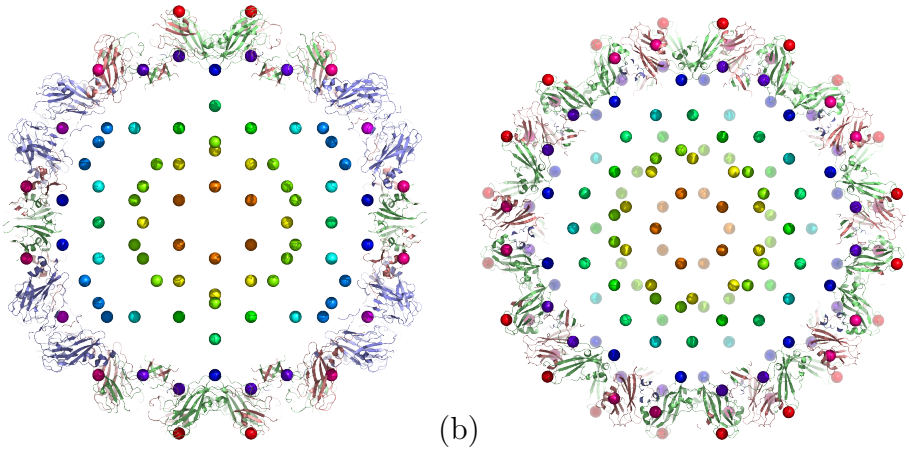


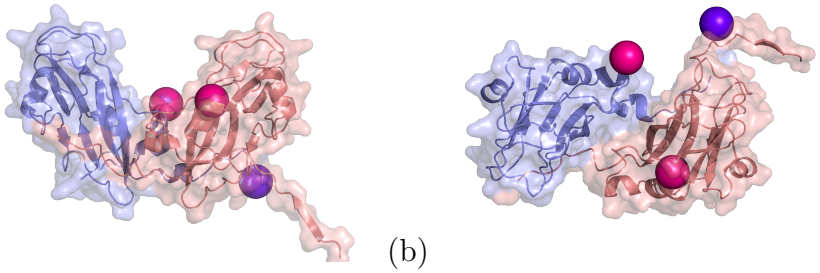
Figure 4.25: The extra point of the best-fit point array (917) compared to those given in [35]. It matches the crystal structure of a complex of B and C chain proteins in CCMV marking where the terminal arm of the B chain protein fits into a hollow in the C chain protein.

Figure 4.26 (both (a) and (b)) show that the best-fit point array fits to the top and bottom of the capsid material well — delimiting its thickness — and Figures 4.27 and 4.28 show the points matching to the AB and CC dimers respectively.



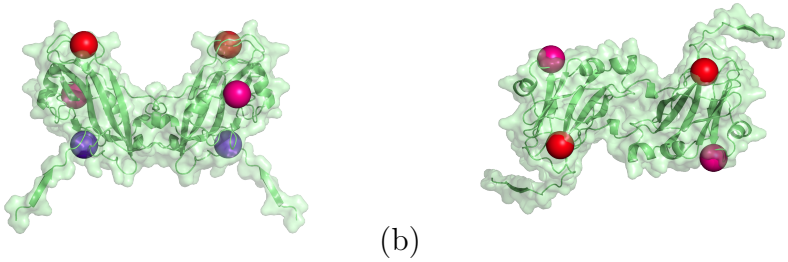
(a) (b)

Figure 4.26: The best-fit point array matching the crystal structure of CCMV down (a) a 2-fold axis and (b) a 5-fold axis showing how it matches the extent of the capsid.



(a) (b)

Figure 4.27: The best-fit point array matching the crystal structure of an AB dimer of CCMV from (a) the side and (b) the top.



(a) (b)

Figure 4.28: The best-fit point array matching the crystal structure of a CC dimer of CCMV from (a) the side and (b) the top.

Combo Number	Outside	Inside	RMSD	Dist. to Tower	Final Score	Combo Radius	Shunt	Number of Hits	Prevalence
917	44	46	0.892	6.983	7.039	136.463	-5	4	23
396	19	21	1.312	6.982	7.104	136.462	-5	4	3
112	19	5	1.51	6.982	7.143	136.462	-5	5	23
520	44	25	1.504	6.983	7.143	136.463	-5	5	3
406	19	40	1.704	6.992	7.196	136.661	-4.8	6	3
547	27	28	0.873	7.177	7.23	136.793	-4.6	5	19
222	27	11	0.922	7.177	7.236	136.793	-4.6	4	31
235	27	12	0.922	7.177	7.236	136.793	-4.6	4	31
248	27	13	0.922	7.177	7.236	136.793	-4.6	4	31
546	27	27	0.922	7.177	7.236	136.793	-4.6	4	31

Table 4.6: The ten lowest scoring combination point arrays for Cowpea Chlorotic Mottle Virus.

Combo Number	Outside	Inside	RMSD	Dist. to Tower	Final Score	Combo Radius	Shunt	Number of Hits	Prevalence
137	19	6	0.462	6.972	6.987	136.262	-5.2	4	2
919	44	48	0.462	6.972	6.987	136.262	-5.2	4	2
917	44	46	0.938	6.972	7.035	136.262	-5.2	4	25
396	19	21	1.332	6.972	7.098	136.262	-5.2	4	5
298	32	15	2.787	6.54	7.11	140.14	-6.8	13	17
112	19	5	1.49	6.972	7.129	136.262	-5.2	5	25
520	44	25	1.513	6.972	7.134	136.262	-5.2	5	5
547	27	28	0.922	7.114	7.173	135.595	-5.8	5	24
222	27	11	0.962	7.114	7.179	135.595	-5.8	4	35
235	27	12	0.962	7.114	7.179	135.595	-5.8	4	35

Table 4.7: The ten lowest scoring combination point arrays for Cowpea Chlorotic Mottle Virus.

4.2.2 Cowpea Chlorotic Mottle Virus — Swollen Form

The swollen form of CCMV is given as a proposed model [65, 97] available from the VIPER website with "PDB-ID" `ccmv_swln_1`. Once again, the data presented in [35] is based on preliminary data, that is, it was obtained before the first filter (of Section 3.3) was in place, communicated privately, just as in Section 4.2.1. The best-fit point array here (547 – composed of arrays 27 and 28) is identical as regards its overlap with the capsid to those point arrays presented in [35] except for the addition of one point just under the A chain protein (the blue point under the red protein in Figure 4.30(a)); it is this point that has raised the total score slightly (i.e. made the RMSD slightly worse) from 4.243 to 4.482 but increased the number of hits above the threshold value. Figure 4.29 shows the best-fit point array overlaid with the X-ray crystallography data viewed down a 5-fold axis, which illustrates the match this point array has to the capsid, representing its thickness accurately.

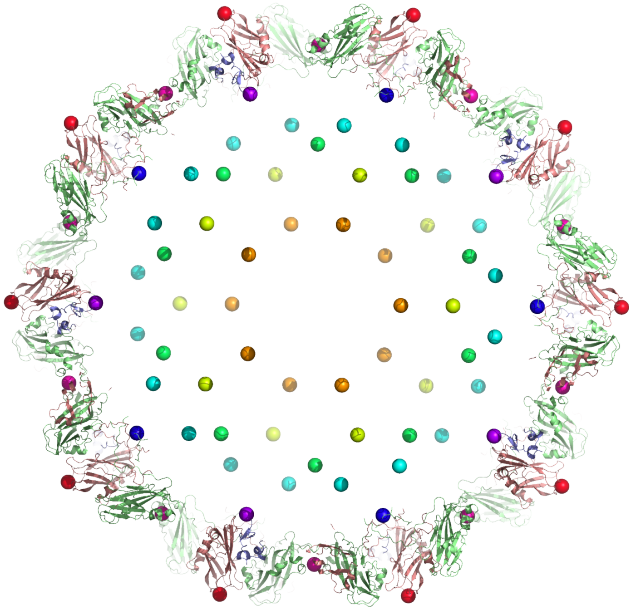
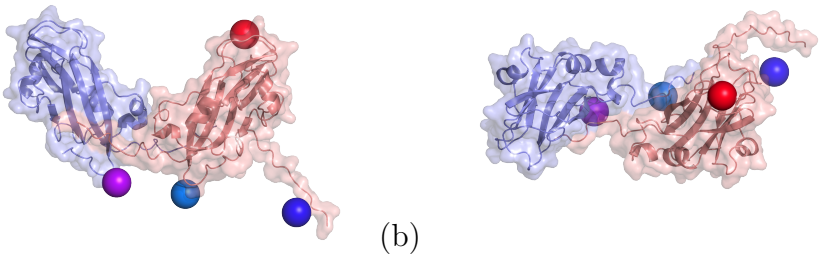


Figure 4.29: The best-fit point array matching the crystal structure of swollen CCMV, viewed down a 5-fold axis.

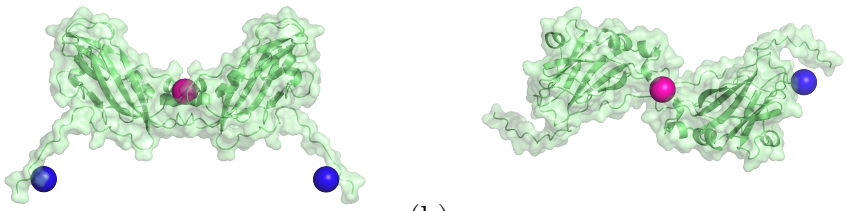


(a) (b)

Figure 4.30: The best-fit point array matching the crystal structure of an AB dimer of swollen CCMV, (a) a side view and (b) a top view.

Combo Number	Outside	Inside	RMSD	Dist. to Tower	Final Score	Combo Radius	Shunt	Number of Hits	Prevalence
547	27	28	2.165	3.925	4.482	162.577	-2.4	4	77
369	18	19	3.372	3.259	4.689	162.374	-3	6	6
936	45	48	3.865	3.255	5.053	162.174	-3.2	4	79
370	18	20	3.974	3.255	5.137	162.174	-3.2	4	67
86	18	4	3.998	3.279	5.171	163.374	-2	5	36
917	44	46	2.428	4.582	5.186	162.743	-2.6	4	77
937	45	49	4.218	3.255	5.328	162.174	-3.2	4	79
395	19	20	2.801	4.593	5.38	163.142	-2.2	5	39
406	19	40	2.847	4.582	5.394	162.742	-2.6	4	66
396	19	21	2.888	4.582	5.416	162.742	-2.6	4	77

Table 4.8: The ten lowest scoring combination point arrays for the swollen form of Cowpea Chlorotic Mottle Virus.



(a) (b)

Figure 4.31: The best-fit point array matching the crystal structure of a CC dimer of swollen CCMV, from (a) a side view and (b) a top view.

4.3 Smaller and Larger Viruses

The viruses considered so far have been $T = 3$ viruses. Here, a smaller ($T = 1$) virus and a larger ($T = 7d$) virus are analysed.

4.3.1 Satellite Tobacco Mosaic Virus

First considered is Satellite Tobacco Mosaic Virus (STMV), which is a $T = 1$ virus with the expected 60 identical capsid proteins. It is deposited on VIPER with PDB-ID `1a34` [57] and has been resolved to a resolution of 1.81Å. The `pdb`-file has fragments of RNA present with 59% of the genome visible [57], which form part of a cage of RNA that comprises approximately 80% of the virus' RNA [58].

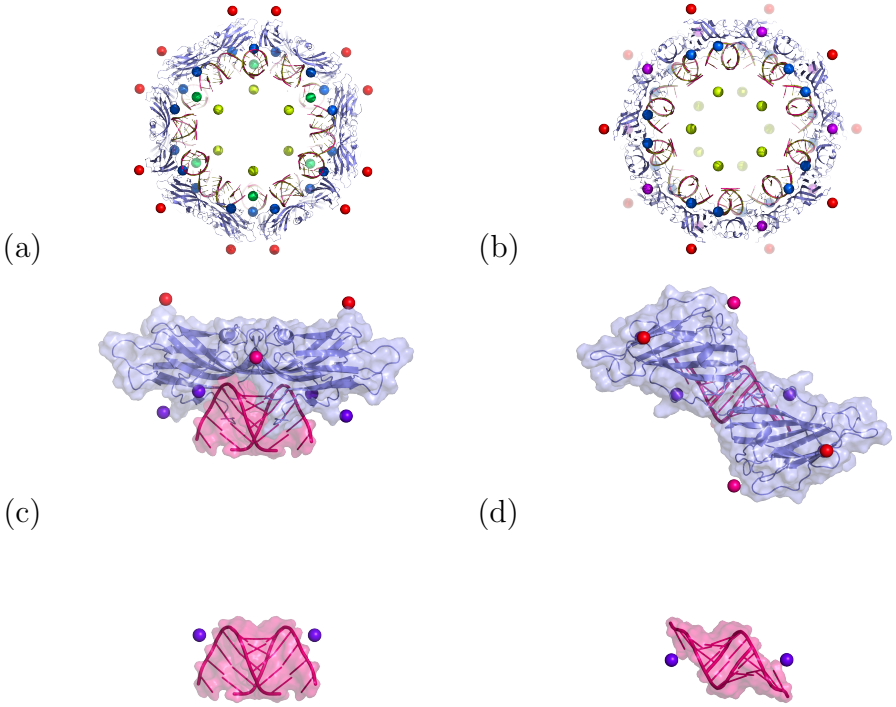
However, STMV has a radius of only around 90Å, meaning that the standard point arrays, while they would have the usual number of points within the capsid, would provide a greater density of points there and hence too many fine detail structural conditions. We therefore use only the pure point arrays given in Tables A.6, A.7 and A.8 in Appendix A (page 172).

If only the 55 pure point arrays are considered, relaxing the normal restriction that there must be three points of contact between the point array and the virus, the results are those shown in Table 4.9. Figure 4.32 shows this best-fit pure point array, that is, point array 8 by itself, matching to the capsid. As can be seen in Figure 4.32(a) and (b), the best-fit point array matches the extent of the capsid well, while still providing some information on the interior of the virus; Figure 4.32(c) and (d) reinforce this, showing a protein dimer with attached RNA.

Number	RMSD	Dist. to Tower	Final Score	Combo Radius	Shunt	Number of Hits	Prevalence
8	0.731	2.781	2.875	93.065	4.7	3	4
29	0.731	2.781	2.875	93.065	4.7	3	4
12	1.595	9.398	9.533	81.618	-5	2	68
11	3.146	10.222	10.695	88.771	2.2	3	6
13	3.146	10.222	10.695	88.771	2.2	3	6
25	5.637	13.913	15.012	94.103	4.8	4	3
24	13.409	7.121	15.183	110.872	2.4	5	27

Table 4.9: The only seven pure point arrays for Satellite Tobacco Mosaic Virus.

Finally, Figure 4.32(e) and (f) show the RNA fragment by itself with the point array, displaying how the fragment is bracketed by this point array.



(a) (b) (c) (d) (e) (f)

Figure 4.32: View down the (a) 3-fold axis and (b) the 5-fold axis of STMV with the best-fit pure point array. (c) Side view and (d) top view of the protein-RNA contact with the best-fit pure point array. (e) Side view and (f) top view of the RNA alone, with the best-fit pure point array. The fit to protein is lessened, so the Number of Hits becomes too low to escape the usual filter and the match to the RNA is greatly lessened.

4.3.2 Simian Virus 40

The second virus in this section is Simian Virus 40, or SV40. It is available from VIPER with PDB-ID `1sva` [96] and is a $T = 7d$ virus (recall Figure 1.11 showing how the various T-numbers are found; the $T = 7$ triangle of Figure 1.10 is $T = 7l$, so the layout of SV40 is the mirror image of that).

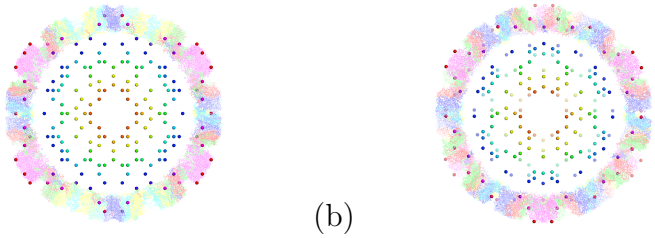
In the same way that STMV is smaller than the viruses previously considered, SV40 is larger (recall STMV was approximately 90Å in radius, SV40 is around 250Å; it may not be particularly large compared to some viruses such as those seen in the Introduction — recall Mimivirus in Figure 1.4 — but the data requirements increase with the square of the radius!), so point arrays with finer detail and more constraints are used; these are the second-iteration point arrays from Section 2.5.2. However, the same range of shunts from +5Å to −5Å is used, as the size of the outermost features of the virus have not scaled similarly in size. Table 4.10 gives the results, showing that point array 30 is the best fitting point array unambiguously, with a combined score of 2.811 compared to the next best array with a score of 8.713, over twice as big.

Figure 4.33 shows the best-fit point array matching to the crystal structure of SV40 from both 2-fold and 3-fold axes, matching the surface of the capsid proteins well while still providing structural constraints within the virus. Figures 4.34 and 4.35 show the best-fit array matching to the 5-fold and quasi-5-fold pentamers of SV40: the proteins are delineated well, and the two pentamers are matched in

Array Number	RMSD	Dist. to Tower	Final Score	Combo Radius	Shunt	Number of Hits	Prevalence
30	1.48	2.39	2.811	243.956	4.5	6	6
8	8.379	2.39	8.713	265.125	4.5	10	6
12	8.857	4.758	10.054	282.646	0.1	4	27
28	3.709	9.548	10.243	249.282	-3.9	6	12
52	10.213	3.371	10.755	271.723	-5	10	7
53	9.506	6.289	11.398	274.751	-4	16	9
13	8.609	8.702	12.241	276.999	-3.3	5	49
51	11.372	6.462	13.079	267.015	-1.5	8	29
11	12.735	9.818	16.08	295.206	-4.1	8	55

Table 4.10: The scoring second iteration pure point arrays for Simian Virus 40.

different places by the points. In particular, the positions of the two different types of C-terminal arm conformations are picked out, showing how even a non quasi-equivalent virus can fit to this theory.

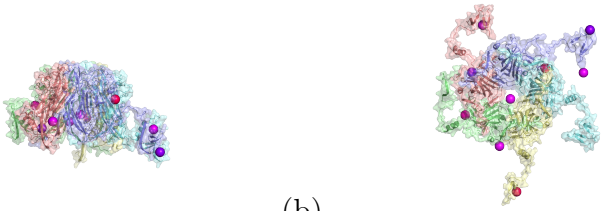


(a) (b)

Figure 4.33: The second iteration point array 30 viewed down (a) a 2-fold axis and (b) a 3-fold axis.

(a) (b)

Figure 4.34: The 5-fold pentamer of Simian Virus 40 with second-iteration point array 30 from (a) the side and (b) the top.



(a) (b)

Figure 4.35: The quasi-5-fold pentamer of Simian Virus 40 with second-iteration point array 30 from (a) the side and (b) the top.

4.4 Polymorphic Interiors

Some viruses do not show any ordered features in the publicised densities. As argued below, this may be because there is a polymorphic genome organisation (that is, the genome may take up one of a number of different arrangements) in these viruses due to the fact that there may be fewer boundary conditions (from the best-fit point array(s)) in these cases.

4.4.1 Hepatitis B

The $T = 4$ structure of Hepatitis B, solved to a resolution of 3.3Å, is deposited at the Protein DataBank with PDB-ID `1qgt`[116] and is available from VIPER. The capsid is formed of two protein dimers (AB and CD), both of which have an unusually large number of $\alpha-$helices.

Table 4.11 shows the results of our best-fit algorithm, which give one point array with a clear lead over the others. Note that point arrays 10 and 27 are structurally related as discussed in Section 2.4, which is why the second-best point array in the table has a *Distance to Tower* score similar to the best-fit point array. Figure 4.36 shows the best-fit point array superimposed on a cross-section of the crystal structure, viewed down a 5-fold axis. Array points are situated on the towers of the CD dimers (see Figure 4.37 (c)) — which, incidentally, lie precisely on the intersection of the "crossbars" of the kite — and the pink and cyan points bracket the capsid from above and below, defining its thickness (best displayed in the full picture in Figure 4.36, but can also be seen to mark the extent of the non-tower portion of

the dimer in Figure 4.37 (a)).

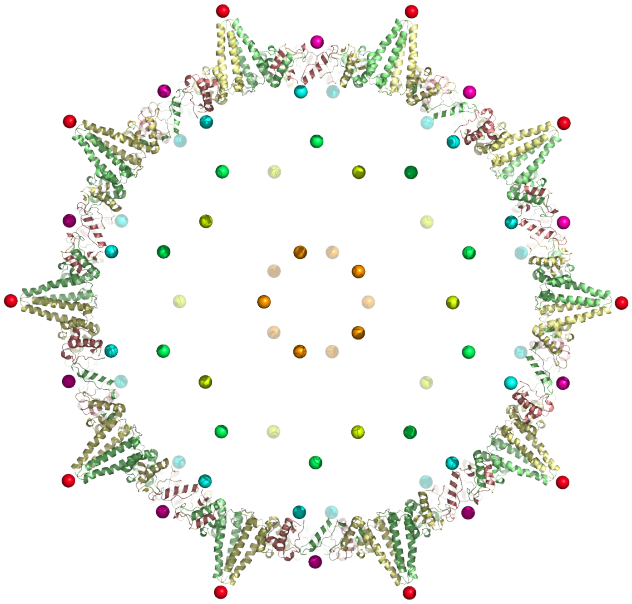


Figure 4.36: A view down the 5-fold axis showing Hepatitis B with the best-fit point array 222 coloured by radial level.

Once again, there are points in the capsid interior, and also for this virus, cryo-EM data are available (EMDB-ID 1400 — although this includes the envelope [93], and also from private communication with Roseman [86]) to probe the predictions of our theory. Figure 4.38 shows a cross-sectional view through the middle of the density, displaying the envelope (purple), capsid protein (cream) and DNA (light blue). The outer shell of DNA clearly occupies the area between the cyan and green points, although there is a further blob of density around the origin. Like the density in the centre of Pariacoto Virus in Figure 4.3, though, this most likely corresponds to noise or disordered genomic material as it occurs at lower signal strength, but it is still interesting that the furthest extent of it reaches almost exactly to the

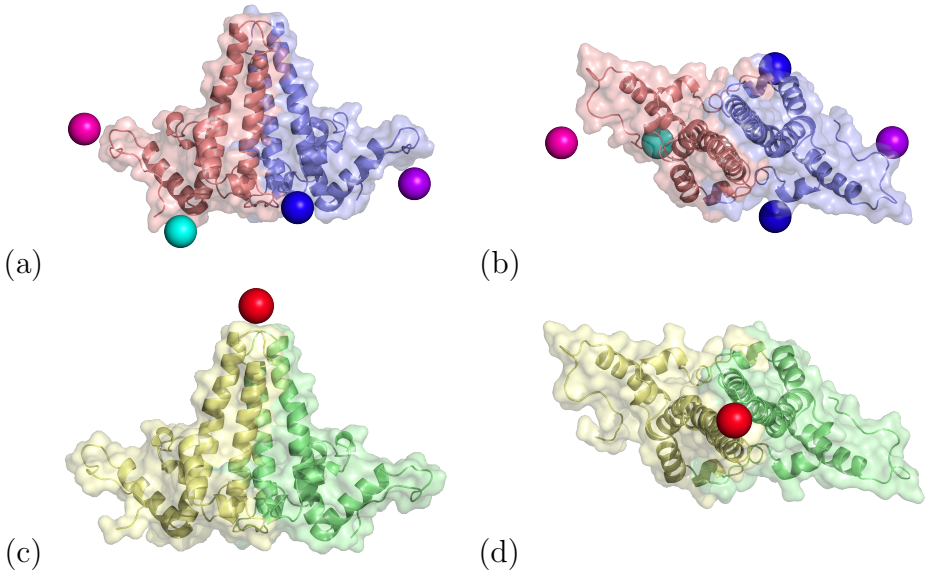


Figure 4.37: The best-fit point array for Hepatitis B (222) against an AB dimer from (a) the side (b) the top and a CD dimer shown in a (c) side view and (d) top view.

yellow points.

Figures 4.39 and 4.40 show cryo-EM data for Hepatitis B with first RNA and and then DNA. The RNA results show a clear shell of RNA (light blue) within the capsid (cream to burgundy), bounded between the cyan and light green points, in a manner very similar to that in Figure 4.38. The DNA results are less good, and are shown at two different threshold levels: 0.33 on the left and 0.25 on the right. Figure 4.40(a) shows turns of DNA in the expected position between the cyan and light green points, but nothing else, while Figure 4.40(b) shows fragmentary information on density just above the orange points and below the yellow, indicating there may well be more genomic material here.

Combo Number	Outside	Inside	RMSD	Dist. to Tower	Final Score	Combo Radius	Shunt	Number of Hits	Prevalence
222	27	11	4.064	1.619	4.375	178.03	3.8	4	13
210	10	26	8.48	1.591	8.628	178.092	0.7	6	52
769	37	41	7.572	5.853	9.57	182.888	-0.1	11	14
716	35	38	7.623	5.853	9.61	182.888	-0.1	11	38
121	37	5	7.898	5.853	9.83	182.888	-0.1	10	3
453	37	21	7.912	5.853	9.841	182.888	-0.1	10	42
119	35	5	7.969	5.853	9.887	182.888	-0.1	10	52
426	35	20	7.988	5.853	9.903	182.888	-0.1	10	52
376	35	18	8.077	5.853	9.974	182.888	-0.1	13	40
768	37	40	8.014	5.98	9.999	186.879	3.7	13	23

Table 4.11: The ten best-fit combination point arrays for Hepatitis B.

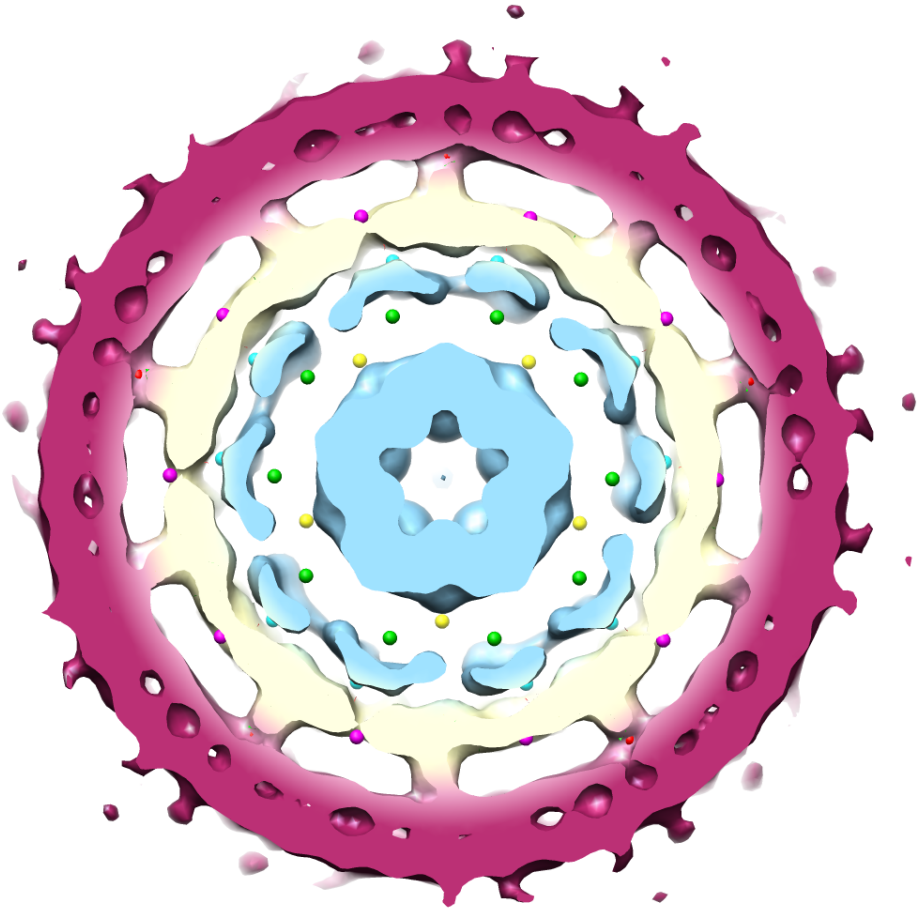


Figure 4.38: A view down the 5-fold axis of Hepatitis B showing cryo-EM data [93] in comparison with the best-fit point array (222 — coloured by radial level). It displays the envelope (purple), capsid protein (cream) and DNA (light blue).

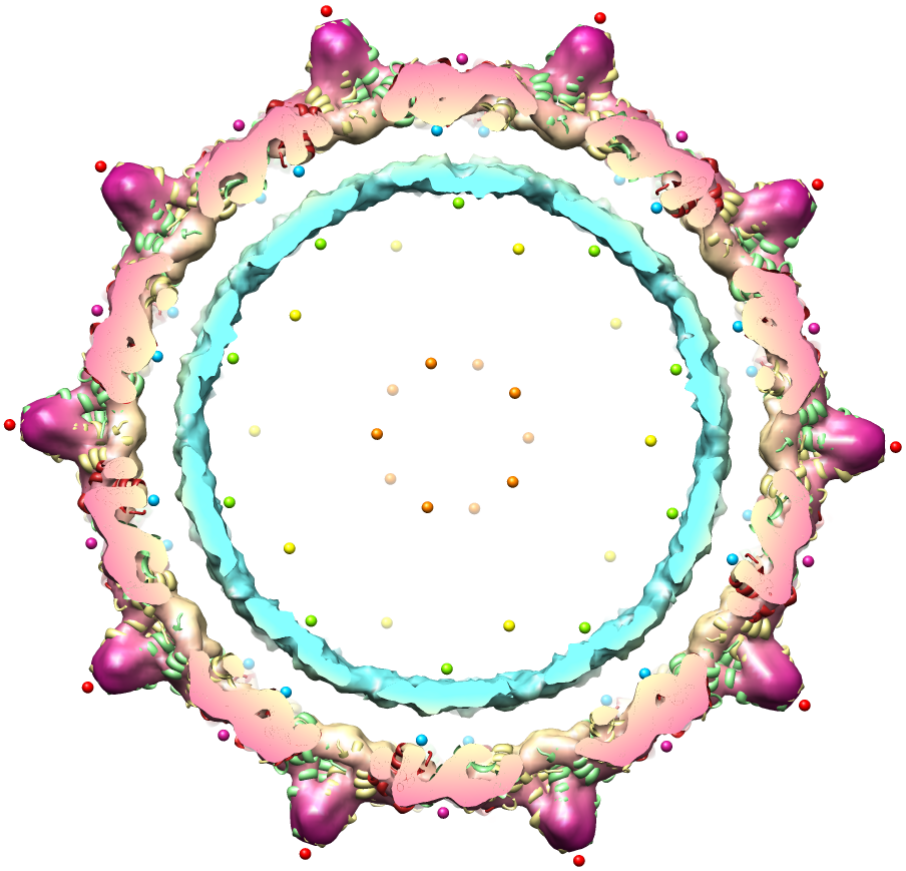


Figure 4.39: A view down the 5-fold axis of Hepatitis B showing cryo-EM data [86] in comparison with the best-fit point array (222 — coloured by radial level). This is displayed at a level of 4.57 showing a very regular shell of RNA (light blue) between the cyan and light green points.

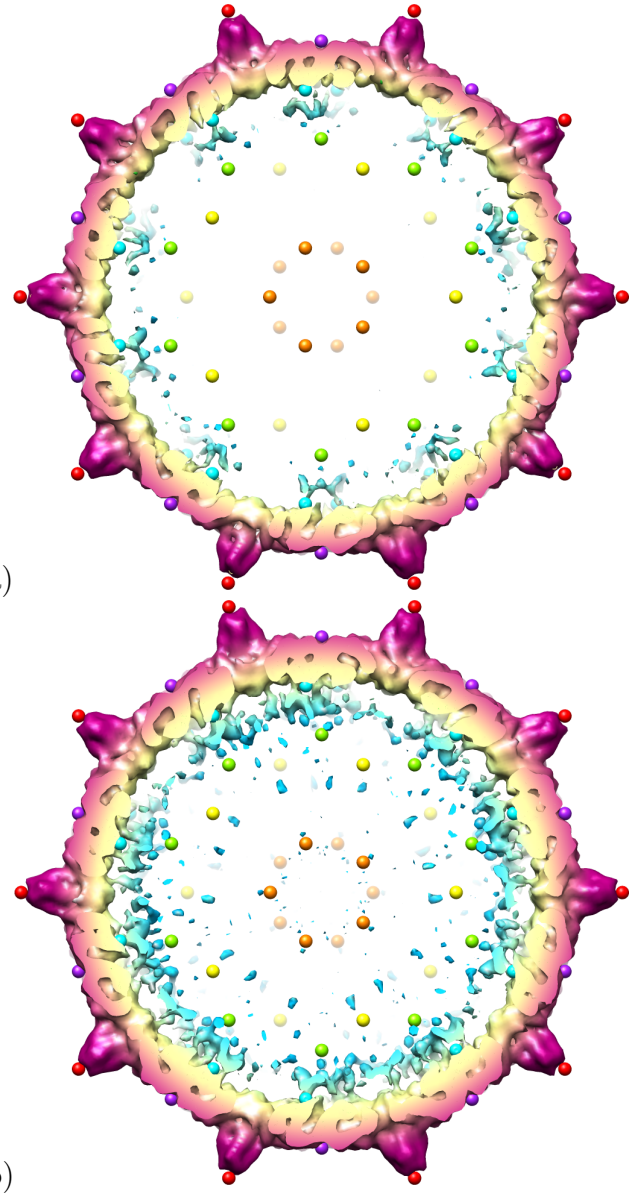


Figure 4.40: A view down the 5-fold axis of Hepatitis B showing cryo-EM data [86] in comparison with the best-fit point array (222 — coloured by radial level). (a) A level of 0.33, showing turns of DNA (light blue) between the cyan and light green points. (b) A level of 0.25 which brings in a little more of the density, notably close to the orange and yellow points.

4.4.2 Tobacco Necrosis Virus

Tobacco Necrosis Virus (TNV) is a $T = 3$ virus; an early `pdb`-file is available from VIPER with PDB-ID 1tnv [3] but this lacks side-chain information. The `pdb`-file with PDB-ID 1c8n [76] resolves the structure to 2.25Å.

Table 2.7 shows that the top 10 point arrays, with one exception, are pure point array 1 as the exterior composed with some other point array for the interior. The exception (1083) has point array 55 as the exterior, and this is twinned with 1 in the sense of Section 2.4.

Note that in Figure 2.7, though, the extent of point array 1 overlaps with the other point arrays that it is paired with; however, these lie entirely below the median line for point array 1 which only has 6 definable radii. Thus, with the exception of point arrays 3, 36 and 19, they do not overlap with the capsid. This is borne out by the "Number of Hits" column of Table 4.12 which shows 4 points interacting with the capsid, whereas combination point arrays 3, 20 and 12 show more than this (6, 6 and 5 points respectively). Point array 3, though, has a very low prevalence (3) so would be rejected as modelling the capsid particularly well; however, due to the similarities with the rest of the ensemble of best-fit point arrays, it is kept in the discussion.

Figure 4.41 shows eight of the best-fit point arrays down a 5-fold axis, while Figure 4.42 shows point array 1 (i.e. the points in common across the majority of the best-fit point arrays) down a 5- and 3-fold axis. All eight are different, giving different minimal radii of points of interest. There are, however, no areas over-represented if all the

point arrays are overlaid, over and above what is shown in Figure 4.42 which shows the points in common. It could be the case, then, that these points in common indicate common structural features of all the different organisations compatable with the symmetry critera.

Figure 4.43 shows point arrays 1, 3 and 20 superimposed on a trimer of TNV. As can be seen in images (a) and (b), the points in common to the best-fit point arrays are in a pentagonal arrangement around the trimer; the uppermost point (red) delimits the outermost points of the capsid and the purple ones mark both the more common outer layer of the capsid as well as the boundaries between trimers.

The additional points of point arrays 3 and 20 include, for both of these, points marking the lowest radius of the capsid which can most easily be seen in Figure 4.43 (d) and (f). Point array 3 includes a point marking the base of the A chain protein (which is probably the highest place of the interior surface of the capsid), while that included in 20 only marks the 2-fold axis and point of closest approach between two trimers on that edge.

Combo Number	Outside	Inside	RMSD	Dist. to Tower	Final Score	Combo Radius	Shunt	Number of Hits	Prevalence
3	1	3	4.632	12.638	13.46	160.326	1.1	6	3
20	1	36	4.839	12.638	13.533	160.326	1.1	6	11
12	1	19	5.244	12.638	13.683	160.326	1.1	5	11
1083	55	55	5.618	12.638	13.83	160.326	1.1	4	11
1	1	1	5.618	12.638	13.831	160.326	1.1	4	11
4	1	4	5.618	12.638	13.831	160.326	1.1	4	11
5	1	5	5.618	12.638	13.831	160.326	1.1	4	11
6	1	6	5.618	12.638	13.831	160.326	1.1	4	11
13	1	20	5.618	12.638	13.831	160.326	1.1	4	11
14	1	21	5.618	12.638	13.831	160.326	1.1	4	11

Table 4.12: The ten lowest scoring combination point arrays for Tobacco Necrosis Virus (1c8n).

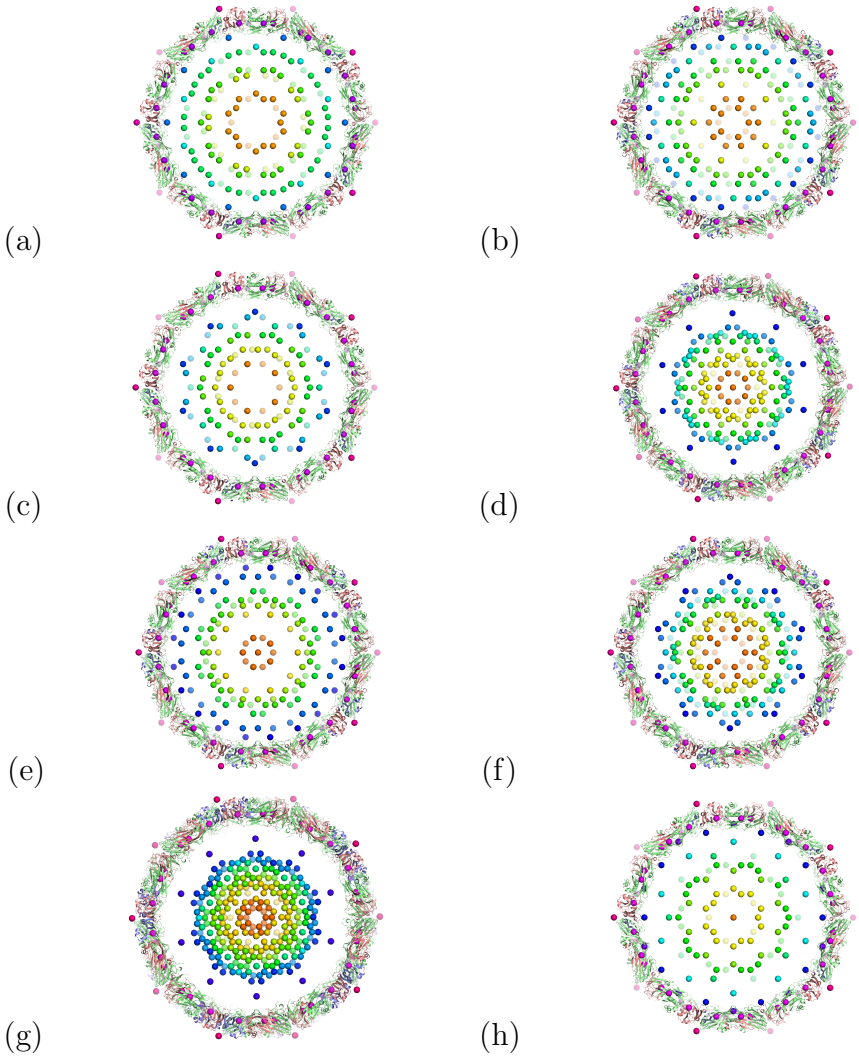
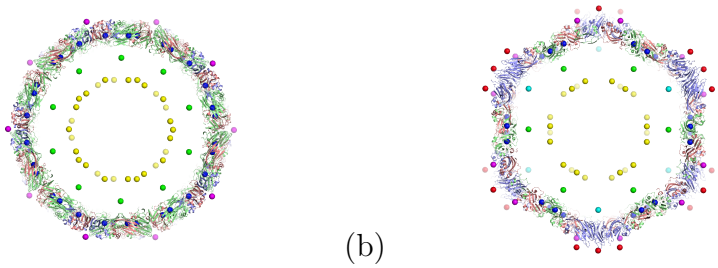
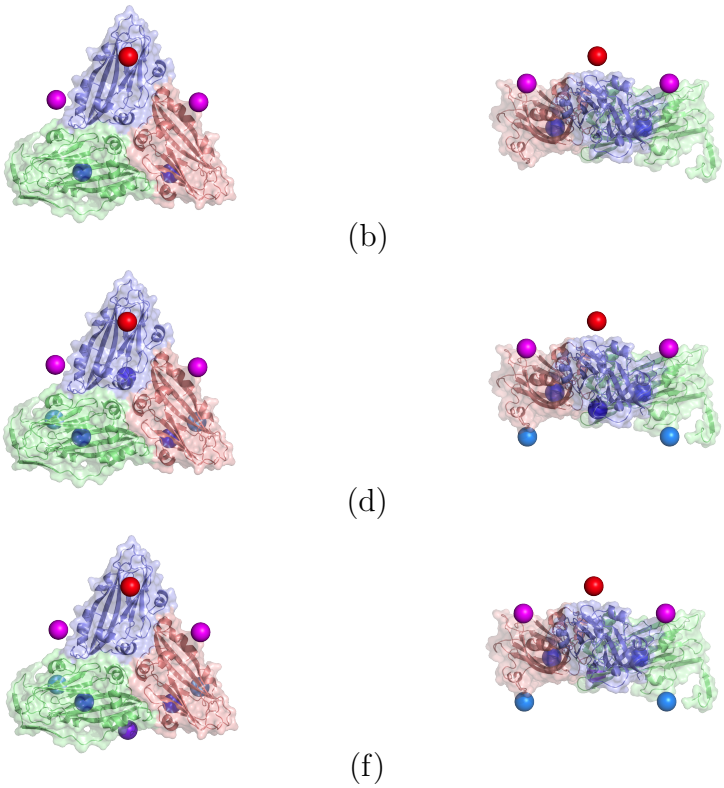


Figure 4.41: Tobacco Necrosis Virus viewed down a 5-fold axis with point array (a) 3 (b) 4 (c) 5 (d) 6 (e) 12 (f) 13 (g) 14 and (h) 20. Since all of these point arrays are possible combinations having pure point array 1 as an exterior overlapping with the capsid, each of these corresponds to a permissible genome organisation.



(a) (b)

Figure 4.42: The points in common to the best-fit point arrays for Tobacco Necrosis Virus viewed down (a) a 5-fold and (b) a 3-fold axis.



(a) (b)

(c) (d)

(e) (f)

Figure 4.43: A trimer of Tobacco Necrosis Virus with point array 1 from (a) the top and (b) the side, point array 3 from (c) the top and (d) the side, and point array 20 from (e) the top and (f) the side.

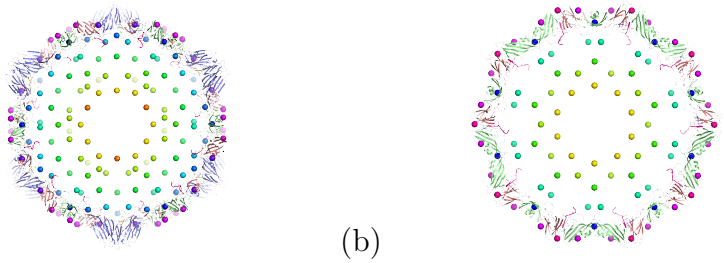
4.4.3 Desmodium Yellow Mottle Tymovirus

Desmodium Yellow Mottle Tymovirus (DYMV) is a $T = 3$ virus with 180 chemically identical subunits arranged into pentamers and hexamers which bulge out around their respective symmetry axes. The `pdb`-file is available with PDB-ID `1ddl` [56] from VIPER, and was determined to 2.7Å resolution.

Table 4.13 shows the results of applying the first iteration combination point arrays. Point arrays 291 and 300 are discarded, due to their (extremely) low prevalence (1 and 3 out of a possible 101 respectively), leaving 217 as the best-fit point array, albeit by a small margin (it scores 1.852 with the next-best arrays scoring 1.897). Note that the exteriors (48 and 15) are complementary in the sense of Table 2.2, so there is effectively only one point array that matches the protein coat of DYMV. The first point array with a different exterior is 59, which achieves a score of 5.438, but only has a prevalence of 2, and so would be discarded. The next point array that has a different exterior and an acceptable prevalence is 1014 (50 and 51) which has a score of 9.160 (with a prevalence also of 78).

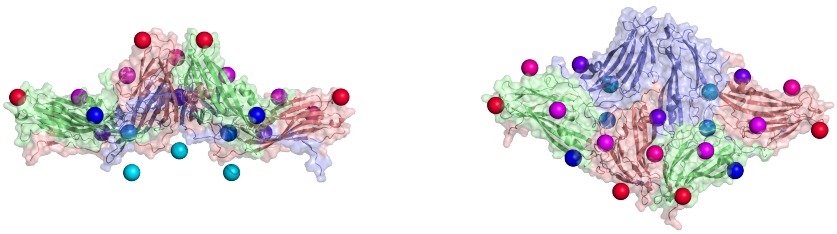
Figure 4.44 shows the best-fit point array (217) against the virus viewed down both a 3-fold and a 5-fold axis, showing how it fits the extent of the capsid; Figure 4.45 shows the same point array against two trimers (compare with Figure 4.2 which shows a trimer of Pariacoto Virus). Figure 4.46(a–d) shows the best-fit point array (217) matching a pentamer and a hexamer of DYMV individually, while (e) and (f) show the pentamer and hexamer together. Notably, the

bulk of the proteins are marked clearly from a top view, and the hexamer's extent is exceptionally well followed, something that no doubt contributes considerably to the remarkably low RMSD score of 1.246 despite 8 points of the point array matching capsid material.



(a) (b)

Figure 4.44: The crystal structure of Desmodium Yellow Mottle Tymovirus with the best-fit point array (217) overlaid, coloured by radius viewed down (a) a 3-fold axis and (b) a 5-fold axis.



(a) (b)

Figure 4.45: The crystal structure of two trimers of Desmodium Yellow Mottle Tymovirus with the best-fit point array (217) overlaid, coloured by radius from (a) the side and (b) the top.

Combo Number	Outside	Inside	RMSD	Dist. to Tower	Final Score	Combo Radius	Shunt	Number of Hits	Prevalence
291	15	16	1.183	1.376	1.814	148.582	-4.1	8	1
217	48	10	1.246	1.37	1.852	147.982	-4.7	7	78
300	15	34	1.275	1.369	1.871	147.882	-4.8	8	3
58	15	3	1.302	1.379	1.897	148.982	-3.7	5	78
83	15	4	1.302	1.379	1.897	148.982	-3.7	5	78
108	15	5	1.302	1.379	1.897	148.982	-3.7	5	78
133	15	6	1.302	1.379	1.897	148.982	-3.7	5	78
290	15	15	1.302	1.379	1.897	148.982	-3.7	5	78
292	15	17	1.302	1.379	1.897	148.982	-3.7	5	78
293	15	18	1.302	1.379	1.897	148.982	-3.7	5	78

Table 4.13: The ten lowest scoring combination point arrays for Desmodium Yellow Mottle Tymovirus.

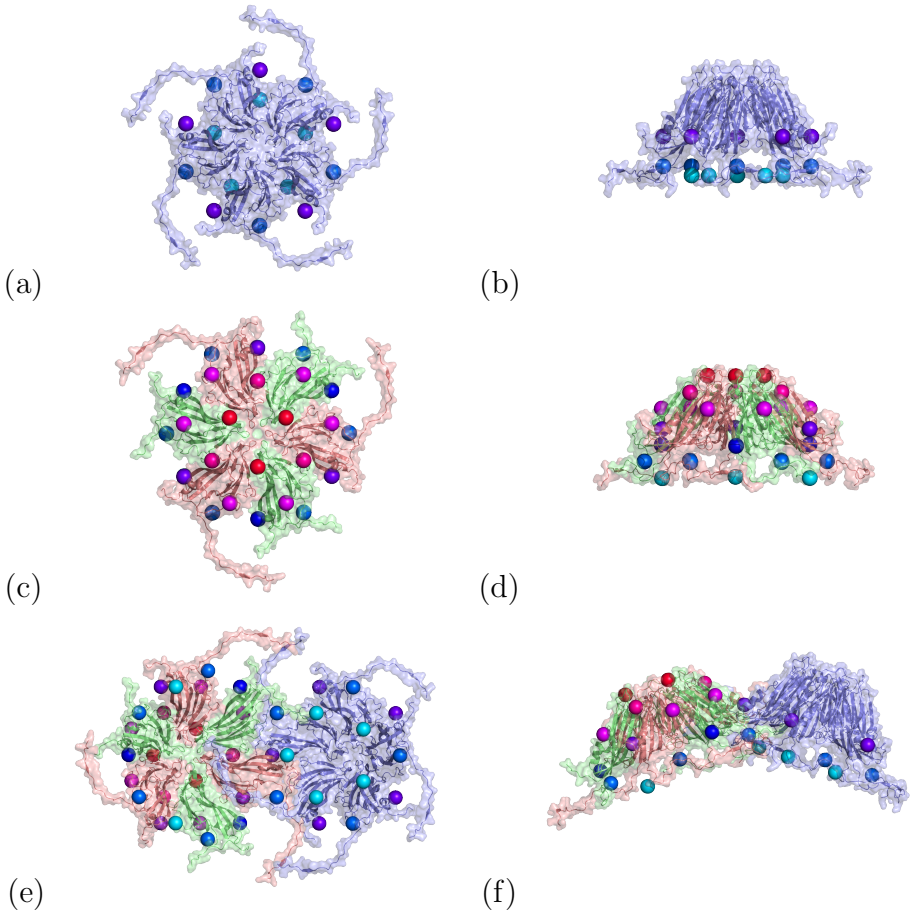


Figure 4.46: The best-fit point array 217 matching a pentamer of Desmodium Yellow Mottle Tymovirus viewed from (a) the top and (b) the side and a hexamer from (c) the top and (d) the side and matching a neighbouring pentamer and hexamer from (e) the top and (f) the side.

4.5 Conclusions

What is clear from these results, and in particular from Section 4.1, is that the symmetry of viruses is not purely tangential (as opposed to radial), to use a mathematical phrase. That is, the influence of symmetry does not only impact the layout of proteins on the surface of the capsid, but also the thickness of that capsid, and also the potential layouts of the genomic material within that capsid.

Section 1.4 of the Introduction discussed how efficient viral genomes are in terms of genetic economy, coding for the correct proteins, folding efficiently and so on, but there appears to be an even deeper connection than that: the method presented here shows that symmetry implies a correlation between the shapes and sizes of different viral components.In particular, given the dimensions of RNA are fixed by nature, a point array that matches them inside the virus leaves little room for variation of the capsid structure. There is therefore evidence of a global viral molecular scaling principle, through which the dimensions of the various viral components are related to one another, and that of the RNA, as illustrated in Figure 4.47.

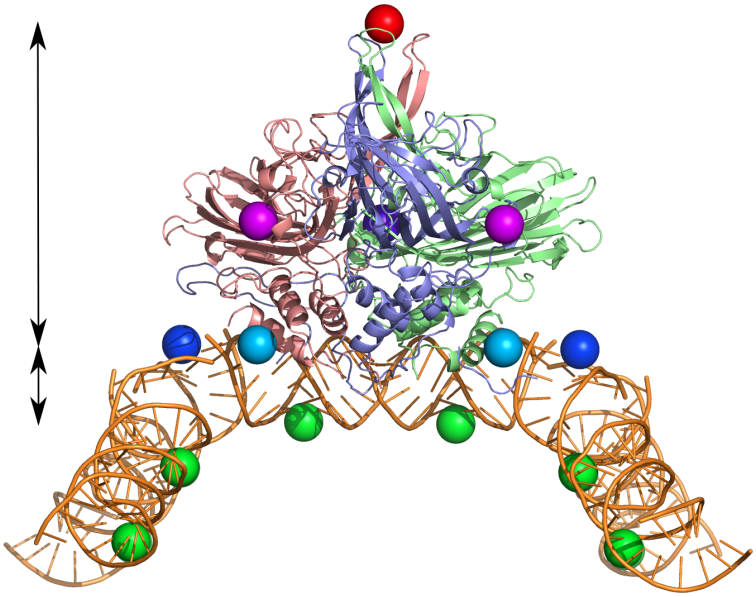


Figure 4.47: The width of RNA dictates the location and scaling of the remaining points, and hence the dimensions of the capsomeres (in this case a trimer) making up the capsid.

Chapter 5

Applications to Fullerenes

Viruses are not the only multishell structures with icosahedral symmetry: such structures also occur in chemistry. We probe here if our mathematical techniques can account for their structures as well.

5.1 Introduction to Fullerenes

In 1980 Iijima reported micrographs of "extremely small particles of less than 100Å in diameter having graphite-like structure" [34]. These carbon cage structures were found in vacuum-deposited films of carbon, and resembled polyhedra made up of 12 pentagonal and otherwise hexagonal faces. Kroto *et al.* confirmed the existence of a structure with 60 carbon atoms via graphite vaporisation [52]. They proposed the structure shown in Figure 5.1 and named this molecule "Buckminsterfullerene", although the alternatives "ballene", "spherene", "soccerene" and "carbosoccer" were also mentioned in the literature.

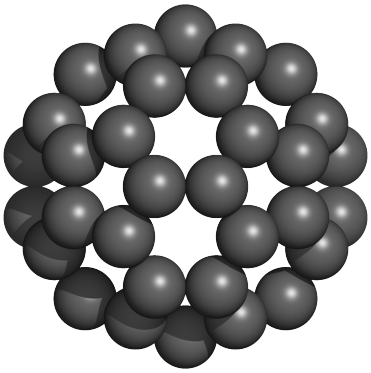


Figure 5.1: C_{60} as proposed by Kroto.

The proposed structure was confirmed by Hawkins via X-ray crystallography [30], before Ugarte [110] showed that graphitic networks can curl up under irradiation by electrons, forming nested shells of carbon exactly like those seen previously by Iijima. Kroto [51] explained that while the structures were very similar, their methods of construction could be quite different.

C_{60} and related structures were analysed by e.g. Leszczynski and Yanov [62], who investigated whether atoms similar to carbon could also form fullerenes. Various properties of carbon fullerenes, such as their polarisability [55], have been probed. The existence of C_{80} was confirmed in 1996 [31], and the possibility of more complicated surfaces has been raised [101].

Icosahedral fullerenes (classified mathematically by Kustov *et al* [54]) occur in two forms: spherical and faceted. The spherical forms are perhaps more stable than the faceted ones [111, 120], with most of the curvature being concentrated around the pentagons [72].

5.2 Point-Arrays as Models of Fullerenes

A check of the exteriors of the existing point arrays shows that none of them are consistent with the characteristic layout of fullerenes in terms of hexagonal and pentagonal rings when treating each point of the array as a carbon atom. Therefore, different arrays must be calculated. Exactly the same procedure as introduced in Chapter 2 (page 37) is followed, but with a base shape of C_{60} rather than an icosahedron, dodecahedron or icosidodecahedron. The vertices used are given in Tables A.4 and A.5, starting on page 170.

The standard procedure results in 49 point arrays: 21 from a translation along a 2-fold axis; 16 from a 3-fold axis and 12 from a 5-fold axis. As an aside, the number of point arrays generated from a base shape with 60 vertices follows the general trend of increasing number of point arrays found as the number of vertices in the base shape increase (with 13, 17 and 25 point arrays resulting from base shapes with 12, 20 and 30 vertices). The allowed translations are given in Tables A.9 and A.10, starting on page 174.

5.2.1 The C_{60} Series

We know, thanks to Ugarte [111], that there is a carbon onion that is realised as an ensemble of shells composed of C_{60} inside C_{240} inside C_{540} (Figure 5.2 shows C_{60} alongside C_{240} and C_{540} for comparison). Therefore, we start by constructing a model of C_{240} based on C_{60} as a base shape. When the exteriors of the point arrays are calculated with the procedure of Section 2.6 (page 54), there are 13 point arrays

whose exteriors have exactly 240 points. Precisely one of these is consistent with *three-connectedness*; that is, each point has precisely three neighbours at approximately the same distance. There is no other point array with the three-connectedness property among all 49 point arrays. It is therefore the only candidate to model the geometry of the C_{240} molecule. For reference, the translation that provides the model of C_{240} is along a 5-fold axis with a multiplier of 3 (see point array 45 in Table A.10 on page 175).

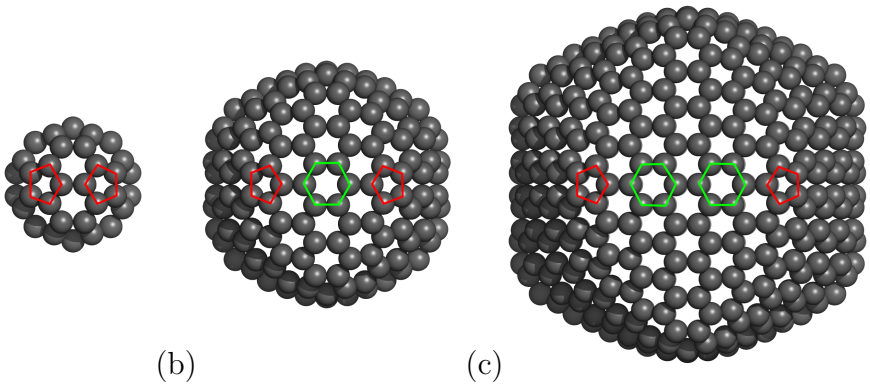
For computational purposes, we will require the following definition that allows us to check the three-connectedness property for our fullerene models.

Definition 1. *Two numbers a and b are defined as* approximately the same *if*

$$\frac{|a-b|}{|a|+|b|} < 0.01$$

Note that this definition is scale-invariant, symmetric in a and b and applies equally to vectors.

Figure 5.2 shows how the structure of C_{240} differs from that of C_{60} by an extra hexagon (shown in green) between the two pentagons (red). This leads to the question as to whether repeating the copy-and-translate process using C_{240} as a start configuration leads to a further shell of this type. If a further iteration step is carried out (i.e. another translation along a 5-fold axis with a multiplier of 3), the structure of C_{540} (shown in Figure 5.2(c)) is obtained. It has one more extra hexagon between the pentagons as demonstrated in the figure.



(a) (b) (c)

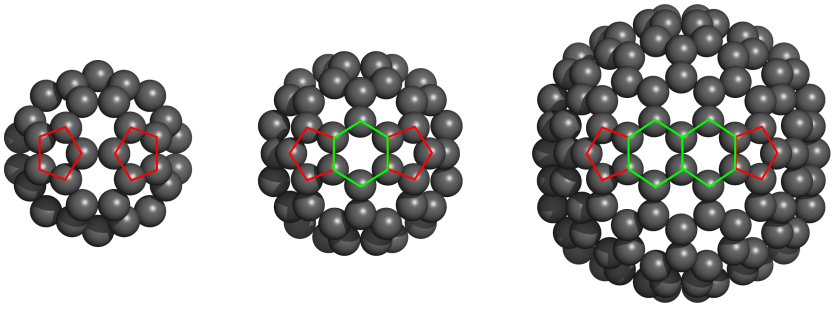
Figure 5.2: The carbon cages in a carbon onion: (a) C_{60}, (b)C_{240} and (c) C_{540}. Note that all three have pentagons (red) oriented vertex-to-vertex with no, one and two hexagons (green) between them respectively.

Indeed, this translation can be repeated, producing models for C_{960}, C_{1500}, C_{2160} and C_{2940} as well. This shows that our procedure simultaneously models different shells of a carbon onion.

5.2.2 The C_{80} Series

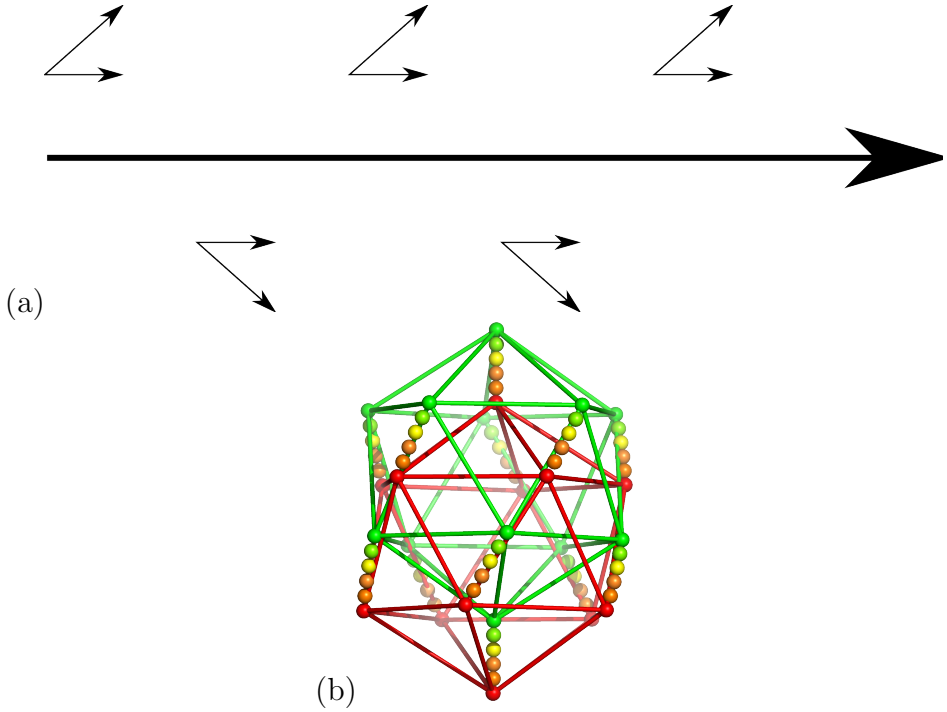
After the confirmation of the existence of C_{80} by Hennrich [31], Furche [20] analysed the different forms available for it, concluding that the icosahedral model was the least stable. As shown in Figure 5.3, the pentagons of C_{80} are oriented differently to those in C_{60}: the C_{60} pentagons are oriented "point-to-point" and those in C_{80} are "edge-to-edge". The affine extensions determined earlier are therefore not able to describe them. (Note that the 49 point arrays generated from C_{60} do not include one that has precisely 80 points in its exterior — the smallest has 150 points.)

In order to account for this phenomenon, a *screw-translation* is nec-



(a) (b) (c)

Figure 5.3: (a) C_{60} has pentagons that are oriented vertex-to-vertex whereas (b) C_{80} and (c) C_{180} have pentagons that are oriented edge-to-edge.



(a)

(b)

Figure 5.4: (a) A glide-reflection. The shape is translated and reflected in the line of translation at each step. (b) A screw-translation of an icosahedron along a 5-fold axis. As the icosahedron is moved along the axis of translation it is rotated around that axis.

essary. This is akin to a *glide-reflection* in two dimensions (see Figure 5.4). A glide-reflection of a footprint produces a pattern akin to someone walking. For each iteration, the repeated motif (the footprint in this case) is reflected through the line (the direction of walking) and translated by one stride-length. A screw-translation is a similar concept in three dimensions. That is, when translating the start shape along an n-fold axis, it is rotated by π/n radians around that translation direction (the direction of rotation is irrelevant, as the end result is identical). The criteria for finding a valid translation remain the same. As before, one of the translated points must lie on one of the symmetry planes.

This procedure generates what will be referred to as the "twisted translations", of which there are 130 (see Tables A.16 to A.20, page 181). Once again, the exteriors have been found and checked for three-connectedness. Precisely three of the twisted point arrays are three-connected: numbers 66, 106 and 113, and these have 120, 80 and 120 points, respectively. In particular, the translation constructing C_{80} is along a 5-fold axis (by length $-1/5 + 2\tau/5$), meaning that the pentagons should line up correctly. Indeed, the structure formed is made up of 12 pentagons and 30 hexagons, each hexagon located on a 2-fold axis (see Figure 5.3(b)). This figure shows that C_{80} contains an additional hexagon between the two pentagons of C_{60}. However, the rotation required to do this precludes the same translation being used to continue the series. Instead, C_{80} must be used as a starting point, and then "straight" (i.e. non-twisted) translations can be generated using C_{80} as a start configuration.

There are 76 standard translations of C_{80}, of which two are three-connected: numbers 64 (along a 5-fold by length $7/5 + \tau/5$) and 69 (along a 5-fold by length $12/5+\tau/5$). They have 180 and 240 points in their exteriors, respectively. The former corresponds to the expected layout of pentagons and hexagons, as depicted in Figure 5.3(c), and, in particular, has two hexagons separating the adjacent pentagons as illustrated in the figure. This translation can be iterated to produce layouts for larger fullerenes, starting with C_{320} and C_{500}.

5.2.3 Other Possibilities

Looking at the other three-connected point array exteriors generated, there are still three to consider; two with 120 points, and one with 240 points, shown in Figure 5.5.

The first alternative structure for C_{120} (as in Figure 5.5(a)) is not very likely to be realised experimentally as the angles required for carbon to create the triangles in that structure are rather acute (and cyclopropane is rather reactive [1]). The second structure (shown in Figure 5.5(b)), is somewhat more feasible, although it does include squares, which may also be rather reactive. It is the only proposed structure for C_{120} to date that is not the "dumbbell" shape of two C_{60} molecules bonded by a shared face [50] in Figure 5.6.

Finally, we revisit all point arrays, twisted and non-twisted with the test for three-connectedness. This reveals more structures that could potentially be realised as fullerenes: arrays 22 and 26 of the basic 55 point arrays (see Table A.7) give a structure[1] for C_{200} (see Figure 5.7). Generating twisted arrays with the "standard" choice of base shapes (namely, the icosahedron, the dodecahedron and the icosidodecahedron) gives one more array (33 — twist translating a dodecahedron along a 2-fold axis by $-4+3\tau$) that corresponds to the buckyball structure, C_{60}. This exhausts all possibilities obtainable with our formalism.

[1] Just one structure: those two point arrays are identical, even if generated in different ways.

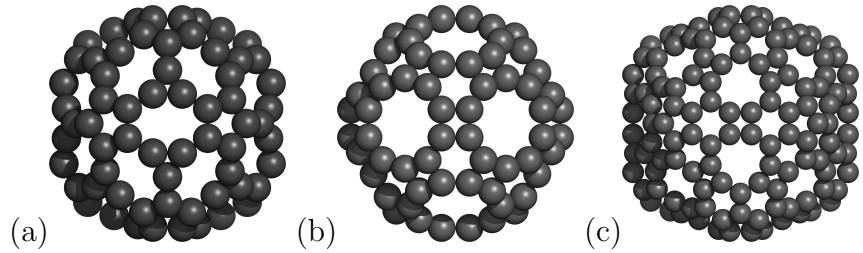


Figure 5.5: (a) C_{120} displaying triangles, pentagons and irregular octagons; (b) C_{120} as a truncated icosidodecahedron, displaying squares, hexagons and decagons; (c) C_{240} displaying squares, pentagons, hexagons and irregular nonagons.

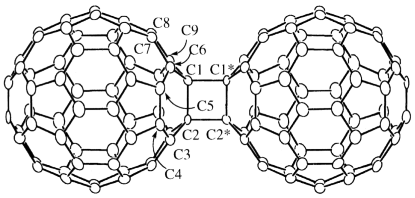


Figure 5.6: C_{120} as a C_{60} dimer as found by X-ray crystallography [50].

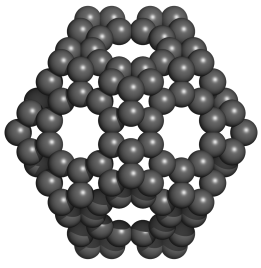


Figure 5.7: A model for a carbon cage structure formed from 200 atoms displaying pentagons, deformed hexagons and decagons.

5.3 Summary

Kustov *et al.* [54] show that from a group theoretical point of view, C_n with $n = 60z$ and $n = 60z + 20$, for $z \in \mathbb{N}$, are "allowable" fullerene structures. Our method has resulted in models for a number of these fullerenes, including two infinite series, and potential alternative structures for C_{120} and C_{240}. The structures found, and how they are related via translations and twist-translations, are shown in diagrammatic form (using `dot` [21]) in Figure 5.8. The links are labelled by the numbers of the translations used to map the corresponding structural blueprints onto each other (the numbers relate to the tables in Appendix A), and a prefix of T refers to a twisted translation.

It is true, though, that, with the exception of those structures shown in Figures 5.5 and 5.7, the structures proposed fit directly into a triangulation scheme differing from that of Caspar and Klug by only the exact locations of the entities involved (carbon atoms as opposed to proteins). It is still interesting to note that a triangulation theory (namely quasi-equivalence) and this affine extension theory agree completely in this case, showing one as an extension of the other.

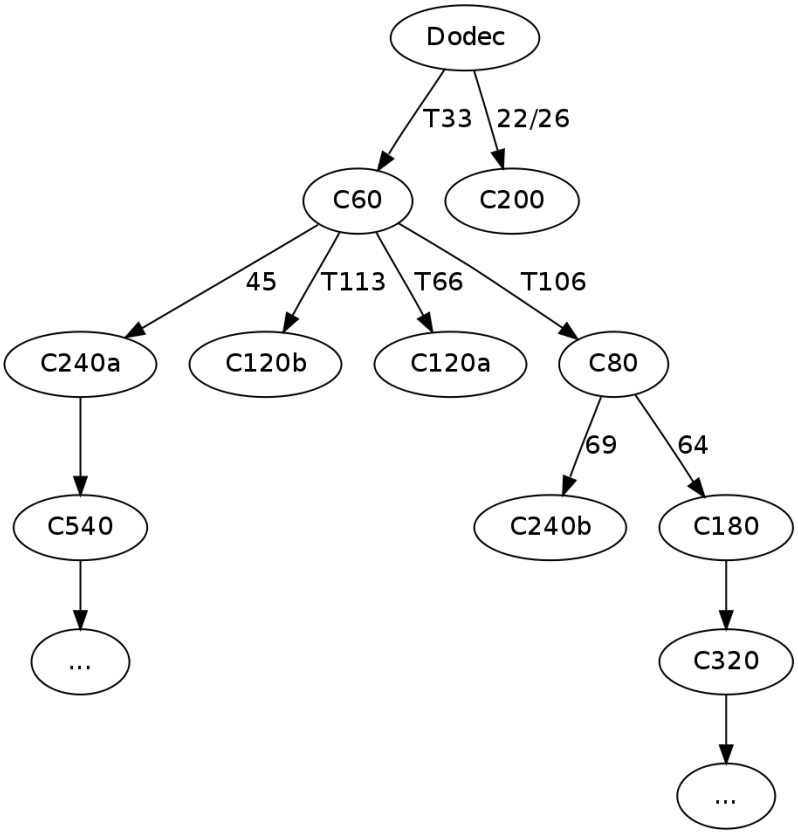


Figure 5.8: A graph displaying the potential fullerene structures created and how they relate via translations and twisted translations.

Chapter 6

Conclusions

In this chapter, the conclusions that can be drawn from the results in the previous chapters are analysed, with particular attention to the results from Chapter 4.

6.1 Predictive Capabilities

The three aspects of the algorithm's predictive capabilities discussed here are the predictions on genomic layouts, the possibilities of multiple genomic layouts for certain viruses (referred to as polymorphic interiors) and providing information for the study of viral transitions.

6.1.1 Predicting Genomic Layout

What would seem to be the outstanding result of this new approach is the correlation between the structure of the protein capsid and the structure of the genomic material (RNA in most of the examples) packaged within that capsid. Previous approaches to viral layout ([9, 107])

have concentrated on the capsid, for which structural data is available at higher resolutions than for genome organisation.This 3D approach, though, links the structure of the capsid with the organisation of the genomic material while providing information on the capsid thickness at various locations.

This correspondence between capsid and genome organisation is particularly noticeable in Sections 4.1.1 (Pariacoto Virus), 4.1.2 (Bacteriophage MS2) and 4.4.1 (Hepatitis B). The Pariacoto Virus result is particularly stunning as not only do the extra internal points match remarkably well to the cryo-EM data (note that every internal point of the array matches to some feature of the cryo-EM map), but there is also modelled X-ray crystallography data for the RNA, and a subset of the internal points are extremely close to the molecular surface of this (recall that the X-ray data of the RNA was not made available to the algorithm). The results for Bacteriophage MS2 and Hepatitis B display a remarkable match to the relevant cryo-EM data, and in several cases, *every single* internal point matches to some cryo-EM feature. It is worth pointing out that while Bacteriophage MS2 and Pariacoto Virus are both $T = 3$ viruses, Hepatitis B is a $T = 4$ virus.

Next, Tomato Bushy Stunt Virus (TBSV - Section 4.1.4) displays extra information about the tertiary genomic structure under neutron scattering (see Figure 4.22), and the point arrays that were chosen by the best-fit algorithm to match to the capsid are all in agreement with this further information (see Figures 4.23 and 4.24). Again, every point in the array (indeed, in each of the 4 best-fitting point arrays) corresponds to material boundaries in the structural information from

the neutron-scattering data.

TBSV exhibits a two-domain [84] protein structure, where the proteins making up the capsid fold into two distinct domains. This is not explicitly delineated by the point arrays, as they do not contain such information, but Figures 4.20 and 4.21, particularly the bottom rows of the combined point arrays, do give some indication that this structure is reflected in the point array.

Finally, STMV does not seem to exhibit an RNA cage, but the `pdb`-file does include RNA fragments where it attaches to the capsid. These fragments are bracketed by the best-fit point array, and even such a small virus can be modelled in this framework. Larger viruses exhibit a number of issues that are discussed later in Section 6.3.4.

There are, however, flaws in the theory. As Section 4.1.1 discusses, while the match to the modelled genomic material looks impressive, it is not the best match of all the point arrays, nor even only those that matched to the capsid proteins. Indeed, of the 196 point arrays matching to the capsid, 563 is 19th by way of scoring to the genome. There is not even, unfortunately, any significant correlation (linear or nonlinear) between the capsid and genome scores as could be hoped were there to be a direct causal link between the two (this does not, though, rule out such a link). Furthermore, of the 117 point arrays that match reasonably to the genome (that is, both have points near the genomic material but not within an atom), the mean score is 1.144 with standard deviation 0.455, meaning the score for array 563 is 1.600 standard deviations below the mean and that 5.48% of scores are at least this good, assuming a normal distribution of scores. However,

of the 18 better-fitting point arrays, 15 of them would fail a stability test such as the "Prevalence" statistic — assuming that points that deep into the capsid move significantly as the target point moves — as illustrated in Figure 4.8. There is certainly a prospect for a different measure of stability, and perhaps a different best-fit algorithm would produce a more clear-cut result.

6.1.2 Polymorphic Interiors

Most icosahedrally–symmetric viruses do not have modelled genomic material in their `pdb`-files, suggesting that X-ray crystallography does not provide sufficient resolution to reliably locate the genome. Cryo-EM data often demonstrate the presence of such genomic material, albeit to a lower resolution than the more icosahedrally regular capsid proteins, but occasionally there is very little information near the centre of the capsid. Two possibilities for this effect are firstly that the genome may not be organised in a symmetric manner (or with much organisation at all), and secondly that there are multiple different arrangements of the genomic material that each look different under icosahedral symmetry and so the averaging process "washes out" the information. In some cases, extra information can be found by *not* averaging so much: [105] applied only 5-fold averaging to investigate how the genomic material of Bacteriophage MS2 lies when the virus is bound to its receptor.

The ability of our approach to explain the second phenomenon is discussed in Section 2.4, Figure 2.7 in particular suggesting that for certain external point arrays, a number of different internal ones

are equally possible, implying that the structural constraints permit a number of different genome organisations for these viruses. Sections 4.4.2 and 4.4.3 show this principle in action: for Tobacco Necrosis Virus, all of the 10 best-scoring point arrays share the same exterior (point array 1), despite the first 4 scoring (very slightly) better than the remainders (13.46 to 13.83) and are all scaled to the same radius (160Å), despite having different interior point arrays; Desmodium Yellow Mottle Tymovirus demonstrates similar behaviour (see Section 4.1.3) – note that point arrays 15 and 48 have identical exteriors in the sense of Table 2.2 – despite the different prevalences and even scalings of the ten best-fit point arrays.

Also, despite Bacteriophage GA (Section 4.1.3) having a distinct best-fit point array (152, which is composed of 7 and 8 and is the same as the best-fit point array for Bacteriophage MS2 (Section 4.1.2)) with a score of 4.76, it has an ensemble of follow-up point arrays with scores of 5.49, each with exterior point array 42 (having a notably high radius as shown in Figure 2.7). This could suggest that the virus prefers one particular arrangement of genomic material, but several other organisations are possible and occur with similar probabilities (albeit lower than that for the arrangement corresponding to point array 152). This could be tested in principle with cryo-EM tomography when that field advances to achieve sufficiently high resolutions.

Finally, Hepatitis B (Section 4.4.1) is a somewhat special case, having a (relatively) ordered genome at two points in the maturation process and matching only one point array well. On the face of it, this would appear to be a further flaw with the theory. However,

the best-fit point array matches both of the tested genome structures well, albeit matching the RNA shell better; this is no doubt partially due to the fact that the RNA cage appears to be more ordered and hence more visible. Furthermore, while the outermost of the predictive points mark the turns of DNA visible in Figure 4.40(a), when more density is brought in, the first patches fit exactly with each of the other predictive points as shown in Figure 4.40(b).

This indicates the algorithm demonstrating that it takes into account (or at least, does not contradict) the known ability of a virus to have a genome capable of folding in multiple ways (in this case, as it transitions between RNA and DNA) despite one and only one point array being picked out as the best-fitting. This adds weight to the supposition that not all sparsely populated cryo-EM models are due to disorganised genomic material, but could well be because of (potentially more radically) different tertiary structures of the genome.

6.1.3 Swelling Transformations

Some viruses undergo various structural transformations during their life cycle, for example as part of a maturation process, such as in the case of Hong Kong 97 [60]. Here, Cowpea Chlorotic Mottle Virus (CCMV) was investigated in Section 4.2 by subjecting both the beginning state (Section 4.2.1) and the proposed end state (Section 4.2.2) to the algorithm described in Chapter 3. The results of the algorithm on the initial state (Figures 4.27 and 4.28) and the final state (Figures 4.30 and 4.31) (this author's contribution) have formed the basis for an analysis of the likely transitions [35].

6.2 Comparison with Random Points

With each point array matching to a feature on the exterior of the viral capsid it is feasible to calculate the probability of a match of a given accuracy to a marked point that a random point distributed over a sphere of a given radius might achieve. That is, if we imagine the exterior of the virus as a sphere and pin targets on it with a radius equal to the amount the best-fit point array was away from the target point of the algorithm (that is, S_1 from equation (3.3)), we can calculate the probability one of a number of darts thrown at random hits one of those targets; that is, how likely it is to get a result at least as good as the one found by the best-fit algorithm. The ratio of the sum of the areas of these targets and the area of the encasing sphere gives the probability each dart hits, x, but what is required is the probability that at least one dart hits out of several. The probability all the darts miss is $(1-x)^n$ where n darts are thrown, and so the probability at least one hits is $1-(1-x)^n$. Each of the t targets has an area of πe^2 where e is the radius of the target, and the large sphere has an area of $4\pi r^2$ where its radius is r. This is illustrated in Figure 6.1. We then have $x = te^2/4r^2$.

The formula for the probability of a random point achieving a result at least as close to an outermost feature as the best-fit algorithm is then

$$P = 1 - \left(1 - \frac{te^2}{4r^2}\right)^n \tag{6.1}$$

where r is the radius of the tower midpoint (taken from the "Combo

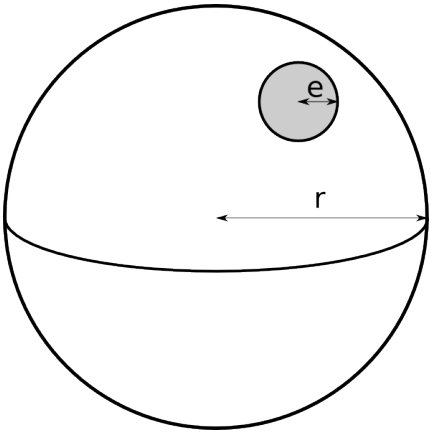


Figure 6.1: One of the t targets of radius e for a random point array compared with the sphere of radius r encasing the virus undergoing analysis.

Radius" columns from results tables in Chapter 4), e is the error "desired" (the "Dist. to Tower" columns from Chapter 4 — S_1 from equation (3.3)), n is the number of distinct outsides (fixed for here at 29 from Section 2.4) and t is the number of tower points available as targets, which has to be considered on a virus-to-virus basis. Much of the time, there are as many distinct outermost features as there are proteins, but notable exceptions are Pariacoto Virus (where there is one tower for each trimer — recall Figure 4.2) and Hepatitis B, where each dimer forms a distinct tower (most obvious in Figure 4.36).

A very large caveat applies, though: this is purely measuring the outermost points; it is *not* a probability of finding a point array to match the entire virus at least as well as the best-fit.

Table 6.1 summarises the results of this applied to the 11 viruses studied here. The 99.9% chance of a match as good or better to the outermost features of TBSV is, however, not as bad as it appears.

Virus	Error (S_1)	Radius	Probability
Bacteriophage GA	4.236	149.068	65.8%
Bacteriophage MS2	5.826	148.797	87.4%
CCMV	6.972	136.262	97.4%
CCMV (Swollen)	3.925	162.577	53.7%
DYMV	1.370	147.982	10.6%
Hepatitis B	1.619	178.03	7.0%
Pariacoto	0.892	173.894	1.1%
Simian 40	2.390	243.956	22.2%
STMV	2.781	93.065	32.4%
TBSV	5.922	174.139	53.4%
TNV	12.638	160.326	99.9%

Table 6.1: The 11 viruses studied, and the probability of matching their outermost features by chance.

There are two components to an overall score, S, being S_1 and S_2 (the match to outermost features analysed here and the RMSD to protein surfaces), and one of these being high can be, in some degree, compensated for by a low score in the other. For TNV, the high score S_1 (12.64) is compensated for by the (comparatively low for that virus) S_2 score of 4.63 — the shortest distance to a target point for TNV is 7.5Å, but the associated RMSD score is 17.73!

On the other hand, the fact that the chance of getting a hit to the towers of Pariacoto Virus as good as the best-fit point array is as low as 1.1% is even more impressive given that this does not take into account any of the further remarkable matches with the capsid.

6.3 Assessment of the Method

In discussions about the method explained here, a few potential criticisms have surfaced, most of which have been underpinned by misunderstandings about the library. This section aims to enlighten the reader as to some of these potential pitfalls.

6.3.1 Icosahedral Symmetry can Manifest in Many Ways

The point arrays are based on icosahedral symmetry, as are viruses (which is why they match). The affine extension generating the point arrays also relies on the symmetry axes of the icosahedron, and so it has been posited that the algorithm is bound to succeed as it matches an icosahedral object to another icosahedral object.

Icosahedral symmetry can be instantiated in many ways (Chapters 2 and 5 contain a plethora of examples) and the results examined here demonstrate that viruses do not follow just icosahedral symmetry, but one of a finite set of restricted instantiations of icosahedral symmetry at different radial levels. Thus, while it is not surprising to see icosahedral symmetry at various radial levels, the way it is organised (e.g. as an icosahedron, dodecahedron, icosidodecahedron or something with more vertices, such as C_{60}) and the exact nature of the radial levels in the multishell structure are non-trivial predictions of the theory.

6.3.2 The Point-Arrays Fill Space

Another criticism has been that given there are so many point arrays, it is trivial to pick and choose certain points so that they fit any icosahedral object. As it happens, if all the "pure" point arrays (the basic 55) are scaled to the same exterior radius, there are indeed some internal points close to any particular radius chosen (however, choosing the radius is not the same as choosing the position, as the location on that sphere is not changeable). However, one can not simply pick and choose subsets of those points freely — they come in packages (55, naturally) and each package must be taken as a whole, or not at all. As explained previously, in particular for Pariacoto Virus, once the gauge point has been picked and scaled correctly, the rest of the point array follows automatically, with a little lee-way to pick the interior of the combination point array, but again, these points come as a "package deal".

6.3.3 Other Matching Algorithms

This algorithm is not the only possible way of fitting the point arrays to viruses; for example, the same "matching to gauge points" method could be used to select the exterior point array, and then the best fitting of the internal point arrays could be chosen, with a different consideration for stability. This may work well for viruses with particularly solid and prominent outermost features such as Pariacoto, but perhaps less so for viruses such as Tobacco Necrosis or Dengue [53]. Again, though, it is impossible to prove that any one algorithm

is the best (or what "best" means in this setting) and what is presented here is merely one possible fitting algorithm, among other as yet untried algorithms, that produces results that clearly demonstrate the existence of a more general symmetry to viruses than previously thought.

6.3.4 Larger Viruses

The viruses studied here have been mostly $T = 3$ viruses, with one smaller and two larger, albeit only $T = 4$ and $T = 7$. Many larger viruses have been studied so as to produce a `pdb`-file, such as the $T = 13$ Bluetongue virus [29] with an outer radius of 353Å or the very large $pT = 169$ PBCV-1 virus [74] with an outer radius of 929Å, but are not studied here. The primary reason for this is computing power — the memory requirements increase as the square of the radius — although the definition of the "outside" of a virus used here becomes less useful at this radius, as the viruses under consideration become more icosahedron-like and less round. However, the higher iteration point arrays also become more faceted, so there is hope there. Lastly, the work on clustering outermost features needs to be updated, as for the smaller viruses the algorithm works well on small areas of protein; with the larger viruses, the outermost features consist of entire proteins, so the current method offers a number of target points for each protein.

6.4 Further Work

There is more work that can be done on the algorithm. As mentioned in Section 6.3.4, the algorithm can be applied to larger viruses (with appropriate computing power) and the necessary tweaks worked out and applied — it is the hope of the author that there could be a scale-invariant algorithm applicable to any (within reason) size of virus, smoothly transitioning from smaller, rounder, $T = 1, 3$ or 4 viruses to larger, more faceted, ones.

Once a truly universal algorithm is worked out, this work could form the basis of a new classification system for viruses, if, say, the algorithm were applied to any appropriate deposition at the Protein DataBank or anything mirrored on VIPER. Given that the algorithm does not detect quasi-equivalence or the lack thereof, though, T number is still useful (as would be a viral tiling, if appropriate).

However, as discussed briefly in Section 6.1.1, there is room for a stability analysis that is not just the Prevalence statistic; the aborted approach with MSMS [123] is unlikely to apply, as it is unsuitable for anything not molecular in size. There is the option for using the B factors in the `PDB` files, if present, to apply Gaussian noise to the atomic coordinates, or perhaps the approach could be to calculate what proportion of space near each point is occupied by atoms, although this is difficult to find analytically.

Lastly, one of the predictions of further RNA within Pariacoto virus (Section 4.1.1) was based on the distance between two adjacent points being exactly that of a turn of ssRNA. It is certainly feasible (on

the lower iteration point arrays at least; higher iterations may require more computing power) that once an array has been fitted to a virus, the distances between adjacent pairs of points can be calculated and any that match neatly to an integer multiple of turns of RNA or DNA could be identified to the user for further investigation. This approach would aid greatly in the predictive power of the algorithm and entire paradigm.

6.5 Uniting Viruses and Fullerenes

This thesis has presented a unified framework for generating geometric representatives of affine-extended icosahedral symmetry and a best-fit algorithm to apply these concepts to icosahedral structures in virology and carbon chemistry. It has shown how the same mathematical principles apply to fullerenes and simultaneously underlie the structure and size of multiple viral components, in a prescriptive rather than descriptive way, where previous theories of viruses have only applied in a surface manner to the capsid. Moreover, this approach, using only icosahedral symmetry, is blind to whether the virus under consideration follows basic quasi-equivalence theory (that is, using a triangulation of the icosahedron's surface), or the extension of viral tiling theory (using other shapes of tiles). It also proffers a potential explanation as to why some viral particles reveal little to nothing of their interiors under (for example) cryo-EM, over and above positing a randomly organised genome. It is the hope of the author that this new understanding of the deeper symmetry of viruses can be used in biological and medical research to help inspire new targeted methods of preventing harmful viruses.

Appendix A

Point Arrays

This Appendix contains supplementary information about the specifics of the point-arrays and their construction. Firstly there are the tables giving the coordinates of the three basic shapes — icosahedron, dodecahedron and icosidodecahedron — and also C_{60}; these are the start configurations as in [35] and all translation lengths given are indicated with respect to these. Then there are tables giving the labels of each start configuration, translation direction and amount for each of those three start configurations, followed by tables giving information about the standard point-arrays of C_{60}. Finally, there are tables for each twisted point-array with the four possible start configurations.

x coordinate	y coordinate	z coordinate
1	0	τ
-1	0	τ
1	0	$-\tau$
-1	0	$-\tau$
0	τ	1
0	$-\tau$	1
0	τ	-1
0	$-\tau$	-1
τ	1	0
$-\tau$	1	0
τ	-1	0
$-\tau$	-1	0

Table A.1: The vertices of the icosahedron.

x coordinate	y coordinate	z coordinate
1	1	1
-1	1	1
1	-1	1
-1	-1	1
1	1	-1
-1	1	-1
1	-1	-1
-1	-1	-1
0	$1-\tau$	τ
0	$-1+\tau$	τ
0	$1-\tau$	$-\tau$
0	$-1+\tau$	$-\tau$
$1-\tau$	τ	0
$-1+\tau$	τ	0
$1-\tau$	$-\tau$	0
$-1+\tau$	$-\tau$	0
τ	0	$1-\tau$
$-\tau$	0	$1-\tau$
τ	0	$-1+\tau$
$-\tau$	0	$-1+\tau$

Table A.2: The vertices of the dodecahedron.

x coordinate	y coordinate	z coordinate
τ	0	0
$-\tau$	0	0
0	τ	0
0	$-\tau$	0
0	0	τ
0	0	$-\tau$
$1/2$	$\tau/2$	$1/2+\tau/2$
$-1/2$	$\tau/2$	$1/2+\tau/2$
$1/2$	$-\tau/2$	$1/2+\tau/2$
$-1/2$	$-\tau/2$	$1/2+\tau/2$
$1/2$	$\tau/2$	$-1/2-\tau/2$
$-1/2$	$\tau/2$	$-1/2-\tau/2$
$1/2$	$-\tau/2$	$-1/2-\tau/2$
$-1/2$	$-\tau/2$	$-1/2-\tau/2$
$\tau/2$	$1/2+\tau/2$	$1/2$
$-\tau/2$	$1/2+\tau/2$	$1/2$
$\tau/2$	$-1/2-\tau/2$	$1/2$
$-\tau/2$	$-1/2-\tau/2$	$1/2$
$\tau/2$	$1/2+\tau/2$	$-1/2$
$-\tau/2$	$1/2+\tau/2$	$-1/2$
$\tau/2$	$-1/2-\tau/2$	$-1/2$
$-\tau/2$	$-1/2-\tau/2$	$-1/2$
$1/2+\tau/2$	$1/2$	$\tau/2$
$-1/2-\tau/2$	$1/2$	$\tau/2$
$1/2+\tau/2$	$-1/2$	$\tau/2$
$-1/2-\tau/2$	$-1/2$	$\tau/2$
$1/2+\tau/2$	$1/2$	$-\tau/2$
$-1/2-\tau/2$	$1/2$	$-\tau/2$
$1/2+\tau/2$	$-1/2$	$-\tau/2$
$-1/2-\tau/2$	$-1/2$	$-\tau/2$

Table A.3: The vertices of the icosidodecahedron.

x coordinate	y coordinate	z coordinate
1	0	3τ
-1	0	3τ
1	0	-3τ
-1	0	-3τ
0	3τ	1
0	-3τ	1
0	3τ	-1
0	-3τ	-1
3τ	1	0
-3τ	1	0
3τ	-1	0
-3τ	-1	0
2	τ	$1+2\tau$
-2	τ	$1+2\tau$
2	$-\tau$	$1+2\tau$
-2	$-\tau$	$1+2\tau$
2	τ	$-1-2\tau$
-2	τ	$-1-2\tau$
2	$-\tau$	$-1-2\tau$
-2	$-\tau$	$-1-2\tau$
τ	$1+2\tau$	2
$-\tau$	$1+2\tau$	2
τ	$-1-2\tau$	2
$-\tau$	$-1-2\tau$	2
τ	$1+2\tau$	-2
$-\tau$	$1+2\tau$	-2
τ	$-1-2\tau$	-2
$-\tau$	$-1-2\tau$	-2
$1+2\tau$	2	τ
$-1-2\tau$	2	τ

Table A.4: The vertices of C_{60} (A).

x coordinate	y coordinate	z coordinate
$1+2\tau$	-2	τ
$-1-2\tau$	-2	τ
$1+2\tau$	2	$-\tau$
$-1-2\tau$	2	$-\tau$
$1+2\tau$	-2	$-\tau$
$-1-2\tau$	-2	$-\tau$
1	2τ	$2+\tau$
-1	2τ	$2+\tau$
1	-2τ	$2+\tau$
-1	-2τ	$2+\tau$
1	2τ	$-2-\tau$
-1	2τ	$-2-\tau$
1	-2τ	$-2-\tau$
-1	-2τ	$-2-\tau$
2τ	$2+\tau$	1
-2τ	$2+\tau$	1
2τ	$-2-\tau$	1
-2τ	$-2-\tau$	1
2τ	$2+\tau$	-1
-2τ	$2+\tau$	-1
2τ	$-2-\tau$	-1
-2τ	$-2-\tau$	-1
$2+\tau$	1	2τ
$-2-\tau$	1	2τ
$2+\tau$	-1	2τ
$-2-\tau$	-1	2τ
$2+\tau$	1	-2τ
$-2-\tau$	1	-2τ
$2+\tau$	-1	-2τ
$-2-\tau$	-1	-2τ

Table A.5: The vertices of C_{60} (B).

Number	Start	Translation Direction	Translation Amount
1	Icos	IDD	$-1+\tau$
2	Icos	IDD	$4-2\tau$
3	Icos	IDD	1
4	Icos	IDD	$-2+2\tau$
5	Icos	IDD	2
6	Icos	IDD	2τ
7	Icos	Dodec	$-1+\tau$
8	Icos	Dodec	1
9	Icos	Dodec	τ
10	Icos	Dodec	$1+\tau$
11	Icos	Icos	$-1+\tau$
12	Icos	Icos	1
13	Icos	Icos	τ

Table A.6: The point-arrays with an icosahedral start.

Number	Start	Translation Direction	Translation Amount
14	Dodec	IDD	$2-\tau$
15*	Dodec	IDD	$-6+4\tau$
16*	Dodec	IDD	$-1+\tau$
17	Dodec	IDD	$4-2\tau$
18	Dodec	IDD	1
19	Dodec	IDD	$-2+2\tau$
20	Dodec	IDD	2
21*	Dodec	IDD	2τ
22*	Dodec	Dodec	$2-\tau$
23	Dodec	Dodec	$-1+\tau$
24	Dodec	Dodec	1
25	Dodec	Dodec	τ
26**	Dodec	Dodec	$1+\tau$
27	Dodec	Icos	$2-\tau$
28	Dodec	Icos	$-1+\tau$
29	Dodec	Icos	1
30	Dodec	Icos	τ

Table A.7: The point-arrays with a dodecahedral start.

Number	Start	Translation Direction	Translation Amount
31*	IDD	IDD	$-1/2 + \tau/2$
32	IDD	IDD	$2 - \tau$
33*	IDD	IDD	$1/2$
34	IDD	IDD	$-1 + \tau$
35*	IDD	IDD	$\tau/2$
36	IDD	IDD	1
37*	IDD	IDD	$-2 + 2\tau$
38	IDD	IDD	τ
39*	IDD	IDD	2
40	IDD	IDD	$1 + \tau$
41*	IDD	IDD	2τ
42*	IDD	Dodec	$-1/2 + \tau/2$
43	IDD	Dodec	$1/2$
44	IDD	Dodec	$\tau/2$
45	IDD	Dodec	1
46	IDD	Dodec	$1/2 + \tau/2$
47*	IDD	Dodec	τ
48*	IDD	Dodec	$1/2 + \tau$
49	IDD	Dodec	$1 + \tau$
50	IDD	Icos	$-1/2 + \tau/2$
51	IDD	Icos	$1/2$
52	IDD	Icos	$\tau/2$
53	IDD	Icos	1
54	IDD	Icos	$1/2 + \tau/2$
55	IDD	Icos	τ

Table A.8: The point-arrays with an icosidodecahedral start.

Number	Start	Translation Direction	Translation Amount
1	C60	IDD	$-1+\tau$
2	C60	IDD	$4-2\tau$
3	C60	IDD	1
4	C60	IDD	$-2+2\tau$
5	C60	IDD	$8-4\tau$
6	C60	IDD	2
7	C60	IDD	$-1+2\tau$
8	C60	IDD	$-4+4\tau$
9	C60	IDD	$1+\tau$
10	C60	IDD	$6-2\tau$
11	C60	IDD	3
12	C60	IDD	2τ
13	C60	IDD	$-6+6\tau$
14	C60	IDD	4
15	C60	IDD	$-2+4\tau$
16	C60	IDD	$2+2\tau$
17	C60	IDD	6
18	C60	IDD	4τ
19	C60	IDD	$4+2\tau$
20	C60	IDD	$2+4\tau$
21	C60	IDD	6τ

Table A.9: The point-arrays with a start of C_{60} translated along 2-fold axes.

Number	Start	Translation Direction	Translation Amount
22	C60	Dodec	$-1+\tau$
23	C60	Dodec	1
24	C60	Dodec	$-2+2\tau$
25	C60	Dodec	τ
26	C60	Dodec	2
27	C60	Dodec	$-1+2\tau$
28	C60	Dodec	$1+\tau$
29	C60	Dodec	3
30	C60	Dodec	2τ
31	C60	Dodec	$2+\tau$
32	C60	Dodec	$1+2\tau$
33	C60	Dodec	3τ
34	C60	Dodec	$2+2\tau$
35	C60	Dodec	$1+3\tau$
36	C60	Dodec	$2+3\tau$
37	C60	Dodec	$3+3\tau$
38	C60	Icos	$-1+\tau$
39	C60	Icos	1
40	C60	Icos	$-2+2\tau$
41	C60	Icos	τ
42	C60	Icos	2
43	C60	Icos	$-1+2\tau$
44	C60	Icos	$1+\tau$
45	C60	Icos	3
46	C60	Icos	2τ
47	C60	Icos	$2+\tau$
48	C60	Icos	$1+2\tau$
49	C60	Icos	3τ

Table A.10: The point-arrays with a start of C_{60} translated along 3- and 5-fold axes.

Number	Start	Translation Direction	Translation Amount
1	Icos	IDD	$-3+2\tau$
2	Icos	IDD	$2-\tau$
3	Icos	IDD	$-1+\tau$
4	Icos	IDD	$-4+3\tau$
5	Icos	IDD	1
6	Icos	IDD	$3-\tau$
7	Icos	IDD	τ
8	Icos	IDD	$-1+2\tau$
9	Icos	IDD	$1+\tau$
10	Icos	IDD	$2+\tau$
11	Icos	Dodec	$2/3-\tau/3$
12	Icos	Dodec	$-1/3+\tau/3$
13	Icos	Dodec	$-2/3+2\tau/3$
14	Icos	Dodec	$-1/3+2\tau/3$
15	Icos	Dodec	$1/3+\tau/3$
16	Icos	Dodec	$5/3-\tau/3$
17	Icos	Dodec	$2/3+\tau/3$
18	Icos	Dodec	$1/3+2\tau/3$
19	Icos	Dodec	$2/3+2\tau/3$
20	Icos	Dodec	$1/3+5\tau/3$
21	Icos	Icos	$7/5-4\tau/5$
22	Icos	Icos	$3/5-\tau/5$
23	Icos	Icos	$2/5+\tau/5$
24	Icos	Icos	$-2/5+4\tau/5$
25	Icos	Icos	1
26	Icos	Icos	$3/5+4\tau/5$

Table A.11: The twisted point-arrays with an icosahedral start.

Number	Start	Translation Direction	Translation Amount
27	Dodec	IDD	$5 - 3\tau$
28	Dodec	IDD	$-3 + 2\tau$
29	Dodec	IDD	$2 - \tau$
30	Dodec	IDD	$-6 + 4\tau$
31	Dodec	IDD	$-1 + \tau$
32	Dodec	IDD	$4 - 2\tau$
33	Dodec	IDD	$-4 + 3\tau$
34	Dodec	IDD	1
35	Dodec	IDD	$6 - 3\tau$
36	Dodec	IDD	$-2 + 2\tau$
37	Dodec	IDD	$3 - \tau$
38	Dodec	IDD	τ
39	Dodec	IDD	$-3 + 3\tau$
40	Dodec	IDD	2
41	Dodec	IDD	$-1 + 2\tau$
42	Dodec	IDD	$1 + \tau$
43	Dodec	IDD	3
44	Dodec	IDD	2τ

Table A.12: The twisted point-arrays with a dodecahedral start translated along a 2-fold axis.

Number	Start	Translation Direction	Translation Amount
45	Dodec	Dodec	$5/3 - \tau$
46	Dodec	Dodec	$2/3 - \tau/3$
47	Dodec	Dodec	$-4/3 + \tau$
48	Dodec	Dodec	$1/3$
49	Dodec	Dodec	$-5/3 + 4\tau/3$
50	Dodec	Dodec	$2/3$
51	Dodec	Dodec	$1/3 + \tau/3$
52	Dodec	Dodec	$-2/3 + \tau$
53	Dodec	Dodec	1
54	Dodec	Dodec	$-1/3 + \tau$
55	Dodec	Dodec	$-2/3 + 4\tau/3$
56	Dodec	Dodec	$5/3$
57	Dodec	Dodec	$1/3 + \tau$
58	Dodec	Dodec	$2/3 + \tau$
59	Dodec	Dodec	$1/3 + 4\tau/3$
60	Dodec	Icos	$-4/5 + 3\tau/5$
61	Dodec	Icos	$3/5 - \tau/5$
62	Dodec	Icos	$-8/5 + 6\tau/5$
63	Dodec	Icos	$-1/5 + 2\tau/5$
64	Dodec	Icos	$2/5 + \tau/5$
65	Dodec	Icos	$9/5 - 3\tau/5$
66	Dodec	Icos	$-6/5 + 7\tau/5$
67	Dodec	Icos	$8/5 - \tau/5$
68	Dodec	Icos	$-3/5 + 6\tau/5$
69	Dodec	Icos	$4/5 + 2\tau/5$

Table A.13: The twisted point-arrays with a dodecahedral start translated along 3- and 5-fold axes.

Number	Start	Translation Direction	Translation Amount
70	IDD	IDD	$-3/2+\tau$
71	IDD	IDD	$1-\tau/2$
72	IDD	IDD	$-1/2+\tau/2$
73	IDD	IDD	$2-\tau$
74	IDD	IDD	$-2+3\tau/2$
75	IDD	IDD	$1/2$
76	IDD	IDD	$-1+\tau$
77	IDD	IDD	$3/2-\tau/2$
78	IDD	IDD	$\tau/2$
79	IDD	IDD	$5/2-\tau$
80	IDD	IDD	$-3/2+3\tau/2$
81	IDD	IDD	1
82	IDD	IDD	$-1/2+\tau$
83	IDD	IDD	$2-\tau/2$
84	IDD	IDD	$1/2+\tau/2$
85	IDD	IDD	$-1+3\tau/2$
86	IDD	IDD	$3/2$
87	IDD	IDD	τ
88	IDD	IDD	$1+\tau/2$
89	IDD	IDD	$-1/2+3\tau/2$
90	IDD	IDD	$1/2+\tau$
91	IDD	IDD	$3/2+\tau/2$
92	IDD	IDD	$3\tau/2$
93	IDD	IDD	$1+\tau$
94	IDD	IDD	$1/2+3\tau/2$
95	IDD	IDD	$3/2+\tau$

Table A.14: The twisted point-arrays with an icosidodecahedral start translated along a 2-fold axis.

Number	Start	Translation Direction	Translation Amount
96	IDD	Dodec	$1/6$
97	IDD	Dodec	$1/2 - \tau/6$
98	IDD	Dodec	$\tau/6$
99	IDD	Dodec	$-1/2 + \tau/2$
100	IDD	Dodec	$1/6 + \tau/6$
101	IDD	Dodec	$1/2$
102	IDD	Dodec	$\tau/3$
103	IDD	Dodec	$-1/2 + 2\tau/3$
104	IDD	Dodec	$-1/6 + \tau/2$
105	IDD	Dodec	$2/3$
106	IDD	Dodec	$-1/3 + 2\tau/3$
107	IDD	Dodec	$1/2 + \tau/6$
108	IDD	Dodec	$\tau/2$
109	IDD	Dodec	$2/3 + \tau/6$
110	IDD	Dodec	$1/2 + \tau/3$
111	IDD	Dodec	$2\tau/3$
112	IDD	Dodec	$1/6 + 2\tau/3$
113	IDD	Dodec	$1/2 + \tau/2$
114	IDD	Dodec	$2/3 + \tau/2$
115	IDD	Dodec	$1/2 + 2\tau/3$
116	IDD	Dodec	$2/3 + 2\tau/3$
117	IDD	Dodec	$1/3 + \tau$
118	IDD	Dodec	$1/2 + \tau$
119	IDD	Dodec	$1/2 + 7\tau/6$
120	IDD	Icos	$-2/5 + 3\tau/10$
121	IDD	Icos	$3/10 - \tau/10$
122	IDD	Icos	$-1/2 + \tau/2$
123	IDD	Icos	$-1/5 + 2\tau/5$
124	IDD	Icos	$1/2$
125	IDD	Icos	$1/10 + 3\tau/10$
126	IDD	Icos	$4/5 - \tau/10$
127	IDD	Icos	$2/5 + \tau/5$
128	IDD	Icos	$\tau/2$
129	IDD	Icos	$-2/5 + 4\tau/5$
130	IDD	Icos	$3/10 + 2\tau/5$
131	IDD	Icos	$1/2 + \tau/2$
132	IDD	Icos	$1/10 + 4\tau/5$
133	IDD	Icos	$4/5 + 2\tau/5$

Table A.15: The twisted point-arrays with an icosidodecahedral start translated along 3- and 5-fold axes.

Number	Start	Translation Direction	Translation Amount
1	C60	IDD	$5-3\tau$
2	C60	IDD	$-3+2\tau$
3	C60	IDD	$2-\tau$
4	C60	IDD	$7-4\tau$
5	C60	IDD	$-1+\tau$
6	C60	IDD	$-4+3\tau$
7	C60	IDD	1
8	C60	IDD	$-7+5\tau$
9	C60	IDD	$-2+2\tau$
10	C60	IDD	$3-\tau$
11	C60	IDD	$-5+4\tau$
12	C60	IDD	τ
13	C60	IDD	$5-2\tau$
14	C60	IDD	$-3+3\tau$
15	C60	IDD	2
16	C60	IDD	$-6+5\tau$
17	C60	IDD	$7-3\tau$
18	C60	IDD	$-1+2\tau$
19	C60	IDD	$4-\tau$
20	C60	IDD	$1+\tau$
21	C60	IDD	$-7+6\tau$
22	C60	IDD	$-2+3\tau$
23	C60	IDD	3
24	C60	IDD	$-5+5\tau$
25	C60	IDD	$8-3\tau$

Table A.16: The twisted point-arrays with a start of C_{60} translated along a 2-fold axis (A).

Number	Start	Translation Direction	Translation Amount
26	C60	IDD	$5-\tau$
27	C60	IDD	$-3+4\tau$
28	C60	IDD	$2+\tau$
29	C60	IDD	$7-2\tau$
30	C60	IDD	$-1+3\tau$
31	C60	IDD	$1+2\tau$
32	C60	IDD	$6-\tau$
33	C60	IDD	$3+\tau$
34	C60	IDD	3τ
35	C60	IDD	5
36	C60	IDD	$-3+5\tau$
37	C60	IDD	$-1+4\tau$
38	C60	IDD	$4+\tau$
39	C60	IDD	$1+3\tau$
40	C60	IDD	$-2+5\tau$
41	C60	IDD	$2+3\tau$
42	C60	IDD	$-1+5\tau$
43	C60	IDD	$1+4\tau$
44	C60	IDD	$3+3\tau$
45	C60	IDD	5τ
46	C60	IDD	$5+2\tau$
47	C60	IDD	$4+3\tau$
48	C60	IDD	$3+4\tau$
49	C60	IDD	$5+3\tau$

Table A.17: The twisted point-arrays with a start of C_{60} translated along a 2-fold axis (B).

Number	Start	Translation Direction	Translation Amount
50	C60	Dodec	$-1+2\tau/3$
51	C60	Dodec	$-4/3+\tau$
52	C60	Dodec	$1/3$
53	C60	Dodec	$-2/3+2\tau/3$
54	C60	Dodec	$5/3-2\tau/3$
55	C60	Dodec	$2/3$
56	C60	Dodec	$-4/3+4\tau/3$
57	C60	Dodec	$-2/3+\tau$
58	C60	Dodec	1
59	C60	Dodec	$-5/3+5\tau/3$
60	C60	Dodec	$2\tau/3$
61	C60	Dodec	$-1+4\tau/3$
62	C60	Dodec	$2/3+\tau/3$
63	C60	Dodec	$-1/3+\tau$
64	C60	Dodec	$1+\tau/3$
65	C60	Dodec	$-5/3+2\tau$
66	C60	Dodec	τ
67	C60	Dodec	$5/3$
68	C60	Dodec	$-4/3+2\tau$
69	C60	Dodec	$1/3+\tau$
70	C60	Dodec	2
71	C60	Dodec	$-2/3+5\tau/3$
72	C60	Dodec	$-1+2\tau$
73	C60	Dodec	$2/3+\tau$
74	C60	Dodec	$4/3+2\tau/3$
75	C60	Dodec	$3-\tau/3$
76	C60	Dodec	$-2/3+2\tau$

Table A.18: The twisted point-arrays with a start of C_{60} along a 3-fold axis (A).

Number	Start	Translation Direction	Translation Amount
77	C60	Dodec	$1+\tau$
78	C60	Dodec	$8/3$
79	C60	Dodec	$5\tau/3$
80	C60	Dodec	$2/3+4\tau/3$
81	C60	Dodec	$-1/3+2\tau$
82	C60	Dodec	$4/3+\tau$
83	C60	Dodec	3
84	C60	Dodec	2τ
85	C60	Dodec	$5/3+\tau$
86	C60	Dodec	$1/3+2\tau$
87	C60	Dodec	$2+\tau$
88	C60	Dodec	$5/3+4\tau/3$
89	C60	Dodec	$2/3+2\tau$
90	C60	Dodec	$4/3+5\tau/3$
91	C60	Dodec	$2+4\tau/3$
92	C60	Dodec	$1+2\tau$
93	C60	Dodec	$4/3+2\tau$
94	C60	Dodec	$2+5\tau/3$
95	C60	Dodec	$1+7\tau/3$
96	C60	Dodec	$5/3+2\tau$
97	C60	Dodec	$2+2\tau$
98	C60	Dodec	$5/3+7\tau/3$
99	C60	Dodec	$5/3+3\tau$
100	C60	Dodec	$1+11\tau/3$
101	C60	Dodec	$4/3+11\tau/3$
102	C60	Dodec	$4/3+4\tau$

Table A.19: The twisted point-arrays with a start of C_{60} translated along a 3-fold axis (B).

Number	Start	Translation Direction	Translation Amount
103	C60	Icos	$-11/5 + 7\tau/5$
104	C60	Icos	$3/5 - \tau/5$
105	C60	Icos	$-8/5 + 6\tau/5$
106	C60	Icos	$-1/5 + 2\tau/5$
107	C60	Icos	$6/5 - 2\tau/5$
108	C60	Icos	$-9/5 + 8\tau/5$
109	C60	Icos	$-2/5 + 4\tau/5$
110	C60	Icos	$-6/5 + 7\tau/5$
111	C60	Icos	$12/5 - 4\tau/5$
112	C60	Icos	$8/5 - \tau/5$
113	C60	Icos	$4/5 + 2\tau/5$
114	C60	Icos	$-7/5 + 9\tau/5$
115	C60	Icos	$11/5 - 2\tau/5$
116	C60	Icos	$3/5 + 4\tau/5$
117	C60	Icos	$-1/5 + 7\tau/5$
118	C60	Icos	$6/5 + 3\tau/5$
119	C60	Icos	$13/5 - \tau/5$
120	C60	Icos	$2/5 + 6\tau/5$
121	C60	Icos	$-2/5 + 9\tau/5$
122	C60	Icos	$-6/5 + 12\tau/5$
123	C60	Icos	$1/5 + 8\tau/5$
124	C60	Icos	$8/5 + 4\tau/5$
125	C60	Icos	$7/5 + 6\tau/5$
126	C60	Icos	$-1/5 + 12\tau/5$
127	C60	Icos	$13/5 + 4\tau/5$
128	C60	Icos	$8/5 + 9\tau/5$
129	C60	Icos	$4/5 + 12\tau/5$
130	C60	Icos	$11/5 + 8\tau/5$

Table A.20: The twisted point-arrays with a start of C_{60} translated along a 5-fold axis.

Bibliography

[1] Chemspider, accessed 23/05/2011. URL www.chemspider.com. 148

[2] T. S. Baker, N. H. Olson, and S. D. Fuller. Adding the third dimension to virus life cycles: three-dimensional reconstruction of icosahedral viruses from cryo-electron micrographs. *Microbiol. Mol. Biol. Rev.*, 4(63):862–922, 1999. 6, 17

[3] M. Bando, Y. Morimoto, T. Sato, T. Tsukihara, Y. Yokota, K. Fukuyama, and H. Matsubara. Crystal structural analysis of tobacco necrosis virus at 5Å resolution. *Acta Crystallographica Section D*, 50(D50):878–883, 1994. 129

[4] G. Basnak, V. L. Morton, Óttar Rolfsson, N. J. Stonehouse, A. E. Ashcroft, and P. G. Stockley. Viral genomic single-stranded RNA directs the pathway toward a $T = 3$ capsid. *J. Mol. Biol.*, 395(5):924–936, 2010. 33

[5] F. C. Bernstein, T. F. Koetzle, G. J. Williams, E. E. Meyer Jr., M. D. Brice, J. R. Rodgers, O. Kennard, T.Shimanouchi, and M. Tasumi. The protein data bank: A computer-based archival

file for macromolecular structures. *J. Mol. Biol.*, (112):535, 1977. 31

[6] A. Bravais. Mémoire sur les systèmes formés par les points distribués régulièrement sur un plan ou dans l'espace. *J. Ecole Polytech.*, (19):1–128, 1850. 38

[7] M. Breitbart and F. Rohwer. Here a virus, there a virus, everywhere the same virus? *Trends Microbiol*, 13:278–284, 2005. 16

[8] B.R. Brooks, C. L. Brooks III, A.D. MacKerell Jr., L. Nilsson, R.J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A.R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R.W. Pastor, C.B. Post, J.Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D.M. York, and M. Karplus. CHARMM: The biomolecular simulation program. *J. Comput. Chem.*, 30:1545–1614, 2009. 65

[9] D. L. D. Caspar and A. Klug. Physical principles in the construction of regular viruses. *Cold Spring Harbor Symp. Quant. Biol.*, 27:1–24, 1962. 20, 152

[10] H. S. M. Coxeter. *A Spectrum of Mathematics*, chapter Virus Macromolecules and Geodesic Domes, pages 98–107. Auckland U.P., 1972. 20

[11] F. H. C. Crick and J. D. Watson. The structure of small viruses. *Nature*, 177:473–475, 1956. 20

[12] R. A. Crowther, N. A. Kiselev, B. Böttchera, J. A. Berriman, G. P. Borisova, V. Ose, and P. Pumpens. Three-dimensional structure of hepatitis B virus core particles determined by electron cryomicroscopy. *Cell*, 77:943–950, 1994. 22

[13] B. Devkota, A. S. Petrov, S. Lemieux, M. B. Boz, L. Tang, A. Schneemann, J. E. Johnson, and S. C. Harvey. Structural and electrostatic characterization of pariacoto virus: Implications for viral assembly. *Biopolymers*, 91(7):530–538, 2009. 75

[14] E. C. Dykeman, N.E. Grayson, K. Toropova, N.A. Ranson, P.G. Stockley, and R. Twarock. Simple rules for efficient assembly predict the layout of a packaged viral RNA. *J. Mol. Biol.*, 408(3):399–407, 2011. 34

[15] E. C. Dykeman and O. F. Sankey. Atomistic modeling of the low-frequency mechanical modes and Raman spectra of icosahedral virus capsids. *Phys Rev E*, 81(2), 2010. 31

[16] W. C. Earnshaw and S. C. Harrison. DNA arrangement in isometric phage heads. *Nature*, 268(5621):598–602, 1977. 33

[17] D. Eberly. Distance between point and triangle in 3d, accessed 7th January 2011. URL www.geometrictools.com/Documentation/ Distance-Point3Triangle3.pdf. 54

[18] W. Fiers, R. Contreras, F. Duerinck, G. Haegeman, D. Iserentant, J. Merregaert, W. Min Jou, F. Molemans, A. Raeymaekers, A. Ven den Berghe, G. Volckaert, and M. Ysebaert. Complete nucleotide sequence of bacteriophage MS2 RNA: primary and secondary structure of the replicase gene. *Nature*, 260(5551):500–507, 1976. 33

[19] G. Fischer. Distance between a point and a triangle in 3d, accessed 7th January 2011. URL http://www.mathworks.com/matlabcentral/fileexchange/22857-distance-between-a-point-and-a-triangle-in-3d. 54

[20] F. Furche and R. Ahlrichs. Fullerene C80: Are there still more isomers? *Journal of Chemical Physics*, 114(23):10362–10367, 2001. 144

[21] E. R. Gansner and S. C. North. An open graph visualization system and its applications to software engineering. *Software - Practice and Experience*, 30(11):1203–1233, 2000. 150

[22] W. M. Gelbart and C. M. Knobler. Virology: Pressurized viruses. *Science*, (323):1682–1683, 2009. 35

[23] T. D. Goddard, C. C. Huang, and T. E. Ferrin. Visualizing density maps with UCSF Chimera. *J. Struct. Biol.*, (157):281–287, 2007. 31

[24] M. Goldberg. A class of multi-symmetric polyhedra. *Tohoku Math. J.*, 43:104–108, 1937. 22, 24

[25] M. Goldberg. Viruses and a mathematical problem. *J. Mol. Biol.*, 24:337–338, 1967. 32

[26] R. Golmohammadi, K. Valegård, K. Fridborg, and L. Liljas. The refined structure of Bacteriophage MS2 at 2.8Å resolution. *J Mol Biol.*, 3(234):620–39, Dec 1993. 89

[27] R. Grasman and R. B. Gramacy. *Geometry: Mesh Generation and Surface Tesselation*, 2008. R package version 0.1-7. 54

[28] N.E. Grayson, A. Taormina, and R. Twarock. DNA duplex cage structures with icosahedral symmetry. *Theor. Comp. Sci.*, 410(15):1440–1447, April 2009. 33

[29] J. M. Grimes, J. N. Burroughs, P. Gouet, J. M. Diprose, R. Malby, S. Zintara, P. P. C. Mertens, and D. I. Stuart. The atomic structure of the bluetongue virus core. *Nature*, 395:470–478, Oct 1998. 58, 163

[30] J.M. Hawkins, A. Meyer, T. A. Lewis, S. D. Loren, and F. J. Hollander. Crystal structure of osmylated C60: Confirmation of the soccer ball framework. *Science*, 252:312–313, 1991. 141

[31] F. H. Hennrich, R. H. Michel, A. Fischer, S. Richard-Schneider, S. Gilb, M. M. Kappes, D. Fuchs, M. Bürk, K. Kobayashi, and S. Nagase. Isolation and characterization of C80. *Angewandte Chemie International Edition in English*, 35:1732–1734, 1996. 141, 144

[32] P. Hopper, S.C. Harrison, and R.T. Sauer. Structure of tomato bushy stunt virus. v. coat protein sequence determination and its structural implications. *J. Mol. Biol.*, 4(177):701–713, Aug 1984. 99

[33] R. W. Horne and P. Wildy. Symmetry in virus architecture. *Virology*, 15:348–373, 1961. 20

[34] S. Iijima. Direct observation of the tetrahedral bonding in graphitized carbon black by high resolution electron microscopy. *J. Cryst. Growth.*, 50(3):675–683, 1980. 140

[35] G. Indelicato, P. Cermelli, D. G. Salthouse, S. Racca, G. Zanzotto, and R. Twarock. A crystallographic approach to structural transitions in icosahedral viruses. *J. Math. Biol.*, 64:745–773, 2012. 39, 107, 108, 112, 157, 167

[36] A. Janner. Form, symmetry and packing of biomacromolecules. I. concepts and tutorial examples. *Acta Crystallographica Section A*, 2010. 30

[37] A. Janner. Form, symmetry and packing of biomacromolecules. II. serotypes of human rhinovirus. *Acta Crystallographica Section A*, 2010. 30

[38] A. Janner. Form, symmetry and packing of biomacromolecules. III. antigenic, receptor and contact binding sites in picornaviruses. *Acta Crystallographica Section A*, 67:174–189, 2011. 30

[39] A. Janner. Form, symmetry and packing of biomacromolecules. IV. filled capsids of cowpea, tobacco, MS2 and pariacoto RNA viruses. *Acta Crystallographica Section A*, 67:517–520, 2011. 30, 31, 80, 83

[40] A. Janner. Form, symmetry and packing of biomacromolecules. V. shells with boundaries at anti-nodes of resonant vibrations in icosahedral RNA viruses. *Acta Crystallographica Section A*, 67:521–532, 2011. 30

[41] N. Jonoska, A. Taormina, and R. Twarock. DNA cages with icosahedral symmetry in bionanotechnology. In A. Condon, D. Harel, J. N. Kok, A. Salomaa, and E. Winfree, editors, *Algorithmic Bioprocesses*, Natural Computing Series, pages 141–158. Springer Berlin Heidelberg, 2009. 33

[42] T. Keef and R. Twarock. New insights into viral architecture via affine extended symmetry groups. *Comp. and Math. Methods in Medicine*, 9:221–229, September 2008. 15, 30, 41, 45, 46, 51

[43] T. Keef and R. Twarock. Affine extensions of the icosahedral group with applications to the three-dimensional organisation of simple viruses. *J Math Biol*, 59(3):287–313, 2009. 15, 30, 41, 45, 46

[44] T. Keef and R. Twarock. *Emerging Topics in Physical Virology*, chapter 3. Imperial College Press, London, 2010. 15

[45] T. Keef, R. Twarock, and K. M. ElSawy. Blueprints for viral capsids in the family of Papovaviridae. *J. Theor. Biol.*, 253(4):808–816, August 2008. 30

[46] T. Keef, J. Wardman, N.A. Ranson, P. G. Stockley, and R. Twarock. Structural constraints on the 3d geometry of simple viruses. *Acta Crystallographica Section A*, page to appear, 2012. 15

[47] B. Keller, J. Dubochet, M. Adrian, M. Maeder, M. Wurtz, and E. Kellenberger. Length and shape variants of the bacteriophage T4 head: mutations in the scaffolding core genes 68 and 22. *J. Virol.*, 62(8):2960–2969, 1988. 19

[48] J. Kindt, S. Tzlil, A. Ben-Shaul, and W. M. Gelbart. DNA packaging and ejection forces in bacteriophage. *PNAS*, 98(24):13671–13674, 2001. 35

[49] T. Klose, Y. G. Kuznetsov, C. Xiao, S. Sun, A. McPherson, and M. G. Rossmann. The three-dimensional structure of mimivirus. *Intervirology*, (53):268–273, 2010. 16, 18

[50] K. Komatsu, K. Fujiwara, T. Tanaka, and Y. Murata. The fullerene dimer C_{120} and related carbon allotropes. *Carbon*, 38:1529–1534, 2000. 148, 149

[51] H. Kroto. Carbon onions introduce new flavour to fullerene studies. *Nature*, 359:670–671, 1992. 141

[52] H. Kroto, J. R. Heath, S. C. O'Brien, R. F. Curl, and R. E. Smalley. C60: Buckminsterfullerene. *Nature*, 318:162–163, 1985. 140

[53] R. J. Kuhn, W. Zhang, M. G. Rossmann, S. V. Pletnev, J. Corver, E. Lenches, C. T. Jones, S. Mukhopadhyay, P. R. Chipman, E. G. Strauss, T. S. Baker, and J. H. Strauss. Structure of dengue virus: implications for flavivirus organization, maturation, and fusion. *Cell*, 108(5):717–725, 2002. 162

[54] E. F. Kustov, V. I. Nefedov, A. V. Kalinin, and G. S. Chernova. Classification system for fullerenes. *Russian Journal of Inorganic Chemistry*, 53(9):1384–1395, 2008. 141, 150

[55] R. Langlet, A. Mayer, N. Geuquet, H. Amara, M. Vandescuren, L. Henrard, S. Maksimenko, and Ph. Lambin. Study of the polarizability of fullerenes with a monopole–dipole interaction model. *Diamond & Related Materials*, 16:2145–2149, 2007. 141

[56] S. B. Larson, J. Day, M. A. Canady, A. Greenwood, and A. McPherson. Refined structure of Desmodium Yellow Mottle Tymovirus at 2.7Å resolution. *J. Mol. Biol.*, (301):625–642, 2000. 134

[57] S. B. Larson, J. Day, A. Greenwood, and A. McPherson. Refined structure of satellite tobacco mosaic virus at 1.8Å resolution. *J. Mol. Biol*, 277(1):37–59, 1998. 16, 18, 116

[58] S. B. Larson and A. McPherson. Satellite tobacco mosaic virus RNA: structure and implications for assembly. *Current Opinion in Structural Biology*, 11(1):59–65, 2001. 116

[59] C. L. Lawson, M. L. Baker, C. Best, C. Bi, M. Dougherty, P. Feng, G. van Ginkel, B. Devkota, I. Lagerstedt, S. J. Ludtke, R. H. Newman, T. J. Oldfield, I. Rees, G. Sahni, R. Sala, S. Velankar, J. Warren, J. D. Westbrook, K. Henrick, G. J. Kleywegt, H. M. Berman, and W. Chiu. EMDataBank.org: unified data resource for CryoEM. *Nucleic Acids Res.*, (39(Database issue)):D456–464, 2011. 30, 33, 35, 90, 94

[60] K. K. Lee, L. Han, H. Turuta, C. Moyer, J. F. Conway, R. L. Duda, R. W. Hendrix, and A. C. Steven. Virus capsid expansion driven by the capture of mobile surface loops. *Structure*, (16):1491–1502, 2008. 157

[61] P. G. Leiman, S. Kanamaru, V. V. Mesyanzhinov, F. Arisaka, and M. G. Rossmann. Structure and morphogenesis of bacteriophage T4. *Cell Mol Life Sci.*, 60(11):2356–2370, 2003. 16

[62] J. Leszczynski and I. Yanov. Possibility of the existence of non-carbon fullerenes: Ab initio HF and DFT/B3LYP studies of the IV main group fullerene-like species. *J. Phys. Chem. A*, (103):396–401, 1999. 141

[63] L.S. Levitov and J. Rhyner. Crystallography of quasicrystals; application to icosahedral symmetry. *J. Phys. France*, 49(49):1835–1849, 1988. 38

[64] J. Lidmar, L. Mirny, and D. R. Nelson. Virus shapes and buckling transitions in spherical shells. *Phys Rev E*, 68(5):051910, 2003. 31

[65] H. Liu, C. Qu, J. E. Johnson, and D. A. Case. Pseudo-atomic models of swollen CCMV from cryo-electron microscopy data. *J. Struct. Biol.*, (142):356–363, 2003. 112

[66] L. F. Liu, J. L. Davis, and R. Calendar. Novel topologically knotted DNA from bacteriophage P4 capsids: studies with DNA topoisomerases. *Nucleic Acids Research*, 9(16):3979–3989, 1981. 35

[67] L. F. Liu, L. Perkocha, R. Calendar, and J. C. Wang. Knotted DNA from bacteriophage capsids. *Proc. Natl. Acad. Sci. USA*, 78(9):5498–5502, 1981. 35

[68] J. M. Maisog, Y. Wang, G. Luta, and J. Liu. *ptinpoly: Point-In-Polyhedron Test (3D)*, 2010. R package version 1.4/r7. 64

[69] R. V. Mannige and C. L. Brooks III. Tilable nature of virus capsids and the role of topological constraints in natural capsid design. *Phys Rev E Stat Nonlin Soft Matter Phys*, 77(5 Pt 1), 2008. 26

[70] R. V. Mannige and C. L. Brooks III. Geometric considerations in virus capsid size specificity, auxiliary requirements, and buckling. *Proc Natl Acad Sci*, 106(21):8531–8536, 2009. 32

[71] R. V. Mannige and C. L. Brooks III. Periodic table of virus capsids: Implications for natural selection and design. *PLoS ONE*, 5(3), 2010. 26

[72] K. G. McKay, H. W. Kroto, and D. J. Wales. Simulated transmission electron microscope images and characterisation of concentric shell and icospiral graphitic microparticles. *J. Chem. Soc., Faraday Trans.*, 88:2815–2821, 1992. 141

[73] K. Namba, R. Pattanayek, and G. Stubbs. Visualization of protein-nucleic acid interactions in a virus. refined structure of intact tobacco mosaic virus at 2.9Å resolution by X-ray fiber diffraction. *J. Mol. Biol.*, (208):307–325, 1989. 16, 19

[74] N. Nandhagopal, A.A. Simpson, J.R. Gurnon, X. Yan, T.S. Baker, M.V. Graves, J.L. Van Etten, and M.G. Rossmann. The structure and evolution of the major capsid protein of a large, lipid-containing DNA virus. *PNAS*, 23(99):14758–14763, 2002. 163

[75] NIAID. HIV virion, accessed 28th March 2011. URL http://www.niaid.nih.gov/topics/HIVAIDS/ Understanding/Biology/Pages/hivVirionLargeImage.aspx. 16, 19

[76] Y. Oda, K. Saeki, Y. Takahashi, T. Maeda, H. Naitow, T. Tsukihara, and K. Fukuyama. Crystal structure of tobacoo necrosis virus at 2.25Å resolution. *J. Mol. Biol.*, 300(1):153–169, 2000. 129

[77] K. Peeters and A. Taormina. Group theory of icosahedral virus capsids: a dynamical top-down approach. *J. Theo. Biol.*, 256(4):607–624, 2009. 31

[78] R. Penrose. The role of aesthetics in pure and applied mathematical research. *Bull. Inst. Math. Appl.*, 10:266–271, 1974. 27

[79] E.F. Pettersen, T.D. Goddard, C.C. Huang, G.S. Couch, D.M. Greenblatt, E.C. Meng, and T.E. Ferrin. UCSF Chimera - a visualization system for exploratory research and analysis. *J. Comput. Chem.*, 25(13):1605–1612, 2004. 30, 31

[80] O. Pornillos, B.K. Ganser-Pornillos, and M. Yeager. Atomic-level modelling of the HIV capsid. *Nature*, 469:424–427, 2011. 16, 19

[81] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0. 54

[82] I. Rayment, T. Baker, D. L. D. Caspar, and W. T. Murakami. Polyoma virus capsid structure at 22.5Å resolution. *Nature*, 295:110–115, 1982. 26

[83] V. S. Reddy, P. Natarajan, B. Okerberg, K. Li, K.V. Damodaran, R.T. Morton, C. L. Brooks III, and J. E. Johnson. Virus particle explorer (VIPER), a website for virus capsid structures and their computational analyses. *J. Virol.*, 75(24):11943–11947, 2001. 31, 74

[84] J. S. Richardson. The anatomy and taxonomy of protein structure. *Advances in Protein Chemistry*, 34:167–339, 1981. 154

[85] I. K. Robinson and S. C. Harrison. Structure of the expanded state of tomato bushy stunt virus. *Nature*, (297):563–568, 1982. 30

[86] A.M. Roseman, J.A. Berriman, S.A. Wynee, P.J. Butler, and R.A. Crowther. A structural model for maturation of the hepatitis b virus core. *PNAS*, 44(102):15821–15826, 2005. 123, 127, 128

[87] M. G. Rossmann. Constraints on the assembly of spherical virus particles. *Virology*, (134):1–11, 1984. 26

[88] V. V. Rybenkov, N. R. Cozzarelli, and A. V. Vologodskii. Probability of DNA knotting and the effective diameter of the DNA double helix. *Proc. Natl. Acad. Sci. USA*, 90(11):5307–5311, 1993. 35

[89] C. Sachse, J.Z. Chen, P. D. Coureux, M. E. Stroupe, M. Fandrich, and N. Grigorieff. High-resolution electron microscopy of helical specimens: a fresh look at tobacco mosaic virus. *J. Mol. Biol.*, 371:812–835, 2007. 16

[90] M. F. Sanner, A. J. Olson, and J. C. Spehner. Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers*, 38(3):305–320, 1996. 64

[91] W. Scherrer. Die einlagerung eines regularen vielecks in ein gitter. *Elemente der Mathematik*, 1(6):97–98, 1946. 38

[92] Schrödinger. *The PyMOL Molecular Graphics System*. LLC. 18, 31

[93] S. Seitz, S. Urban, C. Antoni, and B. Bottcher. Cryo-electron microscopy of hepatitis B virions reveals variability in envelope capsid interactions. *EMBOJ*, (26):4160–4167, 2007. 33, 35, 123, 126

[94] M. Senechal. *Quasicrystals and Geometry*. Cambridge University Press, 1996. 30, 38

[95] J.A. Speir, S. Munshi, G. Wang, T.S. Baker, and J. E. Johnson. Structures of the native and swollen forms of Cowpea Chlorotic Mottle virus determined by X-ray crystallography and cryo-electron microscopy. *Structure*, 3(1):63–78, 1995. 107

[96] T. Stehle, S. J. Gamblin, Y. Yan, and S. C. Harrison. The structure of simian virus 40 refined at 3.1Å resolution. *Structure*, 4(2):165–182, 1996. 119

[97] F. Tama and C. L. Brooks III. The mechanism and pathway of pH induced swelling in Cowpea Chlorotic Mottle virus. *J. Mol. Biol.*, 318:733–747, 2002. 112

[98] L. Tang, K.N. Johnson, L.A. Ball, T. Lin, M. Yeager, and J. E. Johnson. The structure of Pariacoto virus reveals a dodecahedral

cage of duplex RNA. *Nat. Struct. Biol.*, 1(8):77–83, Jan 2001. 33, 75, 81

[99] K. Tars, M. Bundule, K. Fridborg, and L. Liljas. The crystal structure of Bacteriophage GA and a comparison of bacteriophages belonging to the major groups of *Escherichia coli* leviviruses. *J. Mol. Biol.*, (271):759–773, 1997. 95

[100] SymPy Development Team. *SymPy: Python library for symbolic mathematics*, 2008. 44

[101] H. Terrones and M. Terrones. Quasiperiodic icosahedral graphite sheets and high-genus fullerenes with nonpositive Gaussian curvature. *Phys. Rev. B*, 55(15):9969–9974, 1997. 141

[102] M. Tihova, K. A. Dryden, T. L. Le, S. C. Harvey, J. E. Johnson, M. Yeager, and A. Schneemann. Nodavirus coat protein imposes dodecahedral RNA structure independent of nucleotide sequence and length. *J. Virology*, 78(6):2897–2905, 2004. 33

[103] P. A. Timmins, D. Wild, and J. Witz. The three-dimensional distribution of RNA and protein in the interior of Tomato Bushy Stunt virus: a neutron low-resolution single-crystal diffraction study. *Structure*, 2:1191–1201, 1994. 99, 104, 105, 106

[104] K. Toropova, G. Basnak, R. Twarock, P.G. Stockley, and N.A. Ranson. The three-dimensional structure of genomic RNA in bacteriophage MS2: Implications for assembly. *J. Mol. Biol.*, 375(3):824–836, January 2008. 33, 35, 90, 94

[105] K. Toropova, P. G. Stockley, and N. A. Ranson. Visualising a viral RNA genome poised for release from its receptor complex. *J. Mol. Biol*, 408(3):408–419, May 2011. 155

[106] R. Twarock. A tiling approach to virus capsid assembly explaining a structural puzzle in virology. *J. Theor. Biol.*, 226(4):477–482, 2004. 26, 27

[107] R. Twarock. The architecture of viral capsids based on tiling theory. *J. Theor. Medicine*, 6:87–90, 2005. 152

[108] R. Twarock. A toolkit for the construction of icosahedral particles with local symmetry axes, 2005. arXiv:q-bio/0508015v1. 20

[109] C. Uetrecht, C. Versluis, N. R. Watts, W. H. Roos, G. J. L. Wuite, P. T. Wingfield, A. C. Steven, and A. J. R. Heck. High-resolution mass spectrometry of viral assemblies: Molecular composition and stability of dimorphic hepatitis B virus capsids. *PNAS*, 105(27):9216–9220, 2008. 22

[110] D. Ugarte. Curling and closure of graphitic networks under electron-beam irradiation. *Nature*, 359:707–709, 1992. 141

[111] D. Ugarte. Onion-like graphitic particles. *Carbon*, 33:989–993, 1995. 141, 142

[112] K. Valegård, J. B. Murray, N. J. Stonehouse, S. van den Worm, P. G. Stockley, and L. Liljas. The three-dimensional structures of two complexes between recombinant MS2 capsids and RNA

operator fragments reveal sequence-specific protein-RNA interactions. *J Mol Biol*, 5(270):724–738, 1997. 89, 91, 94

[113] K. Valegård, J.B. Murray, P.G. Stockley, N.J. Stonehouse, and L. Liljas. Crystal structure of an RNA bacteriophage coat protein operator complex. *Nature*, 371:623 – 626, Oct 2002. 33, 89

[114] G. van Rossum. *The Python Reference Manual*, 1995. 44

[115] R. S. Williams, G. J. Williams, and J. A. Tainer. A charged performance by gp17 in viral packaging. *Cell*, (135):1169–1171, 2008. 35

[116] S.A. Wynne, R.A. Crowther, and A.G. Leslie. The crystal structure of the human hepatitis B virus capsid. *Mol Cell*, 3(6):771–780, 1999. 122

[117] C. Xiao, Y. G. Kuznetsov, S. Sun, S. L. Hafenstein, V. A. Kostyuchenko, P. R. Chipman, M. Suzan-Monti, D. Raoult, A. McPherson, and M. G. Rossmann. Structural studies of the giant mimivirus. *PLoS Biol*, 7(4):958–966, 04 2009. 16, 18

[118] X. Yan, Z. Yu, P. Zhang, A. J. Battisti, H. A. Holdaway, P. R. Chipman, C. Bajaj, M. Bergoin, M. G. Rossmann, and T. S. Baker. The capsid proteins of a large, icosahedral dsDNA virus. *J. Mol. Biol.*, 385:1287–1299, 2009. 16, 18

[119] A. M. Yoffe, P. Prinsen, A. Gopal, C. M. Knobler, W. M. Gelbart, and A. Ben-Shaul. Predicting the sizes of large RNA molecules. *PNAS*, 105(42):16153–16158, 2008. 33, 34

[120] D. York, J. P. Lu, and W. Yang. Density-functional calculations of the structure and stability of $C240$. *Phys. Rev. B*, 49(12):8526–8528, 1994. 141

[121] R. Zandi, D. Reguera, R. F. Bruinsma, W. M. Gelbart, and J. Rudnick. Origin of icosahedral symmetry in viruses. *Proc. Natl. Acad. Sci.*, 101(44):15556–15560, 2004. 32, 33, 37

[122] R. Zandi and P. van der Schoot. Size regulation of ss-RNA viruses. *Biophysical Journal*, 96:9–20, 2009. 33

[123] H. Zhu. MSMS plugin for PyMOL, 2010. Biotechnology Center (BIOTEC), TU Dresden. 64, 164