Automated Model Building of Nucleic Acids and Carbohydrates Using Experimental Data and Deep Learning Models

Jordan Singh Dialpuri

Doctor of Philosophy

University of York
Department of Chemistry

January 2026

Abstract

Understanding the structural information of proteins, nucleic acids, and carbohydrates is fundamental to gaining mechanistic and functional insights into biological processes. X-ray crystallography and, more recently, cryogenic electron microscopy are frequently used methods to study the three-dimensional structures of biological molecules. These techniques do not record three-dimensional atomic positions directly, instead volumetric density data is used to create an atomic model. This process of building models from density data is often time-intensive and requires substantial manual effort, which automated model-building methods aim to alleviate. While automated methods for protein modelling are mature, methods for nucleic acid and carbohydrate modelling often fall short or require manual intervention. The main challenge that automated model-building methods face is identifying probable regions of experimental density associated with a specific atomic group.

The use of convolutional neural networks to identify regions of experimental density corresponding to the phosphate, sugar and base groups of nucleotides was explored. Extensive experimentation with model architectures enabled the training of a single convolutional neural network that precisely identifies regions of experimental density associated with the nucleic acid phosphate, sugar, and base groups in both crystallographic and electron microscopy experimental density data. These predicted regions can then be used as a guide to automatically model nucleic acids into experimental density, with greater completeness than existing methods can provide. This new method was released as a software package called *NucleoFind*.

The model architecture designed for nucleic acids was also applied to carbohydrates to identify potential glycosylation sites successfully. Using glycosylation geometry data obtained from the Protein Data Bank, these potential sites can then be modelled with a simple method of recursive carbohydrate addition and critical assessment. The resultant method can produce complete models of N, O, and C-linked glycosylation, and lays the groundwork for a future automated carbohydrate model-building method.

Contents

List of Figures

List of Tables

Acknowledgements

While writing this thesis, I was reminded that no work can arise from a solitary mind, but instead occurs through a culmination of influences, conversations and moments of clarity provided by others. I therefore wish to acknowledge those who have shaped this undertaking directly or indirectly.

I express my sincere gratitude to my supervisors, Kathryn Cowtan and Jon Agirre, for their guidance, invaluable support, and profound expertise throughout this PhD. I would also like to extend my appreciation to Tony Wilkinson for the insightful discussions that we shared during numerous Thesis Advisory Panel meetings.

My gratitude extends to the academic and administrative staff members of the York Structural Biology Laboratory, who have created a supportive and stimulating environment conducive to academic research. I am thankful to my colleagues and friends in the Cowtan and Agirre groups for the plentiful discussions, collaborative efforts and guidance. In particular, I am indebted to Paul Bond and Lucy Schofield for their friendship and constructive discussions, which have sustained me over the years.

To my friends, thank you for offering calm in moments of uncertainty and perspective in moments of frustration. Finally, I am grateful to my family, whose quiet constancy provided the foundation on which this work was built. Their support was unwavering, needing no explanation and asking for nothing in return.

Declaration

I declare that this thesis is a presentation of original work and I am the sole author, with the exception of the published or collaborative work listed below. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References.

The following papers include work produced during this PhD and form the basis of parts of the thesis:

- **J. S. Dialpuri**, J. Agirre, K. D. Cowtan and P. S. Bond, NucleoFind: a deep-learning network for interpreting nucleic acid electron density, Nucleic Acids Research, 2024, 52 forms the basis of Part 1.
- **J. S. Dialpuri**, H. Bagdonas, M. Atanasova, L. C. Schofield, M. L. Hekkelman, R. P. Joosten, J. Agirre, Analysis and validation of overall N-glycan conformation in Privateer, Acta Crystallogr D Struct Biol, 2023, 79, 462–472. forms the basis of Chapter 5.

The following papers include work produced during this PhD, but form no content in the thesis:

- **J. S. Dialpuri**, H. Bagdonas, L. C. Schofield, P. T. Pham, L. Holland, J. Agirre, Monitoring carbohydrate 3D structure quality with the Privateer database, Beilstein J. Org. Chem., 2024, 20, 931–939.
- **J. S. Dialpuri**, H. Bagdonas, L. C. Schofield, P. T. Pham, L. Holland, P. S. Bond, F. S. Rodriguez, S. J. McNicholas, J. Agirre, Online carbohydrate 3D structure validation with the Privateer web app, Acta Crystallogr F Struct Biol Commun, 2024, 80, 30– 35.

The following papers include work contributed to during this PhD, but form no content in the thesis:

- L. C. Schofield, **J. S. Dialpuri**, G. N. Murshudov and J. Agirre, Post-translational modifications in the Protein Data Bank, Acta Crystallogr D Struct Biol, 2024, 80, 647–660.
- L. Holland, P. T. Pham, H. Bagdonas, **J. S Dialpuri**, L. C Schofield, J. Agirre, Methods for detecting, building, and improving tryptophan mannosylation in glycoprotein structures, Protein Science, 2025, 34(2).

Acronyms

CCD Chemical Component Dictionary. 37, 59, 60

FSC Fourier Shell Correlation. 40, 157

PDB Protein Data Bank. 31, 37

pp percentage point. 120

ReLU Rectified Linear Unit. 72, 90

RSCC Real Space Correlation Coefficient. 37, 40, 147, 168, 207

SNFG Symbol Nomenclature for Glycans. 10, 59, 60, 62

Glossary

activation function a non-linear function that is used to transform the output of any given regression model inside a neural network. 70

aldose a linear sugar chain capped with an aldehyde group. 56

anomer a stereoisomer which differs only about a single chiral carbon atom. 58

anomeric carbon a carbon atom in a carbohydrate that becomes an anomer after cyclisation. 58

anti conformation the most common conformation of a nucleotide where the base is positioned away from the furanose ring. 46

argmax an operation which locates the maximum value in a function. 69

automated model-building pipeline a pipeline used to automatically solve structures, commonly consisting of density modification, automated model-building and refinement. 152

backpropagation an algorithm used to efficiently update the weight matrix and bias vector of a neural network. 73

batch a small selection of the training dataset which allows more efficient computation. 72

beam-induced motion movement of a specimen in an electron microscope after the application of radiation. 38

bias vector a vector of offset parameters for each regression model inside a neural network. 72

bottleneck an architectural pattern in machine learning used to reduce the dimensionality of a machine learning model. 90

centroids the central position of a group of points. 77

chair the most stable conformation of a six-membered ring. 57

cloverleaf a model which describes the global molecular structure of a t-RNA molecule. 50

confusion matrix a table that allows for the comparison of predicted labels and actual labels. 75

convolution operation an operation used to multiply functions element-wise. 73

convolutional neural network a neural network that uses convolution operations to connect layers. 73

covariance matrix a square matrix that describes the variance between each pair of elements in a vector. 78

cross entropy a measurement for how well predicted probabilities match the actual class labels . 70

crystal soaking the process of submerging a crystal in a solution containing a compound of interest. 54

dataset a collection of data, often used to train a machine learning model. 66

deep neural networks a neural network with greater than one hidden layer. 73

deep U-Net A U-Net convolutional neural network that has a bottleneck portion as a linear vector representation. 119

endo face an atom which deviates away the carbon-5 atom of a furanose sugar. 45

Endo H an enzyme used to cleave the first NAG-β1,4-NAG linkage of a glycan. 65

epoch a complete cycle through the training dataset. 73

Euclidean distance the length of a straight line between two points . 77

exo face an atom which deviates toward the carbon-5 atom of a furanose sugar. 45

feature an individual data point. 66

feature vector a collection of features. 66

full map a three-dimensional volume reconstructed from either two half maps, or all isolated and imaged particles in single particle analysis. 39

furanose a five-membered cyclic hemiacetal sugar. 57

glycan a carbohydrate monomer or polymer attached to a protein or nucleic acid macromolecule. 56

glycoglycerolipid a glycolipid based on the triol glycerol. 63

glycoprotein a protein which has an attached monosaccharide or oligosaccharide. 60

glycosphingolipid a glycolipid based on the amino alcohol sphingosphine. 63

glycosyltransferase enzyme a type of enzyme which forms glycosidic bonds between a sugar and an acceptor. 60

gradient descent an algorithm that minimises a function by moving parameters in the direction with the highest negative gradient . 67

half map a three-dimensional volume reconstructed from one half of the isolated and imaged particles in single particle analysis. 39

hidden layer a set of hidden units that is between the input and output layers of a neural network. 71

hidden units a regression model that is in a hidden layer of a neural network. 71

hydrophobic effect the tendency for hydrophobic groups to move toward other hydrophobic groups. 53

in silico experiments which are performed on a computer. 54

inference the process of predicting a value or classification using a machine learning model. 68

kernel in convolutional neural networks, a kernel is a small set of weights that the input is convolved with. 73

ketose a linear sugar chain capped with a ketone group. 56

learn in the context of machine learning, learn refers to the process where a model improves its performance iteratively. 67

learning rate a parameter that determines how much the optimisation algorithm updates at each step . 67

likelihood in statistics, likelihood measures the probability of an output given a specific input . 69

linear regression model a statistical model which attempts to model the data as a linear function. 66

local resolution a measure of an approximate resolution for a given area, commonly used in cryogenic electron microscopy analyses. 39

logistic regression a statistical model that models the probability of some number of outcomes. 68

loss function a function that measures the difference between actual and predicted values, producing a loss value. 67

loss value a scalar metric that often describes the difference between actual and predicted values. 67

maximum likelihood a method for estimating parameters such that the probability of the desired output is maximised. 67

motion correction a computational correction of beam-induced motion calculated by realigning sequential images from an electron microscope. 38

neural network a collection of interconnected non-linear regression models . 70

nucleic acid biological molecules consisting of individual nucleotides, which are crucial for almost all biological processes.. 41

nuclein a mixture of proteins and nucleic acids discovered by Friedrich Miescher. 41

nucleoside a precursor to a nucleotide consisting of a base group and sugar ring, which becomes esterified to form a nucleotide. 41

nucleotide the primary building block of nucleic acid consisting of a base group, sugar ring and phosphate group.. 41

oligosaccharide a polymeric carbohydrate consisting of multiple sugar monomers. 60

one-hot encoding a method to transform a category into a vector representation. 70

overfit an error in a machine learning that occurs when the model accounts for noise in the training data and is not able to generalise. 75

padding the addition of extra data points around the boundaries of the input to ensure consistent sizing through a convolutional neural network . 73

parameters a scalar variable that represents a relationship in data. 67

PNGase F an enzyme used to cleave the first NAG-ASN linkage of a glycan. 65

principal component analysis a dimensionality reduction method that simplifies high dimensional data into two or three dimensions. 78

probe helix a binding motif which uses an alpha helix to probe for double stranded DNA. 51

pucker the deviation of a ring-atom from the median plane of the ring. 45

pyranose a six-membered cyclic hemiacetal sugar.. 57

reductase a type of enzyme which reduces a chemical group, such as the reduction of an OH group into a H. 59

redundant a term used to describe a dataset which has multiple similar copies of information. 72

sequon a set of consecutive amino acids which form the basis of a recognition motif. 61

shallow neural networks a neural network with one hidden layer. 73

shallow U-Net A U-Net convolutional neural network that does not have a bottleneck portion as a linear vector representation. 119

single particle analysis a technique used in cryogenic electron microscopy to reconstruct a three-dimensional volume from a series of two-dimensional particle images. 38

softmax a function which converts a set of real numbers into a probability distribution. 70

sparse connectivity a connection strategy where each unit in a neural network is not connected to all other units. 73

standard logistic function a smooth S-shaped function commonly used in logistic regression, also known as the sigmoid function . 68

stride the distance between each consecutive convolution in a convolutional neural network . 73

supervised learning category of machine learning where data is labelled. 66

syn conformation an uncommon conformation of a nucleotide where the base is positioned toward the furanose ring. 46

training dataset a collection of examples that a model uses to optimise parameters . 67

transcription a process which makes a copy of a DNA strand with complementary RNA. 49

transferase a type of enzyme which transfers a chemical group, such as the transfer of a carbohydrate onto another carbohydrate. 61

translation a process which takes transcribed RNA to form a protein using t-RNA. 49

unsupervised learning category of machine learning where data is unlabelled. 66

weight matrix a matrix of parameters for each regression model inside a neural network. 72

Chapter 1

Introduction

Proteins are the building blocks of life and are critical for the survival of nearly all organisms. It is well established that the functionality and activity of a protein are directly related to the three-dimensional structure of the macromolecule.[1,2] In cells, proteins commonly interact with other proteins, nucleic acids, small molecules, lipids, carbohydrates, and ions to complete necessary functions.[3] Disruptions in these intricate intermolecular functions are the basis of many diseases, which can cause life-altering symptoms.[4,5] With such severe consequences, structural information must be obtained to afford essential insights into mechanistic causes and potential treatments.[6]

X-ray crystallography has been an invaluable technique for the structural analysis of macromolecules since the mid-20th century.[7] More recently, techniques based on electron microscopy have gained traction as a complementary method to X-ray crystallography to elucidate the structures of larger and more complex biological molecules, assemblies and molecular machines.[8]

1.1 X-ray Crystallography

1.1.1 Background

X-ray crystallography is a method used to obtain structural information from crystals. This technique can be used to determine the structures of small molecules, such as simple salts, and large macromolecules, such as proteins and nucleic acids. Before X-ray crystallography, visible-light microscopy was a standard method for determining structural information of small targets.[9] The principles behind visible-light microscopy are similar to those behind X-ray crystallography. With visible-light microscopy, visible light waves are created and scattered off a target with a particular *intensity* and *phase*. These waves are then refracted using a lens to create an image on a surface. Attempts to gain structural information of macromolecular targets using visible-light microscopy methods will fail due to the fundamental size limitation of visible-light-based techniques. Individual macromolecules are commonly orders of magnitude smaller than the shortest wavelength of visible light, therefore, it is necessary to study such small targets using shorter wavelengths of light, such as X-rays. The wavelength of X-rays used in X-ray crystallography is approximately on the order of a single covalent bond. With X-ray crystallography, these X-rays can be created and scattered off a crystal in an attempt to gain information about the target. In contrast to visible light in visible-light microscopy, these scattered X-rays cannot be refocused with a conventional lens to obtain an image on a surface. The scattered X-rays are measured directly to get an X-ray scattering pattern, also known as an X-ray diffraction pattern. The diffraction pattern, shown in Figure 1.1, provides invaluable detail regarding the structure of the target molecule and has formed the basis of structural studies into small molecules and macromolecules over the last century.[7,10,11]

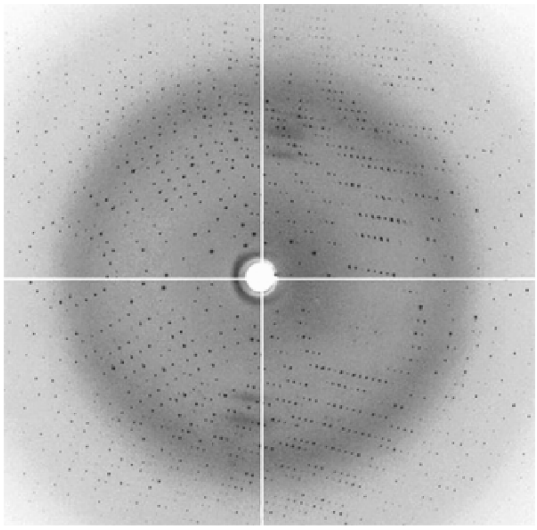


Figure 1.1: An X-ray diffraction pattern of a protein crystal reproduced with permission from Chen (2002).[12] Each black dot corresponds to a recorded reflection.

1.1.2 Scattering

A range of chemical characterisation techniques utilises the principles of light scattering to provide structural information. In the case of X-ray diffraction, X-ray waves are directed toward a target to gather information about the chemical structure. The measurement of the elastic scattering of these incoming X-ray waves from electrons in the target offers this insight. Scattering an X-ray wave from a single atom or even molecule alone would be unlikely to be reliably measurable, but measurements from many repetitions of a target molecule, like that in a crystal, improve the signal-to-noise ratio enough to often produce measurable data.

The interatomic arrangement of a crystal can be described well by a series of parallel, repeating planes, known as lattice planes. When an incoming X-ray wave hits a specific lattice plane at some angle, θ, a proportion of the incoming light will be reflected, with the remainder travelling further into the crystal lattice. This process continues for many lattice planes in the crystal, but not all diffracted waves are directly measurable. Since many waves will be travelling simultaneously, the interference between these waves may lead to destructive or constructive interference. Only when the waves exhibit *phase coherence* will a diffraction spot be measurable on some detector. This behaviour occurs according to Bragg's Law,[13] described in Equation 1.1 where d is the crystal plane spacing, θ is the angle of incidence, λ is the wavelength, and n is the layer of crystal on which the reflection is occurring. The orientation of these measurable planes is identified by integer Miller indices, h, k, l, which describe the fractional interception of these imaginary planes about the axes of the crystals. The intensity of the measured reflection for each Miller index provides information about the relative position of atoms with respect to the lattice plane. In a randomly oriented crystal, few lattice planes will be positioned correctly to satisfy Bragg's law, which limits the amount of information that can be collected. To address this, the crystal is often rotated to collect more data, thereby providing more information about the target chemical.

$$2d \sin\theta = n\lambda \tag{1.1}$$

1.1.3 X-ray Sources

Standard X-ray diffraction methods rely on a monochromatic source of X-ray radiation, with the behaviour of the beam having a strong influence on the quality of the measured signal.[14] Early sources of X-rays for X-ray diffraction utilised X-ray tubes that accelerated electrons from a cathode toward a metal anode, producing X-rays in turn.[15] While helpful in generating X-rays, these X-ray tubes often suffered from poor light output and were prone to rapid damage. Later, X-ray sources with rotating anodes increased the reliability

and quality of X-ray generation by dissipating heat more effectively. Both of these sources are common in many laboratories and are generally referred to as *home sources*. In many modern experiments, to obtain the highest-quality experimental data, it is desirable to use a higher-intensity source of X-rays, which can be generated at large national facilities called *synchrotrons*. These facilities work by accelerating electrons in a large ring and then forcing the electrons to bend around corners using a powerful magnet. This bending process releases intense tangential X-rays, which can be used for X-ray diffraction and other X-ray-related scientific studies. Synchrotron facilities have been the cornerstone of macromolecular crystallography for the last few decades, however, a new generation of X-ray sources known as free-electron lasers produce even more intense X-rays and can improve speed and quality of data collection for specific systems.[16]

1.1.4 X-ray Detectors

Detecting the diffracted X-ray waves is perhaps one of the most critical parts of X-ray crystallography. In the early years of crystallography, X-rays were detected using photographic films, which were wrapped around samples to collect a large number of reflections.[17] While recording diffraction spots on a single photographic film is helpful when identifying crystalline materials, recording multiple crystal rotations on a single film is impractical due to the likely overlap of diffraction spots. Commonly, many different photographic films were used over the course of an X-ray diffraction experiment, which increased the time and effort required to measure and analyse all of the diffraction spots. Data collection using photographic film has largely fallen out of use due to the advancements in computerisation of detectors. By the 1960s, X-ray diffraction experiments with some element of computer control were commonplace, with the photographic film being replaced with a *scintillation counter*.[18] This electronically controlled device was capable of measuring a single diffraction spot at a time and was therefore rotated to collect the maximum amount of available diffraction data. Advancements in detector technology allowed for the collection of data over a larger area, known as area detectors, which increase the throughput of X-ray data collection.[19] Modern detectors include photon-counting detectors, which can measure X-rays with reduced noise and high frame rates, enabling the determination of target structures with increasing accuracy.[20]

1.1.5 Data Processing

The interaction of the incoming X-ray waves with all atoms in the target dictates the diffraction pattern collected during an X-ray diffraction experiment. Each spot, or reflection, observed on the diffraction pattern can be described by a term known as the *structure factor*, F_{hkl}, which is used to describe the amplitude and phase of an X-ray wave that is diffracted off a specific lattice plane (h, k, l). The intensity of each reflection,

I_{hkl} can be measured by a detector and is proportional to the structure factor as shown in Equation 1.2.

$$I_{hkl} \propto (F_{hkl})^2 \tag{1.2}$$

Since diffracted waves must exhibit phase coherence to be visible on a given detector, the diffraction pattern describes the spatial frequency components of a three-dimensional Fourier transform of the electron distribution in the target. The reconstruction of the electron distribution, commonly referred to as *electron density*, can be theoretically calculated through the inverse Fourier transform of the diffraction pattern, shown in Equation 1.3. Each point in the electron density has a contribution from the amplitude, $|F_{hkl}|$, and phase, ϕ_{hkl}, of each reflection. The level of detail observable in the electron density map is measured by the *resolution* and controlled by the reflections measured in the experiment.

$$\rho(xyz) = \frac{1}{V} \sum_{hkl} |F_{hkl}| e^{-2\pi i[hx+ky+lz-\phi_{hkl}]} \tag{1.3}$$

where:

$\rho(xyz)$ is the electron density at a given point in the crystal xyz
$|F_{hkl}|$ is the structure factor amplitude for a given reflection hkl
ϕ_{hkl} is the structure factor phase for a given reflection hkl

Unfortunately, the electron density cannot be computed directly from the X-ray diffraction experiment, since the phase of each reflection cannot be measured. This fundamental limitation of X-ray crystallography is known as the *phase problem*, however, a variety of techniques have been developed to estimate or calculate the phase associated with each reflection.

1.1.5.1 Patterson Methods

One of the earliest solutions to the phase problem was to eliminate the dependence on the phase entirely. By performing the calculation of vector-space density using the square of the structure factor amplitudes and no phases, the resultant density describes all interatomic vectors within the target structure.[21] This map, known as a Patterson map, of N atoms contains $N(N-1)$ peaks, with the height of each peak proportional to the product of the number of electrons in the two atoms. This information can be used to determine the atomic structure in certain instances, but as the number of atoms in the structure increases, this method becomes increasingly complex. In macromolecular crystallography, determining the structure directly from the Patterson map is likely too complicated to be achievable. However, if the macromolecule contains atoms with a large number of electrons, known as heavy atoms, the interatomic vectors between these two heavy atoms may be used in tandem with other methods to obtain an estimate of the phase of each reflection.

1.1.5.2 Isomorphous Replacement

The first general solution to the phase problem in macromolecular crystallography was isomorphous replacement, which utilises diffraction data from multiple crystals to constrain the phase estimation for each reflection. Diffraction data can be collected for a crystal target of interest, known as the native crystal, which can be compared to another which has been soaked in a solution of heavy atoms, known as the heavy-atom derivatised crystal. Assuming that the target structure remains unchanged when heavy atoms are incorporated, comparing the diffraction patterns using Patterson methods is likely to provide information about the locations of the added heavy atoms. Using these positions, in conjunction with data from the native and heavy-atom derivatised crystals, allows for an estimation of the phases of each reflection. Using a single heavy-atom derivatised dataset, referred to as single isomorphous replacement, results in phase estimations with large errors. Therefore, multiple heavy-atom derivatised datasets, referred to as multiple isomorphous replacement, are typically used to provide a more accurate phase estimation. While isomorphous replacement may be effective, the addition of heavy atoms can affect the target structure, so it should be treated with caution during interpretation.[22]

1.1.5.3 Anomalous Dispersion

When X-rays are applied to a crystal, most atoms diffract the incoming X-ray beam elastically. If the incoming X-ray has an energy that is close to the inner-shell electron binding energy of an atom, known as an absorption edge, the atom may absorb a small amount of energy. If this occurs, the diffracted wave will exhibit a different phase relative to the incoming beam, known as *anomalous dispersion*. In routine X-ray diffraction experiments, the X-ray energy is not close to the absorption edges of the most common atomic components of biological macromolecules, resulting in minimal anomalous scattering. Heavier atoms, such as selenium or zinc, do have absorption edges within the standard energy ranges of synchrotron X-ray radiation.[22]

Anomalous dispersion can be used to estimate the phases of each reflection, since the dispersion effect disrupts a general rule present in the diffraction pattern known as Friedel's law.[23] This law states that reflections $F_{h,k,l}$ and $F_{-h,-k,-l}$ must have the same magnitude and opposite phases, but in the presence of anomalous dispersion, this law is broken. The difference between $F_{h,k,l}$ and $F_{-h,-k,-l}$ can be used to estimate the positions of the heavy atoms, similarly to isomorphous replacement. Two popular methods of exploiting this property of heavy atoms are single-wavelength anomalous dispersion (SAD) and multi-wavelength anomalous dispersion (MAD).[24,25] The phase estimates obtained from SAD are often more ambiguous than those of MAD and likely require enhancement from downstream methods. In contrast, MAD may provide better phase estimates, but by irradiating a target crystal multiple times, the likelihood of damage increases.

1.1.5.4 Molecular Replacement

Methods to obtain phase estimates that rely on physical principles of diffraction are abundantly capable and were the *de facto* method of choice for a considerable period of crystallographic history. These methods are commonly referred to as *experimental phasing*, but have largely been replaced with methods which rely on obtaining phase estimates from another atomic model, known as molecular replacement.[26] The fundamental idea behind molecular replacement is that structures that have already been solved and are similar to the target of interest may serve as a good starting point for estimating phases. Molecular replacement is currently the most common method for phase estimation in X-ray crystallography, with popularity only likely to grow as more structures are solved and become available for use in molecular replacement.[27,28]

For molecular replacement to provide a reasonable estimate of the phases of each reflection, it is essential to select a suitable template model. The most appropriate model for molecular replacement is the structure that is most similar to the structure under investigation. Similarity is generally assumed when two structures have a high sequence identity, although this is not necessarily guaranteed. Sequentially similar proteins may undergo substantial conformational changes in secondary or tertiary structure, which can pose significant challenges when performing molecular replacement. Whilst biochemically related structures are obvious candidates as templates for molecular replacement, the use of *in silico* models has become increasingly popular.[29]

Assuming a template structure is similar to that of the structure present in the crystal, the template can be used to estimate the phases for each reflection measured in the diffraction pattern of the unknown structure. Estimating the phases of each reflection directly from the template structure should be completed with care, since the orientation and translation of that template structure is likely not the same as that of the unknown structure. Molecular replacement methods redefine the orientation and translation of the template structure such that a calculated diffraction pattern from the template structure best matches the observed diffraction pattern from the experiment. This approach of computationally optimising the position of the target can be represented by six parameters, three defining the orientation and three defining the position of the target. Optimising all six parameters in a six-dimensional search is likely to be computationally prohibitive, therefore, the placement of the target structure is commonly split into an initial rotation search, followed by a translational search once a sufficient estimate of the rotation is obtained. Computing an appropriate rotation and then translation for the template structure is commonly achieved through Patterson methods[30] or probabilistic methods.[31] Once the rotationally and translationally optimised template structure has been placed, the phases calculated from the template can be used in conjunction with

the amplitudes derived from experimental data to begin downstream processing.

1.1.5.5 *In silico* Structure Prediction

The ability to understand structural information of a biological molecule without significant physical experimentation is an attractive prospect in almost all areas of biochemistry. Methods which attempt to enable this have been developed over time, with the most successful of approaches relying on the application of machine learning methods. Initial machine learning methods tried to predict which secondary structure an input sequence would adopt, and achieved a reported accuracy around 60 % in this task.[32] Secondary structure prediction was later improved further after supplying the neural network with data from homologous models of the target sequence. The intuition behind this method was rooted in the fact that protein secondary structure is relatively more conserved than the protein sequence.[33] As a result, using the probability of each amino acid appearing at a specific position in the sequence, relative to other homologous structures, enables the capture and encoding of evolutionary information. This technique is known as a multiple sequence alignment, and when combined with a neural network, the reported accuracy of secondary structure prediction increased to around 70 %.[34]

While gaining information about the secondary structure from a sequence is significantly helpful, understanding the entire structure from a sequence, or a set of sequences, alone is likely the end goal of most *in silico* structure determination methods. Progress on this goal dramatically increased upon the use of *coevolutionary methods*. While multiple sequence alignments encode information about local-scale inter-amino acid interactions, coevolutionary methods attempt to explore other functionally and structurally dependent areas of the protein.[35] Co-evolutionary methods were applied to protein structure prediction, significantly accelerating the field and culminating in the release of the highly successful *AlphaFold2*.[36] *AlphaFold2* attempts to consider both local and longer-range interactions to form a suggested protein model with high accuracy in many cases. Unlike some other methods, *AlphaFold2* does not impose any strict restraints based on known protein biophysics, instead, it encodes these geometrical restraints within the neural networks implicitly. Following the success of *AlphaFold2*, numerous powerful neural network-based protein prediction software packages have emerged, which approach or surpass the accuracy of *AlphaFold2* in some cases.[37,38] Understanding the structure of proteins has been the primary focus of structure determination methods, however, with the power of neural network-based methods, methodological developments have been applied to other systems, including protein complexes,[39] nucleic acids,[40–42] and carbohydrates.[40]

In silico derived models often offer a good understanding of structural information for a given biological sequence. With most modern methods, almost all experimentally derived

structures in publicly and privately held datasets have been utilised in some way to enable the method to function. Resultantly, predictions are often most accurate when the sequence under investigation is similar to previously determined structural information. Conversely, if the amount of structural information on a given target sequence is scarce, then predictions are unlikely to hold a high degree of accuracy. The *Protein Data Bank (PDB)*,[43] the largest public database of macromolecular structures, contains the most information for proteins, and so protein predictions are most often of acceptable accuracy. Much less structural information is known about nucleic acids, forcing even the best *in silico* prediction software package to struggle with these sequences.[44]

1.1.6 Manual Model Building

After the estimation of the phase of each reflection, an electron density map can be calculated, as shown in Equation 1.3. In theory, this electron density map outlines the distribution of electrons within the target crystal. Using this information, an atomic model which explains this distribution can be created. Generally, it is advisable to visualise the electron density in three-dimensional graphics software such as *O*[45] or more recently *Coot*.[46] Graphical software packages often allow for an atomic model to be manually created, in a process referred to as manual model building.

Manual model building aims to model atoms, residues or chains as they were found in the crystal used in the X-ray diffraction experiment.[47] The electron density map, as well as sequence data, are often required to accurately form a realistic and complete atomic model of the target structure. Depending on the resolution of the data and method used to phase the reflections used to create the electron density, this process of manual model building can range in difficulty from trivially straightforward to impossibly difficult, and often requires a significant time investment. A multitude of tools within the most commonly used software package, *Coot*, aim to facilitate the interpretation and manipulation of atomic models for proteins, nucleic acids, carbohydrates, and ligands.[48–52]

1.1.7 Automated Model Building

Manual model building is an essential step for structure solution, but is often regarded as a significant bottleneck in the process. Given advancements in computing hardware, attempts to automate this process of model building in some way were a vital research goal in the field. The underlying principle behind all model building requires a software method to consider what the electron density represents, whether that be an amino acid, nucleotide, carbohydrate, ligand, or solvent. Some of the earliest approaches to automating protein model building attempted to understand the electron density map by first *skeletonising* it to reveal a path of main-chain atoms.[53] While this method ultimately

fell short of fully automated protein model building, it provided the groundwork for a range of other software methods to attempt to find the Cα atoms of amino acids in the experimental density. One of the first reportedly successful methods of automated atomic position assignment was in the software package *FRODO*, which used precomputed Cα positions and a library of amino acid fragments to automatically select the best-fitting residue.[54] Similar techniques were introduced into the *TEXTAL* software package[55,56] and formed the foundation of many graphical software packages which included automated model-building capabilities, such as *O*,[57] *MAIN*,[58] *QUANTA*[59] and *XTALVIEW*.[60]

The majority of these methods are now mainly of historical significance due to the advancements and success of other software packages for automated model building, such as *Coot* and suites like *CCP4*[61] and *Phenix*,[62] which contain a variety of automated model-building software packages. While legacy software packages are of little modern utility, the availability of such methods certainly facilitated the completion of a large number of macromolecular structures and should not be overlooked. In recent times, automated model building is most commonly completed using software like *Buccaneer*,[63] *Nautilus*,[64] *Phenix Autobuild*[65] or *ARP/wARP*.[66]

1.1.7.1 Nautilus

Nautilus is a popular software package for the rapid and automated model building of nucleic acids.[64] *Nautilus* was designed with a focus on speed to allow compatibility with graphical software packages like *Coot*. Similar to other automated software packages, *Nautilus* attempts to identify which areas of density can be attributed to nucleotides, followed by growing initial positions to form a chain. Careful and considerate optimisations in both the searching and scoring functions often allows *Nautilus* to achieve sub-second nucleic acid chain building.

Locating the likely position of atomic fragments in electron density often requires a six-dimensional search considering both translation and rotation. Much like six-dimensional search approaches in molecular replacement (see Section 1.1.5.4), this approach is unlikely to be computationally efficient. The key to obtaining rapid performance is to minimise the number of computational operations which are necessary at each point in the electron density. *Nautilus* employs a technique that relies on sampling the electron density at a series of probe points, with the expectation that some will contain high-density values and others will contain low-density values if a correctly positioned fragment is at a given location. These probe points, also referred to as *fingerprints*, are shown in Figure 1.2, with spheres representing points expected to be positive and crosses representing points expected to be negative. Each potential position and rotation is scored using an efficient but crude method, which rejects fragments as soon as possible, providing a fast

determination of whether a given position in the electron density map is likely to contain either a sugar or a phosphate group of a nucleotide. The initial list of positions can then be refined using a more accurate scoring method to yield a set of candidate positions.

Following the selection of candidate positions, sugar candidates are converted into mononucleotides, and phosphate candidates are converted into dinucleotides. To determine what geometry of nucleotide best fits the electron density, fragments from a precomputed library are trialled. *Nautilus* uses a library calculated from the crystal structure of a *Tetrahymena thermophila* group I intron resolved at 2.25 Å (PDB code: 1HR2[67]). These initial mononucleotide or dinucleotide fragments are iteratively extended in both the 3' and 5' directions until no further nucleotides can be fit into the electron density. This process is likely to result in some overlapping nucleotides, as candidate positions are often found within the same nucleic acid chain. Overlapping fragments are removed,[63] and any other clashes are resolved, producing an atomic model of the nucleic acids which can be recycled into the procedure for three cycles.

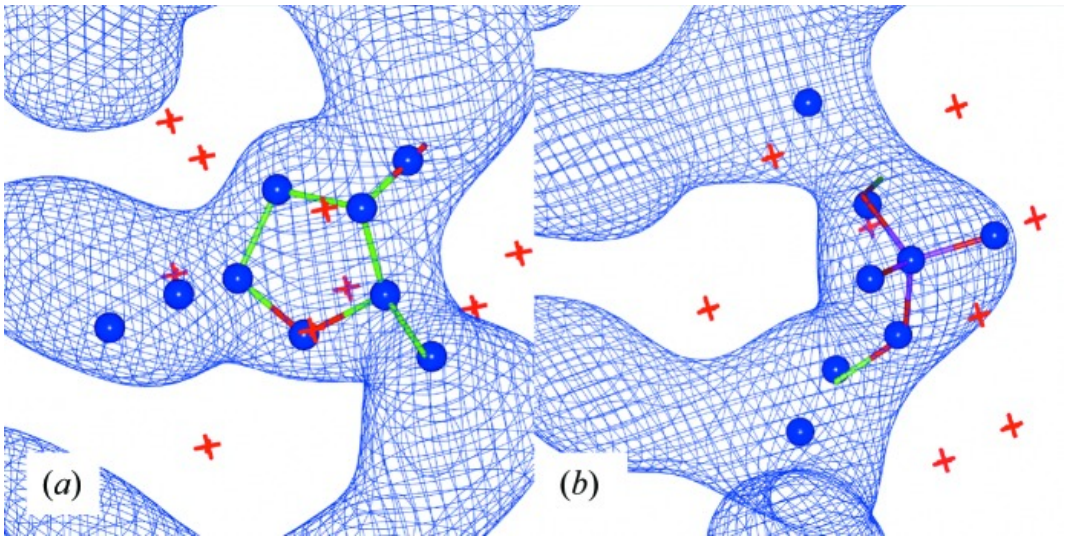


Figure 1.2: Probe points used in the software package *Nautilus* for determining whether a specific grid point position is likely to contain either a sugar (a) or phosphate (b) position. The spheres represent points that are expected to be high, and the crosses represent points that are expected to be low. Reproduced with permission from Cowtan (2014).[64]

1.1.7.2 ARP/wARP

The automated model-building suite *ARP/wARP* is capable of modelling both proteins and nucleic acids with reasonable accuracy. The initial aim of *ARP/wARP* was to improve the estimates of the phases of each reflection in the X-ray diffraction pattern by placing atomic positions in areas of high local density.[68] These atoms, known as free atoms or dummy atoms, aim to reduce the error associated with each phase by increasing the completeness of the atomic model. *ARP/wARP* attempts to build protein structures

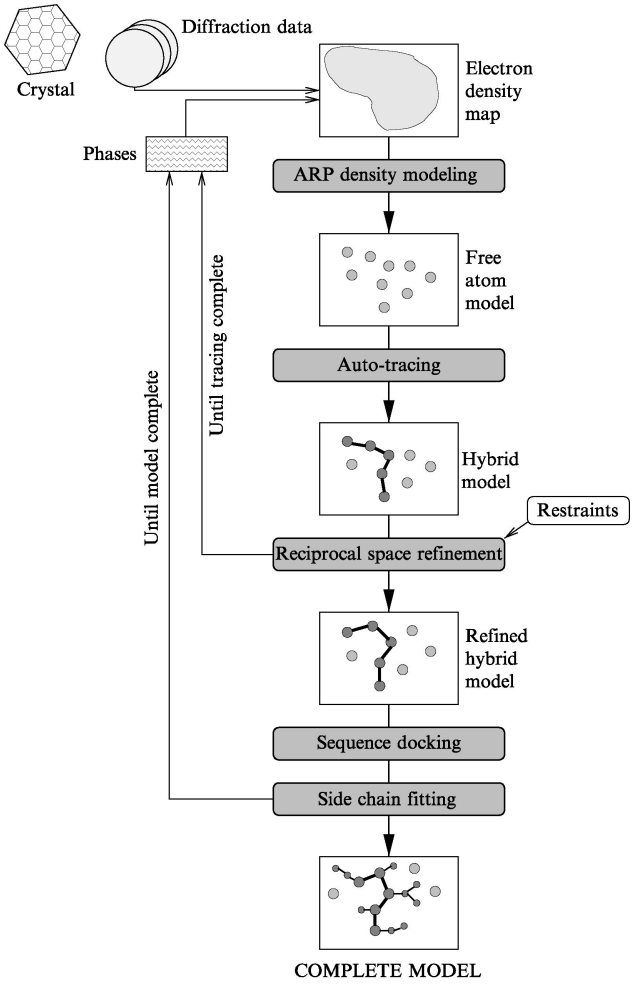


Figure 1.3: Flowchart of *ARP/wARP* software package reproduced with permission from Morris (2003).[69] A free atom model is generated from the electron density map before a backbone is traced and refined.

by tracing a backbone through these free-atom positions, where the backbone follows standard protein Cα geometric expectations.[69] This often results in a peptide backbone surrounded by free atoms, known as the hybrid model. External refinement programs then attempt to refine the atomic model to resemble the experimental data better, and to improve the estimation of each phase (see Section 1.1.8). This process of free-atom backbone tracing and refinement continues cyclically, followed by sequential assignment of each partial amino acid. The most probable amino acid type is then modelled in one of a set number of known side chain conformations.[70] This process of automated model building is shown in Figure 1.3. Since the initial version of *ARP/wARP*, which was built to model proteins exclusively, new capabilities have been added to incorporate non-crystallographic symmetry information into model building,[71] to apply these methods to lower-resolution information,[72] and to model other types of molecules like ligands[73] and nucleic acids.[74] *ARP/wARP* attempts to automatically model nucleic acids fundamentally differently from that of *Nautilus*. While *Nautilus* attempts to locate areas of electron density corresponding to sugars or phosphates, *ARP/wARP* instead looks for

planar regions of density corresponding to the nitrogenous base groups of nucleotides. For any spherical region of density, the variation in electron density can be measured and classified against precomputed expectations for either a small single-ring structure, a large single-ring structure, a double-ring structure or a no-ring structure. This classification enables *ARP/wARP* to identify potential nucleic acid sites within the density for use in subsequent model-building steps.

1.1.7.3 Phenix Autobuild

Another popular automated model-building package is *Phenix Autobuild*, which is capable of modelling proteins and nucleic acids, and ligands from X-ray diffraction data. Model building is carried out using the *RESOLVE* program,[75] which was created to improve the quality of electron density maps following experimental phasing. Later additions to *RESOLVE* allowed it to model protein main-chains automatically.[76–78]

To model proteins, *RESOLVE* employs a similar template approach to *Nautilus*, positioning larger-scale features after searching through a library of features. *RESOLVE* attempts to locate α-helices and β-sheets using a Fast Fourier Transform convolution method,[79] followed by positioning of the best fitting library fragment scored by the mean density at the main-chain atomic positions. The highest-scoring fragment continues and is extended in both directions to continue building the chain. Once a range of chains has been proposed, any clashing areas are pruned, and the longest chain is kept as the theoretical structure. The identity of side-chains is then scored by comparing each position with a precomputed template to yield a probability for each side-chain.

Whilst the majority of tools were initially developed exclusively for proteins, later editions added capabilities to model nucleic acids into electron density. *Phenix Autobuild* attempts to model nucleic acids by identifying phosphate and base positions, as well as the helical nature of nucleic acids. Using a precomputed library of RNA backbones, defined by the RNA Ontology Consortium,[80] nucleic acids can be modelled in a similar way to that of proteins. This fragment library clusters representative nucleic acids from ≤ 3 Å resolution structures deposited in the Nucleic Acid Database as of February 2005.[81] Nucleic acids were clustered based on the backbone torsion angles between sugar-sugar units known as *suites*, yielding 46 distinct conformational clusters.

After initial modelling, the torsion angles in nucleic acids are refined to ensure a good fit to the density, while conserving geometric reason. The *Phenix Autobuild* package utilises cycles of model building and refinement to attempt to produce a complete atomic model with minimal user input.

1.1.8 Refinement

Once the atomic positions have been modelled, either entirely or partially, the atomic model should be refined to create a better agreement between the observed experimental data and the model, and to ensure that the model is chemically sensible. By refining the atomic model to better align with the experiment, a more accurate estimate of the phase of each reflection is likely to be obtained. With better phase estimates, a potentially more accurate electron density map can be created, enabling a critical assessment of the atomic model before making any structural inferences. Popular software packages for macromolecular crystallographic refinement include *REFMAC5*,[82] *Servalcat*,[83] and *phenix.refine*.[62]

These popular methods refine the atomic model to the experimental data using the *maximum likelihood* method. To best align the calculated structure factors, F_{calc}, with the observed structure factors, F_{obs}, it would be natural to attempt to minimise the square difference of the two values, known as least-squares refinement. In practice, since experimental data often contains measurement errors and the molecular model has non-normally distributed errors, least-squares refinement is unlikely to produce an accurate and reliable result. A more effective approach is to estimate the likelihood of observing the experimental diffraction data from the current model. Refinement programs aim to find an atomic model which maximises this likelihood, which should produce a good agreement between the experimental data and the atomic model.

It is often helpful to calculate a parameter which quantifies whether a model agrees with the experimental data. The most commonly used parameter for this purpose is the crystallographic R-factor, shown in Equation (1.4), which assesses the fit of observed structure factors to the calculated structure factors calculated from the atomic model. Refining the atomic model to better align with experimental observations is likely to lower the R-factor, but may also introduce bias in the atomic model by fitting to noise. To alleviate this problem, a subset of reflections from the X-ray diffraction pattern are reserved from refinement and used to calculate an R-factor, known as R_{free}. Typically 5 % of the reflections are omitted for this purpose, and R_{free} has become a standard parameter used to monitor refinement progress.

$$R = \frac{\sum ||F_{obs}| - |F_{calc}||}{\sum |F_{obs}|} \tag{1.4}$$

Refining the atomic coordinates in the model to better align with experimental observations is certainly possible, but without careful consideration, the chemical feasibility of the atomic model is unlikely to be preserved. To prevent this, geometric restraints are often introduced in refinement to ensure the model retains appropriate chemistry. Geometric constraints can be obtained from sources of high-quality structural information,

such as the Cambridge Structural Database,[84] *PDB Chemical Component Dictionary (CCD)*,[85] or CCP4 Monomer Library.[61] After refinement is complete, the new atomic model can be used to create a potentially more accurate electron density map for further model building, critical assessment, and validation.

1.1.9 Validation

Once a target structure has been modelled and refined to a reasonable degree, the structure must be validated to identify any potential sources of error. Standard validation metrics calculate geometric properties of the atomic model and ensure they match expected values in precomputed databases. For protein models, torsion angles in the peptide backbone can be compared to expectations in a Ramachandran plot.[86] Most of the early validation software packages focus exclusively on proteins, such as *PROCHECK*.[87] More modern software packages, such as *MolProbity*,[88] enable the assessment of both main-chain and side-chain atoms in proteins and nucleic acids. For specialised validation of nucleic acids, *DNATCO*[89] and *DoubleHelix*[90] aim to validate the conformation and base pairing of nucleotides. The validation of carbohydrates is commonly carried out using *Privateer*,[91] and ligands can be validated using *Coot*.[50]

Geometric validation is crucial to ensure that the atomic model is chemically reasonable, however, it is also imperative that the atomic model can be explained by experimental data. The most common way to measure this is through the *Real Space Correlation Coefficient (RSCC)*, which measures the agreement between the experimental electron density and a calculated electron density for any number of atoms.[45] RSCC is defined in Equation (1.5) and has a codomain of $[-1, 1]$. High RSCC values indicate that the model is well-supported by the experimental electron density, and conversely, low RSCC values suggest issues with the atomic model.

$$\mathrm{RSCC} = \frac{\sum(\rho_{\mathrm{obs}} - \langle\rho_{\mathrm{obs}}\rangle)(\rho_{\mathrm{calc}} - \langle\rho_{\mathrm{calc}}\rangle)}{\sqrt{\sum(\rho_{\mathrm{obs}} - \langle\rho_{\mathrm{obs}}\rangle)^2 \sum(\rho_{\mathrm{calc}} - \langle\rho_{\mathrm{calc}}\rangle)^2}} \tag{1.5}$$

1.2 Cryogenic Electron Microscopy

1.2.1 Background

Electron microscopy encompasses a range of techniques used to obtain structural information about biological samples. Principally, these techniques are similar to visible-light microscopy, where a sample is irradiated and the scattered radiation is collected. In electron microscopy techniques, this source of radiation is not visible light but electrons fired at a sample, which pass through and are collected to realise near-atomic-level detail. The wavelength of electrons commonly used in these transmission electron microscopes is around 0.02 Å,[92] which is much smaller than visible light and allows for high-resolution investigation. The theoretical power of electrons as a tool is somewhat limited by the effects electrons have on biological material, with a high likelihood of covalent bond disruption and free radical generation leading to conformational changes in the biological sample.[93] The primary strategy to minimise this issue is to image a sample at cryogenic temperatures, which has been shown to reduce the amount of observable damage compared to ambient conditions.[94] Samples can be rapidly cryo-cooled to encapsulate the sample in a layer of amorphous ice,[95] which reduces the spread of electron-induced damage.[96] This cryogenic electron microscopy (cryo-EM) technique generally preserves the native structure of the sample, but sufficient doses of radiation are still likely to cause damage.

Imaging in cryo-EM yields a two-dimensional representation of the biological sample, commonly displayed in the form of a micrograph shown in Figure 1.4. A single image can encompass many biological particles, and various techniques can be employed to extract useful information from a set of images. The most common technique for near-atomic level information of biological samples is *single particle analysis*, which aggregates many two-dimensional projections of the sample to form a three-dimensional reconstruction.

1.2.2 Data Processing

The process of obtaining a three-dimensional representation of a biological sample from many two-dimensional images in single particle analysis begins with correcting for physical effects associated with collecting an image from an electron microscope. As the electron beam hits the sample, particles in the beam experience *beam-induced motion* which can mask high-resolution detail for certain particles. The first common step of data processing, *motion correction*, aims to correct this movement in a series of images by realigning and aggregating individual particles.[98] The result, known as a micrograph, is a representation of the true sample, but the image is unlikely to be identical due to fundamental physical limitations of the electron microscope, which causes a spatially vari-

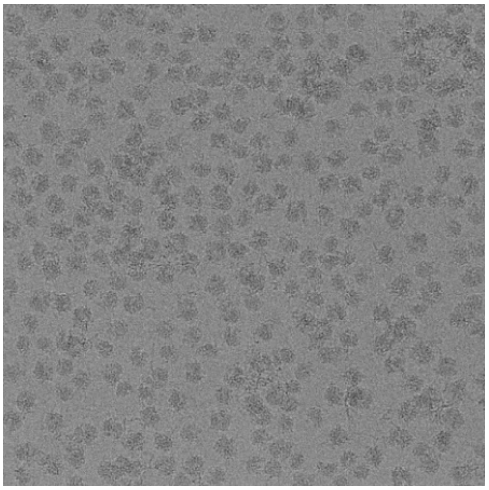


Figure 1.4: Cryogenic electron microscopy micrograph of ribosome particles. Reproduced with permission from Fromm *et al.* (2023).[97]

ant modulation of scattered electrons. This variation can be described by a sinusoidal contrast transfer function, which must be estimated during data processing to recover some of the lost information.[99]

The set of resultant micrographs, which contains many particles, may then be used to extract images of individual particles, known as particle picking. Manual, semi-automatic,[100] and fully automated[101] particle picking methods have been developed, which produce a set of particles that are then grouped into classes. This two-dimensional classification process aims to group similarly orientated and positioned particles, and can be averaged to enhance the level of observable detail.[102]

To obtain a three-dimensional volume associated with the biological sample, the picked particles must be reconstructed in three dimensions. The fundamental difficulty with this process is that the relative orientation of particles is unknown and must be established computationally. One common method is *projection mapping*, which compares the picked particle to a reference three-dimensional model.[102] Reconstruction algorithms commonly divide the dataset of particles into two halves, with each half reconstructed completely independently. The resultant two *half maps* can then be compared to assess uncertainty and global resolution, and are combined to produce a *full map* which approximately represents the Coulomb potential of the structure. Depending on the heterogeneity of the particles used in the reconstruction, the reconstructed map may not be equally well resolved in all areas.[103] The *local resolution* of a map can be calculated to assess these regions, which is often helpful in downstream processing.[104] Overall, this intricate data processing pipeline is a relatively manual process, commonly accomplished in software suites like *RELION*[105] or *cryoSPARC*.[106]

1.2.3 Model Building, Refinement, and Validation

After data processing, the three-dimensional Coulomb potential map represents the biological sample imaged in the electron microscope, but it is difficult to directly extract much biological meaning. In a manner analogous to X-ray diffraction, an atomic model must be produced which explains the observed experimental density. The methods used for this process are principally similar to those in X-ray diffraction, but are potentially more straightforward, given that the experimental data obtained in cryo-EM are similar to accurately phased X-ray diffraction data. Most software methods designed for X-ray diffraction also work well with cryo-EM data of sufficient resolution, but purpose-built cryo-EM exclusive automated methods often produce the best results.[107–109] In some cases, the map may need post-processing to aid the identification of structurally relevant features, such as amino acid side chains. Sharpening or blurring the reconstructed map can yield additional detail, and is most commonly used during interactive model building.[110] Refinement is also broadly similar to X-ray diffraction, with refinement programs optimising the model with respect to the experimental map while considering geometric restraints.

Validating an atomic model against cryo-EM data ensures that the model accurately represents the features in the reconstructed map. The overall model can be assessed using global metrics to summarise validity, such as the *Fourier Shell Correlation (FSC)*[111] or on an individual residue basis to identify specific problematic regions.[112] Metrics such as *RSCC*, commonly used in X-ray diffraction, have some utility in cryo-EM validation, but because the experimental data obtained by the two methods are not identical, other specialist metrics have been developed to assess atomic models critically.[113]

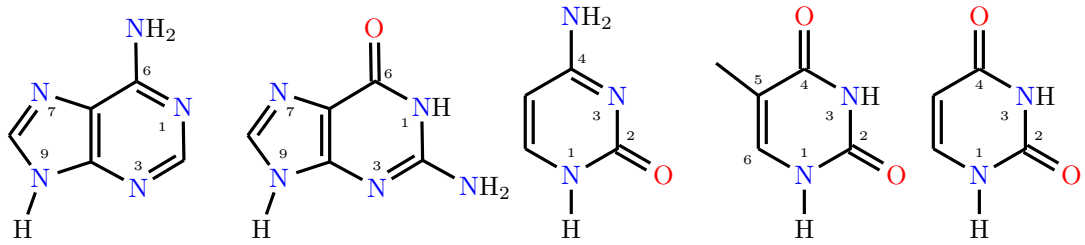
1.3 Nucleic Acids

Nucleic acids are one of the most essential components of life. They encode genetic information necessary to manufacture proteins and other macromolecules that enable cells to function. The discovery of nucleic acids dates back to 1869 when Friedrich Miescher crudely isolated a phosphorus-dense precipitate that was determined to originate only from the nucleus of cells and was named *nuclein*.[114] Initially, Miescher found nuclein to be isolatable only in complexes with other so-called contaminants, which are now known to have been proteins.[115] Miescher had unknowingly isolated the first group of protein-nucleic acid complexes. Efforts to further purify this novel compound, as isolated by Richard Altman in 1889, led to the discovery of a protein-free nuclein that Altman termed *nucleic acid*. At the time, little evidence for the function of nucleic acids was available, although there was private speculation by Miescher that these compounds could be the basis of heredity.[116] Elucidating the function of nucleic acids would be advanced dramatically by structural studies in the 20$^{\text{th}}$ century.

1.3.1 Nucleotide Structure

Two classes of nucleic acid are predominant in nature, deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). Both DNA and RNA are long linear polymers of monomeric units called *nucleotides*, which themselves are phosphate esters of *nucleosides*. The nucleotides in DNA and RNA share a similar chemical nature, consisting of a nitrogenous base group, a pentose sugar group, and a phosphate group. In DNA, the pentose sugar is 2'-deoxyribose, whereas in RNA, the pentose sugar is ribose. The difference between ribose and 2'-deoxyribose lies in the absence of a hydroxyl group at the carbon-2 position of the sugar ring. The vast majority of nitrogenous base groups in each nucleotide consist of either a monocyclic pyrimidine or a bicyclic purine group, although tricyclic groups have been noted in certain conditions.[117] There are five commonly observed nitrogenous bases in nucleic acids: two purines, adenine (A) and guanine (G) and three pyrimidines, cytosine (C), thymine (T) and uracil (U). Both DNA and RNA contain both purines, adenine and guanine, as well as the pyrimidine cytosine. RNA additionally contains the pyrimidine uracil, whereas DNA contains the closely related pyrimidine thymine. The five nitrogenous bases are shown in Figure 1.5. In a nucleotide, the base group is attached to the carbon-1 atom of the pentose sugar in furanose form. The pyrimidine bases attach to the furanose sugar through the nitrogen-1 atom, and the purine bases attach through the nitrogen-9 atom.

A nucleotide is formed from a nucleoside after one of the hydroxyl groups on the furanose sugar is esterified by a phosphate ester. In the simplest case, a nucleoside can be esterified by a phosphate monoester to form a nucleotide. Nucleosides can also be es-

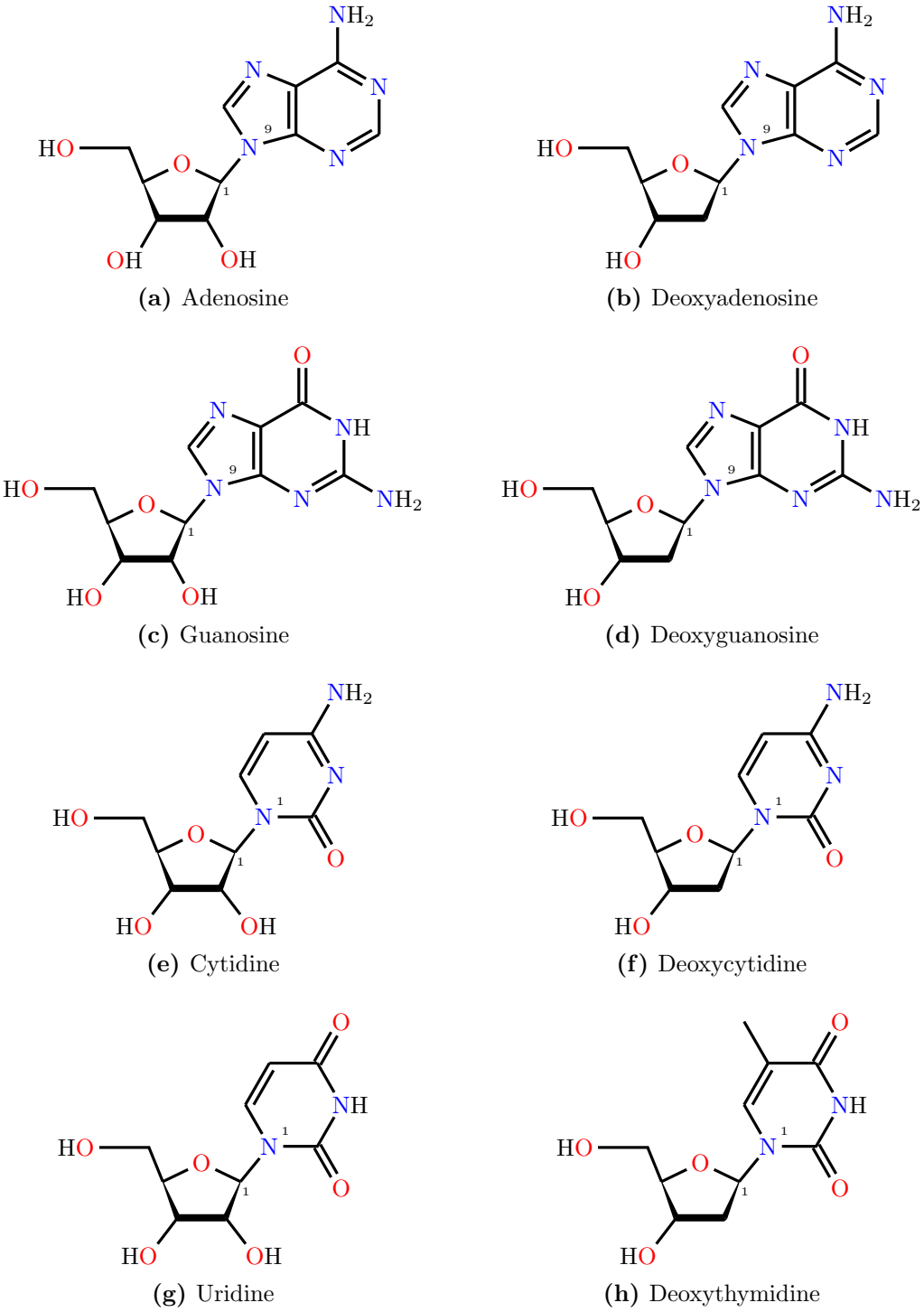


(a) Adenine **(b)** Guanine **(c)** Cytosine **(d)** Thymine **(e)** Uracil

Figure 1.5: Chemical structures of the five main nitrogenous bases found in nucleic acids. All nucleic acids are shown in their keto forms.

terified at multiple hydroxyl groups, forming more complex nucleoside phosphates. The structures of all eight common nucleosides from RNA and DNA are shown in Figure 1.6. Modified nucleotides can exist in nature in specific environments, such as bacteriophages, where a methoxyl or glucose group can be added to the cytosine base group.[118]

The phosphate esterification from a nucleoside to a nucleotide allows the nucleotide to become a monomeric unit in a much larger linear polymer. In nucleic acids, two nucleotides are linked together through a 3'-5' phosphate diester bond originating from the 3' and 5' hydroxyl groups. This strong phosphate ester enables nucleic acids to resist chemical hydrolysis, a crucial characteristic for molecules that encode genetic information.



(a) Adenosine **(b)** Deoxyadenosine

(c) Guanosine **(d)** Deoxyguanosine

(e) Cytidine **(f)** Deoxycytidine

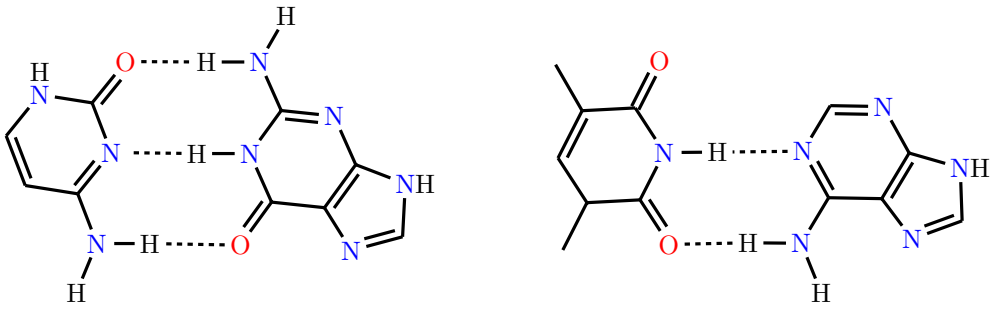
(g) Uridine **(h)** Deoxythymidine

Figure 1.6: Left: Chemical structures of the four nucleosides found in RNA. Right: Chemical structures of the four nucleosides found in DNA.

1.3.2 Three Dimensional Structure

After the primary structure of nucleic acids had been established, attention focused on the secondary structure of nucleic acids, and in particular, DNA. X-ray diffraction experiments played a crucial role in advancing the understanding of the higher-order structure of

DNA. In 1953, Francis Crick and James Watson proposed the double-stranded structure of DNA for the first time, after initial data was collected by Rosalind Franklin. Watson had deduced from X-ray diffraction experiments that the number of nucleotides in the crystallographic unit cell favoured a two-stranded structure. This, in addition to the symmetry of the diffraction pattern, suggested that DNA must contain two chains running in opposite directions. Inspired by the recent identification that pyrimidine-purine hydrogen bonds could form between nucleotide bases,[119] Watson paired adenine with thymine and cytosine with guanine. This complementary base pairing, later known as Watson-Crick base pairing, allowed the two-fold symmetry of the double-stranded DNA to be retained despite an irregular sequence of bases. The Watson-Crick base pairs are shown in Figure 1.7.



(a) Cytosine - Guanine **(b)** Thymine - Adenine

Figure 1.7: Watson-Crick base pairs between cytosine-guanine and thymine-adenine. Three hydrogen bonds stabilise the base pair in cytosine-guanine, and two hydrogen bonds stabilise the base pair in adenine-thymine.

This complementary pairing is fundamental for DNA transcription and translation and is underpinned by two (A-T) or three (C-G) hydrogen bonds. N-H groups are effective hydrogen bond donors, while electron-rich carbonyl oxygens and ring nitrogen atoms can accept hydrogen bonds well. The hydrogen bonds exist between 2.80 - 2.95 Å with a consistent glycosidic bond angle, which enables the nucleotide to have an isomorphous geometry with respect to other base pairs.[120] These characteristics allow all four base pair combinations for DNA (AT, TA, CG, GC) to exist within the DNA double helix without distortion.

The overall conformation of a nucleic acid helix can be described by the torsion angles along both the phosphate backbone and sugar ring. These torsion angles dictate the three-dimensional conformation of the nucleic acid, however, many of the torsion angles are interdependent and redundant. A more compact analogue which can be used to describe the conformation is the sugar pucker, glycosidic bond conformation, C4'-C5' bond orientation and phosphate ester shape.

1.3.2.1 Sugar Pucker

To minimise steric and electrostatic interactions between substituents, the furanose sugar ring in a nucleotide is often non-planar and twisted. This twisting, or *pucker*, can be described as the relative deviation of an atom from the median plane of the ring. Atoms can either deviate into the *exo* or *endo* face, where the endo face is on the same side as the carbon-5 atom and exo on the opposing face. For example, if the deviation of the carbon-2 atom of the furanose ring is above that of the median ring plane, on the same side as the carbon-5 group, this ring conformation can be described as C2'-endo. Pucker can also be measured numerically by calculating the phase of the sugar pucker in the pseudorotation cycle. Given the five-ring torsion angles shown in Figure 1.8, the pseudorotation phase (P) can be calculated by Equation 1.6.[121,122]

$$P = \frac{(\theta_2 + \theta_4) - (\theta_1 + \theta_3)}{2\theta_0(\sin 0.2\pi + \sin 0.4\pi)} \tag{1.6}$$

The two most frequent sugar puckers in nucleic acid structures are C3'-endo ($P \approx 18°$) and C2'-endo ($P \approx 160°$), shown in Figure 1.9. The position of the carbon-3 atom has a profound impact on the overall conformation of the nucleic acid. With the C3'-endo conformation, the carbon-3 atom and the connected oxygen-3 atom are pulled to the endo face of the furanose ring, reducing the inter-phosphate distance and therefore the nucleotide spacing. Whereas, with the C2'-endo conformation, the carbon-3 atom is pushed toward the exo face, increasing inter-phosphate distance and thus changing the overall conformation of the nucleic acid structure.

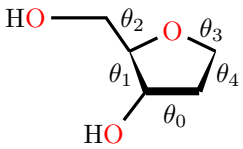
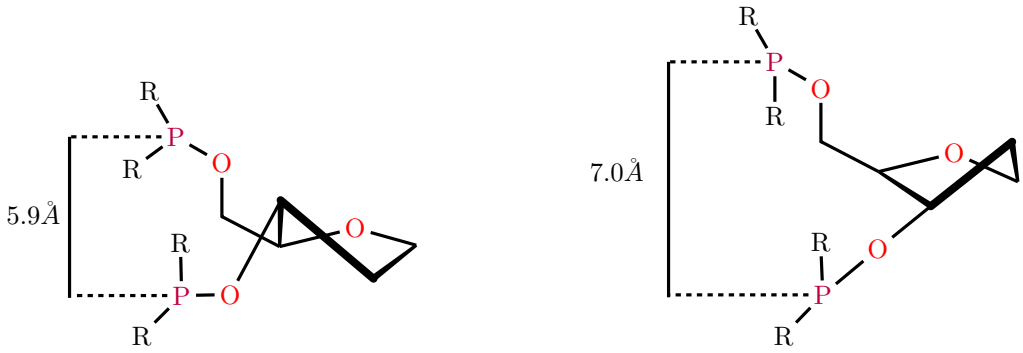


Figure 1.8: Five ring torsion angles present in 2'-deoxyribose in furanose form. These five torsion angles describe the overall conformation of the ring.

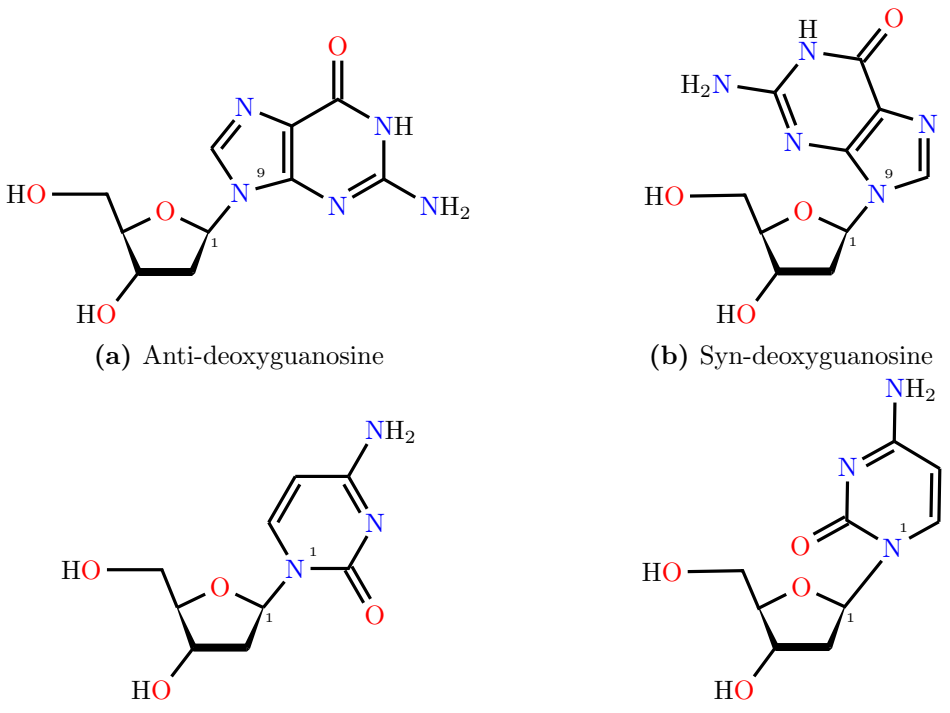


(a) C3'-endo **(b)** C2'-endo

Figure 1.9: Inter-phosphate distances for C3'-endo and C2'-endo conformations of a furanose ring. The carbon-3 atom in the C3'-endo conformation is above the plane of the ring, causing a shorter inter-phosphate distance when compared to the C2'-endo conformation.

1.3.2.2 Syn and Anti Base Conformation

The relative position of the nitrogenous base group in relation to the furanose sugar is another essential conformational feature of nucleic acids that enables large-scale assembly. The planar base lies almost perpendicularly to the furanose ring, enabling two primary orientations to emerge, the *syn* and *anti* conformations, shown in Figure 1.10. In the more common anti-conformation, both pyrimidine and purine bases have hydrogen atoms positioned above the furanose ring, with the bulk of the base group positioned away



(a) Anti-deoxyguanosine **(b)** Syn-deoxyguanosine

(c) Anti-deoxycytidine **(d)** Syn-deoxycytidine

Figure 1.10: Depictions of the syn and anti base conformations for pyrimidines and purines. In the syn conformation, the base lies over the furanose ring, whereas in the more common anti conformation, the bulk of the base group lies away from the furanose ring.

from the ring. Whereas, in the more uncommon syn-conformation, the bulk of the base group is positioned over the furanose ring, creating a more compact nucleotide. This base group position has a significant impact on the hydrogen bonding interactions that are possible with other base groups. In double-stranded DNA, the anti-conformation nucleotide is often exclusively observed, as it is more energetically favoured and facilitates a higher degree of hydrogen bonding. Nevertheless, the syn-conformation can be observed experimentally, particularly in active single-strand RNA molecules.[123]

1.3.3 Secondary Structure of DNA

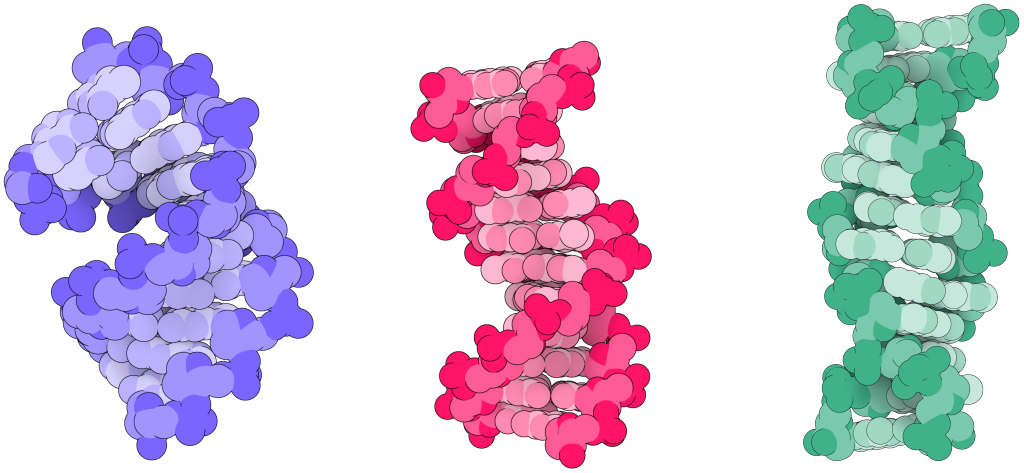
Early in DNA research, it was clear that multiple forms of DNA could be observed in varying reaction conditions. In a low-humidity environment, the A-form of DNA is present, whereas at higher humidity, the B-form of DNA is preferred. These two structural polymorphs are the basis of a wide array of DNA conformations, with most conformations falling into the A or B family. The main differences between the two polymorphs are a direct consequence of the sugar pucker conformations (see Section 1.3.2.1), base stacking interactions and the environment.

1.3.3.1 A-DNA

The overall conformation of A-DNA is similar to the Watson-Crick model, with two strands of DNA forming a right-handed double helix. A-DNA helices often form in dehydrating conditions, with relative humidities $\leq 75\%$,[124] albeit an uncommon condition in biological systems. The sugar-phosphate backbone of A-DNA lies on the outside of the helix, approximately 12 Å away from the helical centre, with sugar rings in the C3'-endo conformation. This sugar conformation reduces inter-nucleotide distances and causes adjacent bases in A-DNA to be approximately only 2.45 Å apart. Such small inter-base distances necessitate a helical tilt of 20 ° to minimise repulsive interactions. These conformational characteristics force the A-DNA conformation to be more compact and shorter in length than other forms of DNA.

1.3.3.2 B-DNA

The B conformation of DNA is more commonly found in biological systems and is present in higher humidity conditions. In B-DNA, the sugar rings are most commonly observed in the C2'-endo conformation, with attached base groups lying perpendicular to the helical axis, shown in Figure 1.11b. This geometry results in an inter-base spacing of approximately 3.4 Å, resulting in a more extended and straighter overall conformation. The predominance and stability of this helix geometry can be partially attributed to the solvated phosphate-sugar backbone. In crystal structures, ordered water molecules can



(a) A-DNA **(b)** B-DNA **(c)** Z-DNA

Figure 1.11: Non-photorealistic representations of the three most common DNA helix conformations. The A-DNA form is a compact form of DNA with a large helical tilt (PDB code: 117D[126]). The B-DNA form is the most commonly observed DNA conformation found in biological systems and is more extended and straighter than A-DNA (PDB code: 3R86[127]). Z-DNA is an unusual form of DNA with a left-handed helix that requires both syn and anti base conformations to form (PDB code: 3P4J[128]).

be observed, which solvate and stabilise exposed heteroatoms such as carbonyl oxygen and amino nitrogens.

1.3.3.3 Z-DNA

The conformation of Z-DNA has little similarity to that of A-DNA or B-DNA. The most significant difference is the preference for a left-handed helix over the more conventional right-handed helices of A and B DNA, shown in Figures 1.11a and 1.11b. The phosphate backbone of Z-DNA, shown in Figure 1.11c, exhibits an alternating, 'zig-zag' conformation, which is a direct result of the conformation of the base group at each nucleotide. Theoretically, the most stable left-handed DNA helix would require all nucleotide groups to adopt the syn conformation, with the base groups lying over the sugar ring. Steric interactions between the carbonyl group of pyrimidines and the furanose oxygen generally limit this conformation to purines. Alternating purine-pyrimidine base groups often allow this unusual anti-syn zig-zag conformation of DNA to form. Based on the glycosidic bond conformation, the conformation of the sugar ring also adapts with syn-pyrimidine nucleotide exhibiting a C3'-endo conformation and the anti-purine nucleotide exhibiting a C2'-endo conformation. The utility of Z-DNA in nature is a natural question, since theoretically, this conformation is less energetically favourable than the more common B-DNA conformation. However, it was found that the conversion from B-DNA to Z-DNA is often spontaneous in high salt conditions due to more energetically favourable $\pi - \pi$ stacking interactions in Z-DNA and more optimal salt-phosphate interactions.[125]

1.3.3.4 Other Forms

Whilst DNA is most often observed in helical form, interactions with other biomolecules facilitate a host of other DNA structural conformations. These non-ideal helices are functionally crucial in a wide array of cell regulation involving DNA.[129]

1.3.3.4.1 Hairpin Loops Hairpin loops occur when a single nucleotide strand contains a palindromic sequence or inverted repeats. This section of DNA can form both intra-strand interactions and inter-strand interactions. Hairpin loops are often the basis of other, more complex DNA structures, such as cruciforms.[130]

1.3.3.4.2 Cruciforms Cruciforms form when two sets of self-complementary sequence segments join together to form a larger, four-way junction section of DNA. Cruciforms are essential in a wide array of biological processes, including genomic regulation and recognition.[131,132]

1.3.4 Structure of RNA

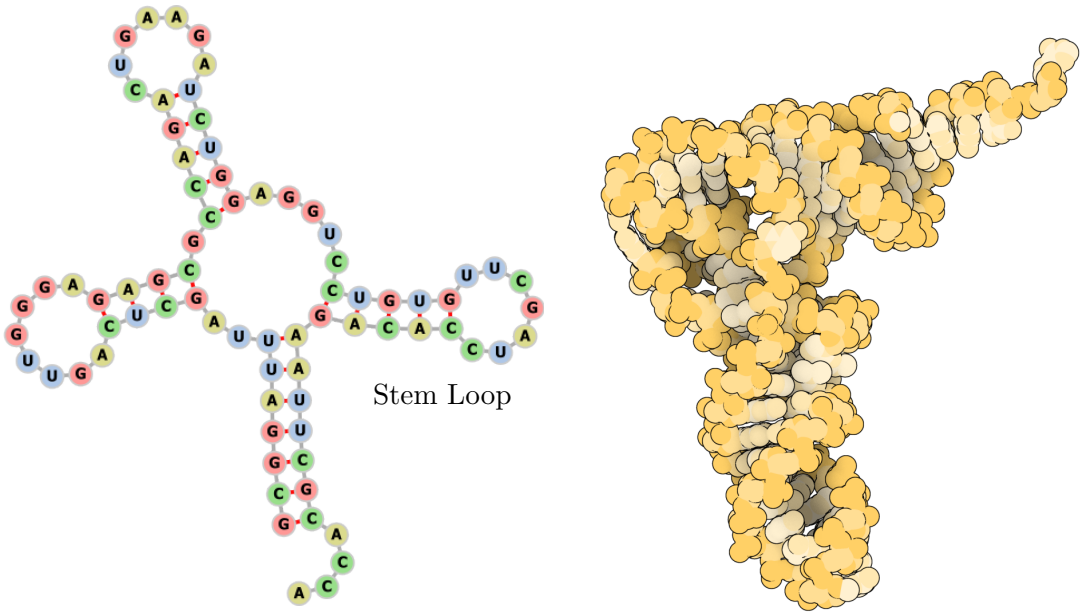
DNA can often exhibit a regular and predictable secondary structure, whereas the secondary structure of RNA has a greater degree of variability and more complexity. RNA has been observed to form regular DNA-like double-stranded structures, as well as larger hybrid double-stranded and single-stranded globular structures. Like DNA, the driving force behind the formation of secondary structure elements in RNA is complementary pairing between bases. However, the type of secondary structure observed for RNA depends on the type of RNA, and by extension, the RNA function.

1.3.4.1 mRNA

Messenger RNA (mRNA) is a polymer critical for protein synthesis. It is *transcribed* from DNA and contains the genetic information required to manufacture proteins in the ribosome. mRNA exists as a single-stranded polymer, which can include short self-complementary regions, allowing the formation of more highly ordered structures. One of the most common and most critical structural formations of mRNA is the hairpin.[133] Hairpins can be crucial for both the chemical stability and recognition of RNA. RNA hairpins form in a similar way to DNA hairpin loops, i.e. the mRNA sequence contains a palindromic region which can form inter-strand interactions, leaving an area of unpaired bases, as shown in Figure 1.12a.

1.3.4.2 tRNA

Transfer RNA, or tRNA, is another type of RNA crucial to protein synthesis. tRNA allows the ribosome to *translate* the mRNA into protein, acting as the physical bridge



(a) Secondary structure **(b)** Tertiary structure

Figure 1.12: A - Secondary structure of t-RNA with four stems and three loops stabilised by base pairing. Image generated with *Forna*.[135] B - Tertiary structure of tRNA (PDB code: 3A3A[136]). The secondary structure of tRNA has multiple stem loops, which form an L-shaped tertiary structure.

between the mRNA and the amino acid. The secondary structure of tRNA exhibits multiple internal loops, known as a *cloverleaf* structure, shown in Figure 1.12a. In three dimensions, this cloverleaf structure manifests as an L-shaped tertiary structure, characteristic of t-RNA, shown in Figure 1.12b.

1.3.4.3 dsRNA

Double-stranded RNA or dsRNA has recently been recognised as a major component of a host of viral infections.[134] The structure of dsRNA is similar geometrically to that of A-DNA, in a conformation known as A-RNA. A-RNA has a right-handed helical structure with sugar rings in the C3'-endo conformation, with standard Watson-Crick pairing between bases. The formation of double-helical RNA strands is reported to play a significant role in the innate immune responses of eukaryotes, highlighting the importance of this complex RNA formation.

1.3.4.4 RNA-DNA Complexes

The structural interface between RNA and DNA is another essential interaction which facilitates the regulation and growth of cells. For instance, the RNA polymerase enzyme is responsible for transcribing DNA into complementary mRNA. To enable this enzyme to function, an RNA-DNA complex must be formed. This hybrid duplex is stabilised by the Watson-Crick base pairing between the two strands, as would be expected for

a homodimer.[137] Conformationally, the RNA-DNA duplex is similar to that of a DNA-DNA or RNA-RNA helix in the A-conformation. This fundamental intercompatibility between RNA and DNA is possible due to the chemical similarity between thymine and uracil bases, which both can donate and accept hydrogen bonds in similar spatial areas.

1.3.5 Protein-Nucleic Acid Complexes

The importance of protein-nucleic acid complexes cannot be understated. These interactions underpin a significant portion of cellular biology, underscoring the overwhelming importance of understanding the chemical and structural interactions responsible for these complexes.

1.3.5.1 Recognition

The recognition of nucleic acids by nucleic acid-binding proteins is often mediated by sequential or structural specificity. Whilst DNA and RNA are monomerically similar, protein-DNA and protein-RNA recognition are frequently different due to the differences in observed secondary structure. The inherent flexibility of single-stranded RNA enables more specific binding to discriminating protein functional groups. In contrast, the homogeneity across DNA helices creates a more challenging environment for achieving structural specificity.

1.3.5.1.1 Double Stranded Recognition

The ability of a protein to recognise and bind to a double-stranded helical nucleic acid occurs through the protein effectively examining the exposed functional moieties of the helix. The characteristic parts of the duplex which can affect binding affinity are the accessible gaps left in the double helical structure, known as the major and minor grooves. These grooves are solvated regions that can accommodate protein secondary structure, provided they have sufficient spatial dimensions. For example, in B-DNA, the major groove is 11.7 Å wide and 8.8 Å deep, which provides enough space for a single protein α-helix or two β-sheets,[138] shown in Figure 1.13. The alpha helix binding motif is common among helix-turn-helix proteins, where a *probe helix* is inserted into the major groove of double-stranded DNA and forms multiple specific non-covalent interactions between the protein sidechains and nucleic acid bases. Protein binding to double-stranded RNA occurs in a manner similar to that of double-stranded DNA, despite ideal A-form RNA helices having major grooves too small for protein α-helix access. Recognition can occur, due to the inherent disorder of RNA duplexes, which convenes enough structural flexibility to accommodate protein secondary structure in RNA grooves.[139]

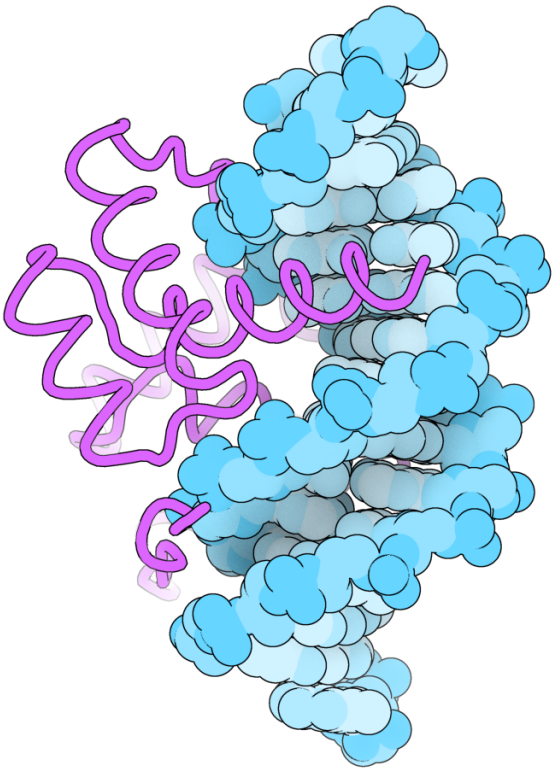


Figure 1.13: Helix-turn-helix protein-DNA complex (PDB code: 1FJL[143]). A protein α-helix is bound in the major groove of the B-DNA, forming the recognition motif.

1.3.5.1.2 Single Stranded Recognition

The recognition of single-stranded nucleic acids by proteins is another fundamental interaction to understand due to the reliance on these complexes for gene expression.[140,141] The comparatively less ordered single-stranded nucleic acid structures exhibit alternate binding motifs compared to double-stranded nucleic acids. One of the reasons for this binding motif shift is that the relative ease of accessing base functional groups in single-stranded structures allows for more optimal protein-nucleic acid interaction and recognition. In RNA-protein assemblies, single-stranded RNA recognition can be achieved through the RNA recognition motif or other conserved protein sequences such as the KH domain.[142]

1.3.5.2 Binding Interactions

1.3.5.2.1 Hydrogen Bonding As is ubiquitous for inter-nucleic acid interactions, hydrogen bonding comprises a significant and selective portion of the binding for protein-nucleic acid interactions.[144] Since hydrogen bonds are highly geometrically specific and require a particular arrangement of donor and acceptor atoms, they can probe recognition moieties precisely. Hydrogen bonding acceptors in proteins are often polar side chains such as asparagine and glutamine, as these polar groups usually have a δ^- charge which can equalise the δ^+ charge of the nucleic acid hydrogen bond donor. With single-stranded

nucleic acids, these protein sidechains can interact with the hydrogen bond donors of the exposed base groups as a pseudo-Watson-Crick base pair. Commonly, the carbonyl oxygens of cytosine, guanine, thymine and uracil can act as hydrogen bond acceptors, and the saturated nitrogen groups in all of the base groups can act as hydrogen bond donors. In certain conditions, water molecules can form a bridge between the protein and nucleic acid, particularly when the nucleic acid functionality is less accessible. While protein-base hydrogen bonding comprises a significant portion of nucleic acid interactions, other important hydrogen bonding pairs exist between the nucleic acid sugar-phosphate backbone and protein sidechains. The oxygen-rich backbone contains a host of potential hydrogen bond acceptor sites, and in the case of the hydroxylated carbon-2 group of RNA, a hydrogen bonding group,[145] although these interactions are comparatively weaker than base-protein interactions.[146]

1.3.5.2.2 Electrostatic Interactions In physiological conditions, the phosphate backbone of nucleic acids often exists in a negatively charged state, which is primarily localised on the non-bridging oxygen atoms of the phosphate group. These charged groups can interact with oppositely charged groups of the protein, creating an attractive electrostatic interaction. Most commonly, these interactions occur with the positively charged amino acid side chains of arginine, histidine and lysine, however electrostatic interactions with the terminal amino acid of a chain are also possible.[147] In contrast to hydrogen bonding, electrostatic interactions are not highly geometrically dependent, leading to minimal sequence-specific binding interactions. The energy of electrostatic interactions depends primarily on the distance between oppositely charged groups. Therefore, these interactions often facilitate a larger-scale examination of the binding profile to gauge whether the protein and nucleic acid are roughly intercompatible.

1.3.5.2.3 Hydrophobic Effect Conceptually, the formation of a complex between two monomers is entropically disfavored, however, the liberation of the solvent molecules on the surface of both the protein and nucleic acid interfaces is a driving factor for energetically favourable complex formation. This principle is known as the *hydrophobic effect* and is another key interaction which guides non-specific interactions.[148,149]

1.3.6 Nucleic Acids in X-ray Crystallography

The structural solution of nucleic acids through X-ray crystallography is often a more complex task than the structure solution of other macromolecules, such as proteins. The additional complexity is primarily due to the difficulty in obtaining well-diffracting crystals. This difficulty commonly results in an exhaustive trial of a range of different crystallisation conditions to facilitate crystal growth. An essential part of crystal growth is crystal contact formation, and the negatively charged backbone of nucleic acid structures

often disrupts this delicate process.[150] In an attempt to aid this situation, nucleic acids can be engineered to promote crystallisation without significantly affecting the structure.[151]

Once diffraction data is obtained for a given nucleic acid-containing crystal, another barrier presents itself for the reconstruction of a nucleic acid, namely, acquiring the phase information. Several strategies can be employed to address this inherent issue, including molecular replacement, anomalous dispersion, or multiple isomorphous replacement (see Section 1.1.5). If a known homologue has already been solved, then molecular replacement is a viable strategy for structure solution, however, if not, further experimental techniques are required. For nucleic acids, the incorporation of heavy atoms in the structure can be obtained through *soaking* or through the construction of the nucleic acid with synthetic base groups.

1.3.6.1 Molecular Replacement

Molecular replacement for nucleic acids is a more difficult task when compared to proteins, primarily due to the relative lack of solved nucleic acid structures in the Protein Data Bank.[152] However, if a homologous model is available, using molecular replacement to solve a nucleic acid structure can be straightforward. In particular, molecular replacement is particularly beneficial when solving protein-nucleic acid complex structures. The abundance of experimentally solved protein binding targets and *in silico* predicted proteins can act as fruitful molecular replacement search targets.[153]

1.3.6.2 Experimental Phasing

Experimental phasing using anomalous scattering or isomorphous replacement has historically been crucial for solving nucleic acid structures. Both experimental methods rely on the introduction of heavy atoms in the structure. Often, heavy atoms that mimic the native metal ions in the structure are added to reduce the likelihood that heavy atom addition disrupts or alters the crystal structure. In RNA, hydrated Mg^{2+} ions can be replaced with cobalt-III, osmium-III or iridium-III ions with minimal disruption. Heavy atoms can also be engineered to bind to specific surfaces in nucleic acid structures, by the introduction of particular base pairs into the structure, such as the G-U base pair in RNA.[154]

1.3.6.3 Challenges

The major bottleneck in the structure solution pipeline for nucleic acids remains the ability to crystallise nucleic acid-only structures. Incorporating other, more readily crystallisable macromolecules in tandem with the nucleic acids is a viable strategy for getting

a nucleic acid-containing crystal.[155] However, it can be argued that the solved structure after co-crystallisation may not fully represent the structure of the native nucleic acid. The degree to which the crystallisation process affects the crystal structure is often hard to determine, particularly when crystal contacts can introduce structural changes to the target structure. This point should be considered when conclusions are drawn from such structures.[146]

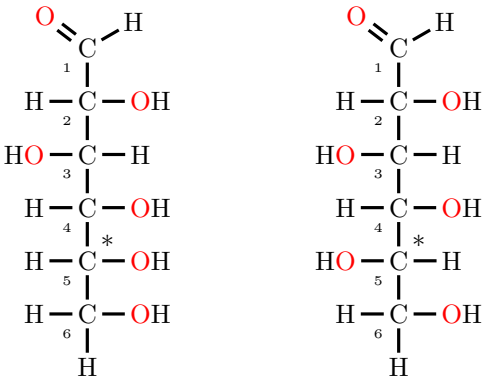
1.4 Glycosylation

The process of glycosylation involves the addition of a carbohydrate group to a biomolecule. Glycosylation underpins a wide array of complex and crucial biological processes which allow cells to function normally. The study of glycosylation, known as glycobiology, has often lagged behind that of protein or nucleic acid biochemistry owing to the inherent complexity associated with glycosylated macromolecules. Unlike proteins or nucleic acids, glycosylation does not follow a specified template and is instead driven by enzymatic processes, often co-translationally or post-translationally in the case of protein glycosylation.[156] The attached carbohydrates, known as *glycans*, exhibit extensive structural complexity which can be attributed to the possibility of branching and conformational isomerism.

1.4.1 Monosaccharides

The basis of a complex carbohydrate moiety is a single monosaccharide, also known as a sugar. Commonly, sugars exist with the empirical formula $(CH_2O)_n$ where $1 \leq n \leq 9$ and consist of a chain of hydroxymethlene (HCOH) groups capped at one end with a hydroxylmethyl group (H_2COH) and at the other end with a carbonyl-containing group. Sugar chains capped with an aldehyde take the name *aldoses* and sugar chains capped with α-hydroxy ketones take the name *ketoses*. With the exception of three carbon ketoses, all monosaccharides exhibit some amount of stereoisomerism, with the number of stereoisomers defined by the formula 2^k where k is the number of chiral carbon atoms.

The overall stereoisomeric configuration can be classified as either D or L based on the position of the chiral carbon furthest from the carbonyl group.[157] If the hydroxyl group on the furthest chiral carbon is positioned to the right in the Fischer projection, the sugars are described as the D stereoisomer, shown in Figure 1.14a. In contrast, if the hydroxyl group is positioned to the left, it is described as the L stereoisomer, shown in Figure 1.14b. Through evolution, the D stereoisomer of sugars became more naturally prevalent, although specific biochemical pathways still rely on the L stereoisomer of certain sugars.[158]



(a) D-glucose **(b)** L-glucose

Figure 1.14: Fischer projections of D-glucose and L-glucose with the furthest chiral carbon (carbon-5) from the carbonyl highlighted. The position of the hydroxyl on the furthest chiral carbon determines the stereoisomer of the sugar.

Sugars often exist in an equilibrium state between acyclic and cyclic forms.[159] From an acyclic sugar, an intramolecular nucleophilic addition between a hydroxyl group and the carbonyl group forms a cyclic hemiacetal or hemiketal structure. Multiple cyclic structures can result from a single acyclic sugar based on the position of the nucleophilic hydroxyl group. For example, six-carbon-long aldehyde-capped acyclic chains, commonly known as aldohexoses, often form two stable cyclic structures. A ring closure between the carbon-1 and carbon-5 atoms results in a six-membered cyclic hemiacetal known as a *pyranose*, whereas a ring closure between the carbon-1 and carbon-4 atoms results in a five-membered cyclic hemiacetal known as a *furanose*, shown in Figure 1.15. Other cyclic structures, such as seven-membered rings, are possible, although they may be less chemically stable in solution.[160]

(a) Pyranose form of D-glucose **(b)** Furanose form of D-glucose

Figure 1.15: Haworth projections of the six-membered ring pyranose and five-membered ring furanose forms of D-glucose. From an aldohexose, the pyranose form is formed through a 1,5 ring closure whereas the furanose form is formed through a 1,4 ring closure. D-glucose is most commonly found in pyranose form, although it may be found in small quantities in the furanose form.

1.4.1.1 Three-Dimensional Structure

Whilst carbohydrates can often be drawn as planar ring structures, these configurations are improbable in three dimensions. The most stable conformation of a six-membered ring, such as a pyranose, is the *chair*, which can exist in two conformations, either $^4C_1$

or $^1C_4$.[161] These designations detail which carbon atom is above (superscript) and below (subscript) the median ring plane designated by the carbon-2, carbon-3, carbon-5, and oxygen atoms when the oxygen is at the top right of the configuration. Almost all six-membered ring carbohydrates exist in either chair conformation, but distortions to this low-energy ring conformation are likely during enzyme action.[162] Five-membered ring carbohydrates exhibit less rigid conformational preferences and can be found in a variety of different conformations as described in Section 1.3.2.1.

1.4.1.2 Anomericity

During cyclisation, the nucleophilic hydroxyl group can attack the carbonyl from above or below the plane of the resulting ring, resulting in the formation of two *anomers*. The carbon of the carbonyl, which becomes chiral after cyclisation, is termed the *anomeric carbon* and exists in either the α or β form. The α form is defined when the anomeric carbon has the same chiral configuration as the chiral carbon furthest from the anomeric carbon, and the β form is present when the anomeric carbon and the furthest chiral carbon have different configurations. For example, if the carbon-1 atom of an aldohexose is in the R configuration and the carbon-5 atom is in the S configuration, the anomeric carbon would be in the β anomer, as shown in Figure 1.16.

(a) α-anomer of D-glucopyranose **(b)** β-anomer of D-glucopyranose

Figure 1.16: α and β anomers of D-glucopyranose shown in the $^4C_1$ chair conformation. When the anomeric carbon is in the same stereochemical configuration as the furthest chiral carbon, the anomer is defined as α, and when they are different, the anomer is defined as β.

1.4.1.3 Modifications

Beyond the strict definition of a carbohydrate, a sugar can exist with a range of chemical modifications, which most commonly occur on a hydroxyl group. These modifications can fine-tune and extend the functionality of modified glycans in many systems.[163]

1.4.1.3.1 Amino Groups Amino-derivatised sugars exist ubiquitously in nature and are known to possess a variety of crucial functions in biological processes.[164] Commonly, one or multiple hydroxyl groups are replaced with either a free amino group (NH_2) or an N-acetamido group ($NHCOCH_3$).[165]

1.4.1.3.2 Esterification The multiple hydroxyl functional groups on sugars can enable targeted functionalisation for glycan-specific analyses. This technique has shown

particular utility in stabilising sugars during mass spectrometry analysis,[166] although esterified sugars can also be found as intermediate structures during the synthesis of complex sugar structures.[167]

1.4.1.3.3 Reduction of Hydroxyl Groups The reduction of hydroxyl groups in sugars is a common feature of nucleic acid chemistry, and glycosylated sugars can also exhibit similar behaviour. The reduction of an OH group to a H atom often requires multi-step reduction chemistry, however natural enzymes known as *reductases* can also perform this modification, allowing basic sugars to be converted into more specialised forms.[168]

1.4.1.3.4 Methylation Methylation is a common modification on other biomolecules like proteins and nucleic acids, and is known to affect the modulation and regulation of important cellular processes such as cell signalling. The methylation of sugars, however, is a rare process that has been observed in bacteria and fungi, but not in mammals. There is little evidence to suggest that methylated sugars function differently from non-methylated sugars despite being found in nature.[169]

1.4.1.4 Common Monosaccharides

In eukaryotic cells, the majority of monosaccharide functionality can be attributed to a small selection of common sugars. These sugars are shown in Table 1.1 alongside a pictorial representation known as the *Symbol Nomenclature for Glycans (SNFG)*.[170] Commonly, sugars are referred to by the code assigned by the *CCD*.[85]

Table 1.1: Common carbohydrates found in glycoproteins. Each carbohydrate is displayed with a code assigned in the *Chemical Component Dictionary (CCD)*, the standard name with stereochemical descriptions, the standard abbreviation used in literature and the symbol assigned when used in the *Symbol Nomenclature for Glycans (SNFG)* representation.

CCD Code	Name	Abbreviation	SNFG Symbol
NAG	β-D-*N*-acetylglucosamime	β-GlcNAc	
BMA	β-D-mannose	β-Man	
MAN	α-D-mannose	α-Man	
GAL	β-D-galactopyranose	β-Gal	
NGA	β-D-*N*-acetylgalactosamine	β-GalNac	
GLC	α-D-glucopyranose	α-Glc	
FUC	α-L-fucopyranose	α-Fuc	
SIA	*N*-acetyl-α-neuraminic acid	α-Neu5Ac	

1.4.2 Oligosaccharides

Individual sugars can exhibit many different stereoisomers with various degrees of modification, but the immense complexity of glycans arises through the linkage of multiple monosaccharides into an *oligosaccharide*. Linkages in oligosaccharides occur through a glycosidic bond formed between two hydroxyl groups, catalysed by a *glycosyltransferase enzyme*.[171] These linkages can occur through either the α or β monosaccharide anomers and between any of the plentiful reactive hydroxyl groups. Additionally, a single monosaccharide can be linked to multiple other monosaccharides allowing the potential for highly branched glycan structures. Seemingly small changes in the linkage between two monosaccharides can give rise to major changes in the function of the resulting structure.[172]

1.4.3 Protein Glycosylation

The glycosylation of proteins, forming a *glycoprotein*, is the most common post-translational modification with over 50 % of mammalian proteins reportedly exhibiting some form of glycosylation.[173] Many cellular functions are known to rely on the glycosylation of proteins, such as cell signalling and immune responses.[174,175] Glycosylation can also aid the folding of specific proteins by increasing protein solubility and reducing aggregation, or

through intermolecular interactions in the endoplasmic reticulum.[176] Protein glycosylation can occur through the covalent modification of a range of different amino acid residues, each with a specific function. The most common amino acid modified with a glycan is asparagine (Asn), but serine (Ser), threonine (Thr), tryptophan (Trp) and cysteine (Cys) have all been found to be glycosylated.

1.4.3.1 *N*-linked Glycosylation

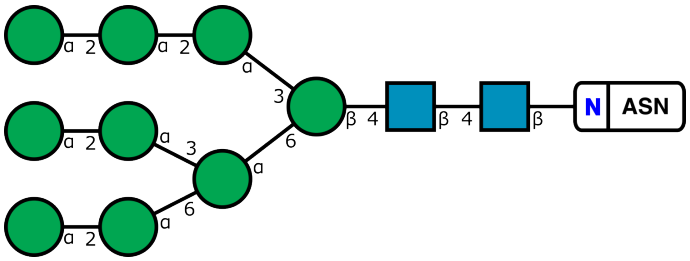
N-linked glycosylation involves the covalent attachment of a carbohydrate to the nitrogen atom of an asparagine residue. The attached carbohydrate moiety is commonly referred to as an *N*-glycan and is found abundantly in eukaryotic proteins. Early studies into *N*-glycans highlighted the inability for all asparagine residues to be *N*-glycosylated and found that a specific set of preceding amino acid residues was required. The most common specific sequence requirement, known as a *sequon*, for *N*-glycans was shown to be Asn-X-Ser/Thr, where X can be any amino acid except for proline.[177]

In eukaryotic cells, *N*-glycosylation occurs in the endoplasmic reticulum and Golgi apparatus. The process is initiated with the production of a dolichol-linked precursor oligosaccharide in the endoplasmic reticulum through the step-wise addition of two β-D-*N*-acetylglucosamine (GlcNAc) monosaccharides by two *transferase* enzymes, followed by the step-wise addition of five mannose monosaccharides. This intermediate oligosaccharide is then transferred to the lumen of the endoplasmic reticulum, where four additional α-D-mannose (Man) monosaccharides are added, followed by three α-D-glucose (Glc) monosaccharides, producing the dolichol-linked precursor.[178–180]

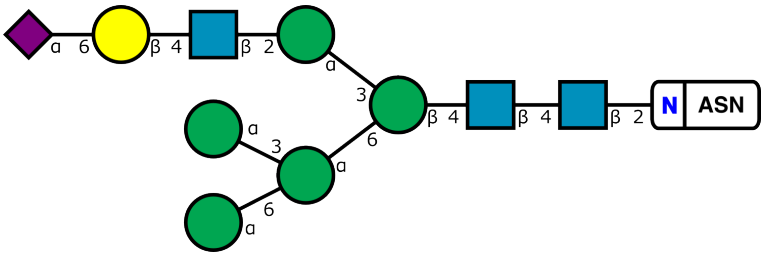
Following the generation of this precursor molecule, an oligosaccharyltransferase complex transfers the oligosaccharide onto the recently translated protein guided by the Asn-X-Ser/Thr sequons. Further processing of the new *N*-glycan removes the terminal Glc monosaccharides, and the protein is encouraged to fold by chaperone proteins.[181,182] After release from the endoplasmic reticulum, the glycoprotein is transported to the Golgi apparatus, where the *N*-glycans can be heavily processed by a variety of enzymes. The resultant *N*-glycans from this complex process can be classified into three main types: high mannose, complex, and hybrid, shown in Figure 1.17. The likelihood of formation for each subtype of *N*-glycosylation varies with the organism and available enzymes.

In addition to protein fold stabilisation, *N*-glycans can also perform various other integral cellular tasks. Functional insight is afforded by targeted suppression of *N*-glycosylation with mutations or inhibition and has revealed the significant role *N*-glycosylation plays in disease.[183] For example, *N*-glycans on the surface of cells facilitate virus-host interactions, which play a key role in the proliferation of viruses such as SARS-CoV-2.[184] In

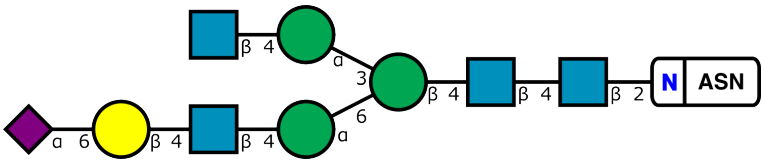
other cases, underglycosylation of essential proteins can lead to diseases with life-altering symptoms or embryonic lethality.[185]



(a) High mannose



(b) Hybrid



(c) Complex

Figure 1.17: *Symbol Nomenclature for Glycans (SNFG)* representations of the three main types of *N*-glycosylation. The three forms shown consist of the same core structure with varying terminal sugars depending on the degree of enzymatic processing. A - High mannose structure from a *N*-glycosylated *β*-D-glucosidase enzyme (PDB code: 5FJI[186]). B - Hybrid structure from a *N*-glycosylated Fab antibody (PDB code: 4DQO[187]). C - Complex structure from a *N*-glycosylated transferrin binding protein (PDB code: 3V8X[188]).

1.4.3.2 *O*-linked Glycosylation

O-linked glycosylation involves the covalent attachment of a carbohydrate group to the oxygen atom in the side chain of a serine or threonine residue. A range of different carbohydrates have been reported to attach to form *O*-glycans, such as GlcNAc, *α*-Man, *β*-D-*N*-acetylgalactosamine (GalNAc), *β*-D-galactose (Gal), and *α*-L-fucose (Fuc).[189] Specific glycosyltransferase enzymes catalyse these additions, with further attachment of a range of possible monosaccharide components, often yielding a heterogeneous and large glycan chain.

O-glycans are synthesised relatively more simply than *N*-glycans, with enzymatic addition of a single carbohydrate, which forms the substrate for further addition. These potentially large *O*-glycan chains are commonly found in clusters on cell surface pro-

teins, secreted proteins, and muscular proteins and play significant roles in many cellular functions.[183] *O*-glycan deficiencies have been shown to play a significant role in diseases, such as bone deformation,[190] but studies into *O*-glycans are relatively underdeveloped compared to those on *N*-glycosylation. Perhaps one of the main reasons for this is the lack of a predictable amino acid sequence that indicates where *O*-glycosylation will occur. *O*-glycans are most commonly studied with mass spectrometry or nuclear magnetic resonance, with structural studies limited due to *O*-glycan heterogeneity, size, and lack of tooling.[156]

1.4.3.3 *C*-linked Glycosylation

C-linked glycosylation is a relatively rare form of glycosylation which mainly involves the covalent addition of a single α-Man carbohydrate in the $^1C_4$ chair conformation to the aromatic sidechain of a tryptophan.[191] Whilst only a single carbohydrate, adding α-Man to tryptophan can significantly alter the chemical affinity of the residue. The heterocyclic indole moiety of the tryptophan sidechain is highly hydrophobic, but after modification with an α-Man, the group becomes more hydrophilic. This increased hydrophilicity can promote certain intramolecular and intermolecular interactions which affect the three-dimensional conformation of the protein.[192] Additionally, the binding of carbohydrates can also increase the likelihood of additional localised sugar attachment.[193] *C*-linked glycosylation occurs in the endoplasmic reticulum of mammalian cells, where *C*-mannosyltransferase enzymes attach an α-Man to the tryptophan following the Trp-X-X-Trp/Cys sequon.[194] Research into the functionality of *C*-linked glycosylation is still in its infancy, as tools and techniques that are used for *N*-linked and *O*-linked glycosylation become more functional for *C*-linked glycosylation, it is likely the function and importance of *C*-linked glycosylation will be better understood.

1.4.4 Lipid Glycosylation

Glycosylated lipids, or glycolipids, are lipids which have been functionalised by the addition of a carbohydrate group and are present in all eukaryotic cell membranes. Glycolipids are produced in the Golgi apparatus through the glycosyltransferase-catalysed addition of carbohydrates to an unmodified hydroxyl group on a lipid. Following modification, the lipid is transported to the cell membrane via a vesicle.[195] Glycolipids are crucial in stabilising the membrane in many cells and the precise structure of a given glycolipid can facilitate specific intercellular recognition. This microscopic change can result in macroscopic effects, exemplified well by human blood group antigens, which are determined by the specific oligosaccharide present on the extracellular glycolipid on red blood cells.[196] Two of the most important and common forms of glycolipids are the *glycosphingolipid*, based on the amino alcohol sphingosine, and the *glycoglycerolipid*, which is based on

the triol glycerol. These glycolipids are often modified with a single potentially acetylated galactose or glucose monosaccharide, but other monosaccharides have been found naturally.[197]

1.4.5 Nucleic Acid Glycosylation

Recently, the glycosylation of nucleic acids was discovered and confirmed as a third scaffold for glycosylation behind proteins and lipids,[198] shown in Figure 1.18. Glycosylated RNA, or glycoRNA, are small non-coding RNA structures which have been found on the surface of many cell membranes in a variety of species. GlycoRNAs are thought to function similarly to other glycosylated macromolecules present on the cell surface and play a role in intercellular interactions and recognition. GlycoRNAs also have the potential to elucidate new causes of human disease, as abnormal glycosylation is known to be a major factor in various diseases.[199] The structure of glycoRNAs is thought to resemble a complex *N*-linked protein glycan, as it may be formed through a similar biosynthetic pathway. However, further research into this area is required to elucidate the mechanism of formation and potential three-dimensional structures.

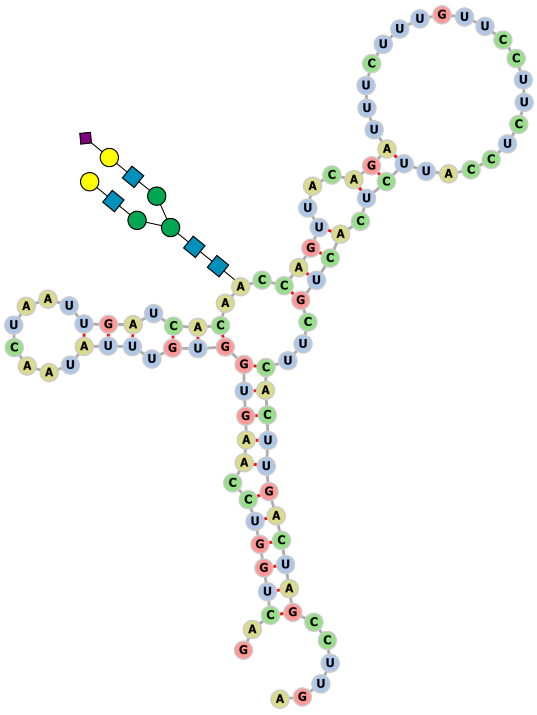


Figure 1.18: Potential secondary structure of glycosylated Y-RNA motif.[200] Currently, the attachment point between the first sugar and nucleic acid is unknown, but is shown arbitrarily to be bound to adenosine. RNA secondary structure generated using *Forna*.[135]

1.4.6 Glycosylation in X-ray Crystallography

X-ray crystallography is one of the most important techniques for elucidating the three-dimensional structures of many biomolecules that contain carbohydrates. Carbohydrate-

containing structures currently represent around 10 % of the total depositions in the Protein Data Bank.[201] However, despite this relatively high abundance, protein glycosylation is often seen as a problem for crystallographers due to difficulties in obtaining crystals of glycoproteins.

1.4.6.1 Experimental Challenges

The purpose of the wide variety of glycan compositions in glycosylated structures can be well attributed to the vast array of functionality glycosylation provides in a cellular environment.[202] However, it is in this variety that problems can arise when attempting to study glycoproteins using crystallography. The glycans of glycosylated structures are inherently heterogeneous and can crystallise in a variety of forms, creating a challenging environment when trying to resolve the structure with a high resolution.[203] Extending these issues is the fact that the potentially large glycans are exceedingly mobile and often protrude into the solvent surrounding a macromolecule, creating issues with efficient and regular packing which are fundamental in obtaining good-quality X-ray diffraction data. To mitigate these problems, crystallographers often resort to removing the glycans mostly or entirely using specific glycosidase enzymes, commonly *PNGase F* or *Endo H*.[204]

1.4.6.2 Carbohydrates in the Protein Data Bank

Once a suitable crystal and experimental dataset have been obtained, the next challenge is modelling a structure to the observed experimental data. After the backbone of the macromolecule has been identified, ligands and post-translational modifications can be modelled. For many years, the process of adding glycans into a model was not well supported by refinement programs, which caused various issues with carbohydrate models. In the absence of suitable restraints, the carbohydrate rings were often distorted into unfavourable, high-energy conformations instead of the more likely $^4C_1$ chair conformation.[162] Many mismodelled structures were deposited into the Protein Data Bank, creating a dangerous situation where errors could propagate into new structures or lead to wrongly directed research. Fortunately, these problems have been largely corrected through the addition of appropriate carbohydrate ring restraints in refinement software,[205] validation software packages like *Privateer*,[91] and re-refined databases such as *PDB-REDO*.[206]

1.5 Machine Learning

Data points rarely exist in isolation and often correlate with other sources of information, allowing for the study of interesting relationships and the inference of new information. Identifying these trends can be trivial in some instances, but frequently, correlating multidimensional data points extends past what is easily achievable by humans. Machine learning attempts to address these difficulties by creating statistical algorithms that can model trends in data.

After collecting some data, each data point can be described as labelled or unlabelled, and the machine learning algorithms that apply to any given dataset depend on this condition. Labelled data, or data with a specific category assigned to it, lends itself well to *supervised learning* techniques, whereas unlabelled data often are analysed by *unsupervised learning* algorithms. The choice of algorithm type that is most appropriate for a given research question is highly dependent on the data available.

1.5.1 Supervised Learning

When a labelled dataset is available, supervised learning is the most logical choice of machine learning algorithm. Supervised learning techniques often involve attempting to build a statistical model that can assign a value to a given input. This input is referred to as a *feature vector* and is commonly denoted using the term $\boldsymbol{x_i}$. Each value in this feature vector is known as a *feature*. In supervised learning, each feature vector has an associated label denoted by y_i and may be classified into several distinct categories ($y_i \in \{0, 1, ..., C\}$), be a real number ($y_i \in \mathbb{R}$) or represent other complex data structures. A collection of labelled feature vectors, $\{(\boldsymbol{x_i}, y_i)\}_{i=0}^{N}$ comprises a *dataset*. Many supervised learning algorithms exist, ranging from more simplistic linear or logistic regression models to more complex neural network-based models.

1.5.1.1 Linear Regression

It is often helpful to understand how two variables correlate, and regression models can parametrise these relationships as a function of a single variable. The mathematical definition of a simple regression model is shown in Equation 1.7, where y is the dependent variable, f is the regression model, x is the independent variable or variables, and ϵ is the error term corresponding to what is not parameterisable inside the regression model. [207]

$$y = f(\boldsymbol{x}) + \epsilon \tag{1.7}$$

The most simplistic regression model is the *linear regression model* which assumes a linear relationship between the dependent variables and the independent variables, shown in

Equation 1.8, where p represents the number of *parameters* in a given model and is equal to the dimensionality of the input variable $\boldsymbol{x_i}$, in the simplistic case, $p = 1$.[208]

$$f(\boldsymbol{x}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p \tag{1.8}$$

The parameters of the model, denoted by $\boldsymbol{\beta}$ ($\boldsymbol{\beta} \in \mathbb{R}$) are initially unknown, and the machine needs to *learn* them using a set of labelled feature vectors known as a *training dataset*, denoted by $\mathcal{T} = \{(\boldsymbol{x_i}, y_i)\}_{i=0}^{N}$. In this context, machine learning amounts to finding a set of parameters which allow the model to fit the matrix multiplication shown in Equation 1.9 where $\boldsymbol{y}$ is an n-dimensional column vector of y_i values, $\boldsymbol{X}$ is a matrix of shape $n \times (p+1)$ containing the $\boldsymbol{x_i}$ values.

$$\boldsymbol{y} = \boldsymbol{X} \cdot \boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{1.9}$$

Finding a set of parameters that best satisfies Equation 1.9 often occurs through the *maximum likelihood* method. This method aims to find the value of $\boldsymbol{\beta}$ which has the highest statistical likelihood of producing the observed $\boldsymbol{y}$ values. To assess how well a model is performing, a *loss value* is calculated with a *loss function* which measures the deviation between the true value y_i and the predicted value $\hat{y}_i$ and is often denoted by $\boldsymbol{\mathcal{L}}$. In linear regression, the loss function is often defined as the mean square error (MSE), shown in Equation 1.10.

$$\mathcal{L}(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{1.10}$$

With a modest number of parameters, a closed-form solution can be obtained through matrix inversion, however as the number of parameters increases, this method becomes computationally infeasible. With large models, the most common method of parameter optimisation occurs through the *gradient descent* algorithm. The first step of gradient descent is to make an initial estimation of the parameters $\boldsymbol{\beta}$, followed by many alterations to $\boldsymbol{\beta}$ to minimise the value of $\mathcal{L}(\boldsymbol{\beta})$. The direction of the parameter alteration is decided after calculation of the partial derivative of the loss function with respect to each parameter of $\boldsymbol{\beta}$, shown in Equation 1.11, where $\nabla_{\boldsymbol{\beta}}\mathcal{L}$ is the gradient vector.[209]

$$\nabla_{\boldsymbol{\beta}}\mathcal{L}(\boldsymbol{\beta}) = \left(\frac{\partial \mathcal{L}}{\partial \beta_1}, \frac{\partial \mathcal{L}}{\partial \beta_2}, ..., \frac{\partial \mathcal{L}}{\partial \beta_p} \right) \tag{1.11}$$

The parameters $\boldsymbol{\beta}$ are then updated as shown in Equation 1.12, where η is the update rate, more commonly referred to as the *learning rate*. This process is repeated until the model reaches convergence and the loss value no longer decreases.

$$\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} - \eta \nabla_{\boldsymbol{\beta}}\mathcal{L}(\boldsymbol{\beta}) \tag{1.12}$$

Once the optimal parameters, $\boldsymbol{\beta}^*$, are found for a given training set, the model is ready to perform *inference* on examples not part of the training set, as shown in Equation 1.13 for $p = 1$.

$$\hat{y}_{\text{new}} = \beta_0^* + \beta_1^* x_{\text{new}} \tag{1.13}$$

1.5.1.2 Logistic Regression

Linear regression is a key tool in modelling the relationship between an input variable or variables and a continuous output, however in many cases, the output variable can be better described as one of several categories. Applying linear regression to such a problem is not straightforward since the codomain of a linear regression model spans all real numbers ($\hat{y} \in \mathbb{R}$). A solution to this issue involves transforming the output of a linear regression model with the *standard logistic function*. This method is known as *logistic regression* but is better defined as a categorisation rather than a regression. The standard logistic function, also called the sigmoid function, is defined in Equation 1.14 and shown graphically in Figure 1.19.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1.14}$$

The sigmoid function has a codomain ranging from 0 to 1, allowing the output of the linear regression model to be mapped to a probability of a given class occurring, given the input, $P(y = k \mid x)$ where $k \in \{0, 1\}$. The resultant logistic regression model can be defined formally as shown in Equation 1.15.[210]

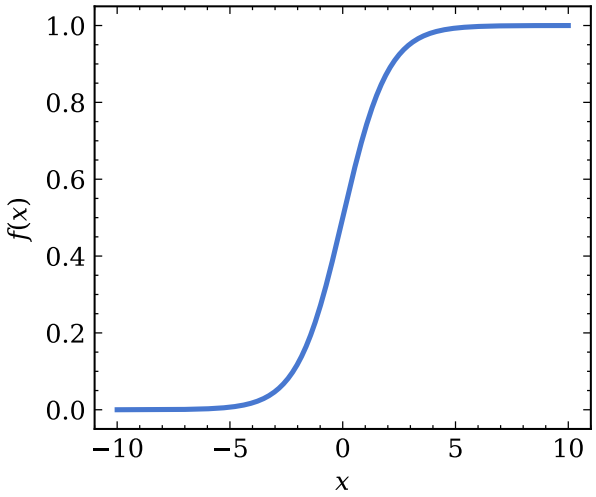


Figure 1.19: Standard logistic function with domain $x = \mathbb{R}$ and codomain $f(x) = (0, 1)$.

$$f_{\boldsymbol{\beta}}(x) \stackrel{\text{def}}{=} \frac{1}{1+e^{-\boldsymbol{\beta x}}} \quad \text{where} \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} \tag{1.15}$$

Since this model outputs probabilities of a given class, instead of minimising some loss function as is done in linear regression, the model can be optimised by maximising the *likelihood* of the training data, $\mathcal{T}$, given the model $f_{\boldsymbol{\beta}}$. The likelihood, ℓ, is expressed as the probability of $\boldsymbol{y}$ given the model input, $\boldsymbol{x}$, and the model parameters, $\boldsymbol{\beta}$, and evaluates to the product of the model outputs over all samples and categories, shown in Equation 1.16.

$$\begin{aligned} \ell &\stackrel{\text{def}}{=} P(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\beta}) \\ &= \prod_{i=0}^{n} P(y_i \mid x_i; \boldsymbol{\beta}) \\ &= \prod_{i:y_i=0}^{n} P(y_i = 0 \mid x_i; \boldsymbol{\beta}) \prod_{i;y_i=1}^{n} P(y_i = 1 \mid x_i : \boldsymbol{\beta}) \\ &= \prod_{i:y_i=0}^{n} 1 - \frac{1}{1+e^{-\boldsymbol{\beta x_i}}} \prod_{i;y_i=1}^{n} \frac{1}{1+e^{-\boldsymbol{\beta x_i}}} \\ &= \prod_{i=0}^{n} \left(1 - \frac{1}{1+e^{-\boldsymbol{\beta x_i}}}\right)^{1-y_i} \left(\frac{1}{1+e^{-\boldsymbol{\beta x_i}}}\right)^{y_i} \end{aligned} \tag{1.16}$$

To optimise the parameters, $\boldsymbol{\beta}$, more easily, it is convenient to take the logarithm of the likelihood to create a log-likelihood value, as shown in Equation 1.17. Following an initial guess of $\boldsymbol{\beta}$, gradient descent can be used to modify $\boldsymbol{\beta}$ to maximise the log-likelihood value throughout the training dataset. After optimal parameters, $\boldsymbol{\beta}^*$, have been calculated for the logistic regression model, the most probable output category for a given unseen input value, x_{new}, can be selected as the predicted category, $\hat{y}$, after application of an *argmax* function, shown in Equation 1.18.

$$\begin{aligned} \log \ell &= \sum_{i}^{n} \left[(1-y_i) \log \left(1 - \frac{1}{1+e^{-\boldsymbol{\beta x_i}}}\right) - y_i \log \left(\frac{1}{1+e^{-\boldsymbol{\beta x_i}}}\right) \right] \\ &= \sum_{i}^{n} (1-y_i) \boldsymbol{\beta x_i} - \log(1 + \exp^{-\boldsymbol{\beta x_i}}) \end{aligned} \tag{1.17}$$

$$\hat{y} = \underset{k=0,1}{\operatorname{argmax}} (P(y = k | x_{\text{new}})) \tag{1.18}$$

In certain cases, it may be useful to model more than two possible output categories, which basic logistic regression cannot accomplish. To generalise logistic regression to K categories, the input vector must be converted from a single real number into a vector

representation. The method most often used for this purpose is *one-hot encoding*. One-hot encoding involves transforming the integral category number ($k \in \{1, 2, ..., K\}$) into a vector representation of length K, where K is the total number of categories. After one-hot encoding, the output vector has a value of 1 in the k^{th} position, with all other values set to 0. For example, if the total number of categories K is 4 and a given training example y_i had an integral value of 2, the one-hot encoded vector $\boldsymbol{y_i}$ would be $[0, 1, 0, 0]$. The standard logistic function is unable to operate on this one-hot encoded vector, therefore, the multinomial logistic function, or *softmax* function, is used. The softmax function is a generalisation of the logistic function which produces an output vector that has a sum of 1, with all elements lying between 0 and 1, shown in Equation 1.19. These conditions allow the output of a softmax function to be interpreted as probabilities for a direct comparison to the one-hot encoded vector.[211]

$$P(k \mid \boldsymbol{x_i}; \boldsymbol{\beta}) = \frac{e^{\boldsymbol{\beta_k x_i}}}{\sum_{l=1}^{K} e^{\boldsymbol{\beta_l x_i}}} \tag{1.19}$$

In a similar way to logistic regression, softmax regression can use the maximum likelihood approach to optimise the parameters $\boldsymbol{\beta^*}$. With softmax regression there are K sets of parameters, each modelling the probability of a given category. Resultantly, slight alterations to the likelihood function are required to accommodate this, commonly referred to as *cross entropy*, shown in Equation 1.20. Once all sets of parameters have been calculated, the model is ready to be used to infer the category given an unseen input x_{new}.

$$\begin{aligned} \text{let } \boldsymbol{\theta} &= [\boldsymbol{\beta_1}, \boldsymbol{\beta_2}, ..., \boldsymbol{\beta_K}] \\ \log \ell(\boldsymbol{\theta}) &= \sum_{i=1}^{n} logP(y_i \mid \boldsymbol{x_i}; \boldsymbol{\theta}) \\ &= \sum_{i;y_i=1}^{n} logP(1 \mid \boldsymbol{x_i}; \boldsymbol{\theta}) + ... + \sum_{i;y_i=K}^{n} logP(K \mid \boldsymbol{x_i}; \boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log P(k \mid \boldsymbol{x_i}; \boldsymbol{\theta}) \end{aligned} \tag{1.20}$$

1.5.1.3 Neural Networks

Certain datasets can exhibit trends that are too complex for a single linear regression to model. In such instances, aggregating multiple simple models can be a viable strategy to create a functional model. The most popular type of aggregate model is the *neural network*, which is a non-linear function consisting of multiple generalised linear regression models. The generalised linear regression model differs from a standard linear regression model by the addition of a scalar *activation function*, denoted by $\sigma : \mathbb{R} \to \mathbb{R}$ and shown in Equation 1.21 and as a diagram in Figure 1.20.

$$z = \sigma(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p) \tag{1.21}$$

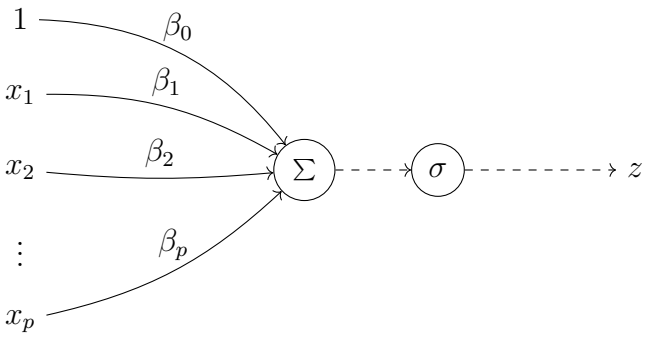


Figure 1.20: Diagram of a generalised linear regression model with inputs variables x_i multiplied by parameters β_i and summed, followed by application of the activation function σ. Solid lines represent parameterised connections, dashed lines are non-parameterised connections.

A single generalised linear regression model has little chance of modelling complex non-linear relationships, so multiple models must be combined. In a neural network, the output, z, is calculated as the summation of N generalised linear regression models, each with a local set of parameters and an output denoted by h_i, shown in Equation 1.22. These intermediate generalised linear regression models are known as *hidden units* and can be logically organised into a *hidden layer*. The output of a hidden layer then forms the input of the subsequent layer, whether that be another hidden layer or the output. This process is shown diagrammatically in Figure 1.21 with a single hidden layer.[212]

$$z = \sigma(\beta_0 + \beta_1 h_1 + \beta_2 h_2 + ... + \beta_N h_N) \tag{1.22}$$

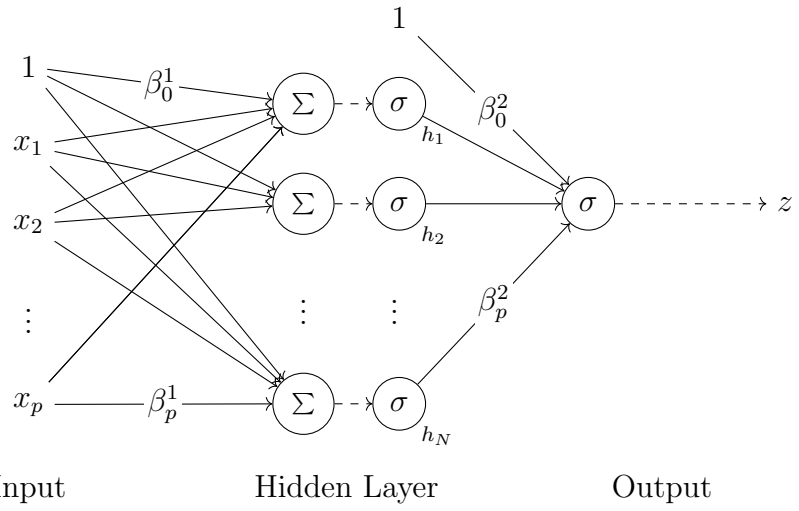


Figure 1.21: Diagram of neural network with one input layer, one hidden layer, and one output layer. Solid lines represent parameterised connections, dashed lines are non-parameterised connections.

The parameters of a given network can be represented concisely as a matrix representation where $\boldsymbol{W}$ is known as the *weight matrix* and $\boldsymbol{b}$ is known as the *bias vector*. These representations allow the neural network shown in Figure 1.21 to be expressed as a series of equations, shown in Equation 1.23, where l represents the layer of the network.

$$\text{let } \boldsymbol{b} = \begin{bmatrix} \beta^l_{0,1} \\ \vdots \\ \beta^l_{0,N} \end{bmatrix} \quad \text{and} \quad \boldsymbol{W} = \begin{bmatrix} \beta^l_{1,1} & \cdots & \beta^l_{p,1} \\ \vdots & \cdots & \vdots \\ \beta^l_{1,M} & \cdots & \beta^l_{p,M} \end{bmatrix} \tag{1.23}$$

$$\boldsymbol{h} = \sigma(\boldsymbol{W^1 x} + \boldsymbol{b^1})$$
$$z = \sigma(\boldsymbol{W^2 h} + \boldsymbol{b^2})$$

The choice of activation function, σ, for hidden layers is problem-dependent, with the *Rectified Linear Unit (ReLU)* being the most common choice, defined in Equation 1.24. *ReLU* is often convenient to use as it is fast to compute and alleviates issues with performing calculations on very small numbers during network training. The activation function of the hidden layers is comparatively less important than the activation function of the output layer, which determines the function of the network. If a logistic function is chosen, the network will become a categorical model ($f : X \to \{1, 2, ..., K\}$), whereas, if a linear function is chosen the network will become a regression model ($f : X \to \mathbb{R}$).

$$\text{ReLU}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \tag{1.24}$$

Training a neural network follows the same principles as linear or logistic regression, namely, the network parameters ($\boldsymbol{W}$ and $\boldsymbol{b}$) must be optimised such that an appropriate loss function is minimised. One major difference between training a neural network and other simpler models is that a neural network can be sensitive to the initial set of parameters, and unfavourable choices of initial parameters can lead to slow loss convergence or even failure to train altogether. Most commonly, neural networks are initialised with small, positive numbers in the hope that the various hidden units highlight different trends in the data.

Neural networks tend to be trained on extensive datasets, and computing a loss score which averages the error over all data points would be computationally infeasible if the dataset were extremely large. To mitigate this, the dataset is often assumed to be *redundant* such that drawing n samples from the dataset represents the dataset well as a whole. This small sample is known as a *batch* and the value selected will dictate the speed and efficacy of training. During training, after every batch has been supplied to the network, the parameters are updated using an efficient gradient calculation method known

as *backpropagation*.[213] Once all data from a dataset has been passed to the network, an *epoch* has passed and the dataset is shuffled before another epoch begins.

1.5.1.3.1 Deep Neural Networks

Single hidden layer neural networks, known as *shallow neural networks*, are an improvement over a single linear regression model at modelling non-trivial relationships in data. Neural networks with more than one hidden layer are termed *deep neural networks* and are substantially more powerful at modelling complex correlations in data, for example, the correlation between an image and a description. Deep neural networks are currently the state-of-the-art machine learning model architecture, finding uses in various practical scenarios such as computer vision and generative models.[214] As a given deep neural network grows, the number of parameters that must be optimised during training also grows, which can become computationally infeasible or intractable for very large networks.[215]

1.5.1.3.2 Convolutional Neural Networks

When working with regularly structured data such as a two-dimensional image or a three-dimensional grid, a common strategy to reduce the number of trainable parameters in a neural network is to take advantage of the *convolution operation* to form a *convolutional neural network*. The main difference between fully-connected neural networks and a convolutional neural network is the removal of the many inter-layer connections, which are replaced by convolving a learnable *kernel* over the input space to produce the subsequent layer. This *sparse connectivity* enables the detection of features in the input with equivariance to translation, which is particularly useful when working with image data.

The convolution operation amounts to the sum of the dot products over each overlapping region between the input space and the kernel, the result of which is then deposited into the output space. A one-dimensional convolution with a kernel of length 3 is shown in Figure 1.22 and a two-dimensional convolution with a kernel of size (3×3) is shown in Figure 1.23. In a convolutional neural network, multiple kernels can be used in a single layer to attempt to highlight many different features. The choice of kernel size and other parameters such as *stride*, the distance between each consecutive convolution, can dictate the size of the convolved output. Often it is desirable for the output of a convolutional layer to be the same dimensions as the input. In these cases, *padding* must be used to ensure a fixed dimensionality.[216]

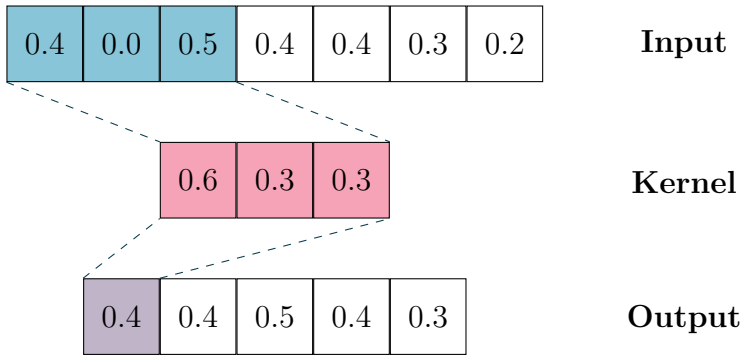


Figure 1.22: One-dimensional convolution operation on an input of shape (7) convolved with a kernel of shape (3) with a stride of 1 forming an output of shape (5).

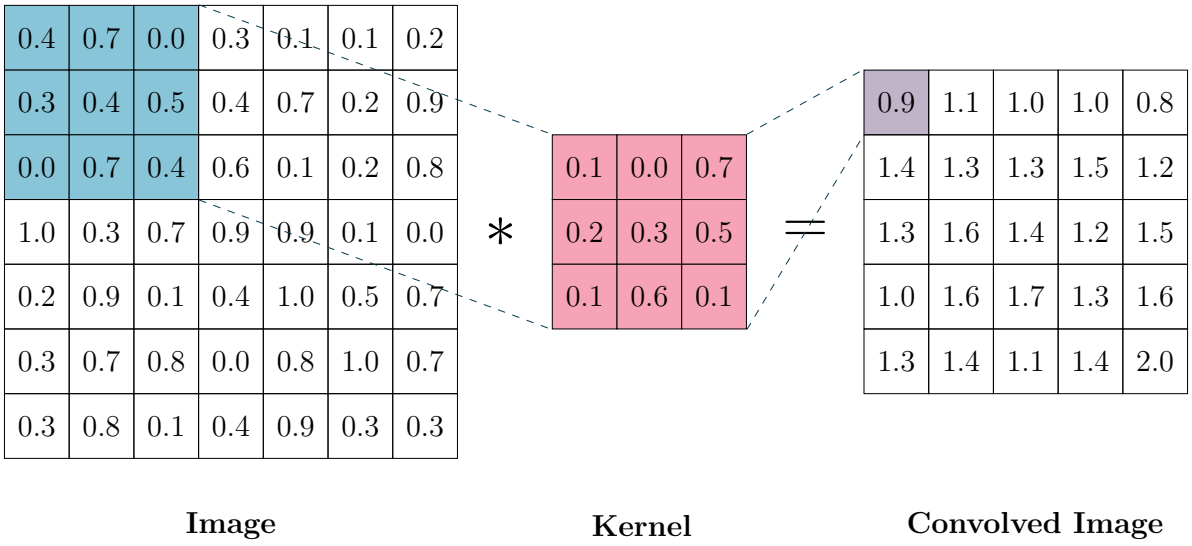


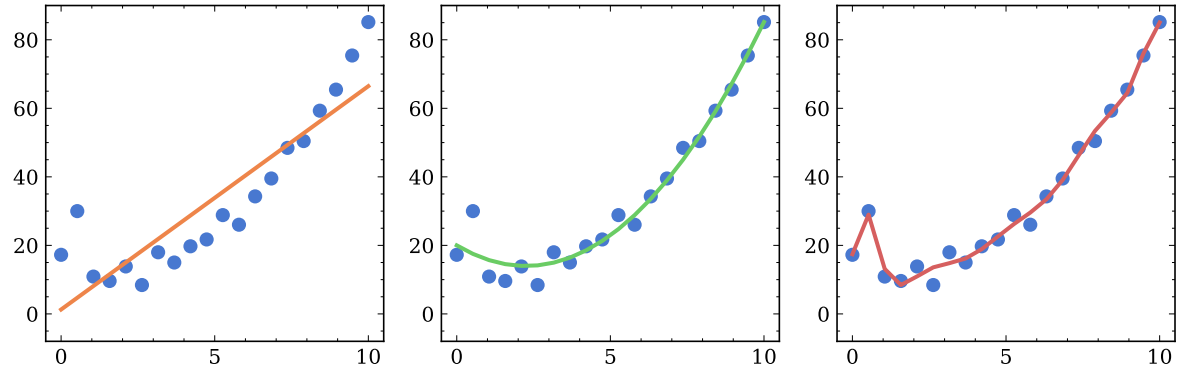
Figure 1.23: Two-dimensional convolution operation on an input image of shape (7×7) convolved with a kernel of shape (3×3) with a stride of 1, forming an output image of shape (5×5).

1.5.1.4 Model Evaluation

When designing and implementing any machine learning model it is important to evaluate the model during and after training, to ensure that the model is trending toward an optimal solution. A good indicator of model improvement is the minimisation of the loss function which can be monitored during training. Solely observing the loss function score however, does not fully encapsulate the current performance of a given model, and is liable to become biased toward training data. The combination of loss function score and other metrics are key tools to ensure that a machine learning model is performing well.

1.5.1.4.1 Underfitting and Overfitting

One of the main benefits of linear regression models over non-linear regression models is the reluctance of a linear regression model to *overfit* the data. Overfitting is a common problem when working with machine learning models and describes a model which performs well on data points in the training set but performs poorly on any other data points. Underfit, well-fit and overfit models are shown in Figure 1.24 with a non-linear regression model. The underfit model has modelled a linear relationship into the non-linear data which does not describe the complex relationship well. The overfit model has modelled the noise in the non-linear data resulting in the potential for erroneous inferred values. The well-fit model accurately encompasses the non-linear trend in the data while avoiding becoming heavily reliant on the noise in the data.



(a) Underfit model **(b)** Well-fit model **(c)** Overfit model

Figure 1.24: Fitness levels of regression models on non-linear synthetic data. The underfit model was modelled using linear regression, the well-fit model was modelled with a 2^{nd}-degree polynomial, and the overfit model was modelled with a 10^{th}-degree polynomial. Axis labels are hidden for clarity since this data is synthetic.

1.5.1.4.2 Other Metrics

A variety of other metrics can be calculated which are useful when evaluating the strength of a supervised machine learning model. For regression tasks, mean squared error is often an appropriate metric which exaggerates large differences between the true and predicted values, shown in Equation 1.10. In classification, a range of additional metrics are used to visualise the performance of the model, many of which are based fundamentally on the *confusion matrix*. The confusion matrix highlights the similarities and differences between the K predicted categories and the true categories, shown in Figure 1.25. These comparisons can be categorised into one of four types: true positives, false positives, false negatives, and true negatives, which enable the calculation of multiple scalar metrics that can be monitored during model training. The most common metrics used in modern machine learning are accuracy, precision, recall and F1 score, defined in Equations 1.25.

Accuracy can be intuitively understood as the overall proportion of correct predictions by a given model. Precision measures the proportion of predicted positives that are actually positives, giving insight into how reliable positive predictions are. Recall provides an understanding of how many of the true positives the model could correctly identify. F1 score is defined as the harmonic mean of precision and recall, and provides a single metric that is commonly used to measure the performance of the model.

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Figure 1.25: Confusion matrix commonly used to monitor machine learning model performance. The actual and predicted positive and negative sample results can be used to measure the number of true positives, true negatives, false positives and false negatives.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.25a}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1.25b}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{1.25c}$$

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1.25d}$$

1.5.2 Unsupervised Learning

Unsupervised learning techniques attempt to build a statistical model of a dataset which does not contain specific labels for each data point. The dataset consists of N unlabelled feature vectors, $\{\boldsymbol{x_i}\}_{i=1}^{N}$ and unsupervised learning models locate structure or patterns in the data which can be exploited to help solve a practical problem. Two of the most useful forms of unsupervised learning are clustering and dimensionality reduction which both allow for a more nuanced understanding of a given dataset.

1.5.2.1 Clustering

Clustering is a useful technique for locating and assigning subgroups in a dataset where data points within each subgroup have similar properties. Many clustering algorithms leverage different statistical models for determining clusters, and the most appropriate method is often difficult to determine without exhaustive computation. The most popular clustering methods rely on either assigning the dataset to a known number of partitions based on similarity or rely on the density of data points to determine clusters.

1.5.2.1.1 K-Means

If the number of desired clusters, k, is known prior to clustering, k-means clustering is an efficient and effective choice for clustering a dataset. The algorithm first randomly assigns k feature vectors, known as *centroids* and denoted by $\boldsymbol{\mu}$. The centroids have the same shape as the feature vectors and exist in the same dimensionality. The distance between each dataset feature vector, $\boldsymbol{x_i}$, and each centroid, $\boldsymbol{\mu_j}$, is calculated and each feature vector is assigned to the nearest centroid. The distance metric chosen can depend on the application but most commonly the *Euclidean distance*, d, is used, defined in Equation 1.26.

$$d(\boldsymbol{x_i}, \boldsymbol{\mu_j}) = \sqrt{\sum_{d=1}^{D} (\boldsymbol{\mu_{j,d}} - \boldsymbol{x_{i,d}})^2} \tag{1.26}$$

An average is calculated for the set of feature vectors assigned to each centroid, which forms the new feature vector for that given centroid. This process of distance calculation and centroid assignment continues until the centroid assignments no longer change. Each set of feature vectors assigned to each centroid is then known as a cluster. If the algorithm succeeds, the output of the clustering process is k distinct clusters, an example of which is shown in Figure 1.26. The clusters generated rely on the random initial centroid positions which may result in different results for the same dataset. Other algorithms attempt to address this by removing the variability in the starting positions.[217]

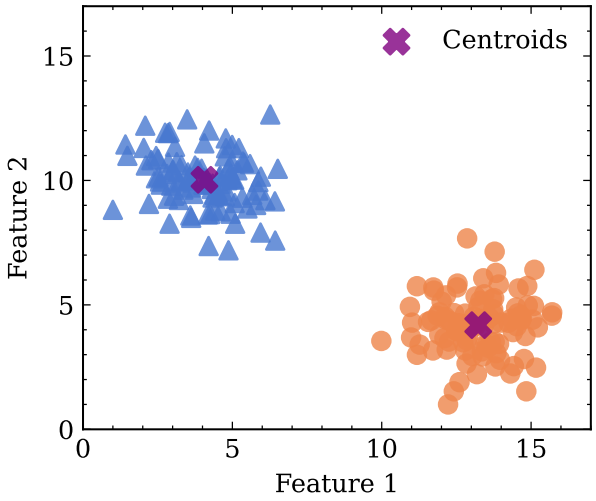


Figure 1.26: Results of k-means clustering with an example dataset. The algorithm highlighted two clusters shown with blue triangles and orange circles. The centroids of each cluster from the last iteration of the algorithm are shown with a purple cross.

1.5.2.1.2 DBSCAN

In many cases, the number of clusters in a dataset is not known or impractical to determine prior to analysis. In such cases, it is useful to inspect the density of the data points to assign clusters, which is the method behind the DBSCAN algorithm. Before clustering, two parameters must be selected, the minimum distance between two points, ϵ, and the minimum number of points to constitute a cluster, n. To start, a random feature vector is selected from the dataset, and the distance to all surrounding points is calculated using a specific distance metric, d. The number of points, p, within the distance, ϵ, is calculated and if p is greater than the minimum number of points, n, all p points are considered to be in the same cluster. The clusters are then expanded by repeating this locality calculation until all feature vectors in the dataset have been considered. The flexibility of DBSCAN to work with any number of clusters can be beneficial in many use cases but often comes at the cost of increased computational expense.[218]

1.5.2.2 Dimensionality Reduction

When working with practical or complex datasets, data points can exist with a very large number of dimensions, and visualising this data is largely incomprehensible to humans. To tackle this, the dimensionality of the dataset can be reduced to allow it to be visualised more simply in two or three dimensions. Commonly, dimensionality reduction is accomplished through *principal component analysis* which transforms the data into a new coordinate system, with each new direction capturing the parameters with the most variance in the dataset. The variables with the highest variance can be determined through the calculation of a *covariance matrix*, shown in Equation 1.27, which describes

the relationship between each pair of parameters as a covariance value. By performing a decomposition of this matrix, the directions of maximum variance, known as principal components, can be obtained. The two or three directions corresponding to the largest principal components can then be used to define a new coordinate system. This method reduces the amount of information lost in the reduction by reorienting the dataset with respect to the most important parameters. Modern computing hardware has advanced such that working with high-dimensional datasets is trivial, so the necessity of dimensionality reduction for data processing has dwindled, nevertheless it is still a very useful technique when visualising data or removing noise in a dataset.[219]

$$\mathrm{Cov}\boldsymbol{X} = \begin{bmatrix} \mathrm{var}(X_1) & \dots & \mathrm{cov}(X_1, X_N) \\ \vdots & \ddots & \vdots \\ \mathrm{cov}(X_N, X_1) & \dots & \mathrm{var}(X_N) \end{bmatrix} \tag{1.27}$$

1.5.3 Machine Learning in X-ray Crystallography

Determining a macromolecular structure using X-ray crystallography encompasses a range of experimental and computational techniques that can be difficult or impractical to accomplish by hand. To increase the likelihood of structure solution, various computer algorithms have been developed to enable X-ray crystallography to be accessible to non-expert users. Many of these methods involve some form of unsupervised or supervised machine learning technique to automatically manipulate or model trends in the data. The first reported machine learning-like methods described the assignment of amino acid features into protein electron density maps in 1977. Despite this method lacking resemblance to modern techniques, this work outlined the ideas behind many of the crystallographic machine learning algorithms used today.[220]

Much of the attention in algorithmic development has been focused on helping model the macromolecule of interest into the collected experimental data. One of the first programs to incorporate a machine learning model for automated backbone tracing in proteins was *CAPRA*, released in 2002.[221] *CAPRA* used a shallow neural network to predict the positions of the Cα backbone from a series of geometric patterns from placed pseudo-atoms. This method was reportedly successful but the reliance on an electron density map originating from well-phased reflections may have limited the practical use of the software for solving new structures. This model design was the basis for the backbone tracing machine learning models in the automated model-building program *TEXTAL*.[56] After the main chain is traced and fits well into density, one of the next steps is modelling the side chains of amino acids. For each residue, the side chain type must be determined in consultation with the known sequence of the protein, however this can be a challenge at low resolution. A residue-type classification model was introduced into the automated model-building software package *ARP/wARP* which uses logistic regression to output the probabilities of all 20 amino acids given a vector description of the side chain electron density.[222]

Following automated model building, model validation ensures that erroneous residues can be identified and fixed. Classically, this is done by measuring parameters for each residue such as density fit or average B-factor. These values can appear obscure to a non-expert crystallographer and so often thresholds are used in programs like *Coot* to alert users to residues in need of inspection. This process has also been accomplished by using a shallow neural network to interpret all of the available calculated parameters and output a correctness score, which can more intuitively guide users to problematic residues.[223] In addition to model building, machine learning methods have also been applied in other areas of crystallography, such as high-throughput crystallisation,[224] diffraction quality analysis,[225,226] data reduction,[227] and beamline operations.[228]

Part I

Automated Model Building of Nucleic Acids

Preface

This part is based on the published article 'NucleoFind: a deep-learning network for interpreting nucleic acid electron density'.[229] A summary of author contributions is as follows:

- Jordan S. Dialpuri - model development, model evaluation, model optimisation, figure generation, writing,
- Kathryn. D. Cowtan - supervision and support,
- Jon Agirre - supervision and support,
- Paul S. Bond - initial model development, supervision, and support. Initial model development consisted of implementing a similar encoder-decoder network in TensorFlow for an unrelated project.

Chapter 2 is reproduced similarly to the published article, with minor changes in prose, descriptions and figures. Chapter 3 represents work completed after publication to enhance the method. Chapter 4 uses some ideas discussed in the published article, but contains new methodological advancements. The completed software package, *NucleoFind*, is scheduled for incorporation into the *CCP4* Software Suite.

Chapter 2

Identification of Nucleic Acids in Density

2.1 Introduction

Interpretation of macromolecular electron density maps from X-ray crystallography can be trivial for an experienced crystallographer, but is a conceptually challenging problem for computer algorithms to solve. Despite this, many software packages have been successful at automatically building macromolecular structures into electron density. In protein atomic model building, algorithmic approaches that rely on orientation-dependent likelihood functions[63] or free atom placement[230] are mature and work well. However, the electron density of nucleic acid-containing structures is often more difficult to interpret than that of proteins, especially after phase estimates are obtained by protein molecular replacement when solving a protein-nucleic acid complex. Nevertheless, automated model building can work well in some cases after molecular replacement, but many still require further manual attention.[64]

The technical challenge of a program understanding complex 3-dimensional shapes with high variability across instances aligns well with the capabilities of deep learning-based methods. This chapter presents a set of deep learning models for the interpretation and segmentation of electron density maps derived from structures containing nucleic acids. The models can positively identify the three constituent parts of a nucleotide, the phosphate group, the ribose sugar and the nitrogenous base before an atomic model is built. The predictions from these models are helpful when attempting to build nucleic acid features into electron density, after molecular replacement. The context obtained from the predictions has been used to enhance the capabilities of automated model-building software in historically difficult cases, such as when building large protein-nucleic acid complexes after molecular replacement using a protein template.

2.1.1 Background

The neural network architecture at the core of this software method is based upon the U-Net architecture.[231] The U-Net is a convolutional neural network designed to analyse and segment two-dimensional biomedical images, with a strong focus on utilising the relatively limited number of data samples efficiently and effectively. As opposed to taking in the entire data sample at once, the U-Net instead relies on a small portion, or 'chunk', of data being supplied to the network for analysis, with the results then combined to classify the entire data sample. In the original U-Net deployment, this data was two-dimensional, but subsequent research expanded the U-Net to three dimensions for use in other biomedical areas.[232] The original 3D U-Net implementation used a collection of two-dimensional images to generate a three-dimensional dataset, however, other sources of three-dimensional data, such as a crystallographic electron density map, may also be used.

This type of convolutional neural network can also be described as an encoder-decoder network with opposing downsampling and upsampling portions. Similar network architectures have already been shown to be extremely useful for characterising experimental data in structural biology. *Haruspex*[233] demonstrated the impressive utility of these convolutional neural networks by annotating the secondary structure of cryo-EM Coulomb potential maps. The network at the core of *Haruspex* assigned a probability of each point in the map corresponding to an α-helix, β-sheet, nucleotide, or an unassigned feature. The network received cubes of density between 40^3 Å^3 and 48^3 Å^3 in volume as input, which provided sufficient secondary structure coverage while minimising model complexity. This annotation proved useful in informing downstream automated model-building pipelines.

Since then, approaches that replace classical algorithmic model-building software pipelines with methods more reliant on neural networks have emerged. *DeepTracer*[108] uses four separate encoder-decoder (U-Net) networks to obtain precise structural information from cryo-EM Coulomb potential maps alone. The first network categorises each point in the map as belonging to specific atom types, while the second analyses each point for its proximity to the protein backbone. In a similar way to the network in *Haruspex*, a third U-Net network classifies each point by its secondary structure, and the final network assigns each point to an amino acid type. Combining the outputs of these classifications, in particular the protein backbone and atom-type networks, enables efficient chain tracing. Using more classical optimisation algorithms on the atomic positions and other classification networks, *DeepTracer* can quickly build models from cryo-EM density maps.

A similar network, the modified feature pyramid, is present in the popular software pack-

age *ModelAngelo*. This encoder-decoder network characterises each point in a cryo-EM Coulomb potential map as either the Cα atom in an amino acid or the phosphorus atom in a nucleotide in a similar way to the atom-type network in *DeepTracer*. However, while *DeepTracer* relies on classical algorithmic model-building methods to transform network classifications into accurate protein models, *ModelAngelo* achieves this by adding a further graph neural network. This graph neural network optimises the positions of the located residues using information from the map, the sequence, and the geometry of neighbouring residues. A significant advantage of *ModelAngelo* over *DeepTracer* for model building is the ability to build nucleic acids in addition to protein features.

Recently, another program capable of building nucleic acid structures, *CryoREAD*, was released.[109] Again, the U-Net architecture was used to identify and classify parts of the cryo-EM density. In this case, the networks classified each point in the map as sugar, phosphate, base or none. Following this, classical chain tracing and sequence assignment produce an accurate full-atom model. The utility of these convolutional neural networks in cryo-EM is clear, but their applicability has also been shown with protein crystallographic density maps. Using a 3D fully-convolutional DenseNet, which has similar downsampling and upsampling stages to a basic U-Net, Á. Godó *et al.*[234] segmented crystallographic protein density maps into each residue type without requiring sequence information.

2.1.2 Aims

This chapter aims to investigate the use of deep learning to consistently identify nucleic acid features in electron density. A deep learning model that can locate nucleic acid features more accurately than current-generation software methods may lay the groundwork for more accurate and complete automated model building of nucleic acid-containing structures. Additionally, any output from a deep learning model may also be used to guide interactive model building. Both applications aim to accelerate the throughput of structure solution of nucleic acid-containing biological molecules.

2.2 Test Set Creation

A test set was generated to evaluate the performance of software methods for identifying nucleic acid electron density in realistic molecular replacement examples. Commonly, when attempting to solve a nucleic acid structure in complex with a protein, the protein structure is either known or can be predicted *in silico*. Therefore, to generate this realistic test set, 1000 protein-nucleic acid structures solved by X-ray crystallography were randomly selected from the Protein Data Bank, without resolution or size filtering. The structure factors and sequences for this set were downloaded and each sequence was run

through the *MrParse* software package[235] to search the AlphaFold Structural Database[36] for suitable molecular replacement search models. The search model with the highest *MrParse* score was selected for each sequence, representing the structure with the best sequence alignment. These structures were then run through molecular replacement using the software package *Slice'N'Dice*.[236] *Slice'N'Dice* attempts to split up the search model into domains before providing the models to *PHASER*,[237] increasing the likelihood of a successful molecular replacement solution. These solutions were filtered to contain only solutions with a minimum protein completeness of 50 %, i.e. the molecular replacement solution has to make up at least 50% of the deposited protein structure. Applying this filter ensures that the test set contains a range of possible solutions that could realistically be found during routine structure solution. This process yielded 288 realistic molecular replacement examples with a resolution range of 1.35 Å to 4.00 Å and an average resolution of 2.29 Å, shown in Figure 2.1. The dataset consists of 228 protein-DNA complexes, 57 protein-RNA complexes and 3 protein-DNA/RNA complexes. Contained within the test set are a variety of different biological molecules with a range of functions, such as transcription factors, exonucleases, polymerases, ligases, and recognition particles. These 288 structures were excluded from any other dataset where this test set may be used to prevent accidental bias.

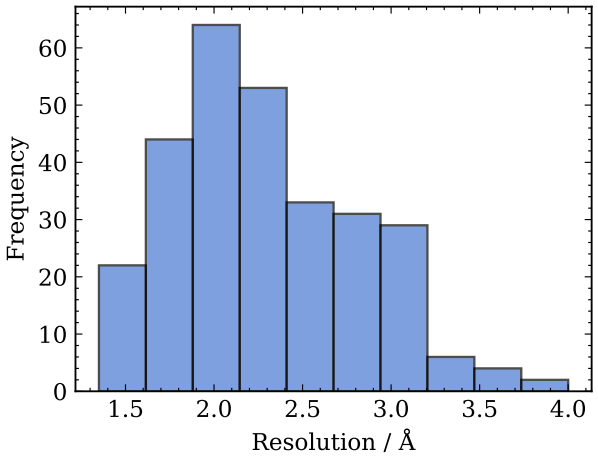


Figure 2.1: Histogram of resolutions for 288 protein-nucleic acid complexes in the X-ray diffraction molecular replacement test set, ranging from 1.35 Å to 4.00 Å.

2.3 Convolutional Neural Network for Binary Segmentation of Nucleic Acid Density

Three-dimensional convolutional neural networks have been shown to work well at segmenting molecule types in Coulomb potential maps from cryo-EM, therefore, the goal of this work is to investigate the application of these complex machine learning models to nucleic acid crystallographic density. Three convolutional neural networks were trained to be able to segment electron density into regions which represent a likely position of either the phosphate group, the sugar group or the base group of nucleic acid electron density, shown in Figure 2.2, where each model, f, can be expressed as a mapping shown in Equation 2.1, where D, W, and H represent the depth, width and height of the three-dimensional input respectively. The more rudimentary method of binary segmentation was chosen initially to prove this method had sufficient complexity to understand nucleic acid electron density.

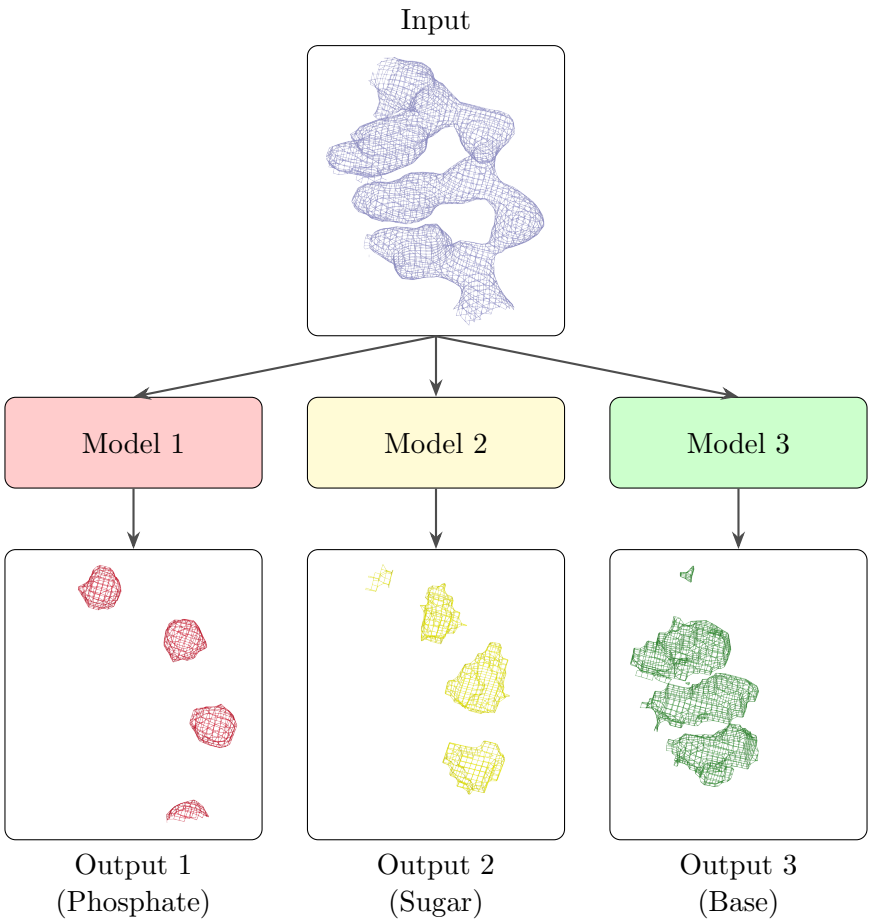


Figure 2.2: Schematic of inputs and outputs of three convolutional neural networks, which each perform binary segmentation of a given input to produce a spatially identical output corresponding to phosphate features, sugar features and base features.

$$f : \mathbb{R}^{D\times H\times W\times 1} \rightarrow \mathbb{R}^{D\times H\times W\times 2} \tag{2.1}$$

2.3.1 Neural Network Architecture

The three models created for nucleic acid binary segmentation were based on the 3D U-Net architecture with slight alterations to the normalisation functions, as shown in Figure 2.3. The model takes in a batch of input boxes, $\boldsymbol{X}$, with three spatial dimensions of length 32 and one filter dimension of length 1, denoted formally as $\boldsymbol{X} \in \mathbb{R}^{B\times 32\times 32\times 32\times 1}$ where B is the size of the batch. In crystallographic terms, the spatial dimensions represent a cubic grid of 32 points in each of the three real-space dimensions with a 0.70 Å grid spacing, where each point has a single value in the final filter dimension corresponding to the electron density value at that point. This spacing was chosen to mimic the conventional grid spacing of an average 2.10 Å crystallographic structure.

Following this input layer is a series of downsampling blocks that perform a range of operations on the input data. Each downsampling block halves the number of spatial dimensions and doubles the number of filter dimensions using a set of convolutions and pooling operations. Following a series of downsampling blocks is a series of upsampling blocks which operate oppositely, namely taking an input, doubling the number of spatial dimensions and halving the number of filter dimensions. These two sections of the model interface through the bottleneck section and via connections from each downsampling block to the corresponding upsampling block. The intuition behind such an architecture

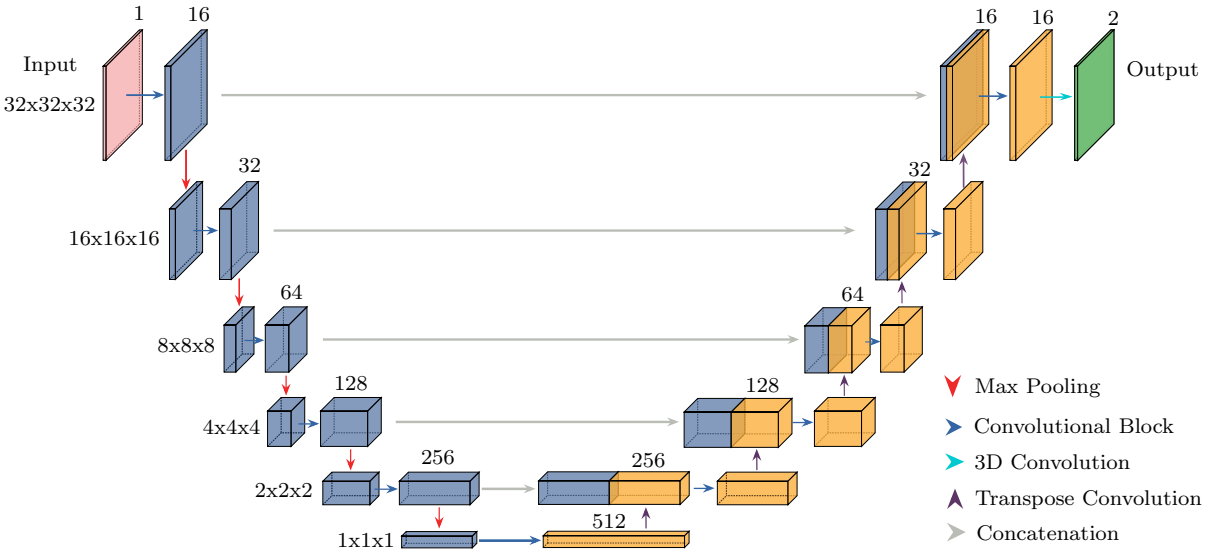


Figure 2.3: Schematic view of the three-dimensional U-Net architecture. The encoder-decoder network first downsamples the data of shape $(32, 32, 32, 1)$ to a vector form of shape $(1, 1, 1, 512)$. The vector is then upsampled back to an output of shape $(32, 32, 32, 2)$, where the two output channels represent the probability of the target group being, or not being, at each point in the output box.

is that the downsampling section of the model can conceptualise global features in the input, and the upsampling section can localise any important features and apply them to the output of the connected downsampling block.

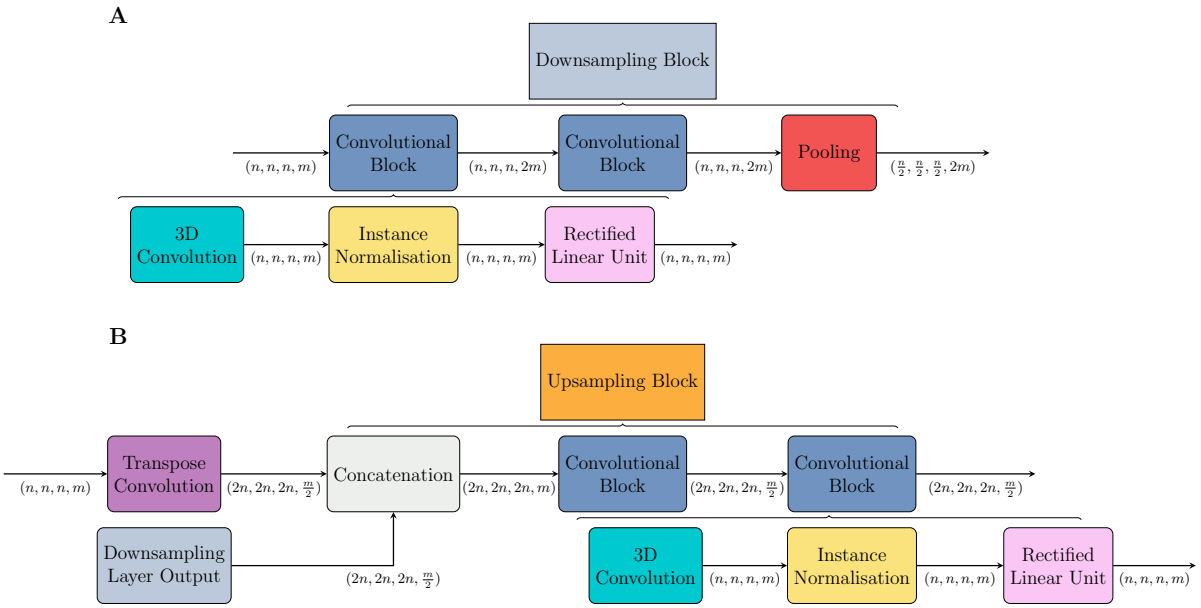


Figure 2.4: A - Schematic representation of the downsampling block which takes in a tensor of shape (n, n, n, m) and downsamples it to a tensor of shape $(\frac{n}{2}, \frac{n}{2}, \frac{n}{2}, 2m)$. Each downsampling block contains two convolutional blocks followed by a single pooling operation. B - Schematic representation of the upsampling block which takes in a tensor of shape (n, n, n, m) and converts it into a tensor of shape $(2n, 2n, 2n, \frac{m}{2})$. Each upsampling block contains a transpose convolution that doubles the input spatial dimensions, followed by a concatenation operation which combines the filter dimensions of the corresponding downsampling operation with the filter dimension of the transpose convolution. Following this are two standard convolutional blocks.

2.3.1.1 Downsampling

The downsampling portion of the model encodes the input data, $\boldsymbol{X} \in \mathbb{R}^{B\times 32\times 32\times 32\times 1}$, into a vector representation, $\boldsymbol{X'} \in \mathbb{R}^{B\times 1\times 1\times 1\times 512}$, through a series of downsampling blocks, denoted as d. The first downsampling block converts the singular value in the input filter dimension to a set of 16 values, and all subsequent downsampling blocks double the length of the filter dimension whilst halving the lengths of the spatial dimensions, shown in Equation 2.2.

$$d : \mathbb{R}^{B\times D\times H\times W\times C} \rightarrow \mathbb{R}^{B\times \frac{D}{2}\times \frac{H}{2}\times \frac{W}{2}\times 2C}. \tag{2.2}$$

where:

B is the batch size, or number of samples provided to the model at one time
D is the depth of the input
H is the height of the input
W is the width of the input

C is the number of filters, commonly referred to as channels

The downsampling block, d, consists of two consecutive convolutional blocks, followed by a max-pooling operation to reduce the spatial dimensionality. Each convolution block itself encompasses a convolution of the input to the block with some learnable kernel of size $(3 \times 3 \times 3 \times C_{in})$ with a stride of 1, followed by instance normalisation[238] and the application of the *Rectified Linear Unit*, defined in Equation 1.24. By default, a convolution operation with a kernel with spatial dimensions greater than 1 reduces the spatial dimensionality of the input, which is undesirable with many consecutive convolutional operations present in this model architecture. To rectify this, the convolution can be carried out with a zero-padded input such that the output spatial dimensions of the convolutional block match those of the input. This process is shown schematically in Figure 2.4A.

2.3.1.2 Bottleneck

The vector representation resulting from the downsampling portion of the model is further processed during the so-called *bottleneck* portion to allow for any required adjustments to the critical vector representation. The bottleneck consists of two convolutional blocks with normalisation operations removed.

2.3.1.3 Upsampling

Following the bottleneck, begins the upsampling portion of the model which decodes the vector representation, $\boldsymbol{X'} \in \mathbb{R}^{B\times1\times1\times1\times512}$, back to the exact spatial dimensions as the input to the entire model, $\boldsymbol{Y} \in \mathbb{R}^{B\times32\times32\times32\times C}$. This is accomplished through five consecutive upsampling blocks, denoted by u, which each double the lengths of the spatial dimensions and half the length of the filter dimensions of a given input, shown in Equation 2.3.

$$u : \mathbb{R}^{B\times D\times H\times W\times C} \rightarrow \mathbb{R}^{B\times 2D\times 2H\times 2W\times \frac{C}{2}}. \tag{2.3}$$

where:

B is the batch size, or number of samples provided to the model at one time
D is the depth of the input
H is the height of the input
W is the width of the input
C is the number of filters, commonly referred to as channels

The upsampling block, u, consists of a transposed convolution, which applies an independent learnable kernel, k, of size $(3 \times 3 \times 3 \times C_{in})$ to the input. This is the reverse

of what occurs during regular convolution, where the kernel is convolved across the input. This operation doubles the lengths of the spatial dimensions and halves the length of the filter dimension. The result is then concatenated with the corresponding (same spatial dimensions) output from the downsampling portion of the model. This concatenation can be thought of as aiding the localisation of any positive features in the model through comparison with the partially encoded input. The upsampling block concludes with two regular convolutional blocks that preserve the spatial dimensions and reduce the filter dimensionality to match the output of the transposed convolution. This is shown schematically in Figure 2.4B.

2.3.1.4 Output

After the final upsampling block, a convolution operation followed by a softmax activation function is applied to transform the filter dimension from a length of 16 to a length of 2 at each spatial point. These two values at each spatial point represent the classification, which can be interpreted as a probability of each class due to the application of the softmax activation function. The first value can be interpreted as the probability of the negative class existing at a specific spatial point, and the second value can be interpreted as the probability of the positive class existing at that spatial point.

2.3.2 Training

A complex model such as the 3D U-Net has a large number of parameters, $\boldsymbol{\beta}$, which control the flow of data throughout the model. These parameters are mostly present in the form of independent convolutional kernels and bias vectors applied in convolutional blocks, with some additional parameters used for normalisation. The selection of these parameters is crucial for the creation of a useful model that accomplishes a given task successfully. The process of training a model involves constantly updating these parameters to obtain the desired result. For the task of identifying nucleic acid features from electron density, a large amount of structural data in the form of density maps can be provided to the model until it exhibits good performance. The first step in this process involves obtaining a large, variable dataset.

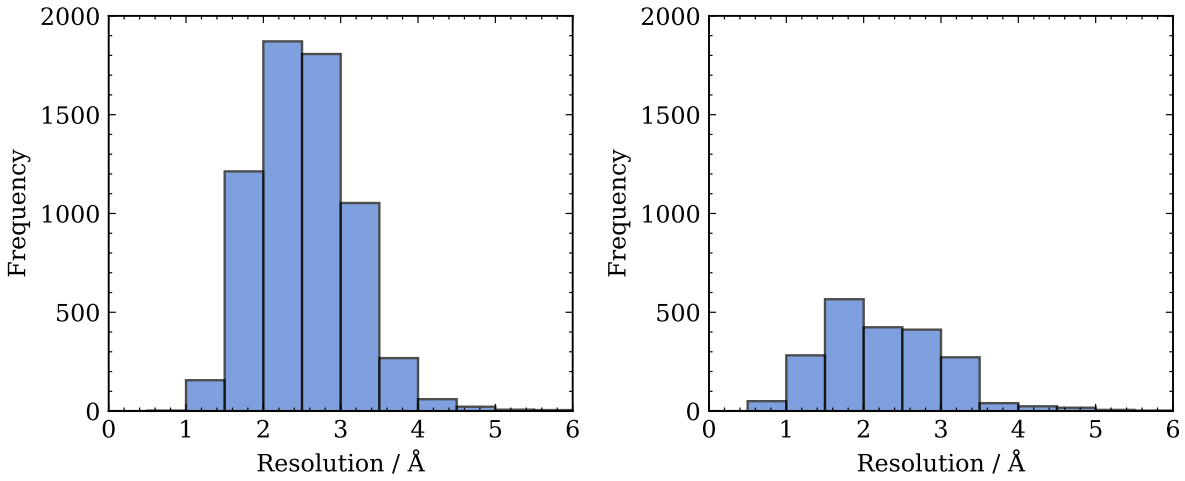
2.3.2.1 Dataset Creation

The dataset used for the model training originated from the Protein Data Bank. All structures used were obtained using X-ray diffraction, with no resolution filter applied. From this collection of nucleic acid-containing structures, 1,000 protein-nucleic acid structures were reserved for use in later unseen testing. In total, the starting dataset contained 2,711 structures comprising only nucleic acids and 8,369 structures consisting of both protein and nucleic acid polymers. Of the 2,711 nucleic acid-only structures, 1,558 were DNA,

1,119 were RNA, and 34 were DNA/RNA complexes. Of the 8,369 protein-nucleic acid structures, 5,754 were protein-DNA complexes, 2,182 were protein-RNA complexes, and 433 were protein-DNA/RNA complexes.

Maps to interpret were taken from the RCSB[239] using phases from the final deposited structure, calculated by the RCSB using *DCC*.[240] To supplement the dataset with more realistic cases, the maps of all protein and nucleic acid-containing models were recalculated to better resemble the output of molecular replacement. To achieve this pseudo-molecular replacement, all non-protein molecules were removed, and the B-factors for the remaining protein residues were set to the average B-factor value of the remaining model. This model was then refined with *REFMAC5*[241] to obtain a more realistically phased map. This map in MTZ form and the deposited model were then added to the dataset.

The protein-nucleic acid structure dataset contains structures with an average resolution of 2.51 Å, with a resolution range of 0.98 Å to 9.00 Å. The nucleic acid-only structure dataset contains structures with an average resolution of 2.24 Å, with a resolution range of 0.55 Å to 10.0 Å. Histograms of the resolutions for both datasets are shown in Figure 2.5.



(a) Protein-nucleic acid structures

(b) Nucleic acid only structures

Figure 2.5: Resolutions of structures in the dataset used for nucleic acid binary segmentation model training, consisting of both protein-nucleic acid structures and nucleic acid structures. The resolution range for both histograms is capped at 6.00 Å for clarity.

2.3.2.2 Dataset Preprocessing

The 3D U-Net is designed such that it should receive a five-dimensional input consisting of a batch of multiple input boxes, with a single value at each point, $\boldsymbol{X} \in \mathbb{R}^{B\times D\times H\times W\times 1}$. Structural data is rarely found in such a format, so the data must be preprocessed before being provided to the model during training. Additionally, it is helpful to enforce consistency in the grid spacing so that features in the input remain on the same scale across

multiple samples. To achieve this, a given density map must first be interpolated onto a regular orthogonal grid with a grid spacing of 0.70 Å.

After all density data in the dataset is interpolated, a target for the model can be devised. For the task of locating nucleic acid features, namely the phosphate, sugar, and base groups, a model that can highlight areas of density originating from a particular group may be helpful. A target map can be defined with the exact spatial dimensions as the interpolated orthogonal grid with positive areas of density within 1.5 Å of an atom from the given target group. This distance-based target function is shown in Equation 2.4 where $\mathbf{r}$ is a position vector. After target map generation, the interpolated orthogonal grid and the three target maps corresponding to the phosphate, sugar, and base targets are output in CCP4 map file format for ease of access. This yields three complete datasets originating from three sources: nucleic acid-only structures, protein-nucleic acid structures with deposited density, and protein-nucleic acid structures with density obtained through the pseudo-molecular replacement. Each dataset can then be split into a training set consisting of 80 % of samples and a test set consisting of 20 % of samples.

$$f(\mathbf{r}) = \begin{cases} 1 & \text{if } \|\mathbf{r} - \mathbf{r}_{\text{atom}}\| < 1.5\,\text{Å} \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

2.3.2.3 Training Scheme

A useful model should exhibit good performance across a range of structures and sources, and this performance is significantly affected by how a given model is trained. For example, if a model sees only samples from dataset A during the first n epochs of training, it is most likely to converge on a solution that satisfies this sample type. If a model is then provided with samples from dataset B after n epochs, the performance of the model may deteriorate for both datasets. To prevent this, it is a good idea to cycle through all possible data sources during an epoch.

During each iteration of training, a random structure is selected from the dataset corresponding to the current map source, and the map source is cycled to help the model learn information from a range of sources. Both the interpolated orthogonal grid and appropriate target grid are loaded from CCP4 map files using *GEMMI*.[242] A random rotation, $\boldsymbol{R}$, and a random translation, $\boldsymbol{T}$, can be drawn from uniform distributions, shown in Equation 2.5, where Rotation$(\mathbf{u}, \theta)$ corresponds to the 3D rotation matrix defined by the axis $\mathbf{u}$ and the angle θ. A scale matrix, $\boldsymbol{S}$, must also be applied to the rotation matrix to ensure the sample vector, $\mathcal{V}$, retains the same scale as the original grid. It should be noted that this method of random rotation is imperfect, since there is an implicit bias for rotations with smaller angles due to the properties of the surface area of a sphere. A

more robust method using randomly generated quaternions may be an avenue for future work.

$$\mathbf{T} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, \quad t_x \sim \mathcal{U}(0,1),\, t_y \sim \mathcal{U}(0,1),\, t_z \sim \mathcal{U}(0,1) \tag{2.5}$$

$$\mathbf{R} = \text{Rotation}(\mathbf{u}, \theta), \quad \mathbf{u} \sim \mathcal{U}(\mathbb{S}^2), \quad \theta \sim \mathcal{U}(0, 2\pi).$$

$$\mathbf{S} = s\mathbf{I}_3 \quad \text{where s} = 0.70 \text{ and } \mathbf{I} \text{ is the identity matrix}$$

This random transformation can then be used to interpolate the respective grid into a vector of any shape, but for model training, this must match the input to the model, $\boldsymbol{X} \in \mathbb{R}^{B,32,32,32,1}$. This interpolation is defined in Equation 2.6.

$$\mathcal{V}(i,j,k) = \mathcal{I}(\mathcal{G}, \mathbf{RS}\boldsymbol{x}_{ijk} + \mathbf{T}), \quad \forall i,j,k \in [0,31] \tag{2.6}$$

where:

$\mathcal{V}$ is the sampled vector
$\mathcal{I}$ is the trilinear interpolation function
$\mathcal{G}$ is the interpolated orthogonal grid
$\boldsymbol{x}_{ijk}$ is the index in the reference frame of the sample

A batch of inputs can then be created using this sampling procedure, one-hot encoded and fed to the model. During initial trials, it was noted how the model had limited success when the sampling procedure had no restrictions on the amount of target visible per sample. The model, observing no nucleic features during the first part of training, learnt to predict everything as the absence of nucleic features. To prevent this, during the first 200 epochs of training, a restriction was placed on the sampling procedure, shown in Equation 2.7. After 200 epochs, this restriction was removed, and any sampled vector was allowed to be used in training to fine-tune the model to better emulate what would be encountered during inference.

$$\sum_{i=1}^{32}\sum_{j=1}^{32}\sum_{k=1}^{32} \mathcal{V}(i,j,k) > 0. \tag{2.7}$$

2.3.2.4 Infrastructure

The network was trained using the TensorFlow library in Python.[243] The network was trained for 1,000 epochs with 1,000 steps per epoch and 8 samples per batch. Training was run on a single NVIDIA A40 GPU and took approximately 60 hours per model. The sigmoid focal cross-entropy loss function[244] and Adam optimiser were used for all

three networks. This loss function is critical for the success of training these three models since it accounts for the class imbalance present in this application. Focal cross-entropy loss applies a modulating factor to the calculation of the cross-entropy loss such that misclassified positive classes contribute more to the loss function than correctly classified negative classes, where the positive class is the minority. To optimise training speed, most of the dataset preparation was precomputed to minimise the time spent generating samples between batches. In addition, the TensorBoard package was used to optimally match the floating-point calculations to the hardware available.

2.3.3 Inference

Once a model has been trained and an updated set of parameters, $\boldsymbol{\beta}$, is known, the model can be used to make predictions on previously unseen examples. It is important to consider that the performance of a given model depends heavily on the format and type of data it has previously seen. To obtain the best results during inference, the preprocessing steps used during training should be replicated.

When processing the density of a new potentially nucleic acid-containing structure, the grid must first be interpolated onto an orthogonal grid with a grid spacing of 0.70 Å. This regular grid can then be split into segments, with a length of 32 in each spatial dimension and the density value in the filter dimension, $\boldsymbol{X} \in \mathbb{R}^{32\times32\times32\times1}$. Each segment can be taken such that there is an overlap of 16 points between consecutive segments, and a margin consistent with this overlap can be added to the orthogonal grid so that every part of the orthogonal grid can be predicted by the model multiple times.

Each segment is then passed to the model alone and the probabilistic output of the model, $\boldsymbol{Y} \in \mathbb{R}^{1\times32\times32\times32\times2}$, can be trivially converted to a classification using the argmax function, defined in Equation 1.18. Once each segment has been predicted and processed, each segment can be placed into an output grid with the exact spatial dimensions of the regular orthogonal grid, and a pointwise average can be calculated. The reverse process of the initial orthogonalisation using trilinear interpolation yields a grid with the same unit cell dimensions as the original input. Since the argmax function was used for classification and the final output was interpolated, the resulting points may be non-integral in the range 0 to 1.

2.3.3.1 Uncertainty Estimation

When attempting to infer meaning from the predictions of any machine learning model, it is often helpful to understand the relevant uncertainty of the prediction. From this 3D U-Net architecture, uncertainty can be calculated or obtained in two distinct ways. The first method is to calculate variance across multiple predictions, and the second is

to obtain uncertainty directly from the model.

Variance calculations are possible since every point in the input grid has been processed and predicted by the model multiple times, therefore a point-wise variance can be calculated to infer some meaning from the agreement between differing predictions from the model. Points with a high variance indicate a larger disagreement, which may be inferred as a lower confidence in the output at a given point.

Alternatively, the confidence of a prediction at any point can be directly obtained from the output of the model. During standard inference, the argmax function is applied to each point in the output to infer a classification from the binary probability. If the probability of the positive class is greater than the probability of the negative class, the output value is set to 1, and the alternative is set to 0. However, this presents an opportunity to obtain a confidence metric directly by skipping the application of this argmax function. If instead, the raw probability of the positive class is used in the output for each point, then each point represents a confidence from the model at that point.

2.3.4 Results and Discussion

The result of inference using all three trained models is shown in Figure 2.6 for a protein-DNA complex solved to a resolution of 2.80 Å. The input map was generated through molecular replacement using a homologous protein search model (PDB code: 1HFO[245]). Molecular replacement was performed with *PHASER*[237] and the molecular replacement solution model was refined to the experimental data with *REFMAC5*. The resultant free-R factor was 0.469, indicating the structure was not yet solved and more atoms may need to be placed correctly. This example represents a procedure commonly encountered during structure solution of a protein-nucleic acid complex. It is therefore a good candidate for visually assessing the performance of the three binary segmentation models. This particular protein-DNA complex was not included in the training dataset for the training of the deep learning models, but two structures containing homologous protein models with bound DNA were. To what extent this biases the result cannot be rationalised since training was done at random, but this example still serves the purpose of contextualising the output predictions in a realistic case.

The $2mF_o - DF_c$ density corresponding to the missing nucleic acid, shown in black at 1.5 σ in Figure 2.6a, clearly resembles a characteristic B-conformer DNA duplex. Although this is somewhat clear to the eye, obtaining such an assessment entirely computationally is difficult. Only 69 % of the scattering matter in the deposited model is available after molecular replacement to be used in the calculation of the $2mF_o - DF_c$ map. This causes an associated error in the phase of each experimental reflection, such that the density

map in the area of the unbuilt nucleic acid is noisy and discontinuous. This is especially evident in regions farther from the protein, such as the bottom portion of the unbound nucleic acid. Existing algorithms which intend to locate nucleic acid features heavily rely on small-scale features resembling known nucleic acid electron density, however, when density is noisy or difficult to interpret, these algorithms may struggle. An approach relying on a deep learning method may be beneficial, as the model can encode larger-scale context that may inform a more localised understanding of features in real space. For example, visually inspecting a DNA duplex and the regular trends in the backbone can inform a human of where a nucleic acid should be, even when the density is poor. Analogously, a deep learning model may infer a similar understanding by taking in a larger input context enclosing multiple nucleic acid residues, which may lead to better feature-finding performance than traditional algorithms.

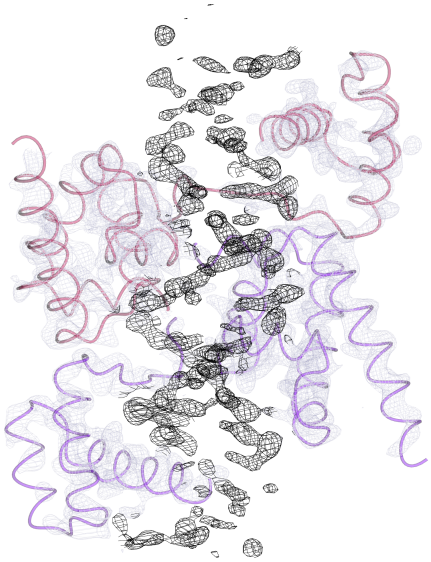
The predicted map from the phosphate model is shown in Figure 2.6b and, in a similar way to the $2mF_o - DF_c$ density, this prediction resembles a B-conformer DNA duplex. The positive areas in this predicted map are localised in regions of density that correspond to atoms in the phosphate group of the nucleic acid and are well separated from each other. This indicates that the phosphate model is performing well on this example, as virtually all phosphate positions are highlighted in the predicted output. This is especially promising, given the poor, noisy input density supplied to the model in the region near the bottom of the nucleic acid helix. It is possible to infer from this that the model has been trained to understand the global relationship between the input density and the phosphate features, regardless of the local, small-scale shape of each phosphate feature. In this example, the phosphate model seems to be able to recall the majority of the nucleic acid features, however, a small number of predicted features lie inside or close to the electron density of the protein and are therefore false positives. False positives, like false negatives, are undesirable and can lead to confusion when attempting to infer biochemical meaning.

In a similar way to the phosphate prediction, the prediction from the sugar model corresponds well to the B-conformer DNA duplex of this protein-DNA complex, shown in Figure 2.6c. The areas highlighted by the sugar model generally correspond well to the atomic positions defined in the deposited model, with predicted regions typically much larger than those of the predicted phosphate positions. This is unsurprising since the training target of the model consisted of highlighting points within 1.5 Å of any atom corresponding to the furanose ring. Consequently, the predicted areas are larger, and in some instances, neighbouring areas of highlighted density may overlap. While this can make it more challenging to interpret, predicted areas remain correct even in areas of poor and noisy input density which confirms the good performance of this model for this

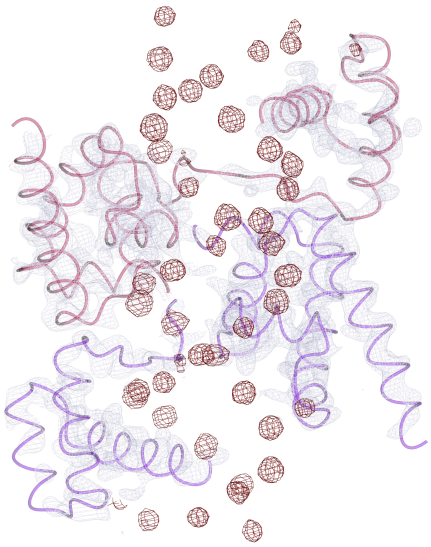
example.

After protein molecular replacement of a protein-DNA complex, it is common to see some structured electron density for the sugar-phosphate backbone, but it is relatively uncommon to see well-defined density for the stacked base pairs. Despite this, the predictions from the base model, shown in Figure 2.6d, resemble the stacked base pairs with a slight turn as expected for the B-conformer of DNA. There is little separation between the predictions of the base groups from the base model, which may make it difficult to segment each base group individually. As with any model, this characteristic is likely due to the data provided to the model during training, and since the average inter-base distance in nucleic acid molecules is 3.4 Å,[246] highlighting positions within 1.5 Å of any base atom likely causes this characteristic.

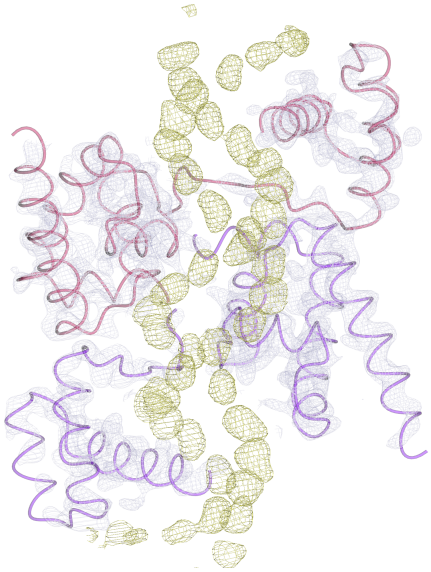
It is evident that the three binary segmentation models work well for nucleic acid feature identification on this protein-DNA complex example. To ensure this performance is not an anomaly and is consistent across a range of samples, predictions were run for all molecular replacement solution maps from a test set of 288 protein-nucleic acid examples, described in Section 2.2. The standard machine learning metrics of accuracy, precision, recall and F1 score were calculated by comparing the output of each model with a calculated target map consistent with what was provided to the model during training, shown in Table 2.1 and Figure 2.7. These metrics are defined in Section 1.5.1.4.2, and can help describe the performance of a given model. The calculated accuracy for most molecular replacement solution maps in this test set was evaluated to be over 98 %. This value indicates all models are performing outstandingly with near-perfect accuracy, however this statistic is significantly misleading with this type of input. The majority of space sampled by these models corresponds to areas where none of the target nucleic acid atoms are present, i.e. areas of protein, solvent or other biological molecules. Since accuracy measures the proportion of correctly predicted samples, both positive and negative, when the ratio of negative to positive samples is very high, the accuracy tends toward 100 %. Further analysis deliberately excludes accuracy as a metric to prevent misinterpretation and instead focuses on precision and recall, which are both primarily dependent on positive predictions.



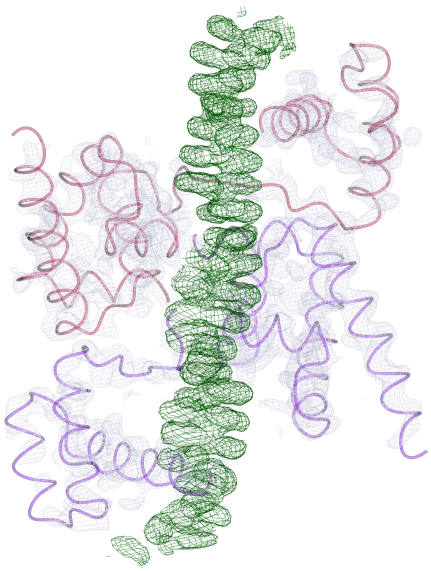
(a) Input density



(b) Predicted phosphate density



(c) Predicted sugar density



(d) Predicted base density

Figure 2.6: Output of all three deep-learning models corresponding to phosphate group, sugar group and base group predictions. To generate the input density, the deposited structure factors were collected from the Protein Data Bank for a POU DNA binding domain resolved to 2.80 Å (PDB code: 3L1P[247]). Molecular replacement was performed using a homologous model (PDB code: 1HFO[245]). The placed model was refined with *REFMAC5*, resulting in a free-R factor of 0.469. The $2mF_o - DF_c$ input density, shown in black at 1.5 σ, outlines a characteristic B-DNA duplex. The $2mF_o - DF_c$ electron density is noisy and discontinuous, which can often cause automated model-building software packages to struggle with locating features. The three trained binary segmentation models can identify the phosphate, sugar and base positions well from the input density, underscoring the potential usefulness of these models as a post-molecular replacement tool.

The precision, recall, and F1 scores for all three binary segmentation models are reasonable, indicating successful training and potential utility. In particular, the models exhibit excellent recall metrics but poorer precision metrics, suggesting that while the models often predict most of the expected nucleic acid features correctly, there are a number of additional false positive predictions. Achieving both high recall and high precision metrics is an inherent difficulty when predicting such a large number of data points (32^3) per prediction, and a trade-off is often seen between precision and recall. Nevertheless, it is important to remember the purpose of the network, to identify nucleic acid features, therefore a more informative metric would be to calculate how many of the target groups were within the correctly segmented density. An atom inclusion metric can be measured by the proportion of atomic positions that sit in positive predicted density output from the corresponding model, defined in Equation 2.8. A high atom inclusion score indicates that the most features are correctly located within the predicted density.

$$\text{Atom Inclusion} = \frac{\sum_i^N f(\mathbf{r_i})}{N} \tag{2.8}$$

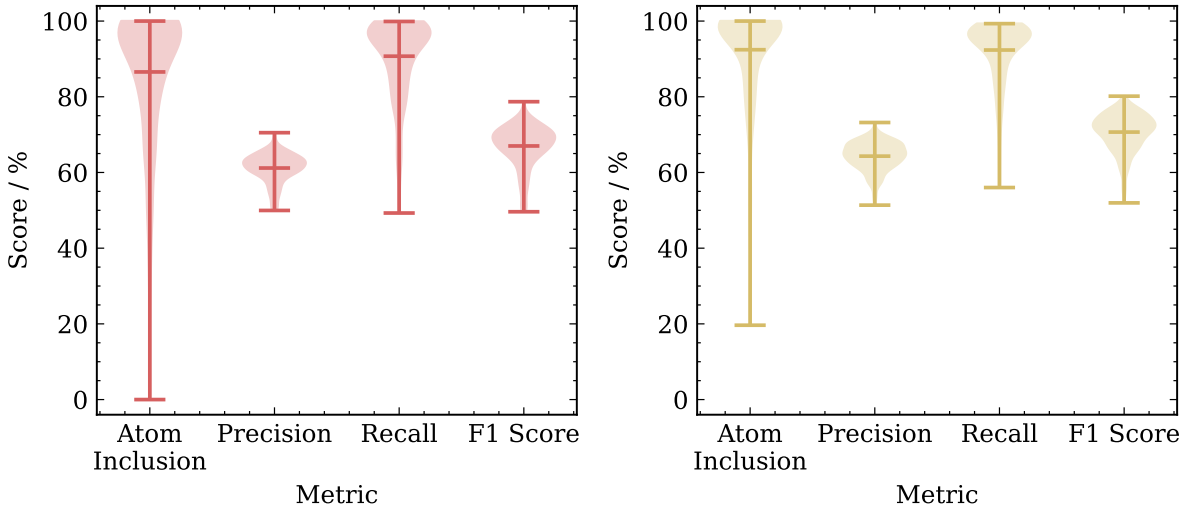
where:

$$f(\mathbf{r}) = \begin{cases} 1 & \text{if } \rho_{\text{pred}}(\mathbf{r}) > 0 \\ 0 & \text{otherwise} \end{cases}$$

The atom inclusion scores for all models are above 86 % on average over the molecular replacement test set, indicating that despite the potential issues with the quality of the $2\text{mF}_\text{o} - \text{DF}_\text{c}$ density map, these three models can still pick out nucleic features well. However, the relatively low precision metrics across all three models highlight an area for improvement.

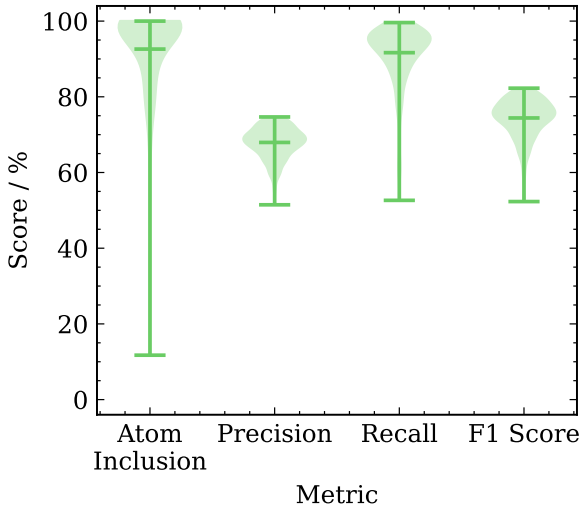
Table 2.1: Model metrics calculated as an average from a test set of 288 real molecular replacement solution maps. Uncertainty here is represented as the standard deviation across the samples.

Model	Atom Inc. / %	Precision / %	Recall / %	F1 / %
Phosphate	86.6 ± 17.8	61.2 ± 3.7	90.7 ± 10.1	67.0 ± 5.3
Sugar	92.4 ± 12.9	64.3 ± 3.6	92.4 ± 8.0	70.7 ± 4.7
Base	92.6 ± 13.2	67.9 ± 4.0	91.7 ± 7.4	74.4 ± 5.0



(a) Phosphate model

(b) Sugar model



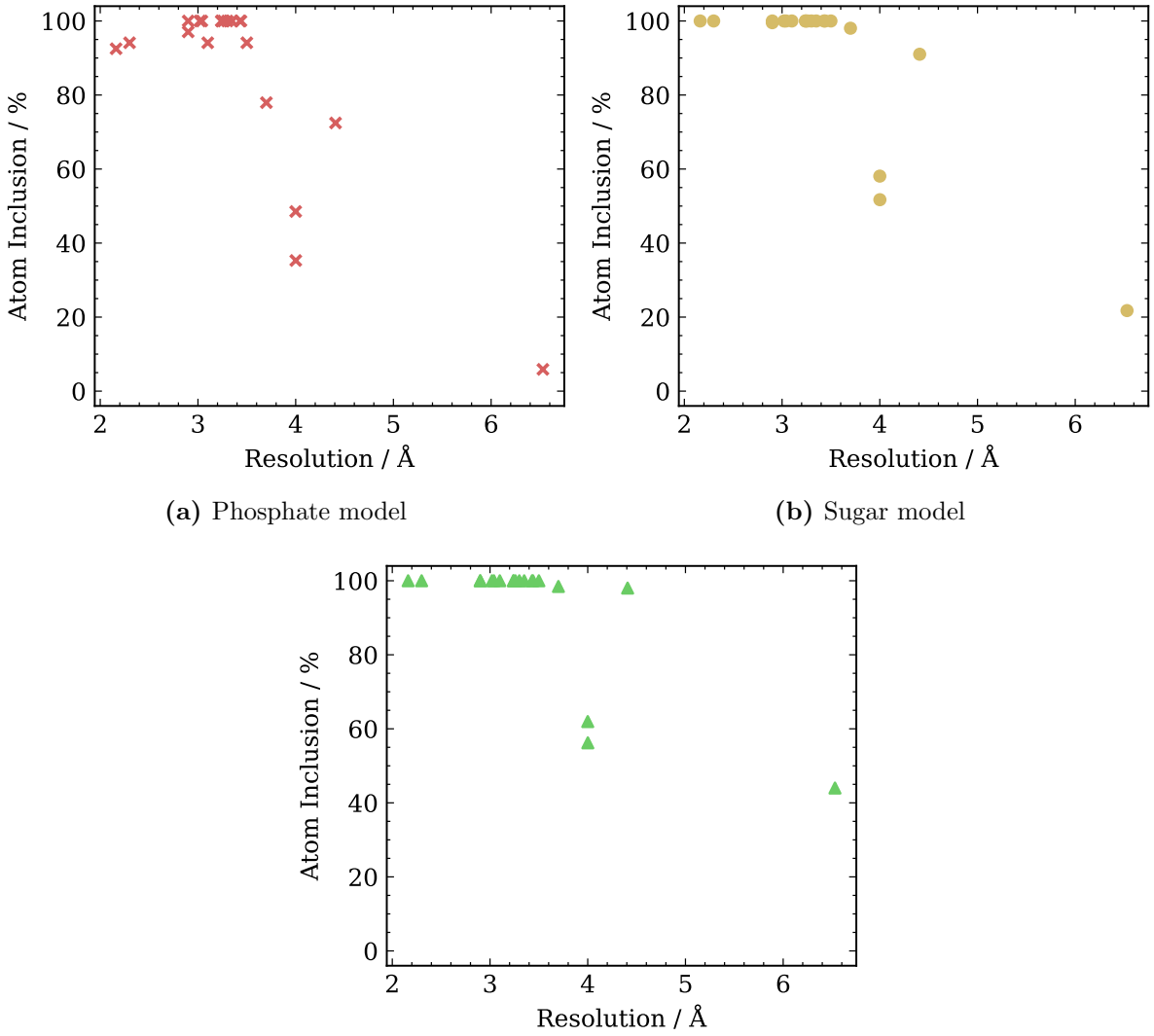
(c) Base model

Figure 2.7: Violin plot showing atom inclusion, precision, recall and F1 score calculated across predictions from all three binary segmentation models using 288 real molecular replacement solution maps as inputs. Bars represent the maximum, median and minimum score for a given category and the shaded area represents the proportion of data at a given point.

2.3.4.1 Resolution Dependence

A method that can reliably locate nucleic acid features from electron density should perform well across a range of resolutions commonly encountered during structure solution. To test the dependence on experimental resolution for each of the three models, predictions were run on 20 DNA-bound DNA topoisomerase proteins in the Protein Data Bank. These 20 structures were excluded from deep learning model training, and were resolved at varying resolutions between 2.11 Å and 6.35 Å, shown in Figure 2.8. Pseudo-molecular replacement maps were generated by calculating a $2mF_o - DF_c$ map using the experimental reflections and only the protein portion of the protein-DNA complex, with B-factors set to the average of the deposited protein.

The results indicate that all three models can locate nucleic acid features in structures with an experimental resolution better than 4 Å. Similar to the molecular replacement test set, the sugar and base models often outperform the phosphate model. It is promising to see that the sugar and base models can pick out features even at very low resolutions that were very uncommon in the training set. These results suggest that the trained models have some understanding of the general structure and topology of the sugar and base electron density, which may be helpful in downstream processing applications.



(a) Phosphate model

(b) Sugar model

(c) Base model

Figure 2.8: Atom inclusion scores of 20 predictions of DNA-bound DNA topoisomerase structures deposited in the Protein Data Bank with resolutions from 2.11 to 6.35 Å. Input maps to each model were calculated using only the protein portion of the protein-nucleic acid complex to emulate molecular replacement.

2.3.5 Conclusions

In conclusion, the application of a binary segmentation convolutional neural network for locating nucleic acid electron density features was successful. Features can now be located with relative certainty across a range of structures and resolutions. The three models output three predicted maps, which could serve as a guide for interactive or automated model building. However, the relatively high likelihood that the model will produce false positive results is certain to cause a degree of confusion in any downstream application. In addition, the three models all perform vaguely similar tasks, and having individual models with independent sets of parameters suggests that some of these parameters are likely redundant. A single model that performs similarly to these three may be more desirable from a usage and efficiency standpoint.

Chapter 3

Optimisation of Convolutional Neural Networks for Segmentation of Nucleic Acid Density

The binary segmentation models described in Chapter 2 highlight the promise of this method for identifying nucleic acid features in experimental density. While the performance of this set of binary segmentation models was acceptable, the lack of precision exhibited by these models limits direct application. It can be hypothesised that this poor performance arises from the independence of each model, where no parameters are shared. This may create a disconnect between each nucleic acid feature and reduce the importance of environmental context. For example, the phosphate model has been trained to understand only phosphate features and is likely to have little knowledge that a sugar ring should explicitly exist between two adjacent phosphate positions. A model that understands this spatial context should, in theory, perform more precisely.

To encode this additional context, the binary segmentation model could be designed with multiple output classes, implicitly allowing parameter sharing that could increase precision and efficiency. Training and optimising such a model requires significant resources, therefore before training, it is pertinent to consider other potential areas for performance improvement or alternate applications of these methods. One such application would be to apply these techniques to cryo-EM Coulomb potential maps, which are generally similar to those obtained in X-ray crystallography.[248] A single model that can locate features of the phosphate, sugar, and base groups, while being indifferent to the source of density, would be helpful for downstream software methods.

To investigate the feasibility of this approach, a single multiclass classification model, f, can be created as defined in Equation 3.1 which takes a single input and identifies the nucleic acid features directly in the output, as shown in Figure 3.1. This model can

be trained on both X-ray crystallographic electron density maps and cryo-EM Coulomb potential maps in an attempt to create a unified method.

$$f : \mathbb{R}^{D \times H \times W \times 1} \rightarrow \mathbb{R}^{D \times H \times W \times 4} \tag{3.1}$$

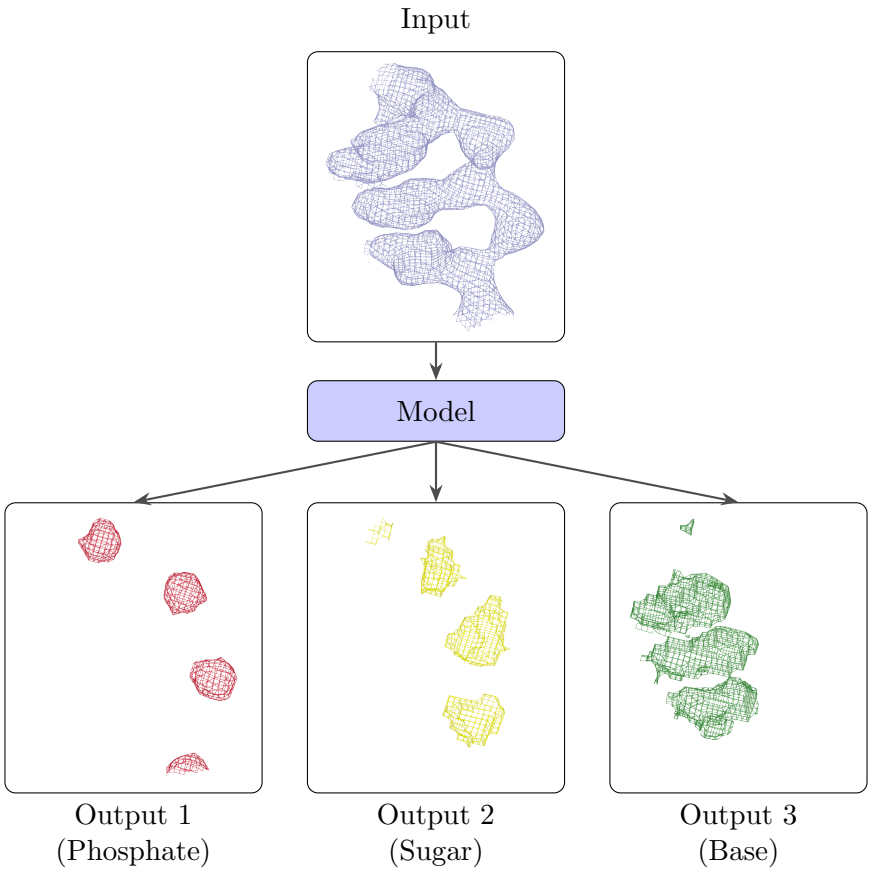


Figure 3.1: Schematic of inputs and outputs of a single multiclass convolutional neural network, which performs multiclass segmentation of a given input to produce a spatially identical output, corresponding to phosphate features, sugar features and base features.

3.1 Baseline Multiclass Segmentation Model

3.1.1 Neural Network Architecture

A baseline multiclass segmentation convolutional neural network can be created using an architecture almost identical to the binary segmentation models defined in Section 2.3.1. The only required change is to the final output layer of the model, which should output 4 classes rather than just 2. To accomplish this, the final upsampling block can be altered such that the output is convolved with 4 filters instead of the 2 filters in the binary segmentation model, creating 4 output channels per spatial point. After application of the standard softmax activation function, these channels represent the probability that a given point represents: no feature, a phosphate feature, a sugar feature, and a base feature, respectively. Additionally, the binary sigmoid focal cross-entropy loss function used in binary segmentation must be replaced with a multiclass sigmoid focal cross-entropy loss function to allow a multiclass model to train.

3.1.2 Training

The dataset for multiclass segmentation was regenerated from scratch using a process similar to that of the binary segmentation model. This was done to collect new samples from the Protein Data Bank that have been deposited since the initial method, and to include cryo-EM data. For X-ray diffraction data, nucleic acid-containing structures from the Protein Data Bank were collected, as described in Section 2.3.2.1, yielding 8,444 protein-nucleic acid complexes and 2,595 nucleic acid-only structures. In the time between dataset generation for the binary segmentation model and the multiclass segmentation model, electron density maps calculated using the software package *DCC* were removed from the Protein Data Bank, therefore maps were recalculated using the deposited structure factor intensities or amplitudes and the deposited model using *REFMAC5*. These recalculated maps were then orthogonalised and interpolated before the generation of a single target map. This target map contained the value 0 at spatial points where there are no nucleic acid features, the value 1 when the spatial point is within 1.5 Å of a phosphate group atom, the value 2 when within 1.5 Å of a sugar group atom, and the value 3 when within 1.5 Å of a base group atom. If any spatial point overlapped two regions, the lower value was selected. These values were chosen to allow for efficient one-hot encoding before training.

To supplement the electron density map dataset, cryo-EM Coulomb potential maps with a fit model containing any nucleic acid were obtained from the Electron Microscopy Data Bank.[249] A resolution filter of 5 Å was applied to remove very low-resolution maps that could pollute the dataset. This yielded Coulomb potential maps for 3,046 protein-nucleic

acid complexes and 50 nucleic acid-only structures. The overall map resolution for this dataset ranges between 1.67 Å and 4.98 Å, with an average of 3.34 Å. After collection, all Coulomb potential maps were trimmed using *Servalcat* to remove the large unnecessary regions surrounding the molecule of interest. These maps were then processed identically to the electron density maps, yielding a source and a single target map.

After the data is prepared, a model can be trained using a similar training scheme to the binary segmentation model. The model is first trained for 20 epochs with restricted sampling, then for 20 epochs without restrictions. The training length was reduced 10-fold due to the observation that the loss value would plateau after 15 epochs of training, indicating that the parameters would not update further and that additional training would be fruitless.

3.1.3 Cryo-EM Test Set

An unseen test set of cryo-EM charge maps was created to evaluate the performance of any segmentation model at identifying nucleic acid features from cryo-EM. 50 entries from the cryo-EM density dataset were randomly selected and removed from training to prevent bias. This relatively small number of entries was chosen to allow validation to be completed on a reasonable time scale. The distribution of the overall resolution of the 50 cryo-EM Coulomb potential maps is shown in Figure 3.2. 22 of these structures were protein-DNA complexes, and 28 were protein-RNA complexes.

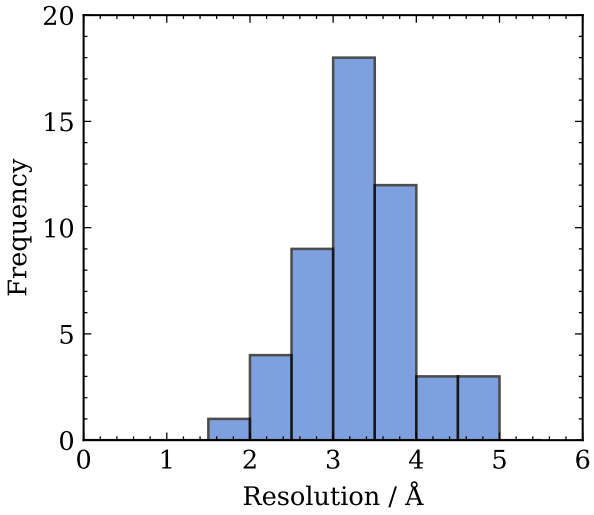


Figure 3.2: Histogram of overall resolutions of 50 cryo-EM Coulomb potential maps randomly selected for use in an unseen test set, designed to evaluate the performance of any deep learning model at identifying nucleic acid features in cryo-EM Coulomb potential maps.

3.1.4 Inference

The inference procedure for a multiclass segmentation model is similar to that of the binary segmentation model. However, the model output, $\boldsymbol{Y}$, must be processed differently to isolate each predicted class for a given spatial point. For each possible class, spatial points which match the class value are masked with the value 1, with the value 0 elsewhere, as defined in Equation 3.2.

$$\boldsymbol{Y}_c(i,j,k) = \begin{cases} 1 & \boldsymbol{Y}(i,j,k) = c \\ 0 & \text{otherwise} \end{cases} \quad \forall c \in \{0,1,2,3\} \tag{3.2}$$

These four masked segments can be used in the point-wise averaging and re-interpolation steps described in Section 2.3.3.

3.1.5 Results and Discussion

The performance of this baseline multiclass segmentation model across the 288 molecular replacement examples described in Section 2.2 is shown in Table 3.1a. The precision metrics for the outputs of the multiclass segmentation model are higher than the precision metrics from all three binary segmentation models. This suggests the hypothesis of combining the parameters of three models into a single model to aid spatial context is correct. However, this increase in precision does not come in isolation, instead, the performance of the model in recall and atom inclusion decreases. These metrics describe a model that is unlikely to successfully identify all nucleic features, but any features that are identified are likely to be correct. Despite this, the objective performance of this model for molecular replacement electron density maps, as measured by the F1 score, is better across all nucleic acid features for the multiclass segmentation model compared to the binary segmentation models.

The promising performance of the multiclass segmentation model for crystallographic maps is not mirrored when applied to cryo-EM Coulomb potential maps. The results of the baseline multiclass segmentation model across the 50 cryo-EM examples reflect a model with poor performance and little utility, shown in Table 3.1b. The low atom inclusion scores may be partly attributed to inaccuracies or bias in the deposited electron microscopy models, but the majority of the poor performance is likely due to difficulty in interpreting the lower-resolution data in this cryo-EM test set.

Based on these results for the baseline multiclass model, the method does not yet yield enough performance in identifying nucleic acid features to be useful for downstream processing. Commonly, when the performance of a model is inadequate, either the model architecture requires re-evaluation or the capacity of the model must be extended. Since

this method performs well with crystallographic maps, the architecture is likely reasonable, and so the capability of the model should be expanded. With convolutional neural networks, the simplest, and often most effective way, to accomplish this is to increase the number of convolutional filters per layer.

Table 3.1: Atom inclusion, precision, recall and F1 score metric results for the binary segmentation models and the baseline multiclass segmentation model. Metrics were calculated as averages across an X-ray diffraction test set phased with molecular replacement and a cryo-EM Coulomb potential map test set. Uncertainty here represents the standard deviation across the samples.

(a) X-ray diffraction molecular replacement test set

Model	Output	Atom Inc. / %	Precision / %	Recall / %	F1 Score / %
Binary segmentation	Phosphate	86.6 ± 17.8	61.2 ± 3.7	90.7 ± 10.1	67.0 ± 5.3
	Sugar	92.4 ± 12.9	64.3 ± 3.6	92.4 ± 8.0	70.7 ± 4.7
	Base	92.6 ± 13.2	67.9 ± 4.0	91.7 ± 7.4	74.4 ± 5.0
Baseline multiclass segmentation	Phosphate	72.2 ± 27.2	72.7 ± 5.4	81.2 ± 13.4	75.0 ± 7.9
	Sugar	72.5 ± 25.4	79.0 ± 5.0	78.7 ± 12.0	77.3 ± 8.4
	Base	73.3 ± 28.2	81.2 ± 5.9	79.3 ± 12.9	78.3 ± 9.8

(b) Cryo-EM Coulomb potential map test set

Model	Output	Atom Inc. / %	Precision / %	Recall / %	F1 Score / %
Binary segmentation	Phosphate	28.5 ± 24.7	55.3 ± 6.2	60.4 ± 9.9	56.1 ± 6.7
	Sugar	49.1 ± 31.0	56.8 ± 7.1	67.8 ± 13.5	58.5 ± 8.1
	Base	67.3 ± 28.7	59.4 ± 7.3	78.4 ± 14.1	63.2 ± 9.4
Baseline multiclass segmentation	Phosphate	27.9 ± 28.3	65.4 ± 12.1	59.8 ± 11.3	60.1 ± 10.0
	Sugar	34.3 ± 30.5	74.1 ± 12.9	60.9 ± 11.6	62.8 ± 11.1
	Base	50.1 ± 31.3	76.1 ± 10.4	68.6 ± 14.1	69.0 ± 11.5

3.2 Effect of Increasing the Number of Convolutional Filters

A convolutional neural network can locate features in an input with translational invariance by applying multiple convolutional filters to an input. Theoretically, each convolutional filter attempts to match a different characteristic of the input and when combined, can determine whether a feature is present. By extension of this theory, applying more filters in the model should allow for a more nuanced understanding of a given feature.[250] The 3D U-Net model architecture is implemented such that for a given layer, l, the number of convolutional filters applied at that layer is defined by Equation 3.3. This indicates that the total number of convolutional filters can simply be controlled by altering

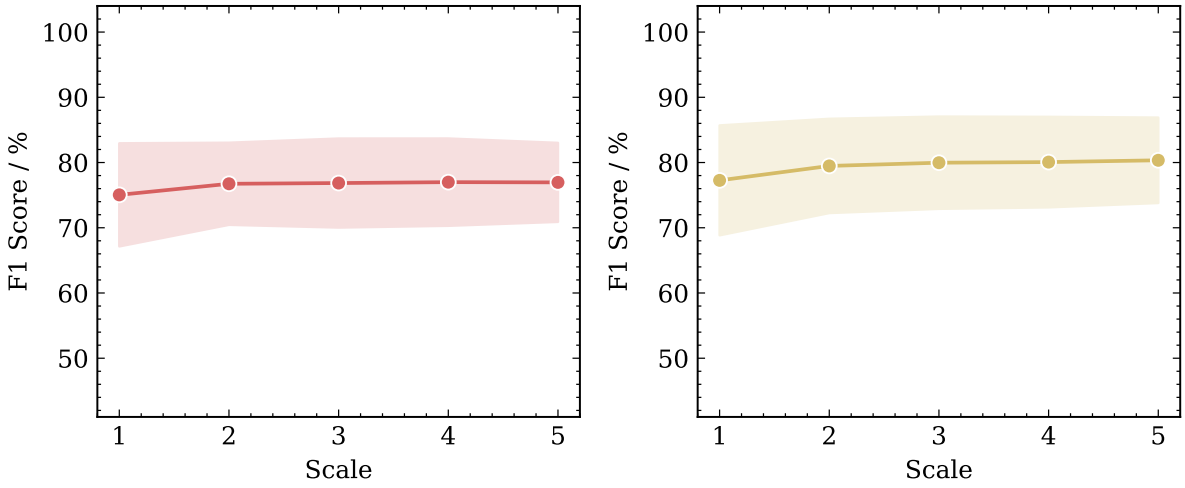
the number of filters in the initial convolutional layers, N_0^{filters}.

$$N_l^{\text{filters}} = N_0^{\text{filters}} \cdot 2^l \tag{3.3}$$

In the baseline multiclass segmentation model and the binary segmentation models, $N_0^{\text{filters}} = 16$. To investigate whether increasing the number of convolutional filters in each layer of the convolutional neural network improves performance, the number of initial filters can be scaled. It is hypothesised that adding more convolutional filters will cede more performance to the model, but at some point, the additional parameters may become redundant, leading to little performance increase. Many additional filters may even cause the model to overfit, harming performance. Four additional multiclass segmentation models were created with a scale ranging from 2 ($N_0^{\text{filters}} = 32$) to 5 ($N_0^{\text{filters}} = 80$). These new models were then trained identically to the baseline multiclass segmentation model, with results shown in Table 3.2.

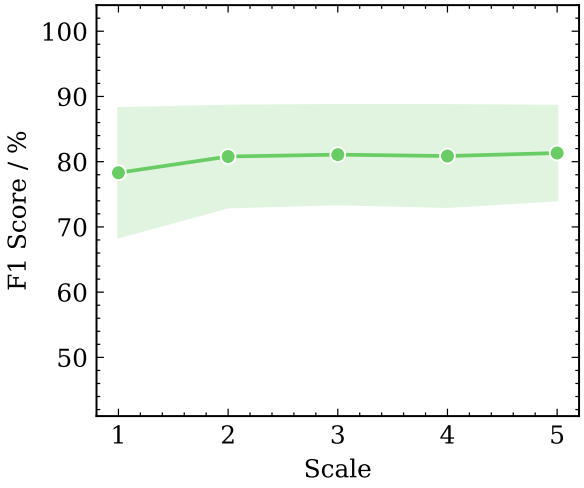
As expected, increasing the number of initial convolutional filters yields better average performance than the baseline multiclass model. Almost all metrics for models with a scale larger than 1 evaluate higher than those of the baseline model, indicating that the initial number of convolutional filters is one of the limiting factors of the performance of the baseline model. However, these results show that the performance increases observed when comparing the baseline model to the model with a scale of 2 do not continue linearly as the scale increases. The F1 scores of the scaled models across the molecular replacement test set are shown in Figure 3.3, where a plateau can be observed after the model with a scale of 2. This trend is incredibly informative, as it reveals that the model architecture can identify nucleic acids with a minimum scale of 2, and subsequent additional parameters yield little added performance. It is likely these additional parameters are ignored or the training process cannot extract any further relevant features.

While the average metrics for the model of scale 2 are larger than the baseline model, it is important to verify statistically that these differences are significant. The first step in accomplishing this is to determine whether the results are parametric, thereby indicating which statistical test to use. A Shapiro-Wilk test was run on the atom inclusion, and F1 score results for each model, and all models were evaluated to have $P \approx 0$, indicating that they were not normally distributed,[251] as expected for a randomly selected test set. Following this, a non-parametric Wilcoxon signed-rank test was run on the paired data between each scaled model and the baseline model.[252] The Wilcoxon signed-rank test is an appropriate choice in this instance, as it assesses statistical significance for non-parametric paired data and is commonly used to compare two machine learning models on the same test set.[253] The results of these statistical tests are shown in Table 3.3.



(a) Phosphate output

(b) Sugar output



(c) Base output

Figure 3.3: F1 score results for models with varying numbers of initial convolutional filters expressed as a scale to the baseline multiclass segmentation model. Shaded regions represent the standard deviation across the 288 molecular replacement test set examples. A performance plateau above a scale of 2 can be observed from these results, suggesting that further additional convolutional filters do not improve model performance.

Table 3.2: Atom inclusion, precision, recall and F1 score metric results for a range of models with varying numbers of initial convolutional filters expressed as a scale to the baseline multiclass segmentation model. Metrics were calculated as averages across a crystallographic test set phased with molecular replacement and a cryo-EM Coulomb potential map test set. Uncertainty here represents the standard deviation across the samples.

(a) X-ray diffraction molecular replacement test set

Model	Output	Atom Inc. / %	Precision / %	Recall / %	F1 Score / %
Scale 1	Phosphate	72.2 ± 27.2	72.7 ± 5.4	81.2 ± 13.4	75.0 ± 7.9
	Sugar	72.5 ± 25.4	79.0 ± 5.0	78.7 ± 12.0	77.3 ± 8.4
	Base	73.3 ± 28.2	81.2 ± 5.9	79.3 ± 12.9	78.3 ± 9.8
Scale 2	Phosphate	78.3 ± 23.1	73.3 ± 3.3	84.6 ± 12.1	76.7 ± 6.3
	Sugar	79.9 ± 22.3	78.6 ± 3.9	82.9 ± 11.2	79.5 ± 7.2
	Base	81.1 ± 23.4	80.6 ± 4.2	83.4 ± 11.2	80.8 ± 7.7
Scale 3	Phosphate	77.1 ± 24.1	73.9 ± 3.9	84.0 ± 12.5	76.9 ± 6.8
	Sugar	80.0 ± 22.4	79.3 ± 3.3	83.4 ± 11.3	80.0 ± 7.1
	Base	81.1 ± 22.8	81.4 ± 3.6	83.4 ± 11.1	81.1 ± 7.6
Scale 4	Phosphate	77.9 ± 23.8	73.7 ± 4.1	84.5 ± 12.3	77.0 ± 6.7
	Sugar	81.0 ± 21.8	78.5 ± 3.5	84.1 ± 11.1	80.1 ± 6.9
	Base	80.4 ± 23.1	81.5 ± 3.6	83.0 ± 11.3	80.9 ± 7.8
Scale 5	Phosphate	79.8 ± 22.7	72.9 ± 3.6	85.6 ± 11.9	77.0 ± 6.0
	Sugar	82.4 ± 21.2	78.6 ± 2.7	85.0 ± 11.0	80.3 ± 6.5
	Base	82.4 ± 21.9	81.0 ± 3.1	84.3 ± 10.9	81.3 ± 7.2

(b) Cryo-EM Coulomb potential map test set

Model	**Output**	Atom Inc. / %	Precision / %	Recall / %	F1 Score / %
Scale 1	Phosphate	27.9 ± 28.3	65.4 ± 12.1	59.8 ± 11.3	60.1 ± 10.0
	Sugar	34.3 ± 30.5	74.1 ± 12.9	60.9 ± 11.6	62.8 ± 11.1
	Base	50.1 ± 31.3	76.1 ± 10.4	68.6 ± 14.1	69.0 ± 11.5
Scale 2	Phosphate	33.3 ± 31.7	67.1 ± 11.0	62.0 ± 13.1	61.8 ± 10.8
	Sugar	43.6 ± 33.4	75.5 ± 11.5	63.9 ± 12.4	65.8 ± 11.9
	Base	49.8 ± 33.2	78.0 ± 11.2	69.1 ± 14.5	69.6 ± 12.4
Scale 3	Phosphate	33.5 ± 31.0	67.4 ± 11.4	62.1 ± 13.0	62.2 ± 10.7
	Sugar	41.6 ± 33.3	77.0 ± 11.6	63.7 ± 12.6	65.5 ± 12.1
	Base	52.3 ± 33.1	78.2 ± 10.7	69.6 ± 14.5	70.2 ± 12.5
Scale 4	Phosphate	31.0 ± 30.8	66.9 ± 12.0	61.2 ± 12.7	61.2 ± 10.6
	Sugar	43.9 ± 32.9	74.1 ± 11.6	64.9 ± 12.9	66.3 ± 11.8
	Base	50.9 ± 32.8	77.3 ± 10.4	69.4 ± 14.5	70.0 ± 12.2
Scale 5	Phosphate	34.7 ± 32.5	66.6 ± 11.6	63.1 ± 14.0	62.5 ± 11.0
	Sugar	44.4 ± 33.1	76.4 ± 10.8	65.1 ± 13.2	66.6 ± 12.0
	Base	50.6 ± 33.1	78.6 ± 11.2	69.2 ± 14.7	69.9 ± 12.5

The molecular replacement test set results indicate that all scaled models are statistically significantly different from the baseline model with $P \approx 0$ for all comparisons. The cryo-EM test set results are significantly better for the phosphate and sugar outputs, whereas

the base outputs are only slightly significantly different for the Scale 3 and Scale 5 models. This is partially interesting because it reveals that base groups in cryo-EM Coulomb potential maps can be identified similarly with any number of parameters tested, however since the overall performance of all models on cryo-EM Coulomb potential maps is poor, some other systematic issue is likely preventing better performance.

This experiment has highlighted the promising performance of a model with 32 initial convolutional filters, whilst also demonstrating how additional parameters can become redundant in larger models. Since this experiment only coarsely sampled possible architectures with integral scale factors, the model with 32 initial convolutional filters may also include some redundant parameters. Non-integral scale factors could be applied to the model, but it is perhaps more helpful to determine which parts of the U-Net architecture are most essential for performance. If redundant and non-performant areas of the model architecture could be identified and removed, this could lead to a model with fewer parameters that retains good performance. Reducing the size of a deep learning model is an important area of study, as smaller models generally enable inference on more limited computing hardware, thereby increasing method accessibility.

Table 3.3: Statistics calculated by comparing the baseline multiclass segmentation model against models with varying numbers of convolutional filters across an X-ray diffraction test set phased with molecular replacement and a cryo-EM Coulomb potential map test set. Pairwise significance estimations were completed with a Wilcoxon signed-rank test. Significance levels are represented by stars where $P \geq 0.05$ = n.s., $0.01 \leq P < 0.05 = *$, $0.001 \leq P < 0.01 = **$, $P < 0.001 = ***$. Delta values are the difference between the average performance of both models for a given output type, expressed in percentage points (pp).

(a) X-ray diffraction molecular replacement test set

Model	**Output**	**Atom Inclusion**			**F1 Score**		
		P-value	Sig.	Delta / pp	P-value	Sig.	Delta / pp
Scale 2	Phosphate	0.00	***	6.1	0.00	***	1.7
	Sugar	0.00	***	7.4	0.00	***	2.2
	Base	0.00	***	7.8	0.00	***	2.5
Scale 3	Phosphate	0.00	***	4.9	0.00	***	1.8
	Sugar	0.00	***	7.5	0.00	***	2.7
	Base	0.00	***	7.9	0.00	***	2.8
Scale 4	Phosphate	0.01	*	5.6	0.00	**	2.0
	Sugar	0.00	***	8.5	0.00	***	2.8
	Base	0.01	**	7.0	0.00	**	2.6
Scale 5	Phosphate	0.00	***	7.5	0.00	***	1.9
	Sugar	0.00	***	9.9	0.00	***	3.1
	Base	0.00	***	9.2	0.00	***	3.0

(b) Cryo-EM Coulomb potential map test set

Model	Output	**Atom Inclusion**			**F1 Score**		
		P-value	Sig.	Delta / pp	P-value	Sig.	Delta / pp
Scale 2	Phosphate	0.00	***	5.4	0.00	***	1.7
	Sugar	0.00	***	9.2	0.00	***	3.0
	Base	0.43	n.s.	-0.3	0.03	*	0.6
Scale 3	Phosphate	0.00	***	5.6	0.00	***	2.1
	Sugar	0.00	***	7.3	0.00	***	2.8
	Base	0.02	*	2.2	0.00	**	1.2
Scale 4	Phosphate	0.62	n.s.	3.1	0.64	n.s.	1.1
	Sugar	0.16	n.s.	9.6	0.19	n.s.	3.5
	Base	0.95	n.s.	0.1	0.69	n.s.	1.0
Scale 5	Phosphate	0.00	***	6.9	0.00	***	2.3
	Sugar	0.00	***	10.1	0.00	**	3.8
	Base	0.01	n.s.	0.5	0.00	***	0.9

3.2.1 Identification of Redundant Parameters

Redundant parameters in any machine learning model take up valuable space and require additional time during training and inference. It is therefore beneficial to mitigate these by reducing the number of parameters in non-critical areas. The U-Net model architecture consists of two key parts: the downsampling portion, which encodes the three-dimensional spatial input into a vector representation, and the upsampling portion, which decodes the vector representation to form an output that is spatially identical to the input. In theory, while both of these portions of the network are important, the upsampling portion attempts to localise features throughout the range of spatial dimensions, which ultimately form the output. It can be hypothesised that the parameters in this portion of the model are more important to the function of the model.

To investigate this, two additional models were trained, the first model consists of a downsampling portion with an initial number of convolutional filters equal to that of the baseline model ($N_0^{\text{filters}} = 16$), and an upsampling portion with the number of convolutional filters equal to that of a model with a scale of 2 ($N_0^{\text{filters}} = 32$). The second model was implemented in a reversed configuration, with a downsampling scale of 2 ($N_0^{\text{filters}} = 32$), and an upsampling scale of 1 ($N_0^{\text{filters}} = 16$). These two models were then trained identically to the baseline multiclass segmentation model, with results shown in Table 3.4 and pairwise statistical analysis for the atom inclusion results are shown in Table 3.5. A similar analysis for the F1 score results is presented in Supplementary Table 8.5.

Comparing the baseline multiclass segmentation model with the newly trained models with increased scale in one portion of the model allows for analysis of the relative importance of both model segments. From these results, a performance benefit is observed when additional parameters are present in the downsampling portion of the model compared to the upsampling portion. Statistical tests completed on the crystallographic test set reveal similar performance between the baseline multiclass segmentation model and the model with increased upsampling parameters in the phosphate and sugar outputs, with a slight increase in performance for the base output. In contrast, the model with increased downsampling parameters exhibits a statistically significant increase in performance compared to the baseline model for all outputs. This performative disparity between the two newly trained models can also be confirmed by comparing the model with increased downsampling parameters against the model with increased upsampling parameters, where there is a statistically significant increase in performance with additional downsampling parameters.

Across the cryo-EM test set, a similar trend can be observed with the model with increased downsampling parameters statistically significantly outperforming the model

with increased upsampling parameters. Further providing evidence that the downsampling portion of the U-Net model is more important for nucleic acid segmentation, with the upsampling portion likely containing some redundancy. If this characteristic were true, it would also be expected that the model with an overall scale of 2 would perform similarly to the model with only increased downsampling parameters, since they only differ in the supposedly redundant upsampling region. Crystallographic examples do not strongly support this theory, with a small increase in performance with the additional upsampling parameters in the model with an overall scale of 2, when compared to the model with only increased downsampling parameters. In contrast to this, across cryo-EM examples, the additional upsampling parameters in the model with an overall scale 2 seem to harm phosphate and base output performance.

While these results are mixed across both structure determination methods, it is clear that the downsampling portion of the model is relatively more important than the upsampling portion of the model, and if parameters must be reduced for computational performance, parameter reduction in the upsampling portion may be sufficient. This result also disproves the original hypothesis that the upsampling region of the model was more important than the downsampling region, providing important insight into how the model operates, however, it is clear that there is more nuance than this coarse experimentation can provide, which may be an interesting area for further study.

Table 3.4: Atom inclusion and F1 score metrics for models with varying numbers of convolutional filters in the downsampling and upsampling portions of the U-Net model, expressed as a scale to the baseline multiclass segmentation model. Metrics were calculated as averages across an X-ray diffraction test set phased with molecular replacement and a cryo-EM Coulomb potential map test set. Uncertainty here represents the standard deviation across the samples.

(a) X-ray diffraction molecular replacement test set

Model		Output	Atom Inclusion / %	F1 Score / %
Downsampling	Upsampling			
Scale 1	**Scale 1**	Phosphate	72.2 ± 27.2	75.0 ± 7.9
		Sugar	72.5 ± 25.4	77.3 ± 8.4
		Base	73.3 ± 28.2	78.3 ± 9.8
Scale 1	**Scale 2**	Phosphate	71.4 ± 26.4	75.6 ± 7.6
		Sugar	72.8 ± 24.9	77.8 ± 8.3
		Base	75.1 ± 26.8	78.9 ± 9.2
Scale 2	**Scale 1**	Phosphate	76.8 ± 24.1	76.7 ± 6.7
		Sugar	77.0 ± 23.1	78.5 ± 7.5
		Base	78.2 ± 24.6	79.9 ± 8.4
Scale 2	**Scale 2**	Phosphate	78.3 ± 23.1	76.7 ± 6.3
		Sugar	79.9 ± 22.3	79.5 ± 7.2
		Base	81.1 ± 23.4	80.8 ± 7.7

(b) Cryo-EM Coulomb potential map test set

Scale		Output	Atom Inclusion / %	F1 Score / %
Downsampling	Upsampling			
Scale 1	**Scale 1**	Phosphate	27.9 ± 28.3	60.1 ± 10.0
		Sugar	34.3 ± 30.5	62.8 ± 11.1
		Base	50.1 ± 31.3	69.0 ± 11.5
Scale 1	**Scale 2**	Phosphate	32.4 ± 30.8	61.7 ± 10.7
		Sugar	38.8 ± 31.8	64.5 ± 11.6
		Base	49.9 ± 31.6	69.9 ± 11.9
Scale 2	**Scale 1**	Phosphate	38.9 ± 33.7	62.8 ± 11.1
		Sugar	44.2 ± 32.8	65.9 ± 11.5
		Base	57.6 ± 30.3	72.1 ± 11.2
Scale 2	**Scale 2**	Phosphate	33.3 ± 31.7	61.8 ± 10.8
		Sugar	43.6 ± 33.4	65.8 ± 11.9
		Base	49.8 ± 33.2	69.6 ± 12.4

Table 3.5: Atom inclusion statistics calculated by comparing the specified reference model against the specified comparison model across an X-ray diffraction test set phased with molecular replacement and a cryo-EM Coulomb potential map test set. Pairwise significance estimations were completed with a Wilcoxon signed-rank test. Significance levels are represented by stars where $P \geq 0.05 =$ n.s., $0.01 \leq P < 0.05 = *$, $0.001 \leq P < 0.01 = **$, $P < 0.001 = ***$. Delta values are the difference between the average performance of both models for a given output type, expressed in percentage points (pp).

(a) X-ray diffraction molecular replacement test set

Reference Model		**Comparison Model**		**Output**	**Atom Inclusion**		
Down.	Up.	Down.	Up.		P-value	Sig.	Delta / pp
Scale 1	**Scale 1**	**Scale 1**	**Scale 2**	Phosphate	0.02	*	-0.8
				Sugar	0.96	n.s.	0.3
				Base	0.00	***	1.8
Scale 1	**Scale 1**	**Scale 2**	**Scale 1**	Phosphate	0.00	***	4.6
				Sugar	0.00	***	4.6
				Base	0.00	***	5.0
Scale 1	**Scale 2**	**Scale 2**	**Scale 1**	Phosphate	0.00	***	5.4
				Sugar	0.00	***	4.2
				Base	0.00	***	3.2
Scale 1	**Scale 2**	**Scale 2**	**Scale 2**	Phosphate	0.00	***	7.0
				Sugar	0.00	***	7.1
				Base	0.00	***	6.0
Scale 2	**Scale 1**	**Scale 2**	**Scale 2**	Phosphate	0.00	***	1.5
				Sugar	0.00	***	2.8
				Base	0.00	***	2.8

(b) Cryo-EM Coulomb potential map test set

Reference Model		**Comparison Model**		**Output**	**Atom Inclusion**		
Down.	Up.	Down.	Up.		P-value	Sig.	Delta / pp
Scale 1	**Scale 1**	**Scale 1**	**Scale 2**	Phosphate	0.00	***	4.5
				Sugar	0.01	*	4.4
				Base	0.54	n.s.	-0.2
Scale 1	**Scale 1**	**Scale 2**	**Scale 1**	Phosphate	0.00	***	11.0
				Sugar	0.00	***	9.8
				Base	0.00	***	7.5
Scale 1	**Scale 2**	**Scale 2**	**Scale 1**	Phosphate	0.00	***	6.5
				Sugar	0.00	***	5.4
				Base	0.00	***	7.7
Scale 1	**Scale 2**	**Scale 2**	**Scale 2**	Phosphate	0.26	n.s.	0.9
				Sugar	0.00	***	4.8
				Base	0.44	n.s.	-0.1
Scale 2	**Scale 1**	**Scale 2**	**Scale 2**	Phosphate	0.00	**	-6.0
				Sugar	0.08	n.s.	-0.6
				Base	0.00	***	-7.7

3.3 Effect of Increasing the Spatial Size of the Input

In the context of real-space density maps, the spatial size of the input to a given model is directly proportional to the space that the input occupies in real space. For example, the input to the baseline multiclass model encloses a box with 32 grid points in each spatial dimension, with a grid spacing of 0.70 Å, evaluating to a box with real spatial dimension lengths of 22.4 Å. Given that a DNA helix is approximately 20 Å in diameter,[254] this size input may envelop the density of many nucleic acid residues, however, it may be too small to identify larger range motifs or conserved structural features. To encode a larger input, two clear options emerge, the grid spacing of the input could be increased, or the length of the spatial dimension could increase. Both options may succeed in encoding a larger real-space volume, however, increasing the grid spacing risks masking the nuanced characteristics of nucleic acid density, so the effect of increasing the spatial dimension length of the input to the U-Net model was investigated. It can be hypothesised that with a larger input, the additional spatial context may allow for a better understanding of larger-scale nucleic acid features, yielding better performance.

The 3D U-Net architecture used in the baseline multiclass segmentation model requires the input spatial dimension length, N, to be an integral power of 2 to allow the spatial dimensions to be halved at each layer, described in Equation 3.4. In light of this requirement, two additional models can be created using the same number of initial filters as the baseline multiclass segmentation model ($N_0^{\text{filters}} = 16$), but with an input spatial dimension length of 64 ($k = 6$) and 128 ($k = 7$).

$$N = 2^k, \quad k \in \mathbb{N} \tag{3.4}$$

Increasing the spatial dimensions length from 32 presents two distinct choices for the model architecture. The first choice is to keep the depth of the model identical to the baseline model architecture, forcing the bottleneck layer to form at a higher spatial dimension. The benefit of this approach, commonly referred to as a *shallow U-Net*, may include the ability to encode more spatial context, without drastically increasing the number of model parameters. Alternatively, the depth of the U-Net can be increased to ensure the bottleneck layer is a linear vector, commonly referred to as a *deep U-Net*. This architecture necessitates additional downsampling and corresponding upsampling blocks, increasing the number of trainable parameters in the model. Both approaches were investigated to determine whether additional model performance can be extracted. It can be hypothesised that the additional spatial input will allow for more nucleic acid features to be identified, and the additional parameters in the deep U-Net compared to the shallow U-Net will also produce better performance, but with the added risk of causing the model to overfit to the training data.

3.3.1 Shallow U-Net Architecture

Shallow U-Net architectures have previously been shown to work well with a range of inputs,[255,256] and reportedly benefit from increased numerical stability during training compared to deeper U-Net models, most likely due to the reduced number of trainable parameters. While enhanced numerical stability is a bonus, the potential power of a shallow U-Net for locating nucleic acid electron density lies in the reduction of trainable parameters. If a shallow U-Net model can function acceptably with a bottleneck layer at a higher spatial dimension, the input spatial dimension length could be increased without additional trainable parameters, since only the convolutional kernels are updated during training. To investigate this, two shallow U-Net models with an input spatial dimension length of 64 ($k = 6$) and 128 ($k = 7$) were trained identically to the multiclass baseline segmentation model. A schematic of the architecture for the model with an input spatial dimension length of 128 is shown in Figure 3.4.

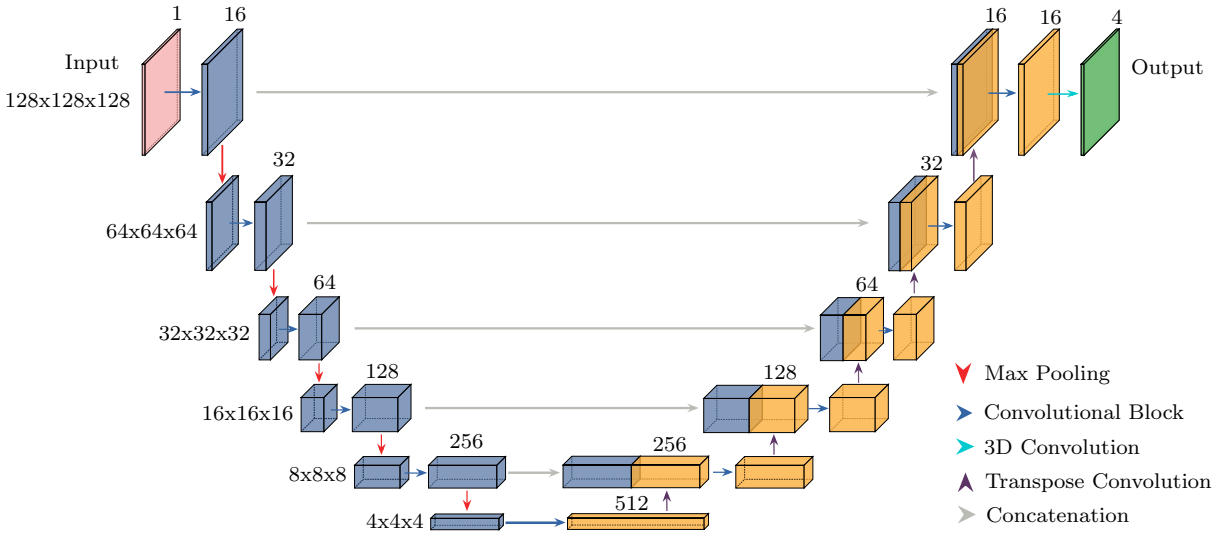


Figure 3.4: Schematic view of the shallow three-dimensional U-Net architecture. The encoder-decoder network first downsamples the data of shape $(128, 128, 128, 1)$ to a vector form of shape $(4, 4, 4, 512)$. The vector is then upsampled back to an output of shape $(128, 128, 128, 4)$, where the four output channels represent the probability of the grid point being no nucleic acid, a phosphate position, a sugar position, or a base position.

The results from the baseline multiclass segmentation model with an input spatial size of 32, and from the two new shallow U-Net models with spatial sizes of 64 and 128 are shown in Table 3.6. Pairwise statistical analysis using a Wilcoxon signed-ranked test is shown in Table 3.7. In both the molecular replacement test set with crystallographic data and the cryo-EM test set, as the spatial size of the input to the model increases, a corresponding increase in performance can be observed.

With crystallographic data, moving from the baseline multiclass segmentation model to the model with a spatial input size of 64 yields an approximate 14 *percentage point*

(pp) increase in average atom inclusion and an approximate 4 pp increase in recall in crystallographic maps. As statistical tests indicate that these differences are significant and the precision metric remains approximately constant, the additional spatial input likely allows more nucleic acid features to be encoded, thereby increasing model performance overall. By extension therefore, further increasing the spatial size to 128 should also allow for even more encoded information and model performance. Statistical tests do show that the atom inclusion performance of the model with a spatial size of 128 is significantly better than the model with a spatial size of 64, but with only a small 2 pp increase on average. Similarly, little change is seen in the F1 score, with only the phosphate and sugar outputs exhibiting any significant differences. It is therefore clear that the additional spatial size in the model with spatial size 128 does not yield a similar performance boost as was observed when moving from the baseline multiclass segmentation model to the model with spatial size 64. Regardless of this trend, the performance of both larger models is strong across the molecular replacement test set.

Importantly, the promising performance of these larger spatial size models is also seen over the cryo-EM Coulomb potential map test set. The baseline multiclass segmentation model performed poorly with cryo-EM data with low atom inclusion and F1 scores. Comparing the baseline model to the model with spatial size 64 evaluates to an approximate 16 pp increase in atom inclusion and an approximate 6 pp increase in F1 score, all of which are statistically significant. This result further confirms the observation that increasing the space encoded by a model improves performance. Increasing the spatial size of the input further in the model with a spatial size of 128, yields additional performance with an approximate 8 pp increase in atom inclusion over the model with a spatial size of 64. This, coupled with small significant increases in F1 score, indicates that this larger model does indeed exhibit markedly better performance than smaller models across both crystallographic and cryo-EM data.

Table 3.6: Atom inclusion, precision, recall and F1 score metrics for the baseline multiclass U-Net model, and the shallow U-Net models with a varying input spatial size ranging from 64 to 128. Metrics were calculated as averages across an X-ray diffraction test set phased with molecular replacement and a cryo-EM Coulomb potential map test set. Uncertainty here represents the standard deviation across the samples.

(a) X-ray diffraction molecular replacement test set

Model	Output	Atom Inc. / %	Precision / %	Recall / %	F1 Score / %
Size 32	Phosphate	72.2 ± 27.2	72.7 ± 5.4	81.2 ± 13.4	75.0 ± 7.9
	Sugar	72.5 ± 25.4	79.0 ± 5.0	78.7 ± 12.0	77.3 ± 8.4
	Base	73.3 ± 28.2	81.2 ± 5.9	79.3 ± 12.9	78.3 ± 9.8
Size 64 Shallow	Phosphate	85.4 ± 19.6	72.1 ± 2.9	88.8 ± 10.6	77.5 ± 5.0
	Sugar	87.4 ± 17.9	77.8 ± 3.1	88.0 ± 9.7	81.4 ± 5.3
	Base	87.0 ± 18.4	80.4 ± 3.0	86.8 ± 9.3	82.6 ± 5.9
Size 128 Shallow	Phosphate	87.5 ± 17.6	72.3 ± 3.0	89.2 ± 9.8	78.0 ± 4.8
	Sugar	89.2 ± 15.9	77.9 ± 2.5	88.3 ± 8.8	81.6 ± 5.0
	Base	89.0 ± 16.5	79.8 ± 3.5	87.7 ± 8.6	82.8 ± 5.4

(b) Cryo-EM Coulomb potential map test set

Model	Output	Atom Inc. / %	Precision / %	Recall / %	F1 Score / %
Size 32	Phosphate	27.9 ± 28.3	65.4 ± 12.1	59.8 ± 11.3	60.1 ± 10.0
	Sugar	34.3 ± 30.5	74.1 ± 12.9	60.9 ± 11.6	62.8 ± 11.1
	Base	50.1 ± 31.3	76.1 ± 10.4	68.6 ± 14.1	69.0 ± 11.5
Size 64 Shallow	Phosphate	43.3 ± 32.8	68.8 ± 8.9	66.3 ± 14.4	65.0 ± 11.0
	Sugar	52.7 ± 31.0	77.3 ± 10.9	67.6 ± 12.8	69.4 ± 11.1
	Base	63.4 ± 30.2	77.6 ± 10.0	75.3 ± 13.5	74.9 ± 10.9
Size 128 Shallow	Phosphate	51.9 ± 32.5	67.9 ± 7.6	69.7 ± 14.9	66.7 ± 10.1
	Sugar	65.0 ± 29.9	75.2 ± 8.4	73.2 ± 12.8	72.7 ± 10.1
	Base	69.2 ± 30.1	76.9 ± 9.0	78.1 ± 13.8	76.0 ± 10.8

Table 3.7: Statistics calculated by comparing the baseline multiclass U-Net model with shallow U-Net models with varying input spatial sizes ranging from 64 to 128 across an X-ray diffraction test set phased with molecular replacement and a cryo-EM Coulomb potential map test set. Pairwise significance estimations were completed with a Wilcoxon signed-rank test. Significance levels are represented by stars where $P \geq 0.05$ = n.s., $0.01 \leq P < 0.05 = *$, $0.001 \leq P < 0.01 = **$, $P < 0.001 = ***$. Delta values are the difference between the average performance of both models for a given output type, expressed in percentage points (pp).

(a) X-ray diffraction molecular replacement test set

Model		Output	**Atom Inclusion**			**F1 Score**		
Ref.	Comp.		P-value	Sig.	Delta / pp	P-value	Sig.	Delta / pp
Size 32	**Size 64 Shallow**	Phosphate	0.00	***	13.2	0.00	***	2.4
		Sugar	0.00	***	14.9	0.00	***	4.1
		Base	0.00	***	13.7	0.00	***	4.3
Size 32	**Size 128 Shallow**	Phosphate	0.00	***	15.2	0.00	***	2.9
		Sugar	0.00	***	16.7	0.00	***	4.3
		Base	0.00	***	15.7	0.00	***	4.5
Size 64 Shallow	**Size 128 Shallow**	Phosphate	0.00	***	2.1	0.00	***	0.5
		Sugar	0.00	***	1.9	0.01	**	0.2
		Base	0.00	***	2.0	0.49	n.s.	0.2

(b) Cryo-EM Coulomb potential map test set

Model		Output	**Atom Inclusion**			**F1 Score**		
Ref.	Comp.		P-value	Sig.	Delta / pp	P-value	Sig.	Delta / pp
Size 32	**Size 64 Shallow**	Phosphate	0.00	***	15.4	0.00	***	4.9
		Sugar	0.00	***	18.4	0.00	***	6.6
		Base	0.00	***	13.3	0.00	***	5.9
Size 32	**Size 128 Shallow**	Phosphate	0.00	***	24.0	0.00	***	6.7
		Sugar	0.00	***	30.7	0.00	***	10.0
		Base	0.00	***	19.1	0.00	***	7.0
Size 64 Shallow	**Size 128 Shallow**	Phosphate	0.00	***	8.6	0.00	***	1.7
		Sugar	0.00	***	12.3	0.00	***	3.4
		Base	0.00	***	5.8	0.00	***	1.1

3.3.2 Deep U-Net Architecture

The original implementation of the U-Net encoded a given input into a linear vector representation known as the bottleneck, a design which has been used in many applications.[108,109,231] As the spatial size of the input increases, the depth of the U-Net must also increase, with more downsampling and corresponding upsampling layers added to ensure that the bottleneck remains a vector representation. These additional downsampling and upsampling blocks add extra parameters to the model, potentially allowing for a better understanding of the features of the input, however with an increased risk of overfitting the training data. To investigate whether the deep U-Net architecture enables efficient

nucleic acid feature identification, two new deep U-Net models were trained with input spatial sizes of 64 and 128. The results of these two models, as well as pairwise Wilcoxon signed-rank statistical tests, are shown in Tables 3.8 and 3.9.

Across the X-ray diffraction test set, the results suggest that increasing the spatial dimensions of the deep U-Net model from 32 to 64 improves the ability of the model to locate all nucleic acid features, with an approximate 13 pp increase in atom inclusion. Statistically significant performance increases were also observed for the sugar and base outputs in F1 score. This result again supports the idea that encoding more real-space information than the baseline multiclass segmentation model leads to a better understanding of the nucleic acid features. Further increasing the spatial size to 128 yields a slight increase in atom inclusion score performance for sugar output, while the phosphate and base outputs remain statistically similar. Minor increases in performance are also seen in F1 score for the phosphate and base outputs, and a minor decrease in performance for the sugar. These results suggest that the sugar prediction has become more imprecise with the additional spatial size, albeit the difference is somewhat negligible.

With cryo-EM data, increasing the spatial dimensions through a deep U-Net model exhibits a comparable trend. Substantial improvements in atom inclusion and F1 score performance are observed when comparing the baseline multiclass model with an input size of 32 to the deep U-Net model with an input spatial size of 64. However, performance gains are not discernible when the input spatial size is further increased to 128. This largest model performs statistically equivalently to the model with an input spatial size of 64 in F1 score for the phosphate and base outputs, as well as in atom inclusion for the sugar and base outputs. The difference between the two models is in phosphate atom inclusion and sugar F1 score, where the larger model is statistically significantly worse.

Table 3.8: Atom inclusion, precision, recall and F1 score metrics for deep U-Net models with a varying input spatial size ranging from 32 to 128. Metrics were calculated as averages across an X-ray diffraction test set phased with molecular replacement and a cryo-EM Coulomb potential map test set. Uncertainty here represents the standard deviation across the samples.

(a) X-ray diffraction molecular replacement test set

Model	Output	Atom Inc. / %	Precision / %	Recall / %	F1 Score / %
Size 32 Deep	Phosphate	72.2 ± 27.2	72.7 ± 5.4	81.2 ± 13.4	75.0 ± 7.9
	Sugar	72.5 ± 25.4	79.0 ± 5.0	78.7 ± 12.0	77.3 ± 8.4
	Base	73.3 ± 28.2	81.2 ± 5.9	79.3 ± 12.9	78.3 ± 9.8
Size 64 Deep	Phosphate	86.1 ± 18.2	70.2 ± 3.7	88.9 ± 10.2	75.9 ± 4.8
	Sugar	85.8 ± 18.3	78.2 ± 3.2	85.4 ± 9.7	80.7 ± 6.0
	Base	86.4 ± 18.7	79.2 ± 3.8	86.4 ± 9.8	81.6 ± 6.5
Size 128 Deep	Phosphate	85.4 ± 18.6	72.2 ± 3.1	88.4 ± 10.2	77.7 ± 5.3
	Sugar	87.0 ± 18.4	77.0 ± 2.8	87.1 ± 9.8	80.5 ± 5.5
	Base	86.3 ± 19.3	80.1 ± 3.5	86.2 ± 9.4	82.3 ± 6.1

(b) Cryo-EM Coulomb potential map test set

Model	Output	Atom Inc. / %	Precision / %	Recall / %	F1 Score / %
Size 32 Deep	Phosphate	27.9 ± 28.3	65.4 ± 12.1	59.8 ± 11.3	60.1 ± 10.0
	Sugar	34.3 ± 30.5	74.1 ± 12.9	60.9 ± 11.6	62.8 ± 11.1
	Base	50.1 ± 31.3	76.1 ± 10.4	68.6 ± 14.1	69.0 ± 11.5
Size 64 Deep	Phosphate	46.2 ± 32.7	65.0 ± 7.4	68.2 ± 15.0	64.8 ± 9.8
	Sugar	51.9 ± 31.7	74.6 ± 9.5	68.3 ± 13.5	69.0 ± 11.3
	Base	62.5 ± 30.4	76.3 ± 9.2	75.4 ± 14.2	73.8 ± 10.7
Size 128 Deep	Phosphate	43.0 ± 33.5	69.2 ± 9.4	65.5 ± 14.0	64.7 ± 11.0
	Sugar	50.3 ± 32.8	75.9 ± 10.5	66.6 ± 12.7	68.0 ± 11.5
	Base	60.2 ± 31.5	77.6 ± 10.8	73.2 ± 13.5	73.5 ± 11.6

Table 3.9: Statistics calculated by comparing deep U-Net models with varying input spatial sizes ranging from 32 to 128 across an X-ray diffraction test set phased with molecular replacement and a cryo-EM Coulomb potential map test set. Pairwise significance estimations were completed with a Wilcoxon signed-rank test. Significance levels are represented by stars where $P \geq 0.05 =$ n.s., $0.01 \leq P < 0.05 = *$, $0.001 \leq P < 0.01 = **$, $P < 0.001 = ***$. Delta values are the difference between the average performance of both models for a given output.

(a) X-ray diffraction molecular replacement test set

Model		Output	**Atom Inclusion**			**F1 Score**		
Ref.	Comp.		P-value	Sig.	Delta / pp	P-value	Sig.	Delta / pp
Size 32 Deep	**Size 64 Deep**	Phosphate	0.00	***	13.8	0.61	n.s.	0.8
		Sugar	0.00	***	13.3	0.00	***	3.4
		Base	0.00	***	13.1	0.00	***	3.3
Size 32 Deep	**Size 128 Deep**	Phosphate	0.00	***	13.2	0.00	***	2.6
		Sugar	0.00	***	14.6	0.00	***	3.2
		Base	0.00	***	13.0	0.00	***	4.0
Size 64 Deep	**Size 128 Deep**	Phosphate	0.52	n.s.	-0.7	0.00	***	1.8
		Sugar	0.00	***	1.3	0.00	***	-0.2
		Base	0.74	n.s.	-0.2	0.00	***	0.6

(b) Cryo-EM Coulomb potential map test set

Model		Output	**Atom Inclusion**			**F1 Score**		
Ref.	Comp.		P-value	Sig.	Delta / pp	P-value	Sig.	Delta / pp
Size 32 Deep	**Size 64 Deep**	Phosphate	0.00	***	18.3	0.00	***	4.7
		Sugar	0.00	***	17.5	0.00	***	6.2
		Base	0.00	***	12.4	0.00	***	4.8
Size 32 Deep	**Size 128 Deep**	Phosphate	0.00	***	15.1	0.00	***	4.5
		Sugar	0.00	***	16.0	0.00	***	5.3
		Base	0.00	***	10.1	0.00	***	4.5
Size 64 Deep	**Size 128 Deep**	Phosphate	0.01	*	-3.2	0.48	n.s.	-0.2
		Sugar	0.35	n.s.	-1.6	0.03	*	-1.0
		Base	0.14	n.s.	-2.3	0.55	n.s.	-0.3

3.3.3 Comparison of Shallow and Deep U-Net Architectures

Comparing the performance of the shallow and deep U-Net architectures allows an assessment of the relative importance of the linear representation in the bottleneck layer. Theoretically, the additional trainable parameters may help the deeper models perform better. Still, it is likely there is a limit at which further additional parameters become redundant, as demonstrated in Section 3.2.

To compare the impact of model parameters on performance, it is useful to compare the shallow and deep U-Net models at the same spatial size. The shallow U-Net with a

spatial size of 64 contains 25,948,868 parameters, whereas the deep U-Net at the same spatial size contains approximately four times as many at 103,820,484 parameters. While a linear increase in performance with additional parameters is not expected, if these additional parameters were non-redundant, the deep U-Net model may realise some increase in performance. A Wilcoxon signed-ranked statistical test was performed on the atom inclusion and F1 scores across the crystallographic and cryo-EM test set to determine if any differences in performance were statistically significant, shown in Table 3.10. Across the crystallographic test set, on average, the deep U-Net performs statistically significantly worse than the shallow U-Net in F1 score by approximately 1 pp. The average atom inclusion results are more mixed, with the base output performing statistically similarly between the models, the phosphate output being slightly significantly better in the deep U-Net model whereas the sugar output is statistically significantly worse in the deep U-Net compared to the shallow U-Net. Results for the cryo-EM test set are similar, with worse F1 score performance in the sugar and base outputs for the deep U-Net model, but with no significant difference in the phosphate output. The atom inclusion result for the base output again shows no significant difference between the shallow and deep U-Net models, whereas the sugar output is slightly significantly worse and the phosphate output is statistically significantly better. Since these results indicate no significant increase in performance, the additional parameters in the deep U-Net model compared to the shallow U-Net model are likely not fully utilised at this spatial scale. The extra parameters are more likely to lead the deep U-Net to overfit the training data, resulting in worse performance in almost all cases.

Performance differences between the shallow and deep U-Net models are also seen at a spatial size of 128. While the shallow U-Net model with an input spatial size of 128 contains an identical number of parameters as the other shallow U-Net models, the deep U-Net model with the same spatial size contains 64 times the number of parameters at 1,660,989,828. This enormous difference in the number of parameters should, in theory, allow the larger model to understand most of the complex features of the nucleic acid and perform well in metrics. However, this model does not perform as well as might be expected, most likely due to significant overfitting. With both X-ray diffraction and cryo-EM data, the deep U-Net model performs statistically significantly worse than the shallow U-Net model across all metrics for all three nucleic acid types. Overcoming this overfitting problem may be possible with additional training data, a smaller learning rate or other regularisation techniques, but the large number of parameters makes such a model impractical to use.

Table 3.10: Statistics calculated by comparing the shallow and deep U-Net models at the same spatial size across an X-ray diffraction test set phased with molecular replacement and a cryo-EM Coulomb potential map test set. Pairwise significance estimations were completed with a Wilcoxon signed-rank test. Significance levels are represented by stars where $P \geq 0.05$ = n.s., $0.01 \leq P < 0.05 = *$, $0.001 \leq P < 0.01 = **$, $P < 0.001 = ***$. Delta values are the difference between the average performance of both models for a given output, expressed in percentage points (pp).

(a) X-ray diffraction molecular replacement test set

Model		Output	**Atom Inclusion**			**F1 Score**		
Ref.	Comp.		P-value	Sig.	Delta / pp	P-value	Sig.	Delta / pp
Size 64 Shallow	**Size 64 Deep**	Phosphate	0.02	*	0.7	0.00	***	-1.6
		Sugar	0.00	***	-1.6	0.00	***	-0.7
		Base	0.13	n.s.	-0.6	0.00	***	-1.0
Size 128 Shallow	**Size 128 Deep**	Phosphate	0.00	***	-2.1	0.02	*	-0.3
		Sugar	0.00	***	-2.2	0.00	***	-1.1
		Base	0.00	***	-2.7	0.00	***	-0.6

(b) Cryo-EM Coulomb potential map test set

Model		Output	**Atom Inclusion**			**F1 Score**		
Ref.	Comp.		P-value	Sig.	Delta / pp	P-value	Sig.	Delta / pp
Size 64 Shallow	**Size 64 Deep**	Phosphate	0.00	**	2.8	0.38	n.s.	-0.2
		Sugar	0.02	*	-0.8	0.00	**	-0.4
		Base	0.39	n.s.	-0.9	0.00	***	-1.1
Size 128 Shallow	**Size 128 Deep**	Phosphate	0.00	***	-8.9	0.00	***	-2.1
		Sugar	0.00	***	-14.7	0.00	***	-4.7
		Base	0.00	***	-9.1	0.00	***	-2.5

3.3.4 Conclusions

Increasing the spatial dimensions of the input to any U-Net model has been shown to increase the performance of the model at identifying nucleic acid features in density. Significant performance improvements have been observed when the spatial size of the input is increased from 32 to 64, with smaller but significant improvements when the spatial size is increased further to 128. Comparing the shallow and deep U-Net architectures underscores the likely redundancy of the additional parameters introduced by the deeper convolutional layers in the deep U-Net model since the performance between the two architectures is comparable. The most performant model identified in this analysis is the shallow U-Net with an input spatial size of 128. Employing such a large spatial input, without additional parameters beyond the baseline multiclass segmentation model, results in strong performance with minimal extra computational expense. To improve these results further, a combination of the additional spatial size and the additional convolutional filters described in Section 3.2 may yield an optimal model.

3.4 Combination of Optimal Spatial Dimensions and Number of Features

A comprehensive search over varying input spatial sizes and the number of convolutional filters has been conducted to identify the most performant and efficient model for estimating nucleic acid density in both crystallographic and cryo-EM data. A shallow U-Net model with a spatial size of 128 outperformed any other model while maintaining an identical number of parameters as the baseline model, furthermore, increasing the number of parameters to a scale of 2 significantly improved model performance with only a modest increase in the number of parameters. The consideration of the number of parameters in a model is crucial, as it determines the computational resources required to utilise a given model. Reducing the number of parameters can make a model more accessible, which is essential for establishing a ubiquitous software methodology. Removing redundant parameters that increase computational requirements is beneficial, and investigations into parameter importance highlighted how reducing the number of convolutional filters in the upsampling portion of the U-Net architecture produced only small performance differences. Combining these findings should, in theory, produce a model with strong performance while maintaining a sensible number of model parameters. If this model is to be successful in comparison to the binary segmentation models designed in Chapter 2.3, this optimised model must exhibit similar atom inclusion and recall performance with a significant increase in precision and F1 score. This would be indicative of a sensitive and specific model which can distinguish nucleic acid features from other areas of density without many false positives.

This combined model was created using a shallow U-Net model architecture with an input spatial dimension of 128, with 32 initial convolutional filters in the downsampling layer, and 16 final convolutional filters in the upsampling layer, shown in Figure 3.5. This model was trained identically to the baseline multiclass segmentation model, and the results are shown in comparison to the binary segmentation model and the baseline multiclass segmentation model in Table 3.11. Statistical tests for all metrics are shown in Supplementary Section 8.1.

The motivation for optimising the three binary segmentation models was to increase precision and reduce the redundancy introduced by having three separate models performing similar tasks. The baseline multiclass segmentation model successfully performed significantly more precisely than the binary segmentation models, however, this came at the cost of poorer recall and atom inclusion performance. The results of further optimising the baseline multiclass segmentation model to form an optimised multiclass segmentation model suggest that this loss in performance has been recovered, without compromising

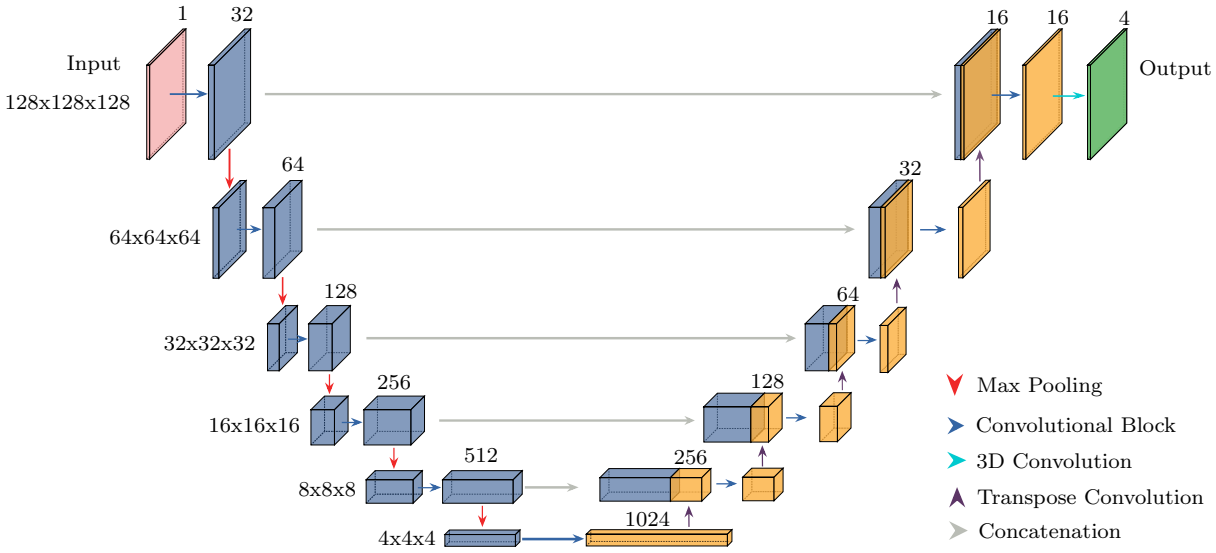


Figure 3.5: Schematic view of the optimised three-dimensional U-Net architecture. The encoder-decoder network first downsamples the data of shape $(128, 128, 128, 1)$ to a vector form of shape $(4, 4, 4, 1024)$. The vector is then upsampled back to an output of shape $(128, 128, 128, 4)$, where the four output channels represent the probability of the grid point being no nucleic acid, a phosphate position, a sugar position, or a base position.

precision or adding many additional parameters. Comparing the atom inclusion performance between the binary segmentation models and the optimised multiclass segmentation model, the results from the crystallographic dataset show no statistically significant differences in any nucleic acid type output. With the cryo-EM dataset, a large statistically significant increase in atom inclusion can be seen for the phosphate and sugar outputs, with the base output remaining statistically similar. These results therefore suggest that the optimised multiclass segmentation model is on par with, or better than, the binary segmentation models in terms of atom inclusion performance.

Observing the differences in precision between the binary segmentation models and the optimised multiclass segmentation model highlights the effectiveness of transitioning from three binary models to a single multiclass model. Across the crystallographic dataset, the optimised model is significantly more precise, with an approximately 13 pp improvement. Furthermore, with the cryo-EM dataset, an even more pronounced precision improvement is observed, with an approximate 17 pp difference. These performance improvements indicate that the optimised model is less likely to produce false positive results, allowing greater importance to be placed on highlighted predictions in downstream applications. A precise model means little without good recall performance, and a trade-off between the two metrics is often observed. The optimised multiclass segmentation model performs much better in precision, yet only a small decrease in recall is observed compared to the binary segmentation models across the crystallographic dataset. This indicates that the optimised model can locate most nucleic acid density without overconfidence, resulting

Table 3.11: Atom inclusion, precision, recall and F1 score metrics for the three binary segmentation models, the baseline multiclass segmentation model and the optimised multiclass segmentation model. Metrics were calculated as averages across an X-ray diffraction test set phased with molecular replacement and a cryo-EM Coulomb potential map test set. Uncertainty here represents the standard deviation across the samples.

(a) X-ray diffraction molecular replacement test set

Model	Output	Atom Inc. / %	Precision / %	Recall / %	F1 Score / %
Binary Segmentation	Phosphate	86.6 ± 17.8	61.2 ± 3.7	90.7 ± 10.1	67.0 ± 5.3
	Sugar	92.4 ± 12.9	64.3 ± 3.6	92.4 ± 8.0	70.7 ± 4.7
	Base	92.6 ± 13.2	67.9 ± 4.0	91.7 ± 7.4	74.4 ± 5.0
Baseline multiclass segmentation	Phosphate	72.2 ± 27.2	72.7 ± 5.4	81.2 ± 13.4	75.0 ± 7.9
	Sugar	72.5 ± 25.4	79.0 ± 5.0	78.7 ± 12.0	77.3 ± 8.4
	Base	73.3 ± 28.2	81.2 ± 5.9	79.3 ± 12.9	78.3 ± 9.8
Optimised multiclass segmentation	Phosphate	87.0 ± 19.0	73.8 ± 2.9	89.1 ± 10.1	79.0 ± 5.3
	Sugar	89.9 ± 17.6	77.5 ± 3.4	89.5 ± 9.2	81.9 ± 5.5
	Base	89.2 ± 17.8	80.4 ± 3.2	88.1 ± 8.8	83.3 ± 5.7

(b) Cryo-EM Coulomb potential map test set

Model	Output	Atom Inc. / %	Precision / %	Recall / %	F1 Score / %
Binary Segmentation	Phosphate	28.5 ± 24.7	55.3 ± 6.2	60.4 ± 9.9	56.1 ± 6.7
	Sugar	49.1 ± 31.0	56.8 ± 7.1	67.8 ± 13.5	58.5 ± 8.1
	Base	67.3 ± 28.7	59.4 ± 7.3	78.4 ± 14.1	63.2 ± 9.4
Baseline multiclass segmentation	Phosphate	27.9 ± 28.3	65.4 ± 12.1	59.8 ± 11.3	60.1 ± 10.0
	Sugar	34.3 ± 30.5	74.1 ± 12.9	60.9 ± 11.6	62.8 ± 11.1
	Base	50.1 ± 31.3	76.1 ± 10.4	68.6 ± 14.1	69.0 ± 11.5
Optimised multiclass segmentation	Phosphate	52.0 ± 30.2	69.0 ± 7.7	69.7 ± 13.6	67.7 ± 9.5
	Sugar	68.2 ± 29.3	75.4 ± 8.5	75.5 ± 12.2	74.4 ± 10.0
	Base	72.2 ± 28.7	76.7 ± 9.0	79.9 ± 12.9	77.4 ± 10.4

in false positives. The recall performance across the cryo-EM dataset is expected to increase with the optimised model because the original binary segmentation models were not trained on cryo-EM data. Indeed, the recall performance is statistically better on average for the optimised model across the phosphate and sugar outputs, with statistically similar performance for the base output.

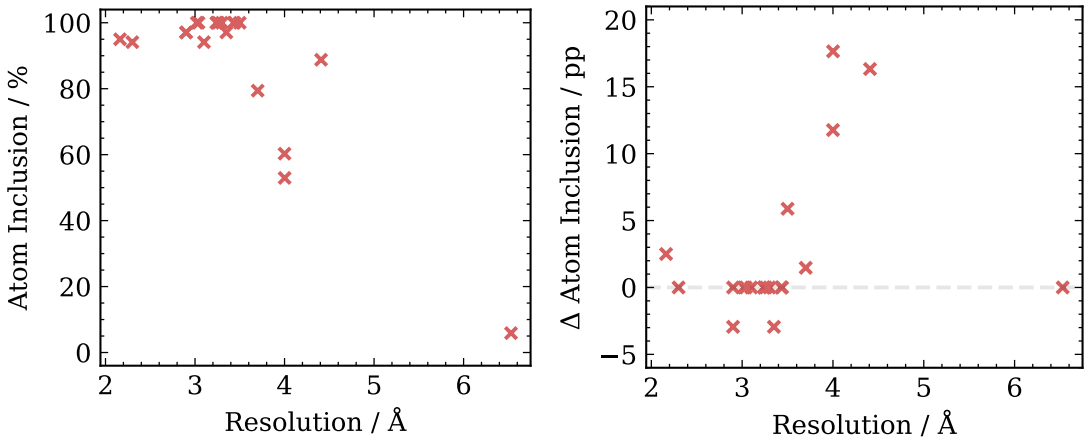
Overall, the optimised multiclass segmentation model performs better than any previous model for identifying nucleic acids in both crystallographic and cryo-EM density. The ability of this model to locate nucleic acid features in post-molecular replacement electron density maps and lower resolution cryo-EM Coulomb potential maps highlights the great potential of this model for automated model building of nucleic acids. Since

this model can work with both crystallographic and cryo-EM data, it is likely that it also has a wider range of tolerable resolutions.

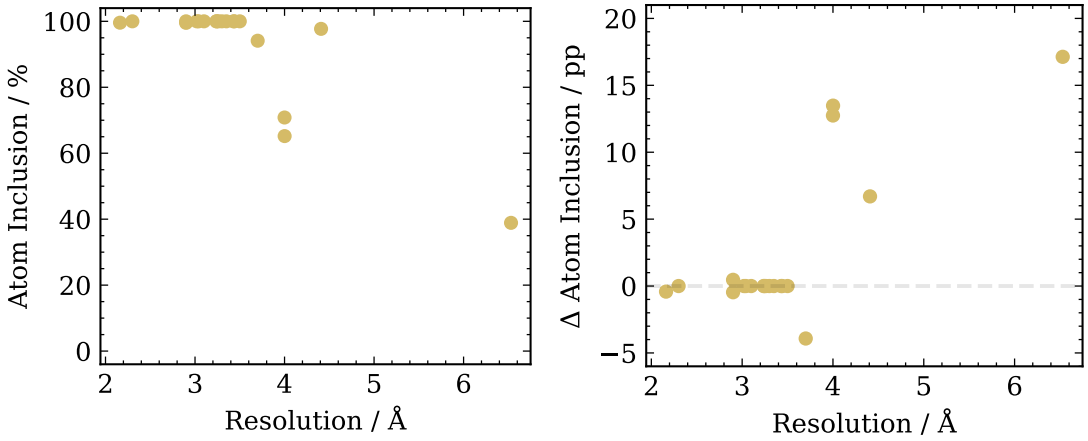
3.4.1 Resolution Dependence

The binary segmentation models outlined in Chapter 2.3 exhibited good performance with crystallographic data at experimental resolutions better than 4 Å. If the optimised multiclass segmentation model was to be useful, it should also exhibit a similar trend and work well with a range of resolution inputs. To investigate this, the same test set of 20 DNA-bound DNA topoisomerase proteins solved with X-ray diffraction were predicted using the optimised model, all of which were excluded from the training set. The atom inclusion results from the optimised multiclass segmentation model, as well as a comparison to the binary segmentation models, are shown in Figure 3.6. This test focused exclusively on crystallographic data since resolution is clearly defined as the limit of reciprocal-space reflections, whereas in cryo-EM, resolution is heavily dependent on the data processing techniques and is more locally variable.[257]

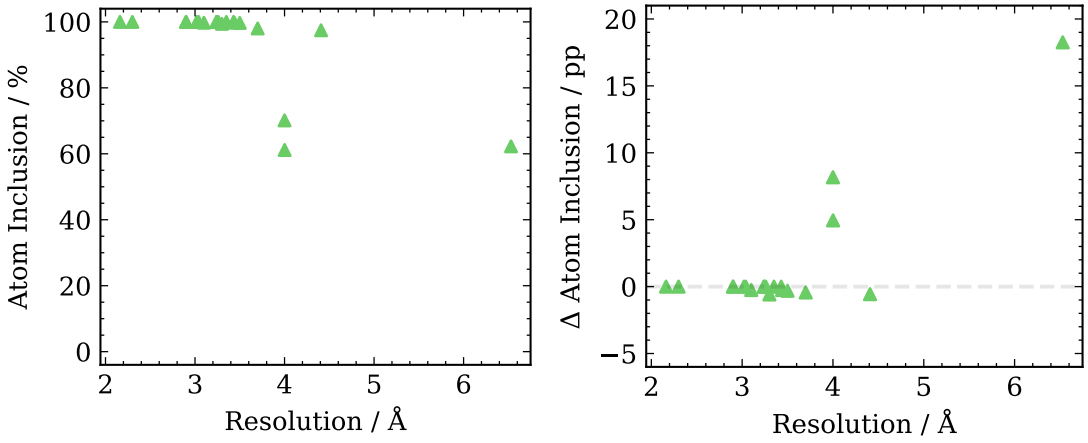
The optimised model performs similarly well with structures solved to resolutions between 2.00 and 3.50 Å, with many structures approaching near-complete atom inclusion. For lower resolution structures, the optimised model performs better in almost all cases, suggesting that either the addition of lower resolution cryo-EM density to the training set aided the low resolution understanding of the model, or that the optimised model architecture helps in some way. The actual cause is likely a combination of the additional data and larger spatial encoding of the optimised model, which may allow predictions to identify specific density features that are often obscured in the absence of high-resolution data.



(a) Phosphate model



(b) Sugar model



(c) Base model

Figure 3.6: Atom inclusion scores of 20 optimised multiclass segmentation model predictions of DNA-bound DNA topoisomerase structures deposited in the Protein Data Bank with resolutions from 2.11 to 6.35 Å. Input maps to each model were calculated using only the protein portion of the protein-nucleic acid complex to emulate molecular replacement. These results suggest that the optimised model has similar performance at high resolution, and better performance at low resolution when compared to the baseline binary segmentation models discussed in Chapter 2.

3.4.2 Conclusions

In conclusion, the convolutional neural networks designed to identify nucleic acid features in density have been optimised for enhanced precision and efficiency compared to the binary segmentation models outlined in Chapter 2.3. The systematic analyses, shown in Figure 3.7, demonstrated that increasing the spatial dimensions of the input to the convolutional neural network improved model performance, with additional parameters in the downsampling component of the network also aiding further performance gains. This trained model can be used in downstream applications to identify nucleic acid features from both crystallographic and cryo-EM data at a range of resolutions, while remaining computationally accessible through careful consideration of the model size and number of parameters.

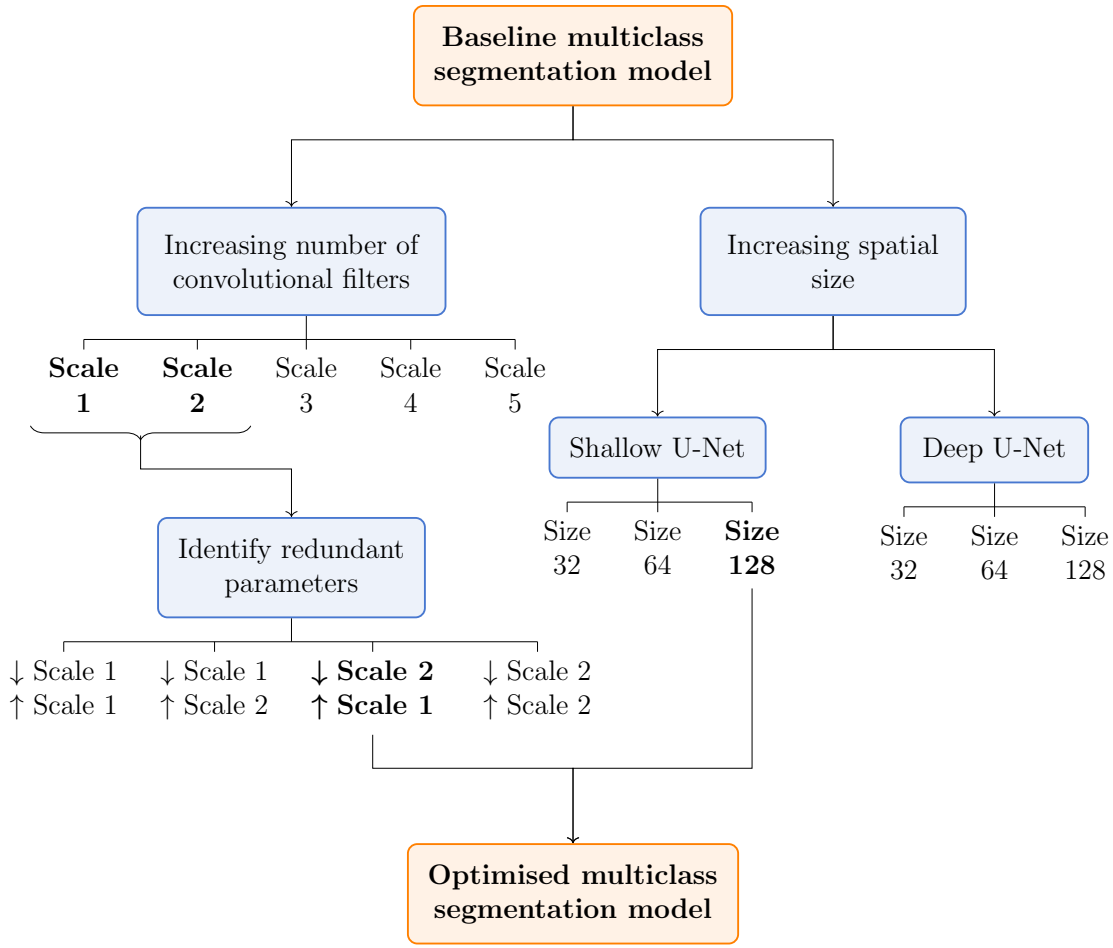


Figure 3.7: Flowchart outlining systematic model analysis and optimisation to yield an optimised multiclass segmentation model from the baseline segmentation model. Initially, both the number of convolutional filters and the spatial size were independently explored. After identifying that the model with a scale of 2 had the optimal number of convolutional filters, redundant parameters in the upsampling portion were found and removed. Investigations into the spatial size of both shallow and deep U-Net models yielded a shallow U-Net with a spatial size of 128 as the most optimal model. The optimised scale and size were combined to form the optimised multiclass segmentation model.

Chapter 4

Automated Model Building of Nucleic Acids Using Deep Learning Predictions

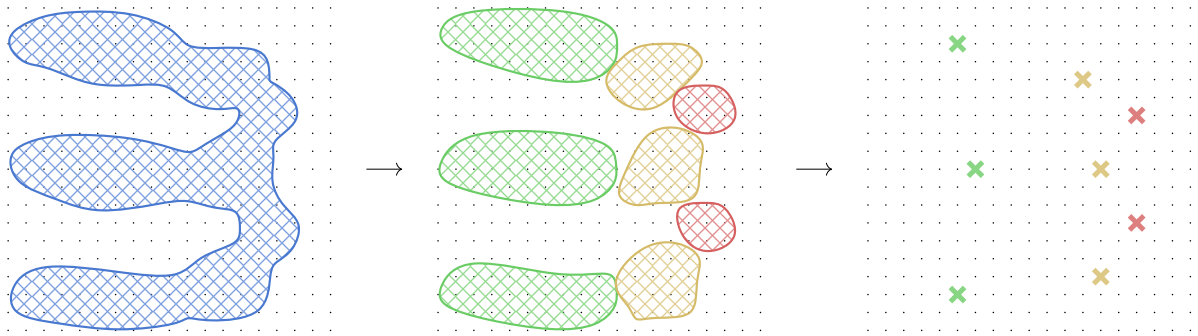
The machine learning predictions designed in Chapter 2 and optimised in Chapter 3 provide a robust methodological foundation for the identification of nucleic acid features from crystallographic and cryo-EM data. In search of the overall objective of automated nucleic acid model building, after potential nucleic acid features have been located, an atomic model which is supported by experimental data must be created.

One approach to building an atomic model from the machine learning predictions is to locate a single point for each predicted volume, which is a similar method to the fingerprint-finding function within the existing nucleic acid model-building program *Nautilus*. Once a single point corresponding to a nucleic acid constituent group has been identified, these points can form the basis of a search strategy for the surrounding areas of experimental density. Following this, fragments can be aligned with these identified positions and joined together to form nucleic acid chains. Existing tidying and refinement algorithms can then process these chains to produce an atomic model of a nucleic acid that is well supported by the experimental data.

The main aim of this method development is to enable the production of atomic models that are more complete than is possible with current-generation software methods, while remaining fast, efficient, and straightforward to use. A time-efficient software method ensures that the structure solution process is as streamlined as possible, aiding the understanding of any system to which it is applied.

4.1 Locating Nucleic Acid Centroids From Predicted Density

The predicted maps generated by the deep learning model helpfully highlight the potential locations of nucleic acid features, yet are mostly too large to pinpoint a single atomic position. This is unsurprising since the deep learning models were not trained for such a task, however this can be accomplished through more classical model-building algorithms. Figure 4.1 shows an overall schematic of this process.



Input density Predicted density Predicted positions

Figure 4.1: Schematic of nucleic acid centroid location from input density. From an input experimental density map, the optimised deep learning model outlined in Chapter 3 can be used to predict areas of nucleic acid features. The phosphate prediction is shown in red, the sugar prediction in yellow, and the base prediction in green. To create an atomic model, these predicted regions can be converted to a single point for each predicted group.

Before algorithmic development, it is important to consider which nucleic acid features predicted by deep learning models would be most useful for automated model building. In a nucleic acid, the base groups are most commonly stacked in some way, and so the predictions for this group will likely outline the general topology of the nucleic acid. While this is useful for identifying where a nucleic acid is generally, it may be difficult to extract specific positional information considering the large volume of the base in comparison to sugars and phosphates. Therefore, a more viable solution may be to utilise the predictions of both the sugar and phosphate groups since they are smaller in volume and contain critical positional information about the nucleic acid backbone. If a singular point for each phosphate and sugar is found, in an ideal situation, the backbone can be easily traced, providing enough information to build a nucleic acid atomic model crudely.

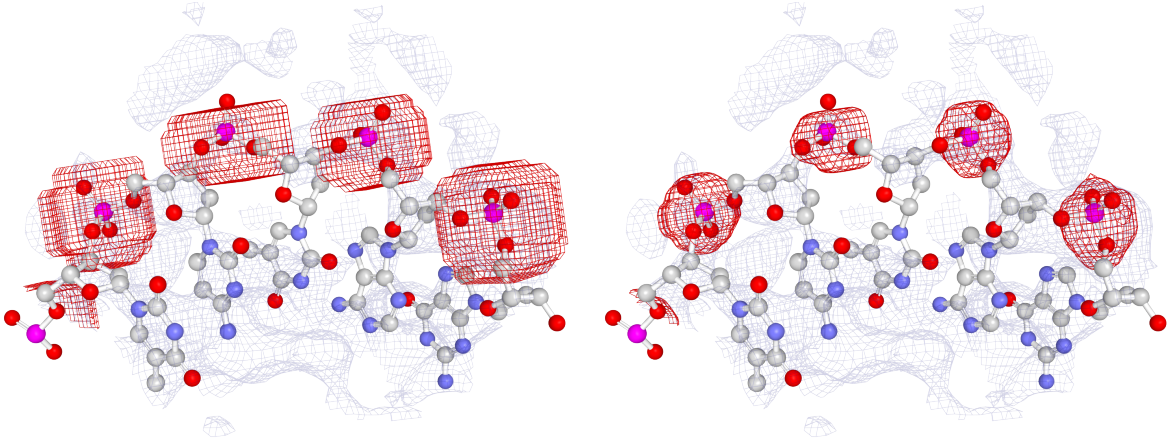
4.1.1 Threshold Selection

Locating a single point algorithmically must begin with an assessment of the predicted map generated from the deep learning model for a given nucleic acid type. While maps are commonly represented visually through specific highlighted areas of interest, the map

itself contains a regular grid of points, each with a value. Visualisation programs often select a specific threshold value to display, and a similar process must be followed when working with a predicted map.

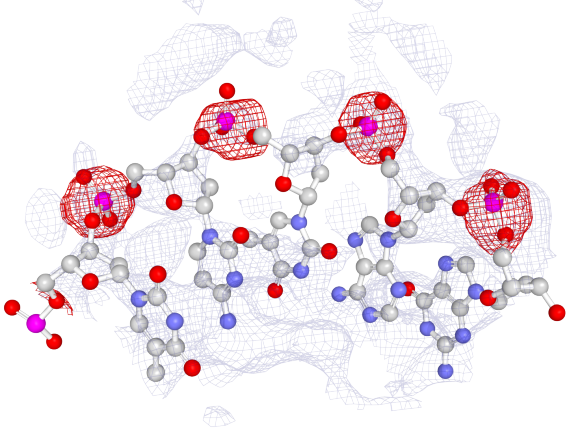
The multiclass deep learning model outputs integral values for each point, which are then split into each map type based on the index of the prediction. Following post-prediction processing and interpolation, the output predicted map for each nucleic acid type contains values in the range of 0 to 1. Intermediate values arise when multiple predictions of a single grid point disagree, for example, over eight predictions for a grid point, seven positive predictions and one negative prediction yield a value of 0.875 for that grid point. This range of values creates a necessity for a threshold for determining when a grid point should be considered a positive prediction. Figure 4.2 illustrates this through the visualisation of a predicted phosphate map. With a threshold of 0, both slight positive predictions and strong positive predictions are equally weighted creating a large predicted area that encompasses the deposited phosphate atoms well. With a slight increase in the threshold value to 0.10, the predicted volumes become much smaller while remaining well fit to the deposited phosphate atoms, suggesting that a low, but non-0 threshold produces a more interpretable result for further processing. Increasing the threshold further again reduces the predicted volumes, but in this case, would not make a substantial difference to the outcome.

A specific optimised threshold can be calculated analytically by measuring precision and recall across a range of thresholds from 0 to 1 and selecting the threshold that maximises the average F1 score. This method is good if false positives and false negatives are equally undesirable, however for automated model building, this is not the case. If the predicted maps miss a nucleic acid feature, i.e., a false negative, at least one residue may be missing in the final built structure. Whereas, if the predictions suggest a false nucleic acid feature, i.e. a false positive, this can be disputed by comparison with the experimental data. For this reason, a threshold value of 0.10 was chosen to maximise the recall performance of the predictions without significantly reducing the precision, as would be the case with a threshold of 0. For the same reasons, this threshold was also used when processing predicted sugar maps.

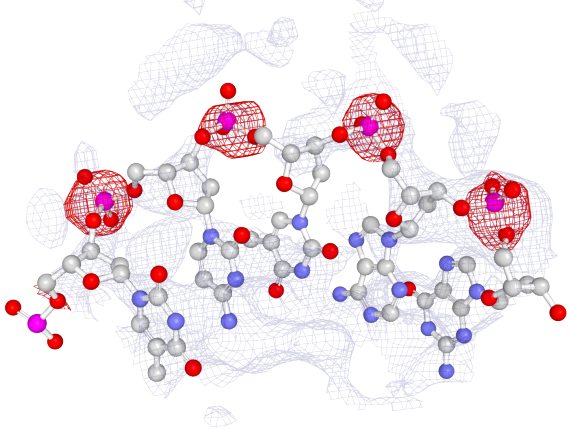


(a) Threshold - 0.00

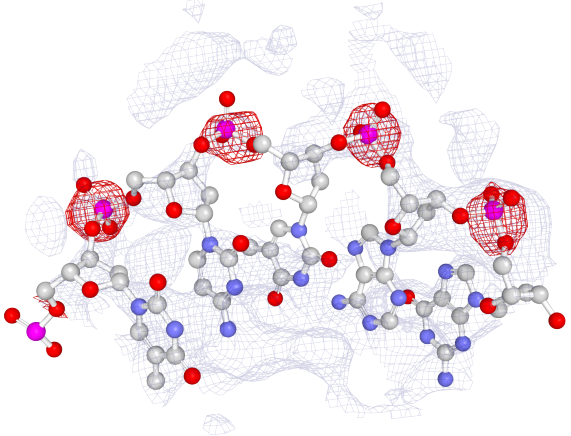
(b) Threshold - 0.10



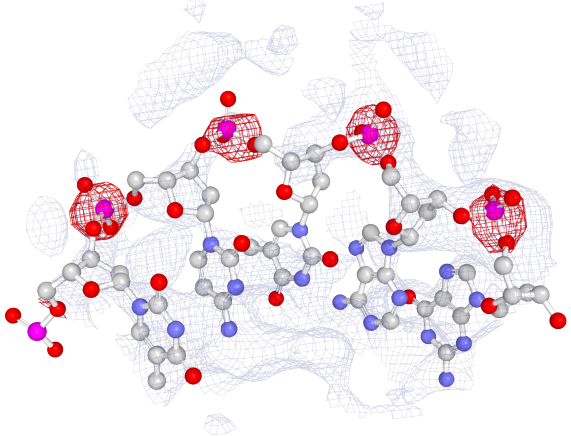
(c) Threshold - 0.20



(d) Threshold - 0.30



(e) Threshold - 0.40



(f) Threshold - 0.50

Figure 4.2: Predicted phosphate maps are shown in red with varying contour threshold values to illustrate how the specific threshold value can influence the processing of predictions. Lower thresholds will include weaker predicted points and inflate predicted volumes, whereas higher thresholds only include stronger predictions and result in smaller predicted volumes. The choice of threshold determines the sensitivity of a given method. The predicted phosphate map was generated from the protein-only molecular replacement solution map of a transcription factor solved with X-ray crystallography to 2.35 Å resolution (PDB code: 5D5W[258]). The atomic model shown is the deposited model obtained from the Protein Data Bank.

4.1.2 Centroid Calculation

To calculate a centroid position for each sugar and phosphate predicted volume, a set of grid points is located which have associated predicted map values greater than the decided threshold of 0.10, defined in Equation 4.1.

$$\mathcal{P} = \{(x, y, z) \mid \rho_{\text{pred}}(x, y, z) \geq \tau\} \tag{4.1}$$

where:

$\mathcal{P}$ is a set of points, each defined by a Cartesian coordinate
ρ_{pred} is a predicted map
τ is the threshold, set to 0.10

Each point in the set, $\mathcal{P}$, represents a highlighted predicted point which must then be combined with neighbouring points to achieve a single point per predicted volume. This is accomplished through an iterative hill ascent algorithm which compares the current point, p, at a given step, s, with all neighbouring points, $\mathbb{N}_{grid}$. The point is then moved to the same position as the neighbouring point with the highest predicted map value, shown in Equations 4.2 and 4.3. This algorithm steps through grid points until no neighbours with higher predicted map values remain, shown in Equation 4.4.

$$(x_{s+1}, y_{s+1}, z_{s+1}) = \underset{(x',y',z') \in \mathbb{N}_{\text{grid}}(x,y,z)}{\operatorname{argmax}} \rho_{\text{pred}}(x', y', z') \quad \forall (x_s, y_s, z_s) \in \mathcal{P} \tag{4.2}$$

$$\text{where} \quad \mathbb{N}_{\text{grid}}(x, y, z) = \left\{ (x', y', z') \;\middle|\; \begin{array}{c} x' \in \{x-1, x, x+1\}, \\ y' \in \{y-1, y, y+1\}, \\ z' \in \{z-1, z, z+1\}, \\ (x', y', z') \neq (x, y, z) \end{array} \right\} \tag{4.3}$$

$$\text{until} \quad \rho_{\text{pred}}(x_{s+1}, y_{s+1}, z_{s+1}) \leq \rho_{\text{pred}}(x_s, y_s, z_s) \tag{4.4}$$

This process leaves all points, $\mathcal{P}$, in local maxima. To ensure that each point reaches the global maximum of a specific predicted volume, all local maxima within 1.5 Å of each other can be joined together and an average position calculated. It is necessary to perform the hill ascent before distance-based averaging to prevent accidental inclusion of points in a neighbouring predicted volume. After averaging, a new set of points, $\mathcal{P}'$, is created, which should contain a single point in each predicted volume. Having a single position that represents the predicted map volume makes further processing much easier, but it is important to compare predictions against the experimental data to ensure consistency and rigour.

4.1.2.1 Phosphate Centroid Refinement

Once a set of centroid positions corresponding to phosphate features, $\mathcal{P}'$, has been identified, each position can be refined to the local maximum grid point value in the

$2mF_o - DF_c$ map in X-ray crystallography. This additional step is not strictly necessary for the function of the algorithm, but it is likely to yield improved performance for X-ray crystallographic data as a result of the elemental composition of phosphate groups. Phosphorus is comparatively electron-rich when compared to the rest of the atomic composition of proteins and nucleic acids, and is often associated with large grid point values in the surrounding area with X-ray data. These high absolute grid point values make it slightly easier to locate the phosphate groups with greater accuracy, but significant caution must be taken when working with lower-resolution data. It is important to note that unconstrained positional refinement at low resolution or with poorly phased crystallographic data may result in identified phosphate feature positions far from where the true feature is. In consideration of this, simplex minimisation[259] of the Cartesian coordinate can be constrained with an objective function, f, shown in Equation 4.5. It is important to negate the score since the simplex algorithm is designed for minimisation. Additionally, for numerical stability, the theoretically infinite condition is replaced with the maximum 64-bit integer. This approach cannot be reliably applied to cryo-EM data, since the Coulomb potential map represents the distribution of electrostatic potential rather than electron density, which is more likely to yield refined positions far from the actual phosphorus position.

$$f(x,y,z) = \begin{cases} -\rho(x,y,z) & \text{if } \rho_{\text{pred}}(x,y,z) > 0 \\ \infty & \text{otherwise} \end{cases} \tag{4.5}$$

After constrained simplex refinement of the phosphate positions, the set of points, $\mathcal{P}'$, should more accurately represent the underlying nucleic acid feature. These points may then be used to identify larger-scale nucleic acid fragments. This type of centroid refinement is not directly applicable to a sugar prediction, since there is no clear maximal point for a nucleic acid sugar ring that this algorithm could identify.

4.2 Identification of Trinucleotide Fragments

The topology of a nucleic acid chain can be aptly described by the positions of the phosphorus atoms in the sugar-phosphate backbone. These positions encode enough information to understand the general three-dimensional conformation of the nucleic acid. Therefore, when attempting to build a nucleic acid atomic model into density, these positions may form the basis of a crude model. Accurately describing the topology of a nucleic acid fragment without rotational ambiguity requires at least three potential atomic positions, known as a triplet, to be found, illustrated in Figure 4.3.

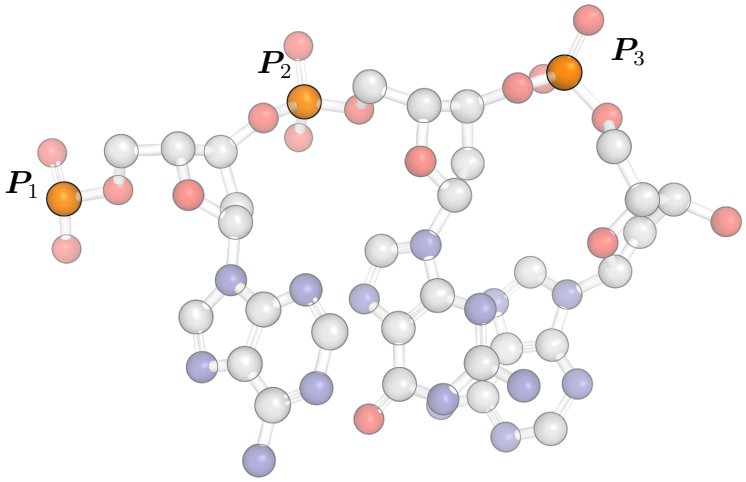


Figure 4.3: Illustration of a trinucleotide fragment of DNA with phosphorus atoms $\boldsymbol{P}_1$ to $\boldsymbol{P}_3$ highlighted. These three atomic positions can crudely describe the topology of a nucleic acid backbone.

After centroid calculation, a set of potential atomic points is available, but they are not organised in any meaningful way, making direct application to model building impractical. Given a set of points $\mathcal{P}'$, the total number of possible combinations of triplets is given by Equation 4.6.[260] For 100 potential points in $\mathcal{P}'$, this evaluates to 161,700 combinations, which, while possible to model, would be largely inefficient. Therefore, a strategy for reducing this potentially large number of configurations must be formulated, which can be achieved by understanding when a set of points is likely to be realistic given known biochemical conditions.

$$\binom{N}{3} = \frac{N!}{3!(N-3)!} \quad \text{where } N = |\mathcal{P}'| \tag{4.6}$$

4.2.1 Analysis of Existing Nucleic Acid Structures

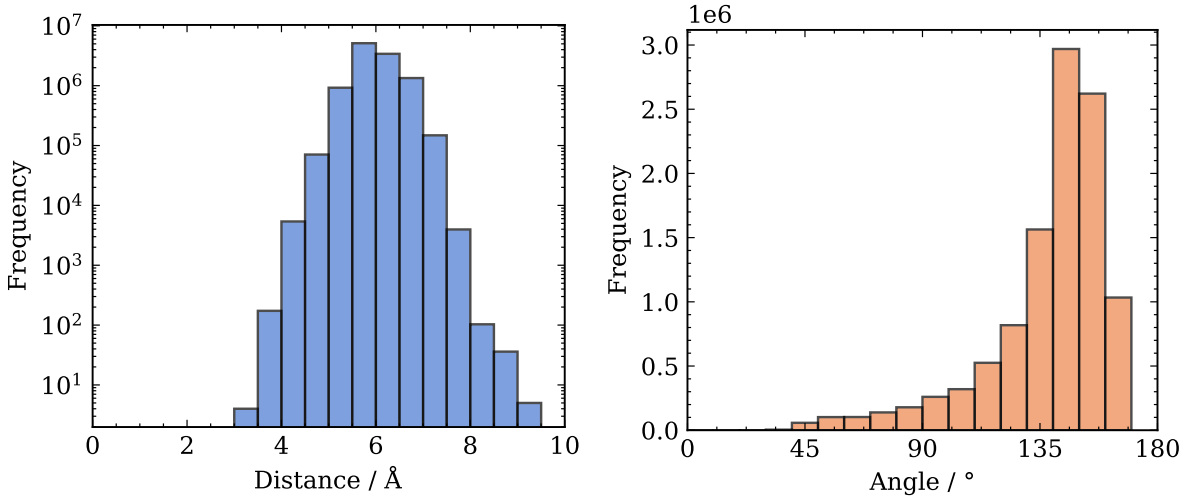
To understand the expected geometry of phosphorus triplets, a search can be conducted across existing nucleic acid structures, where the inter-phosphate atomic distance and

phosphate triplet angles are recorded. To do this, a dataset of nucleic acid-containing structures deposited in the Protein Data Bank was generated. A filter was applied to only include structures solved using X-ray diffraction or cryo-EM, to remove computationally derived models that may bias the dataset. For each nucleic acid chain in the deposited model, the orthogonal distance, d, between phosphorus atom positions in two consecutive nucleotides was calculated, shown in Equation 4.7. Similarly, the angle between three consecutive nucleotides was calculated using the cosine formula, shown in Equation 4.8.

$$d_n = \|\boldsymbol{P}_{n+1} - \boldsymbol{P}_n\| \quad d_{n+1} = \|\boldsymbol{P}_{n+2} - \boldsymbol{P}_{n+1}\| \tag{4.7}$$

$$\theta_n = \arccos\left(\frac{(\boldsymbol{P}_{n+1} - \boldsymbol{P}_n) \cdot (\boldsymbol{P}_{n+2} - \boldsymbol{P}_{n+1})}{d_n d_{n+1}}\right) \tag{4.8}$$

The results highlight clear preferences in inter-phosphate distances and phosphate triplet angles in deposited structures, shown in Figure 4.4. The average inter-phosphate distance evaluates to 6.0 Å with a standard deviation of 0.4 Å, and the average phosphate triplet angle evaluates to 139.5 ° with a standard deviation of 23.5 °. These values align well with what would be expected biochemically, for example, the large angular standard deviation is likely due to the wide range known of nucleic acid conformations. This information can be used to place constraints on the set of points, $\mathcal{P}'$, to reduce the number of likely triplet combinations.



(a) Distance distribution **(b)** Angle distribution

Figure 4.4: Inter-phosphate distance and phosphate triplet angle histograms calculated from nucleic acids deposited in the Protein Data Bank, solved with X-ray diffraction or cryo-EM. The inter-phosphate distance distribution is plotted with a logarithmic scale for clarity. Both geometric values have clear preferences with an average distance value of 6.0 ± 0.4 Å and an average angle value of $139.5 \pm 23.5°$.

4.2.2 Backbone Tracing

Backbone tracing is a longstanding automated model-building algorithm which encompasses many techniques and is utilised in almost every macromolecular automated model-building program.[54,63,64,261,262] The general goal of backbone tracing is to generate a graph-like structure of the macromolecule backbone which can then be used for further processing. Traditional backbone tracing algorithms may use specialised search functions to find candidate backbone positions, whereas this method can obtain them from processed deep learning predictions.

From a set of points, the backbone tracing algorithm must find which nearby points are likely to form a realistic connection. Any two points with inter-point Euclidean distances within a threshold, τ_d, may be considered as potentially linked. The selection of τ_d is crucial and must be derived from empirical observations of real nucleic acid structures. The average inter-phosphate distance derived from the Protein Data Bank was 6.0 Å, and it is important to allow a relatively large tolerance in this distance to accommodate imperfections in the deep learning prediction point-finding algorithm. Consequently, τ_d was set to 6.0 ± 2.0 Å.

Using this threshold, a graph network, G, can be formed from a set of points, $\mathcal{P}'$, and a set of edges, E, where each edge in the graph is composed of two points with Euclidean distances within the range of τ_d, denoted formally in Equation 4.9. Figure 4.5 illustrates this process, where, for instance, points $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are sufficiently close in space to be considered linked.

$$E = \{(\boldsymbol{P}_i, \boldsymbol{P}_j) \mid 4 \leq \|\boldsymbol{P}_i - \boldsymbol{P}_j\| \leq 8, \boldsymbol{P}_i, \boldsymbol{P}_j \in \mathcal{P}', i \neq j\} \tag{4.9}$$

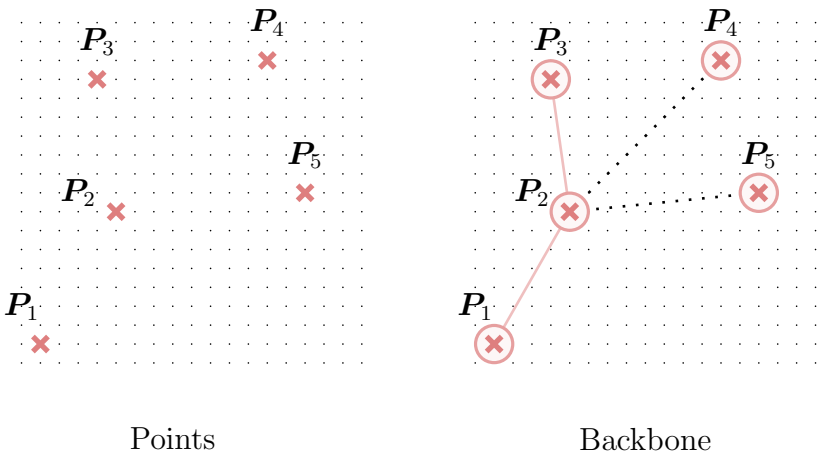


Figure 4.5: Schematic of backbone tracing, an algorithm which takes a set of points and creates a graph network detailing a set of edges and points which describe the potential nucleic acid topology. This approach must have restrictions in place to reduce the chance of unrealistic nucleic acid topologies from being processed further.

Two consecutive edges can be located to form a potential triplet of points, provided that the geometry is consistent with what is expected of a nucleic acid chain. To enforce this expectation, an angular threshold, τ_θ, can be imposed to narrow the range of potential triplet points where the angle, θ, is calculated using Equation 4.8. Given many potential conformations of both DNA and RNA, this threshold should not be excessively restrictive. From observations of the Protein Data Bank, τ_θ was set to $45 \leq \tau_\theta \leq 175$ to account for a wide array of potential conformations and the potential uncertainty in the predicted phosphate positions. This restriction, based on empirical observations, is likely to be biased both toward the data available in the Protein Data Bank and the statistical nature of angular sampling, but is acceptable for the purposes of filtering out non-biochemically relevant potential triplets. Using these two restrictions, τ_d and τ_θ, a triplet, T_n, can be calculated as defined in Equation 4.10.

$$T = \{(\boldsymbol{P}_i, \boldsymbol{P}_j, \boldsymbol{P}_k) \mid (\boldsymbol{P}_i, \boldsymbol{P}_j) \in E, (\boldsymbol{P}_j, \boldsymbol{P}_k) \in E, i \neq k, 45 \leq \theta \leq 175\} \tag{4.10}$$

Using this method, points shown in Figure 4.5 can be grouped into two triplets, $(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3)$ and $(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_5)$. The potential triplet $(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_4)$ must be discounted as the phosphate triplet angle exceeds the maximum allowed by τ_θ. This algorithm works well for locating triplets in many instances, however, it may produce unwanted false positives with certain nucleic acid topologies. Such is the case in for triplet $(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_5)$, which bridges the helical fragment defined by $(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3)$ with another fragment of which points $\boldsymbol{P}_4$ and $\boldsymbol{P}_5$ are part of. This local feature of two separate nucleic acid strands coming into close contact is common in RNA structures and this algorithm will likely assess the two separate helical strands as being linked. To help avoid this problem and ensure that the backbone is correctly traced, another restriction must be placed to ensure that any located triplet is contained within a single helical fragment and does not form an undesirable backbone bridge.

Since this backbone tracing algorithm is most useful when a model is unavailable, the algorithm must understand which points belong to the same helical fragment without external information. To achieve this, a new source of context must be introduced. Fortunately, the deep learning predictions that this algorithm relies on also provide potential sugar ring positions. Given that the positions of a sugar ring and phosphate group are likely to exist nearby to each other, the predicted sugar ring positions can provide crucial context when attempting to trace the backbone of the nucleic acid.

This additional algorithmic step begins with locating sugar positions from predicted sugar volumes, $\mathcal{S}$, as described in Section 4.1. In a nucleic acid chain, a sugar ring is situated proximal to the midpoint of any two adjacent phosphate positions, and this biochemical characteristic can be used to further restrict the edges, thereby preventing backbone errors. For each identified edge, a restriction can be imposed such that the midpoint be-

tween the two points in the edge is within a specified distance of any sugar position, $\boldsymbol{S}_n$. This is denoted formally in Equation 4.11, and is shown schematically in Figure 4.6. Between the predicted phosphate points $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ is a predicted sugar point $\boldsymbol{S}_1$, indicating a high probability of these two phosphate points being topologically joined. Furthermore, between the predicted phosphate points $\boldsymbol{P}_2$ and $\boldsymbol{P}_3$ is another predicted sugar point $\boldsymbol{S}_2$. The triplet $(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3)$ is supported by these two sugar positions, as well as the distance and angular thresholds restrictions, indicating that this is a probable trinucleotide fragment and should be further processed. In contrast, the previously identified triplet $(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_5)$ does not adhere to these new restrictions, as there are no close sugar points to the midpoint of the $(\boldsymbol{P}_2, \boldsymbol{P}_5)$ potential edge. This additional check decreases the probability that the backbone tracing algorithm accidentally produces undesirable results.

$$E' = \{(\boldsymbol{P}_i, \boldsymbol{P}_j) \in E \mid \exists \boldsymbol{S}_n \in \mathcal{S} : \|\mathbf{m}_{ij} - \boldsymbol{S}_n\| \leq 2\} \tag{4.11}$$
$$\text{where} \quad \mathbf{m}_{ij} = \frac{\boldsymbol{P}_i + \boldsymbol{P}_j}{2}$$

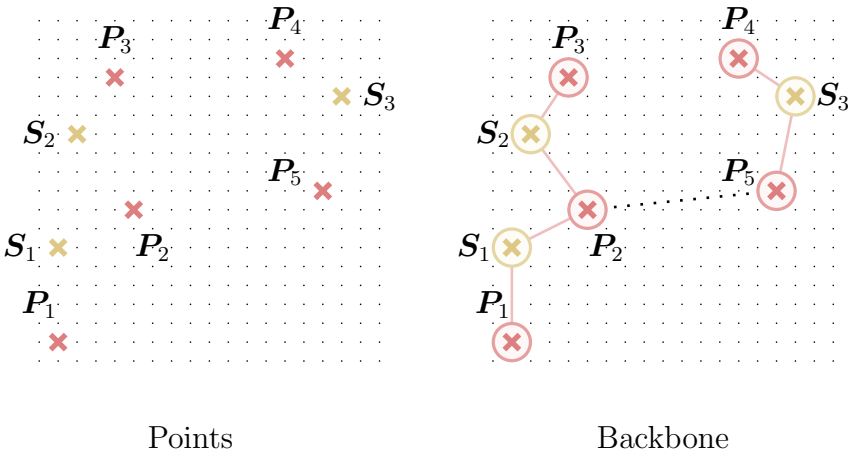


Figure 4.6: Schematic of backbone tracing using predicted sugar points to restrict the identified triplets to more realistic geometries. This algorithm ensures that a predicted sugar point exists somewhere close to the midpoint of two potential phosphate positions, which prevents close nucleic acid backbones from being incorrectly associated.

Once all potential triplets have been identified and verified, larger-scale backbone features can be extracted by aggregating overlapping triplet points. This process is shown in Figure 4.7 where overlapping triplets $(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3)$ and $(\boldsymbol{P}_2, \boldsymbol{P}_3, \boldsymbol{P}_4)$ can be aggregated into a chain consisting of points $(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3, \boldsymbol{P}_4)$. This process continues until a terminal triplet is found, at which point the traced backbone ends. Despite all of the restrictions placed, in certain cases, this algorithm could still produce branch points where three triplets overlap, such that the backbone can continue in multiple ways. It is unlikely that such a nucleic acid topology is realistic and is more likely due to a complex nucleic acid conformation. If branch points remain in the traced backbone, it could lead to modelling

of an improbable nucleic acid chain, therefore, this potential issue must be identified and resolved during the following backbone building steps.

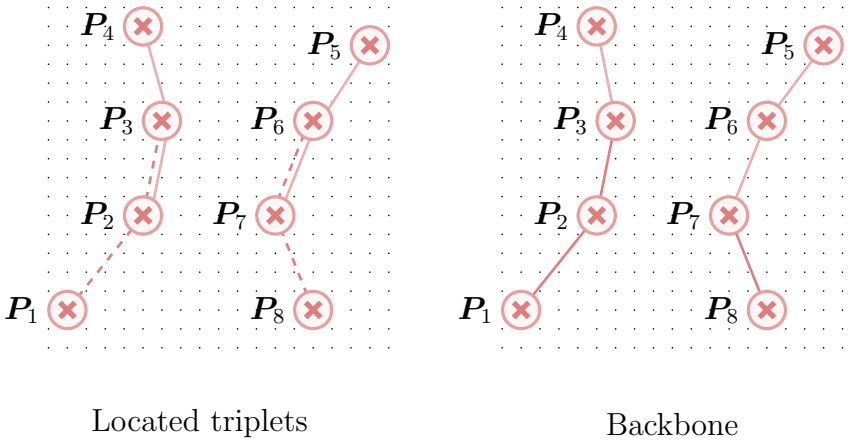


Figure 4.7: Schematic of triplet point aggregation to form a larger-scale backbone. Points $(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3)$ and $(\boldsymbol{P}_2, \boldsymbol{P}_3, \boldsymbol{P}_4)$ can be aggregated into a chain consisting of points $(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3, \boldsymbol{P}_4)$. This continues until a terminal triplet is reached. If any branch points are encountered, the branch which forms the longest chain is favoured.

4.3 Backbone Building

The traced backbone describes the overall topology of the nucleic acid chains, however the goal of an automated model-building software package is to build an atomic model of the structure, so specific residues must be introduced and modelled correctly to satisfy this requirement. Given a known macromolecular backbone, a standard approach is to model the new system using validated fragments from other macromolecular structures.[261] Alternatively, atomic positions could be modelled directly into the density without using known fragments as a guide, but this method is unlikely to produce an unbiased and geometrically sound atomic model without an extremely robust methodology, so was not considered.

Placing a single nucleotide into density at a specific point introduces three degrees of rotational freedom that must be searched explicitly to determine the optimal orientation which satisfies the data. While feasible, modelling nucleotides independently may pose similar challenges to modelling atomic positions independently, potentially introducing biases toward the current density data while leaving neighbouring nucleotides with unfavourable backbone geometries. Attempting to model dinucleotide fragments mitigates some of these issues by reducing the degrees of freedom to one, provided that the dinucleotide spans two fixed backbone positions. These two fixed positions establish an axis of rotation for which a rigid dinucleotide could potentially be modelled in any rotation. Both of these methods are viable, but when searching through large fragment libraries,

searching rotational space for every fragment could become computationally inefficient.

Perhaps a more efficient approach would be to trial trinucleotide fragments, eliminating any rotational degrees of freedom if rigid superposition is applied. This approach would allow more fragments to be fit in a fixed amount of time, compared to using a single or dinucleotide fragment, which both require rotational searches for an optimal fit. The trinucleotide fragments trialled in this algorithm should represent varying conformational states of the sugar and phosphate backbone, and so by attempting to model each fragment into the supporting density, the most optimal fragment can be identified. Increasing the size of the proposed fragment beyond three may allow for a better understanding of the nucleic acid topology, but would require a larger sampling of potential conformational states, which are likely to reduce the computational efficiency of this method, and so was not attempted. To determine the optimality of a trinucleotide fragment, a representative score must be calculated for each potential fragment, with the highest-scoring fragment selected for further processing.

4.3.1 Scoring

Automated model-building algorithms often score any placed fragments by comparing the model with the observed data, either the $2mF_o - DF_c$ density map in X-ray crystallography or the Coulomb potential map in cryo-EM. This method provides a reliable assessment of whether a placed fragment is accurate or if there is insufficient data to support it. Commonly, two main scoring algorithms are used and both have specific advantages and drawbacks for use in automated algorithms. *RSCC*, defined in Equation 1.5, measures the agreement between the current model and the experimental data. A density map is calculated from the model and each grid point is effectively compared to what is seen in the experimental density map, with the score describing the correlation between the two maps. RSCC is a beneficial score algorithmically since it explicitly compares the environment of the residue to the experimental data, allowing for erroneous fragments to be easily identified using specific thresholds. Despite the favour of RSCC as a metric, there are important drawbacks which should be considered. RSCC is sensitive to the quality of the experimental data, the value may reduce when the experimental data is of a lower resolution or has a significant degree of phase uncertainty. A more important drawback for use algorithmically is the requirement of a calculated density map for each fragment. This calculation can become computationally inefficient when dealing with a large fragment library across many potential nucleic acid sites.

To circumvent this speed issue, a less accurate but more computationally efficient alternative is commonly employed. This method involves evaluating the density values at each atomic point in a given residue, which can be summed to provide a score which

reflects how well enclosed a given residue is in the experimental density, shown in Equation 4.12. For a residue with N atoms, this algorithm only requires N map operations, which is a significant reduction in the number of necessary computational operations when compared to calculating RSCC for a standard nucleotide residue.

$$\text{Score} = \sum_{i}^{N} \rho_{obs}(x_i, y_i, z_i) \tag{4.12}$$

Since this atomic position scoring method is more time-efficient, it is a more effective score when attempting to build the backbone of a nucleic acid via fragment placement. The major downside of the atomic scoring method is the additional uncertainty introduced by considering only atomic positions. In theory, this score would produce high scores for any large area of positive density values, which could, for instance, originate from a neighbouring protein residue. Such an imprecise score may lead to incorrect fragments being favoured, depending on the environment of the traced nucleic acid backbone. A more robust method may incorporate knowledge of the difference between experimental density regions originating from nucleic acids and those from other molecules.

The predicted maps that enable the backbone to be precisely traced can also serve as guides for positioning and scoring residues. Specifically, the sugar and base predicted maps can be used to supplement the atomic scoring method to ensure that a fragment has appropriately positioned sugar and base features. The process of using these predicted maps is analogous to the atomic position score, wherein all sugar ring atoms are scored against the predicted sugar map, and all available base atoms are scored against the predicted base map. Both scores are then added to the atomic position score.

4.3.2 Fragment Library Generation

To model any potential nucleic acid backbone using trinucleotide fragments, a range of conformations must be sampled to identify the fragment that best supports the experimental data. These conformations are commonly sampled from a precomputed library of fragments, containing a number of varied nucleic acid conformations. Existing conformational libraries, such as DNATCO,[89] exist for nucleic acid structures, but commonly focus on mononucleotide or dinucleotide conformations. To model trinucleotide fragments, a new fragment library must be generated which encompasses the most common trinucleotide conformations across nucleic acid structures. To do this, all consecutive trinucleotides from all nucleic acid structures in the Protein Data Bank were gathered before being assigned into groups based on the geometry of the sugar-phosphate backbone. Trinucleotide conformational groups were assigned based on the relative positions of the P, O5', C5', C4', C3', and O3' atoms, which must all be within 0.75 Å of another trinucleotide to be considered the same conformation. This threshold was chosen

to ensure that the conformational groups remain sufficiently distinct from each other, whilst allowing small variations in atom positions to be accounted for. To ensure fairness regardless of deposited nucleotide type, all base groups must be removed from each conformation. In total, this process yielded 65,942 trinucleotide conformations from 11,369,359 trinucleotides. This substantial number of conformations indicates a high degree of variability in nucleic acid conformations, however, 60,674 of these conformations have less than 50 occurrences across the Protein Data Bank, suggesting that most conformations are uncommon or may even be improbable. To eliminate these low-frequency conformations, the library was filtered to include only the most frequent 90 % of conformations, yielding 818 trinucleotide fragments. It is likely that this process does not yield a uniform sample of all trinucleotide conformations, nevertheless, when considering that automated model-building programs are almost always followed by real-space or reciprocal-space refinement, this limitation is unlikely to become a bottleneck but may be an avenue for future work.

4.3.3 Fragment Placement

After a set of three candidate phosphate positions has been identified, the three phosphate atoms in a sampled trinucleotide from the precomputed fragment library may be rigidly superposed over the three candidate phosphate positions. Each trinucleotide from the library can then be scored to determine the optimal fragment for a given set of candidate phosphate positions. When scoring a potential trinucleotide fragment, the first two nucleotides and the final phosphate group should be scored using the atomic scoring method described in Section 4.3.1. The sugar group in the final nucleotide is omitted from this calculation to prevent terminal positions from down-weighting the fragment score.

Each nucleic acid chain has a specific associated directionality, which is an important nomenclature for understanding the orientation of each residue in a chain. *In vivo* nucleic acids are assembled in the $5' \rightarrow 3'$ direction,[246] but when attempting automated model building, the directionality of a given chain is unknown. As a result, when attempting to place fragments, both the $3' \rightarrow 5'$ and $5' \rightarrow 3'$ directions must be trialled to determine which best supports the available experimental data.

Before attempting to build entire chains, it is important to remove any branch points that could not be resolved during backbone tracing. Ultimately, all modelled nucleic acids must be well validated by experimental data, so to resolve a branch point, all possible triplet permutations can be trialled and the permutation that best fits the experimental density should remain, while all other permutations should be removed. This method should prevent triplets from bridging nearby nucleic acid chains when there is no

experimental density to support such a conformation.

Following this, fragments can begin to be placed in both the $3' \rightarrow 5'$ and $5' \rightarrow 3'$ directions over each set of triplets across all aggregated chains. Since each pair of points may be part of at most two triplets, if there is an overlap, the nucleotide with the highest score should be kept. This process yields chains of consecutive nucleotides built in both the $3' \rightarrow 5'$ and the $5' \rightarrow 3'$ directions. To determine the most probable direction, the scores for each nucleotide in a chain can be aggregated and compared in both directions, with the highest-scoring direction kept for further processing and the other direction discarded. This method is shown schematically in Figure 4.8.

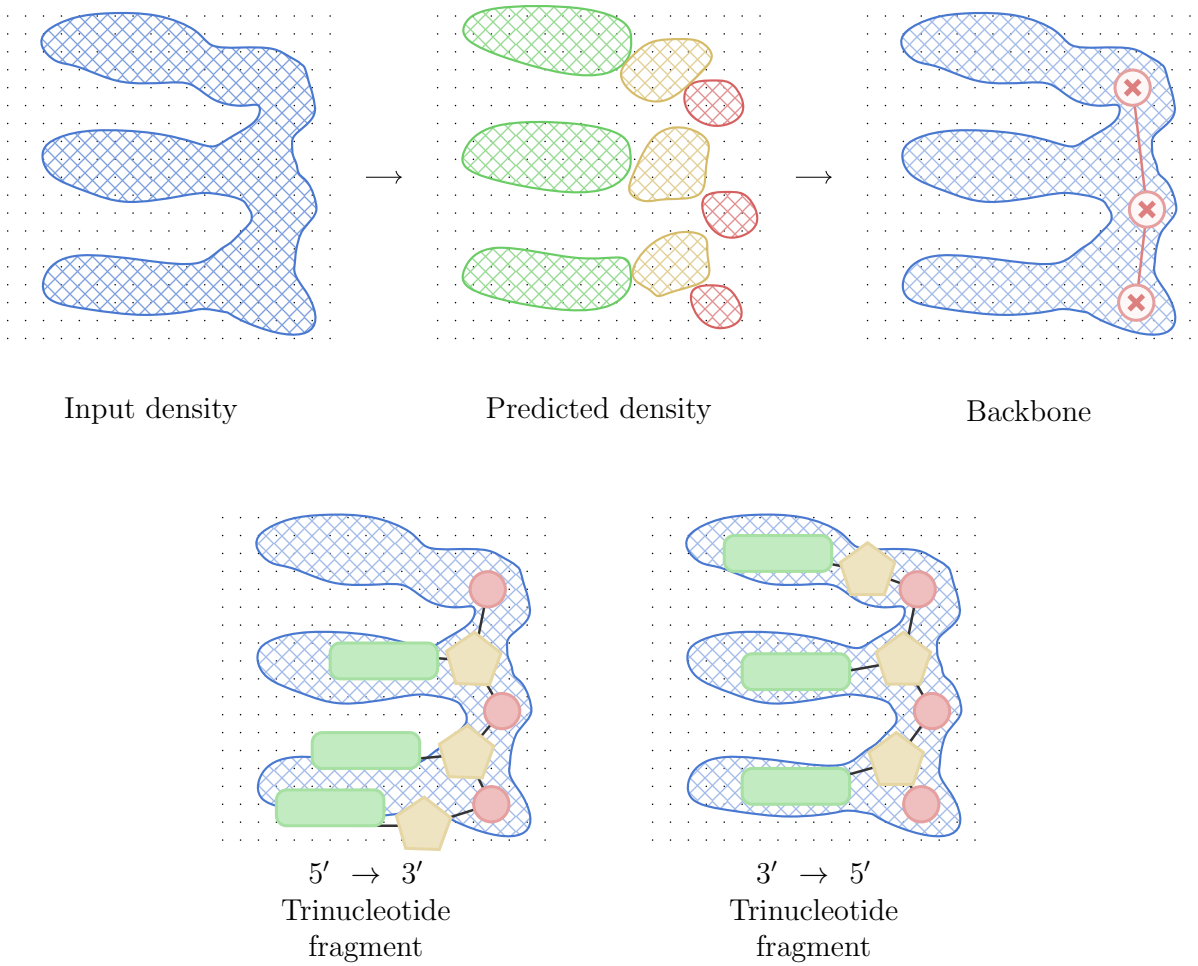


Figure 4.8: Schematic of the fragment placement method for nucleic acid model building. The deep learning model predicts the positions of the phosphate, sugar and base groups from the input experimental density. The phosphate and sugar positions are used to trace the backbone of the nucleic acid. Fragments are rigidly superimposed over the backbone in both the $3' \rightarrow 5'$ and $5' \rightarrow 3'$ directions, and the most probable set of fragments are selected. The $5' \rightarrow 3'$ shows a poor fit to the experimental density, whereas the $3' \rightarrow 5'$ fragment is well fit.

4.4 NucleoFind

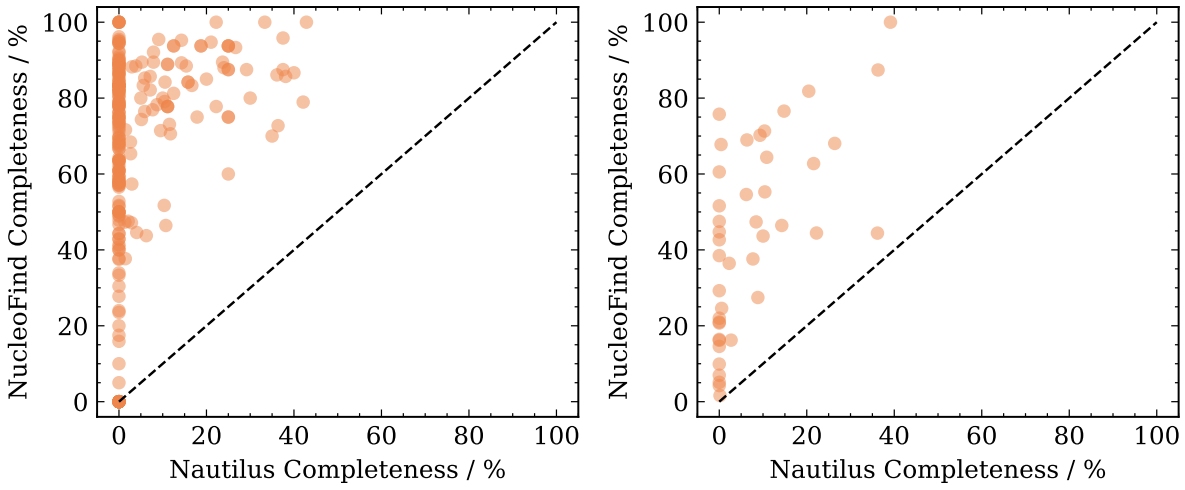
The optimised deep learning models and accompanying optimised software method for automated model building were combined into a new software package named *NucleoFind*. This software package uses predictions from the optimised deep learning model, along with new backbone building algorithms, to locate nucleic acid features in experimental density. The aim of adopting this new method is to supplement or replace the existing methods in the current-generation software package *Nautilus*. After *NucleoFind* locates nucleic acid features, the result is passed to existing growing, joining, pruning and sequencing methods contained within *Nautilus*. These established post-processing methods function well and are important to obtain consistent results across a range of inputs. For *NucleoFind* to be considered successful, the completeness of the automatically built models must improve over the existing method *Nautilus*.

The relative performance of the two software packages may be evaluated by comparing the automatically built models from X-ray crystallographic and cryo-EM experimental data. Each automated model can be scored with a completeness metric,[107] which is defined as the number of modelled nucleotides which have all phosphate and sugar ring atoms within 2 Å of the corresponding atoms in the deposited nucleotide, with respect to the total number of nucleotides in the deposited model. This strict metric was chosen to critically assess the quality of any automatically built model.

To establish the relative power of both *Nautilus* and *NucleoFind*, automated model building was run using both packages across both the X-ray diffraction and cryo-EM test sets outlined in Sections 2.2 and 3.1.3. The completeness of models from both software packages is shown in Figure 4.9, with statistics in Table 4.1. Across the X-ray diffraction test set, the average completeness of models built with *Nautilus* evaluates to 5.2 ± 9.9 %, whereas *NucleoFind* produces models with an average completeness of 67.1 ± 26.7 %. With cryo-EM data, *Nautilus* produces models with an average completeness of 7.8 ± 10.9 %, whereas *NucleoFind* models the nucleic acids more completely with an average completeness of 43.5 ± 25.0 %. This remarkable increase in completeness between *NucleoFind* and *Nautilus* suggests the deep learning model powering *NucleoFind* is indeed able to locate more nucleotides, enabling the modelling algorithms to produce more complete molecular models of nucleic acid-containing structures.

The strong performance of *NucleoFind* with protein-nucleic acid examples from molecular replacement is particularly impressive, since the experimental density generated from a partial molecular replacement solution is often difficult to interpret.[263] Without any refinement, *NucleoFind* can model some or even all of the missing nucleic acid molecules.

However, automated model building rarely concludes the structure solution process. Instead, it is commonly included in an *automated model-building pipeline* that uses cycles of density modification, automated model building and refinement in an attempt to achieve a more complete macromolecular structure. This process is mainly beneficial in crystallographic examples, since additional correctly placed atomic positions only improve the experimental density map through better phase estimates, but may also help in some instances with cryo-EM data.



(a) X-ray diffraction **(b)** Cryo-EM

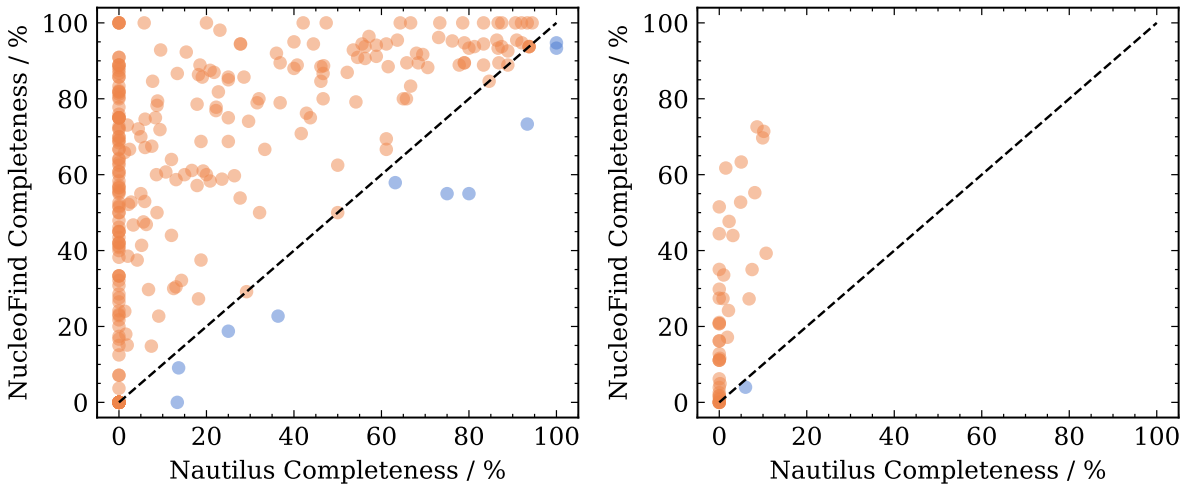
Figure 4.9: Completeness of models generated from automated nucleic acid model-building software packages *Nautilus* and *NucleoFind*. Completeness is defined as the proportion of nucleotide residues with all sugar and phosphate atoms within 2 Å of the deposited model. *NucleoFind* outperforms or is equal to *Nautilus* in every example in a test set of crystallographic molecular replacement examples and a test set of cryo-EM examples.

Table 4.1: Statistics calculated by comparing the completeness of refined models generated by *Nautilus* and *NucleoFind* with default parameters across an X-ray diffraction test set phased with molecular replacement and a cryo-EM Coulomb potential map test set. Pairwise significance estimations were completed with a Wilcoxon signed-rank test. Significance levels are represented by stars where $P \geq 0.05$ = n.s., $0.01 \leq P < 0.05 = *$, $0.001 \leq P < 0.01 = **$, $P < 0.001 = ***$. Delta values are the difference between the average completeness of models created for a given output, expressed in percentage points (pp).

Test Set	**Reference**	**Comparison**	P-value	Significance	Delta / pp
X-ray diffraction	*Nautilus*	*NucleoFind*	0.00	***	62.0
cryo-EM	*Nautilus*	*NucleoFind*	0.00	***	35.7

4.5 Integration Into Model-Building Pipelines

The existing automated nucleic acid model-building program, *Nautilus*, is included in the *ModelCraft* automated model-building pipeline[264] as part of the *CCP4* software suite.[61] It is natural, therefore, to attempt to use *NucleoFind* as a drop-in replacement for *Nautilus*, which should yield similar performance gains to those observed when running both software packages alone. To test this, all 288 molecular replacement examples and 50 cryo-EM structures were run through 5 cycles of both *ModelCraft* with *Nautilus* and *ModelCraft* with *NucleoFind*. By default, *ModelCraft* will run for 25 cycles, however this was reduced to allow this analysis to be conducted on a realistic timescale. The completeness of models from both software pipelines is shown in Figure 4.10. Across the crystallographic examples, *ModelCraft* with *Nautilus* produced models with an average completeness of 24.3 ± 30.1 %, and *ModelCraft* with *NucleoFind* produced models with an average completeness of 62.3 ± 31.0 %. With cryo-EM data, the average completeness of models from *ModelCraft* with *Nautilus* was 2.0 ± 3.3 %, whereas *ModelCraft* with *NucleoFind* produces models with an average completeness of 24.7 ± 22.2 %.



(a) X-ray diffraction **(b)** Cryo-EM

Figure 4.10: Completeness of models generated from automated nucleic acid model-building pipeline *ModelCraft* with nucleic acid model-building software packages *Nautilus* or *NucleoFind*. Orange points represent improvements in completeness for *NucleoFind* over *Nautilus*, blue points represent deterioration in completeness for *NucleoFind* over *Nautilus*. Completeness is defined as the proportion of nucleotide residues with all sugar and phosphate atoms within 2 Å of the deposited model. On average, *NucleoFind* outperforms *Nautilus* in a test set of crystallographic molecular replacement examples and a test set of cryo-EM examples.

Surprisingly, the completeness of models built by *ModelCraft* with *NucleoFind* in both crystallographic and cryo-EM examples falls when compared to *NucleoFind* alone, initially suggesting that the application of refinement and additional *NucleoFind* cycles harms performance. This is a particularly counterintuitive result for crystallographic data, since cycles of refinement and model building should improve the quality of the

phase estimates, potentially allowing for a more complete molecular model to be built. To understand the cause of this unusual performance decline, it is first necessary to examine how the *ModelCraft* pipeline is structured to identify another potential explanation. A schematic of one crystallographic cycle of *ModelCraft* is shown in Figure 4.11, encompassing a range of software methods for creating and improving an atomic model to the experimental density. *ModelCraft* operates linearly and employs automated protein model building with *Buccaneer*, which is eventually followed by automated nucleic acid model building with *Nautilus* or *NucleoFind*. The cryo-EM pipeline is similar, with nucleic acid model building preceded by protein model building.

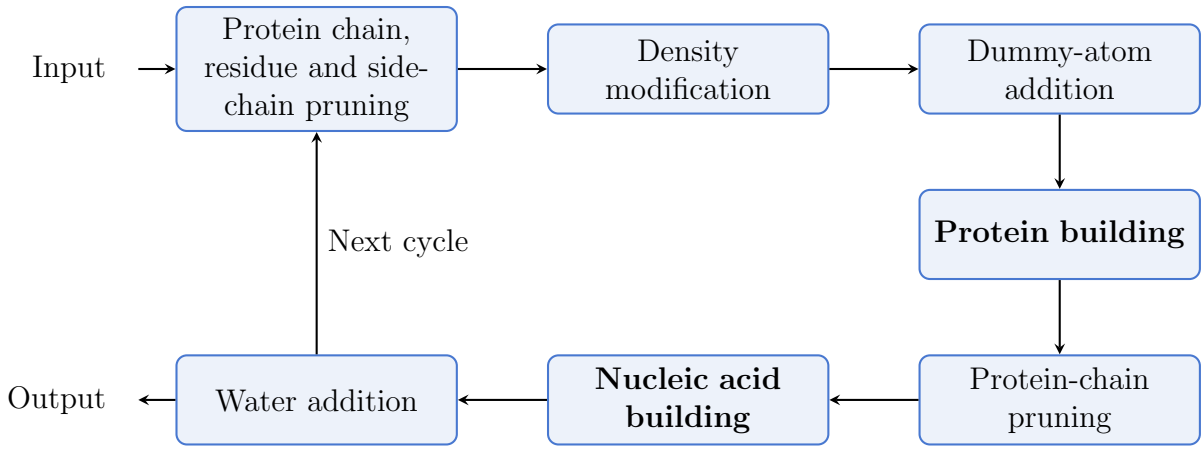
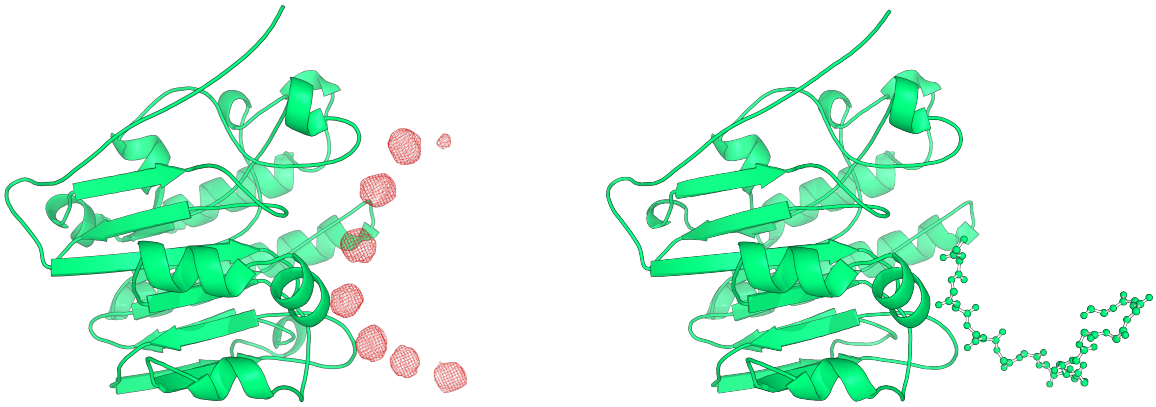


Figure 4.11: General schematic of *ModelCraft* with X-ray diffraction data.[264] *ModelCraft* encompasses a range of software methods for improving and modelling atomic positions, including density modification, automated protein model building, and automated nucleic acid model building. Refinement is omitted from this schematic for clarity but occurs after any modification to the model. This process occurs cyclically until there is no improvement in R_{work}, or until the declared number of cycles is reached.

The schematic reveals no clear indication of why the performance of *ModelCraft* with *NucleoFind* falls with additional cycles, provided that methods perform acceptably. To get a better insight, it is helpful to observe an example where *NucleoFind* predicts a particular region to contain some nucleic acids and identify what is modelled by *ModelCraft*. The protein molecular replacement solution of an exonuclease in complex with DNA (PDB code: 7CD6[265]) is shown in Figure 4.12a with the predicted phosphate density calculated by *NucleoFind* from the protein molecular replacement solution map. This predicted map shows a potential region of nucleic acids in complex with the protein, which is modelled completely by *NucleoFind* alone. Despite this, *ModelCraft* with *NucleoFind* fails to model this area as a nucleic acid, instead modelling a single peptide chain, as shown in Figure 4.12b. The culpability of this can be tied directly to the order in which *ModelCraft* builds macromolecules, first proteins and then nucleic acids. If the automated protein model-building software, *Buccaneer*,[63] mistakenly models a peptide in density better supported by a nucleic acid, there would be no immediate way for *Nautilus* or *NucleoFind* to rectify this. This behaviour is by design as part of the model pruning



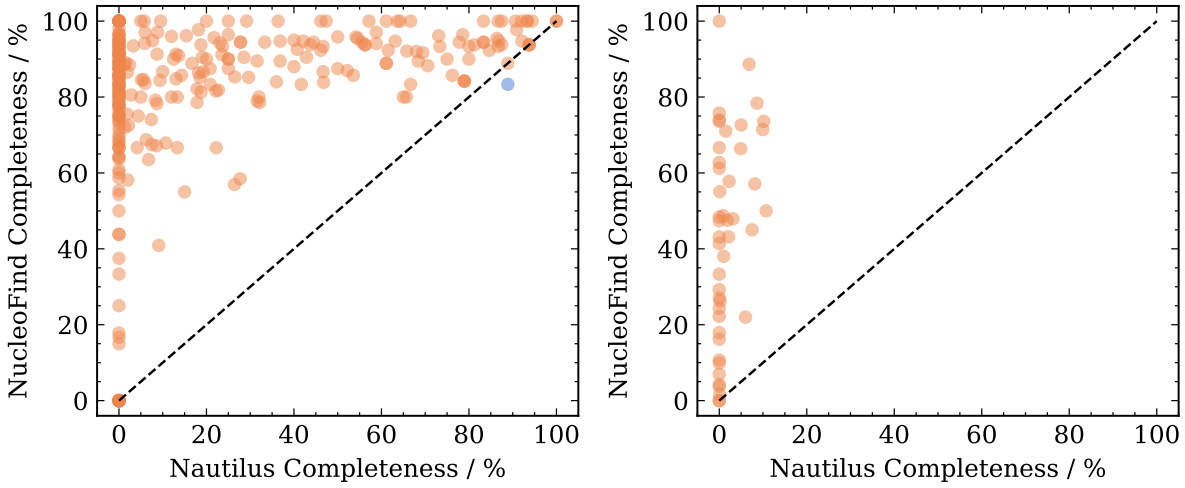
(a) Molecular replacement model **(b)** *ModelCraft* with *NucleoFind* model

Figure 4.12: Automated model building result for *ModelCraft* with *NucleoFind* for a protein-DNA complex resolved to 2.70 Å with X-ray crystallography (PDB code: 7CD6[265]). Molecular replacement was performed with a homologous model, and the result was passed to 5 cycles of *ModelCraft* with *NucleoFind*. The atomic model produced by *ModelCraft* with *NucleoFind* contains a peptide chain in the place where the *NucleoFind* prediction suggests there should be nucleic acid.

functionality in *Nautilus*, which is also used by *NucleoFind*. Given the relative uncertainty of positioned nucleotides in *Nautilus*, it is inappropriate to remove clashing amino acid residues in favour of potentially incorrect nucleotides. While appropriate for *Nautilus*, the deep learning predictions powering *NucleoFind* allow for much more confidence in any placed nucleotide, suggesting that an alternate pruning scheme that can remove incorrectly placed amino acids may be beneficial.

A stronger pruning method was added to *NucleoFind*, which removes any amino acid residues within 2 Å of any nucleotide modelled by the deep learning-based backbone building method. Since this method for building nucleotides is relatively rigorous, there is an implied confidence in the nucleic acid model, which can be used to ensure that only incorrectly built amino acids are removed. This method was chosen over a potential alternative pruning scheme, which would remove any amino acid modelled in positions with positive predicted density. While this method may work in many cases, it could become problematic in the rare instance of a false positive prediction.

ModelCraft with *NucleoFind* was rerun through both the X-ray crystallographic and cryo-EM test sets to confirm that the incomplete models were indeed caused by excessive protein modelling. Over the crystallographic test set, the average completeness of models was 81.3 ± 23.0 % and across the cryo-EM test set, the average completeness was 43.6 ± 26.2 %, shown in Figure 4.13. This large increase in performance compared to the previous version of *ModelCraft* with *NucleoFind* confirms that mismodelled amino acid residues were limiting performance. This may also be impacting the absolute perfor-



(a) X-ray diffraction **(b)** Cryo-EM

Figure 4.13: Completeness of models generated from automated nucleic acid model-building pipeline *ModelCraft* with nucleic acid model-building software packages *Nautilus*, or *NucleoFind* with stronger pruning. Orange points represent improvements in completeness for *NucleoFind* over *Nautilus*, blue points represent deterioration in completeness for *NucleoFind* over *Nautilus*. Completeness is defined as the proportion of nucleotide residues with all sugar and phosphate atoms within 2 Å of the deposited model. On average, *NucleoFind* outperforms *Nautilus* in a test set of crystallographic molecular replacement examples and a test set of cryo-EM examples.

mance of *ModelCraft* with *Nautilus*, which is likely to build more nucleic acids if there were some method for resolving the incorrectly built protein residues. Nevertheless, the final software method produces a statistically significant increase in model building performance over *ModelCraft* with *Nautilus*, with a 57.4 pp increase across crystallographic examples and a 41.6 pp increase in completeness with cryo-EM examples. Nucleic acids that were modelled in the original structure, but were not modelled by *ModelCraft* with *NucleoFind*, are likely the result of difficult-to-interpret or missing density. These unmodellable conditions must be critically considered by researchers before modelling and may be substantiated by other experimental evidence or prior knowledge.

Although the absolute completeness of models is dependent on the validity of the deposited structure used for comparison, the increase in nucleic acid model-building performance from replacing the existing software package *Nautilus* with *NucleoFind* is extremely promising. For an automated model-building method to be considered useful however, both protein and nucleic acid should be accurately modelled. The additional performance in nucleic acid model building exhibited by *ModelCraft* with *NucleoFind* should not come at the cost of weaker protein model building. This is a particularly important point to consider, since the stronger pruning method added to *NucleoFind*, by design, removes amino acids. A Wilcoxon signed-rank test between the protein completeness results of *ModelCraft* with *Nautilus* and *ModelCraft* with *NucleoFind* reveals no statistically significant difference across both the crystallographic and cryo-EM test sets.

This important result suggests that either the stronger pruning method does not remove valid amino acids, or that subsequent cycles of *ModelCraft* fix any incorrectly removed residues.

Completeness is a metric that can be calculated because the ground truth solution is available. However, during structure solution this is an unusual scenario, and other metrics are commonly favoured, such as R_{free} in crystallography and *FSC* in cryo-EM. *ModelCraft* outputs these metrics to measure structure solution success, however, using them as indicators of nucleic acid model-building performance is difficult since they are likely to be jointly dependent on other factors. The success of protein side-chain building, water placement, and the degree of overfitting all relate to these alternate metrics and were therefore not the primary focus of this analysis. The results of R_{free} and FSC across the crystallographic and cryo-EM test sets, respectively, are shown in Supplementary Section 8.2.

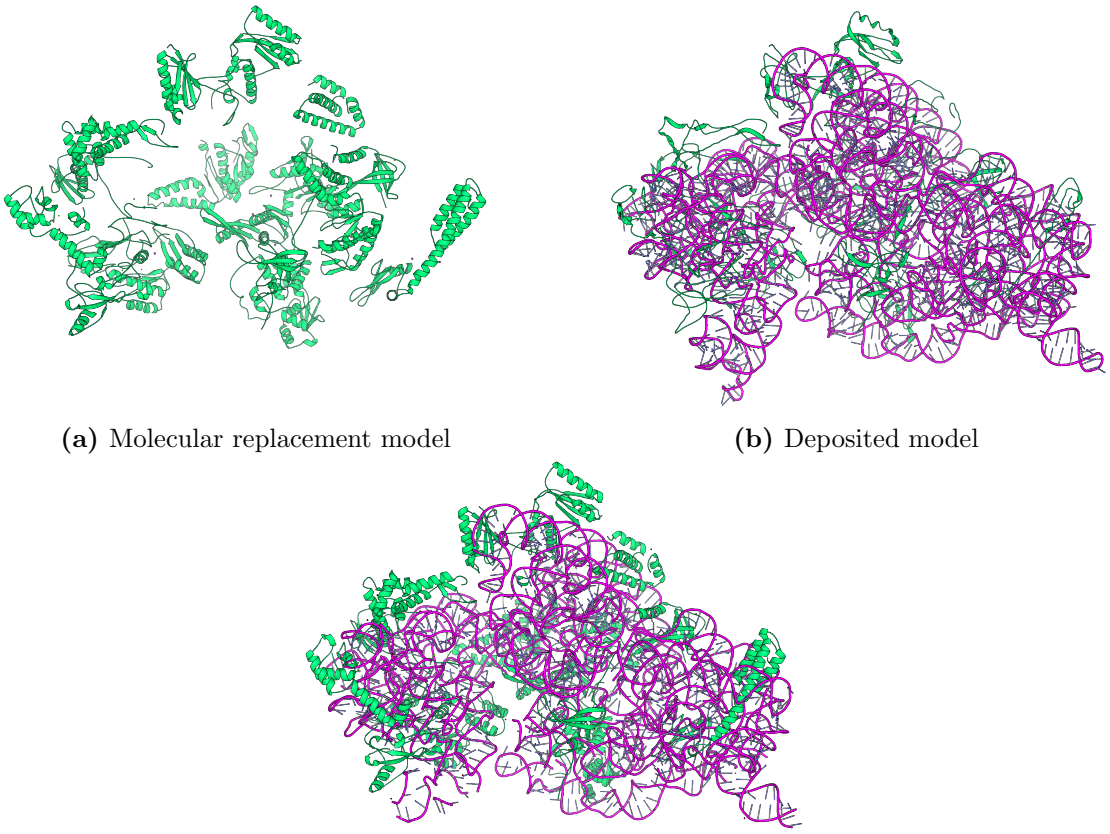
Across the crystallographic test set, each cycle of *NucleoFind* took 74 seconds on average when using 5 CPU cores. Contrastingly, one cycle of *Nautilus* took 12 seconds on average, suggesting that the cost of the significant performance increase is approximately 1 additional minute of run time per cycle. This time loss may be mitigated by allocating more CPU cores to *NucleoFind*, which is implemented so that additional available CPU cores reduce runtime linearly. The model-building performance of *NucleoFind* with cryo-EM examples, while an improvement over *Nautilus*, is not on par with the crystallographic examples and may be an area for future improvement. This relative lack of performance is likely caused by poorer deep learning predictions when working with cryo-EM Coulomb potential maps, as was observed in Chapter 3. To better understand where *NucleoFind* succeeds and fails, three case studies are considered.

4.5.1 Case Study 1: *de novo* building of *Thermus thermophilus* 30S ribosomal subunit from crystallographic data

Understanding the structure of ribosomal subunits has provided critical insight into the biochemistry underpinning protein synthesis.[266] Attempting to build such a large structure highlights several challenges, but perhaps the most difficult challenge after collecting data is obtaining sufficiently good phase estimates. When the 30S ribosomal subunit of *Thermus thermophilus* was initially solved, researchers used multiple experimental phasing strategies based on locating heavy atoms after heavy atom soaking.[267] With the advances in the number of available homologous structures in the Protein Data Bank, molecular replacement has become a more accessible option for obtaining phase estimates. When attempting to find a homologous model for a protein-nucleic acid complex such as the 30S ribosomal subunit, it is common to search only for homologous *protein* models.[152] Given the amount of scattering matter that may be missing from the protein molecular replacement solution of a large protein-nucleic acid complex, solving the structure after this point is non-trivial. To test the ability of *NucleoFind* to model the remaining nucleic acid from a difficult starting case, the structure solution process of the 30S ribosomal subunit was repeated, starting with 3.37 Å resolution merged reflection data (PDB code: 1IBK[268]). A homologous protein model from *Thermus thermophilus* HB8 (PDB code: 2VQF[269]) was found using *MrBump*[270] and used as a search for molecular replacement using *Phaser*.[237] The resultant molecular replacement solution was used as input to *ModelCraft* with *Nautilus* and to *ModelCraft* with *NucleoFind*, both with default parameters. It should be noted that this structure was not used in either the training or test sets of the deep learning model, whereas other ribosome structures were included. Considering that the phase estimations for the reflection data are likely updated during the model-building process, producing varying quality electron density, the inherent bias introduced by including ribosome structures in the deep learning training set is likely to be small. However, this potential source of bias should be considered when appreciating the model-building results.

In this example, *ModelCraft* with *NucleoFind* produces an outstanding result, building 1,338 out of the 1,525 nucleotides in the deposited model with a maximum sugar-phosphate atomic position deviation of 2 Å, as shown in Figure 4.14. This 87.8 % completeness score was achieved in 10 hours of completely automated model building on standard computational hardware, for which each *NucleoFind* run took approximately 6 minutes. In total, 19 cycles of *ModelCraft* were run, at which point the R_{work} metric plateaued for three cycles, causing model building to terminate. The remaining nucleic acids, which could not be modelled by *NucleoFind*, exist almost exclusively in regions of

particularly difficult to interpret electron density, likely caused by structural flexibility or structural heterogeneity. In contrast, *ModelCraft* with *Nautilus* was unable to model any correct nucleic acids, likely due to difficulties interpreting the density calculated using the initial model, in addition to potentially excessive protein model building.



(a) Molecular replacement model

(b) Deposited model

(c) *ModelCraft* with *NucleoFind* model

Figure 4.14: Automated model-building results for a crystallographic structure of the *Thermus thermophilus* 30S ribosomal subunit (PDB code: 1IBK[268]). Molecular replacement was performed using a homologous protein model from another 30S ribosomal subunit (PDB code: 2VQF[269]) with the result passed to *ModelCraft* with *NucleoFind* and ran for 25 cycles. The resultant structure is approximately 88 % complete with the remaining nucleotides existing in difficult to interpret electron density.

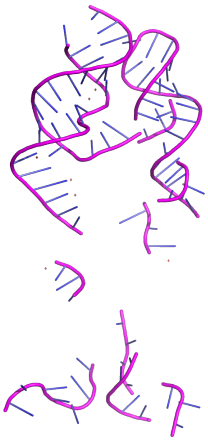
4.5.2 Case Study 2: *de novo* building of novel doubly pseudoknotted Rous sarcoma virus-programmed ribosomal frameshifting element from experimentally phased crystallographic data

Frame-shifting is a mechanism often exploited by viruses to encode more genomic information in a limited amount of nucleic acid. When a ribosome encounters specific sequence and structural elements, it can *slip* and continue translation in an offset *reading frame*.[271] The structure that dictated this behaviour in the Rous sarcoma virus was believed to contain an RNA pseudoknot, but was not confirmed by structural characterisation.[272] Recently, the three-dimensional structure was determined using X-ray crystallography to exhibit a novel double pseudoknot butterfly-like fold.[273] This structure was solved with SAD (see Section 1.1.5.3) at 2.80 Å resolution and has no homologues in the Protein Data Bank, and so may serve as an excellent test of the model-building power of *NucleoFind*. Given that the deep learning model behind *NucleoFind* has never seen an electron density map phased by SAD, a successfully built atomic model may suggest an inherent understanding of nucleic acid electron density characteristics.

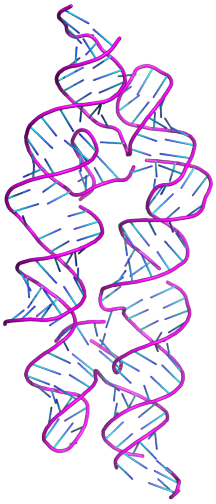
To recreate a realistic starting point for model building, the experimental phasing process was repeated using deposited anomalous intensity data from the Protein Data Bank. An iridium substructure containing 11 atoms was located using *SHELX*,[274] before density modification with *Parrot*.[275] The resultant substructure and reflection data was used as input to *ModelCraft* with *Nautilus* and *ModelCraft* with *NucleoFind* with default parameters. The results of model building, alongside the deposited model and predicted AlphaFold 3 model, are shown in Figure 4.15.

The AlphaFold 3 prediction for this novel fold is poor, with all regions exhibiting very low pLDDT scores, suggesting little confidence in the atomic positions. The lack of homologues in the Protein Data Bank, as well as poor *in silico* predictions of such nucleic acid structures, leaves only experimental phasing as a viable option for phase estimation in X-ray crystallography. The *ModelCraft* with *Nautilus* result yields a modest 31 % nucleic acid completeness, with large areas left unmodelled. In comparison, *ModelCraft* with *NucleoFind* produces an 86 % complete model, which is well modelled overall. Some of the terminal nucleotides remain unmodelled, and the inter-helix junctions are either missing or mismodelled. While a strong result for *ModelCraft* with *NucleoFind*, the small mistakes require some manual correction action.

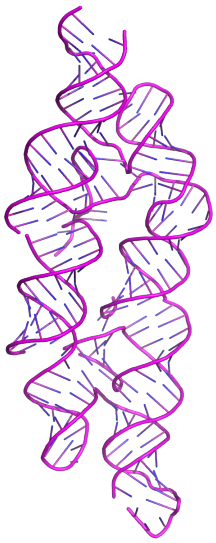
(a) AlphaFold 3 model



(b) *ModelCraft* with *Nautilus* model



(c) *ModelCraft* with *NucleoFind* model



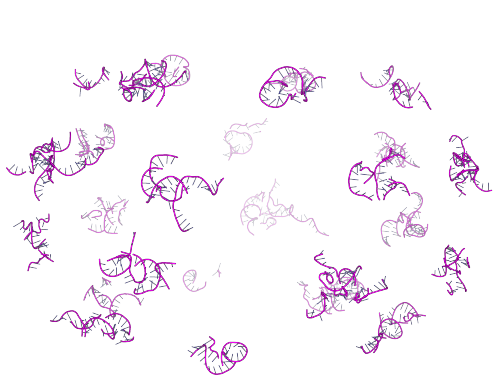
(d) Deposited model

Figure 4.15: Automated model-building results for a novel crystallographic structure of the Rous sarcoma virus frameshifting double pseudoknotted RNA (PDB code: 9DID[273]). Experimental phasing was performed using deposited anomalous intensity data using 11 iridium positions. The result was passed to *ModelCraft* with *Nautilus* and *NucleoFind* as nucleic acid model-building programs. *ModelCraft* with *Nautilus* yielded a 31 % complete model, whereas *ModelCraft* with *NucleoFind* yielded an 86 % complete result, but with mismodelling in the more complex regions of the novel fold.

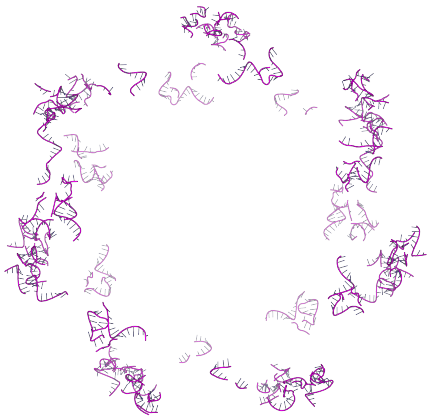
4.5.3 Case Study 3: *de novo* building of novel RNA nanocage from cryo-EM data

The discovery of non-coding RNAs transformed the view of RNA from a simple intermediate molecule in protein synthesis to a functional molecule with significant biological impact.[276] Genomic analysis revealed a set of non-coding RNAs present in bacteria and phages, which form large ordered secondary structures,[277] and have been recently structurally characterised using cryo-EM.[278] The Giant, Ornate, Lake, and Lactobacillales-Derived (GOLLD) RNA nanocage is the largest of these novel RNA-only structures, and so serves as a challenging but relevant case study for the model-building performance of *NucleoFind* from cryo-EM data. 5 cycles of *ModelCraft* with *Nautilus* and *ModelCraft* with *NucleoFind* were run using the two deposited cryo-EM half maps as inputs, with the resultant models shown in Figure 4.16.

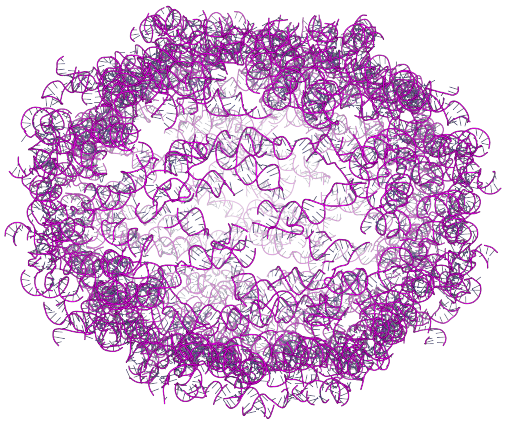
In total, *ModelCraft* with *Nautilus* modelled 632 of the 11,032 deposited residues correctly, whereas *ModelCraft* with *NucleoFind* produced a much more complete model with 8,138 correctly modelled residues. This significant performance improvement highlights the power of the deep learning predictions in identifying nucleic acid positions. When the deep learning model correctly identifies potential phosphate and sugar positions, the model-building procedures function well. However, this example of a complex RNA determined by cryo-EM highlights a flaw in this architecture. In regions of more difficult-to-interpret density, due to lower local resolution, absences in the predicted phosphate positions are very unlikely to yield a modelled nucleotide. This is particularly apparent in the outer regions of the RNA nanocage, where flexibility in the RNA chain likely leads to lower local resolution in the experimental density. Even if both the sugar and base positions were successfully predicted in these areas, without a phosphate prediction, there is no way for *NucleoFind* to build a successful model in these regions. It may be possible to rectify this deficiency in this particular example by averaging the predictions from *NucleoFind* across the multiple symmetric copies of the RNA cage, but a more robust future solution may be required that relies more heavily on the sugar or base predictions.



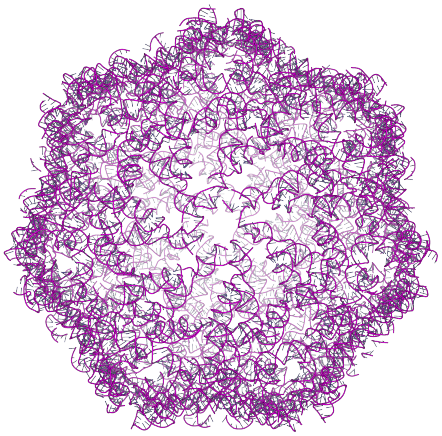
(a) *ModelCraft* with *Nautilus* model - Side



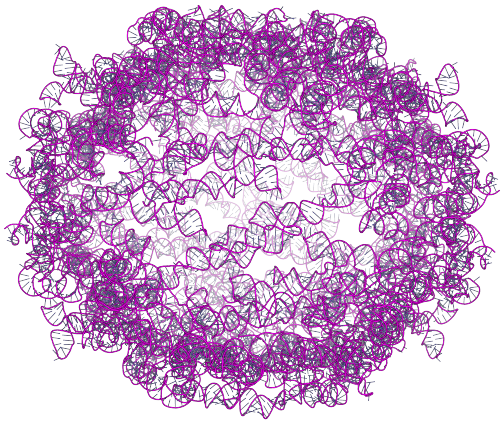
(b) *ModelCraft* with *Nautilus* model - Top



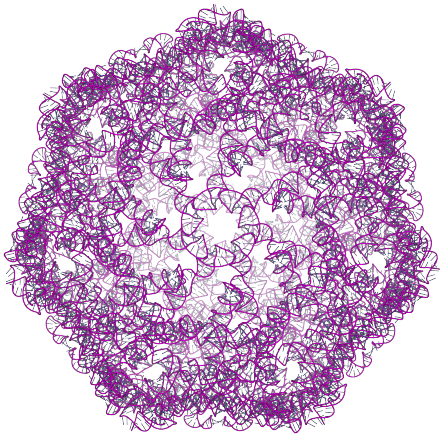
(c) *ModelCraft* with *NucleoFind* model - Side



(d) *ModelCraft* with *NucleoFind* model - Top



(e) Deposited model - Side



(f) Deposited model - Top

Figure 4.16: Automated model-building results for a cryo-EM structure of a GOLLD RNA nanocage. Two cryo-EM half maps with a full reconstruction resolution of 3.00 Å (PDB code: 9MEE[278]) was passed to *ModelCraft* with *Nautilus* and *ModelCraft* with *NucleoFind* to generate an atomic model. *ModelCraft* with *NucleoFind* produced a more complete model compared to *ModelCraft* with *Nautilus*, but failed to build many nucleotides in areas of lower local resolution.

4.6 Conclusions

In conclusion, the predictions produced by the deep learning model discussed in Chapter 3 provide outstanding context for the automated modelling of nucleic acids. The phosphate predictions are processed to yield singular points, which are then connected into a graph structure according to several restrictions. Fragments from a precomputed library are then superimposed over each connected triplet, and the best-fitting fragment is joined to form a chain. Modelled chains can then be further processed and sequenced to yield an atomic model of the nucleic acids, which is statistically significantly better than the automated model-building software package *Nautilus*. This new software package, named *NucleoFind*, was incorporated into the *ModelCraft* pipeline to produce more complete atomic models both from crystallographic and cryo-EM data.

The case studies presented highlight the improvement *ModelCraft* with *NucleoFind* has over *ModelCraft* with *Nautilus*, but critically, also identify the limitations of *NucleoFind*. Identifying the limitations of software is an essential step in methodological development and serves as the basis for future work. While *NucleoFind* is evidently a capable automated model-building package for nucleic acids, the assignment of bases remains a persistent problem that has yet to be addressed. *NucleoFind*, which uses the base building module from *Nautilus*, takes no account of base pairing in the assignment of bases, which is likely to lead to mistakes that require manual correction. Including base pairing in the probabilistic determination of the base may improve the assignment, which is likely to improve the model geometry after refinement.

In addition to the poor base assignment, *NucleoFind* is, by design, very dependent on the predicted maps output from the multiclass deep learning model. Areas of weaker, but still manually interpretable, density have been visually observed to produce worse predictions. Despite the possibility of *NucleoFind* outputting a probabilistic prediction for the phosphates, sugars and bases, the model-building procedure does not consider this. Future iterations of this method could include these probabilities as inputs to the model-building procedure, thereby improving the model-building accuracy of *NucleoFind* by modelling areas of poorer experimental density.

Overall, *NucleoFind* is a significant improvement over the existing method *Nautilus* and is scheduled to be incorporated into the *CCP4* software suite as a replacement for *Nautilus*.

Part II

Automated Model Building of Carbohydrates

Preface

Chapter 5 is partly based on the published article 'Analysis and validation of overall N-glycan conformation in Privateer'.[279] This paper, published in 2023, analysed the Protein Data Bank as it was in August 2021. Since then, thousands of additional structures have been deposited, so this analysis was repeated from scratch, although the techniques and processes were similar. The methods discussed in this chapter were made available via the software package *Privateer* as a web application[280] and a database.[201] Chapter 6 represents new, unpublished work.

Chapter 5

Analysis of Three-Dimensional Carbohydrate Conformations

5.1 Introduction

Glycosylation is an essential biochemical process which often induces significant changes to the structure of a protein.[281] After the co-translational or post-translational modification, the oligosaccharide chain exhibits a specific three-dimensional conformation, which is known to be critical for certain cellular functions.[282] Obtaining accurate models of these potentially complex conformations is key to understanding how some specific biological processes, such as virus-host interactions,[283] unfold. Unfortunately, due to the various stereochemical and regiochemical permutations exhibited by glycans, it is often difficult to obtain such information in the absence of structural data. Even with structural data, determining the exact carbohydrate moieties can be a significant challenge, given the flexibility and heterogeneity of the attached glycans. Glycans are often found on protein surfaces, allowing a high degree of mobility and potentially increasing the difficulty of interpretation of experimental data from both X-ray crystallography and cryo-EM. Furthermore, even if intramolecular interactions rigidly stabilise a glycan, the specific carbohydrate types present within each glycan can vary depending on the degree of enzymatic processing, further reducing interpretability when multiple crystal copies or particles are averaged. Perhaps it is due to these challenges that some structures deposited into the Protein Data Bank contain improbable or incorrect carbohydrate models.[162]

5.2 Aims

Given the challenges often encountered when modelling carbohydrates, the goal of this study is to develop a software method that automatically and consistently models carbohydrates into experimental density. Since glycans are highly conformationally diverse, it is imperative to obtain a strong foundational understanding of glycan geometry. The

three-dimensional model of a glycan chain can be described by three properties: the identity of the sugars, the conformation of the individual sugars and the relationship between individual sugars. It is well understood that individual sugars commonly exist in either the $^4C_1$ or $^1C_4$ chair conformations, but the relationship between individual sugars depends on numerous factors, such as sugar type, degree of branching, and proximity to the protein surface. Previous attempts at understanding these relationships involved creating databases of disaccharide torsion angles from structures deposited in the Protein Data Bank, such as *carp*.[284] However, these databases predate the routine validation of carbohydrate ring conformations,[91] so a new analysis is needed to incorporate only validated carbohydrates into a new geometric library. By understanding the linkage conformations found across many structures in the Protein Data Bank, trends can be extracted for use in downstream applications.

5.3 Library Generation

5.3.1 Methods

To generate the geometric library, glycosylated amino acid residues were searched for in the Protein Data Bank using the RCSB Search API.[239] The results were filtered to include only structures solved with either X-ray crystallography or cryo-EM. For each bound glycan chain found, all monosaccharide conformations were validated using *Privateer*, which computes the modelled ring conformation using the Cremer-Pople algorithm.[285] While this check alone determines a degree of validity, it is crucial to evaluate whether experimental data support the carbohydrate. Provided the experimental data is available, the *RSCC* can be calculated for each monosaccharide, which assesses the agreement between the model and experimental density. As the goal of this library is to understand the relationships between linked monosaccharides across many real structures, a strict *RSCC* threshold of 0.80 was selected. Additionally, *Privateer* checks to ensure that the nomenclature for a carbohydrate is consistent with the model, ensuring that any downstream analysis is valid. Although no experimental resolution filter was used, the strict *RSCC* cutoff applies an implicit resolution filter since it is often difficult to achieve a high *RSCC* with lower resolution experimental data.

The geometric relationship between linked monosaccharides can be defined by a set of torsion angles, bond angles, and a bond length. Since the elemental composition of the linkage heavily constrains the bond angles and bond length, the overall conformation is most aptly described with a set of three torsion angles, shown in Figure 5.1 and defined in Equations 5.1 to 5.3. These torsion angles can be trivially calculated from the atomic positions of validated linked monosaccharides, with minimal computational effort using

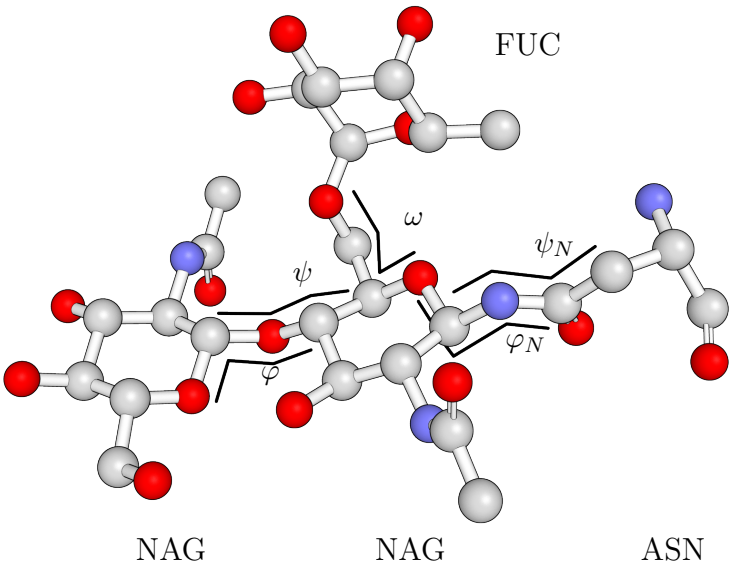


Figure 5.1: Torsion angle definitions for protein-sugar and sugar-sugar linkages.

GEMMI. From the curated dataset of monosaccharides, potential linkages can be identified between any pair of monosaccharides, provided that there are donor and acceptor atoms within an appropriate bonding distance of approximately 1.45 Å. Given that both monosaccharides have been validated to fit the density well and are in the lowest energy conformation, the torsion angles between the two monosaccharides can be calculated and recorded with confidence. The bond angles and bond lengths can also be collected for completeness, though further analysis will focus only on the torsion angles. The means and standard deviations reported refer to circular means and standard deviations due to the periodic nature of angles.

$$\psi = \text{dihedral}(\text{C}_{n-1}^{donor}, \text{C}_{n}^{donor}, \text{O}_{n}^{donor}, \text{C}_{1}^{acceptor}) \tag{5.1}$$

$$\varphi = \text{dihedral}(\text{C}_{n}^{donor}, \text{O}_{n}^{donor}, \text{C}_{1}^{acceptor}, \text{O}_{5}^{acceptor}) \tag{5.2}$$

$$\omega = \text{dihedral}(\text{X}_{n-2}^{donor}, \text{C}_{n-1}^{donor}, \text{C}_{n}^{donor}, \text{O}_{n}^{donor}) \tag{5.3}$$

5.3.2 Results and Discussion

The survey through the Protein Data Bank highlighted several validated linkage types from *N*-glycosylation, *O*-glycosylation, and *C*-glycosylation. To ensure meaningful statistics could be drawn from each linkage type, only linkages with at least 50 validated occurrences were considered for further analysis, yielding 11 linkages in *N*-glycosylation, 2 linkages from *O*-glycosylation and the only known linkage from *C*-glycosylation. Table 5.1 shows the full denomination and abbreviations for each validated linkage identified, and Supplementary Section 8.3 reports the collated and clustered results.

Table 5.1: Full linkage denomination, abbreviations and CCD codes for identified validated linkages obtained through searching the Protein Data Bank. Each linkage has greater than 50 validated occurrences, where validity is determined by conformational and density fit validation using *Privateer*.

(a) *N*-glycosylation

Full Linkage Denomination	Abbreviation	CCD Code
N-Acetyl-*β*-D-glucosamine–asparagine	GlcNAc–*β*–Asn	NAG-ASN
N-Acetyl-*β*-D-glucosamine–1,4–*N*-acetyl-*β*-D-glucosamine	GlcNAc–*β*1, 4–GlcNAc	NAG-1,4-NAG
β-D-Mannose–1,4–*N*-acetyl-*β*-D-glucosamine	Man–*β*1,4–GlcNAc	BMA-1,4-NAG
α-D-Mannose–1,3–*β*-D-mannose	Man–*α*1,3–Man	MAN-1,3-BMA
α-D-Mannose–1,6–*β*-D-mannose	Man–*α*1,6–Man	MAN-1,6-BMA
α-D-Mannose–1,2–*α*-D-mannose	Man–*α*1,2–Man	MAN-1,2-MAN
α-D-Mannose–1,3–*α*-D-mannose	Man–*α*1,3–Man	MAN-1,3-MAN
α-D-Mannose–1,6–*α*-D-mannose	Man–*α*1,6–Man	MAN-1,6-MAN
α-L-Fucose–1,3–*N*-acetyl-*β*-D-glucosamine	Fuc–*α*1,3–GlcNAc	FUC-1,3-NAG
α-L-Fucose–1,6–*N*-acetyl-*β*-D-glucosamine	Fuc–*α*1,6–GlcNAc	FUC-1,6-NAG
N-Acetyl-*β*-D-glucosamine–1,2–*α*-D-mannose	GlcNAc–*β*1,2–Man	NAG-1,2-MAN

(b) *C*-glycosylation

Full Linkage Denomination	Abbreviation	CCD Code
α-D-Mannose–tryptophan	Man-*α*-Trp	MAN-TRP

(c) *O*-glycosylation

Full Linkage Denomination	Abbreviation	CCD Code
α-D-Mannose–serine	Man-*α*-Ser	MAN-SER
α-D-Mannose–threonine	Man-*α*-Thr	MAN-THR

5.3.2.1 Protein-Sugar linkages

The relationship between glycosylated amino acid residues and the corresponding attached pyranose sugars is perhaps the most important geometric property of a glycan chain. This geometric relationship dictates the general topology of a glycan chain, and may also be important in defining the degree of accessibility of a glycan to further glycosylation or processing. The most frequently observed validated linkage in the Protein Data Bank is the first linkage of an *N*-glycan chain between the side chain of asparagine and GlcNAc. The sidechain of asparagine is an amide group and when linked to a pyranose sugar, it would be chemically reasonable to expect the amide group to remain planar, with a resultant torsion angle across the amide bond, ψ, close to 180 °. Indeed, across the 29,671 validated occurrences of this linkage, the average ψ angle is 179.2 ± 23.5 °. The torsion angle φ by definition includes the oxygen atom of the pyranose ring and describes only single bonds. It may therefore be expected that there is greater chemical freedom for the atoms this torsion angle describes. This is reflected well in the greater spread of validated φ angles for GlcNAc–β–Asn, with an average value of -101.6 ± 38.1 °.

The remaining three protein-sugar linkages identified were between α-D-mannose and the side chains of amino acids threonine, serine and tryptophan, respectively. Much like the planar amide bond of the asparagine sidechain, the planar indole moiety of the tryptophan sidechain restricts Man-α-Trp linkages to have a ψ angle of approximately 180 °. This is observed in the distribution of torsion angles across 53 validated Man-α-Trp linkages with an average ψ angle of 183.1 ± 6.0 °. The φ values are relatively conserved across all examples with an average value of 120.0 ± 14.0 °. The positive φ torsion angle is expected considering the $^1C_4$ conformation adopted by the Man monosaccharide to minimise unfavourable steric interactions.

A higher degree of conformational variability is observed for two *O*-glycosylation linkages, Man-α-Ser and Man-α-Thr. The attachment to the α-D-mannose occurs through an alcohol moiety in both serine and threonine, which contains no rigid double bond to restrict torsion angles to near planarity. Subsequently, both linkages are observed in varying conformations across validated entries in the Protein Data Bank. The spread of values is similar between these two linkages with average torsion values of ψ = 172.9 ± 30 ° and φ = 69.2 ± 21.9 ° for Man-α-Ser, and ψ = 129.9 ± 25 ° and φ = 88.0 ± 28.5 ° for Man-α-Thr.

Two-dimensional histograms for each of these four protein-sugar linkages are shown in Figure 5.2 with the frequency of occurrence shown by colour. The three mannosylated amino acid sidechains in this analysis show a clear single conformational cluster, which is a useful property for downstream model-building applications. To model glycosylation

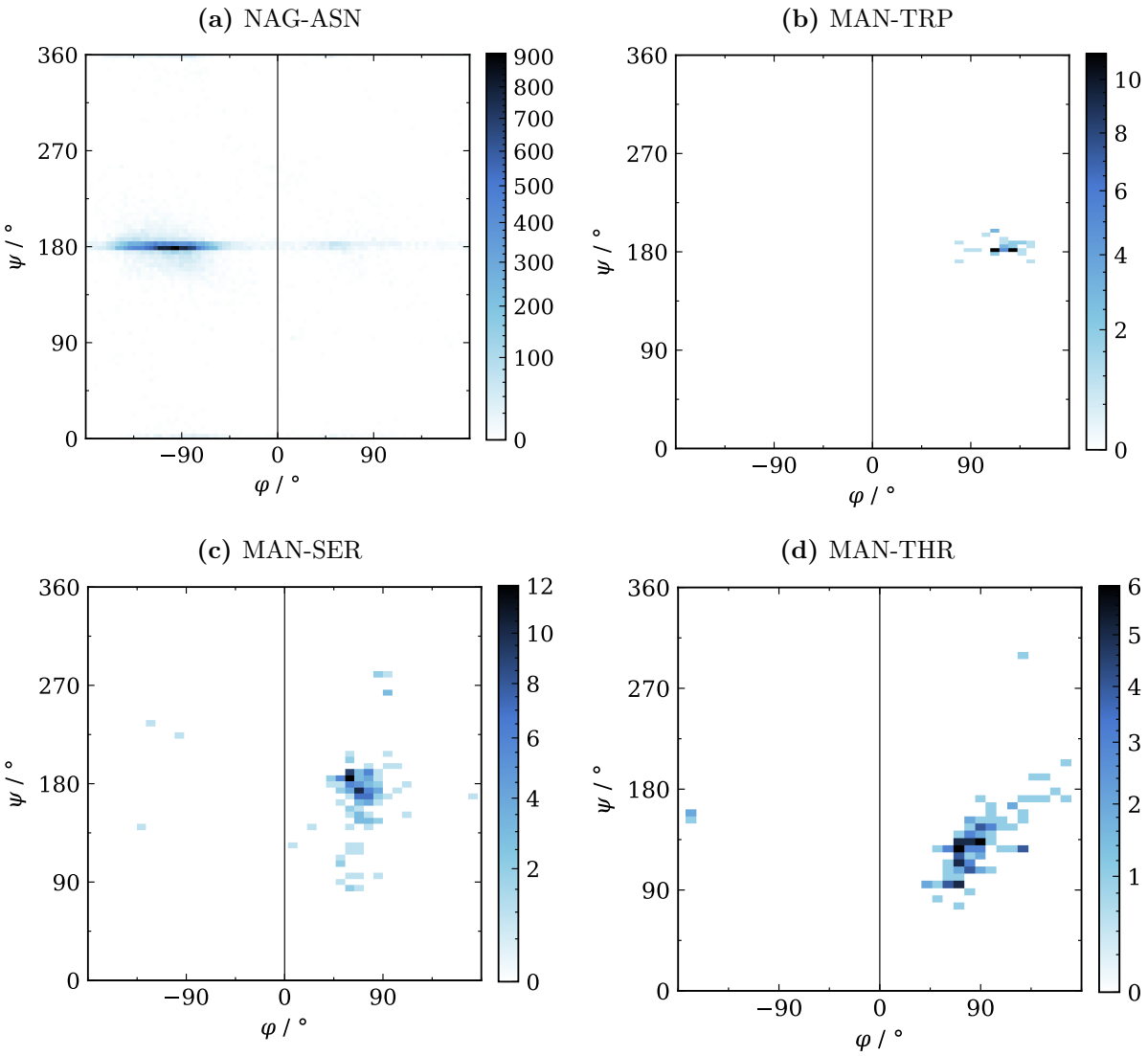
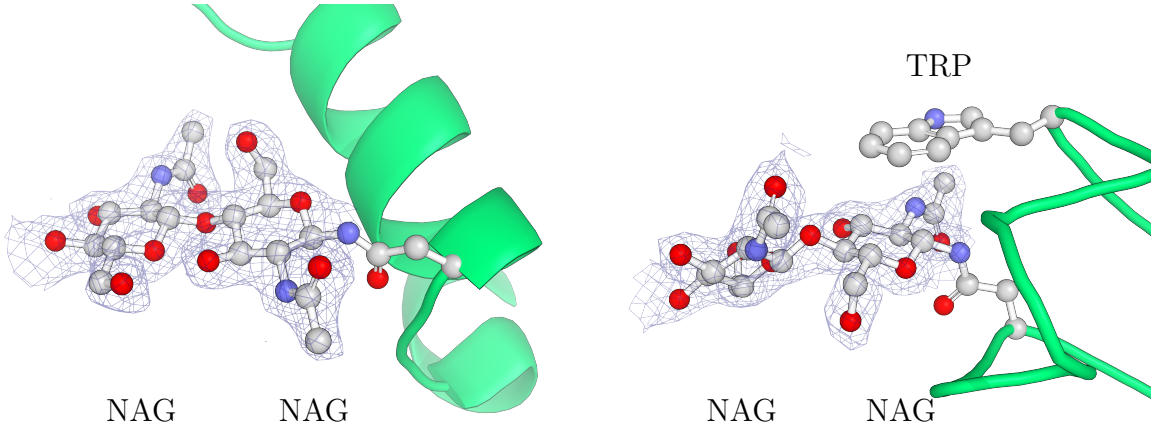


Figure 5.2: Two-dimensional histograms showing ψ and φ torsion angles for validated protein-sugar linkages found in a survey of the Protein Data Bank. Only linkages with greater than 50 validated occurrences are shown, with frequency denoted by colour. The colour bar is plotted with the Power Law distribution[286] to aid visualisation of less frequent bins.

onto these amino acid side chains, a sugar needs to be attached in only a single geometric conformation, whereas if a linkage were to exhibit multiple distinct conformational clusters, the complexity of any method would undoubtedly increase. Despite the additional downstream complexity, since all linkages have been validated, it is crucial to understand any additional conformational clusters highlighted by this analysis. For example, two distinct conformational clusters are visible from the ψ, φ histogram for the GlcNAc–β–Asn linkage, suggesting that while the main conformational cluster with a negative φ angle is preferred, an alternative conformation is observed in the Protein Data Bank. Understanding how these secondary conformations occur may provide greater confidence in the determination of validity. Visually inspecting many of these minor conformations high-



(a) Major conformation **(b)** Minor conformation

Figure 5.3: Major and minor conformations of the NAG-ASN linkage highlighted from surveying the Protein Data Bank for validated linkages. Across the validated dataset, the major conformation is most commonly observed, however, if a stabilising CH-π interaction is present via a neighbouring tryptophan residue, the minor conformation may be allowed. Both of these conformations are seen in fungal GH3 β-glucosidase protein solved to 1.95 Å resolution, where all sugars shown have been validated by *Privateer* (PDB code: 5FJI[186]).

lights a specific CH-π interaction between a local tryptophan residue and GlcNAc that may aid the stability of this conformation.[287] Figure 5.3 illustrates an example from a fungal GH3 β-glucosidase, which is conserved across homologous structures (PDB code: 5FJI[186]).

5.3.2.2 Glycosidic Linkages Between Pyranosides

In *N*-linked and *O*-linked glycosylation, after the addition of a monosaccharide onto the sidechain of an amino acid, the glycan chain can be extended with a wide array of monosaccharides to form a diverse range of oligosaccharides. Commonly, these chains exhibit high variability and branching, resulting in a complex overall glycan conformation. To understand and estimate the conformation of an entire glycan, it is helpful first to consider the distribution of favourable conformations among validated linkages. The survey of the Protein Data Bank highlighted 10 linkages which meet the analytical criteria, all of which arise in *N*-glycans. The torsion angle distributions for all identified linkages are shown in Figure 5.4, with linkages described using the CCD code format for clarity.

N-glycosylated chains attach to the asparagine side chain through a β-linked GlcNAc monosaccharide, followed by another β-1,4 linked GlcNAc monosaccharide. This NAG-1,4-NAG linkage is the most common sugar-sugar linkage found in the Protein Data Bank, with 6,772 validated occurrences. The conformation of this abundant linkage is generally well conserved with ψ = -128.0 ± 22.4 ° and φ = -79.0 ± 23.0 °. Following the second Glc-

NAc sugar, a β-linked Man sugar is commonly attached through another β-1,4 linkage. BMA-NAG and NAG-NAG share a similar torsion angle distribution with ψ = -132.4 ± 24.4 ° and φ = -84.9 ± 27.6 °. This is unsurprising given that both GlcNAc and β-D-Man share a common D-pyranose core and are attached through the same β-1,4 linkage.

The terminal monosaccharides of an *N*-glycan chain depend on the degree of enzymatic processing the chain has been subjected to. With high mannose *N*-glycans, chains often terminate with highly branched Man monosaccharides, whereas with complex or hybrid *N*-glycans, other monosaccharide types can be positioned terminally. α-Man monosaccharides are most commonly found attached to β-Man monosaccharides through an α-1,3 linkage, with 1,298 validated occurrences observed across the Protein Data Bank. However, the positional isomer which attaches through a α-1,6 linkage, has a similar number of validated occurrences at 1,136. These two MAN-BMA linkages exhibit varied torsional distributions. The geometry of MAN-1,3-BMA is relatively well conserved with $\psi = 122.7 \pm 30.4$ ° and $\varphi = 80.6 \pm 24.9$ ° whereas MAN-1,6-BMA varies much more with $\psi = 171.6 \pm 40.7$ ° and $\varphi = 82.0 \pm 34.4$ °. The additional spread of the α-1,6 linkage is understandable when considering the additional rotatability of the exo $C_6 - O_6$ hydroxyl group. Certain glycoproteins exhibit further glycosylation through various MAN-MAN and NAG-MAN linkages, which can increase the complexity and branching of the glycan chain. After the first α-Man monosaccharide, the glycan chain can be further mannosylated via α-1,2, α-1,3, or α-1,6 linkages. Both MAN-1,2-MAN and MAN-1,3-MAN show a similar torsion angle distribution with $\psi = 133.9 \pm 29.3$ ° and $\varphi = 84.3 \pm 22.7$ ° for the α-1,2 linkage and $\psi = 129.9 \pm 31.2$ ° and $\varphi = 82.5 \pm 20.8$ ° for the α-1,3 linkage. Analogous to the MAN-1,6-BMA linkage, the MAN-1,6-MAN linkage exhibits a higher degree of spread when compared to the more rigid α-1,3 linkage with ψ = -174.4 ± 30.3 ° and $\varphi = 77.9 \pm 33.8$ °. Again, this is likely due to the additional flexibility of the exo-hydroxyl group.

The three least common validated linkages observed in this survey occur only in hybrid or complex *N*-glycan trees. With complex *N*-glycans, the first GlcNAc sugar can be fucosylated through either a α-1,3 or α-1,6 linkage. From the 124 validated occurrences of the FUC-1,3-NAG linkage, the torsion angles seem to prefer a very rigid range of values with $\psi = 137.6 \pm 14.6$ ° and φ = -70.9 ± 12.5 °. The more common FUC-1,6-NAG linkage, much like other α-1,6 linkages, adopts a wider array of conformations with $\psi = 167.9 \pm 42.4$ ° and φ = -79.3 ± 23.5 °. The final linkage type identified in this analysis occurs in both hybrid and complex *N*-glycan trees. After mannosylation, a new GlcNAc sugar can be linked to a terminal α-Man through a β-1,2 linkage. Despite the general flexibility of an oligosaccharide chain with such a large extent, this linkage shows a small torsion angle distribution with $\psi = 148.4 \pm 20.2$ ° and φ = -85.2 ± 20.5 ° across 185

validated occurrences.

Analysis of these sugar-sugar linkages yields a clear trend in the value of the φ torsion angle. The most common φ values appear to be determined by the anomeric form of the monosaccharide which accepts the glycosidic linkage. For D-pyranosides, this means that the β-anomer should exhibit negative φ values ($\varphi \in [-180, 0]$) and α-anomers should exhibit positive φ values ($\varphi \in [0, 180]$). This trend, along with the general torsion angle distribution of validated linkages, provides a strong understanding of the geometric relationships between the most commonly linked monosaccharides.

This survey also reveals that certain geometric occurrences lie far from the average geometry. Since both monosaccharides are explicitly validated by *Privateer*, these abnormal geometries likely result from interesting biochemical interactions that may be considered in downstream applications.

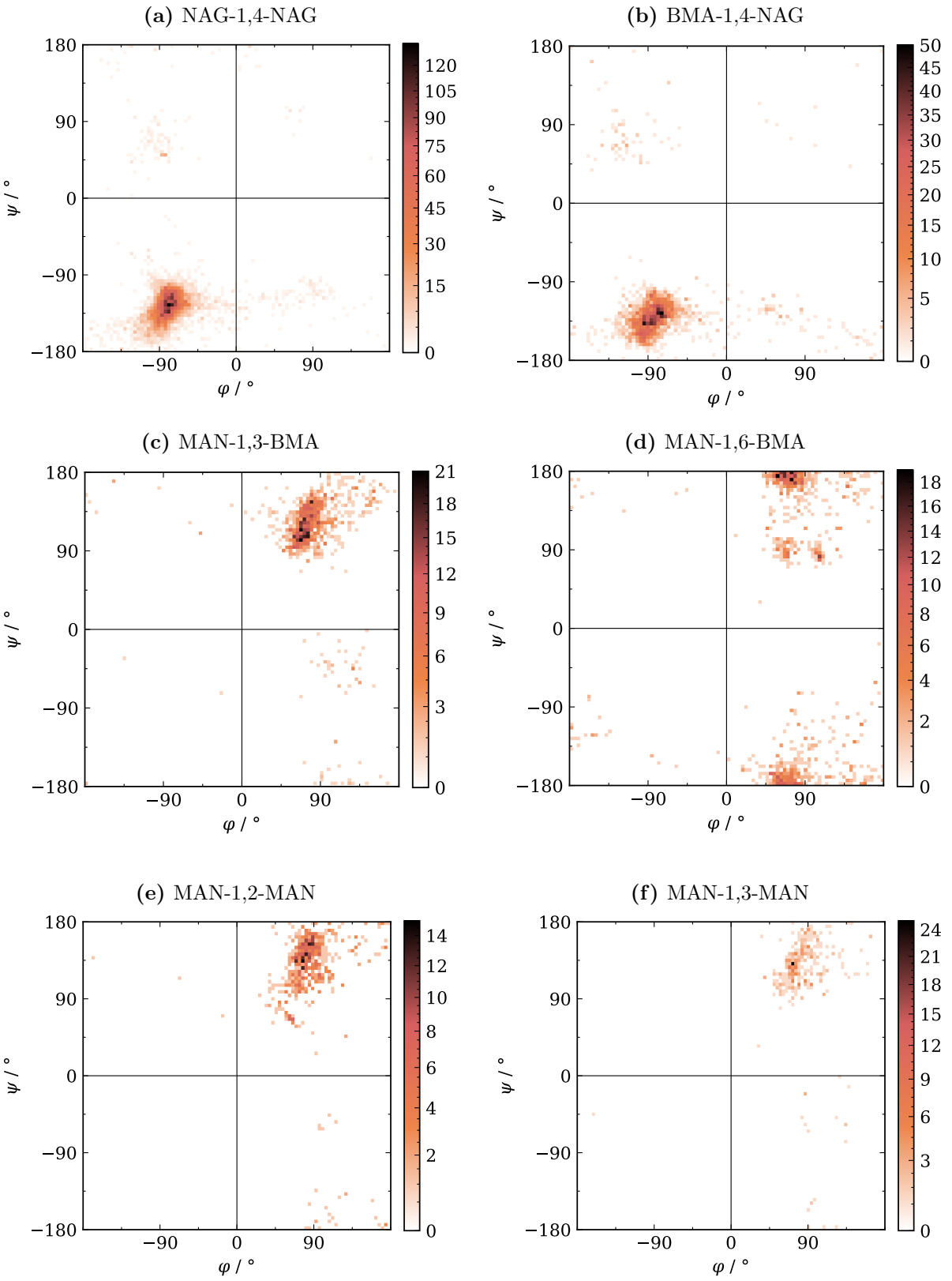


Figure 5.4: Two-dimensional histograms showing ψ and φ torsion angles for validated sugar-sugar linkages found in a survey of the Protein Data Bank. Only linkages with greater than 50 validated occurrences are shown, with frequency denoted by colour. The colour bar is plotted with the Power Law distribution[286] to aid visualisation of less frequent bins.

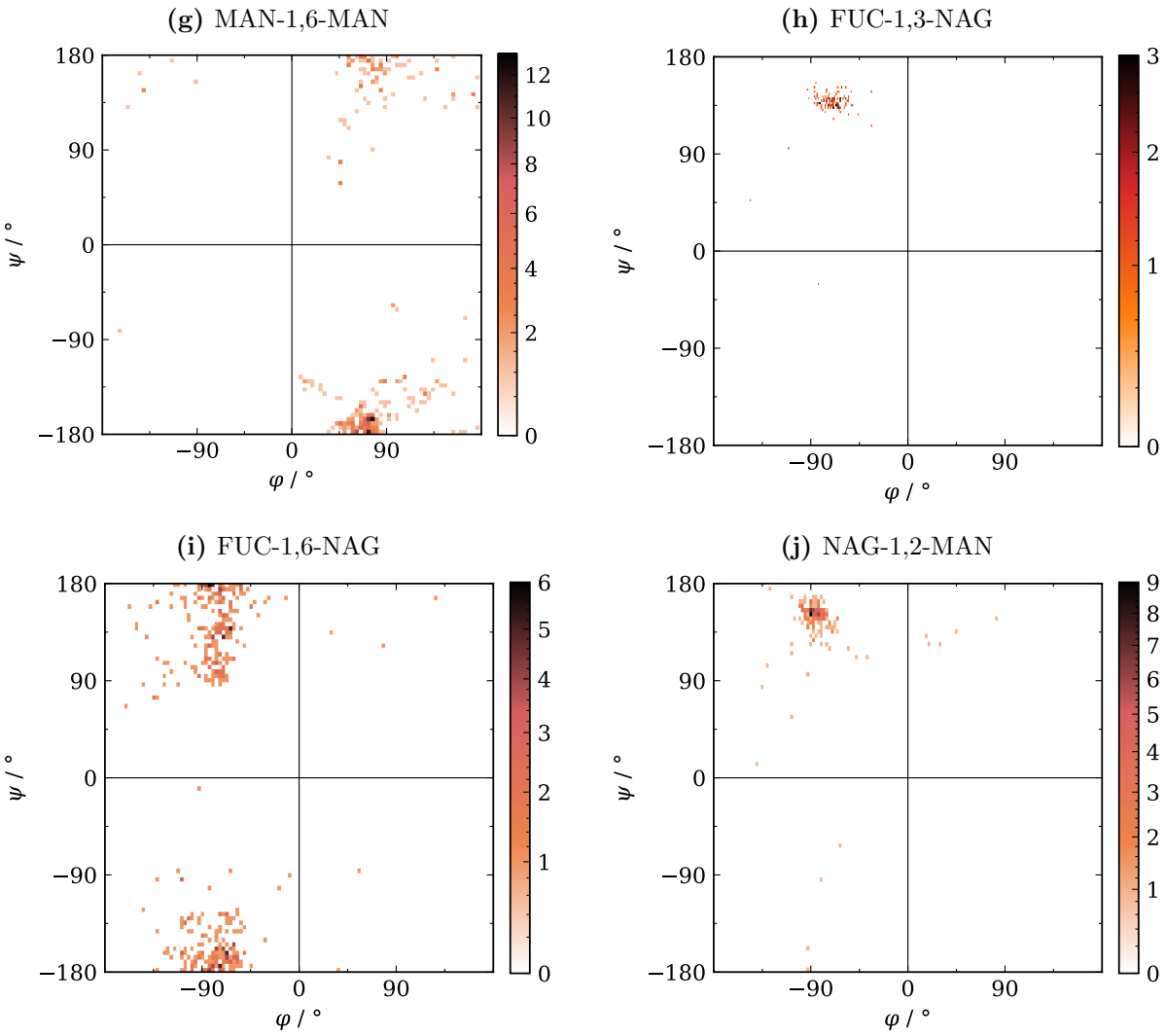


Figure 5.4 (continued): Two-dimensional histograms showing ψ and φ torsion angles for validated sugar-sugar linkages found in a survey of the Protein Data Bank. Only linkages with greater than 50 validated occurrences are shown, with frequency denoted by colour. The colour bar is plotted with the Power Law distribution[286] to aid visualisation of less frequent bins.

5.3.2.2.1 Interactions Stabilising Uncommon Glycosidic Linkages

In the absence of any external factors, a glycosidic linkage will likely exist in the most stable conformation. Despite this, the observed spread of data in the Protein Data Bank suggests that glycosidic linkages with conformations far from the expected ideal are stable. Since glycosidic linkages are not commonly found in isolation in the Protein Data Bank, these deviations must be explained by specific protein-sugar interactions which stabilise uncommon conformations.

The functionality and stability of certain proteins are attributed to repulsive and attractive electrostatic forces, which are generally mediated by amino acid side chains.

Some side chains are positively charged, such as lysine and arginine, while others are negatively charged, such as aspartic acid and glutamic acid.[288,289] Depending on the positions of these electrostatic side chains, they may also interact with the location of other biological molecules, such as carbohydrates, nucleic acids or ligands.

Figure 5.5 illustrates an example of a MAN-1,2-MAN linkage, with torsion angles which deviate from the modal linkage conformation. This linkage is found in a receptor-like kinase BRI1 (PDB code: 4J0M[290]) resolved to 2.50 Å resolution using X-ray diffraction. Since both sugars are deemed correct by *Privateer* and can be visually observed to explain the experimental density well, this uncommon linkage conformation is unlikely to arise from erroneous model building or refinement. Examining both sugars reveals a network of electrostatic interactions, which are likely influencing the positions of both monosaccharides. A hydrogen bond is likely to exist between the amino acid side chain of a local histidine residue (HIS-286) and the O4 hydroxyl group in MAN-5. Additionally, the presence of a potentially positively charged arginine group close to MAN-4 and MAN-5 could more generally influence the position of the glycan chain. Another potential hydrogen bond between the terminal MAN-6 sugar and a proximal asparagine sidechain may also affect the position of this terminal sugar. Since hydrogens are not well resolved in this 2.50 Å resolution electron density map, the positions of hydrogens cannot be directly deduced to confirm hydrogen bonding. However, the environment of this uncommon MAN-1,2-MAN conformation suggests that strong protein-sugar interactions indeed influence the

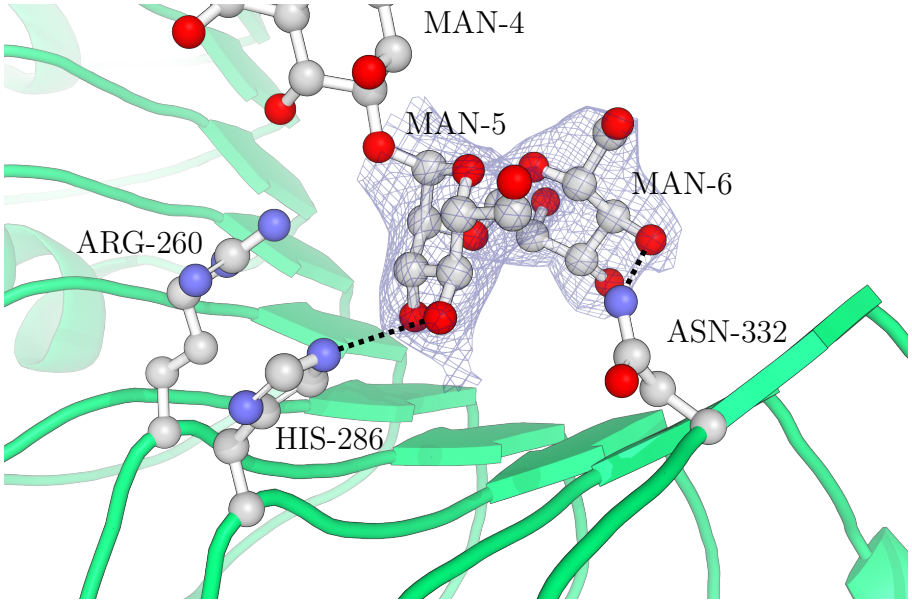


Figure 5.5: An uncommon conformation of the validated MAN-1,2-MAN linkage found in a receptor-like kinase structure (PDB code: 4J0M[290]) with φ = 82.9 ° and ψ = -179.8 °. Both MAN-5 and MAN-6 sugars fit the electron density well, suggesting that this uncommon conformation is stabilised by some protein-carbohydrate interaction. In this example, both electrostatic interactions between a potentially charged arginine side chain and two hydrogen bonds from a histidine and an asparagine side chain stabilise this conformational distortion.

observed conformation.

5.3.3 Conclusions

The survey through the Protein Data Bank revealed numerous validated protein-sugar and sugar-sugar glycosidic linkages, enabling an understanding of the expected geometric relationships within the most frequently structurally validated glycans. These geometric relationships are generally similar to those defined in prior studies,[291] but biochemical inference can be made with greater confidence because this analysis surveyed more data and because each linkage has been deemed valid. Furthermore, investigating the electrostatic interactions underlying uncommon glycosidic linkage conformations provides a more nuanced understanding of the intricate interactions at play within glycans. The knowledge curated through this survey is instrumental for accurate novel glycosidic linkage model building, but may also be used to validate and remodel existing glycoprotein structures.

Direct extrapolation from this dataset should be done with some caution. The data analysed here represent only well-resolved structural data and take no account of redundancy or completeness. If a specific linkage conformation does not appear in this dataset, it does not mean that that linkage is automatically improbable. To alleviate this flaw, it is necessary to understand the bounds beyond which all conformations become improbable. In combination with the most common conformations observed in the Protein Data Bank, any calculated conformational probability may be used directly to validate existing structures or, possibly, as prior information in an automated carbohydrate model-building software package.

5.4 *Ab initio* Conformational Analysis

When attempting to model a glycosidic linkage, it is helpful to have knowledge of which conformations are energetically favourable. These favourable geometric regions may be used as a guide for interactive model building, as bounds for automated model building, or as alert thresholds for validation software packages. Methods which utilise similar prior-chemical knowledge are beneficial when experimental density is poor, as is often the case with glycans.[162] Energetic information for carbohydrates is most commonly calculated using molecular mechanics measurements,[292] or from quantum mechanical calculations.[293,294] The two approaches balance energetic accuracy and computational feasibility, but since molecular mechanics approaches commonly rely on precomputed libraries for comparison, a quantum mechanical calculation is more appropriate for this study.

5.4.1 Methods

For each of the most common validated glycosidic linkages observed in the Protein Data Bank, shown in Table 5.1, a set of conformers must be generated that sufficiently samples the ψ, φ torsional space. The other torsion angle, which commonly describes a torsional linkage, ω, was excluded from this analysis since increasing the dimensionality of the sample space would render the analysis computationally infeasible, likely requiring months of computation. Additionally, only free linkages were considered to remove any bias from interacting biological macromolecules and enable energetic calculations to become computationally feasible. A 6 ° torsion angle increment was chosen to ensure a reasonable conformational fidelity whilst remaining computationally efficient. While the ψ and φ torsion angles vary, the ω torsion angle and all bond angles were set to the mean value obtained through the survey of the Protein Data Bank described in Section 5.3. This process yields 3,600 conformations, which can be assessed to determine which are energetically favourable.

Energetics were obtained using Density Functional Theory[295] (DFT) in *Gaussian16*,[296] where a single point energy was calculated for each conformation using DFT-B3LYP and the 6-31G(d,p) basis set. Given the relatively large number of conformations, this specific set of functionals should provide sufficient accuracy for a comparative study whilst remaining computationally feasible. While it may be true that other basis sets or DFT methods may predict the absolute energy of each conformation more accurately, the main focus of this work is not on quantum chemical theory but on applying these methods to understand which linkage conformations are likely the most energetically favourable. Following the calculation of a single point energy for each conformation of a given linkage, relative energies can be calculated by simply normalising the absolute energies with re-

spect to the minimum and maximum values, yielding a normalised energy between 0 and 1. In this analysis, energy refers to the self-consistent field energy, as calculated by the Kohn-Sham equation,[295] which accounts for the interactions between molecular orbitals.

5.4.2 Results and Discussion

5.4.2.1 Protein-Sugar Linkages

Normalised energetic contour diagrams are shown in Figure 5.6 for the 4 protein-sugar linkages identified through the survey of the Protein Data Bank. Contour diagrams are useful when attempting to identify regions of conformational space which are energetically favourable, and regions which require more substantial external driving factors to form. These diagrams have been normalised purposefully to retract any meaning from the absolute energetic differences, which are not comparable between linkage types. The lightest region on the diagram represents the most energetically stable conformations, and the darkest region on the diagram represents the least energetically stable conformation. It should be noted that these plots are weighted relative to the worst conformational energy, which aggregates many different conformations into the same colour due to the dynamic range of the plots. An alternate approach could be to compute the relative probability of each conformation using the Boltzmann equation, which may provide insight into relative stability. This approach was not pursued, but may be an excellent avenue for future work.

The two planar protein donor groups of asparagine and tryptophan were theorised to restrict the ψ torsion angle to near 180 °. Indeed, this can be seen in the contour diagrams for both NAG-ASN and MAN-TRP, although the region of low energy ψ angles extends further than initially anticipated. While this may initially seem like each conformation in this region is equally likely, this region encompasses conformations with different stabilities, which have been aggregated into a single contour level due to the dynamic range of the diagram. This was done purposefully to dissuade any notion of absolute stability, and instead should be seen as simply an estimation. For example, the NAG-ASN results suggest there is a general allowed region when conformations contain a ψ torsion angle near 180 ° with any φ angle, containing both the major and minor conformations observed in the Protein Data Bank (see Section 5.3.2.1). When attempting to model a new NAG-ASN linkage, conformations outside of this region should not be modelled, since they would likely represent an unstable high-energy conformation. An example of a conformation inside this region and one outside this region is shown in Figure 5.7. Despite the high-energy conformation exhibiting no clear atomic overlaps, the proximity of the *N*-acetyl modification to the asparagine likely increases steric clashes, resulting in a more unstable conformation. Without planar donor groups, both serine and threonine appear to exhibit more varied allowable conformation regions. Interest-

ingly, the allowed region for MAN-SER is much larger than that for MAN-THR, most likely due causes by interactions with the additional methyl group on the side chain of threonine when compared to that of serine.

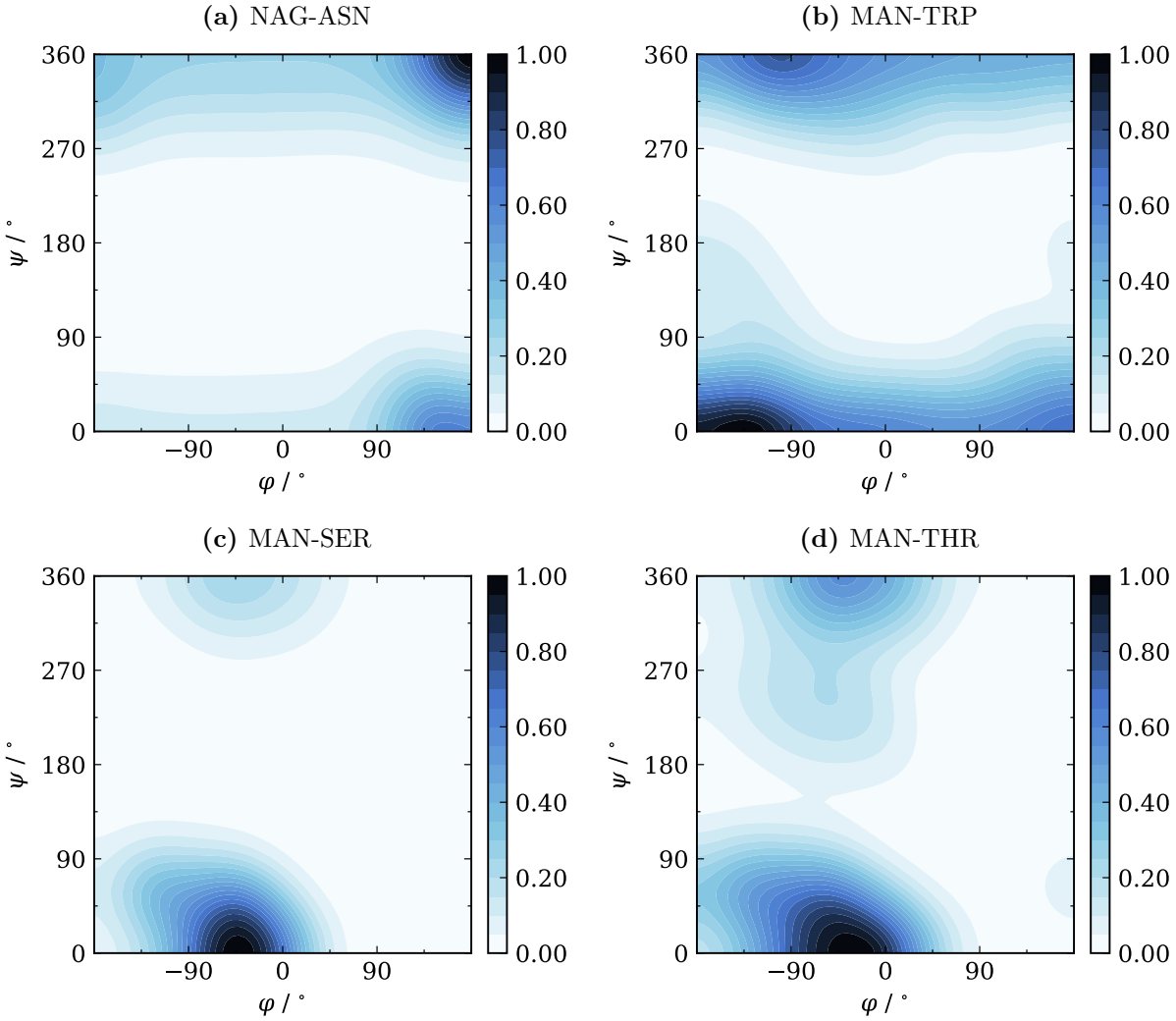
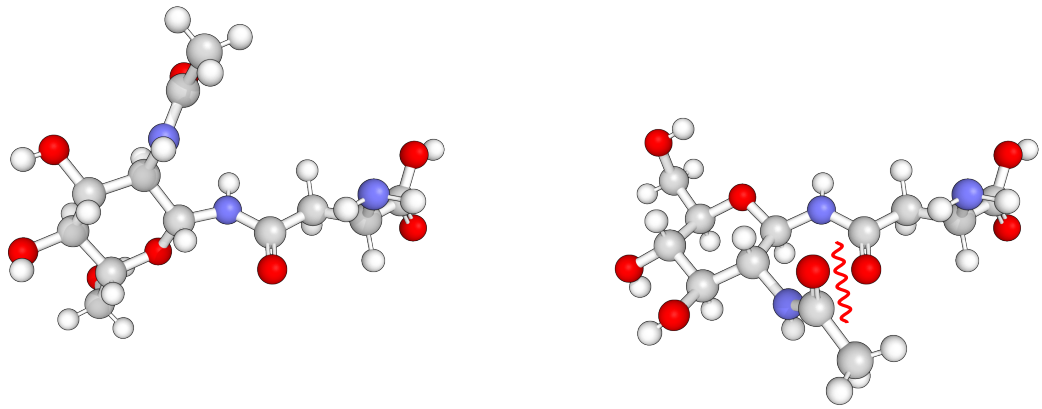


Figure 5.6: Normalised energy contour diagrams for 4 protein-sugar linkages, with colour representing the normalised energy. The self-consistent field energy values were calculated for each combination of ψ,φ torsion angles using *Gaussian16*.[296] Energy values were normalised within each linkage, which can be used to identify regions of high-energy conformations for a given linkage.



(a) Low-energy conformation **(b)** High-energy conformation

Figure 5.7: High and low energy example conformations for the NAG-ASN linkage. The low-energy conformation has non-bonding atoms sufficiently far apart, thereby limiting unfavourable steric interactions. The high-energy conformation exhibits unfavourable steric interactions between the *N*-acetyl group of the GlcNAc sugar and the carbonyl of the asparagine side chain, indicated by the red line.

5.4.2.2 Glycosidic Linkages Between Pyranosides

Normalised energetics were calculated across conformational space for the 10 sugar-sugar linkages identified through the survey of the Protein Data Bank, shown in Figure 5.8. In an identical way to the normalised energetic contour diagrams in Section 5.4.2.1, regions with high normalised energy values indicate unstable conformations of a given linkage, which can be used to inform downstream model-building methods.

As was seen with validated linkage from the Protein Data Bank, when the attached sugar is in the β-anomer, the allowed regions generally exist with negative φ values only. This can be seen with both the NAG-1,4-NAG and BMA-1,4-NAG linkages, although the trend is slightly dependent on the ψ value of the conformation. Similarly, when the attached sugar is in the α-anomer, the allowed regions were theorised to exist only in conformations with positive φ values. While allowed regions do exist with positive φ values, these energetic results suggest that negative φ conformations are favourable when accompanied with specific ψ values.

5.4.2.3 Comparison to Validated Linkage Data

If this abundance of conformational data is to be used in downstream structural inference, then it must tolerate uncommon conformations that are stabilised by external interactions, as was outlined in Section 5.3.2.2.1. The example shown describes a MAN-1,2-MAN linkage with linkage torsion angles which deviate far from the mean value observed across the validated linkage dataset. The normalised energy for this specific set of torsion angles is ≈ 0, indicating that these calculations are sensitive enough not to discount realistic but uncommon conformations. Extending this idea, since all linkage conformations were

validated with *Privateer*, all conformations should exist in the lowest energy region identified for a given linkage. On average, 99 % of validated protein-sugar linkages and 98 % of validated sugar-sugar linkages fall within the lowest energy region. While this is a large percentage, some linkages are in non-favourable conformations, whilst both the donor and acceptor groups have been deemed correct by *Privateer*.

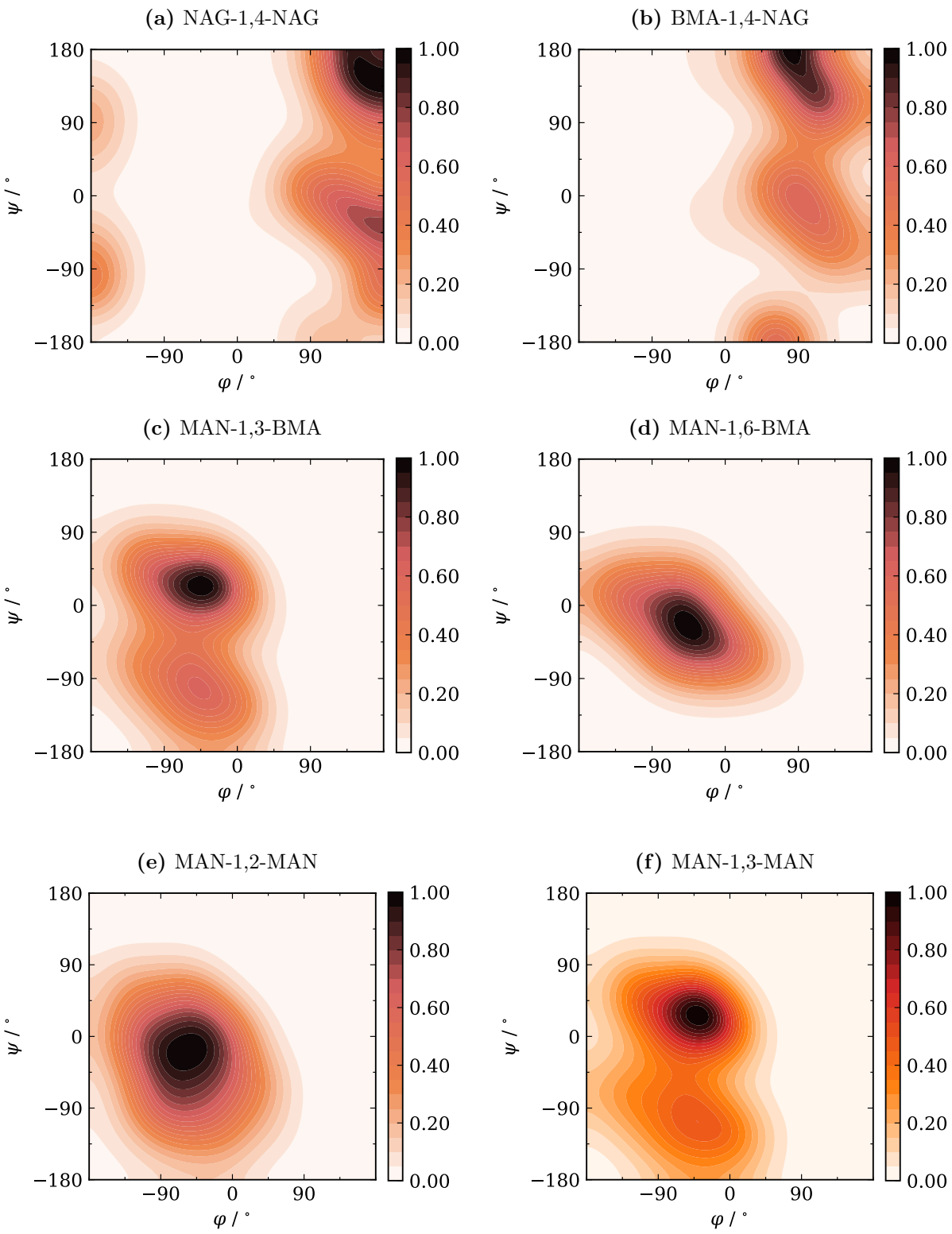


Figure 5.8: Normalised energy contour diagrams for 10 sugar-sugar linkages, with colour representing the normalised energy. The self-consistent field energy values were calculated for each combination of ψ,φ torsion angles using *Gaussian16*.[296] Energy values were normalised within each linkage, which can be used to identify regions of high-energy conformations for a given linkage.

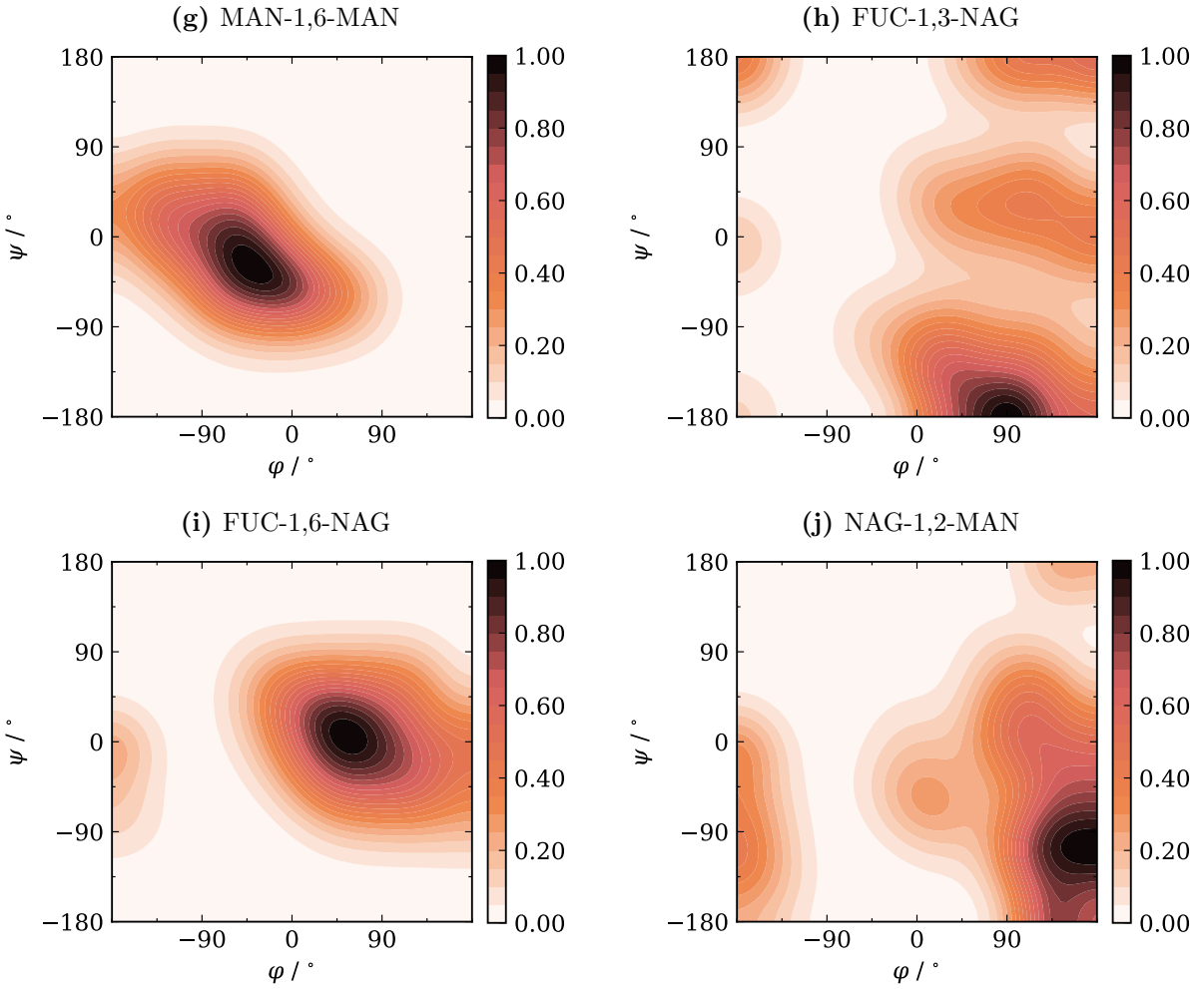


Figure 5.8 (continued): Normalised energy contour diagrams for 10 sugar-sugar linkages, with colour representing the normalised energy. The self-consistent field energy values were calculated for each combination of ψ,φ torsion angles using *Gaussian16*.[296] Energy values were normalised within each linkage, which can be used to identify regions of high-energy conformations for a given linkage.

5.4.3 Conclusions

Generally favourable and unfavourable conformational regions have been identified by analysing the energetics of the most abundant validated glycosidic linkages in the Protein Data Bank. These conformational regions may be used as bounds for both validation and model building,

The primary objective of this study was to establish the foundation for an efficient, automated model-building software. However, some linkages exhibit unfavourable conformations despite being deemed correct by the validation software *Privateer*. To mitigate this problem, this dataset of energetic information could be used to inform linkage conformation validation, which may potentially reduce the chance of future modelling inconsistencies.

5.5 Applications to Structural Validation

Validation software is routinely used during the structure solution process to ensure that atomic models are geometrically and experimentally sound. The software package *Privateer* has become the *de facto* tool for validating structurally determined carbohydrates. It is common to rely on the output of validation software to guide structure solution, so all validation tools must perform accurate validation. Analysis of the energetics of glycosidic linkages highlighted several linkages which passed all of the validation checks in *Privateer*, but appear to be in geometrically unfavourable conformations, described in Section 5.4.2.3. To ensure the validation methods are robust, it may be helpful to highlight glycosidic linkages that are conformationally distant from the most expected conformation across the Protein Data Bank or distant from the known energetically favourable regions.

5.5.1 Validation Using Structural Data

When scrutinising a linkage during glycan validation, the linkage conformation can be effectively searched for in the validated linkage data for that specific linkage. The intuition behind this is that if the linkage under scrutiny has been seen and already validated, then it is likely to be correct. Whilst theoretically simple, this method requires some care in implementations since the chance that any linkage has the exact same conformation as any other is low, given the relatively small validated sample size. Instead, the validated data can be grouped into bins to ensure that minute geometric variability does not impact the validation method. For each validated linkage, the ψ, φ torsional space was split into two-dimensional bins with a 2 ° bin spacing, in a similar method to the protein validation software *Tortoize*.[297] This is defined formally in Equations 5.4 and 5.5 for a sugar-sugar glycosidic linkage.

$$i_{\psi} = \left\lfloor \frac{\psi + 180}{2} \right\rfloor \quad \text{where } \psi \in [-180, 180] \tag{5.4}$$

$$i_{\varphi} = \left\lfloor \frac{\varphi + 180}{2} \right\rfloor \quad \text{where } \varphi \in [-180, 180] \tag{5.5}$$

Counting the number of validated linkages found in the same bin as the linkage under scrutiny provides an idea of how likely the new linkage is, but without a comparison to the rest of the conformational distribution, this value has little meaning. Instead, a Z-score can be calculated, which weights a given bin in relation to the other bins in the distribution, allowing for easier identification of unusual conformations. The Z-score,

$z_{\psi,\varphi}$, is calculated as shown in Equations 5.6 - 5.8.[298]

$$\langle c \rangle = \frac{1}{N^2} \sum_{i_\psi=0}^{N-1} \sum_{i_\varphi=0}^{N-1} c(i_\psi, i_\varphi) \tag{5.6}$$

$$\sigma(c) = \sqrt{\frac{1}{N^2} \sum_{i_\psi=0}^{N-1} \sum_{i_\varphi=0}^{N-1} (c(i_\psi, i_\varphi) - \langle c \rangle)^2} \tag{5.7}$$

$$z_{\psi,\varphi} = \frac{c(i_\psi, i_\varphi) - \langle c \rangle}{\sigma(c)} \tag{5.8}$$

A positive Z-score suggests that the linkage conformation is well represented in the validated dataset, whereas a negative Z-score suggests that it is uncommon. An additional benefit to the Z-score is that values are normalised to the specific linkage of concern, allowing Z-scores to be compared across linkage types. This Z-score calculation was added to *Privateer* and output during routine glycan validation in the *CCP4* software suite. Z-scores of less than -1 are flagged for inspection since they are unlikely to have been seen in the validated dataset, however as suggested earlier, uncommon conformations may be perfectly valid but unsolved in the Protein Data Bank. To supplement this score and provide a better insight for linkage validation, the conformation may be compared to energetically favourable conformations.

5.5.2 Validation Using Energetic Data

Energetic calculations for each validated linkage outlined in Section 5.4 suggest regions of favourable linkage conformations and regions of conformations which are unlikely to exist naturally. When validating a linkage, the linkage conformation can be simply compared to the pre-calculated normalised energy distribution to determine whether a linkage is valid or not. This approach is methodologically similar to validation using structural data, but the calculation of a Z-score is not required since the energy values are already normalised. For a given linkage conformation, an appropriate 6 ° bin can be calculated, in a similar way to Equations 5.4 and 5.5. If this bin is contained within the lowest energy region, then the linkage can be deemed acceptable, provided that it also passes all other forms of validation calculated by *Privateer*.

Chapter 6

Automated Model Building of Carbohydrates

6.1 Identification of Carbohydrates in Density

Modelling carbohydrates into electron density or Coulomb potential maps is a common, yet non-trivial, step in the structure-solution process. Regions of experimental density which are explainable by carbohydrates are often challenging to interpret, particularly when the glycan chains are flexible or heterogeneous. The first step in carbohydrate modelling is to determine whether the target biological molecule contains carbohydrates, which can be particularly difficult when relying solely on experimental density and is often achieved by combining information from other bioinformatic sources.[299]

Sequence information is key for identifying potential glycosylation sites, since both *N*-glycosylation and *C*-glycosylation follow specific consensus sequences or motifs. The identification of potential sites can be followed by inspection of the experimental density in the surrounding region to determine whether any carbohydrates are modellable. Indeed, similar approaches have been incorporated into functional interactive and semi-automatic carbohydrate-building software packages that focus exclusively on modelling *N*-glycosylation,[51,300] or modelling *C*-glycosylation.[301] Few methods focus on the other common type of glycosylation, *O*-glycosylation, likely because no natural consensus sequence has been identified for the modification, and *O*-glycans are commonly found in intrinsically disordered regions, which are not well studied by structural techniques.[302] If *O*-glycans can be resolved, modelling almost certainly occurs through observation of the experimental density or comparisons with homologous structures, in combination with other biophysical techniques such as mass spectrometry.[303]

In the absence of sequence information, some methods attempt to interpret the experimental density directly, but have found limited success.[301,304] The ability to model carbo-

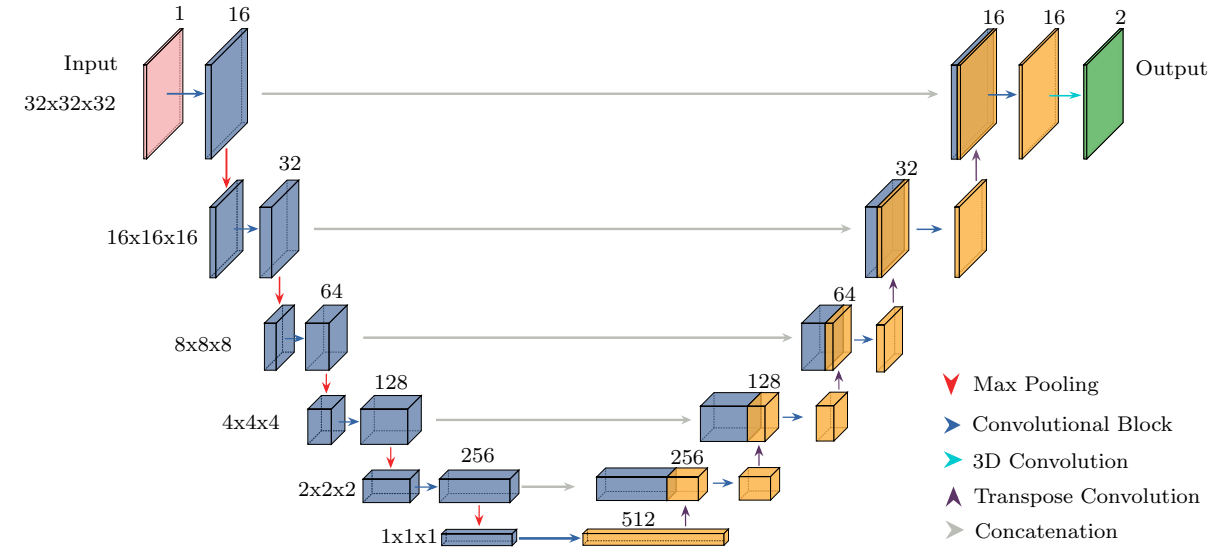
hydrates without requiring sequence information or manual interpretation is appealing, as such methods could be automated for any carbohydrate, including *O*-glycosylation. A clear parallel can be drawn between the generally accepted difficulty of carbohydrate experimental density interpretation and the difficulties associated with interpreting partially phased nucleic acid electron density. It is natural, therefore, to explore whether the successful method developed for nucleic acid identification in Part I can be applied to identify carbohydrates. If this were possible, it could enable the development of an unprecedented, fully automated method for carbohydrate model building.

6.1.1 Application of Established Convolutional Neural Networks for Carbohydrate Identification

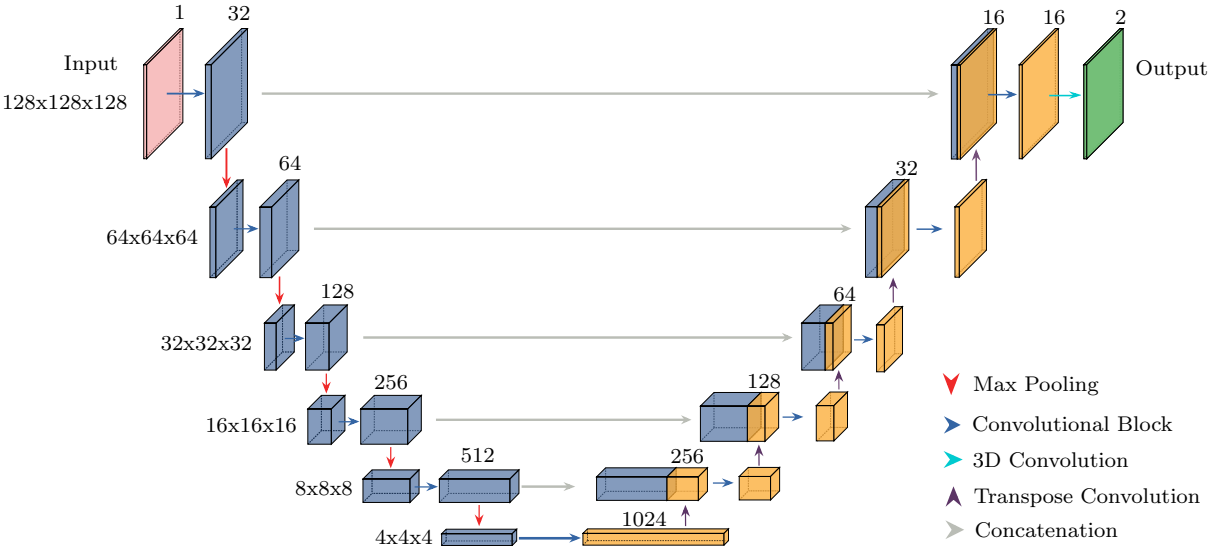
As the task of identifying poorly resolved carbohydrates is similar to that of poorly resolved nucleic acids, a method similar to that described in Part I was investigated to assess whether such models could be applied to carbohydrate systems. Unlike nucleotides, carbohydrates are not easily separable into constituent groups, so an initial experiment was conducted to predict regions of experimental density corresponding to any pyranose carbohydrate, rather than the biochemically segmented output designed for nucleic acids.

6.1.1.1 Neural Network Architecture

Two new binary segmentation models were created, which mirror the baseline nucleic acid segmentation and optimised nucleic acid segmentation architectures from Chapter 3. The previously developed baseline segmentation and optimised segmentation architectures were designed to output four channels per point, representing no nucleic acid, a phosphate position, a sugar position and a base position. Such a design is superfluous for simple binary segmentation of carbohydrates, so both model architectures were modified to reduce the number of output channels from 4 to 2, representing no carbohydrate and a carbohydrate position. The baseline binary segmentation model, shown in Figure 6.1a, operates with an input spatial dimension, $\boldsymbol{X}$, of length 32 ($\boldsymbol{X} \in \mathbb{R}^{32,32,32,1}$), whereas the optimised binary segmentation model, shown in Figure 6.1b, operates with an input spatial dimension of length 128 ($\boldsymbol{X} \in \mathbb{R}^{128,128,128,1}$). Despite the optimised nucleic acid segmentation model outperforming the baseline segmentation model across all metrics for the task of nucleic acid identification (see Section 3.4), both models were used in this initial experiment, as the optimality of the nucleic acid model may not transfer to carbohydrates.



(a) Baseline binary segmentation three-dimensional U-Net



(b) Optimised binary segmentation three-dimensional U-Net

Figure 6.1: A - Schematic view of the baseline binary segmentation three-dimensional U-Net architecture. The encoder-decoder network first downsamples the data of shape $(32, 32, 32, 1)$ to a vector form of shape $(1, 1, 1, 512)$. The vector is then upsampled back to an output of shape $(32, 32, 32, 2)$, where the two output channels represent the probability of the grid point being no carbohydrate and the probability of the grid point being a carbohydrate. B - Schematic view of the optimised binary segmentation three-dimensional U-Net architecture. The encoder-decoder network first downsamples the data of shape $(128, 128, 128, 1)$ to a vector form of shape $(4, 4, 4, 1024)$. The vector is then upsampled back to an output of shape $(128, 128, 128, 2)$, where the two output channels represent the probability of the grid point being no carbohydrate and the probability of the grid point being a carbohydrate.

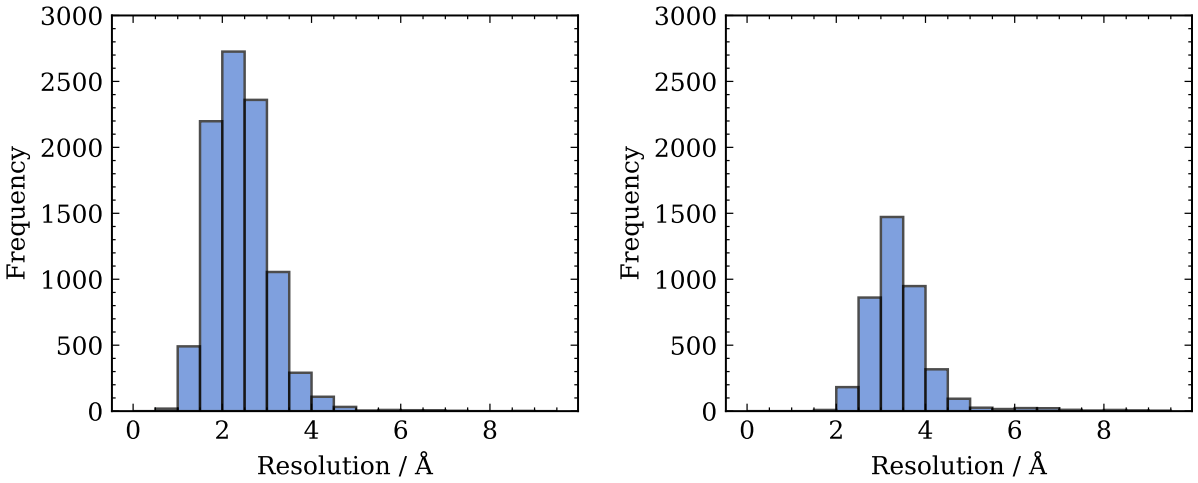
6.1.1.2 Training

6.1.1.2.1 Dataset Creation

The dataset used to train the two binary segmentation models originated from structures deposited in the Protein Data Bank[305] as of August 2025. All structures containing *N*-glycosylation, *O*-glycosylation, and *C*-glycosylation were obtained from the Protein Data Bank, and filtered to contain structures solved with either X-ray crystallography or cryo-EM. A maximum resolution cut-off of 5 Å was enforced for cryo-EM structures to protect the dataset from very low-resolution cases, where the glycosylation state should not be determined to avoid erroneous interpretation. This process yielded 9,530 X-ray crystallographic structures and 4,083 cryo-EM structures. The majority of structures in the dataset contained exclusively *N*-glycans, with 8,920 *N*-glycan structures from X-ray crystallography and 3,883 *N*-glycan structures from cryo-EM. There were comparatively few, exclusively *O*-glycosylated structures with only 174 *O*-glycan structures from X-ray crystallography and 18 *O*-glycan structures from cryo-EM. The remainder of the dataset consisted of 7 exclusively *C*-glycosylated structures from X-ray crystallography, 206 X-ray crystallographic structures that contained multiple types of glycosylation, and 50 multiply glycosylated cryo-EM structures.

For all X-ray crystallographic structures, electron density maps were calculated using the deposited structure factor intensities or amplitudes and the deposited model using *REFMAC5*. To supplement the dataset with more realistic electron density examples, electron density maps were also calculated using the deposited structure factor intensities or amplitudes, and a model with all glycans removed, with B-factors of the remaining atoms set to the average of the entire structure. This deglycosylated electron density emulation aims to replicate the commonly observed state of electron density after completing protein model building. For the cryo-EM structures, full maps were obtained from the Electron Microscopy Data Bank and trimmed using *Servalcat* to remove the large unnecessary regions surrounding the molecule of interest.

This carbohydrate dataset contains X-ray crystallographic structures with an average resolution of 2.39 Å, with a resolution range of 0.91 Å to 8.69 Å. Cryo-EM structures in the dataset exhibited an average reconstruction resolution of 3.32 Å, with a reconstruction resolution range of 1.70 Å to 4.99 Å. The distribution of resolutions for both structure determination methods is shown in Figure 6.2.



(a) X-ray diffraction **(b)** Cryo-EM

Figure 6.2: Resolutions of structures used during training of the carbohydrate deep learning model dataset across both X-ray diffraction and cryo-EM structures. The average resolution of the structures solved with X-ray diffraction is 2.39 Å, and the average reconstruction resolution of the cryo-EM structures is 3.32 Å.

6.1.1.2.2 Dataset Preprocessing

To ensure uniformity between samples, each density map in the dataset is interpolated and orthogonalised onto a regular orthogonal grid with a grid spacing of 0.70 Å, in an identical way to the dataset used to train the nucleic acid segmentation models in Part I. A target map can then be generated, which has the same spatial dimensions as a given density map example in the dataset, but where grid points within 1.5 Å of any atom in any carbohydrate pyranose group are set to the value 1, with values of 0 elsewhere, as shown in Equation 2.4. 80 % of the dataset can be randomly assigned to the training set, with the remaining 20 % allocated to the test set.

6.1.1.2.3 Training Scheme

Training a convolutional neural network to identify carbohydrate groups is a principal example of a problem with many imbalanced classes. The majority of grid points in the target map will have values of 0, and if samples were drawn randomly from such a target map, the model may often observe only negative data points during training, leading to reduced model performance, as observed in training models for nucleic acid segmentation. One option to combat this issue is to randomly draw samples and ensure that each sample encompasses a specific number of target points. If the sample does not encompass enough target points, another is drawn, as was described in Section 2.3.2.3. While this approach may work, drawing incompatible samples only serves to slow down training and reduce computational efficiency. A better option is to draw samples pseudo-randomly originating

around a set of known positive target points, and subsequently apply a random rotation and small random translation. This approach reduces the chance that a solely negative sample will be drawn and increases the speed at which data can be processed for training. The pseudo-random sampling method is effectively centred around a uniformly sampled atomic position, $\boldsymbol{c}$, drawn from the set of atomic positions for all pyranose atoms in the current structure, $\mathcal{A}_{\text{pyr}}$, shown in Equation 6.1. To alleviate the bias associated with drawing samples with a positive target at the centre of the sample, a random offset, $\boldsymbol{\epsilon}$, is applied after being drawn from the normal distribution, shown in Equation 6.2. The standard deviation, σ, of the normal distribution is set to a quarter of the sample spatial dimension length, N, as a compromise between removing central bias and keeping some target points within the sampled cube.

$$\mathcal{A}_{\text{pyr}} = \{\boldsymbol{a_0}, \boldsymbol{a_1}, ..., \boldsymbol{a_M}\}, \quad \boldsymbol{a_i} \in \mathbb{R}^3 \tag{6.1}$$
$$\boldsymbol{c} \sim \mathcal{U}(\mathcal{A}_{\text{pyr}})$$

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \end{bmatrix}, \quad \epsilon_x \sim \mathcal{N}(0, \sigma^2), \epsilon_y \sim \mathcal{N}(0, \sigma^2), \epsilon_z \sim \mathcal{N}(0, \sigma^2) \tag{6.2}$$
$$\boldsymbol{c'} = \boldsymbol{c} + \boldsymbol{\epsilon}$$

Following the selection of a pseudo-random centre point, a random rotation and scale can be applied identically to the sampling procedure previously described for nucleic acids, shown in Equation 6.3.

$$\mathbf{R} = \text{Rotation}(\mathbf{u}, \theta), \quad \mathbf{u} \sim \mathcal{U}(\mathbb{S}^2), \quad \theta \sim \mathcal{U}(0, 2\pi) \tag{6.3}$$
$$\mathbf{S} = s\mathbf{I}_3 \quad \text{where s} = 0.7 \text{ and } \mathbf{I} \text{ is the identity matrix}$$

The sample can then be interpolated using this random rotation and pseudo-random centre point as shown in Equation 6.4.

$$\mathcal{V}(i, j, k) = \mathcal{I}(\mathcal{G}, \mathbf{RS}\boldsymbol{x}_{ijk} + \boldsymbol{c'}), \quad \forall i, j, k \in [0, N] \tag{6.4}$$

where:

- $\mathcal{V}$ is the sampled vector
- $\mathcal{I}$ is the trilinear interpolation function
- $\mathcal{G}$ is the interpolated orthogonal grid
- $\boldsymbol{x}_{ijk}$ is the index in the reference frame of the sample

Both the baseline binary segmentation and optimised binary segmentation models were trained using this sampling procedure for the first 20 epochs. Following this pseudo-random sampling, the models were trained for an additional 20 epochs using random sampling, as described in Equation 2.5. Once a model has been trained, inference can be performed using the same scheme outlined in Section 2.3.3.

6.1.1.3 Test Set Creation

A subset of structures was selected from the dataset to test the performance of both binary segmentation models in identifying carbohydrates based on density. To ensure the performance of the models is tested fairly, structures that contained any glycans deemed geometrically invalid by *Privateer* were excluded from consideration in the test set. No fit-to-density metrics were enforced to test the ability of a deep learning model to identify weaker experimental density.

The geometrically validated dataset was first partitioned by experimental method to ensure the test set contained examples from both X-ray crystallography and cryo-EM. The dataset was then divided by glycan type to include examples containing both *N*-glycans and *O*-glycans. *C*-glycosylation was excluded from this test set, due to the minute number of *C*-glycans examples in the dataset. Considering the dependence on the interpretability of glycan density based on resolution, a 2.5 % sample of each integral resolution bin was taken for each of the four partitions of the dataset. This small sample was chosen to ensure that the test set remained large enough to be indicative of model performance, yet not so large as to be computationally inefficient.

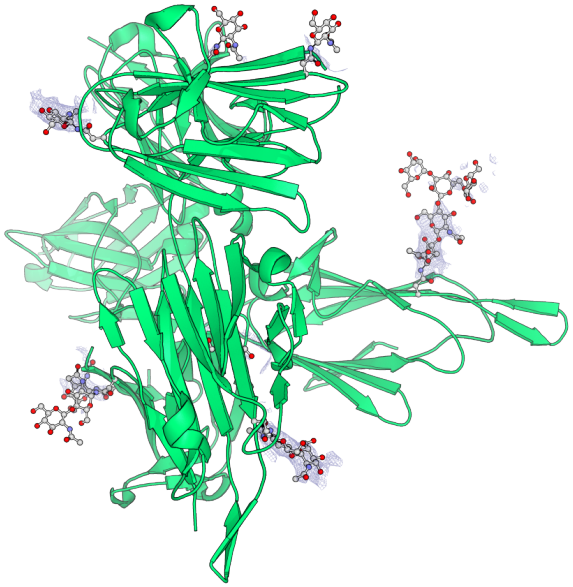
This process yielded 234 examples from X-ray crystallography, 225 of which were exclusively *N*-glycan containing structures, 8 of which were solely *O*-glycan containing structures and 1 of which included a mixture of both *N*-glycans and *O*-glycans. The remainder of the test set consisted of 109 *N*-glycan-containing structures determined with cryo-EM. No *O*-glycan structures from cryo-EM were included due to the limited number included in the overall dataset. Since each partition was sampled uniformly across resolution bins, the distribution of the test set closely mirrors the resolution distribution of the entire dataset, shown in Figure 6.2.

For the X-ray crystallographic structures, structure factor intensities or amplitudes were obtained from the Protein Data Bank, along with the complete deposited model. Carbohydrates and waters were removed from the deposited model before the B-factors of every atom were set to the average B-factor of the remaining structure. After 10 cycles of refinement using *Servalcat*, an electron density map can be obtained which aims to resemble realistic electron density used for modelling carbohydrates. For cryo-EM structures, full maps were downloaded from the Electron Microscopy Data Bank and trimmed using *Servalcat*. All structures from this test set were removed from any further deep learning model training to prevent bias.

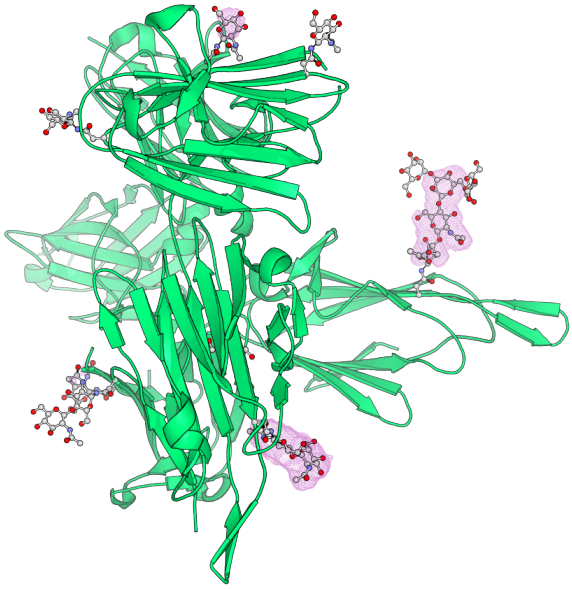
6.1.1.4 Results and Discussion

The results of inference using both the baseline binary segmentation and the optimised binary segmentation models are shown in Figure 6.3 for the *Plasmodium falciparum* Pfs230 protein (PDB code: 9E7N[306]) resolved to 2.48 Å using X-ray crystallography. Inference was performed on an input electron density map calculated using deposited structure factor amplitudes and a model with no carbohydrates and solvent, as outlined in Section 6.1.1.3. This model was not part of the training or testing set of the deep learning models, and no structures related by homology to the Pfs230 protein were included in the dataset. Any identification of carbohydrates by a deep learning model using this structure is most likely a factor of generalisation capability and not regurgitation.

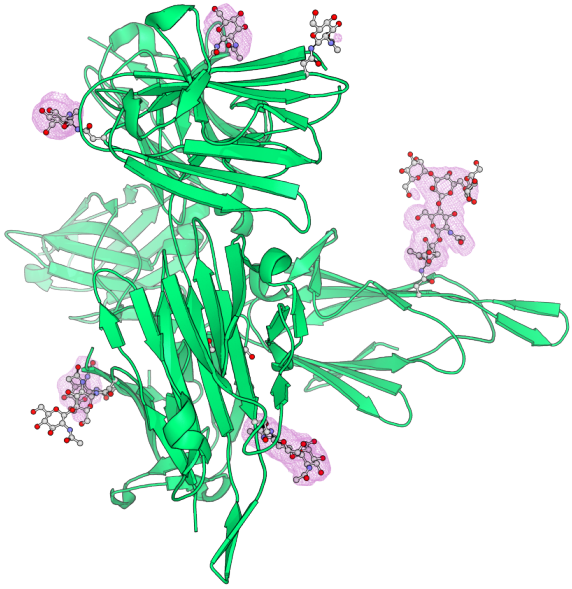
Without the contribution of the atoms in the carbohydrates to the estimation of the phases of each experimental reflection, the electron density in the region of the carbohydrates in the Pfs230 protein is generally poorer than that of the protein region. Some carbohydrates modelled in the deposition exhibit little visible electron density at a 1.5 σ contour level, which may pose challenges for interpretation. The output of the baseline binary segmentation model, shown in Figure 6.3b, exhibits an average atom inclusion of 45.1 %, with at least one positive prediction in the area of five of the seven glycan chains. Comparatively, the optimised binary segmentation model, shown in Figure 6.3c, produces an output with at least one positive prediction for all glycans, with an average atom inclusion of 71.1 %. The ability of both binary segmentation models to identify challenging carbohydrate electron density is promising, suggesting that the method has been successful in this example. It is pertinent to recall the original purpose of these deep learning models, which was to identify carbohydrates in experimental density without using sequence information. For the intended goal, the output predictions for the optimised model are much more useful and complete than those of the baseline model, although it should be noted that some carbohydrates remain unpredicted in the outputs from both models. The inability of either model to locate certain carbohydrates could indicate a lack of performance, but is most likely attributed to the challenging characteristics of the electron density in those regions. To ensure that this relatively good performance of the models is observable across a range of structures, determination methods and glycosylation types, inference was run for both models across the carbohydrate test set described in Section 6.1.1.3.



(a) Input density



(b) Baseline binary segmentation model



(c) Optimised binary segmentation model

Figure 6.3: Output of the baseline and optimised binary segmentation models corresponding to predicted carbohydrate locations for a Pfs230 protein (PDB code: 9E7N[306]) resolved to 2.48 Å using X-ray crystallography. To generate the input density, carbohydrates and solvent were removed from the deposited model, and refined using the deposited structure factor amplitudes with *Servalcat*. The resultant electron density map, shown at 1.5 σ, was provided to both the baseline and optimised carbohydrate segmentation models using the inference procedure originally implemented for nucleic acids.

The results of inference over the 343 structures in the carbohydrate test set are shown in Table 6.1, with statistics shown in Table 6.2. As was demonstrated when using these model architectures for nucleic acid identification, the optimised segmentation model outperforms the baseline segmentation model in all metrics for both crystallographic and cryo-EM examples.

The baseline binary segmentation model shows mediocre performance on crystallographic data and very poor performance on cryo-EM data. Although the performance is inadequate, the model produces some positive predictions, suggesting it can understand carbohydrate density to some extent and is not a product of failed convergence or other training artefact. In comparison, the optimised binary segmentation model performs well across the crystallographic examples in the test set, predicting a high percentage of atomic positions despite the weaker density associated with carbohydrates. The performance of the optimised model with cryo-EM data is not as strong, with relatively few atomic positions correctly identified.

The statistically significant increase in performance from the baseline model to the optimised model suggests that carbohydrate identification is likely aided by both the increased spatial size of the input and the additional parameters within the optimised model. This is slightly surprising, since carbohydrates often exist in a more limited volume of density when compared to the nucleic acid target the model architecture was optimised to identify. One explanation for this behaviour may be that the additional spatial context provided by the larger input allows the model to better understand which areas are likely to be solvent and which are likely to contain macromolecules. Since carbohydrates most commonly protrude into the solvent surrounding a macromolecule, understanding the difference between these two areas of density may be helpful for performance. The relatively smaller input spatial dimension of the baseline model is likely to limit what environmental context can be encoded. While this hypothesis may provide some rationalisation, it

Table 6.1: Atom inclusion, precision, recall and F1 score metric results for carbohydrate identification with the baseline binary segmentation model and the optimised binary segmentation model. Metrics were calculated as averages across an X-ray diffraction test set and a cryo-EM Coulomb potential map test set. Uncertainty here represents the standard deviation across the samples.

(a) X-ray diffraction test set

Model	Atom Inclusion / %	Precision / %	Recall / %	F1 Score / %
Baseline binary segmentation	53.2 ± 30.8	69.6 ± 12.2	68.8 ± 12.6	67.2 ± 10.5
Optimised binary segmentation	71.4 ± 30.5	75.8 ± 10.0	79.5 ± 14.5	75.5 ± 10.5

(b) Cryo-EM test set

Model	Atom Inclusion / %	Precision / %	Recall / %	F1 Score / %
Baseline binary segmentation	12.8 ± 23.0	58.0 ± 12.4	54.0 ± 8.3	52.8 ± 5.0
Optimised binary segmentation	42.3 ± 31.2	68.2 ± 12.0	64.2 ± 12.1	63.4 ± 9.7

Table 6.2: Statistics calculated by comparing the baseline binary segmentation model against the optimised binary segmentation model for carbohydrate identification. Pairwise significance estimations were completed with a Wilcoxon signed-rank test. Significance levels are represented by stars where $P \geq 0.05$ = n.s., $0.01 \leq P < 0.05 = *$, $0.001 \leq P < 0.01 = **$, $P < 0.001 = ***$. Delta values are the difference between the average performance of both models for a given output, expressed in percentage points (pp).

(a) X-ray diffraction test set

Model		**Atom Inclusion**			**F1 Score**		
Ref.	Comp.	P-value	Sig.	Delta / pp	P-value	Sig.	Delta / pp
Baseline binary segmentation	**Optimised binary segmentation**	0.00	***	18.3	0.00	***	8.2

(b) Cryo-EM test set

Model		**Atom Inclusion**			**F1 Score**		
Ref.	Comp.	P-value	Sig.	Delta / pp	P-value	Sig.	Delta / pp
Baseline binary segmentation	**Optimised binary segmentation**	0.00	***	29.5	0.00	***	10.6

is challenging to extract a definite conclusion from only two data points. Further experimentation on the spatial size may provide further proof, but was not pursued to preserve the limited resources available for other experiments.

6.1.1.5 Conclusions

Both binary segmentation models were able to detect some carbohydrate density, even though the model architectures were initially designed for a nucleic acid system. This achievement is remarkable and suggests that this architecture can understand electron density and cryo-EM Coulomb potential maps across various systems. Given the performative similarity between nucleic acids and carbohydrates, it is likely that the methodological changes developed to improve nucleic acid identification performance will also be applicable to this carbohydrate system. Perhaps the most significant improvement in nucleic acid identification precision came from moving from a binary to a multiclass segmentation model, which likely allowed the model to better localise features. Since glycosylation generally occurs through an amino acid donor, if a model were able to locate both the protein donor and carbohydrate group, it may be possible to realise additional performance, as was seen for nucleic acids.

6.1.2 Optimisation of Convolutional Neural Networks for Carbohydrate Identification

Deep learning models tasked with identifying atomic features in crystallographic and cryo-EM density have been shown to exhibit higher precision when the number of output classes is increased to include other spatially similar targets (see Chapter 3). Intuitively, if a deep learning model were encouraged to learn the distinction between two neighbouring and distinct classification targets, it must be able to predict either class if the other is proximal. This approach allows the model to become more prudent in identification and, in theory, decreases the likelihood of false positives. To ensure performance improvements, the number of parameters in the model must be large enough to accommodate this additional task, otherwise other performance metrics are likely to falter.

For the task of identifying glycosylation, the choice of the proximal target group is straightforward. Since glycosylation most commonly occurs through an asparagine, threonine, serine, or tryptophan, the donor amino acid, which attaches the glycan to the protein, is certain to be near the carbohydrate, barring any significant modelling errors in a given structure. To investigate this, the optimised binary segmentation model architecture was altered to produce three output channels: one for no carbohydrate or protein, one for a carbohydrate position, and one for a protein position, respectively, shown in Figure 6.4. This optimised multiclass segmentation model was trained using a similar training scheme to the baseline and optimised binary segmentation models, but with an additional protein target where any point within 1.5 Å of any donor amino acid atom was set to 1 with 0 elsewhere. The resultant two targets were one-hot encoded, with overlapping points assigned to the protein target. Samples were drawn using the same procedure as the binary segmentation models (see Section 6.1.1.2.3), and the model was trained using the multiclass sigmoid focal cross-entropy loss function. After 20 epochs of pseudo-random sampling, a further 20 epochs of random unrestricted sampling completed the training of the model. Following training, inference can be performed with the same method as outlined for nucleic acids, described in Section 3.1.4.

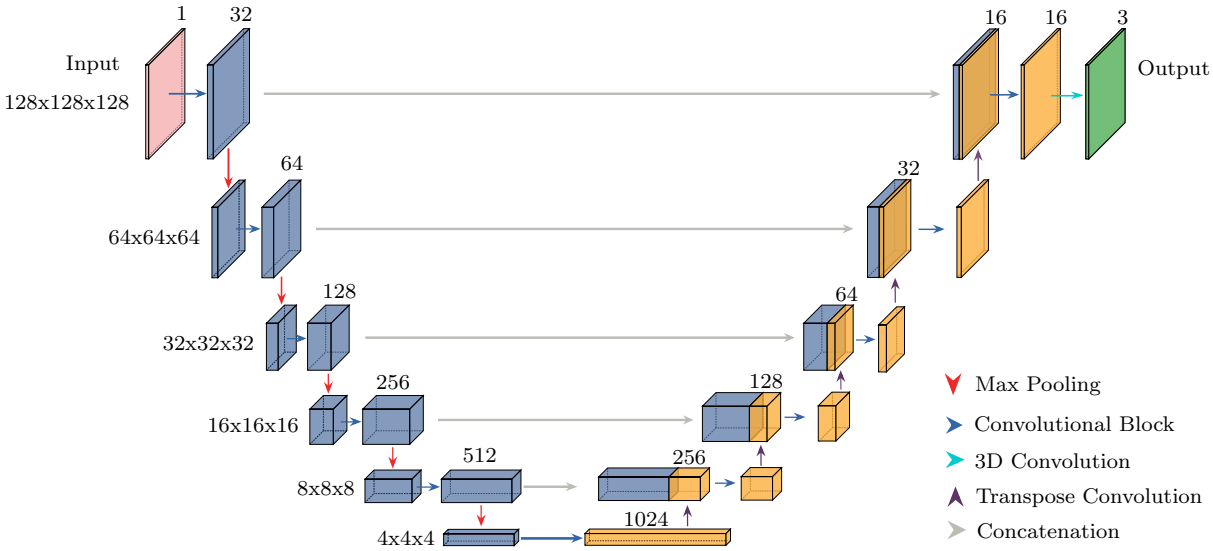


Figure 6.4: Schematic view of the optimised multiclass segmentation three-dimensional U-Net architecture. The encoder-decoder network first downsamples the data of shape $(128, 128, 128, 1)$ to a vector form of shape $(4, 4, 4, 1024)$. The vector is then upsampled back to an output of shape $(128, 128, 128, 3)$, where the three output channels represent the probability of the grid point being no carbohydrate, the probability of the grid point being a carbohydrate, and the probability of the grid point being a carbohydrate amino acid donor.

The inference results over the 343 examples in the carbohydrate test set for the optimised multiclass segmentation model are shown in Table 6.3, with statistics shown in Table 6.4. These results suggest that the performance of the optimised multiclass model improves when compared to the optimised binary model. The inclusion of the protein donor target, although a relatively trivial addition, produces the expected statistically significant increase in precision with a 3.3 pp increase in crystallographic examples and a 5.4 pp increase in cryo-EM examples. This higher precision suggests the model enforces some level of correlation between predicted protein and carbohydrate regions, but it is critical that this improvement does not come at the cost of recall. Across crystallographic examples, no statistically significant difference in recall is observed between the binary and multiclass models, and a slightly statistically significant increase in recall was observed across the cryo-EM examples in the test set. The additional precision and subtle improvements in recall contribute to statistically significant increases in the F1 score for both structural determination methods. The small but significant improvement in performance of the optimised multiclass segmentation model compared to the optimised binary segmentation model is an encouraging result, and further confirms that the methodological developments explored for nucleic acid identification are applicable to another system. The optimised multiclass model also provides a prediction for where the donor protein is likely to be without a significant increase in the number of model parameters, which is a helpful bonus.

Overall, this exploration into developing a convolutional neural network for carbohy-

drate identification has been successful. The finalised model performs well on realistic crystallographic examples and suboptimally but still acceptably on cryo-EM data. Given the similarity between the methods for nucleic acid and carbohydrate identification, and the similarly weak performance in cryo-EM data, it is likely that this characteristic is inherent to the method. Significant changes to the model architecture are likely needed to ensure better performance for cryo-EM data, which is likely to be an excellent avenue for future work. Regardless, the development of a model which can identify both proteins and carbohydrates involved in glycosylation is certain to be beneficial for a variety of downstream methodological applications.

Table 6.3: Atom inclusion, precision, recall and F1 score metric results for carbohydrate identification with the optimised binary segmentation model and the optimised multiclass segmentation model. Metrics were calculated as averages across an X-ray diffraction test set and a cryo-EM Coulomb potential map test set. Uncertainty here represents the standard deviation across the samples.

(a) X-ray diffraction test set

Model	Output	Atom Inc. / %	Precision / %	Recall / %	F1 Score / %
Optimised binary segmentation	Carbohydrate	71.4 ± 30.5	75.8 ± 10.0	79.5 ± 14.5	75.5 ± 10.5
Optimised multiclass segmentation	Carbohydrate	74.0 ± 29.2	79.1 ± 9.6	80.0 ± 13.6	77.7 ± 10.1
	Protein	75.5 ± 29.8	76.8 ± 9.6	81.5 ± 14.3	77.4 ± 10.5

(b) Cryo-EM test set

Model	Output	Atom Inc. / %	Precision / %	Recall / %	F1 Score / %
Optimised binary segmentation	Carbohydrate	42.3 ± 31.2	68.2 ± 12.0	64.2 ± 12.1	63.4 ± 9.7
Optimised multiclass segmentation	Carbohydrate	46.6 ± 28.9	74.1 ± 12.2	64.5 ± 10.3	66.5 ± 9.7
	Protein	43.8 ± 32.4	74.1 ± 13.2	63.7 ± 11.5	65.3 ± 10.6

Table 6.4: Statistics calculated by comparing the output of the optimised binary segmentation model against the carbohydrate output of the optimised multiclass segmentation model. Pairwise significance estimations were completed with a Wilcoxon signed-rank test. Significance levels are represented by stars where $P \geq 0.05 =$ n.s., $0.01 \leq P < 0.05 = *$, $0.001 \leq P < 0.01 = **$, $P < 0.001 = ***$. Delta values are the difference between the average performance of both models for a given output, expressed in percentage points (pp).

(a) X-ray diffraction test set

Atom Inclusion		**Precision**		**Recall**		**F1 Score**	
Sig.	Delta / pp	Sig.	Delta / pp	Sig.	Delta / pp	Sig.	Delta / pp
**	2.6	***	3.3	n.s.	0.4	***	2.2

(b) Cryo-EM test set

Atom Inclusion		**Precision**		**Recall**		**F1 Score**	
Sig.	Delta / pp	Sig.	Delta / pp	Sig.	Delta / pp	Sig.	Delta / pp
***	4.3	***	5.9	*	0.3	***	3.1

6.2 Automated Model Building of Carbohydrates Using Machine Learning Predictions

Tools for modelling carbohydrates into crystallographic or cryo-EM density maps often rely on initial manual input to guide the locations and type of glycosylation.[51,299] This is likely because identifying where glycosylation is expected to occur and which carbohydrate type to model is incredibly challenging. Guiding modelling with assistive interactive tools is helpful for this purpose, but perhaps a more optimal approach would be to automate the process.

Location of glycosylated sites can be trivial in a fully complete model where sequential information can be used to isolate potential *N*-glycosylated or *C*-glycosylated sites. However, in the typical instance of incomplete models during structure solution, such information may not always be available. Additionally, when sequential information is unavailable or unpredictable, as is the case for *O*-glycosylation, it is necessary to manually isolate potentially glycosylated sites using software such as *Coot*. This commonly occurs through visual inspection of the density around the modelled macromolecule, with a particular focus on regions unexplained by other molecules. Once potential sites have been identified, prior knowledge may be used to decide which carbohydrates should be modelled. This general strategy of glycosylation site identification, followed by modelling, often yields excellent results but requires a relatively high degree of skill and experience.

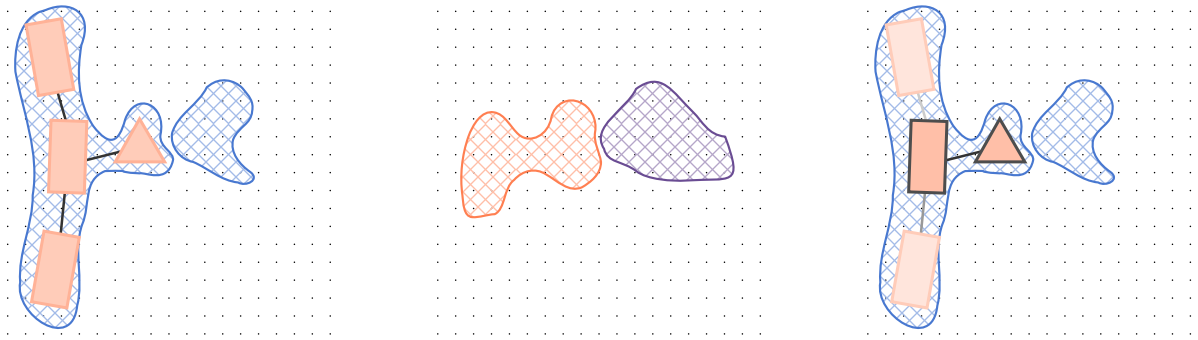
To create a more automated approach, the predicted outputs from the deep learning model corresponding to donor amino acids and carbohydrates could serve as a guide to identify unmodelled glycosylation sites. Once sites are identified, the prior-knowledge-based manual modelling of carbohydrates could be replaced by a method that relies on the vast array of data gathered during the validation of carbohydrates described in Chapter 5. In theory, this approach could provide a strong automated model-building method for carbohydrates, and an initial proof-of-concept was investigated.

6.2.1 Location of Glycosylated Sites

Assigning which amino acid residues are likely to be glycosylated is the first step required in an automated method for carbohydrate modelling. To ensure that most sites that can be glycosylated are investigated, it is pertinent to use both the predictions from the deep learning model and any available sequence information. Consensus sequons for *N*-glycosylation and *C*-glycosylation can be supplemented by assigning potential donor groups based on the location of positively predicted regions.

Since amino acid donors and carbohydrates should exist proximally, the regions of predicted density can be used to isolate modelled amino acid side chains that are likely to host glycosylation, which can be denoted by $\mathcal{S}$. All amino acid side chains with potential donor atoms, $\mathcal{A}_{\text{donor}}$, that are positioned inside positive protein donor predicted density, and are within 2 Å of a positive carbohydrate predicted position, may be classified as a potential site. This can be formally defined as shown in Equation 6.5 and schematically shown in Figure 6.5. This 2 Å distance requirement was chosen to approximate a glycosidic bond length with a small tolerance for perturbations in atomic positions.

$$\mathcal{S} = \{\boldsymbol{a_i} \in \mathcal{A}_{\text{donor}} \mid \rho_{\text{protein}}(\boldsymbol{a_i}) > 0 \wedge \exists \boldsymbol{g_j} \in \rho_{\text{carb}} : \|\boldsymbol{a_i} - \boldsymbol{g_j}\| < 2 \wedge \rho_{\text{carb}}(\boldsymbol{g_j}) > 0\} \quad (6.5)$$



Input density Predicted density Predicted site

Figure 6.5: Schematic of predicted site location using predicted density from a deep learning model for carbohydrate identification. If an amino acid side chain donor group is located within positive protein density (centre panel - left, orange) and is within 2 Å of positive carbohydrate density (centre panel - right, purple), it may be classified as a potential glycosylated site.

6.2.2 Modelling Carbohydrate Chains

After identifying a potential glycosylation site, multiple strategies can be used to build an atomic model of the glycan. Some methods rely on attaching pre-built chains to an amino acid side chain,[307] while others aim to add individual monomers one-by-one.[51] The choice of strategy depends mainly on the amount of prior knowledge available and the quality of the experimental data. For an automated method, relying on manually supplied glycan chain information is inconvenient. Therefore, without prior information, the natural choice is to attempt to model monomers individually, guided by the experimental data.

When modelling a carbohydrate onto a potentially glycosylated amino acid side chain, the choice of the first carbohydrate monomer can be trivially determined by the amino acid type. The geometric relationship between a specific carbohydrate monomer and a donor amino acid side chain can be defined entirely by a set of torsion angles (ψ, ϕ, ω),

a set of angles (α, β, γ), and a bond length. After a site is identified, a given protein-carbohydrate linkage can be modelled in an initial state using the mean torsion angles, angles and bond length for each geometric cluster observed in the Protein Data Bank (see Chapter 5 and Supplementary Section 8.3). While useful as an initial starting point, the range of observed validated linkages across the Protein Data Bank, as well as the *ab initio* energy distribution, suggests all protein-carbohydrate linkages can exist in a variety of possible conformations. Modelling a linkage in this average conformation is unlikely to explain the observed experimental data adequately. A more robust solution would be to refine the geometric parameters of each linkage to better fit the experimental data.

Parametric refinement can be carried out using constrained simplex minimisation, chosen over alternative gradient-based methods for efficiency. A mapping function, g, can suppose a potential location of the added carbohydrate group, $\boldsymbol{c}$, from the M geometric parameters, $\boldsymbol{\theta}$, shown in Equation 6.6.

$$\begin{aligned} \boldsymbol{\theta} &= [l, \alpha, \beta, \gamma, \psi, \phi, \omega] \\ \boldsymbol{c} &= g(\boldsymbol{\theta}) \end{aligned} \tag{6.6}$$

The fit of this potential carbohydrate group can then be scored based on density, considering deviations from the calculated ideal geometric parameters. This objective function, f, is shown in Equation 6.7. The difference calculated in the angular penalty term is circular, but omitted from the equation for clarity.

$$f(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{i}^{N} \rho_{\text{exp}}(\boldsymbol{c_i}) + w_{geom}\sum_{j}^{M}\frac{(\theta_j - \mu_j)^2}{\sigma_j^2} + w_{length}\frac{(l - l_{ideal})^2}{\sigma_l^2} \tag{6.7}$$

where:

ρ_{exp} is the experimental density map
λ is a weighting factor
θ_j is the proposed angular value
μ_j is the average angular value
σ_j is the standard deviation of the angular value
l_{ideal} is the ideal bond length
σ_l is the standard deviation of the bond length

After constrained simplex refinement of the parameters, an updated set of parameters, $\boldsymbol{\theta}^*$, should result in better positioning of the carbohydrate group with respect to geometric expectations and experimental density, shown schematically in Figure 6.6. For all the trialled geometries, the carbohydrate that best fits the experimental density can be kept for further processing. For this initial proof-of-concept investigation, the w_{geom} was set to 1×10^{-2} and w_{length} was set to 1 after a small trial-and-error investigation. It should

be noted that this is not a robust estimation of these values, but was deemed acceptable for the purposes of determining whether this method was tractable.

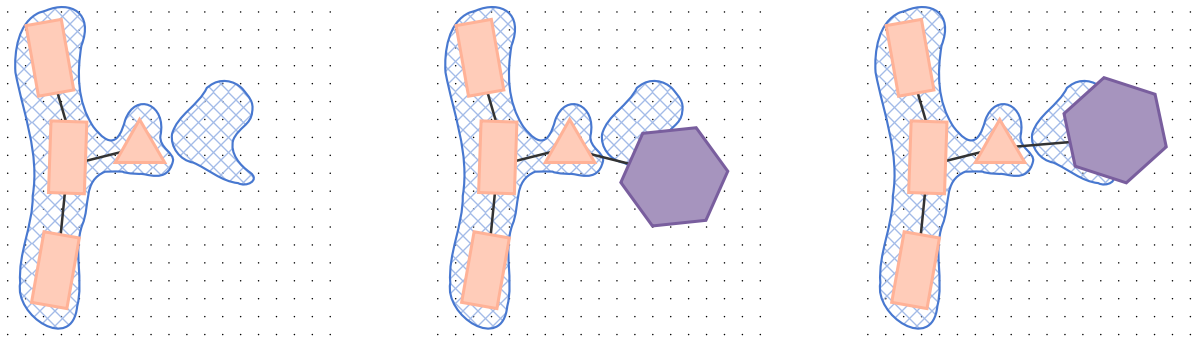


Figure 6.6: Schematic of carbohydrate geometry refinement. After identification of a potential site (left), a carbohydrate can be placed in a mean geometry (centre) with mean bond lengths, bond angles, and torsion angles, but may not fit the experimental density well. Refinement of the geometry can yield a better model of the carbohydrate (right).

When modelling carbohydrates into crystallographic data, after adding carbohydrate atomic positions, it is important to include these positions when calculating the structure factor amplitudes and phases for the current model. After recalculation of the electron density map with the most up-to-date phase information, the region of density supporting the added carbohydrate likely becomes more interpretable, as shown in Figure 6.7.

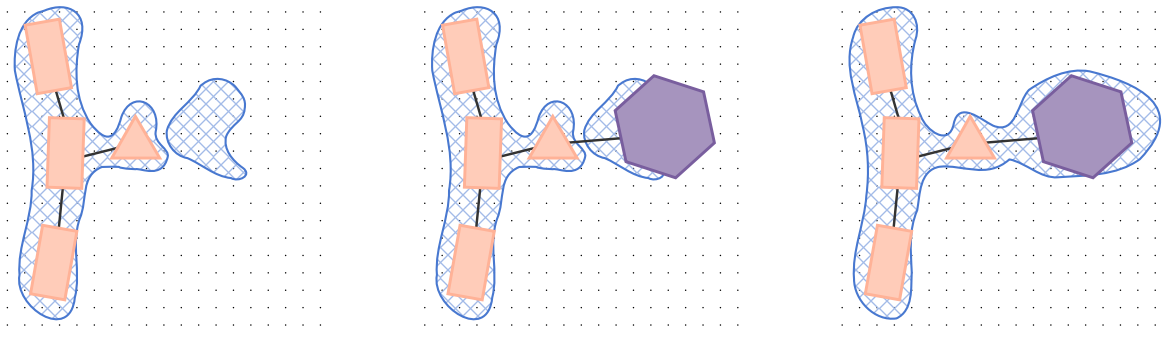


Figure 6.7: Schematic of carbohydrate-addition-induced map recalculation with crystallographic data. Given a potential site with initial electron density (left), a carbohydrate can be modelled (centre). The contributions to the phase of each reflection must be updated to include the newly modelled carbohydrate, which may yield a more interpretable electron density map after recalculating the map (right).

After all potential glycosylation sites are identified and an appropriately positioned carbohydrate is modelled, each added carbohydrate must be critically assessed to ensure support from the experimental density. With crystallographic data, density fit is typically established using a metric such as *RSCC*, whereas modern cryo-EM methods may

use an alternate metric known as Q-score.[308] Placed carbohydrates which exhibit a high score are likely to be correct. In contrast, lower scores indicate a modelled position that is not adequately supported by the experimental data and should therefore be removed.

The threshold at which carbohydrates should be removed directly dictates the sensitivity of this method, and is a crucial choice. In crystallography, an RSCC threshold of 0.80 is commonly used to describe a valid carbohydrate,[91] but is likely too high to serve as a threshold for an automated method that does not refine atomic positions or B-factors, so a lower internal threshold of 0.60 was used as a compromise. With cryo-EM data, a threshold for the Q-score is commonly calculated with respect to the resolution of the experimental data using a cubic polynomial.[309] Since cryo-EM experimental data is not expected to change during automated model building, this threshold can be used to assess any modelled carbohydrate critically. If any of the potentially glycosylated sites were modelled with an additional carbohydrate that scores under this internal threshold, then it should be removed to prevent erroneous model building.

Following validated addition of an initial carbohydrate group, the glycan may be extended by trialling and refining known linkages. For a given carbohydrate, the most frequent linkages observed in the survey of the Protein Data Bank can be trialled and refined at all possible donor positions for each expected geometry. This process can be repeated at all terminal ends of the glycan chain until no further carbohydrates can be added, whilst remaining geometrically valid, chemically valid and explainable by the experimental density. Chemical validity can be enforced simply by ensuring that no two non-bonded atoms are within van der Waals radii of each other.

After this process of recursive addition and critical assessment is complete, the atomic model can be refined with *Servalcat*. Provided geometric restraints for the rings are used during refinement,[205] all carbohydrates should remain in the lowest energy conformation. The refined atomic positions can finally be assessed against the updated electron density map or Coulomb potential map to ensure the added carbohydrates fit the density well. The stricter crystallographic carbohydrate RSCC threshold of 0.80 may be enforced, with any carbohydrates with a poorer fit to density removed. Similarly, carbohydrates modelled into cryo-EM data which do not meet the Q-Score threshold may be discarded.

6.2.2.1 Modelling *N*-glycosylation

N-glycosylation is the most commonly reported glycosylation type in the Protein Data Bank. It is initiated by the addition of a β-GlcNAc sugar to the nitrogen atom of an asparagine side chain by a glycosyltransferase enzyme. As a result of enzymatic addition, nearly all *N*-glycans should exhibit the Man-β1,4-GlcNAc-β1,4-GlcNAc-β-Asn core

group. This relatively conserved glycan core is a helpful characteristic when attempting to automatically model *N*-glycosylation, since the type and linkage of the first three sugars are known and do not need to be determined through trial and error. After this core group, the degree of enzymatic processing can yield a range of other sugars and linkages, with the most frequently observed in the Protein Data Bank outlined in Table 5.1.

To test the proof-of-concept strategy of recursive addition and critical assessment for *N*-glycosylation model building, the method was run on a leukocyte myeloperoxidase structure resolved to 2.49 Å with X-ray diffraction (PDB code: 9SDS[310]). This structure was not used for training the deep learning model, but it should be noted that homologous structures may have been included. Since samples were drawn randomly during training, it is not possible to be certain of the degree of implicit bias. Regardless, the automated model-building method does not use these predictions beyond identifying a potential site, so the use of this structure in assessing the automated method remains valid.

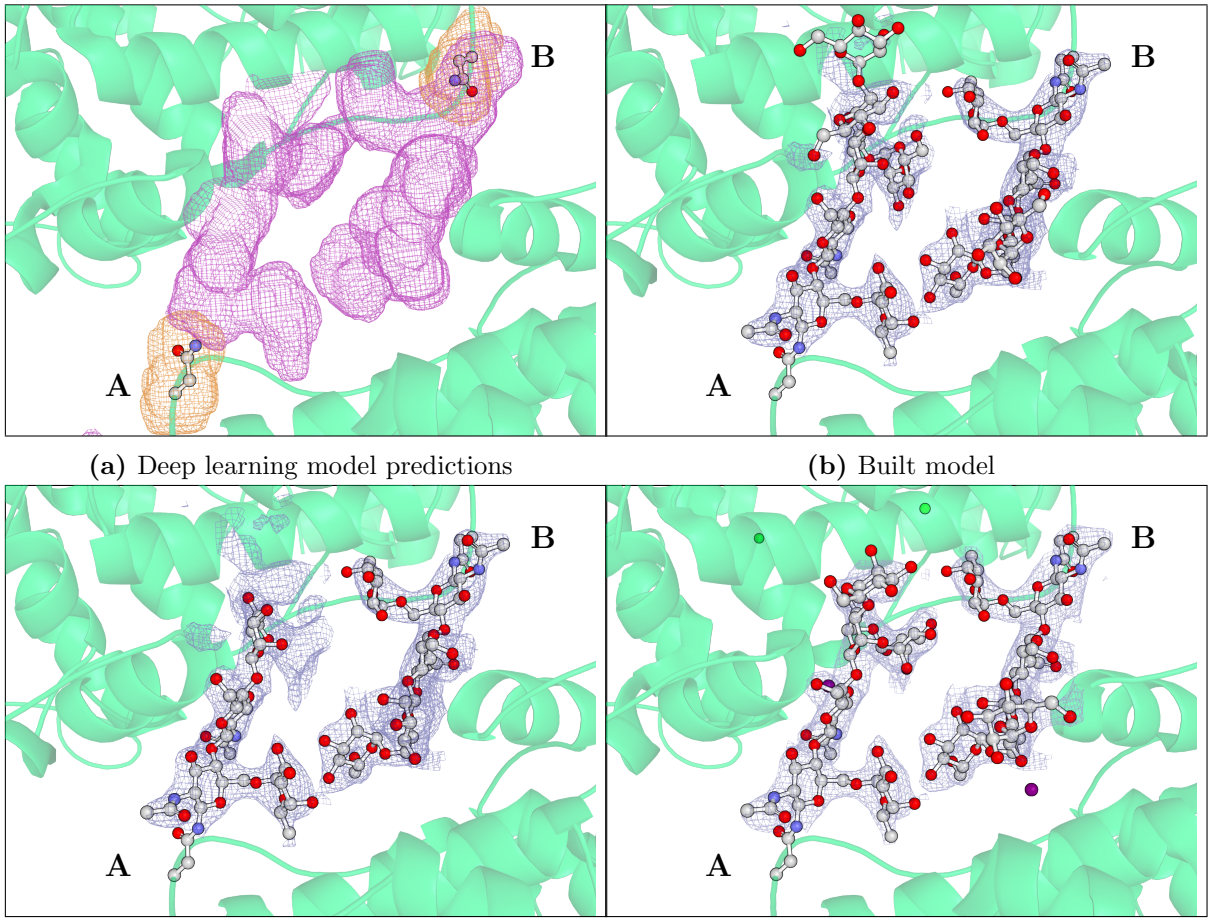
To prepare the structure, the coordinate and structure factor data were downloaded from the Protein Data Bank. All solvent and carbohydrates were removed, followed by 10 cycles of reciprocal-space refinement with *Servalcat*. This process aims to mimic a realistic carbohydrate model-building scenario following protein molecular replacement or model building. The resultant electron density from this deglycosylation process served as the basis for identifying potential glycosylation sites using the deep learning model. The output predicted protein and predicted carbohydrate density is shown in Figure 6.8a for a heavily glycosylated region supporting the dimerisation of the leukocyte myeloperoxidase protein.

Both the predicted protein and carbohydrate density are well defined in this region, revealing a clear protein-carbohydrate binding motif that yields two potential glycosylated sites, labelled A and B. In this example, the available sequence information also identifies these potential sites, but was redundant given the strong predictions. The predicted maps also identified four other potentially glycosylated sites, but were not shown for clarity.

After five cycles of the proposed automated model-building process, the glycosylated chains A and B contain a total of 13 modelled carbohydrates. Both chain A and B contain the Man-β1,4-GlcNAc-β1,4-GlcNAc-β-Asn glycan core, with additional Fuc-α1,6-GlcNAc, Man-α1,3-Man and Man-α1,6-Man linkages. This process also modelled an additional terminal Man-α1,2-Man linkage in chain A, after the Man-α1,3-Man linkage. The resultant glycosylated protein model was then refined with *Servalcat*, shown in Fig-

ure 6.8b.

Following refinement, a critical assessment of the resultant structure is necessary to reduce the likelihood of incorrect results. Using the refined model and refined electron density map, any carbohydrate which does not meet the minimum RSCC requirement of



(a) Deep learning model predictions **(b)** Built model

(c) Built and validated model **(d)** Deposited model

Figure 6.8: Automated model-building results for a leukocyte myeloperoxidase structure resolved to 2.49 Å with X-ray diffraction (PDB code: 9SDS[310]), with a focus on C-ASN-248 (Label A) and D-ASN-248 (Label B). A - Protein (orange) and carbohydrate (purple) density predictions from the optimised multiclass segmentation deep learning model. The input to the deep learning model was calculated from the carbohydrate and solvent removed deposited model and deposited structure factors with *Servalcat*. The clear regions of predicted density suggest areas of potential glycosylation, which have not been modelled. B - Refined atomic model built using the recursive addition method. Both chains A and B exhibit all sugars found in the deposited model, with chain A containing another terminal α-Man carbohydrate. The $2mF_o - DF_c$ density map, shown at 1.5 σ, was obtained from the output of reciprocal-space refinement with *Servalcat*. C - Validated atomic model with three carbohydrates with an RSCC of less than 0.80 removed. The $2mF_o - DF_c$ density map, shown at 1.5 σ, is the electron density map used for the RSCC calculation, not one calculated after re-refinement of the validated atomic model. D - Deposited model with deposited $2mF_o - DF_c$ density map, shown at 1.5 σ. The deposited model contains two well-modelled glycan chains, which are likely to form intermolecular stabilising interactions which keep the leukocyte myeloperoxidase dimer in complex.

0.80 is removed to leave a resultant structure with 10 modelled carbohydrates, shown in Figure 6.8c. The three carbohydrates which did not meet the criteria were terminal α-Man sugars, all with an RSCC around 0.65. Compared with the deposited structure, the remaining 10 carbohydrates were modelled well with all ring atoms within 2 Å to those deposited, and of the correct carbohydrate and linkage types. This positive result suggests that the relatively simple method of recursive addition, refinement, and validation works well to model *N*-glycosylation when the supporting electron density is well defined. The method evidently becomes less effective in the terminal regions of this example, where the electron density is of poorer quality, resulting in the strict validation thresholds remaining unmet. This characteristic is particularly interesting, since two of the three terminal carbohydrates removed after automated model building were well modelled in the original deposition, with RSCC values greater than 0.80. Slight atomic position deviations in the terminal carbohydrates likely cause this weaker RSCC, which cannot be corrected during either the internal real-space or external reciprocal-space refinement processes.

6.2.2.2 Modelling *C*-glycosylation

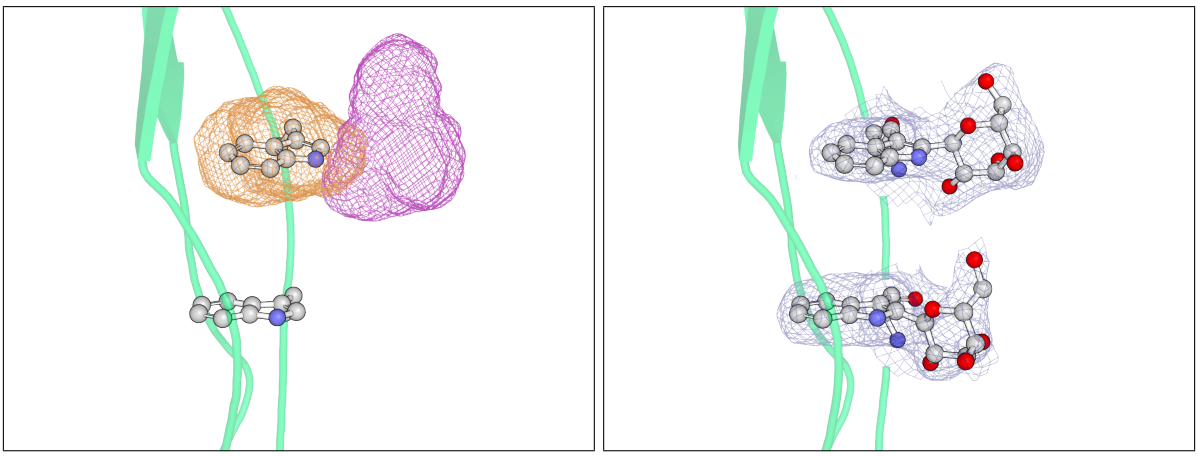
C-glycosylation is a relatively simple but rare form of glycosylation, which occurs through the attachment of a single α-Man sugar in the $^1C_4$ chair conformation to the indole side chain of a tryptophan. This attachment should be relatively trivial to model, given sufficient data quality. To test this, automated carbohydrate model building was run on a mouse ectodomain, which was resolved to 2.80 Å resolution using X-ray diffraction (PDB code: 6OOL) and was not involved in the deep learning model training. A deglycosylated model and accompanying electron density map were calculated using the procedure described in Section 6.2.2.1 and served as the starting point for the automated method.

The predicted protein and carbohydrate density in the area of the *C*-glycosylation is shown in Figure 6.9a. The predictions identify only one of the two potentially *C*-glycosylated sites in this particular region. This is an interesting result, since it suggests that the deep learning model is correctly able to isolate the experimental density corresponding to the $^1C_4$ Man residue, but not well enough to predict both sites correctly. The likely reason for this subpar prediction is the relatively less well-defined electron density in the area around the second tryptophan, as well as the inherent ambiguity of *C*-glycosylation, which can occur either singularly or in pairs depending on the sequence. Despite the poor prediction, the method identifies the second site by recognising the conserved Trp-X-X-Trp sequence.

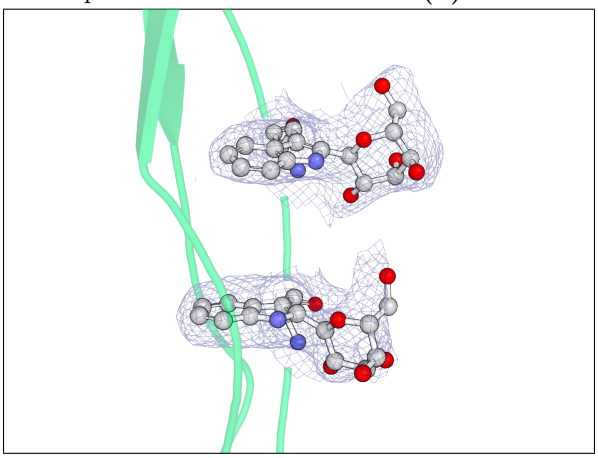
Following a single cycle of the automated model-building method and reciprocal-space refinement, both potential *C*-glycosylation sites were successfully modelled with RSCC values greater than 0.80, surpassing those of the deposited model, shown in Figure 6.9b

and Figure 6.9c. By default, the automated method identifies all potential glycosylation sites from the predicted maps, resulting in the output model also accurately modelling the first sugar of an *N*-glycan chain located in an alternate region of the protein.

Overall, in this singular example, the method was successfully able to model *C*-glycosylation accurately. The predictions failed to identify one of the glycosylated sites, so without sequence information, it is possible that the method would only partially model *C*-glycosylation correctly.



(a) Deep learning model predictions **(b)** Built and validated model



(c) Deposited model

Figure 6.9: Automated model-building results for a mouse ectodomain structure resolved to 2.80 Å with X-ray diffraction (PDB code: 6OOL[311]), with a focus on A-TRP-252 (top) and A-TRP-249 (bottom). A - Protein (orange) and carbohydrate (purple) density predictions from the optimised multiclass segmentation deep learning model. The input to the deep learning model was calculated from the carbohydrate and solvent removed deposited model and deposited structure factors with *Servalcat*. Only one of the two *C*-mannosylated sites was successfully predicted by the deep learning model, highlighting a clear limitation in predictive performance. B - Automatically built, refined and validated model using the recursive addition method. Two TRP-MAN linkages were modelled, both with an RSCC greater than 0.80. The $2mF_o - DF_c$ electron density map, shown at 1.5 σ, was obtained from the output of reciprocal-space refinement of the built atomic model. C - Deposited model showing two TRP-MAN linkages, shown with the deposited $2mF_o - DF_c$ electron density map at 1.5 σ.

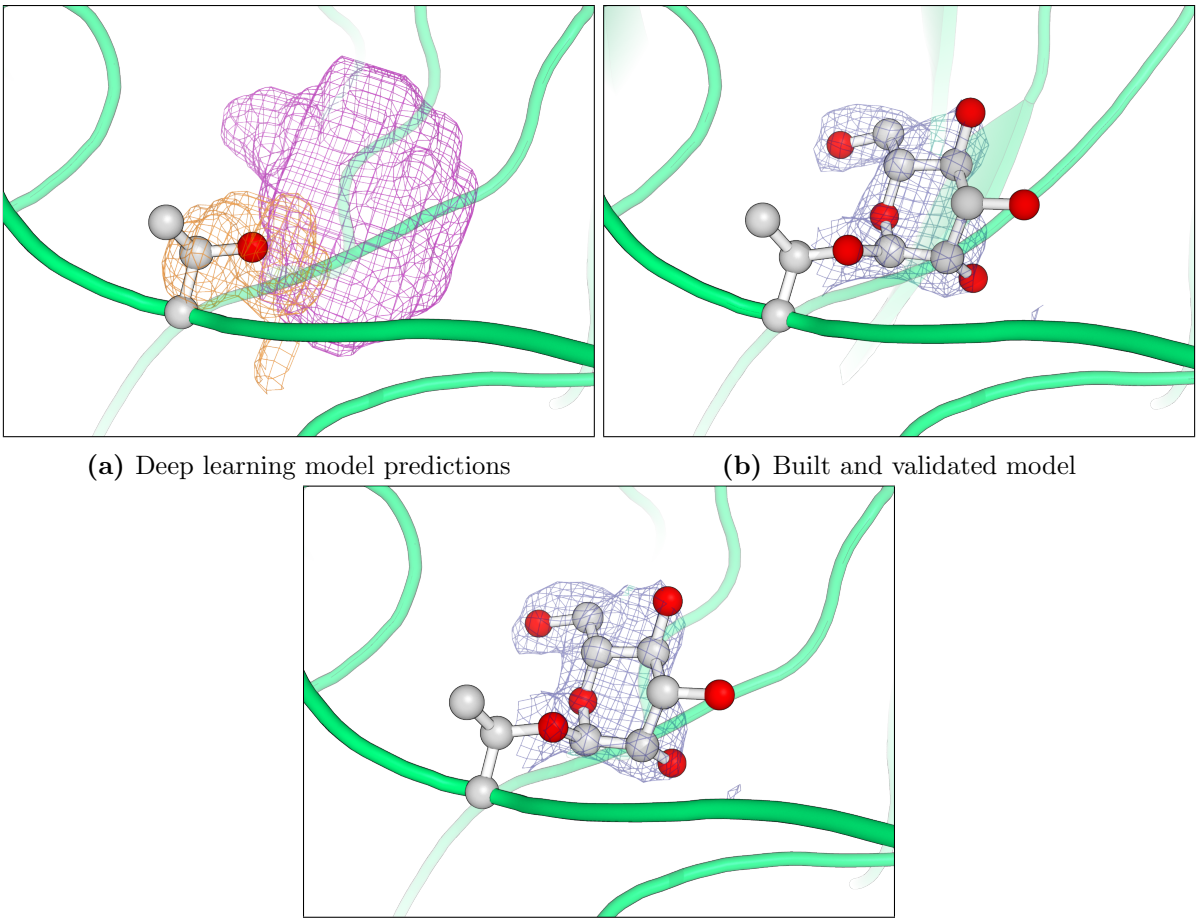
6.2.2.3 Modelling *O*-glycosylation

A wide array of *O*-linked carbohydrate addition has been found in natural systems, but no conserved sequence has been identified to be able to predict where sites of *O*-glycosylation will occur. Modelling most commonly occurs through manual inspection of the density surrounding the side chains of serine or threonine amino acids. The need for manual model building for *O*-glycosylation underscores the promise of this automated method, provided the deep learning model correctly captures and identifies potential sites. Only two *O*-glycosylated linkages were identified in the Protein Data Bank survey that had sufficient validated occurrences to calculate statistics. As a result, the current automated method may only model the most common linkages, Man-α-Ser and Man-α-Thr. If this method is successful, it could be extended provided accurate information can be obtained for the other known *O*-glycan linkages.

To test the method for *O*-mannosylation identification and model building, the procedure was run on a fructofuranosidase enzyme solved to 1.86 Å using X-ray diffraction (PDB code: 8BES[312]). This structure was chosen because it was not part of deep learning training, and because the glycosylated form is structurally novel.

After predicting the protein and carbohydrate density from a deglycosylated electron density map, 8 of the 14 modelled O-glycosylation sites were identified, with an example shown in Figure 6.10. The automated model-building method successfully models $\alpha-$Man sugars onto four of these sites, with RSCCs of approximately 0.77 and 0.80, and two sites with RSCCs greater than 0.80, leaving the remaining two sites unmodelled.

While this result is not as strong as the results for *N*-glycosylation or *C*-glycosylation, it does show significant promise. The ability to locate 8 of the 14 *O*-glycosylation sites, alongside 35 *N*-glycosylation sites from density alone, suggests the method can be helpful in certain circumstances. With sufficient quality data, this automated method is likely to work well with *O*-glycosylation. The limitations in this result are less likely to be due to the technique but rather to the resolvability of the density in the example structure.



(a) Deep learning model predictions **(b)** Built and validated model

(c) Deposited model

Figure 6.10: Automated model-building results for a fructofuranosidase enzyme solved to 1.86 Å using X-ray diffraction (PDB code: 8BES[312]) with a focus on C-THR-163. A - Protein (orange) and carbohydrate (purple) density predictions from the optimised multiclass segmentation deep learning model. The input to the deep learning model was calculated from the carbohydrate and solvent removed deposited model and deposited structure factors with *Servalcat*. The predictions reveal a clear, potentially glycosylated threonine residue. B - Automatically built, refined and validated model using the recursive addition method. A THR-MAN linkage was modelled with an RSCC greater than 0.80. The $2mF_o - DF_c$ electron density map, shown at 1.5 σ, was obtained from the output of reciprocal-space refinement of the built atomic model. C - Deposited model showing the deposited THR-MAN linkage, shown with the deposited $2mF_o - DF_c$ electron density map at 1.5 σ.

6.2.2.4 Results and Discussion

To confirm that the relatively successful results shown for singular examples are not anomalies, a wider array of structures must be examined. The automated model-building process was run for the 234 X-ray diffraction structures and 109 cryo-EM structures that formed the unseen validation set used during deep learning training, outlined in Section 6.1.1.3. The success of the method may be determined by two parameters, the percentage of correctly identified carbohydrate sites and the completeness of the resultant atomic model. Completeness here is measured in the same way as described for nucleic acids (see Chapter 4), where a correctly positioned residue is determined when all

six ring atoms are within 2 Å of all six corresponding ring atoms in the deposited model, and is of the correct type.

The underlying assumption of this method of tracking success is that the models in the Protein Data Bank are accurately modelled. While the geometric validation of these models by *Privateer* certainly aid this assumption, the possibility that the carbohydrates in this test set are incorrectly modelled should be considered. Manual investigation of every carbohydrate in each structure would provide certainty, but it was deemed impractical for this initial investigation into an automated method. Instead, both the completeness of carbohydrates with a validated fit to density, and the completeness of all carbohydrates are reported in Table 6.5.

Table 6.5: Results of site identification and automated model building for the test set of *N*-linked and *O*-linked glycans determined with both X-ray diffraction and cryo-EM. Both deposited and validated counts are shown here to account for any potential modelling errors in the deposited models.

(a) Site identification results

Method	**Type**	**Deposited Count**		**Validated Count**	
		Found	Total	Found	Total
X-ray diffraction	*N*-linked	575	748	354	378
	O-linked	51	77	24	34
cryo-EM	*N*-linked	732	1836	573	1148

(b) Model building results

Method	**Type**	**Deposited Count**		**Validated Count**	
		Modelled	Total	Modelled	Total
X-ray diffraction	*N*-linked	854	1205	517	589
	O-linked	33	80	17	36
cryo-EM	*N*-linked	1211	2700	1068	1718

6.2.2.4.1 *N*-glycosylation

Across N-glycosylated structures in the test set solved with X-ray crystallography, 575 out of the 748 reported N-glycosylated sites were correctly identified using the predictions output from the deep learning model. This is a promising result, suggesting that the site identification technique works well. The unfound sites could be attributed to poor deep learning model performance, but may also result from the reported sites not being well supported by the associated electron density. This may be confirmed by measuring the proportion of sites found that were deposited with a well-modelled first sugar, as indicated by an RSCC greater than 0.80. This measurement shows that 354 of the 378 validated sites were successfully identified using the method, suggesting that the deep learning model can isolate potential N-glycosylated sites effectively with sufficient electron density. Once sites are identified, they are supplemented with all available sequence information to maximise the likelihood of successful model building. Hence, the results of model building in this investigation are effectively independent of the identified sites.

The recursive addition approach for model building yields successful modelling of 854 carbohydrate monomers out of the 1,205 deposited. While this method should not produce a monomer with a poor electron density fit, the same cannot be said for the deposited models. Therefore, results in comparison to carbohydrates with an arbitrary electron density fit should be taken with caution. A better approach is to measure the completeness of the automated model with respect to well-modelled deposited carbohydrates, which reveals that the method successfully modelled 517 of the 582 validated carbohydrate monomers. This high level of model-building completeness is a promising result and provides evidence that this relatively simplistic method can be used to model N-glycosylated carbohydrates efficiently. Investigations into why some validated carbohydrates were left unmodelled reveal two clear weaknesses in this approach. The first problem is that this method assumes an expected structure of an N-glycan, the $\mathrm{Man}-\beta 1,4-\mathrm{GlcNAc}-\beta 1,4-\mathrm{GlcNAc}-\beta-\mathrm{Asn}$ core group is almost always present, but any deviation from this due to specific engineering or rare biological pathways leave this method unable to cope. Secondly, this method implicitly enforces the geometric relationships between carbohydrates to conform to expected distributions, however, ideality may diverge from reality in specific circumstances. Interactions which stabilise uncommon geometric conformations, as discussed in Section 5.3.2.2.1, are not well considered in this method.

Compared with crystallographic N-glycan site identification, this method struggles with cryo-EM data. 732 of the total reported 1,836 N-glycan sites were successfully identified, and only 573 of the 1,148 validated sites were found. These results are approximately

in line with what could be expected given the poor performance metrics observed after deep learning model training on the same cryo-EM data (see Section 6.1.2). Similarly, the results of model building suggest the method is weaker with cryo-EM data than with crystallographic data. Only 1,211 of the 2,700 deposited carbohydrate monomers and 1,068 of the 1,718 validated carbohydrate monomers were successfully modelled, yielding a lower completeness than observed for *N*-glycans across crystallographic examples.

This method of carbohydrate addition is highly dependent on the quality of the available experimental data, since both geometric refinement and scoring explicitly incorporate experimental data. Some evidence for this factor being the leading cause of the poorer performance can be gathered by measuring the B-factor for successfully and unsuccessfully modelled carbohydrate monomers in the deposited structures. The average deposited B-factor for a successfully modelled carbohydrate in crystallography was 68.4 ± 40.7 Å^2, whereas unsuccessfully modelled carbohydrate monomers had an average deposited B-factor of 90.9 ± 49.6 Å^2. With cryo-EM data, the average deposited B-factor for successful carbohydrates was 87.5 ± 41.1 Å^2 and for unsuccessful carbohydrates was 124.4 ± 70.4 Å^2. Although a crude method of analysis, higher B-factors generally indicate poorer experimental data quality around carbohydrates, suggesting that this method works best when data quality is highest. A highly detailed analysis of the performance of this initial method for *N*-glycosylation model building was not completed, since the technique evidently requires further careful optimisation to achieve better performance.

6.2.2.4.2 *O*-glycosylation

For *O*-glycosylated X-ray diffraction models, 51 of the 77 reported *O*-glycan sites were identified by the automated method, with successful identification of 25 of the 34 validated sites. This is an outstanding result and represents a novel software capability. Since *O*-glycosylation has no known consensus sequence, identification using this method is likely to be abundantly helpful in structure solution, but also means that if any sites are unfound, then nothing will be modelled using this method. 17 of the 25 identified validated sites were successfully modelled with an α-Man, with the remaining 8 unmodelled due to the method failing to account for linkages other than Man-α-Ser and Man-α-Thr. Automatically modelling a large proportion of *O*-glycosylation in the test set is an excellent achievement that no current method can achieve. Depositions of *O*-glycan models in the Protein Data Bank are few, likely due to difficulties attributed to identification and modelling. This method may serve as a solid foundation for developing a more potent *O*-glycosylation model-building method that can adequately depict the heterogeneity and complexity of *O*-glycosylation.

6.2.2.5 Conclusions and Future Work

The novel application of deep learning network predictions to identify potentially glycosylated sites has been a success overall. *N*-glycosylated, *O*-glycosylated and *C*-glycosylated sites have been identified using realistic crystallographic data with high accuracy. The deep learning predictions on cryo-EM data are comparatively weaker, yielding less accurate predictions of potentially glycosylated sites, albeit still good enough to be useful during interactive structure solution. Improving the capability of cryo-EM site identification is likely to be a fruitful avenue for future work. A variety of post-processing techniques applied to the Coulomb potential map may improve interpretability, which in turn may allow deep learning models to perform better. Alternatively, exploring other model architectures or embedding structural statistics into the deep learning model may yield a more sensitive method that performs better across both crystallography and cryo-EM.

The complementary method of automated model building has shown varying levels of success. To avoid producing models with errors and potentially perpetuating known issues in carbohydrate modelling,[162] the method was designed to be very stringent, requiring very high thresholds to produce carbohydrate models. As a result, this model-building method works well when the experimental data is of good quality and the carbohydrates meet expectations, but may fail outside these specifications. It is most likely that the naive approach to geometric refinement is at the root of some of these issues, and more appropriate weighting and scoring may produce better models in edge cases, which could be an area for future exploration. One area that should certainly be explored further is the modelling of *O*-glycosylation, another novel capability this group of methods provides. While *O*-mannosylation is the most commonly reported type of *O*-glycosylation in the Protein Data Bank, it is far from the only one and automated model-building methods should take into account many more potential types of *O*-glycosylation, since they are crucial to a variety of biological systems.[174] Geometric data for all these potential linkages are unlikely to be available in the Protein Data Bank alone, so *ab initio* calculations may be required to obtain this data robustly.

6.3 Identification and Automated Model Building of Unmodelled Glycans

The capability of this automated method to isolate and potentially model glycans from experimental density alone affords a unique opportunity for discovery. Given the vast number of deposited structures in the Protein Data Bank, it is likely that some protein glycosylation remains unmodelled despite adequate support from experimental data. To investigate whether this hypothesis is true, the glycosylation-site identification method followed by automated model building was applied to all X-ray diffraction structures in the Protein Data Bank. In theory, carbohydrates modelled by the automated method that lack a modelled analogue in the deposited structure may be novel. Given the rigorousness of the technique, designed to reduce false positives, any new carbohydrate modelled may provide new insights into the function of a given biological system.

A survey of the Protein Data Bank revealed 7,054 potentially unmodelled *N*-glycosylated sites across 3,136 structures. The automated building method modelled 2,463 carbohydrates in 1,839 *N*-glycans across 942 deposited structures, all with an RSCC greater than 0.80. In addition to the potential *N*-glycans, 1,572 potential *O*-glycosylated sites were identified in 1,348 structures, as well as 198 potential *C*-glycosylation sites in 170 structures. Surprisingly, only 48 *O*-mannosylation sites and 2 *C*-glycosylation sites could be modelled with density fit validity. A number of these probable sites may be false positives arising from the binding of other biologically relevant molecules in the vicinity of potential donor amino acid side chains. For instance, phosphorylation often occurs at serine or threonine residues and may lead to false-positive site identification. The rigour of the automated model-building method is unlikely to model in such instances, but the final conclusion, as to whether each of these added carbohydrates is correct, must come from manual verification, which was not completed in this rudimentary analysis. Therefore, without verification, these results should be taken with significant caution and should form the basis of further study. A single case study is considered to reinforce the power of this automated method when combined with manual verification.

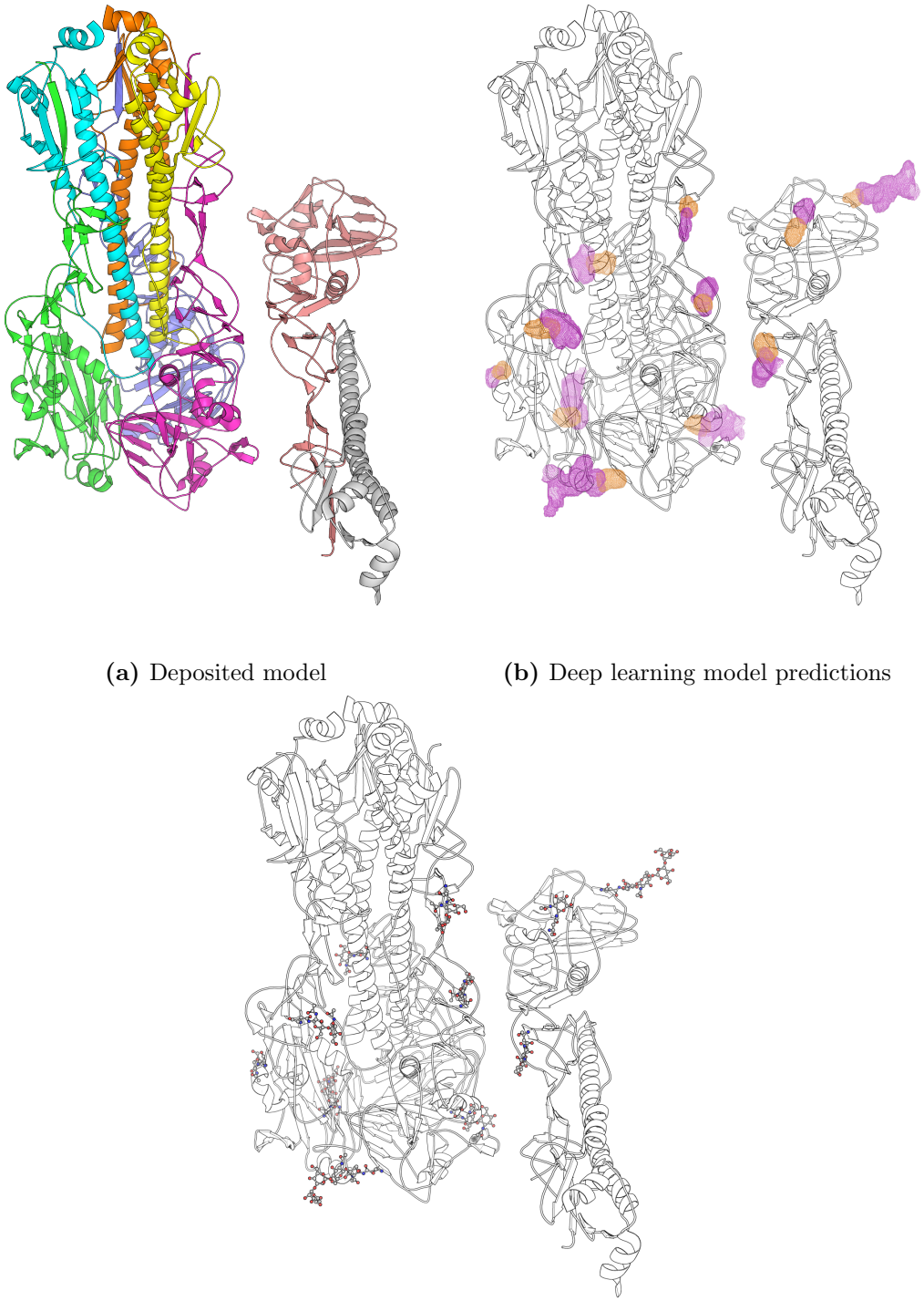
6.3.1 Case Study: *de novo* carbohydrate model building of a hemagglutinin from an unidentified influenza virus

The hemagglutinin protein on the surface of influenza viruses enables the virus to bind to and enter host cells. Hemagglutinins (HA) are commonly heavily glycosylated, which is used as a buffer to evade host immune systems.[313] A novel strain of the canine influenza virus (CIV) was detected in the spring of 2015 in Illinois, USA, and subsequently caused widespread infections across the USA. The structure of two HA proteins was resolved

to 3.01 Å using X-ray crystallography (PDB code: 6N4F[314]). Although glycoproteins, the HA proteins were deposited with no modelled carbohydrates. The authors, Pulit-Penaloza *et al*, comment in the original publication that "seven asparagine-linked glycosylation sites (NXS/T) were predicted in the CIV HA monomer; however, no interpretable carbohydrate electron density was observed at any location". The carbohydrate model-building efforts undertaken by the original authors are almost certainly more nuanced than this simple comment provides, but allude to a definite difficulty in interpretation.

Using the deposited structure factors and the deposited model, *Servalcat* was used to calculate an electron density map, which was provided to the deep learning model-based identification method. 11 potentially glycosylated sites were identified across the entire HA complex, 3 of which were on the HA monomer reported by the original authors. The automated model-building method was able to model 7 *N*-glycosylated carbohydrates at all three identified sites on the HA monomer and an additional 18 carbohydrates elsewhere on the complex, shown in Figure 6.11. All carbohydrates were validated as correct by *Privateer*, and manual inspection supports all glycosylated positions.

The *N*-glycans modelled by the automated method may provide further understanding for the particular viral protein, perhaps influencing complex formation or host-protein interactions. If this automated method had been available to the original authors, some of the difficulty associated with carbohydrate model building would likely have been alleviated. While somewhat speculative, the power and usefulness of this automated method are clear in this case study.



(a) Deposited model **(b)** Deep learning model predictions

(c) Automatically built model

Figure 6.11: Automated model-building results for two previously unknown HA proteins, resolved to 3.01 Å using X-ray crystallography (PDB code: 6N4F[314]). A - The deposited model, which has no modelled carbohydrates. B - The protein (orange) and carbohydrate (purple) deep learning predictions, which suggest many areas of potential unmodelled glycosylation. C - Automatically built model using the recursive addition method, yielding 25 novel carbohydrates which were not determined in the deposition.

Chapter 7

Overall Conclusions and Future Work

> All models are wrong, but some are useful.
>
> *George E. P. Box*[315]

The initial goal of this PhD project was to improve automated model-building methods for nucleic acids solved with X-ray diffraction. The existing software package *Nautilus* is a robust software package for rapid nucleic acid modelling, using statistical fingerprinting to identify regions of experimental density likely to originate from nucleotides. The major limitation of this method is the quality of the experimental density map. As is a common scenario during the structure solution of nucleic acid-containing biological molecules, errors in the phase estimates for crystallographic data, as well as nucleic acid flexibility, contribute to difficult-to-interpret regions of experimental density.

Chapter 2 describes a proof-of-concept deep learning based method for isolating probable nucleic acid regions. Using three U-Net convolutional neural networks, the positions of the phosphate, sugar and base groups could be isolated with respectable accuracy. The methodological advancements outlined in this chapter allowed a deep learning model to understand relationships in the electron density that are incredibly difficult even for a trained human to perceive. The major drawback of this method was a lack of precision, resulting in unwanted false-positive predictions. Since the phosphate, sugar, and base predictions were contained within separate models, it is unlikely that the inherent biochemical relationship between the three nucleic acid constituents was understood.

Chapter 3 builds on this deep learning method to increase performance. A greater consensus between the three output targets was achieved by migration to a multiclass U-Net. Sharing parameters between the three outputs significantly increased the precision of the

model, but initially came at the cost of recall. Extensive experimentation on the architecture of the deep learning model was completed to recover this performance loss while maintaining the desirable high precision. The number of convolutional filters, as well as the spatial size and depth, were independently explored to determine an optimised spatial dimension and number of filters. Throughout this, the total number of parameters in a given model was strongly considered in decisions made about the model architecture. By creating a smaller model, the method becomes more accessible to more people, since there is little need for significant computational investment. This was of paramount importance, since these techniques should be accessible to everyone, regardless of financial status. Additionally, since the method was effectively redeveloped, it enabled the inclusion of cryo-EM data, producing an optimised and unified method that isolates the positions of the phosphate, sugar, and base groups with excellent performance in realistic experimental data scenarios.

Chapter 4 applies these deep learning predictions to guide automated nucleic acid model building. After isolation of a nucleic acid backbone using the phosphate and sugar predictions, a library of trinucleotide fragments can be trialled to produce a nucleic acid model. This software package, *NucleoFind*, produces more complete models than the existing method *Nautilus* in realistic crystallographic and cryo-EM examples. *NucleoFind* was used as a replacement for *Nautilus* in the model-building pipeline *ModelCraft* to realise significant automated model-building completeness improvements. This new method generally meets the initial goal of improving automated nucleic acid model building, but is not perfect. Commonly, models output from *ModelCraft* with *NucleoFind* contain arbitrary assignments of bases, since the sequencing module within *NucleoFind* was taken directly from *Nautilus* without any modification. An interesting avenue for future work would be improving the sequencing ability of *NucleoFind*. Perhaps the development or use of similar deep learning methods could better enable the identification of the base type supported by the experimental density, thereby improving the output model.

The development of the deep learning model for the identification of nucleic acids in difficult-to-interpret experimental density is one of the most significant advancements made during this PhD. As the goal of improving nucleic acid model building was generally achieved, thought turned to other interesting applications of similar models. One area of biochemistry that is often difficult to study is glycosylation, since the experimental density of glycans is often hard to interpret. If carbohydrates could be identified using a similar deep learning model, this information could be combined with data obtained from the Protein Data Bank to guide automated carbohydrate model building.

Chapter 5 outlines the background data collection for the goal of automated carbohydrate

model building. Linkage geometry data between validated carbohydrates was obtained from the Protein Data Bank for the purposes of model building, but were also shown to be useful in carbohydrate validation. By observing trends in the Protein Data Bank and *ab initio* calculated geometric parameters, structures can be compared against expected geometric distributions to ensure validity. These expected geometric parameters may be used as initial trials in model building once potential glycosylated sites have been identified.

Chapter 6 investigates whether the deep learning models designed for nucleic acid identification could be used to identify carbohydrates. The deep learning models trialled showed a similar trend to that observed for nucleic acids, with an initial model that performed generally well but was seen to be imprecise. Adding another predicted class to force the model to be more conservative, as described in Chapter 3, yielded more precise results with similar recall. This multiclass model yields predicted locations for carbohydrates and glycosylated amino acids, which can be used to isolate potentially glycosylated sites using experimental density alone. Following this, a relatively primitive method of recursive addition and critical assessment, using the geometric data obtained in Chapter 5 can produce an atomic model of the carbohydrates with reasonable accuracy. This technique has been shown to work with *N*-glycans, *O*-glycans, and *C*-glycans and provides a solid foundation for further development into a potent automated carbohydrate model-building method.

Overall, the methods outlined in this thesis allow both nucleic acids and carbohydrates to be modelled without significant manual intervention. X-ray crystallography was a primary focus for method development, but since cryo-EM generally produces similar experimental data, methods were tested using both experimental techniques. The optimised deep learning model architecture evidently works well in identifying difficult-to-interpret regions of experimental data, but certainly struggles more with cryo-EM data. This has been seen as a consistent factor throughout carbohydrate and nucleic acid model building, and may be another area where future studies can improve.

7.1 Implications for the Field

Given the general desire for complete automation in the structure solution process from X-ray diffraction or cryo-EM data, the methodological advancements made during this PhD are well-placed and likely to be a positive contribution to the field. For X-ray analysis, unattended data collection at many synchrotrons,[316,317] automated data processing,[318] and automated molecular replacement[235,236] can yield a starting point for the structure solution of nucleic acid or carbohydrate-containing biological molecules without

much or any manual involvement. The automated model-building methods developed for nucleic acids and carbohydrates are able to continue this trend of automation to yield an atomic model. In some cases, this atomic model is sufficient for structural inference, whereas in others, manual intervention remains necessary.

The optimised U-Net deep learning model, which underpins these two new automated methods, has been shown to perform well for both nucleic acids and carbohydrates, despite being optimised solely for nucleic acid identification. The relatively successful application of this model architecture to carbohydrates suggests that this design may also apply to other systems across the field of structural biology. Applying this optimised model architecture to similar modern convolutional neural networks for density modification,[319] crystal solvent estimation[320] or electron density sharpening[321] may yield performance gains, though this remains a conjecture without further work.

Additionally, convolutional neural networks are widely used in deep-learning applications for phase recovery in small molecule crystallography,[322,323] and are an active area of study for obtaining phase estimates for diffraction data of biological molecules.[324] If a software method could successfully synthesise the phase information lost during X-ray diffraction, it would likely alleviate the long-standing phase problem in X-ray crystallography and make structure solution routine. Leveraging the knowledge gained from experimenting with convolutional neural network architectures in this thesis, in combination with existing other work,[325] could constitute a more performant method for achieving this ambitious goal.

Tuning and optimising a machine learning model often involves searching for a set of hyperparameters that yields the best performance. For smaller machine learning tasks, this can be a completely automated process, however, for larger machine learning models, extensive parameter trials which vary hyperparameters individually can be computationally prohibitive. The more manual protocol described and developed for model optimisation in this thesis may be helpful as a learning aid or template for other investigators in the field to obtain small and performant machine learning models.

Overall, the use of deep learning to understand crucial structural information of biological macromolecules from experimental or sequence data is a transformative approach that will undoubtedly be relied upon in many future methods. As more structural data is obtained in the future through various techniques, software methods will likely become better, and perhaps more importantly, biological understanding will improve. Developing methods that aid the determination of structural data is therefore extremely important, since advanced software methods provide a robust data foundation on which future tech-

niques are likely to be based. As a showcase of the functionality of these automated model-building methods with crystallographic data developed during this PhD, a final case study is considered.

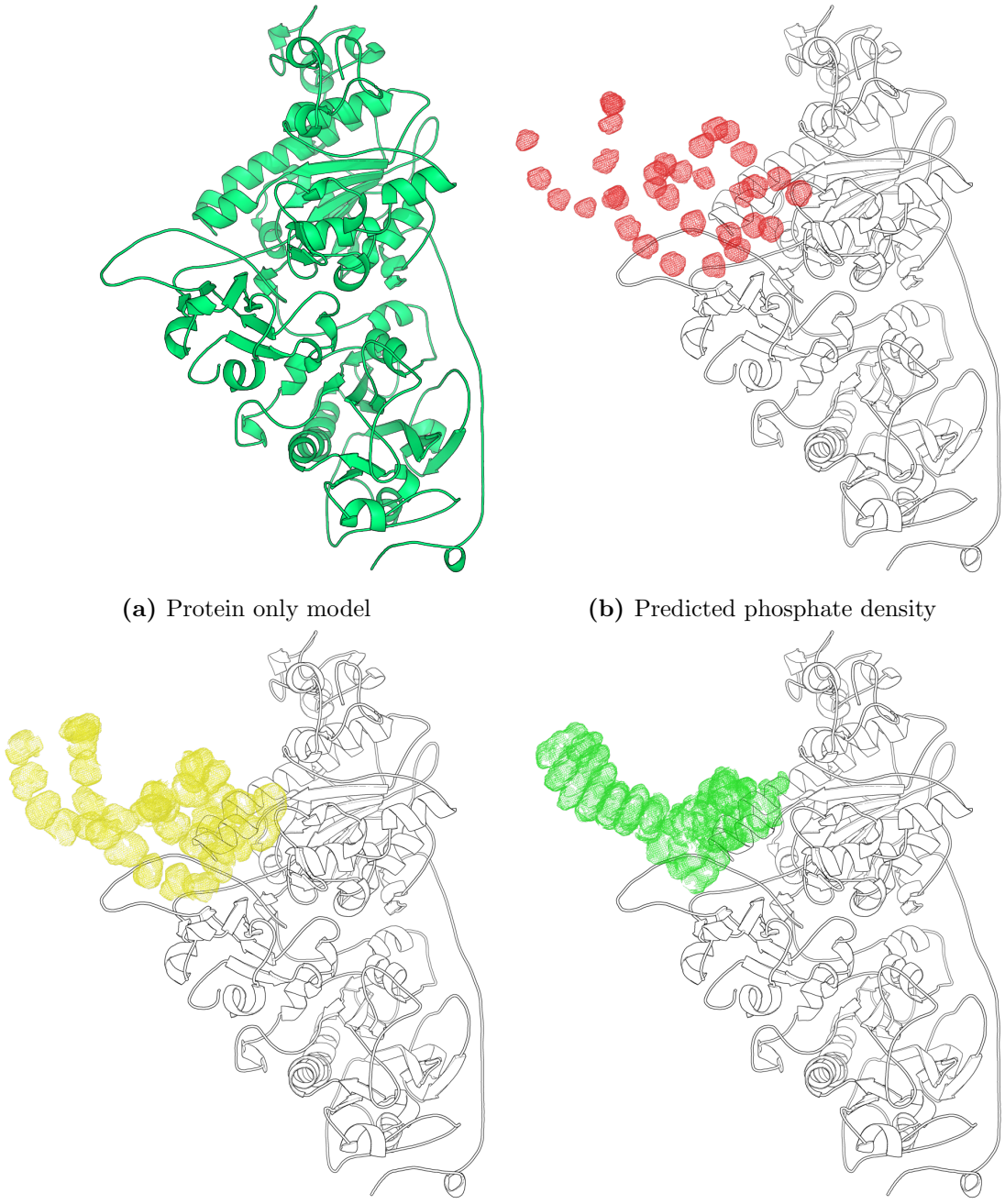
7.2 Case Study: *de novo* nucleic acid and carbohydrate model building of a glycosylated mouse autotaxin-DNA complex

Autotaxin is an enzyme that hydrolyses extracellular lysophospholipids to form lysophosphatidic acid (LPA). LPA is used in a variety of signalling pathways and induces growth factor-like responses in certain cells. Autotaxin, therefore, is an attractive target for inhibition to target specific diseases such as pulmonary fibrosis.[326] The structure of a mouse autotaxin in complex with an inhibiting DNA aptamer was solved at 2.00 Å resolution with X-ray crystallography (PDB code: 5HRT[326]). As an exercise in automated model-building, all nucleic acids and carbohydrates were removed from the deposited structure, and the protein B-factors were set to the average value. The deposited structure factors and protein-only structure were used to calculate a starting electron density map after 10 cycles of refinement with *Servalcat*. This process aims to mimic the result of protein molecular replacement from an *in silico* or homologous model, shown in Figure 7.1a.

The starting electron density map was provided to both the nucleic acid and carbohydrate identification deep learning models, with results shown in Figures 7.1b-7.1f. All predictions are strong, isolating areas of the density corresponding to the nucleic acid phosphate, nucleic acid sugar, nucleic acid base, carbohydrate protein donor and carbohydrate. To model the nucleic acid, *ModelCraft* with *NucleoFind* was run with default parameters after providing the protein-only atomic model and structure factor data. The resultant model, shown in Figure 7.1g, contains 28 automatically modelled nucleotides with all backbone atoms within 2 Å of the corresponding nucleotide in the deposited model. This equates to a 93.3 % completeness, with only two terminal nucleotides shifted relative to the deposition due to poorer electron density in the likely flexible region. This output model was then used in the newly developed automated carbohydrate model-building method, yielding an accurate model for 6 of the 11 deposited carbohydrates. Only 7 of the 11 deposited carbohydrates were modelled with an RSCC of over 0.80, resulting in the method successfully modelling 6 of the 7 deposited well-fit carbohydrates, as shown in Figure 7.1g.

The majority of nucleic acids and carbohydrates in this structure were modelled using these two automated methods. A small amount of manual work is required to finalise

the structure, but the difficult stages of initial interpretation and modelling have been completed without manual effort. This example underscores the capability of the software packages developed in this PhD, demonstrating strong performance from a starting point of only protein, but lacking the sensitivity required to model nucleic acids or carbohydrates completely. Such intricacy may not be possible from rigorous automated methods, since the quality of the experimental data is almost always the limiting factor. While automated methods certainly help accelerate the process of structure solution, manual verification will almost always be required for the foreseeable future.



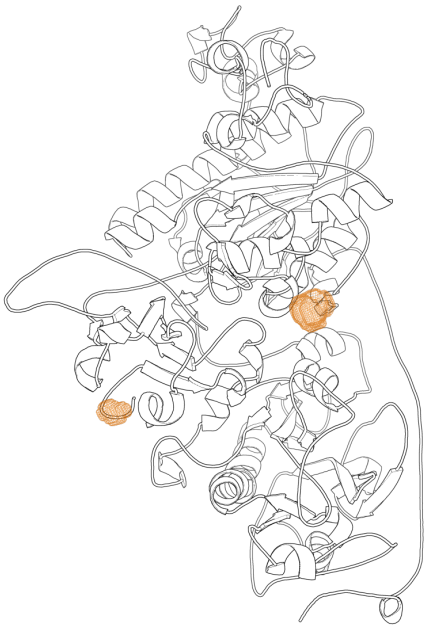
(a) Protein only model

(b) Predicted phosphate density

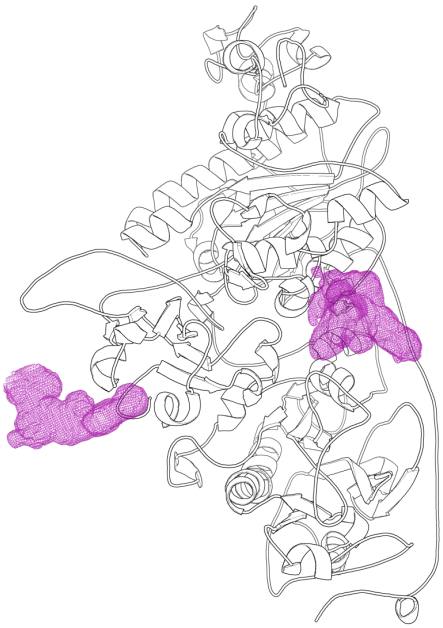
(c) Predicted sugar density

(d) Predicted base density

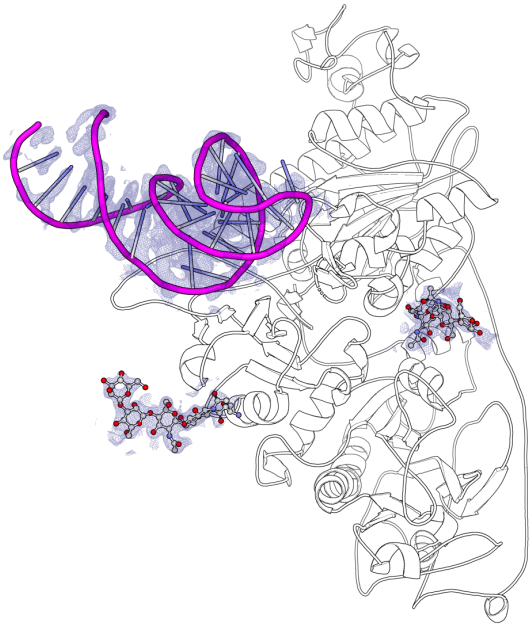
Figure 7.1: Automated model-building results for a mouse autotaxin in complex with an inhibiting DNA aptamer resolved at 2.00 Å resolution with X-ray crystallography (PDB code: 5HRT[326]). A - Protein-only model used as a starting point for automated model-building. B - Predicted phosphate density from *NucleoFind* after inputting an electron density map calculated using deposited structure factor amplitudes and the protein-only atomic model. C - Predicted sugar density from *NucleoFind* after inputting an electron density map calculated using deposited structure factor amplitudes and the protein-only atomic model. D - Predicted base density from *NucleoFind* after inputting an electron density map calculated using deposited structure factor amplitudes and the protein-only atomic model.



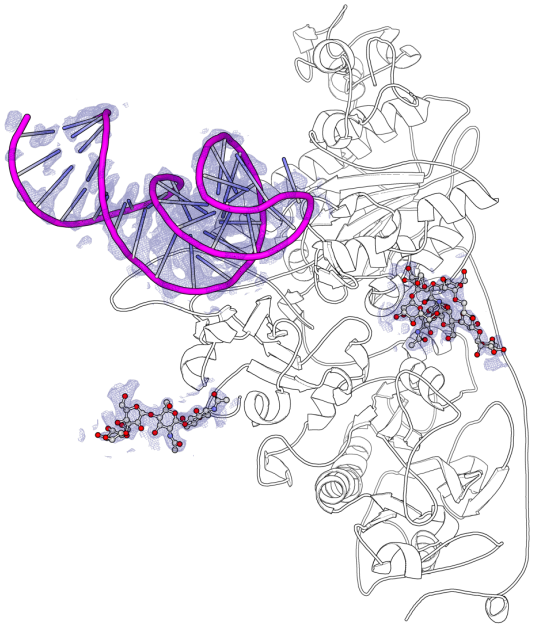
(e) Predicted carbohydrate donor density



(f) Predicted carbohydrate density



(g) Automatically built model



(h) Deposited model

Figure 7.1 (continued): E - Predicted carbohydrate protein donor density from the optimised multiclass segmentation model for carbohydrates after inputting an electron density map calculated using deposited structure factor amplitudes and the protein-only atomic model. F - Predicted carbohydrate density from the optimised multiclass segmentation model for carbohydrates after inputting an electron density map calculated using deposited structure factor amplitudes and the protein-only atomic model. G - Atomic model built after *ModelCraft* with *NucleoFind* and the automated carbohydrate model-building protocol with nucleic acids and carbohydrates highlighted. H - Deposited model with nucleic acids and carbohydrates highlighted.

7.3 Reflection

Aside from the technical and methodological challenges, perhaps the most troublesome aspect of this PhD has been testing new methods effectively, while achieving a balance between large-scale testing campaigns and in-depth individual examples. Individual test cases highlight the intricacies of the methodology, allowing strengths and weaknesses to be identified with a particular example. This is an essential aspect of method development, as testing on individual structures enables a method to progress. While useful, individual examples do not represent the full range of possible use cases for a method, so larger-scale tests must be conducted. Small changes in methodology may introduce significant differences in a single example, but averaging across an entire test set yields a more confident overall assessment of the performance of the method. Yet, it is impractical to comment on every aspect of the performance of every example in the test set, so general conclusions must be made. Throughout this thesis, attempts were made to balance these two testing methodologies, using case studies and test sets. In some instances, this balanced approach may not have fully captured the nuances of the technique under investigation, but compromises were consciously made for brevity.

Another consideration ever-present throughout the PhD process was the use of resources. Access to high-performance computers enabled the training of powerful deep learning models, but at almost every step, a cost-benefit analysis was performed to ensure the desired rigour could be achieved while minimising wasted computation, which costs both research time and real energy resources. The move to a high-performance computer run on 100 % renewable energy, alleviated a significant ethical barrier and allowed for the in-depth optimisation of the deep learning model described in Chapter 3. Overall, the energetic and environmental costs of this work were deemed acceptable, since developing powerful and accessible automated methods is likely to be a net positive for future scientific endeavours.

Bibliography

[1] I. Clark-Lewis, K.-S. Kim, K. Rajarathnam, J.-H. Gong, B. Dewald, B. Moser, M. Baggiolini and B. D. Sykes, *Journal of Leukocyte Biology*, 1995, **57**, 703–711.

[2] A. R. Fersht, R. J. Leatherbarrow and T. N. C. Wells, *Biochemistry*, 1987, **26**, 6030–6038.

[3] Committee on Research Opportunities in Biology Board on Biology Commission on Life Sciences, National Research Council (US), in *Opportunities in Biology*, National Academies Press (US), 1989.

[4] P. Sweeney, H. Park, M. Baumann, J. Dunlop, J. Frydman, R. Kopito, A. McCampbell, G. Leblanc, A. Venkateswaran, A. Nurmi and R. Hodgson, *Translational Neurodegeneration*, 2017, **6**, 6.

[5] T. K. Chaudhuri and S. Paul, *The FEBS Journal*, 2006, **273**, 1331–1349.

[6] F. U. Hartl, *Annual Review of Biochemistry*, 2017, **86**, 21–26.

[7] J. C. Kendrew, G. Bodo, H. M. Dintzis, R. G. Parrish, H. Wyckoff and D. C. Phillips, *Nature*, 1958, **181**, 662–666.

[8] J. L. S. Milne, M. J. Borgnia, A. Bartesaghi, E. E. H. Tran, L. A. Earl, D. M. Schauder, J. Lengyel, J. Pierson, A. Patwardhan and S. Subramaniam, *The FEBS Journal*, 2013, **280**, 28–45.

[9] W. J. Croft, *Under The Microscope: A Brief History Of Microscopy*, World Scientific, 2006.

[10] M. Jaskolski, Z. Dauter and A. Wlodawer, *The FEBS journal*, 2014, **281**, 3985–4009.

[11] D. Crowfoot, *Nature*, 1935, **135**, 591–592.

[12] C.-J. Chen, J. Rose, M. Newton, Z.-J. Liu and B.-C. Wang, in *Modern Protein Chemistry: Practical Aspects*, 2002, pp. 7–36.

[13] W. H. Bragg and W. L. Bragg, *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 1913, **88**, 428–438.

[14] M. S. Smyth and J. H. J. Martin, *Molecular Pathology*, 2000, **53**, 8–14.

[15] S. Galli, *Journal of Chemical Education*, 2014, **91**, 2009–2012.

[16] N. Huang, H. Deng, B. Liu, D. Wang and Z. Zhao, *The Innovation*, 2021, **2**, year.

[17] H. Kaufman and I. Fankuchen, *Analytical Chemistry*, 1949, **21**, 24–29.

[18] R. W. Pringle, *Nature*, 1950, **166**, 11–14.

[19] S. M. Gruner, M. W. Tate and E. F. Eikenberry, *Review of Scientific Instruments*, 2002, **73**, 2815–2842.

[20] A. Förster, S. Brandstetter and C. Schulze-Briese, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2019, **377**, 20180241.

[21] A. L. Patterson, *Physical Review*, 1934, **46**, 372–376.

[22] D. Blow, in *Outline of Crystallography for Biologists*, ed. D. Blow, Oxford University Press, 2002, p. 0.

[23] G. Friedel, *CR Acad. Sci. Paris*, 1913, **157**, 1533–1536.

[24] W. A. Hendrickson, J. L. Smith and S. Sheriff, in *Methods in Enzymology*, Academic Press, 1985, vol. 115 of Diffraction Methods for Biological Macromolecules Part B, pp. 41–55.

[25] W. A. Hendrickson and C. M. Ogata, in *Methods in Enzymology*, Academic Press, 1997, vol. 276 of Macromolecular Crystallography Part A, pp. 494–523.

[26] P. Argos and M. G. Rossman, in *Theory and Practice of Direct Methods in Crystallography*, ed. M. F. C. Ladd and R. A. Palmer, Springer US, Boston, MA, 1980, pp. 361–417.

[27] P. Evans and A. McCoy, *Acta Crystallographica Section D: Biological Crystallography*, 2008, **64**, 1–10.

[28] C. Abergel, *Acta Crystallographica Section D: Biological Crystallography*, 2013, **69**, 2167–2173.

[29] T. G. Flower and J. H. Hurley, *Protein Science : A Publication of the Protein Society*, 2021, **30**, 728–734.

[30] R. Engh and R. Huber, *International Tables for Crystallography, Vol. F, edited by MG Rossmann & E. Arnold*, 2001.

[31] R. J. Read, *Acta Crystallographica. Section D, Biological Crystallography*, 2001, **57**, 1373–1382.

[32] N. Qian and T. J. Sejnowski, *Journal of Molecular Biology*, 1988, **202**, 865–884.

[33] I. P. Crawford, T. Niermann and K. Kirschner, *Proteins: Structure, Function, and Bioinformatics*, 1987, **2**, 118–129.

[34] B. Rost and C. Sander, *Proceedings of the National Academy of Sciences*, 1993, **90**, 7558–7562.

[35] S. H. P. de Oliveira, J. Shi and C. M. Deane, *Bioinformatics*, 2017, **33**, 373–381.

[36] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli and D. Hassabis, *Nature*, 2021, **596**, 583–589.

[37] M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, C. Millán, H. Park, C. Adams, C. R. Glassman, A. DeGiovanni, J. H. Pereira, A. V. Rodrigues, A. A. van Dijk, A. C. Ebrecht, D. J. Opperman, T. Sagmeister, C. Buhlheller, T. Pavkov-Keller, M. K. Rathinaswamy, U. Dalwadi, C. K. Yip, J. E. Burke, K. C. Garcia, N. V. Grishin, P. D. Adams, R. J. Read and D. Baker, *Science*, 2021, **373**, 871–876.

[38] G. Ahdritz, N. Bouatta, C. Floristean, S. Kadyan, Q. Xia, W. Gerecke, T. J. O'Donnell, D. Berenberg, I. Fisk, N. Zanichelli, B. Zhang, A. Nowaczynski, B. Wang, M. M. Stepniewska-Dziubinska, S. Zhang, A. Ojewole, M. E. Guney, S. Biderman, A. M. Watkins, S. Ra, P. R. Lorenzo, L. Nivon, B. Weitzner, Y.-E. A. Ban, P. K. Sorger, E. Mostaque, Z. Zhang, R. Bonneau and M. AlQuraishi, *bioRxiv*, 2022.

[39] R. Evans, M. O'Neill, A. Pritzel, N. Antropova, A. Senior, T. Green, A. Žídek, R. Bates, S. Blackwell, J. Yim, O. Ronneberger, S. Bodenstein, M. Zielinski, A. Bridgland, A. Potapenko, A. Cowie, K. Tunyasuvunakool, R. Jain, E. Clancy, P. Kohli, J. Jumper and D. Hassabis, *Protein complex prediction with AlphaFold-Multimer*, 2022, `https://www.biorxiv.org/content/10.1101/2021.10.04.463034v2`, Pages: 2021.10.04.463034 Section: New Results.

[40] J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick, S. W. Bodenstein, D. A. Evans, C.-C. Hung, M. O'Neill, D. Reiman, K. Tunyasuvunakool, Z. Wu, A. Žemgulytė, E. Arvaniti, C. Beattie, O. Bertolli, A. Bridgland, A. Cherepanov, M. Congreve, A. I. Cowen-Rivers, A. Cowie, M. Figurnov, F. B. Fuchs, H. Gladman, R. Jain, Y. A. Khan, C. M. R. Low, K. Perlin, A. Potapenko, P. Savy, S. Singh, A. Stecula, A. Thillaisundaram, C. Tong, S. Yakneen, E. D. Zhong, M. Zielinski, A. Žídek, V. Bapst, P. Kohli, M. Jaderberg, D. Hassabis and J. M. Jumper, *Nature*, 2024, **630**, 493–500.

[41] C. Discovery, J. Boitreaud, J. Dent, M. McPartlon, J. Meier, V. Reis, A. Rogozhnikov and K. Wu, *bioRxiv*, 2024.

[42] J. Wohlwend, G. Corso, S. Passaro, N. Getz, M. Reveiz, K. Leidal, W. Swiderski, L. Atkinson, T. Portnoi, I. Chinn, J. Silterra, T. Jaakkola and R. Barzilay, *bioRxiv*, 2025.

[43] H. Berman, K. Henrick and H. Nakamura, *Nature Structural & Molecular Biology*, 2003, **10**, 980–980.

[44] C. Bernard, G. Postic, S. Ghannay and F. Tahi, *bioRxiv*, 2024.

[45] T. A. Jones, J.-Y. Zou, S. W. Cowan and M. Kjeldgaard, *Acta Crystallographica Section A: Foundations of Crystallography*, 1991, **47**, 110–119.

[46] P. Emsley and K. Cowtan, *Acta Crystallographica Section D: Biological Crystallography*, 2004, **60**, 2126–2132.

[47] A. M. Karmali, T. L. Blundell and N. Furnham, *Acta Crystallographica Section D: Biological Crystallography*, 2009, **65**, 121–127.

[48] P. Emsley, B. Lohkamp, W. G. Scott and K. Cowtan, *Acta Crystallographica Section D: Biological Crystallography*, 2010, **66**, 486–501.

[49] J. Debreczeni and P. Emsley, *Acta Crystallographica Section D: Biological Crystallography*, 2012, **68**, 425–430.

[50] P. Emsley, *Acta Crystallographica Section D*, 2017, **73**, 203–210.

[51] P. Emsley and M. Crispin, *Acta Crystallographica Section D: Structural Biology*, 2018, **74**, 256–263.

[52] A. Casañal, B. Lohkamp and P. Emsley, *Protein Science*, 2020, **29**, 1055–1064.

[53] J. Greer, *Journal of Molecular Biology*, 1974, **82**, 279–301.

[54] L. Holm and C. Sander, *Journal of Molecular Biology*, 1991, **218**, 183–194.

[55] T. Holton, T. R. Ioerger, J. A. Christopher and J. C. Sacchettini, *Acta Crystallographica Section D-Biological Crystallography*, 2000, **56**, 722–734.

[56] T. R. Ioerger and J. C. Sacchettini, in *Methods in Enzymology*, Academic Press, 2003, vol. 374 of Macromolecular Crystallography, Part D, pp. 244–270.

[57] T. A. Jones, M. Bergdoll and M. Kjeldgaard, in *Crystallographic and Modeling Methods in Molecular Design*, ed. C. E. Bugg and S. E. Ealick, Springer New York, New York, NY, 1990, pp. 189–199.

[58] D. Turk, *Acta Crystallographica Section D: Biological Crystallography*, 2013, **69**, 1342–1357.

[59] T. Oldfield, *Acta Crystallographica. Section D, Biological Crystallography*, 2002, **58**, 487–493.

[60] D. E. McRee, *Journal of Structural Biology*, 1999, **125**, 156–165.

[61] J. Agirre, M. Atanasova, H. Bagdonas, C. B. Ballard, A. Baslé, J. Beilsten-Edmands, R. J. Borges, D. G. Brown, J. J. Burgos-Mármol, J. M. Berrisford, P. S. Bond, I. Caballero, L. Catapano, G. Chojnowski, A. G. Cook, K. D. Cowtan, T. I. Croll, J. Debreczeni, N. E. Devenish, E. J. Dodson, T. R. Drevon, P. Emsley, G. Evans, P. R. Evans, M. Fando, J. Foadi, L. Fuentes-Montero, E. F. Garman, M. Gerstel, R. J. Gildea, K. Hatti, M. L. Hekkelman, P. Heuser, S. W. Hoh, M. A. Hough, H. T. Jenkins, E. Jiménez, R. P. Joosten, R. M. Keegan, N. Keep, E. B. Krissinel, P. Kolenko, O. Kovalevskiy, V. S. Lamzin, D. M. Lawson, A. A. Lebedev, A. G. W. Leslie, B. Lohkamp, F. Long, M. Malý, A. J. McCoy, S. J. McNicholas, A. Medina, C. Millán, J. W. Murray, G. N. Murshudov, R. A. Nicholls, M. E. M. Noble, R. Oeffner, N. S. Pannu, J. M. Parkhurst, N. Pearce, J. Pereira, A. Perrakis, H. R. Powell, R. J. Read, D. J. Rigden, W. Rochira, M. Sammito, F. Sánchez Rodríguez, G. M. Sheldrick, K. L. Shelley, F. Simkovic, A. J. Simpkin, P. Skubak, E. Sobolev, R. A. Steiner, K. Stevenson, I. Tews, J. M. H. Thomas, A. Thorn, J. T. Valls, V. Uski, I. Usón, A. Vagin, S. Velankar, M. Vollmar, H. Walden, D. Waterman, K. S. Wilson, M. D. Winn, G. Winter, M. Wojdyr and K. Yamashita, *Acta Crystallographica. Section D, Structural Biology*, 2023, **79**, 449–461.

[62] D. Liebschner, P. V. Afonine, M. L. Baker, G. Bunkóczi, V. B. Chen, T. I. Croll, B. Hintze, L.-W. Hung, S. Jain, A. J. McCoy, N. W. Moriarty, R. D. Oeffner, B. K. Poon, M. G. Prisant, R. J. Read, J. S. Richardson, D. C. Richardson, M. D. Sammito, O. V. Sobolev, D. H. Stockwell, T. C. Terwilliger, A. G. Urzhumtsev, L. L. Videau, C. J. Williams and P. D. Adams, *Acta Crystallographica Section D: Structural Biology*, 2019, **75**, 861–877.

[63] K. Cowtan, *Acta Crystallographica Section D: Biological Crystallography*, 2006, **62**, 1002–1011.

[64] K. Cowtan, *IUCrJ*, 2014, **1**, 387–392.

[65] T. C. Terwilliger, M. M. Los Alamos National Laboratory, O. C. R. Lawrence Berkeley National Laboratory and U. o. C. Department of Haematology, 2008.

[66] A. Perrakis, R. Morris and V. S. Lamzin, *Nature Structural Biology*, 1999, **6**, 458–463.

[67] K. Juneau, E. Podell, D. J. Harrington and T. R. Cech, *Structure*, 2001, **9**, 221–231.

[68] V. S. Lamzin and K. S. Wilson, *Acta Crystallographica Section D: Biological Crystallography*, 1993, **49**, 129–147.

[69] R. J. Morris, A. Perrakis and V. S. Lamzin, in *Methods in Enzymology*, Academic Press, 2003, vol. 374 of Macromolecular Crystallography, Part D, pp. 229–244.

[70] S. C. Lovell, J. M. Word, J. S. Richardson and D. C. Richardson, *Proteins: Structure, Function, and Bioinformatics*, 2000, **40**, 389–408.

[71] T. Wiegels and V. S. Lamzin, *Acta Crystallographica Section D: Biological Crystallography*, 2012, **68**, 446–453.

[72] G. Chojnowski, P. Heuser, J. Pereira and V. Lamzin, *Acta Crystallographica Section A Foundations and Advances*, 2018, **74**, e151–e151.

[73] G. X. Evrard, G. G. Langer, A. Perrakis and V. S. Lamzin, *Acta Crystallographica Section D: Biological Crystallography*, 2007, **63**, 108–117.

[74] J. Hattne and V. S. Lamzin, *Acta Crystallographica Section D: Biological Crystallography*, 2008, **64**, 834–842.

[75] T. C. Terwilliger, *Acta Crystallographica Section D: Biological Crystallography*, 2001, **57**, 1755–1762.

[76] T. C. Terwilliger, *Acta Crystallographica. Section D, Biological Crystallography*, 2003, **59**, 38–44.

[77] T. C. Terwilliger, in *Methods in Enzymology*, Academic Press, 2003, vol. 374 of Macromolecular Crystallography, Part D, pp. 22–37.

[78] T. Terwilliger, *Journal of Synchrotron Radiation*, 2004, **11**, 49–52.

[79] K. Cowtan, *Biological Crystallography*, 1998, **54**, 750–756.

[80] J. S. Richardson, B. Schneider, L. W. Murray, G. J. Kapral, R. M. Immormino, J. J. Headd, D. C. Richardson, D. Ham, E. Hershkovits, L. D. Williams, K. S. Keating, A. M. Pyle, D. Micallef, J. Westbrook and H. M. Berman, *Rna*, 2008, **14**, 465–481.

[81] C. L. Lawson, H. M. Berman, L. Chen, B. Vallat and C. L. Zirbel, *Nucleic Acids Research*, 2024, **52**, D245–D254.

[82] A. A. Vagin, R. A. Steiner, A. A. Lebedev, L. Potterton, S. McNicholas, F. Long and G. N. Murshudov, *Acta Crystallographica Section D: Biological Crystallography*, 2004, **60**, 2184–2195.

[83] K. Yamashita, M. Wojdyr, F. Long, R. A. Nicholls and G. N. Murshudov, *Acta Crystallographica Section D: Structural Biology*, 2023, **79**, 368–373.

[84] C. R. Groom, I. J. Bruno, M. P. Lightfoot and S. C. Ward, *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials*, 2016, **72**, 171–179.

[85] J. D. Westbrook, C. Shao, Z. Feng, M. Zhuravleva, S. Velankar and J. Young, *Bioinformatics*, 2015, **31**, 1274–1278.

[86] G. N. Ramachandran, C. Ramakrishnan and V. Sasisekharan, *Journal of Molecular Biology*, 1963, **7**, 95–99.

[87] R. A. Laskowski, M. W. MacArthur, D. S. Moss and J. M. Thornton, *Journal of Applied Crystallography*, 1993, **26**, 283–291.

[88] C. J. Williams, J. J. Headd, N. W. Moriarty, M. G. Prisant, L. L. Videau, L. N. Deis, V. Verma, D. A. Keedy, B. J. Hintze, V. B. Chen, S. Jain, S. M. Lewis, W. B. Arendall III, J. Snoeyink, P. D. Adams, S. C. Lovell, J. S. Richardson and D. C. Richardson, *Protein Science*, 2018, **27**, 293–315.

[89] J. Černý, P. Božíková, J. Svoboda and B. Schneider, *Nucleic Acids Research*, 2020, **48**, 6367–6381.

[90] G. Chojnowski, *Nucleic Acids Research*, 2023, **51**, 8255–8269.

[91] J. Agirre, J. Iglesias-Fernández, C. Rovira, G. J. Davies, K. S. Wilson and K. D. Cowtan, *Nature Structural & Molecular Biology*, 2015, **22**, 833–834.

[92] L. De Broglie, *Nature*, 1923, **112**, 540–540.

[93] R. M. Glaeser, *Journal of Ultrastructure Research*, 1971, **36**, 466–482.

[94] R. Glaeser, *JOURNAL DE MICROSCOPIE*, 1971, **12**, 133–+.

[95] K. A. Taylor and R. M. Glaeser, *Science*, 1974, **186**, 1036–1037.

[96] L. A. Passmore and C. J. Russo, *Methods in Enzymology*, 2016, **579**, 51–86.

[97] S. A. Fromm, K. M. O'Connor, M. Purdy, P. R. Bhatt, G. Loughran, J. F. Atkins, A. Jomaa and S. Mattei, *Nature Communications*, 2023, **14**, 1095.

[98] S. H. Scheres, *eLife*, 2014, **3**, e03665.

[99] R. K. Bryan, in *Maximum-Entropy and Bayesian Methods in Science and Engineering: Volume 2: Applications*, ed. G. J. Erickson and C. R. Smith, Springer Netherlands, Dordrecht, 1988, pp. 171–179.

[100] S. H. W. Scheres, *Journal of Structural Biology*, 2015, **189**, 114–122.

[101] A. Dhakal, R. Gyawali, L. Wang and J. Cheng, *Briefings in Bioinformatics*, 2025, **26**, bbaf011.

[102] S. H. Scheres, in *Methods in Enzymology*, ed. G. J. Jensen, Academic Press, 2010, vol. 482, pp. 295–320.

[103] A. E. Leschziner and E. Nogales, *Annual Review of Biophysics*, 2007, **36**, 43–62.

[104] A. Kucukelbir, F. J. Sigworth and H. D. Tagare, *Nature methods*, 2014, **11**, 63–65.

[105] S. H. W. Scheres, *Journal of Structural Biology*, 2012, **180**, 519–530.

[106] A. Punjani, J. L. Rubinstein, D. J. Fleet and M. A. Brubaker, *Nature Methods*, 2017, **14**, 290–296.

[107] K. Jamali, L. Käll, R. Zhang, A. Brown, D. Kimanius and S. H. W. Scheres, 2023, 2023.05.16.541002.

[108] J. Pfab, N. M. Phan and D. Si, *Proceedings of the National Academy of Sciences*, 2021, **118**, e2017525118.

[109] X. Wang, G. Terashi and D. Kihara, *Nature Methods*, 2023, 1–9.

[110] T. C. Terwilliger, O. V. Sobolev, P. V. Afonine and P. D. Adams, *Acta Crystallographica. Section D, Structural Biology*, 2018, **74**, 545–559.

[111] A. P. Joseph, I. Lagerstedt, A. Jakobi, T. Burnley, A. Patwardhan, M. Topf and M. Winn, *Journal of Chemical Information and Modeling*, 2020, **60**, 2552–2560.

[112] G. Pintilie and W. Chiu, *Acta Crystallographica. Section D, Structural Biology*, 2021, **77**, 1142–1152.

[113] I. Farabella, D. Vasishtan, A. P. Joseph, A. P. Pandurangan, H. Sahota and M. Topf, *Journal of Applied Crystallography*, 2015, **48**, 1314–1323.

[114] J. F. Miescher, *Ueber die chemische Zusammensetzung der Eiterzellen*, 1871.

[115] J. F. Miescher, *Die Spermatozoen einiger Wirbelthiere: ein Beitrag zur Histochemie*, 1874.

[116] E. Lamm, O. Harman and S. J. Veigl, *Genetics*, 2020, **215**, 291–296.

[117] A. Noma, Y. Kirino, Y. Ikeuchi and T. Suzuki, *The EMBO Journal*, 2006, **25**, 2142–2154.

[118] J. H.Gommers-Ampt and P. Borst, *The FASEB Journal*, 1995, **9**, 1034–1042.

[119] J. Creeth, *PhD Thesis*, University of London, 1948.

[120] C. Fonseca Guerra, F. M. Bickelhaupt, J. G. Snijders and E. J. Baerends, *Journal of the American Chemical Society*, 2000, **122**, 4117–4128.

[121] M. Sundaralingam, *Journal of the American Chemical Society*, 1965, **87**, 599–606.

[122] H. R. Wilson, A. Rahman and P. Tollin, *Journal of Molecular Biology*, 1969, **46**, 585–589.

[123] J. E. Sokoloski, S. A. Godfrey, S. E. Dombrowski and P. C. Bevilacqua, *RNA*, 2011, **17**, 1775–1787.

[124] V. N. Potaman and R. R. Sinden, in *Madame Curie Bioscience Database [Internet]*, Landes Bioscience, 2013.

[125] A. Rich and S. Zhang, *Nature Reviews Genetics*, 2003, **4**, 566–572.

[126] C. Bingman, S. Jain, G. Zon and M. Sundaralingam, *Nucleic Acids Research*, 1992, **20**, 6637–6647.

[127] S. Venkadesh, P. Mandal and N. Gautham, *Crystal structure of d(CCGGTACCGG)2 as B-DNA duplex grown with 5 mM CoCl2: 3r86*, 2011, `https://www.wwpdb.org/pdb?id=pdb_00003r86`, Institution: Worldwide Protein Data Bank.

[128] K. Brzezinski, A. Brzuszkiewicz, M. Dauter, M. Kubicki, M. Jaskolski and Z. Dauter, *Nucleic Acids Research*, 2011, **39**, 6238–6248.

[129] G. Wang and K. M. Vasquez, *DNA Repair*, 2014, **19**, 143–151.

[130] S. V. Kuznetsov, C.-C. Ren, S. A. Woodson and A. Ansari, *Nucleic Acids Research*, 2008, **36**, 1098–1112.

[131] V. Brázda, R. C. Laister, E. B. Jagelská and C. Arrowsmith, *BMC Molecular Biology*, 2011, **12**, 33.

[132] R. P. Bowater, N. Bohálová and V. Brázda, *International Journal of Molecular Sciences*, 2022, **23**, 6171.

[133] P. Svoboda and A. Di Cara, *Cellular and molecular life sciences: CMLS*, 2006, **63**, 901–908.

[134] Y. G. Chen and S. Hur, *Nature reviews. Molecular cell biology*, 2022, **23**, 286–301.

[135] P. Kerpedjiev, S. Hammer and I. L. Hofacker, *Bioinformatics*, 2015, **31**, 3377–3379.

[136] Y. Itoh, S. Chiba, S.-i. Sekine and S. Yokoyama, *Nucleic Acids Research*, 2009, **37**, 6259–6268.

[137] A. P. Nygaard and B. D. Hall, *Journal of Molecular Biology*, 1964, **9**, 125–142.

[138] N. M. Luscombe, S. E. Austin, H. M. Berman and J. M. Thornton, *Genome Biology*, 2000, **1**, reviews001.1–reviews001.37.

[139] D. E. Draper, *Journal of Molecular Biology*, 1999, **293**, 255–270.

[140] N. H. Hopcroft, A. Manfredo, A. L. Wendt, A. M. Brzozowski, P. Gollnick and A. A. Antson, *Journal of Molecular Biology*, 2004, **338**, 43–53.

[141] F. Gebauer, T. Schwarzl, J. Valcárcel and M. W. Hentze, *Nature Reviews Genetics*, 2021, **22**, 185–198.

[142] C. Maris, C. Dominguez and F. H.-T. Allain, *The FEBS journal*, 2005, **272**, 2118–2131.

[143] D. S. Wilson, B. Guenther, C. Desplan and J. Kuriyan, *Cell*, 1995, **82**, 709–719.

[144] S. Jones, D. T. A. Daley, N. M. Luscombe, H. M. Berman and J. M. Thornton, *Nucleic Acids Research*, 2001, **29**, 943–954.

[145] C. Hélène and G. Lancelot, *Progress in Biophysics and Molecular Biology*, 1982, **39**, 1–68.

[146] M. Egli, A. Flavell, A. M. Pyle, W. D. Wilson, S. I. Haq, B. Luisi, J. Fisher, C. Laughton, S. Allen, J. Engels, J. A. Grasby and S. Neidle, *Nucleic Acids in Chemistry and Biology*, The Royal Society of Chemistry, 2006.

[147] A. Hierro, J. M. Arizmendi, S. Bañuelos, A. Prado and A. Muga, *Biochemistry*, 2002, **41**, 6408–6413.

[148] F. Xiao, Z. Chen, Z. Wei and L. Tian, *Advanced Science*, 2020, **7**, 2001048.

[149] A. Mozo-Villarías, J. Cedano and E. Querol, *European Biophysics Journal*, 2021, **50**, 951–961.

[150] A. Ke and J. A. Doudna, *Methods*, 2004, **34**, 408–414.

[151] A. R. Ferré-D'Amaré, K. Zhou and J. A. Doudna, *Journal of Molecular Biology*, 1998, **279**, 621–631.

[152] M. Marcia, E. Humphris-Narayanan, K. S. Keating, S. Somarowthu, K. Rajashankar and A. M. Pyle, *Acta Crystallographica Section D: Biological Crystallography*, 2013, **69**, 2174–2185.

[153] R. W. Jackson, C. M. Smathers and A. R. Robart, *Molecules*, 2023, **28**, 2111.

[154] A. Y. Keel, R. P. Rambo, R. T. Batey and J. S. Kieft, *Structure*, 2007, **15**, 761–772.

[155] W. G. Scott, *Acta Crystallographica Section D: Biological Crystallography*, 2012, **68**, 441–445.

[156] *Essentials of Glycobiology*, ed. A. Varki, R. D. Cummings, J. D. Esko, H. H. Freeze, P. Stanley, C. R. Bertozzi, G. W. Hart and M. E. Etzler, Cold Spring Harbor Laboratory Press, Cold Spring Harbor (NY), 2nd edn., 2009.

[157] R. S. Shallenberger, T. E. Acree and C. Y. Lee, *Nature*, 1969, **221**, 555–556.

[158] K.-I. Sasajima and A. J. Sinskey, *Biochimica et Biophysica Acta (BBA) - Enzymology*, 1979, **571**, 120–126.

[159] F. Shafizadeh, in *Advances in Carbohydrate Chemistry*, ed. M. L. Wolfrom, Academic Press, 1958, vol. 13, pp. 9–61.

[160] S. Hanessian, *The Journal of Organic Chemistry*, 1969, **34**, 675–681.

[161] J. Boeyens, *Journal of Crystal and Molecular Structure*, 1978, **8**, 317–320.

[162] J. Agirre, G. Davies, K. Wilson and K. Cowtan, *Nature Chemical Biology*, 2015, **11**, 303–303.

[163] H. Yu and X. Chen, *Organic & Biomolecular Chemistry*, 2007, **5**, 865–872.

[164] J. Yang, D. Xie and X. Ma, *Molecules*, 2023, **28**, 4724.

[165] M. Mobli and A. Almond, *Organic & Biomolecular Chemistry*, 2007, **5**, 2243–2251.

[166] H. M. Kayili, N. Barlas, D. B. Demirhan, M. E. Yaman, M. Atakay, Güler, M. Kara, K. S. Tekgunduz and B. Salih, *Food Chemistry*, 2023, **421**, 136166.

[167] E. T. Sletten, G. Fittolani, N. Hribernik, M. C. S. Dal Colle, P. H. Seeberger and M. Delbianco, *ACS Central Science*, 2024, **10**, 138–142.

[168] V. Dimakos and M. S. Taylor, *Chemical Reviews*, 2018, **118**, 11457–11517.

[169] E. Staudacher, *Biological chemistry*, 2012, **393**, 675–685.

[170] A. Varki, R. D. Cummings, M. Aebi, N. H. Packer, P. H. Seeberger, J. D. Esko, P. Stanley, G. Hart, A. Darvill, T. Kinoshita, J. J. Prestegard, R. L. Schnaar, H. H. Freeze, J. D. Marth, C. R. Bertozzi, M. E. Etzler, M. Frank, J. F. Vliegenthart, T. Lütteke, S. Perez, E. Bolton, P. Rudd, J. Paulson, M. Kanehisa, P. Toukach, K. F. Aoki-Kinoshita, A. Dell, H. Narimatsu, W. York, N. Taniguchi and S. Kornfeld, *Glycobiology*, 2015, **25**, 1323–1324.

[171] S. J. Williams, in *Encyclopedia of Biophysics*, ed. G. C. K. Roberts, Springer, Berlin, Heidelberg, 2013, pp. 216–222.

[172] A. Payen, *Comptes rendus*, 1838, **7**, 1052–1056.

[173] R. Apweiler, H. Hermjakob and N. Sharon, *Biochimica et Biophysica Acta (BBA) - General Subjects*, 1999, **1473**, 4–8.

[174] A. Varki, *Glycobiology*, 1993, **3**, 97–130.

[175] R. A. Dwek, *Chemical Reviews*, 1996, **96**, 683–720.

[176] K. Ohtsubo and J. D. Marth, *Cell*, 2006, **126**, 855–867.

[177] A.-J. Petrescu, A.-L. Milac, S. M. Petrescu, R. A. Dwek and M. R. Wormald, *Glycobiology*, 2004, **14**, 103–114.

[178] S. C. Hubbard and R. J. Ivatt, *Annual Review of Biochemistry*, 1981, **50**, 555–583.

[179] M. A. Lehrman, *Glycobiology*, 1991, **1**, 553–562.

[180] F. Schwarz and M. Aebi, *Current Opinion in Structural Biology*, 2011, **21**, 576–582.

[181] E. C. Mandon, S. F. Trueman and R. Gilmore, *Cold Spring Harbor Perspectives in Biology*, 2013, **5**, a013342.

[182] D. J. Kelleher and R. Gilmore, *Glycobiology*, 2006, **16**, 47R–62R.

[183] C. Reily, T. J. Stewart, M. B. Renfrow and J. Novak, *Nature Reviews Nephrology*, 2019, **15**, 346–366.

[184] Y. Huang, C. Yang, X.-f. Xu, W. Xu and S.-w. Liu, *Acta Pharmacologica Sinica*, 2020, **41**, 1141–1149.

[185] K. Yamashita, H. Ideo, T. Ohkura, K. Fukushima, I. Yuasa, K. Ohno and K. Takeshita, *Journal of Biological Chemistry*, 1993, **268**, 5783–5789.

[186] J. Agirre, A. Ariza, W. A. Offen, J. P. Turkenburg, S. M. Roberts, S. McNicholas, P. V. Harris, B. McBrayer, J. Dohnalek, K. D. Cowtan, G. J. Davies and K. S. Wilson, *Acta Crystallographica Section D: Structural Biology*, 2016, **72**, 254–265.

[187] M. Pancera, S. Shahzad-ul Hussan, N. A. Doria-Rose, J. S. McLellan, R. T. Bailer, K. Dai, S. Loesgen, M. K. Louder, R. P. Staupe, Y. Yang, B. Zhang, R. Parks, J. Eudailey, K. E. Lloyd, J. Blinn, S. M. Alam, B. F. Haynes, M. N. Amin, L.-X. Wang, D. R. Burton, W. C. Koff, G. J. Nabel, J. R. Mascola, C. A. Bewley and P. D. Kwong, *Nature Structural & Molecular Biology*, 2013, **20**, 804–813.

[188] N. Noinaj, N. C. Easley, M. Oke, N. Mizuno, J. Gumbart, E. Boura, A. N. Steere, O. Zak, P. Aisen, E. Tajkhorshid, R. W. Evans, A. R. Gorringe, A. B. Mason, A. C. Steven and S. K. Buchanan, *Nature*, 2012, **483**, 53–58.

[189] G. J. Gerwig, in *The Art of Carbohydrate Analysis*, ed. G. J. Gerwig, Springer International Publishing, Cham, 2021, pp. 11–50.

[190] A. M. Boyce, A. E. Lee, K. L. Roszko and R. I. Gafni, *Frontiers in Endocrinology*, 2020, **11**, year.

[191] J. Hofsteenge, D. R. Mueller, T. De Beer, A. Loeffler, W. J. Richter and J. F. G. Vliegenthart, *Biochemistry*, 1994, **33**, 13524–13530.

[192] A. Shcherbakova, M. Preller, M. H. Taft, J. Pujols, S. Ventura, B. Tiemann, F. F. Buettner and H. Bakker, *eLife*, 2019, **8**, e52978.

[193] S. L. Crine and K. R. Acharya, *The FEBS Journal*, 2022, **289**, 7670–7687.

[194] S. Minakata, S. Manabe, Y. Inai, M. Ikezaki, K. Nishitsuji, Y. Ito and Y. Ihara, *Molecules*, 2021, **26**, 5258.

[195] W. Curatolo, *Biochimica et Biophysica Acta (BBA) - Reviews on Biomembranes*, 1987, **906**, 137–160.

[196] I. H. Erb, *Canadian Medical Association Journal*, 1940, **42**, 418–421.

[197] H. Wiegandt, *Glycolipids*, Elsevier, 2011.

[198] R. A. Flynn, K. Pedram, S. A. Malaker, P. J. Batista, B. A. H. Smith, A. G. Johnson, B. M. George, K. Majzoub, P. W. Villalta, J. E. Carette and C. R. Bertozzi, *Cell*, 2021, **184**, 3109–3124.e22.

[199] I. J. Chang, M. He and C. T. Lam, *Annals of Translational Medicine*, 2018, **6**, 477.

[200] N. Valkov and S. Das, *Advances in experimental medicine and biology*, 2020, **1229**, 327–342.

[201] J. S. Dialpuri, H. Bagdonas, L. C. Schofield, P. T. Pham, L. Holland and J. Agirre, *Beilstein Journal of Organic Chemistry*, 2024, **20**, 931–939.

[202] W. B. Struwe and C. V. Robinson, *Current opinion in structural biology*, 2019, **58**, 241–248.

[203] J. H. Prestegard, *The Journal of Biological Chemistry*, 2021, **296**, 100556.

[204] A. L. Tarentino, R. B. Trimble and T. H. Plummer, *Methods in Cell Biology*, 1989, **32**, 111–139.

[205] M. Atanasova, R. A. Nicholls, R. P. Joosten and J. Agirre, *Acta Crystallographica Section D: Structural Biology*, 2022, **78**, 455–465.

[206] R. P. Joosten, F. Long, G. N. Murshudov and A. Perrakis, *IUCrJ*, 2014, **1**, 213–220.

[207] F. Galton, *The Journal of the Anthropological Institute of Great Britain and Ireland*, 1886, **15**, 246–263.

[208] J. F. Kenney and E. Keeping, *Mathematics of statistics*, 1962, **1**, 252–285.

[209] T. Zhang, Proceedings of the twenty-first international conference on Machine learning, New York, NY, USA, 2004, p. 116.

[210] D. R. Cox, *Journal of the Royal Statistical Society: Series B (Methodological)*, 1958, **20**, 215–232.

[211] L. Boltzmann, *Studien über das Gleichgewicht der lebendigen Kraft zwischen bewegten materiellen Punkten*, 1868.

[212] W. S. McCulloch and W. Pitts, *The bulletin of mathematical biophysics*, 1943, **5**, 115–133.

[213] D. E. Rumelhart, G. E. Hinton and R. J. Williams, *Nature*, 1986, **323**, 533–536.

[214] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.

[215] Y. LeCun, Y. Bengio and G. Hinton, *Nature*, 2015, **521**, 436–444.

[216] A. Krizhevsky, I. Sutskever and G. E. Hinton, Advances in Neural Information Processing Systems, 2012.

[217] X. Jin and J. Han, in *Encyclopedia of Machine Learning*, ed. C. Sammut and G. I. Webb, Springer US, Boston, MA, 2010, pp. 563–564.

[218] M. Ester, H.-P. Kriegel, J. Sander, X. Xu and others, kdd, 1996, pp. 226–231.

[219] K. Pearson, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1901, **2**, 559–572.

[220] E. A. Feigenbaum, R. S. Engelmore and C. K. Johnson, *Acta Crystallographica Section A*, 1977, **33**, 13–18.

[221] T. R. Ioerger and J. C. Sacchettini, *Acta Crystallographica Section D: Biological Crystallography*, 2002, **58**, 2043–2054.

[222] G. Chojnowski, J. Pereira and V. S. Lamzin, *Acta Crystallographica Section D: Structural Biology*, 2019, **75**, 753–763.

[223] P. S. Bond, K. S. Wilson and K. D. Cowtan, *Acta Crystallographica Section D: Structural Biology*, 2020, **76**, 713–723.

[224] J. A. Aguiar, M. L. Gong and T. Tasdizen, *Computational Materials Science*, 2020, **173**, 109409.

[225] H. Yanxon, J. Weng, H. Parraga, W. Xu, U. Ruett and N. Schwarz, *Journal of Synchrotron Radiation*, 2023, **30**, 137–146.

[226] V.-A. Surdu and R. Győrgy, *Applied Sciences*, 2023, **13**, 9992.

[227] V. Rahmani, S. Nawaz, D. Pennicard, S. P. R. Setty and H. Graafsma, *Journal of applied crystallography*, 2023, **56**, 200–213.

[228] J. Schurmann, I. Lindhè, J. W. Janneck, G. Lima and Z. Matej, 2019 53rd Asilomar Conference on Signals, Systems, and Computers, 2019, pp. 978–983.

[229] J. Dialpuri, J. Agirre, K. Cowtan and P. Bond, *Nucleic Acids Research*, 2024, **52**, e84.

[230] G. G. Langer, S. X. Cohen, V. S. Lamzin and A. Perrakis, *Nature protocols*, 2008, **3**, 1171–1179.

[231] O. Ronneberger, P. Fischer and T. Brox, 2015.

[232] Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox and O. Ronneberger, 2016.

[233] P. Mostosi, H. Schindelin, P. Kollmannsberger and A. Thorn, *Angewandte Chemie International Edition*, 2020, **59**, 14788–14795.

[234] Á. Godó, K. Aoki, A. Nakagawa and Y. Yagi, *IEEE Access*, 2022, **10**, 28760–28772.

[235] A. J. Simpkin, J. M. H. Thomas, R. M. Keegan and D. J. Rigden, *MrParse: Finding homologues in the PDB and the EBI AlphaFold database for Molecular Replacement and more*, 2021, `https://www.biorxiv.org/content/10.1101/2021.09.02.458604v1`, Pages: 2021.09.02.458604 Section: New Results.

[236] A. J. Simpkin, L. G. Elliott, K. Stevenson, E. Krissinel, D. J. Rigden and R. M. Keegan, *Slice'N'Dice: Maximising the value of predicted models for structural biologists*, 2022, `https://www.biorxiv.org/content/10.1101/2022.06.30.497974v1`, Pages: 2022.06.30.497974 Section: New Results.

[237] A. J. McCoy, R. W. Grosse-Kunstleve, P. D. Adams, M. D. Winn, L. C. Storoni and R. J. Read, *Journal of Applied Crystallography*, 2007, **40**, 658–674.

[238] D. Ulyanov, A. Vedaldi and V. Lempitsky, 2017.

[239] S. K. Burley, C. Bhikadiya, C. Bi, S. Bittrich, H. Chao, L. Chen, P. A. Craig, G. V. Crichlow, K. Dalenberg, J. M. Duarte, S. Dutta, M. Fayazi, Z. Feng, J. W. Flatt, S. Ganesan, S. Ghosh, D. S. Goodsell, R. K. Green, V. Guranovic, J. Henry, B. P. Hudson, I. Khokhriakov, C. L. Lawson, Y. Liang, R. Lowe, E. Peisach, I. Persikova, D. W. Piehl, Y. Rose, A. Sali, J. Segura, M. Sekharan, C. Shao, B. Vallat, M. Voigt, B. Webb, J. D. Westbrook, S. Whetstone, J. Y. Young, A. Zalevsky and C. Zardecki, *Nucleic Acids Research*, 2023, **51**, D488–D508.

[240] H. Yang, E. Peisach, J. D. Westbrook, J. Young, H. M. Berman and S. K. Burley, *Journal of Applied Crystallography*, 2016, **49**, 1081–1084.

[241] G. N. Murshudov, P. Skubák, A. A. Lebedev, N. S. Pannu, R. A. Steiner, R. A. Nicholls, M. D. Winn, F. Long and A. A. Vagin, *Acta Crystallographica Section D: Biological Crystallography*, 2011, **67**, 355–367.

[242] M. Wojdyr, *Journal of Open Source Software*, 2022, **7**, 4200.

[243] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu and Xiaoqiang Zheng, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015, `https://www.tensorflow.org/`.

[244] T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, 2018.

[245] A. Reményi, A. Tomilin, E. Pohl, K. Lins, A. Philippsen, R. Reinbold, H. R. Schöler and M. Wilmanns, *Molecular Cell*, 2001, **8**, 569–580.

[246] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts and P. Walter, *Molecular Biology of the Cell*, Garland Science, 4th edn., 2002.

[247] D. Esch, J. Vahokoski, M. R. Groves, V. Pogenberg, V. Cojocaru, H. vom Bruch, D. Han, H. C. A. Drexler, M. J. Araúzo-Bravo, C. K. L. Ng, R. Jauch, M. Wilmanns and H. R. Schöler, *Nature Cell Biology*, 2013, **15**, 295–301.

[248] M. Beckers, D. Mann and C. Sachse, *Progress in Biophysics and Molecular Biology*, 2021, **160**, 26–36.

[249] The wwPDB Consortium, *Nucleic Acids Research*, 2024, **52**, D456–D465.

[250] K. O'Shea and R. Nash, *An Introduction to Convolutional Neural Networks*, 2015, `http://arxiv.org/abs/1511.08458`, arXiv:1511.08458 [cs].

[251] S. S. SHAPIRO and M. B. WILK, *Biometrika*, 1965, **52**, 591–611.

[252] D. Rey and M. Neuhäuser, in *International Encyclopedia of Statistical Science*, ed. M. Lovric, Springer, Berlin, Heidelberg, 2011, pp. 1658–1659.

[253] O. Rainio, J. Teuho and R. Klén, *Scientific Reports*, 2024, **14**, 6086.

[254] S. Neidle and M. Sanderson, *Principles of Nucleic Acid Structure*, Academic Press, 2021.

[255] S. Mahmud, N. Ibtehaz, A. Khandakar, A. Tahir, T. Rahman, K. R. Islam, M. S. Hossain, M. S. Rahman, M. T. Islam and M. E. H. Chowdhury, *Sensors*, 2022, **22**, 919.

[256] S. Z. Li, M. G. French, K. M. Pavlov and H. T. Li, Developments in X-Ray Tomography XIV, 2022, pp. 393–404.

[257] P. V. Afonine, B. P. Klaholz, N. W. Moriarty, B. K. Poon, O. V. Sobolev, T. C. Terwilliger, P. D. Adams and A. Urzhumtsev, *Acta Crystallographica. Section D, Structural Biology*, 2018, **74**, 814–840.

[258] T. Neudegger, J. Verghese, M. Hayer-Hartl, F. U. Hartl and A. Bracher, *Nature Structural & Molecular Biology*, 2016, **23**, 140–146.

[259] J. A. Nelder and R. Mead, *The Computer Journal*, 1965, **7**, 308–313.

[260] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*, Courier Corporation, 1965.

[261] E. Alharbi, R. Calinescu and K. Cowtan, *Acta Crystallographica Section D: Structural Biology*, 2023, **79**, 326–338.

[262] F. DiMaio, J. Shavlik and G. N. Phillips, *Bioinformatics*, 2006, **22**, e81–e89.

[263] E. Dodson, *Acta Crystallographica Section D: Biological Crystallography*, 2008, **64**, 17–24.

[264] P. S. Bond and K. D. Cowtan, *Acta Crystallographica. Section D, Structural Biology*, 2022, **78**, 1090–1098.

[265] T.-C. Liu, C.-T. Lin, K.-C. Chang, K.-W. Guo, S. Wang, J.-W. Chu and Y.-Y. Hsiao, *Nature Communications*, 2021, **12**, 601.

[266] V. Ramakrishnan, *Cell*, 2002, **108**, 557–572.

[267] J. A. Doudna, *Current biology: CB*, 1999, **9**, R731–734.

[268] J. M. Ogle, D. E. Brodersen, W. M. Clemons, M. J. Tarry, A. P. Carter and V. Ramakrishnan, *Science*, 2001, **292**, 897–902.

[269] S. Kurata, A. Weixlbaumer, T. Ohtsuki, T. Shimazaki, T. Wada, Y. Kirino, K. Takai, K. Watanabe, V. Ramakrishnan and T. Suzuki, *Journal of Biological Chemistry*, 2008, **283**, 18801–18811.

[270] R. M. Keegan and M. D. Winn, *Acta Crystallographica Section D: Biological Crystallography*, 2007, **63**, 447–457.

[271] J. D. Dinman, *Microbe (Washington, D.C.)*, 2006, **1**, 521–527.

[272] T. Jacks and H. E. Varmus, *Science*, 1985, **230**, 1237–1242.

[273] C. P. Jones and A. R. Ferré-D'Amaré, *Proceedings of the National Academy of Sciences*, 2025, **122**, e2418418122.

[274] G. M. Sheldrick, *Acta Crystallographica Section A: Foundations of Crystallography*, 2008, **64**, 112–122.

[275] K. Cowtan, *Acta Crystallographica Section D: Biological Crystallography*, 2010, **66**, 470–478.

[276] K. Nemeth, R. Bayraktar, M. Ferracin and G. A. Calin, *Nature Reviews Genetics*, 2024, **25**, 211–232.

[277] Z. Weinberg, J. Perreault, M. M. Meyer and R. R. Breaker, *Nature*, 2009, **462**, 656–659.

[278] R. C. Kretsch, Y. Wu, S. A. Shabalina, H. Lee, G. Nye, E. V. Koonin, A. Gao, W. Chiu and R. Das, *Nature*, 2025, **643**, 1135–1142.

[279] J. S. Dialpuri, H. Bagdonas, M. Atanasova, L. C. Schofield, M. L. Hekkelman, R. P. Joosten and J. Agirre, *Acta Crystallographica Section D: Structural Biology*, 2023, **79**, 462–472.

[280] J. S. Dialpuri, H. Bagdonas, L. C. Schofield, P. T. Pham, L. Holland, P. S. Bond, F. Sánchez Rodríguez, S. J. McNicholas and J. Agirre, *Acta Crystallographica. Section F, Structural Biology Communications*, 2024.

[281] F. Xin and P. Radivojac, *Bioinformatics*, 2012, **28**, 2905–2913.

[282] L. Casalino, Z. Gaieb, J. A. Goldsmith, C. K. Hjorth, A. C. Dommer, A. M. Harbison, C. A. Fogarty, E. P. Barros, B. C. Taylor, J. S. McLellan, E. Fadda and R. E. Amaro, *ACS Central Science*, 2020, **6**, 1722–1734.

[283] N. L. Miller, T. Clark, R. Raman and R. Sasisekharan, *Frontiers in Molecular Biosciences*, 2021, **8**, 666756.

[284] T. Lütteke, M. Frank and C.-W. von der Lieth, *Nucleic Acids Research*, 2005, **33**, D242–D246.

[285] D. Cremer and J. Pople, *Journal of the American Chemical Society.*, 1975, **97**, year.

[286] A. Clauset, C. R. Shalizi and M. E. J. Newman, *SIAM Review*, 2009, **51**, 661–703.

[287] A. Imberty and S. Pérez, *Protein Engineering, Design and Selection*, 1995, **8**, 699–709.

[288] M. J. Law, M. E. Linde, E. J. Chambers, C. Oubridge, P. S. Katsamba, L. Nilsson, I. S. Haworth and I. A. Laird-Offringa, *Nucleic Acids Research*, 2006, **34**, 275–285.

[289] R. E. Hubbard and M. K. Haider, 2010.

[290] J. She, Z. Han, B. Zhou and J. Chai, *Protein & Cell*, 2013, **4**, 475–482.

[291] T. Lütteke, *Acta Crystallographica Section D: Biological Crystallography*, 2009, **65**, 156–168.

[292] R. J. Woods and M. B. Tessier, *Current Opinion in Structural Biology*, 2010, **20**, 575–583.

[293] V. Lutsyk and W. Plazinski, *The Journal of Physical Chemistry B*, 2021, **125**, 10900–10916.

[294] S. A. Samsonov, S. Theisgen, T. Riemer, D. Huster and M. T. Pisabarro, *BioMed Research International*, 2014, **2014**, 808071.

[295] W. Kohn and L. J. Sham, *Physical Review*, 1965, **140**, A1133–A1138.

[296] M. e. Frisch, G. Trucks, H. B. Schlegel, G. Scuseria, M. Robb, J. Cheeseman, G. Scalmani, V. Barone, G. Petersson, H. Nakatsuji and others, *Gaussian 16*, 2016.

[297] B. van Beusekom, K. Joosten, M. L. Hekkelman, R. P. Joosten and A. Perrakis, *IUCrJ*, 2018, **5**, 585–594.

[298] R. W. Hooft, C. Sander and G. Vriend, *Bioinformatics*, 1997, **13**, 425–430.

[299] J. Agirre, G. J. Davies, K. S. Wilson and K. D. Cowtan, *Current Opinion in Structural Biology*, 2017, **44**, 39–47.

[300] B. van Beusekom, N. Wezel, M. L. Hekkelman, A. Perrakis, P. Emsley and R. P. Joosten, *Acta Crystallographica. Section D, Structural Biology*, 2019, **75**, 416–425.

[301] L. Holland, P. T. Pham, H. Bagdonas, J. S. Dialpuri, L. C. Schofield and J. Agirre, *Protein Science*, 2025, **34**, e70025.

[302] E. T. Prates, X. Guan, Y. Li, X. Wang, P. K. Chaffey, M. S. Skaf, M. F. Crowley, Z. Tan and G. T. Beckham, *Chemical Science*, 2018, **9**, 3710–3715.

[303] L. Han and C. E. Costello, *Biochemistry. Biokhimiia*, 2013, **78**, 710–720.

[304] M. Atanasova and J. Agirre, *Acta Cryst*, 2021, **77**, C775.

[305] wwPDB consortium, *Nucleic Acids Research*, 2019, **47**, D520–D528.

[306] M. H. Dietrich, J. Chmielewski, L.-J. Chan, L. L. Tan, A. Adair, F. M. T. Lyons, M. Gabriela, S. Lopaticki, T. A. Dite, L. F. Dagley, L. Pazzagli, P. Gupta, M. Kamil, A. M. Vaughan, R. Rojrung, A. Abraham, R. Mazhari, R. J. Longley, K. Zeglinski, Q. Gouil, I. Mueller, S. A. Fabb, R. Shandre-Mugan, C. W. Pouton, A. Glukhova, S. Shakeel and W.-H. Tham, *Science*, 2025, **389**, eady0241.

[307] O. C. Grant, D. Wentworth, S. G. Holmes, R. Kandel, D. Sehnal, X. Wang, Y. Xiao, P. Sheppard, T. Grelsson, A. Coulter, G. Miller, B. L. Foley and R. J. Woods, *Generating 3D Models of Carbohydrates with GLYCAM-Web*, 2025, `https://www.biorxiv.org/content/10.1101/2025.05.08.652828v1`, Pages: 2025.05.08.652828 Section: New Results.

[308] G. Pintilie, K. Zhang, Z. Su, S. Li, M. F. Schmid and W. Chiu, *Nature methods*, 2020, **17**, 328–334.

[309] G. Pintilie, C. Shao, Z. Wang, B. P. Hudson, J. W. Flatt, M. F. Schmid, K. L. Morris, S. K. Burley and W. Chiu, *Acta Crystallographica Section D: Structural Biology*, 2025, **81**, 410–422.

[310] U. Leitgeb, R. Crha, I. Fegerl, P. G. Furtmüller, C. Oostenbrink and V. Pfanzagl, *International Journal of Biological Macromolecules*, 2025, **330**, 148038.

[311] J. Stetefeld, M. McDougall, P. Loewen, A. Moya and M. Meier, *Structural elucidation of the Ectodomain of mouse UNC5H2: 6ool*, 2020, https://www.wwpdb.org/pdb?id=pdb_00006ool, Institution: Worldwide Protein Data Bank.

[312] E. Jiménez-Ortega, E. Narmontaite, B. González-Pérez, F. J. Plou, M. Fernández-Lobato and J. Sanz-Aparicio, *International Journal of Molecular Sciences*, 2022, **23**, 14981.

[313] N. C. Wu and I. A. Wilson, *Cold Spring Harbor Perspectives in Medicine*, 2020, **10**, a038778.

[314] J. A. Pulit-Penaloza, N. Simpson, H. Yang, H. M. Creager, J. Jones, P. Carney, J. A. Belser, G. Yang, J. Chang, H. Zeng, S. Thor, Y. Jang, M. L. Killian, M. Jenkins-Moore, A. Janas-Martindale, E. Dubovi, D. E. Wentworth, J. Stevens, T. M. Tumpey, C. T. Davis and T. R. Maines, *The Journal of Infectious Diseases*, 2017, **216**, S499–S507.

[315] G. E. P. Box, in *Robustness in Statistics*, ed. R. L. Launer and G. N. Wilkinson, Academic Press, 1979, pp. 201–236.

[316] O. Svensson, S. Malbet-Monaco, A. Popov, D. Nurizzo and M. W. Bowler, *Acta Crystallographica Section D: Biological Crystallography*, 2015, **71**, 1757–1767.

[317] S. Arzt, A. Beteva, F. Cipriani, S. Delageniere, F. Felisaz, G. Förstner, E. Gordon, L. Launer, B. Lavault, G. Leonard, T. Mairs, A. McCarthy, J. McCarthy, S. McSweeney, J. Meyer, E. Mitchell, S. Monaco, D. Nurizzo, R. Ravelli, V. Rey, W. Shepard, D. Spruce, O. Svensson and P. Theveneau, *Progress in Biophysics and Molecular Biology*, 2005, **89**, 124–152.

[318] G. Winter, D. G. Waterman, J. M. Parkhurst, A. S. Brewster, R. J. Gildea, M. Gerstel, L. Fuentes-Montero, M. Vollmar, T. Michels-Clark, I. D. Young, N. K. Sauter and G. Evans, *Acta Crystallographica Section D: Structural Biology*, 2018, **74**, 85–97.

[319] P. Skubák, *Acta Crystallographica Section D: Structural Biology*, 2024, **80**, 528–534.

[320] D. McDonagh, D. J. Rigden, D. G. Waterman and R. M. Keegan, *Predicting Protein Crystal Solvent Content from Patterson Maps Using Machine Learning*, 2025, https://www.biorxiv.org/content/10.1101/2025.09.24.678396v1, ISSN: 2692-8205 Pages: 2025.09.24.678396 Section: New Results.

[321] D. Liu, *Deep learning for Electron Density Map Sharpening*, 2025, https://rs-station.github.io/2025/03/21/deep-learning-for-electron-density-map-sharpening.html.

[322] A. S. Larsen, T. Rekis and A. Madsen, *Science*, 2024, **385**, 522–528.

[323] T. Simonnet, S. Grangeon, F. Claret, N. Maubec, M. D. Fall, R. Harba and B. Galerne, *IUCrJ*, 2024, **11**, 859–870.

[324] T. Pan, E. Dramko, M. D. Miller, A. Kyrillidis and G. N. Phillips, *Acta Crystallographica Section D: Structural Biology*, 2025, **81**, 668–677.

[325] C. Dun, Q. Pan, S. Jin, R. Stevens, M. D. Miller, G. N. Phillips and A. Kyrillidis, *CrysFormer: Protein Structure Prediction via 3d Patterson Maps and Partial Structure Attention*, 2023, http://arxiv.org/abs/2310.03899, arXiv:2310.03899 [cs].

[326] A. Perrakis and W. H. Moolenaar, *Journal of Lipid Research*, 2014, **55**, 1010–1018.

[327] L. McInnes, J. Healy and S. Astels, *Journal of Open Source Software*, 2017, **2**, 205.

Part III

Supplementary Information

8.1 Pairwise Statistics for Nucleic Acid Segmentation Deep Learning Models

Table 8.1: Atom inclusion statistics calculated by comparing three binary segmentation models, the baseline multiclass segmentation model and the optimised multiclass segmentation model. Pairwise significance estimations were completed with a Wilcoxon signed-rank test. Significance levels are represented by stars where $P \geq 0.05$ = n.s., $0.01 \leq P < 0.05 = *$, $0.001 \leq P < 0.01 = **$, $P < 0.001 = ***$. Delta values are the difference between the average performance of both models for a given output, expressed in percentage points (pp).

(a) X-ray diffraction molecular replacement test set

Model		Output	**Atom Inclusion**		
Ref.	Comp.		P-value	Sig.	Delta / pp
Binary segmentation	**Baseline multiclass segmentation**	Phosphate	0.00	***	-14.4
		Sugar	0.00	***	-20.0
		Base	0.00	***	-19.3
Binary segmentation	**Optimised multiclass segmentation**	Phosphate	0.32	n.s.	0.4
		Sugar	0.16	n.s.	-2.5
		Base	0.06	n.s.	-3.4
Baseline multiclass segmentation	**Optimised multiclass segmentation**	Phosphate	0.00	***	14.8
		Sugar	0.00	***	17.4
		Base	0.00	***	15.9

(b) Cryo-EM Coulomb potential map test set

Model		Output	**Atom Inclusion**		
Ref.	Comp.		P-value	Sig.	Delta / pp
Binary segmentation	**Baseline multiclass segmentation**	Phosphate	0.73	n.s.	-0.6
		Sugar	0.01	*	-14.8
		Base	0.01	*	-17.1
Binary segmentation	**Optimised multiclass segmentation**	Phosphate	0.00	***	23.5
		Sugar	0.00	**	19.0
		Base	0.43	n.s.	4.9
Baseline multiclass segmentation	**Optimised multiclass segmentation**	Phosphate	0.00	***	24.1
		Sugar	0.00	***	33.8
		Base	0.01	***	22.0

Table 8.2: Precision statistics calculated by comparing three binary segmentation models, the baseline multiclass segmentation model and the optimised multiclass segmentation model. Pairwise significance estimations were completed with a Wilcoxon signed-rank test. Significance levels are represented by stars where $P \geq 0.05 =$ n.s., $0.01 \leq P < 0.05 = *$, $0.001 \leq P < 0.01 = **$, $P < 0.001 = ***$. Delta values are the difference between the average performance of both models for a given output, expressed in percentage points (pp).

(a) X-ray diffraction molecular replacement test set

Model		Output	**Precision**		
Ref.	Comp.		P-value	Sig.	Delta / pp
Binary segmentation	**Baseline multiclass segmentation**	Phosphate	0.00	***	11.5
		Sugar	0.00	***	14.7
		Base	0.00	***	13.3
Binary segmentation	**Optimised multiclass segmentation**	Phosphate	0.00	***	12.6
		Sugar	0.00	***	13.2
		Base	0.00	***	12.5
Baseline multiclass segmentation	**Optimised multiclass segmentation**	Phosphate	0.00	***	1.1
		Sugar	0.00	***	-1.4
		Base	0.00	***	-0.8

(b) Cryo-EM Coulomb potential map test set

Model		Output	**Precision**		
Ref.	Comp.		P-value	Sig.	Delta / pp
Binary segmentation	**Baseline multiclass segmentation**	Phosphate	0.00	***	10.1
		Sugar	0.00	**	17.3
		Base	0.00	***	16.8
Binary segmentation	**Optimised multiclass segmentation**	Phosphate	0.00	***	13.6
		Sugar	0.00	***	18.6
		Base	0.00	***	17.4
Baseline multiclass segmentation	**Optimised multiclass segmentation**	Phosphate	0.22	n.s.	3.5
		Sugar	0.14	n.s.	1.3
		Base	0.43	n.s.	0.6

Table 8.3: Recall statistics calculated by comparing three binary segmentation models, the baseline multiclass segmentation model and the optimised multiclass segmentation model. Pairwise significance estimations were completed with a Wilcoxon signed-rank test. Significance levels are represented by stars where $P \geq 0.05 =$ n.s., $0.01 \leq P < 0.05 = *$, $0.001 \leq P < 0.01 = **$, $P < 0.001 = ***$. Delta values are the difference between the average performance of both models for a given output, expressed in percentage points (pp).

(a) X-ray diffraction molecular replacement test set

Model		Output	**Recall**		
Ref.	Comp.		P-value	Sig.	Delta / pp
Binary segmentation	**Baseline multiclass segmentation**	Phosphate	0.00	***	-9.5
		Sugar	0.00	***	-13.6
		Base	0.00	***	-12.4
Binary segmentation	**Optimised multiclass segmentation**	Phosphate	0.06	n.s.	-1.6
		Sugar	0.00	***	-2.9
		Base	0.00	***	-3.6
Baseline multiclass segmentation	**Optimised multiclass segmentation**	Phosphate	0.00	***	8.0
		Sugar	0.00	***	10.7
		Base	0.00	***	8.8

(b) Cryo-EM Coulomb potential map test set

Model		Output	**Recall**		
Ref.	Comp.		P-value	Sig.	Delta / pp
Binary segmentation	**Baseline multiclass segmentation**	Phosphate	0.51	n.s.	-0.6
		Sugar	0.00	**	-6.9
		Base	0.00	**	-9.8
Binary segmentation	**Optimised multiclass segmentation**	Phosphate	0.00	***	9.3
		Sugar	0.00	**	7.7
		Base	0.68	n.s.	1.5
Baseline multiclass segmentation	**Optimised multiclass segmentation**	Phosphate	0.00	***	10.0
		Sugar	0.00	***	14.6
		Base	0.00	***	11.4

Table 8.4: F1 score statistics calculated by comparing three binary segmentation models, the baseline multiclass segmentation model and the optimised multiclass segmentation model. Pairwise significance estimations were completed with a Wilcoxon signed-rank test. Significance levels are represented by stars where $P \geq 0.05 =$ n.s., $0.01 \leq P < 0.05 = *$, $0.001 \leq P < 0.01 = **$, $P < 0.001 = ***$. Delta values are the difference between the average performance of both models for a given output, expressed in percentage points (pp).

(a) X-ray diffraction molecular replacement test set

Model		Output	**F1 Score**		
Ref.	Comp.		P-value	Sig.	Delta / pp
Binary segmentation	**Baseline multiclass segmentation**	Phosphate	0.00	***	8.0
		Sugar	0.00	***	6.6
		Base	0.00	***	3.9
Binary segmentation	**Optimised multiclass segmentation**	Phosphate	0.00	***	12.0
		Sugar	0.00	***	11.3
		Base	0.00	***	8.9
Baseline multiclass segmentation	**Optimised multiclass segmentation**	Phosphate	0.00	***	3.9
		Sugar	0.00	***	4.6
		Base	0.00	***	5.0

(b) Cryo-EM Coulomb potential map test set

Model		Output	**F1 Score**		
Ref.	Comp.		P-value	Sig.	Delta / pp
Binary segmentation	**Baseline multiclass segmentation**	Phosphate	0.04	*	4.0
		Sugar	0.06	n.s.	4.2
		Base	0.02	*	5.9
Binary segmentation	**Optimised multiclass segmentation**	Phosphate	0.00	***	11.6
		Sugar	0.00	***	15.9
		Base	0.00	***	14.3
Baseline multiclass segmentation	**Optimised multiclass segmentation**	Phosphate	0.00	***	7.6
		Sugar	0.00	***	11.7
		Base	0.00	***	8.4

Table 8.5: F1 score statistics calculated by comparing the specified reference model against the specified comparison model across an X-ray diffraction test set phased with molecular replacement and a cryo-EM Coulomb potential map test set. Pairwise significance estimations were completed with a Wilcoxon signed-rank test. Significance levels are represented by stars where $P \geq 0.05 =$ n.s., $0.01 \leq P < 0.05 = *$, $0.001 \leq P < 0.01 = **$, $P < 0.001 = ***$. Delta values are the difference between the average performance of both models for a given output type, expressed in percentage points (pp).

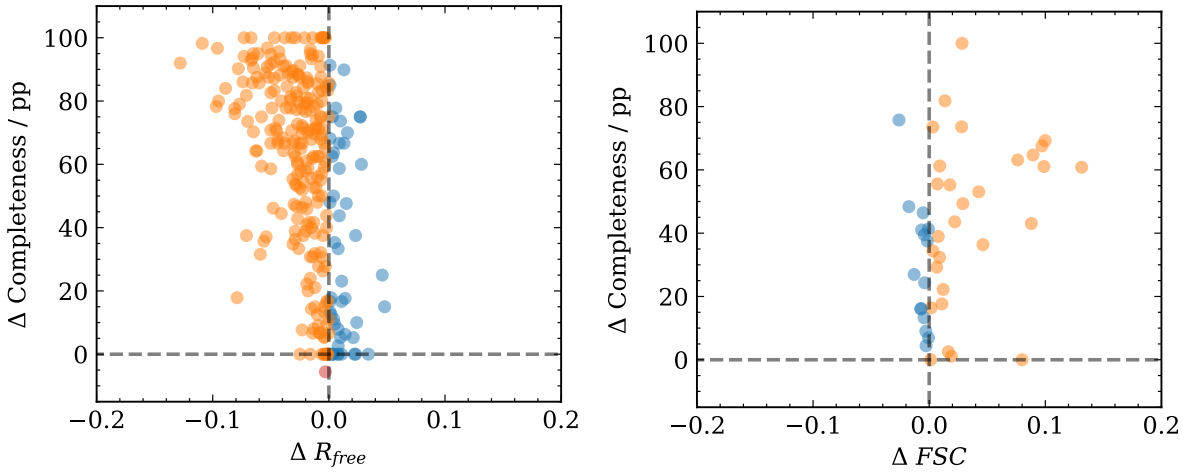
(a) X-ray diffraction molecular replacement test set

Reference Model		**Comparison Model**		**Output**	**F1 Score**		
Down.	Up.	Down.	Up.		P-value	Sig.	Delta / pp
Scale 1	**Scale 1**	**Scale 1**	**Scale 2**	Phosphate	0.00	***	0.6
				Sugar	0.00	***	0.6
				Base	0.00	***	0.6
Scale 1	**Scale 1**	**Scale 2**	**Scale 1**	Phosphate	0.00	***	1.7
				Sugar	0.00	***	1.2
				Base	0.00	***	1.6
Scale 1	**Scale 2**	**Scale 2**	**Scale 1**	Phosphate	0.00	***	1.1
				Sugar	0.00	***	0.7
				Base	0.00	***	1.0
Scale 1	**Scale 2**	**Scale 2**	**Scale 2**	Phosphate	0.00	***	1.1
				Sugar	0.00	***	1.7
				Base	0.00	***	1.9
Scale 2	**Scale 1**	**Scale 2**	**Scale 2**	Phosphate	0.32	n.s.	0.0
				Sugar	0.00	***	1.0
				Base	0.00	***	0.9

(b) Cryo-EM Coulomb potential map test set

Reference Model		**Comparison Model**		**Output**	**F1 Score**		
Down.	Up.	Down.	Up.		P-value	Sig.	Delta / pp
Scale 1	**Scale 1**	**Scale 1**	**Scale 2**	Phosphate	0.00	***	1.6
				Sugar	0.00	***	1.7
				Base	0.00	***	0.9
Scale 1	**Scale 1**	**Scale 2**	**Scale 1**	Phosphate	0.00	***	2.7
				Sugar	0.00	***	3.1
				Base	0.00	***	3.0
Scale 1	**Scale 2**	**Scale 2**	**Scale 1**	Phosphate	0.00	**	1.1
				Sugar	0.00	***	1.4
				Base	0.00	***	2.2
Scale 1	**Scale 2**	**Scale 2**	**Scale 2**	Phosphate	0.34	n.s.	0.1
				Sugar	0.00	**	1.3
				Base	0.73	n.s.	-0.25
Scale 2	**Scale 1**	**Scale 2**	**Scale 2**	Phosphate	0.18	n.s.	-1.0
				Sugar	0.01	*	-0.1
				Base	0.05	n.s.	-2.5

8.2 Structure Solution Metrics for *ModelCraft* With *Nautilus* and *ModelCraft* With *NucleoFind*



(a) R_{free} **(b)** FSC

Figure 8.1: Changes in structure solution performance metrics as a function of additional nucleic acid model-building completeness when comparing *ModelCraft* with *Nautilus* to *ModelCraft* with *NucleoFind*. R_{free} is output by *ModelCraft* across the 288 X-ray diffraction examples, and FSC is output by *ModelCraft* for 50 cryo-EM examples.

8.3 Clustering of Carbohydrate Geometry

When attempting to model a carbohydrate, it is helpful to have knowledge of a representative range of possible linkage conformations which may be independently trialled. To achieve this, clustering was performed for each validated linkage identified in the Protein Data Bank survey. The torsion angles ψ, φ, ω were chosen as representatives for geometry, with each torsion embedded into sin and cos parameters to account for the periodic nature of the torsion angles. This formed a 6-dimensional search space which was clustered using HDBSCAN.[327] The minimum cluster size hyperparameter was adjusted for each linkage after visualising the suggested clusters. The mean and standard deviation for each cluster of torsion angles in linkage are shown in Table 8.6, with the mean and standard deviation of the corresponding bond angles shown in Table 8.7.

It is important to recognise that no clustering method is a perfect solution for any given dataset, and choosing a specific method often involves compromises. In this rudimentary clustering analysis, semi-automatic clustering with HDBSCAN was chosen, as the torsion angles form relatively distinct areas of density, which are most compatible with a density-based clustering method (see Section 1.5.2.1.2). Adjusting the minimum cluster size hyperparameter without an empirical search was chosen to ensure the goal of the clustering process was met rather than optimising a particular scoring metric, but this approach does reduce the repeatability of the analysis. This method is imperfect but was deemed acceptable for the purposes of the investigation into automated carbohydrate model building.

Table 8.6: Clustered torsion data for validated protein-carbohydrate and carbohydrate-carbohydrate linkages obtained through a survey of the Protein Data Bank. Clustering was performed with semi-automatic density-based clustering with HDBSCAN.

Linkage	**Min. Cluster Size**	$\varphi/^\circ$	$\psi/^\circ$	$\omega/^\circ$
NAG-1,2-ASN	400	-97.70 ± 27.29	359.45 ± 6.83	-179.57 ± 8.72
		-102.64 ± 25.78	179.12 ± 6.13	176.27 ± 6.38
		57.63 ± 16.73	180.05 ± 4.69	168.84 ± 8.19
NAG-1,4-NAG	50	-92.05 ± 13.80	61.17 ± 15.52	-178.62 ± 6.18
		-79.72 ± 16.76	-127.93 ± 15.30	178.68 ± 4.64
		85.74 ± 12.10	-109.45 ± 7.65	172.16 ± 6.74
BMA-1,4-NAG	40	-117.12 ± 17.27	69.91 ± 15.99	-177.38 ± 10.38
		-84.64 ± 16.18	-131.41 ± 14.17	179.15 ± 5.33
		51.36 ± 9.51	-121.89 ± 7.71	171.47 ± 3.75
MAN-1,3-BMA	35	79.67 ± 22.25	122.88 ± 21.53	63.17 ± 6.83
		115.03 ± 15.96	-45.78 ± 18.66	60.43 ± 5.04
MAN-1,6-BMA	50	75.67 ± 23.34	178.46 ± 13.23	61.60 ± 5.50
		84.19 ± 19.65	87.97 ± 8.04	59.97 ± 5.41
MAN-1,2-MAN	30	81.87 ± 11.33	135.63 ± 18.96	61.03 ± 4.84
		61.35 ± 3.21	68.13 ± 4.28	61.20 ± 3.49
FUC-1,6-NAG	25	-73.94 ± 8.37	-174.49 ± 11.47	-62.22 ± 4.21
		-72.71 ± 5.71	125.35 ± 17.23	-62.40 ± 3.75
MAN-1,3-MAN	10	81.55 ± 19.60	130.23 ± 22.77	61.84 ± 6.20
		108.76 ± 22.81	-41.20 ± 22.63	59.39 ± 7.29
MAN-1,6-MAN	15	69.71 ± 11.05	-174.85 ± 12.29	60.50 ± 5.13
		120.21 ± 10.10	-143.92 ± 5.03	67.55 ± 2.06
NAG-1,2-MAN	50	-85.24 ± 20.49	148.42 ± 20.21	178.52 ± 5.40
MAN-1,1-SER	50	69.17 ± 21.87	172.89 ± 30.10	60.27 ± 9.14
FUC-1,3-NAG	50	-70.88 ± 12.53	137.66 ± 14.61	-63.85 ± 6.33
MAN-1,1-THR	50	88.01 ± 28.58	129.93 ± 25.83	62.71 ± 10.13
MAN-1,1-TRP	50	120.06 ± 14.00	183.11 ± 5.92	166.49 ± 27.97

Table 8.7: Clustered bond angle data for validated protein-carbohydrate and carbohydrate-carbohydrate linkages obtained through a survey of the Protein Data Bank. Clustering was performed with semi-automatic density-based clustering with HDBSCAN on the corresponding torsion angles.

Linkage	**Min. Cluster Size**	$\alpha/^\circ$	$\beta/^\circ$	$\gamma/^\circ$
NAG-1,2-ASN	400	127.74 ± 5.73	109.11 ± 5.86	114.67 ± 1.94
		121.38 ± 4.18	109.60 ± 3.89	113.85 ± 1.69
		125.12 ± 4.63	112.18 ± 3.97	113.31 ± 2.02
NAG-1,4-NAG	50	118.50 ± 4.22	110.85 ± 2.53	114.83 ± 1.34
		113.04 ± 3.92	111.02 ± 3.09	113.57 ± 1.72
		120.23 ± 3.25	109.47 ± 2.00	114.13 ± 2.33
BMA-1,4-NAG	40	118.89 ± 3.83	110.24 ± 2.47	115.76 ± 2.34
		112.92 ± 3.72	110.73 ± 3.52	113.73 ± 1.85
		117.77 ± 3.19	113.62 ± 1.35	111.69 ± 1.33
MAN-1,3-BMA	35	111.88 ± 3.67	112.37 ± 3.93	115.08 ± 2.10
		116.70 ± 2.74	110.52 ± 1.34	114.33 ± 1.49
MAN-1,6-BMA	50	111.08 ± 3.19	111.65 ± 2.91	114.79 ± 1.86
		112.29 ± 3.59	113.29 ± 3.15	114.58 ± 1.96
MAN-1,2-MAN	30	112.00 ± 3.67	111.89 ± 2.38	114.92 ± 1.88
		116.88 ± 2.86	114.58 ± 2.38	114.96 ± 2.18
FUC-1,6-NAG	25	111.94 ± 3.33	111.52 ± 2.23	115.16 ± 1.96
		111.50 ± 2.81	112.18 ± 1.81	115.28 ± 1.59
MAN-1,3-MAN	10	112.28 ± 4.98	112.17 ± 3.70	114.72 ± 1.94
		116.97 ± 2.36	110.66 ± 2.19	114.47 ± 1.86
MAN-1,6-MAN	15	111.91 ± 3.47	111.64 ± 3.24	114.69 ± 2.04
		107.91 ± 1.05	110.22 ± 0.46	115.45 ± 0.48
NAG-1,2-MAN	50	112.37 ± 4.56	110.39 ± 3.71	113.60 ± 1.86
MAN-1,1-SER	50	113.15 ± 4.54	112.02 ± 2.94	114.81 ± 1.71
FUC-1,3-NAG	50	114.32 ± 4.32	113.36 ± 3.79	114.78 ± 2.16
MAN-1,1-THR	50	112.71 ± 5.56	111.71 ± 2.41	114.50 ± 1.98
MAN-1,1-TRP	50	125.33 ± 3.68	109.29 ± 3.62	117.77 ± 1.93

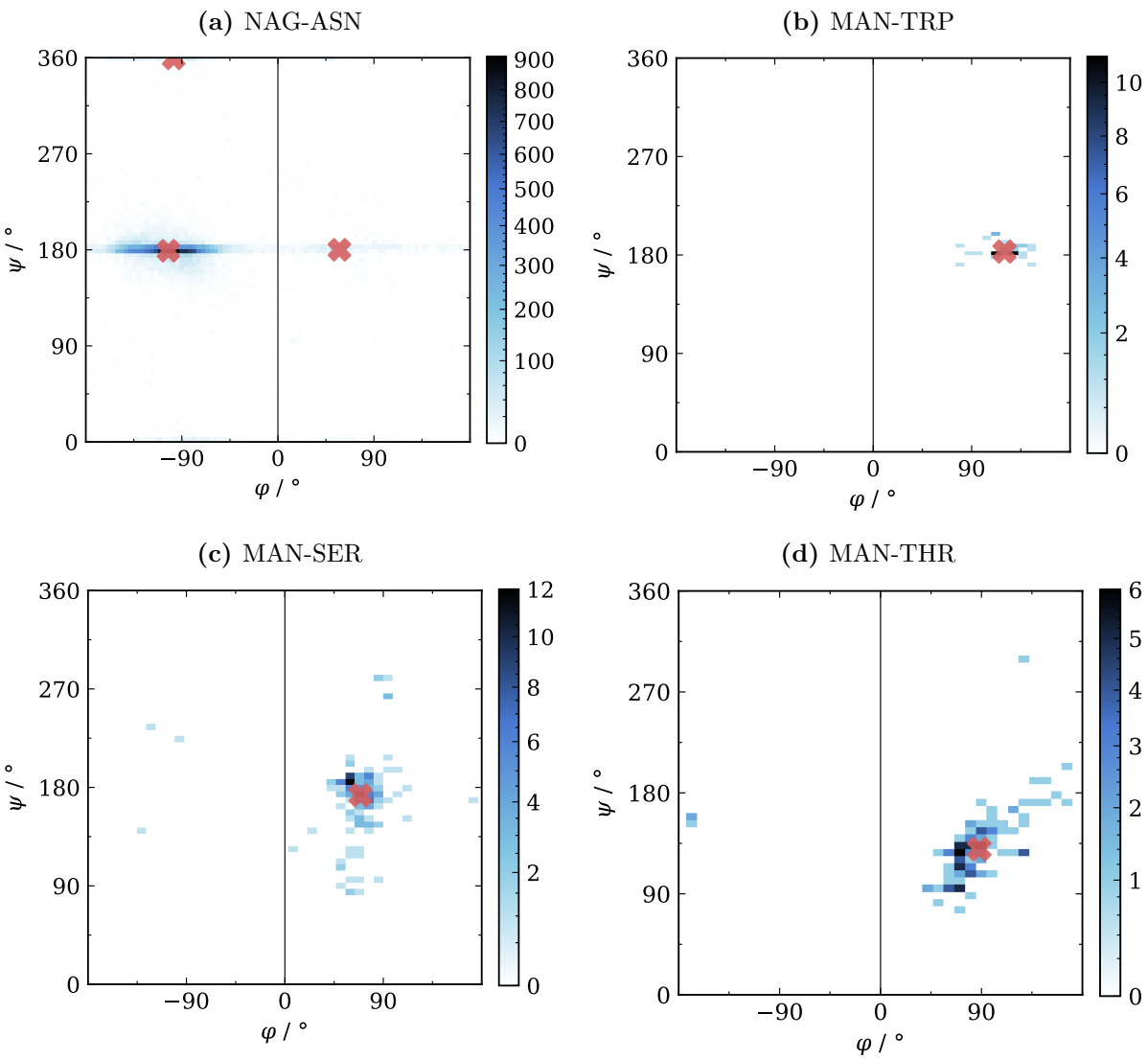


Figure 8.2: Centroids of geometric clusters obtained using HDBSCAN, shown with crosses, with two-dimensional histograms showing ψ and φ torsion angles for validated protein-sugar linkages found in a survey of the Protein Data Bank. Only linkages with greater than 50 validated occurrences are shown, with frequency denoted by colour. The colour bar is plotted with the Power Law distribution[286] to aid visualisation of less frequent bins.

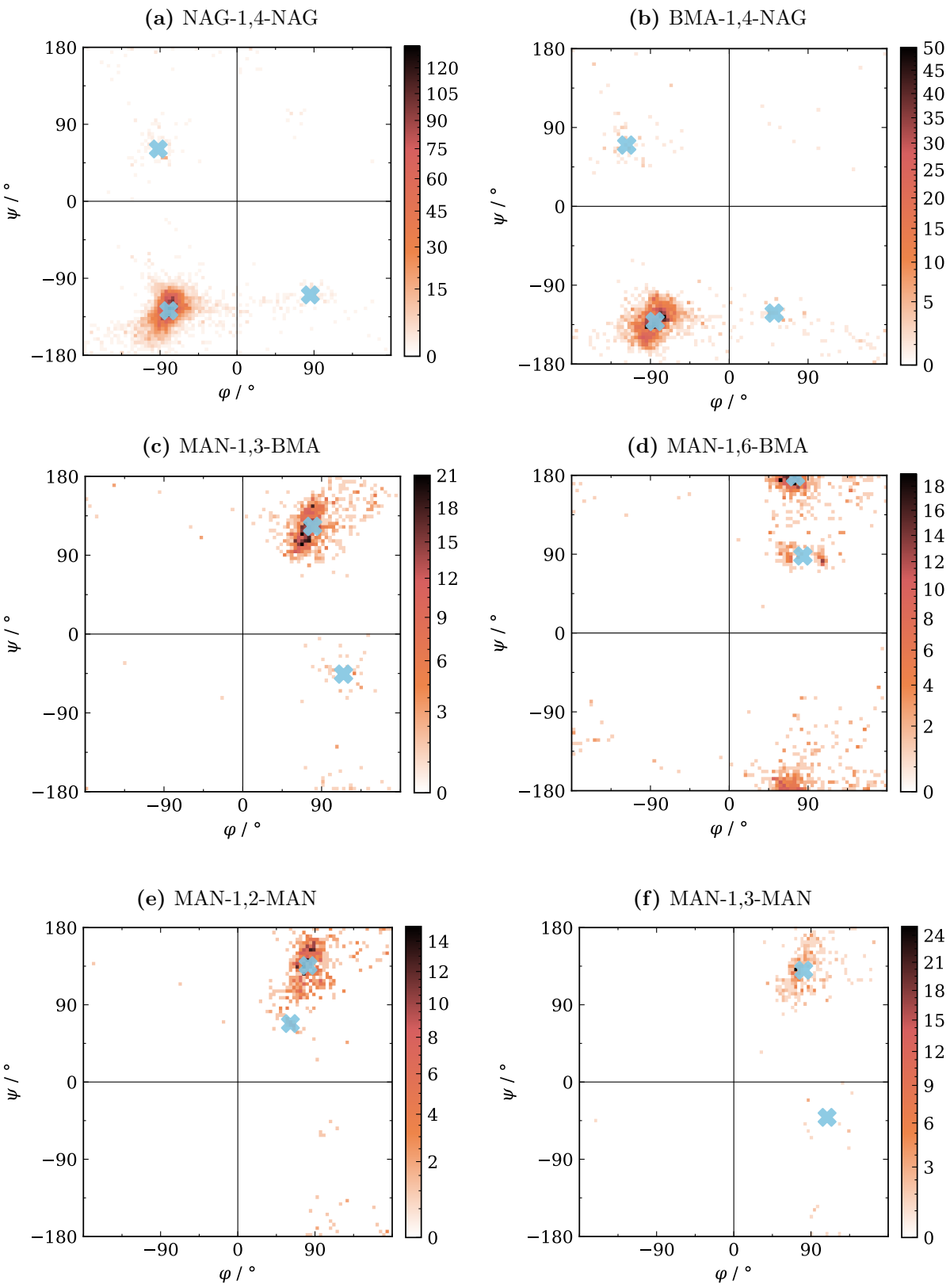


Figure 8.3: Centroids of geometric clusters obtained using HDBSCAN, shown with crosses, with two-dimensional histograms showing ψ and φ torsion angles for validated sugar-sugar linkages found in a survey of the Protein Data Bank.

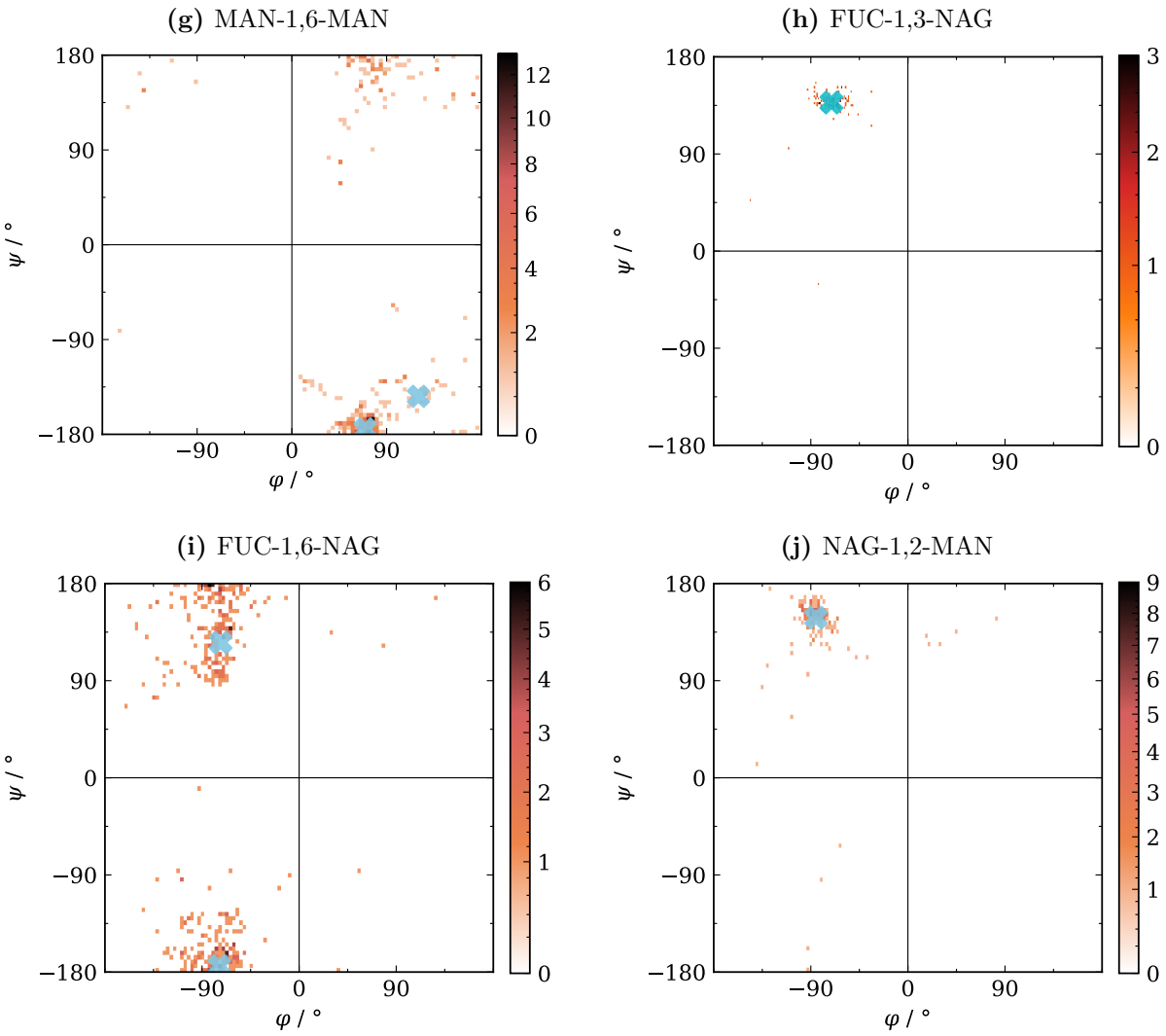


Figure 8.3 (continued): Centroids of geometric clusters obtained using HDBSCAN, shown with crosses, with two-dimensional histograms showing ψ and φ torsion angles for validated sugar-sugar linkages found in a survey of the Protein Data Bank. Only linkages with greater than 50 validated occurrences are shown, with frequency denoted by colour. The colour bar is plotted with the Power Law distribution[286] to aid visualisation of less frequent bins.

Part IV

Appendix

University of York
York Graduate Research School
Research Degree Thesis Statement of Authorship

Note that where a paper has multiple authors, the statement of authorship can focus on the key contributing/corresponding authors.

Candidate name	**Jordan Singh Dialpuri**
Department	**Chemistry**
Thesis title	**Automated Model Building of Nucleic Acids and Carbohydrates Using Experimental Data and Deep Learning Models**

Title of the work (paper/chapter)	**NucleoFind: a deep-learning network for interpreting nucleic acid electron density**	
Publication status	**Published**	**X**
	Accepted for publication	
	Submitted for publication	
	Unpublished and unsubmitted	
Citation details (if applicable)	Jordan S Dialpuri, Jon Agirre, Kathryn D Cowtan, Paul S Bond, NucleoFind: a deep-learning network for interpreting nucleic acid electron density, Nucleic Acids Research, Volume 52, Issue 17, 23 September 2024, Page e84, https://doi.org/10.1093/nar/gkae715	

Description of the candidate's contribution to the work*	Model development, model evaluation, model optimisation, figure generation, writing
Signature of the candidate	
Date (DD/MM/YY)	05/01/26

Co-author contributions

By signing this Statement of Authorship, each co-author agrees that:
(i) the candidate has accurately represented their contribution to the work;
(ii) if required, permission is granted for the candidate to include the work in their thesis (note that this is separate from copyright considerations).

Name of co-author	Jon Agirre
Contact details of co-author	jon.agirre@york.ac.uk
Description of the co-author's contribution to the work*	Supervision and support
Signature of the co-author	
Date (DD/MM/YY)	05/01/26

Name of co-author	Kathryn Cowtan
Contact details of co-author	kathryn.cowtan@york.ac.uk
Description of the co-author's contribution to the work*	Supervision and support
Signature of the co-author	
Date (DD/MM/YY)	05/01/26

Name of co-author	Paul Bond
Contact details of co-author	paul.bond@york.ac.uk
Description of the co-author's contribution to the work*	Initial model development, supervision, and support
Signature of the co-author	
Date (DD/MM/YY)	05/01/26