Leveraging deep learning for identification and structural determination of novel protein complexes from *in situ* electron cryotomography of *Mycoplasma pneumoniae*

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Julius-Maximilians-Universität Würzburg

vorgelegt von

Joseph Christian Campbell Somody

Geburtsort: Toronto, Canada

Würzburg, 2023

Eingereicht am: 2023-02-17

Mitglieder der Promotionskommission:

Vorsitzende: Prof. Dr. Vera Kozjak-Pavlovic

Gutachter: Prof. Dr. Peer Bork

Gutachter: Prof. Dr. Thomas Dandekar

Tag des Promotionskolloquiums: 2023-04-21

Doktorurkunde ausgehändigt am: ....

Eidesstattliche Erklärungen nach §7 Abs. 2 Satz 3, 4, 5 der Promotionsordnung der Fakultät für Biologie

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, die Dissertation "Tiefenlernen als Werkzeug zur Identifizierung und Strukturbestimmung neuer Proteinkomplexe aus der In-situ-Elektronenkryotomographie von *Mycoplasma pneumoniae*", eigenständig, d. h. insbesondere selbständig und ohne Hilfe eines kommerziellen Promotionsberaters, angefertigt und keine anderen, als die von mir angegebenen Quellen und Hilfsmittel verwendet zu haben.

Ich erkläre ausserdem, dass die Dissertation weder in gleicher noch in ähnlicher Form bereits in einem anderen Prüfungsverfahren vorgelegen hat.

Weiterhin erkläre ich, dass bei allen Abbildungen und Texten bei denen die Verwertungsrechte (Copyright) nicht bei mir liegen, diese von den Rechtsinhabern eingeholt wurden und die Textstellen bzw. Abbildungen entsprechend den rechtlichen Vorgaben gekennzeichnet sind sowie bei Abbildungen, die dem Internet entnommen wurden, der entsprechende Hypertextlink angegeben wurde.

Affidavit

I hereby declare that my thesis entitled "Leveraging deep learning for identification and structural determination of novel protein complexes from *in situ* electron cryotomography of *Mycoplasma pneumoniae*" is the result of my own work. I did not receive any help or support from commercial consultants. All sources and/or materials applied are listed and specified in the thesis.

Furthermore, I verify that the thesis has not been submitted as part of another examination process neither in identical nor in similar form.

Besides, I declare that if I do not hold the copyright for figures and paragraphs, I obtained it from the rightsholder and that paragraphs and figures have been marked according to law or, for figures taken from the Internet, the hyperlink has been added accordingly.

Würzburg, den ...2023-05-04....

....
Signature

Abstract

Leveraging deep learning for identification and structural determination of novel protein complexes from *in situ* electron cryotomography of *Mycoplasma pneumoniae*

Joseph Christian Campbell Somody

The holy grail of structural biology is to study a protein *in situ*, and this goal has been fast approaching since the resolution revolution and the achievement of atomic resolution. A cell's interior is not a dilute environment, and proteins have evolved to fold and function as needed in that environment; as such, an investigation of a cellular component should ideally include the full complexity of the cellular environment. Imaging whole cells in three dimensions using electron cryotomography is the best method to accomplish this goal, but it comes with a limitation on sample thickness and produces noisy data unamenable to direct analysis. This thesis establishes a novel workflow to systematically analyse whole-cell electron cryotomography data in three dimensions and to find and identify instances of protein complexes in the data to set up a determination of their structure and identity for success. *Mycoplasma pneumoniae* is a very small parasitic bacterium with fewer than 700 protein-coding genes, is thin enough and small enough to be imaged in large quantities by electron cryotomography, and can grow directly on the grids used for imaging, making it ideal for exploratory studies in structural proteomics. As part of the workflow, a methodology for training deep-learning-based particle-picking models is established.

As a proof of principle, a dataset of whole-cell *Mycoplasma pneumoniae* tomograms is used with this workflow to characterize a novel membrane-associated complex observed in the data. Ultimately, 25 431 such particles are picked from 353 tomograms and refined to a density map with a resolution of 11 Å. Making good use of orthogonal datasets to filter search space and verify results, structures were predicted for candidate proteins and checked for suitable fit in the density map. In the end, with this approach, nine proteins were found to be part of the complex, which appears to be associated with chaperone activity and interact with translocon machinery.

Visual proteomics refers to the ultimate potential of *in situ* electron cryotomography: the comprehensive interpretation of tomograms. The workflow presented here is demonstrated to help in reaching that potential.

Zusammenfassung

Tiefenlernen als Werkzeug zur Identifizierung und Strukturbestimmung neuer Proteinkomplexe aus der In-situ-Elektronenkryotomographie von *Mycoplasma pneumoniae*

Joseph Christian Campbell Somody

Der heilige Gral der Strukturbiologie ist die Untersuchung eines Proteins *in situ*, und dieses Ziel ist seit der Auflösungsrevolution und dem Erreichen der atomaren Auflösung in greifbare Nähe gerückt. Das Innere einer Zelle ist keine verdünnte Umgebung, und Proteine haben sich so entwickelt, dass sie sich falten und so funktionieren, wie es in dieser Umgebung erforderlich ist; daher sollte die Untersuchung einer zellulären Komponente idealerweise die gesamte Komplexität der zellulären Umgebung umfassen. Die Abbildung ganzer Zellen in drei Dimensionen mit Hilfe der Elektronenkryotomographie ist die beste Methode, um dieses Ziel zu erreichen, aber sie ist mit einer Beschränkung der Probendicke verbunden und erzeugt verrauschte Daten, die sich nicht für eine direkte Analyse eignen. In dieser Dissertation wird ein neuartiger Workflow zur systematischen dreidimensionalen Analyse von Ganzzell-Elektronenkryotomographiedaten und zur Auffindung und Identifizierung von Proteinkomplexen in diesen Daten entwickelt, um eine erfolgreiche Bestimmung ihrer Struktur und Identität zu ermöglichen. *Mycoplasma pneumoniae* ist ein sehr kleines parasitäres Bakterium mit weniger als 700 proteinkodierenden Genen. Es ist dünn und klein genug, um in grossen Mengen durch Elektronenkryotomographie abgebildet zu werden, und kann direkt auf den für die Abbildung verwendeten Gittern wachsen, was es ideal für Sondierungsstudien in der strukturellen Proteomik macht. Als Teil des Workflows wird eine Methodik für das Training von Deep-Learning-basierten Partikelpicken-Modellen entwickelt.

Als Proof-of-Principle wird ein Dataset von Ganzzell-Tomogrammen von *Mycoplasma pneumoniae* mit diesem Workflow verwendet, um einen neuartigen membranassoziierten Komplex zu charakterisieren, der in den Daten beobachtet wurde. Insgesamt wurden 25 431 solcher Partikel aus 353 Tomogrammen gepickt und zu einer Dichtekarte mit einer Auflösung von 11 Å verfeinert. Unter Verwendung orthogonaler Datensätze zur Filterung des Suchraums und zur Überprüfung der Ergebnisse wurden Strukturen für Protein-Kandidaten vorhergesagt und auf ihre Eignung für die Dichtekarte überprüft. Letztendlich wurden mit diesem Ansatz neun Proteine als Bestandteile des Komplexes gefunden, der offenbar mit der Chaperonaktivität in Verbindung steht und mit der Translocon-Maschinerie interagiert.

Das ultimative Potenzial der In-situ-Elektronenkryotomographie – die umfassende Interpretation von Tomogrammen – wird als visuelle Proteomik bezeichnet. Der hier vorgestellte Workflow soll dabei helfen, dieses Potenzial auszuschöpfen.

Table of Contents

List of Figures

List of Tables

Abbreviations

2D	two-dimensional
3D	three-dimensional
Å	ångström
AA	amino acid
AlphaFold DB	AlphaFold Protein Structure Database
ANUBIS	Analysis of Unbiased Insertions
ATCC	American Type Culture Collection
BLAST	Basic Local Alignment Search Tool
bp	base pair
BS^3	bis(sulfosuccinimidyl)suberate
°C	degree Celsius
CARE	Content-Aware Image Restoration
CASP	Critical Assessment of Protein Structure Prediction
CCD	charge-coupled device
CCP-EM	Collaborative Computational Project for EM
CLMS	crosslinking mass spectrometry
Cm	chloramphenicol
CNN	convolutional neural network
cryo-EM	electron cryomicroscopy
cryo-ET	electron cryotomography
CTF	contrast transfer function
Da	dalton
DSSO	disuccinimidyl sulfoxide
DUF	domain of unknown function
e$^-$	electron
EM	electron microscopy
EMBL	European Molecular Biology Laboratory
ET	electron tomography
eV	eletronvolt
FBP	filtered backprojection
FSC	Fourier shell correlation
GAN	generative adversarial network
g	gram
HEPES	4-(2-hydroxyethyl)-1-piperazineethanesulfonic acid
h	hour
HPC	high-performance computing
LC–MS/MS	liquid chromatography–tandem mass spectrometry
L	litre
m	metre

mol	mole
ML	machine learning
MSA	multiple sequence alignment
NN	neural network
PBS	phosphate-buffered saline
PCA	principal component analysis
PDB	Protein Data Bank
pH	potential of hydrogen
PPLO	pleuropneumonia-like organism
PSI-BLAST	Position-Specific Iterative BLAST
PUM	pseudouridimycin
px	pixel
RELION	Regularised Likelihood Optimisation
ReLU	rectified linear unit
RMSD	root-mean-square deviation
RNA	ribonucleic acid
SNR	signal-to-noise ratio
STA	subtomogram averaging
STAR	Self-Defining Text Archive and Retrieval
STR	short tandem repeat
TAC	Thesis Advisory Committee
TEM	Transmission electron microscopy
TM	template matching
TMT	tandem mass tag
Tn-seq	transposon sequencing
U	unit
VPP	Volta phase plate
WBP	weighted backprojection
YOLO	You Only Look Once

Chapter 1

Introduction

1.1 Overview

Figure 1.1 is a flowchart depicting an overview of the novel workflow demonstrated in this thesis. This workflow will be used systematically to analyse whole-cell cryo-ET data in 3D, find and identify instances of a protein complex in this data, and determine its structure and identity. Starting from the raw imaging data, tomograms are reconstructed, and template matching is carried out to find instances of a particle of interest within the tomograms. These instances are then used to start the iterative approach of training a convolutional neural network to pick more particles in the data more accurately. In the end, orthogonal datasets are used to focus the search for the constituent proteins of the particle, as well as to verify results along the way. These steps will be

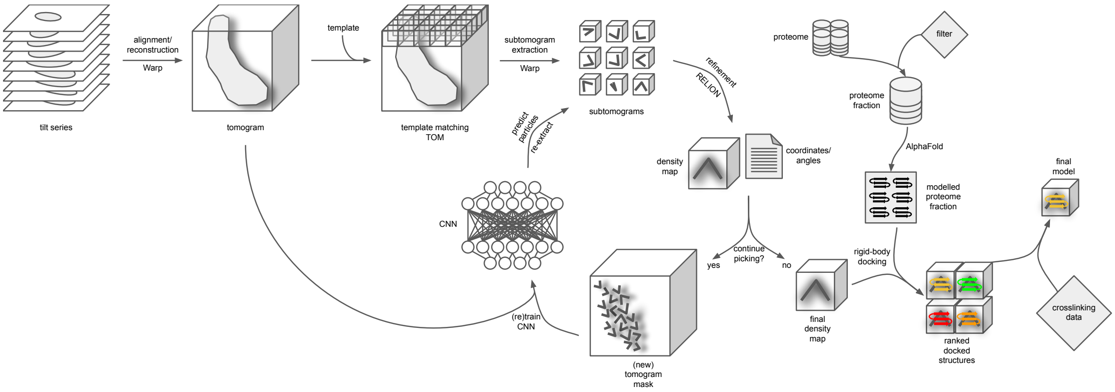


Figure 1.1: A diagram illustrating the workflow demonstrated as part of this thesis, including the iterative picking and refinement process for the particle of interest, as well as orthogonal datasets used for lead prioritization and result verification along the way.

described in much greater detail; Figure 1.1 is meant as a guide. As a proof of principle, a dataset of whole-cell *Mycoplasma pneumoniae* tomograms will be used with this workflow to characterize a novel membrane-associated complex.

1.2 *Mycoplasma pneumoniae*

Mycoplasma pneumoniae is a very small parasitic bacterium, one of the smallest self-replicating organisms, lacking a cell wall and periplasmic space [1]. It is pathogenic to humans and involved in a number of diseases including atypical bacterial pneumonia [2]. It attaches to a host by means of a cellular architecture called the attachment organelle formed at one pole of the cell [3]. The *M. pneumoniae* genome has been fully sequenced since 1996 [1]. Containing 688 annotated open reading frames, its genome of approximately 800 kbp has undergone a reduction in size from ancestral bacteria, explained in least in part by its loss of many metabolic pathways and of genes for the synthesis of complex structures like the bacterial cell wall as it developed a parasitic lifestyle [1, 4].

Despite the simplicity of the *M. pneumoniae* genome—lacking genes, *inter alia*, for the *de novo* synthesis of nucleotides and amino acids [1, 4]—there remains “a remarkable level of structural complexity” in *M. pneumoniae* [5, 6], and it’s no surprise that the species in the *Mycoplasma* genus were already identified in 1992 as ideal candidates for investigating cellular machinery [7]. In 2009, *M. pneumoniae* was the subject of detailed functional analysis [8], including studies into its proteome organization [9], metabolic and regulatory machinery [10], and transcriptome complexity [11].

In terms of their physical characteristics, *Mycoplasma pneumoniae* cells have a spherical to filamentous shape on the order of 0.1–0.2 µm in width and 1–2 µm in length [12, 13], with symmetric round forms during early growth, filamentous or flask-shaped forms in exponential growth, and asymmetric round forms in late-stage growth [14, 15].

With an average cell volume only 5% of that of *Escherichia coli* [5], colonies of *Mycoplasma pneumoniae* are rarely larger than 100 µm in diameter, even when grown on a rich medium [13]. Sterols are a required component of the *M. pneumoniae* cell membrane and are sourced from the host *in vivo*, meaning that growing *M. pneumoniae in vitro* requires a growth medium enriched with sterols [13]. Growth in SP4 medium (serum-supplemented) leads to successful but slow culture of *M. pneumoniae* under atmospheric conditions at 37 °C [13], still with a doubling time of about 8 h [10, 16]. *M. pneumoniae* reproduces via binary fission, wherein the attachment organelle is duplicated and the duplicate then relocated to the opposing polar end of the cell prior to nucleoid separation [17]. *M. pneumoniae* can bind to and move along solid surfaces (e.g. glass) [13], including carbon-coated grids used in electron microscopy [18].

Its small and simple genome makes *M. pneumoniae* an organism suitable for further studies in proteomics or, in this case, a proof of principle in structural proteomics. In addition, *M. pneumoniae* has a simple cellular structure, containing "the minimum set of organelles essential for growth and replication" [12], allowing such efforts in structural proteomics (to be detailed in Section 1.3) to go unoverwhelmed by cellular architecture and complex genome/proteome organization.

A wealth of information has already been published on *M. pneumoniae* omics, which also proves useful; however, approximately one third of its proteome is still without functional prediction, and at least 250 proteins continue to lack any characterization. In particular, compared to its soluble proteome, the *M. pneumoniae* membrane proteome is much less well studied, which is why not much is yet known about its parasitic mechanisms in detail (e.g. attachment to host, uptake of nutrients) [9].

Finally, the fact that *M. pneumoniae* can grow directly on grids used in electron microscopy, with a cellular thickness small enough to do without the need for milling [14, 19], and with a cellular size such that individual cells can efficiently be imaged in bulk, allows it to be an ideal model organism for further large-scale structural proteomic studies, especially using whole-cell cryo-ET to obtain *in situ* imaging data.

1.3 Electron cryotomography

1.3.1 History

Since the discovery of the cell as the biological functional unit around 350 years ago, advances in much of biology have been facilitated by developments in biological imaging technologies. In the late 1870s, however, Ernst Abbe postulated that approximately half of the wavelength of light used in light microscopy is an upper limit to the achievable resolution, which meant a cap of around 1000 Å (100 nm) [20]. The ångström (Å) is a metric unit used commonly in the natural sciences, equal to 10^{-10} m (0.1 nm). This cap lasted until the early 1930s, after magnetic coils were discovered to focus an electron beam in much the same way as a lens can focus light, and Ernst Ruska produced the first electron microscope [20]. First used for imaging inorganic materials [21], biology needed two more decades to work out technical challenges before electron microscopy (EM) could aid in many discoveries in the 1950s and 1960s [22]. One such challenge was handling the thickness and sensitivity of biological samples, which led to the use of chemical fixation and sectioning of samples embedded in paraffin [22]. Even with sectioning, such sample thicknesses required the use of strong vacuums and higher voltages in the microscope, and the resulting increased temperatures necessitated developing more robust materials for embedding and therefore also improved methods of sectioning [21]. EM grids, made of metal and sometimes carbon-coated, are just millimetres in diameter and contain from tens to thousands of holes in the mesh to act as an electron-permissive support for these fragile sectioned samples [21]. The final step in imaging is the recording of the transmitted electrons. Initially, this was done indirectly with the use of a charge-coupled device (CCD) camera, converting electrons to photons before being detected by a standard sensor; nowadays, direct detectors are used, which image from the electron beam directly [23]. Both methods allow for real-time digital output [21].

Reconstructing a three-dimensional (3D) model by interpolating from individual two-dimensional (2D) image slices of the sample can result in a lack of detail [24]. Although such interpolation worked well in specific cases and sufficiently thin serial sections [25], the development of electron tomography (ET) was a breakthrough in reconstructing accurate 3D images. Using transmission electron microscopy (TEM) for ET, the grid with a mounted sample is loaded into a tiltable holder on a movable stage, and images are recorded over a range of tilt angles, whose limits are dictated by the geometry of the held grid and leads to some missing information referred to as the "missing wedge" [24]. With software such as IMOD [26], the output projection images are processed and used to reconstruct a 3D volume called a tomogram via a method such as weighted backprojection (WBP). In particular, the z-axis resolution is greatly improved using ET rather than interpolated reconstructions from serial sections [24]. A schematic of this process is shown in Figure 1.2, where the sample is imaged at different tilt angles (Subfigure 1.2A), each image represents a 2D projection for a particular tilt angle (Subfigure 1.2B), and then the tilt series is used in backprojection to reconstruct a 3D image volume called the tomogram (Subfigure 1.2C).

Further developments in ET were made by transitioning to very low temperatures. Samples can be cryosectioned after cryofixation by high-pressure freezing, for example, which avoids the artefacts of chemical fixation [24]. Cryogenic EM (cryo-EM) is the imaging of such samples using an electron microscope with a liquid-helium-cooled cryostage, and the same semantic extension applies for cryogenic ET (cryo-ET) [27].

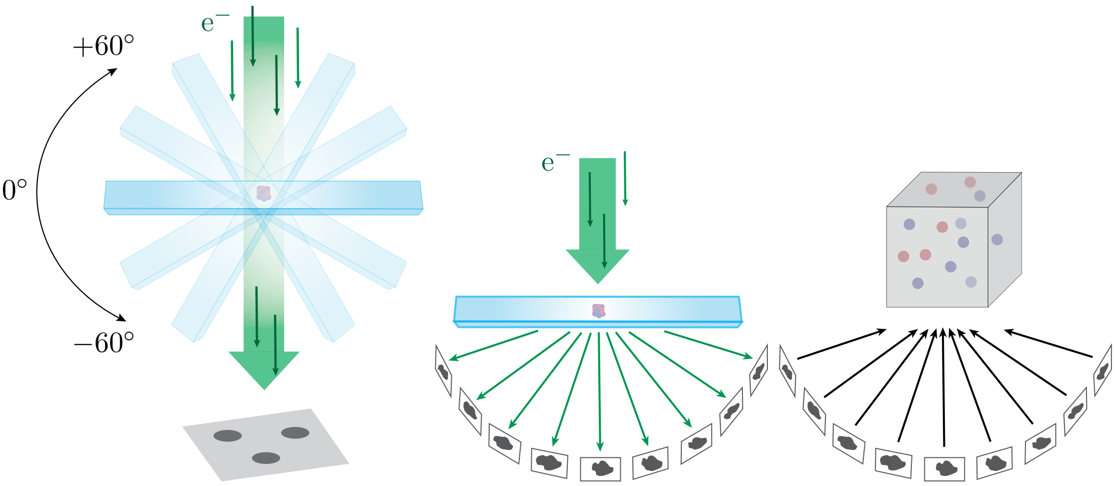
For the twenty years since then, hardware developments have been the major force behind increasing resolution in cryo-EM [28], to the point where "atomic resolution"—a resolution, generally around 1.2 Å, at which individual atoms can be distinguished [29]—is now achievable [30, 31]. For a while, these groundbreaking resolutions were mostly coming from single-particle cryo-EM, where molecules (such as purified protein) in suspension are imaged in 2D, and a 3D reconstruction is possible thanks to the random projection orientations among particles [32].

While single-particle cryo-EM is considered most appropriate for the structural determination of large protein complexes, cryo-ET can also be used for this purpose, especially for heterogeneous samples that would likely not work well with single-particle cryo-EM [23]. In addition to purified protein complexes, organelles and whole bacterial cells were also some of the first samples to be imaged with cryo-ET [33]. Although mostly used to investigate lower-resolution morphology in the beginning, cellular cryo-ET has recently been used to obtain high-resolution *in situ* structures of protein complexes [34]. Molecular cryo-ET (cryo-ET with purified samples) and cellular cryo-ET are two approaches in the same general method, and both require subsequent extraction of 3D subvolumes from the tomogram called subtomograms [35, 36]. An alignment in 3D space of these subtomograms must be performed, and the aligned subtomograms can then be averaged to improve the signal-to-noise ratio and classified into different states or assemblies [34]. In 2019, Zhang argued that this ability to align, average, and classify is possibly the best feature of cryo-ET, since the 3D reconstruction of each particle identified in the tomogram exists independently, allowing for direct analysis of variability in 3D [34].

In spite of all the advantages of cryo-ET, there are some major limitations, the most obvious of which is the maximum sample thickness of approximately 500 nm, which makes whole-cell imaging of nearly all eukaryotes impractical [33]. Efforts are made in sample preparation to have cell cultures in a known dynamic state, but another limitation is that dynamic information is lost in cryo-ET [33], although recently it has been shown that ribosome dynamics could be analysed via a classification and quantification of different ribosome states [37].

1.3.2 Visual proteomics

A cell's interior is not a dilute environment; on the contrary, it is crowded and contains 200–400 g/L of macromolecules [39], a concentration that translates to somewhere on the order of 100 000 proteins per *Mycoplasma pneumoniae* cell. One consequence of this molecular crowding is that an individual macromolecule has a restricted subspace of the cell volume in which to move about, and this increases the likelihood of both specific and non-specific intermolecular associations [39]. These associations are not immune from evolutionary pressure: proteins undergo selection to facilitate productive encounters with appropriate partners and avoid useless or harmful encounters with other partners [39].



(A) The sample is imaged at various tilt angles.

(B) Each image is a 2D projection of the sample at one tilt angle.

(C) The tilt series is back-projected to reconstruct a tomogram.

FIGURE 1.2: A schematic showing the steps involved in data collection and tomogram reconstruction in electron cryotomography. This figure was inspired by a similar one by Galaz-Montoya and Ludtke [38].

Cells certainly contain non-transient multiprotein complexes and have defined signalling pathways, but there remains a large proportion of their macromolecular interactions occurring non-randomly, and stochastic and unbounded models for diffusion and mixing are therefore not suited to the in-cell environment [39]. Another consequence of the crowded cell interior is that protein conformations are pushed to have compact and stable states in such an environment, and this shouldn't be assumed to carry over to a lower-complexity environment [39].

Structural biology has traditionally been a venture in reductionism, purifying biological macromolecules individually or in small complexes, thereby ignoring their context *in situ* and potentially also reporting non-physiological conformations [40]. A protein's affinities for various ligands or substrates, as well its other intrinsic properties, are environment-dependent, and so any investigation of a cellular component should ideally include the full complexity of the cellular environment, a near-impossible task if starting from individual pieces [39]. As Gierasch wrote in 2009, "the holy grail is to study a protein *in situ*", revealing new functional information and also allowing direct observation of the effect of perturbations at the cellular level [39]. Cryo-ET is the tool with which this problem has become approachable, maintaining the context and variability of conformations and interactions [41], and allowing structural characterizations true to their functional environments [42].

The term "visual proteomics" was used as early as 2004 [43] and refers to the ultimate potential of *in situ* cryo-ET: the comprehensive interpretation of tomograms [44]. Visual proteomics has been an idea ever since [45], and some successful attempts at visual proteomics in a limited sense have been made [46, 47], but it was only recently that hardware and software in cryo-ET reached the point where this entered the realm of possibility at scale [40]. The developments needed to reach this point have been truly multidisciplinary: the physics and engineering behind advances in electron microscopes, the computational methods behind advances in image processing, and the bioinformatics behind tools to analyse resulting data, not to mention the computing power to handle the massive computations inherent to such high-resolution tomograms [40].

Distortions in reconstructed tomograms, as well as limitations with contrast and resolution, are among the challenges in full visual proteomics, not to mention the scope of the problem considering the size of proteomes and crowded cellular environment to disentangle [44]. Denoising is often the first step after tomogram reconstruction, since it allows for easier interpretation of tomograms [44]. There are upsides and downsides to using denoised tomograms at various stages of processing, which will be covered in Subsection 1.4.3 in detail. Blurring and artefacts may be introduced by typical denoising processes, and while image contrast can be improved by deconvolution filters, noise may still be prominent [48].

The next general step is particle localization, attempting to locate all instances of a particular preselected particle of interest, the most standard method for which is template matching (TM), where the cross-correlation of a template is calculated throughout the tomogram to identify putative instances [44]. Section 2.3 will cover the various sources for deriving these templates, as well the advantages and drawbacks of TM overall, in more detail. In the past few years, neural networks (NNs) have also been applied to the problems of particle localization. This, too, will be covered in more detail in Subsection 1.4.4.

Subvolumes can now be extracted at the positions indicated by particle localization and aligned and averaged with refinement software, a method that works well for large particles but quickly becomes overwhelmingly more complex for smaller particles [44]. This subtomogram averaging (STA) has the advantage of producing a higher signal-to-noise ratio (SNR) compared to the inherently noisy tomograms [49]. In addition to size, the abundance of particles is also a limitation of STA, since copy numbers within the cell are an upper bound on the number of particles identified [49]. With large and abundant particles, near-atomic resolution has recently been attained [50]. Often more important than resolution, though, is contextual information, depending on the particular problem at hand [49]. Subsection 2.6 will contain more detail on this and STA in general.

Contrast in cryo-ET of frozen-hydrated biological samples is predominantly phase contrast, originating in the electron phase shift during elastic scattering by the sample [48]. This is traditionally produced as defocus phase contrast by defocusing the objective lens such that the electron beam is focused slightly above the sample [51]. The CTF effects cannot be fully removed from the image, since the CTF is zero at some frequencies, leading to no data at those frequencies [52]. CTF effects can be modelled in 3D reconstruction, however, by weighting information from multiple images such that higher-contrast images contribute more to each particular voxel in Fourier space [52]. While defocus phase contrast works well for higher spatial frequencies, this is not the case for low spatial frequencies, giving rise to overall low contrast in the resulting images and thereby hampering interpretability [51]. Using phase plates such as the Volta phase plate (VPP) [53, 54] allows for a significant increase in image contrast, which can be important in visual proteomics in order to help with detecting and identifying smaller particles [48]. The downside to the VPP, however, is that it seems to weaken the signal at higher resolutions [55], leading to a compromise between contrast and resolution [48]. One way of adapting to this compromise is by acquiring two datasets: a smaller one with higher contrast to aid in guiding the workflow (e.g. template generation) and a larger one with lower contrast to drive the aspects more dependent on resolution (e.g. structural determination). On top of all this, there is still the problem of the untargeted portion of visual proteomics: even with many selected proteins of interest mapped into a tomogram, there will remain a proportion of unidentified densities. For smaller macromolecules that evade detection and identification, orthogonal data, such as crosslinking mass spectrometry (CLMS), can be used to infer interactions with larger 'anchor' protein complexes [56].

In 2010, Förster *et al.* detailed the steps involved in the best methods in visual proteomics of the time [47], based mainly on the experiences gained in the 2009 *Leptospira interrogans* study by Beck *et al.* [57]. They imaged and reconstructed tomograms for *L. interrogans* cells under various conditions, each covering approximately 10% of the average cell volume, and generated templates of a handful of protein complexes of interest from atomic maps in the Protein Data Bank (PDB) [57]. They performed template matching with a specialized scoring function that also considered template cross-correlation with tomogram background, competing templates, and simple geometric decoy templates, and found that specificity (based on approximate total number based on quantitative mass spectrometry) was significantly decreased for smaller targets and the detection of low-abundance targets was very challenging [57]. In 2006, Ortiz *et al.* used template matching to map 70S ribosomes in whole-cell tomograms of *Spiroplasma melliferum* and found that, despite all contrast-rich features tending to correlate with the template, the method is feasible, but that only relatively large macromolecular complexes can be reliably detected [58].

It seems clear that there are a number of limitations here that exclude a true proteome atlas: template matching alone doesn't offer the required accuracy needed to faithfully detect more than the lowest-hanging targets; the preselection of targets of interest and use of PDB structures to generate templates reduces the scope of the results; and, finally, using tomograms of cellular subvolumes rather than entire cells fails to produce the full whole-cell spatial context promised by visual proteomics. Using machine learning, deep learning in particular, to make advancements in solving the particle-picking problem has recently become possible. This will be discussed after a brief tangent on Euler angles.

1.3.3 Euler angles

For some parts of the analysis, it will be useful to understand the conventions for defining particle position and orientation in structural biology, particularly in RELION [59]. This data is almost always stored by means of a Self-Defining Text Archive and Retrieval (STAR) file, a flexible file format where most data is tabular and columns are defined by name rather than by position [60–63]. After refining a set of subtomograms, the resulting STAR file contains the list of particles, with the source tomogram and x, y, and z coordinates (in pixels) for each. For each dimension, the origin offset (in Å) is also provided—the shift required to translate the particle into alignment with the reference. RELION follows the 3D Image Conventions [64] and defines a right-handed intrinsic coordinate system for orientating particles via rotation by three successive (ZYZ) Euler angles. The fact that the coordinate system is intrinsic means that the axes of rotation are defined with respect to the axes within the object to be rotated, rather than motionless global axes. The first rotation, "rot", sometimes also called ϕ (phi), is about the z-axis, up to a maximum of $\pm 180°$. The second rotation, "tilt", sometimes also called θ (theta), is about the new y'-axis, up to a maximum of $180°$. The third and final rotation, "psi", sometimes also written as ψ, is about the new z''-axis, up to a maximum of $\pm 180°$. While translations (e.g. origin offsets) shift observations into the reference projection, orientations (e.g. Euler angles) in a RELION STAR file rotate the reference into observations (i.e. particle instances) [59].

In Figure 1.3, an illustration of this rotational framework is shown. The original system x, y, z (in red) becomes orange after the first rotation ϕ. Unchanged axes are shown as dashed lines of alternating colour. The orange system x', y', z' goes through rotation θ, becoming green system x'', y'', z''. The resulting system x''', y''', z''' after the final rotation ψ is shown in blue.

One way to think about how these rotations actually affect a biological structure is to imagine the coordinate space in a particular shape and consider the impact of motion throughout it. Since there is only one rotation not about the z-axis, this is the only opportunity to tilt the structure away from the z-axis. The first rotation about the

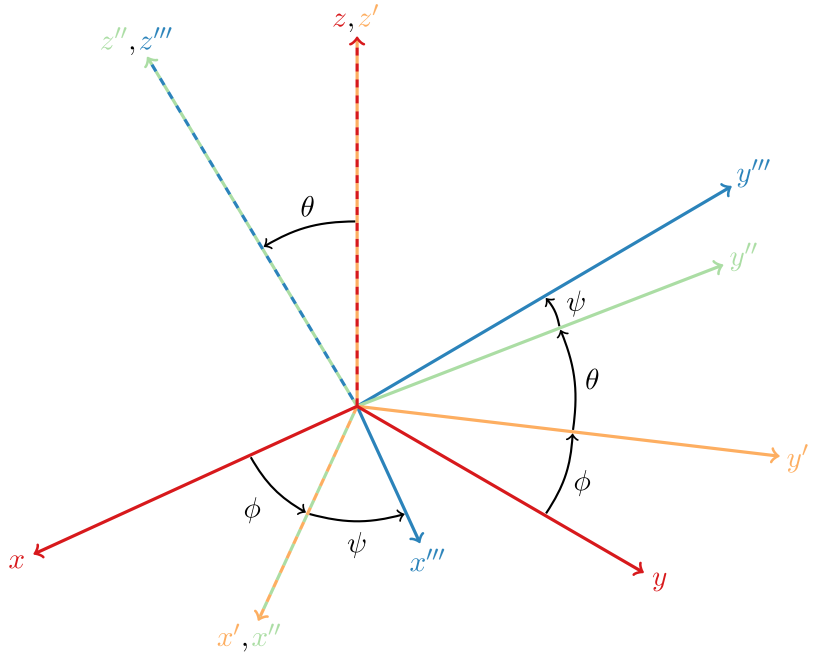


FIGURE 1.3: An illustration of rotation in 3D space using intrinsic Euler angles. Although ZYZ rotations are used in this thesis (and the field of 3D electron microscopy in general), this figure depicts ZXZ rotations. Conceptually, however, there is no difference. This figure was inspired by a similar one by James Diebel [65].

z-axis, then, has to be for the purpose of aligning the tilt axis. Imagining a protein structure sitting at the top pole of a hollow sphere, such that the part of the structure in contact (the lowest z-slice) cannot be removed from the sphere, only slid around it, the first rotation "rot" about z moves the xy-plane into the position where the next rotation "tilt" about y gets the desired tilt. Since the "rot" rotation was exclusively for the purpose of aligning the tilt, there now has to be a final rotation "psi" about z to achieve the correct final orientation.

1.4 Neural networks and deep learning

1.4.1 General principles and architectures

In general, machine learning (ML) refers to the use of any methods that allow for the prediction of new properties of data based on known properties discovered from the data [66]. In other words, a model should be able to generalize from its experience [67]. In order to maximize this ability, the model and the underlying data should have matching levels of complexity, thereby avoiding underfitting in the case of overly simple models and avoiding overfitting in the case of overly complex models [68]. The determination of this optimum is also known as the bias–variance tradeoff: large bias can be from a failure to learn the relevant features of the data [69], large variance can be from attempts to model the noise in the data [70], and both typically cannot be eliminated at once. This applies to both supervised and unsupervised ML. Supervised learning occurs when a labelled training dataset (i.e. set of input–output pairs) is used to inform the ML model, which then predicts labels (outputs) for new inputs. Classification and regression are the two most common forms of supervised learning. Unsupervised learning, on the other hand, uses a dataset of only inputs (i.e. unlabelled data) and finds structure and commonalities in the data. Principal component analysis (PCA) and clustering are the two most common forms of unsupervised learning.

Each of these methods in ML differs in the way it models the observations in order to generalize. Artificial neural networks, or just neural networks (NNs) for short, are one such ML model inspired by the structure and function of biological neural networks found in animal brains [71]. A neural network is composed of nodes (neurons) and edges (synapses) connecting them, such that signals can be transmitted through the network. A neuron receives numerical inputs, combines them in some way, and then produces an output to be propagated to connected downstream neurons. Neurons and edges have properties such as weights that act as parameters in the model and are updated during training in order to modulate the signals in an appropriate way to achieve a particular downstream task. Neurons are typically grouped into layers that, at least in most cases, communicate using only inputs from the previous layer, do not exchange signals within the layer, and only send output to the next layer. The first layer of a neural network is the input layer, which receives the data the network should use to make predictions. The intermediate layers of a neural network are called the hidden layers, and these are where transformations of the inputs take place. The final layer of a neural network is the output layer, which produces a prediction for the given input data and allows it to be read out.

The output value of a neuron at any point takes the sum of the neurons feeding it input weighted by the weights of the edges making those connections. In order to add non-linearity to the model, however, and model complex relationships, neurons are also given activation functions, through which the output is passed before propagating to the next neuron [72]. The activation function can be as simple as the rectified linear unit (ReLU) or logistic function. Without this, the seemingly complex model would ultimately reduce to a single linear function. The process of going from input layer to output layer, calculating the inputs and outputs of each neuron along the way, is called forward propagation, and is how a NN makes predictions from input data. To train a NN, the model calculates predictions for inputs with paired target outputs, evaluates the loss function (a metric to measure how different the targets and predictions are), and uses an algorithm called backpropagation to update edge weights backwards through the network proportionally to their effect on the error (the partial derivative of the error with respect to the weight) [73].

Deep learning refers to a particular scope of NN architectures involving multiple hidden layers. Due to the added power afforded by the hidden layers, increasingly complex problems can be modelled and solved. In traditional ML, it is necessary to extract and select features from the dataset prior to training a model. In deep learning, on the other hand, raw data are given directly as the input, and the features are learned as part of the model [74].

In deep learning, a convolutional neural network (CNN) is a type of NN that uses layers of convolution filters to create features, which works especially well on image data in particular, with applications such as image classification, semantic segmentation, or object detection. A CNN has many types of layers with different functions, and the overall architecture of the network can be adjusted depending on the nature of the data and patterns to be modelled. Convolutional layers perform convolution operations on the data using parametrized filters—matrices that, when overlappingly tiled across the input data, produce a weighted sum of the field in range. Many filters are usually created in each convolutional layer, and the ultimate goal of each filter is to recognize some feature in its receptive field. While convolutional layers increase the amount of data, pooling layers downsample the feature maps produced by convolutional layers by grouping neighbouring data elements into blocks and simplifying each block into a single output element. The most common pooling operation is maxpooling, which simply reduces each block into the largest element of that block. Convolutional layers and pooling layers are often alternated in a CNN, incrementally generating feature complexity and downsampling the data. A fully connected layer, also known as a dense layer, connects every neuron in one layer with every neuron in the next layer, and it is common to use one or more fully connected layers before the output layer in CNNs, in order to make predictions using the derived complex features. The final layer in a CNN is often a fully connected layer with the softmax function, a function that maps class-wise numerical outputs into a coherent probability distribution over the output classes, which can then be interpreted as the confidence for each output. Regularization is a method of preventing overfitting, which is generally done by limiting the model complexity unless it results in much greater performance. This is often implemented in ML by adding an extra term to the loss function that penalizes parameter complexity; for deep learning, it is more common to use dropout regularization: layer outputs are ignored or dropped at random during training, which also has the effect of making the model more robust.

LeNet was one of the first CNNs and has a simple architecture [75, 76]. For single-channel, two-dimensional inputs of handwritten digits, LeNet had the following setup:

convolutional layer with sigmoid activation, pooling layer, another convolution layer with sigmoid activation, another pooling layer followed by flattening, three fully connected layers of decreasing size with sigmoid activation, finally ending at a ten-class output, corresponding to the digits to be recognized from the handwriting. Different types of CNNs have been developed with varying architectures, strengths, and weaknesses, and the choice of which to apply depends on the dataset and task at hand. Other examples of CNNs include AlexNet [77], VGGNet [78], and ResNet [79].

A U-Net is a type of CNN originally developed by Ronneberger *et al.* in 2015 [80] that uses a unique architecture and performs very well on image segmentation. The "U" in U-Net refers to the shape of the architecture: a contracting path at first (much like a standard CNN), followed by an expansive path. The contracting path goes through repeated application of convolution and downsampling layers, but doubling the number of feature channels at each downsampling step. The expansive path then consists of repeated applications of upsampling and convolution, which reduces the number of features and increases the resolution, to which the result from the same level of the contracting path is then concatenated to the feature map. This concatenation step is known as "copy and crop" and transfers information from the downsampled representation of the input image to the upsampled representation, which allows the network to use both high- and low-level features of the image in its output.

Many claim that CNNs are black boxes that lack interpretability, but this is in fact not the case [81]. Perturbation-based approaches (observing the effect on output from a change in input) and backpropagation-based methods (applying backpropagation to check the relative importances of different parts of the input) are two ways to help gain insight into a model's inner workings [82]. Interpretability is not a well-defined concept, however, and it is difficult to strike a balance between pure but complex logic (e.g. a very deep decision tree) and tautological definitions (e.g. *post hoc* explanations like "x is an A because it's most similar to other instances of A") [83].

1.4.2 For protein folding

Predicting the three-dimensional structure of a protein from just its amino-acid sequence has long been a challenge in the field of bioinformatics, computationally difficult due to the truly vast number of conformations that a chain of amino acids can potentially have [84]. Homology modelling has traditionally been the approach of choice, using aligned templates with known structure and similar sequence to the target to infer the target's structure [85, 86]. Recently, OpenAI released AlphaFold, a deep-learning-based method for predicting protein structures [87, 88]. AlphaFold uses a novel NN architecture, using a multiple sequence alignment (MSA) and pairwise features as inputs. The most important component of the overall architecture is a module called Evoformer, which uses the pairwise residue features as boundaries on solving a graph inference problem in 3D space. Evoformer creates improved representations, which are then fed back through again as inputs. A structure module transforms the iteratively updated MSA and pairwise features into a protein structure, first modelling the protein backbone and then additionally predicting finer features of the model, such as side chain angles. The output structure from this module is fed back in to the Evoformer module along with updated inputs multiple times, known as recycling. In the very last step, a relaxation of the structure by gradient descent in a molecular dynamics force field is applied, leading to better stereochemical fit.

This was a greatly simplified explanation of AlphaFold just to demonstrate the complexity of the deep-learning-based approach. In "Critical assessment of methods of protein structure prediction—Round XIV" (CASP14) [89], a competition for the accurate modelling of protein structures from their sequences, AlphaFold structures performed much better than other competitors. Compared to the benchmark, AlphaFold results had a median backbone RMSD of 0.96 Å and an all-atom RMSD of 1.5 Å. Potentially the nicest feature of an NN-based approach, AlphaFold also provides accurate per-residue confidence estimates along with the output.

1.4.3 For tomogram denoising

Cryo-ET data often has a very low signal-to-noise ratio (SNR), especially at higher resolutions [90]. Additionally, the filtered backprojection (FBP) algorithm used for tomogram reconstruction can increase high-frequency noise [91]. Although the human eye is very capable when it comes to pattern recognition, even in noisy data, the levels of noise in these tomograms can exceed this ability, meaning that tomogram denoising can be helpful for picking particles manually or for curating particles picked in another way. The simplest way of denoising a tomogram is to apply a linear filter to its frequency space such as a low-pass filter for removing high frequencies. These kinds of filters are not context-aware, may introduce some blurring or even give rise to artefacts, and recently more advanced methods using neural networks have become popular [48]. It is important to note that, while denoising can and should be used to help with the naked-eye interpretation of raw data, some signal will ultimately be removed along with the noise, which means their use should be discouraged in downstream computations such as subtomogram averaging, in order to retain the most high-resolution data.

In order to train a neural network to denoise a tomogram, it seems logical that input–output pairs of noisy and noise-free data samples would be used to model the removal of the differences between the images. Lehtinen *et al.* published Noise2Noise in 2018 [92], however, which proved that images could be restored using only corrupted examples, upon which a model of the noise-free image can be built. This is useful for cryo-ET, since there exist no noise-free samples on which to train a model.

Noise2Noise is based on a U-Net [80]—a 3D U-Net [93], more specifically—with a depth (levels of downsampling before upsampling) of six. Instead of adapting the architecture and loss function, Noise2Noise simply operates as it would in a U-Net with input–output pairs of noisy and noise-free samples, but with the noise-free samples swapped for noisy samples. This works because the network cannot actually transform one noisy sample into another: for each pair of examples, there would be no way to predict the random target noise. Although the targets are chaotic, the weight gradients during each step of training are smooth due to the Gaussian distribution of the noise throughout the pixels, and the network therefore does converge to parameters that achieve the best average output given the zero-mean Gaussian noise: the noise-free image. Lehtinen *et al.* find that, with some additional tweaks, they can even perform better with only noisy targets over noise-free targets.

In 2019, Tegunov and Cramer published Warp [94], a cryo-EM software package that automates as much of the workflow as possible, from motion correction in data acquisition to CTF correction to particle picking and tomogram denoising. For data acquired as movie frames, one can create independently noisy observations of the underlying data by simply aligning and averaging different sets of frames for the same micrograph output. Splitting them into odd and even frames keeps the number of frames the same between sets and also distributes the later frames (with more radiation damage) equally between sets. The same concept also works at a coarser level for data acquired as tilts, instead using odd and even tilts as the independent sets. During tomogram reconstruction, the user can opt to produce deconvolved tomograms made with just odd and just even frames/tilts, which can then be used to train Noise2Map, the Noise2Noise implementation included with Warp. A trained model is included by default with Warp, but one should train a new model for every new dataset for best performance.

Topaz-Denoise [95] was released by Bepler *et al.* in 2020 as an addition to the Topaz particle-picking pipeline [96]. They also adopt a 3D U-Net [80, 93] architecture with some modifications. Although already outdated, more information on these and other methods in content-aware image restoration for electron microscopy (CARE) are examined in a 2019 book chapter by Buchholz *et al.* [97]. One newer example from 2022 is Noise-Transfer2Clean [98] by Li *et al.*, yet another U-Net approach—this time first selecting patches of pure noise in enhanced tomograms, using them to train a generative adversarial network (GAN) to synthesize noise, and training the denoising U-Net on clean–noisy pairs generated by adding synthetic noise—which can perform better by avoiding the Noise2Noise hypothesis that noise is zero-mean and independent and identically distributed.

1.4.4 For particle picking

Particle picking is one of the tightest bottlenecks in this workflow. Even with the best data and best processing, the number and accuracy of known locations of the particle of interest can make the difference between an interpretable structure and a low-resolution blob. Template matching is the most standard particle-picking approach, whereby the cross-correlation of a particle template is calculated across positions and orientations in each tomogram and the top peaks are selected, which works reasonably well for tomograms of sufficient SNR and contrast [99], although there can be problems with overwhelming numbers of false positives. Recently, deep-learning-based approaches to particle picking have become more popular, especially using CNNs due to their inherent strength in solving image-based problems. DeepPicker [100] and DeepEM [101] were the first deep-learning-based methods for particle picking, using simple CNNs and requiring manually picked particles for training and using the unpicked background as negative training examples. Since then, Topaz [96] has been released, which also uses a simple CNN architecture but considers particle picking as a positive–unlabelled problem, allowing for a smaller training dataset and faster training. Warp [94] was also packaged with its own implementation of a ResNet [79] to segment a micrograph into three classes: background, particle, and high-contrast artefact. Using a methodology called "you only look once" (YOLO) [102], crYOLO [103] is a deep-learning-based particle-picking approach that needs only a single pass of the full image, as opposed to several passes of cropped regions, and is therefore both faster and more sensitive to the particle's context. So far, these methods all deal with 2D data: classifying particles in micrographs based on 2D projections of particles, which is mostly useful for single-particle cryo-EM. In order to pick particles in 3D in tomograms, without simply discarding the valuable 3D context available, more complex approaches had to be developed.

In 2021, Moebel *et al.* released DeepFinder [104], the first 3D deep-learning-based approach to particle picking that doesn't require structural information of the particle. DeepFinder creates training masks for each training tomogram, using the known coordinates and orientations of each particle of interest to paste into the tomogram either a stencil (shape-based) in the shape of that particle or simply a sphere of similar size to the particle (sphere-based). Both strategies suffer from label noise: the average shape of a particle (or especially just a sphere) does not capture the potential variability in particle conformation. DeepFinder samples 3D patches from the training tomograms and trains a multi-class 3D U-Net [80, 93] using the corresponding patch from the training mask as the target label. Patches are sampled only where there are target labels present, avoiding training on too much background. Patches are also bootstrapped by sampling more for under-represented classes, thereby reducing class imbalance. Finally, training data is also augmented by adding jitter (random shifts) and applying a 180° rotation at random to the patches. The trained U-Net combines features at different spatial resolutions, conserving both global and local information. After training, in prediction mode, a tomogram is broken into patches and fed into the trained model. The resulting predictions are reassembled into a segmentation map that assigns a class label to each voxel in the tomogram. These multi-class segmentation maps must then be converted into particle lists: neighbouring voxels are merged into 3D connected components and the centroid of each cluster of voxels is reported as the location of that particle.

DeePiCt [105] is another 3D deep-learning-based approach to particle picking, also using a 3D U-Net for semantic segmentation, but additionally incorporating a 2D U-Net for cellular segmentation. The integration of these two greatly helps exclude false positives based on the localization of the particle of interest. Both DeePiCt and DeepFinder will be discussed in more detail in Section 2.5.

1.5 State-of-the-art workflow

As mentioned in Section 1.1 and Figure 1.1, the novel workflow that I have developed will be presented in this thesis alongside a proof of principle in the form of an uncharacterized membrane-associated complex from whole-cell tomograms of *Mycoplasma pneumoniae*. Based on the background information in this introductory chapter, the challenges and bottlenecks involved are manifold. Any bottleneck in the workflow needs to be eliminated as best as possible, such that the workflow can be scaled up and work in parallel, unlocking the potential for true visual proteomics. Locating enough particles in the noisy data to create a density map of sufficiently high resolution is the first such challenge, which is complicated by the fact that the particles don't appear equally distributed in position or orientation. Manual picking is tricky and time-consuming and with low sensitivity, but automated methods reduce the specificity and result in sometimes overwhelming numbers of false positives to curate manually. This bottleneck will be solved by a specific iterative training method. With a satisfactory density map, the next challenge is characterizing the identity of the protein or proteins in the particle. Even with the reduced genome of *M. pneumoniae*, manually investigating every protein in the proteome to assess suitability would be a challenge. In the workflow, this bottleneck is overcome by reducing the search space of brute-force procedures, an example of which is the use of membrane shaving and mass spectrometry to obtain a list of proteins with extramembranous enrichment when identifying the membrane-associated particle of interest.

1.6 Outline

In Section 2.2 of this thesis, I present the methods I applied and the results I obtained processing the raw cryo-EM data and reconstructing tomograms.

In Section 2.3 of this thesis, I present the methods I applied and the results I obtained picking particles manually to generate a template and performing template matching on the tomograms using a template. In Section 3.1, I discuss the reasons that template matching is required and the different options that a user of this workflow has in addition to manual picking.

In Section 2.4 of this thesis, I present the methods I applied and the results I obtained denoising tomograms using Noise2Map [94] and Topaz-Denoise [95]. In Section 3.1, I discuss the relevance of denoising to the workflow and the relative performance of each denoising tool.

In Section 2.5 of this thesis, I present the methods I applied and the results I obtained picking particles with DeePiCt [105] and DeepFinder [104]. In Section 3.1, I discuss these important results in detail, including strategies they inform, such as stencil optimization and iterative retraining.

In Section 2.6 of this thesis, I present the methods I applied and the results I obtained refining and classifying particle-containing subtomograms with RELION [59]. In Section 3.2, I discuss these results, the problem with angle distribution, and the best obtained density map.

In Section 2.7 of this thesis, I present the methods I applied and the results I obtained folding structural models of *M. pneumoniae* proteins, narrowing down candidates using orthogonal experimental data, and fitting these candidate structures into the density map obtained from refined cage particles. In Section 3.3, I discuss these results and their validation and present the unique assembly of the three proteins identified to make up the trimeric bulk of the cage complex.

In Section 2.8 of this thesis, I present the methods I applied and the results I obtained analysing transposon-sequencing data for *M. pneumoniae.* In Section 3.3, I discuss these results and their implications on essentiality of the genes involved in the cage complex.

In Section 2.9 of this thesis, I present the methods I applied and the results I obtained using both sequence-based and structure-based tools for homology searches of the genes involved in the cage complex. In Section 3.3, I discuss these results and the potential they have to shine a light on the function of the cage complex.

In Section 2.10 of this thesis, I present the methods I applied and the results I obtained investigating the cellular distribution and preferred orientation of the particle of interest. In Section 3.2, I discuss the lack of significant evidence for orientation preference between closely neighbouring particles and what the distribution of cluster sizes means.

Chapter 2

Methods and Results

2.1 Sample preparation and data collection

A cryo-ET dataset from past *M. pneumoniae* work [50] was used for this project. While the methods and results inherent to this thesis follow in the proceeding sections, the details of sample preparation and data collection are important for context, and should therefore first be explained. The sample preparation and data collection were both performed by Liang Xue. The methods described in this section are adapted from his PhD thesis [106].

The *M. pneumoniae* strain M129 (ATCC 29342) was provided by Jörg Stülke's group at the University of Göttingen. Cells were cultivated at 37 °C in cell-culture flasks with modified Hayflick medium [107]: 14.7 g/L Difco PPLO (BD, USA), 20% (v/v) Gibco horse serum (New Zealand origin, Life Technologies, USA), 100 mmol/L HEPES-Na (pH 7.4), 1% (w/w) glucose, 0.002% (w/w) phenol red and 1000 U/mL freshly prepared penicillin G.

Quantifoil gold grids with holey carbon support films (R2/1, 200 mesh; Quantifoil Micro Tools, Germany) were sterilized by ultraviolet irradiation for 30 minutes, glow-discharged for 45 seconds, and finally sterilized for 10 minutes, before being placed into a cell-culture dish with modified Hayflick medium. The medium was inoculated and cells were cultured at 37 °C, thus *M. pneumoniae* was grown directly on cryo-EM grids. The culture time was controlled to be less than 20 hours to make sure cells are in the fast-growing phase before vitrification. Grids were then quickly washed with PBS solution with protein A–conjugated gold beads (10 nm, Aurion, Netherlands), blotted from the back, and plunge-frozen by submersion in a liquid ethane/propane mixture using a manual plunger manufactured at the Max Planck Institute of Biochemistry in Germany.

For antibiotic-treated samples, the procedure was the same as above, except that drugs were added into the culture medium 15–20 minutes before freezing. For chloramphenicol (Cm; Sigma-Aldrich), an antibiotic acting via inhibiting the bacterial ribosome, the final concentration in the medium was 0.5 mg/mL. For pseudouridimycin (PUM; AdipoGen AG, Switzerland), an antibiotic acting via inhibiting bacterial RNA polymerase, the final concentration in the medium was 0.4 mg/mL.

Cryo-ET data were collected on a Titan Krios 300 keV transmission electron microscope (ThermoFisher Scientific). SerialEM [108, 109] was used for data collection. Before tilt-series acquisition, grid and grid square maps were acquired for cell position selection, at 135× and 2250× magnifications, respectively. Tilt series were collected with the dose-symmetric scheme [110]. Images were recorded in dose-fractionation counting mode and raw frames were saved. Frames were motion-corrected with the SerialEM plugin `alignframes` on-the-fly to generate tilt series, and `mdoc` files were created to store relevant information.

A Gatan K2 Summit direct detector camera was used with magnification 81 000×, calibrated pixel size on the specimen 1.7005 Å, targeted defocus range 1.5–3.5 µm, tilt angle range −60° to +60°, tilt increment 3°, constant dose of approximately 3 $e^-/Å^2$ for all tilts, and total dose of approximately 120 $e^-/Å^2$. All tilt series were acquired without a phase plate, except for 14 tilt series of untreated cells imaged with the Volta phase plate (VPP) [53]. The VPP data were primarily used for visualization and generation of data-driven references for template matching. Alignment and operation of VPP were carried out as described previously [111].

Tomograms with ice contamination or particularly poor fiducials were excluded from the datasets. Without the phase plate, there were 421 tilt series: 356 untreated and 65 Cm-treated. With the phase plate, there were 14 tilt series: all untreated. At all stages of data processing, these three datasets were handled independently and without any pooling.

2.2 Data processing and tomogram reconstruction

Each tilt series was aligned using gold-bead fiducials in Etomo (v4.9.x), a program in the IMOD software package [26, 112]. Tomograms for manual particle picking were also reconstructed in Etomo with default settings and voxel size 6.802 Å. Tomograms with significant ice contamination or with more than three bad tilts were excluded, in total representing less than 5% of tilt series collected.

The tilt series were then imported into Warp (v1.0.7b) [94] and a spatially resolved contrast transfer function (CTF) was estimated in 2D for each tilt micrograph. After importing the `xf` alignment file from Etomo for each tilt series into Warp, the CTF for each tomogram was estimated, this time for each tilt series. A 4×-binned tomogram with voxel size 6.802 Å and dimensions 928 × 928 × 450 was reconstructed in Warp for each processed tilt series. Unbinned tomograms with voxel size 1.7005 Å and dimensions 3712 × 3712 × 1800 were not reconstructed.

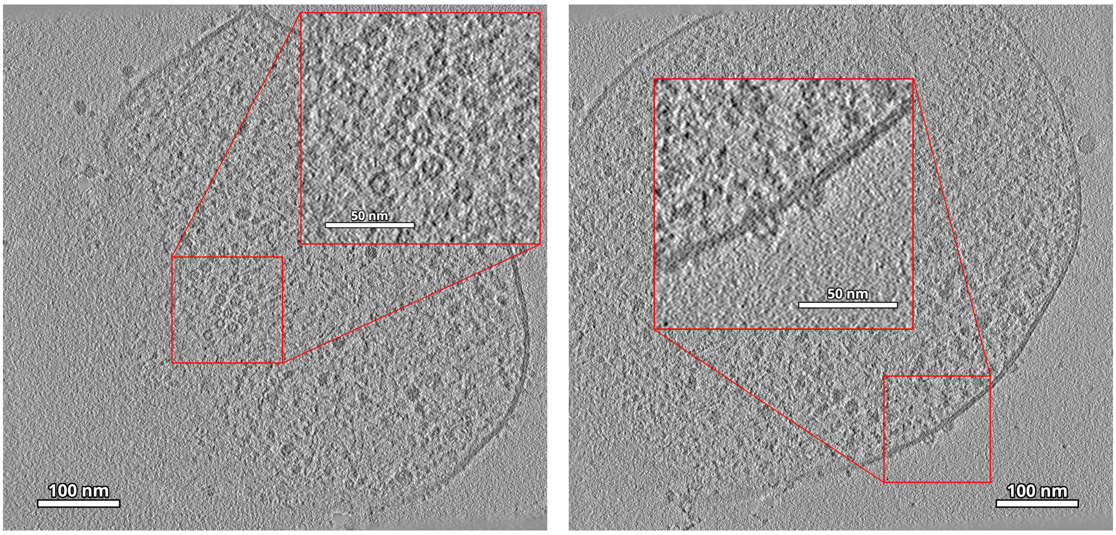
Since these datasets were originally used by Liang Xue for his *M. pneumoniae* expressome study [50], they underwent five rounds of iterative average-based multi-particle refinement in M (v1.0.7b) [113] using refinement in RELION (v3.0.7/v3.0.8) [59] of ribosomes from these tomograms to update tomogram deformation models and CTF parameters. This applies to all tilt series except the 14 VPP ones.

2.3 Particle picking and template matching

Readily visible in high-contrast VPP tomograms, we noticed a recurring feature always in the proximity of the cell membrane: a density on the extracellular side of the membrane, appearing ring-shaped when viewed from the top or bottom (i.e. a z-slice parallel to the membrane) or as a dome-shaped bulge when viewed from the side (i.e. a z-slice perpendicular to the membrane). In Figure 2.1, top views and side views of these particles in VPP tomograms of untreated *M. pneumoniae* cells are shown. With a size somewhere between an average protein complex and a ribosome, and also with unknown identity, this particle seemed a good candidate for guiding the development of the workflow.

Using the `e2spt_boxer` program in the EMAN2 software suite [114], the VPP dataset was used for manual picking of 278 particles across the 14 tomograms. After subtomogram extraction in Warp and multiple refinement attempts in RELION, the resulting density was still disappointing and likely not resolved enough to be used as an effective template for template matching on the other datasets. Various views of this template, after filtering and trimming, are shown in Figure 2.2.

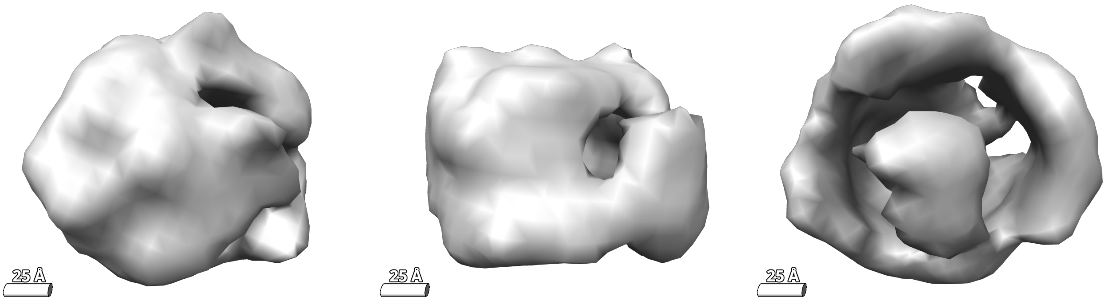
During the work on ribosome refinement carried out by Liang Xue, it was also noticed after a round of classification that some ribosomes in tomograms of Cm-treated cells are seemingly associated with our distinctive membrane complex. Due to the cage shape of our particle of interest, it will be referred to henceforth as simply the cage. A better alignment, driven by the larger ribosome, resulted in a better template of the cage simply by trimming the ribosome away from the average for the class of cage-associated



(A) Top views of the particle of interest in a slice of tomogram `00029`.

(B) Side views of the particle of interest in a slice of tomogram `00054`.

FIGURE 2.1: The particle of interest is strikingly visible with the naked eye in VPP tomograms reconstructed in Etomo. Top views of the particle show a ring shape parallel to the cell membrane, while the particles appear as extracellular bulges on the membrane when viewed from the side.



(A) View of the top (extracellular side) of the template.

(B) View of the side of the template.

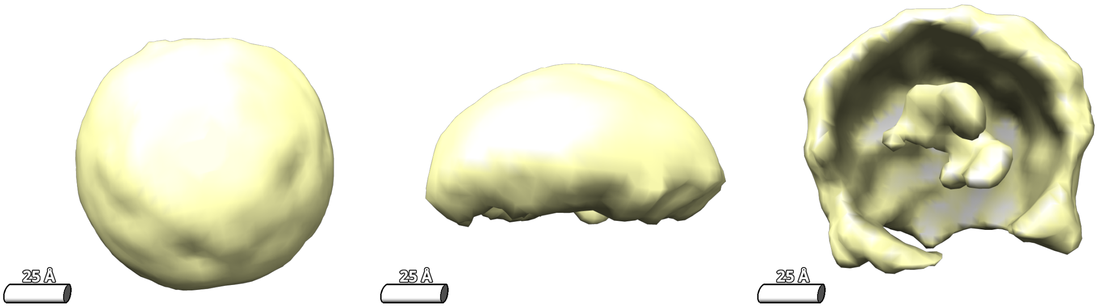
(C) View of the bottom (intracellular side) of the template.

FIGURE 2.2: Various views of the template generated by manually picking instances of the particle of interest in VPP tomograms and aligning and averaging them.

ribosomes and applying a low-pass filter. Various views of this new template are shown in Figure 2.3.

Using PyTom [115] for TM with this new template (box size 32 px with pixel size 6.802 Å), with a spherical mask with a radius of 15 px and smoothing of 2 px, and with rotational search strategy `angles_19.95_1944.em`, the top 400 peaks for each tomogram in the Cm-treated dataset were extracted with a minimum distance between peaks of 26 px and a minimum distance from the tomogram edge of 20 px. Peak extraction, as well as initial curation and visualization, was performed in TOM toolbox [116] in MATLAB.

Results from TM often contain large numbers of false positives, caused both by particles



(A) View of the top (extracellular side) of the template.

(B) View of the side of the template.

(C) View of the bottom (intracellular side) of the template.

FIGURE 2.3: Various views of the density used as a template for template matching generated by trimming the ribosome away from the average of a subclass of ribosomes found to associate with the particle of interest.

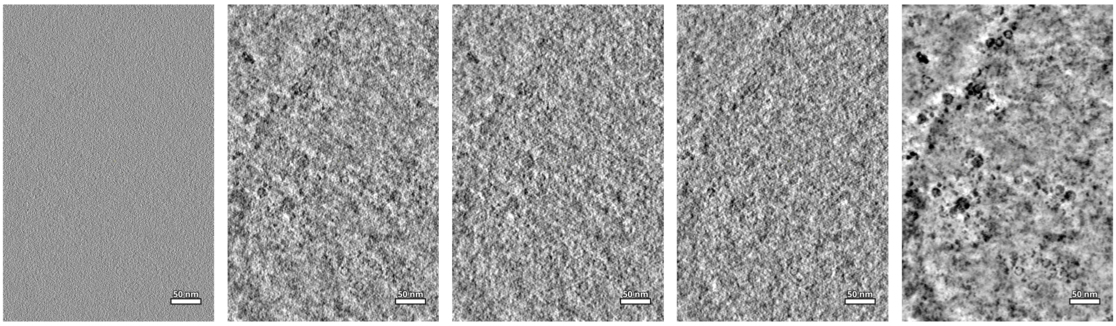
with commonalities and simply by high-contrast elements in the tomogram, such as ribosomes, membranes, or gold-bead fiducials [58, 117, 118]. Lukas Adam, an intern in the group of Julia Mahamid, curated the particle lists manually. This resulted in a total of 1263 identified particles across 55 tomograms.

2.4 Tomogram denoising

In order to obtain more particles, it was decided to use DeePiCt [105], a CNN-based tool for particle picking, under development at the time by Irene de Teresa-Trueba. This will be described in detail in Section 2.5. It was already known internally that the network performs better with denoised data, so it was necessary to first denoise the *M. pneumoniae* tomograms.

The obvious choice for denoising cryo-ET data in 3D, especially since preprocessing was carried out in Warp [94], was to use Noise2Map, a modified version of the Noise2Noise algorithm [92], packaged with Warp. During the phase of tomogram reconstruction in Warp, in addition to the raw reconstructions used for downstream processing, one can opt to save deconvolved reconstructions made from all tilts, only the odd tilts, and old the even tilts. The deconvolution operation that Warp performs artificially boosts the lowest frequencies in the data, resulting in sharper object boundaries in the output, particularly useful for particle picking. The denoising model is trained by learning to recognize noise based on the differences between the deconvolved tomograms from only the odd and only the even tilts.

I first tried to train a Noise2Map model from scratch, but since the tomograms are loaded into memory for training, the memory limits the number of tomograms that can be used,

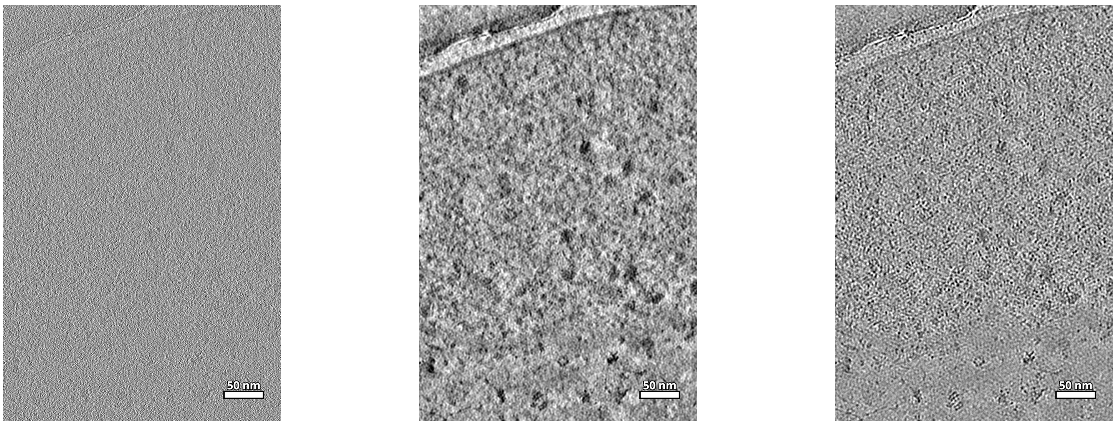


(A) Raw. (B) Deconvolved. (C) Deconvolved using only odd tilts. (D) Deconvolved using only even tilts. (E) Denoised with Noise2Map.

FIGURE 2.4: Tomogram denoising using Noise2Map (Warp [94]). Different versions of the tomogram are shown in each panel. The view shown is always the same region of z-slice $^{128}/_{450}$ of tomogram `00255`.

and the denoising results after training on only a few tomograms weren't encouraging. Instead, I used a pretrained model (`noisenet3dmodel_256_20200915_174215`) provided by Dimitry Tegunov, the developer of Warp. Despite being trained on *M. pneumoniae* tomograms of a different pixel size, the method seems robust enough to produce acceptable output. Denoised output for an example tomogram is shown in Figure 2.4. Comparing the deconvolved version to the original raw version (Subfigures 2.4B/2.4A), the deconvolution already helps a lot with making features more apparent. The denoised version (Subfigure 2.4E) is a step better, removing the majority of grainy noise and enhancing the visibility of larger features in the tomogram. The only downside is a slight 'washed-out' feeling in the image now, where high-resolution information is clearly lacking. This is why, although deconvolved and denoised tomograms are useful for particle picking and visualization, raw tomograms should always be used for extraction for particle refinement. The filtering and denoising process can remove the data needed for high-resolution sub-tomogram averaging, and deep-learning methods can introduce non-existent information into the data (e.g. from training bias).

I also tried the 3D denoising component [95] of Topaz [96], a software package developed for 2D and 3D cryo-EM denoising and particle picking. Topaz's `denoise3d` module is also based on the Noise2Noise framework [92], using noisy pairs of observations of the same signal. With the help of Frosina Stojanovska in the group of Judith Zaugg, a model was trained on a small subset of the *M. pneumoniae* tomograms. Model training was much slower than Noise2Map, regularly exceeding the maximum job time on the HPC cluster. An example of the result of applying the trained model to our data is shown in Figure 2.5. As seen by comparing the deconvolved and denoised tomograms (Subfigures 2.5B/2.5C), there isn't a visual advantage in the denoised version; in fact, it seems that some noise is also enhanced, making it harder to discern features.



(A) Raw.

(B) Deconvolved.

(C) Denoised with Topaz-Denoise.

FIGURE 2.5: Tomogram denoising using Topaz-Denoise [95]. Different versions of the tomogram are shown in each panel. The view shown is always the same region of z-slice $^{231}/_{450}$ of tomogram 00070.

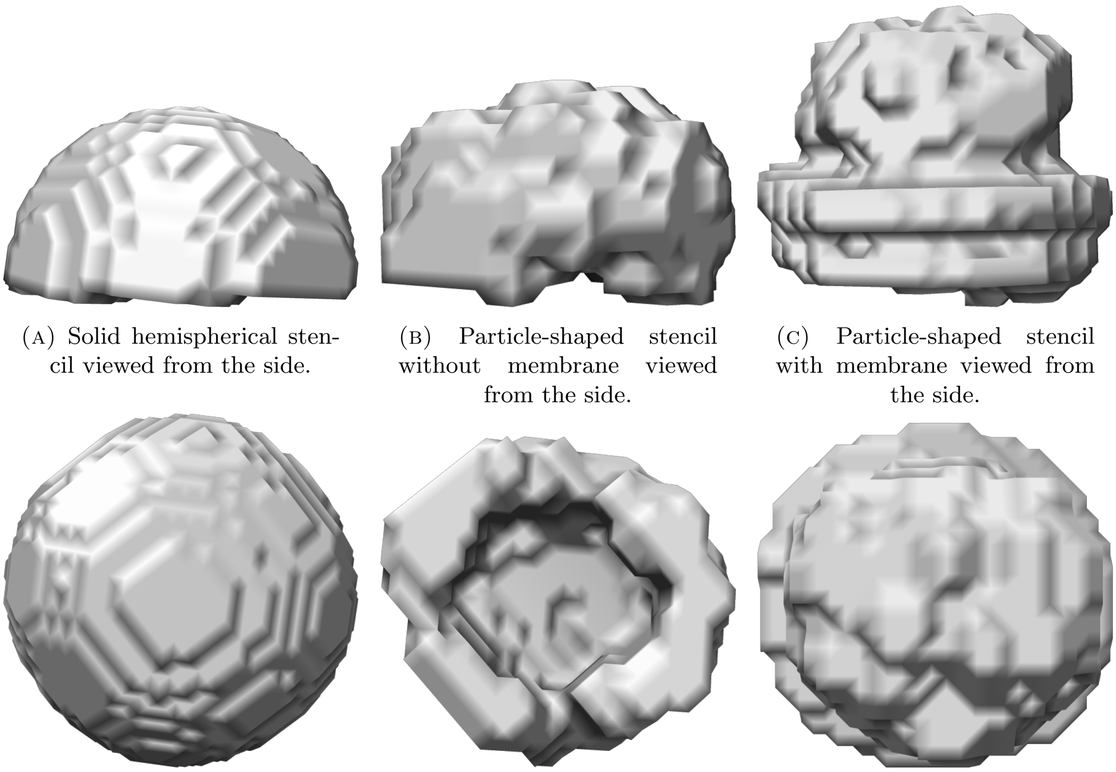
In conclusion, it was decided to use the Noise2Map-denoised tomograms for particle picking.

2.5 Neural networks for particle picking

DeePiCt [105] is a CNN-based tool for segmentation and particle picking, which, at the time of this work, was under development by Irene de Teresa-Trueba, a postdoctoral fellow in the groups of Julia Mahamid and Judith Zaugg. It has since been completed and published alongside a benchmarking study proving its functionality on various complexes and organelles in *Schizosaccharomyces pombe* tomograms [105]. The particle localization component of DeePiCt is based on the 3D U-Net [93], itself an extension of the original 2D U-Net architecture [80]. In addition to the tomogram image data, training this network requires tomogram masks—binary tomograms with the same dimensions as the tomogram image data—for each tomogram and for each semantic class to be trained, where the voxel value is 1 in the subvolumes to be used for training positive examples and 0 otherwise. The pipeline generates training data by saving subvolumes (of adjustable size) of the input tomogram image data centred wherever the corresponding location is masked in the input mask. The training data is then used to train the network (of adjustable depth and number of initial features) for a selected number of epochs. Once training is complete, subvolumes are generated for the tomograms for which predictions should be made, in the same style as the generation of training data, but instead spanning the full tomogram rather than only the masked portions. For each prediction subvolume, the network predicts the probability that it represents a subvolume whose centre is within an instance of each semantic class. For each semantic class, a probability map of the same dimensions as the input tomogram is then reconstructed. Typical postprocessing steps involve thresholding the probability maps to create binary prediction masks, clustering predicted voxels into contiguous blocks of approximately the size of the particle, and determining the coordinates of the centres of these particles.

Before progressing to complex training or even to parameter optimization, I first wanted to test the software to ensure the inputs are correctly read and the output is as expected. A new DeePiCt model was trained on 53 tomograms (`00254`–`00317`) and evaluated on 2 tomograms (`00318`–`00319`). In order to generate the binary mask with the same dimensions as the tomogram, a small binary mask on the order of particle size, which will be referred to as a stencil, was pasted into an empty tomogram in every location containing an instance of training data. Orientation manipulations using Euler angles were performed using the SciPy (v1.7.x/v1.8.x/v1.9.x) [119] and `mrcfile` (v1.4.x) from CCP-EM [120] libraries in Python. In this case, the curated particle list from template matching on Cm-treated tomograms was used, and the stencil was simply a solid sphere with a radius of 16 px (109 Å) centred at each set of particle coordinates. Using denoised tomograms as the image data, a box size of 64 px, a box overlap of 12 px, the model with a depth of 2 and 32 initial features was trained with a batch size of 5 for 150 epochs with a training–validation split of 0.8, batch normalization active, no encoder dropout, decoder dropout of 0.2, no data augmentation, and no cross-validation. The resulting probability map was thresholded at probability 0.5, and voxels were clustered with a clustering connectivity of 3 to a minimum cluster size of 100 and no maximum cluster size. For automated particle-picking statistics, a tolerance radius of 10 px was used. The resulting probability maps and statistics were acceptable, so I proceeded with optimizing the parameters.

The shape of the stencil is clearly important in training and making predictions. At a minimum, since whatever is used for training will ultimately be predicted, it influences the shape emerging from a cluster of predicted voxels in the output: if one uses a spherical stencil, there will be spheres in the probability map. This doesn't necessarily have much of an impact on the accuracy or usability of the output, but it's fair to say that an optimal stencil for a given particle would likely be as big as possible without exceeding the boundaries of the particle—a particle likely has areas of higher and lower distinctiveness, and discarding image data from some areas of the particle would raise the chances of missing the former. As it's unclear exactly which elements should be included, however, a test was carried out with three different stencil shapes: a solid hemisphere facing the extracellular side of the membrane (Subfigures 2.6a/2.6d), a hollow particle-shaped stencil without the membrane (Subfigures 2.6b/2.6e), and a hollow particle-shaped stencil including the membrane (Subfigures 2.6c/2.6f). The



(A) Solid hemispherical stencil viewed from the side.

(B) Particle-shaped stencil without membrane viewed from the side.

(C) Particle-shaped stencil with membrane viewed from the side.

(D) Solid hemispherical stencil viewed from the top.

(E) Particle-shaped stencil without membrane viewed from the bottom.

(F) Particle-shaped stencil with membrane viewed from the top.

FIGURE 2.6: Two views of each of three stencils tested for generating DeePiCt training masks. The first column shows a solid hemispherical stencil to be positioned over the extracellular portion of the particle of interest. The second column shows a stencil with the shape of the extracellular portion of the particle with hollow interior, not including the membrane. The third column shows a similar stencil to the second one, particle-shaped and hollow, but instead also including the membrane.

stencils were generated with the help of Rasmus Kjeldsen Jensen, a postdoctoral fellow in the group of Julia Mahamid.

To test these three stencils, three new DeePiCt models were created, each with its own binary mask generated by repeated pasting of the respective stencil, using the curated particle list from template matching on Cm-treated tomograms. The models were trained on 53 tomograms (`00254`–`00317`) and evaluated on 2 tomograms (`00318`–`00319`). With denoised tomograms as the image data, a box size of 64 px, a box overlap of 12 px, the model with a depth of 2 and 32 initial features was trained with a batch size of 5 for 75 epochs with a training–validation split of 0.8, batch normalization active, no encoder dropout, decoder dropout of 0.2, no data augmentation, and no cross-validation.

Using tomogram `00318`, the three stencils were scored based on manually inspecting the probability map at each coordinate in the curated list of particles from template matching. A score of 0 was assigned to clear misses, a score of 0.5 to near misses, 1.0

to satisfactory detection, and 1.5 to good detection. The score results are shown in Table 2.1. Although there was only a negligible difference, the hollow particle-shaped stencils including the membrane performed marginally better, so DeePiCt training from this point will be with this stencil shape.

Stencil	Figure	Score
Hemisphere	2.6A/2.6D	45.0
Particle without Membrane	2.6B/2.6E	44.5
Particle with Membrane	2.6C/2.6F	45.5

Table 2.1: The three stencils tested and their resulting scores when evaluated against curated particles in tomogram `00318`. The theoretical maximum score is 61.5.

In order to determine whether denoised tomograms are actually needed for DeePiCt, I tried instead training a DeePiCt model using deconvolved tomogram image data. The model was trained on 53 tomograms (`00254`–`00317`) and evaluated on 60 tomograms (`00070`–`00074` and `00254`–`00319`). The binary masks were generated using the hollow particle-shaped stencil including membrane. With deconvolved tomograms as the image data, a box size of 64 px, a box overlap of 12 px, the model with a depth of 2 and 32 initial features was trained with a batch size of 5 for 75 epochs with a training–validation split of 0.8, batch normalization active, no encoder dropout, decoder dropout of 0.2, no data augmentation, and no cross-validation.

Image Data	Clear Misses	Near Misses	Satisfactory	Good	Score
Denoised	6	1	12	22	45.5
Deconvolved	0	3	10	28	53.5

Table 2.2: Denoised and deconvolved tomograms and their resulting scores when evaluated against curated particles in tomogram `00318`. The theoretical maximum score is 61.5.

Just as with evaluating stencils, using tomogram `00318`, the deconvolved results were scored based on manually inspecting the probability map at each coordinate in the curated list of particles from template matching. A score of 0 was assigned to clear misses, a score of 0.5 to near misses, 1.0 to satisfactory detection, and 1.5 to good detection. The score results are shown in Table 2.2, including old results for comparison. Assessed by random visual inspection, other tomograms for which predictions were generated also showed improvements along the same lines. Since using deconvolved tomogram image data works significantly better, any future DeePiCt training will be with deconvolved tomograms.

With the intent to this time predict particles for the whole dataset of tomograms, especially the untreated tomograms for which the most data exists, a new DeePiCt model was trained. The model was trained on 53 tomograms (`00254–00317`) and evaluated on 411 tomograms (`00032–00051`, `00067–00208`, `00240–00319`, and `00447–00678`). The binary masks were generated using the hollow particle-shaped stencil including membrane. With deconvolved tomograms as the image data, a box size of 64 px, a box overlap of 12 px, the model with a depth of 2 and 32 initial features was trained with a batch size of 5 for 75 epochs with a training–validation split of 0.8, batch normalization active, no encoder dropout, decoder dropout of 0.2, no data augmentation, and no cross-validation.

The resulting probability maps were thresholded at probability 0.7, and voxels were clustered with a clustering connectivity of 3 to a minimum cluster size of 300 and no maximum cluster size. In total, 21 022 particles were predicted across all 411 tomograms. Through inspecting the remaining particles while adjusting the score threshold, it was empirically determined to be 3500, leaving 6741 particles. The particles were then extracted as subtomograms in Warp [94] and subtomogram averaging was performed in RELION [59]. Details on this process will be given in Section 2.6.

Especially since the predictions were made on more tomograms than used for training, we can benefit from improved model performance through iterative rounds of training. In Figure 1.1, I presented a diagram of the general iterative workflow for particle picking and refinement, which may help to supplement this dense technical text with the high-level idea. I therefore trained a new DeePiCt model, this time both training and evaluating on the same 411 tomograms (`00032–00051`, `00067–00208`, `00240–00319`, and `00447–00678`). The binary masks were generated using a new hollow particle-shaped stencil including membrane, derived from the newly refined average from the extracted subtomograms from the last DeePiCt round, and of course using the particle coordinates and orientations from the refinement. Tomograms `00582` and `00617` had had no predicted particles after thresholding in the last DeePiCt round, leading to their binary masks for this round being completely empty, in turn meaning they were effectively ignored for training here. With deconvolved tomograms as the image data, a box size of 64 px, a box overlap of 12 px, the model with a depth of 2 and 32 initial features was trained with a batch size of 5 for 150 epochs with a training–validation split of 0.8, batch normalization active, no encoder dropout, decoder dropout of 0.2, no data augmentation, and no cross-validation.

The resulting probability maps were thresholded at probability 0.7, and voxels were clustered with a clustering connectivity of 3 to a minimum cluster size of 300 and no maximum cluster size. In total, after this instance of retraining, 23 714 particles were predicted across all 411 tomograms. Using a new empirically determined score threshold of 4500, I was left with 10 070 particles. After some manual curation, including removing clear false positives and adding some obvious misses, there were 8500 particles. The

Probability Threshold	Clustering Connectivity	Minimum Cluster Size	Maximum Cluster Size	Particles in 00318	Total Particles
0.7	3	300	—	79	23 714
0.8	3	300	—	74	21 761
0.9	3	300	—	66	19 832
0.9	3	500	1000	5	2602
0.9	3	1000	2000	4	2527
0.9	3	2000	4000	5	2965
0.9	3	4000	8000	18	7855
0.9	3	8000	16 000	24	1741
0.9	3	16 000	—	4	48

Table 2.3: The number of particles detected by DeePiCt's clustering algorithm in tomogram 00318, as well as in all 411 tomograms, for each experimental set of clustering parameters.

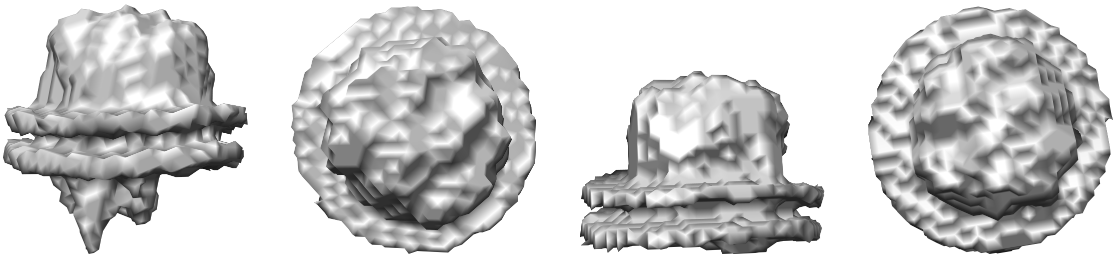
particles were then extracted as subtomograms in Warp [94] and subtomogram averaging was performed in RELION [59].

One problem that became apparent at this point was that, despite very accurate probability maps from this round of retraining, this didn't translate to accurate coordinates or particle counts in the resulting particle lists. Since instances of our particle of interest often appear in tight groupings in the imaged cells, the thresholded probability maps in these cases can have two or more particles with overlapping voxels. The clustering algorithm used by DeePiCt sees them as continuous and predicts one particle located in between them. The coordinates of this particle are often not useful, since the box used for subtomogram extraction likely won't be big enough to capture any of the particles the coordinates came from. We also observe the opposite problem: when predictions for an instance of a particle are not so strong, it can happen (before or after thresholding) that the cluster of voxels becomes discontinuous. In this case, two particles will be predicted where there should have been only one. I performed some experimentation by varying clustering parameters in DeePiCt, and the results are shown in Table 2.3. The probability threshold is the cutoff at which probabilities are binarized before clustering: anything lower than the cutoff becomes 0 and anything else becomes 1. The clustering connectivity refers to the dimensions in which adjacency is counted for the purposes of merging neighbouring blocks of voxels. In three dimensions, a clustering connectivity of 3 means all diagonal neighbours are also counted (i.e. within a step of at most one pixel in *each* dimension). The minimum and maximum cluster sizes are the limits applied to the size of clusters found in the algorithm.

Again taking tomogram 00318 as an illustrative example, as shown in Table 2.3, we lose some predicted particles when increasing the probability threshold, but even at a probability of 0.9, there are 66 particles—greater than our known minimum of 41 particles from the curated list. Looking at the different subranges of cluster size, we see that they tend to be higher than 4000. This makes sense when considering the size of the stencil (i.e. the number of active voxels in the stencil) is approximately 10 000. In the range above 16 000, there are only four results, and they are all from two merged particles due to issues with the clustering algorithm. In the 8000–16 000 range, we see 24 particles, and they all reference real particle positions, verified manually in a denoised tomogram. In the 4000–8000 range, we see 18 particles, which also all reference real particles. It's not surprising, however, that these particles are predicted with such high confidence, since they were the ones used for training, and predictions are now being carried out on the same tomograms as those used for training. Since the goal of retraining rounds is to teach a more varied picture of what particles can look like, it's important not to filter out lower-confidence predictions that may in fact be true positives. In addition, especially with a high probability threshold like this, it's important to inspect lower-scoring predictions, since they will often represent split particles. In order to help with this for future rounds of predictions, it was decided to move to a solid stencil rather than a hollow one, since at least then the likelihood of splitting predicted voxels into discontinuous clusters is reduced. Finally, since DeePiCt doesn't modify the clustering algorithm based on the minimum and maximum cluster sizes given as parameters, it seems there is no point filtering the cluster sizes in postprocessing, and this should rather be done as part of the curation step along with finding an appropriate score threshold.

After refinement of the particles predicted by the retrained network, the average was used to make a new stencil. As mentioned before, in order to avoid cases where weakly predicted particles end up split in two voxel clusters, a solid stencil will now be used. At this point, I also decided to test whether it makes a difference to include the intracellular area of the particle. Two different stencils are illustrated in Figure 2.7—one from the full particle (Subfigures 2.7A/2.7B) and one from just the extracellular and membrane portions (Subfigures 2.7C/2.7D)—and are to be compared in the second round of retraining. The stencils were generated with the help of Rasmus Kjeldsen Jensen. At the same time, I decided to test whether there's a significant difference between using raw tomogram reconstructions for the image data instead of the deconvolved tomogram reconstructions.

Four new DeePiCt models were trained, both training and evaluating on the same 411 tomograms (`00032–00051`, `00067–00208`, `00240–00319`, and `00447–00678`). The binary masks were generated using each of the two new solid particle-shaped stencils shown in Figure 2.7, derived from the newly refined average from the extracted subtomograms from the previous DeePiCt round (the first retraining round), and of course using the



(A) Solid full particle stencil, viewed from the side.

(B) Solid full particle stencil, viewed from the top.

(C) Solid extracellular particle stencil, viewed from the side.

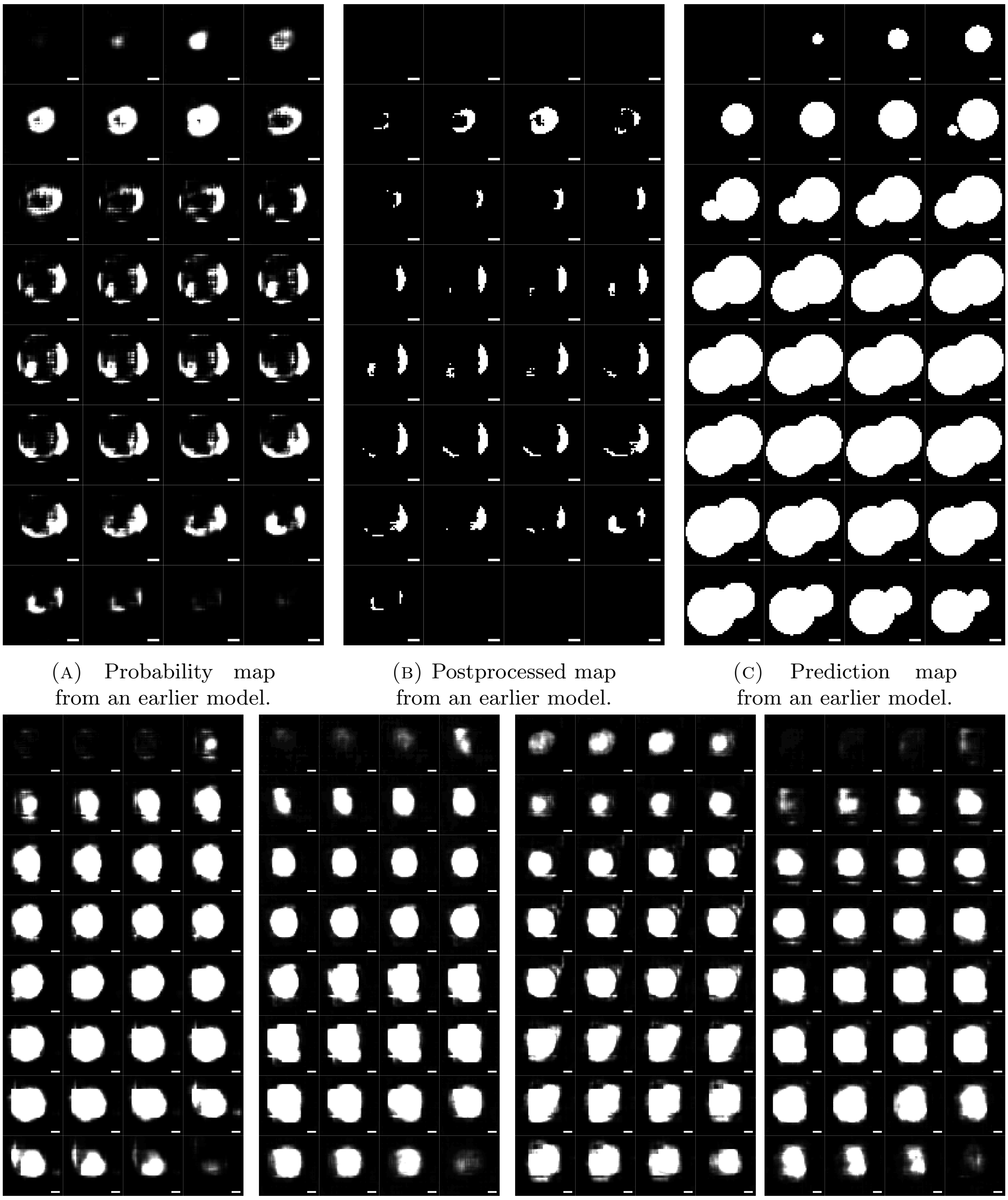
(D) Solid extracellular particle stencil, viewed from the top.

FIGURE 2.7: Side and top views of each of two solid stencils tested in the second round of retraining, one based on the shape of the full particle, and the other based on only the extracellular portion of the particle.

particle coordinates and orientations from this new refinement. With either raw or deconvolved tomograms as the image data, a box size of 64 px, a box overlap of 12 px, the model with a depth of 2 and 32 initial features was trained with a batch size of 5 for 150 epochs with a training–validation split of 0.8, batch normalization active, no encoder dropout, decoder dropout of 0.2, no data augmentation, and no cross-validation.

The solid stencil indeed helped with the problem of split particle predictions. Figure 2.8 shows an example of where this problem has been solved, using montages of prediction results from a past training round to compare to these newest training rounds. Subfigure 2.8A shows the old probability map, and it's notable that there are some weakly predicted voxels (the darker voxels) as well as some small gaps in the prediction. The binarized result after thresholding is shown in Subfigure 2.8B; the gaps are now much larger. After segmenting clusters of voxels and using their centroids as the coordinates for predicted particles, we can see in Subfigure 2.8C what the predictions look like by plotting solid spheres with a radius of 16 px centred at these coordinates: two particles are now predicted, with strongly overlapping spheres. In Subfigures 2.8D–2.8G, montages of probability maps for the same region are shown for the results of the four models just trained. In all four cases, we see that the solid stencil helped the probability map stay continuous for where a single particle should be.

With respect to the deconvolved–raw and extracellular–full comparisons, I manually inspected the predicted probability maps for a few tomograms from each of the four models, and it seemed that deconvolved and extracellular was the combination with the best performance. The probability maps using the stencil of the full particle seemed to focus a bit too much on the cell membrane, perhaps just because the extra intracellular part of the stencil allowed for more training data with membrane within the box. With the raw tomogram data, occasionally it seemed that a particle was missed compared to



(A) Probability map from an earlier model.

(B) Postprocessed map from an earlier model.

(C) Prediction map from an earlier model.

(D) Probability map from the new model, using deconvolved tomograms and the extracellular stencil.

(E) Probability map from the new model, using deconvolved tomograms and the full stencil.

(F) Probability map from the new model, using raw tomograms and the extracellular stencil.

(G) Probability map from the new model, using raw tomograms and the full stencil.

Figure 2.8: Montages of various maps, all for the same region in tomogram `00318`. A montage shows how a view of the xy-plane changes moving incrementally through the z-dimension.

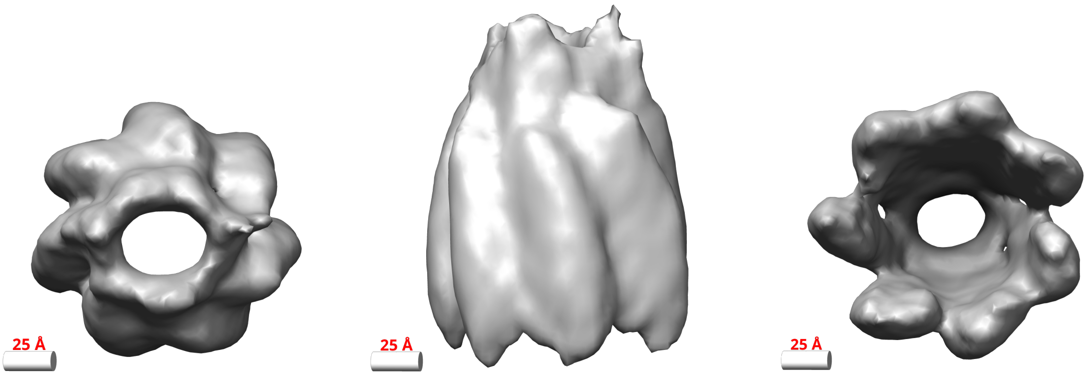
the deconvolved tomogram data, so this is why the deconvolved data and extracellular stencil became my choices going forward.

The resulting probability maps were thresholded at probability 0.9, and voxels were clustered with a clustering connectivity of 3 to a minimum cluster size of 300 and no maximum cluster size. In total, after this instance of retraining, 19 109 particles were predicted across all 411 tomograms. Using a new empirically determined score threshold of 4000, I was left with 11 906 particles. The particles were then extracted as subtomograms in Warp [94] and sent to RELION [59] for refinement. Again, details on this process will be given in Section 2.6.

I briefly tried experimenting with DeePiCt parameters to see how much of a difference box size and box overlap (stride) make. Another cause for this investigation was the appearance of some strange image artefacts with straight edges and right angles in the predicted probability maps, and I wanted to test whether this phenomenon was related to the box size and box overlap. I trained a DeePiCt model for each of the nine combinations of three box sizes—32 px, 48 px, and 64 px—and three box overlaps—6 px, 12 px, and 24 px. Unfortunately, only three of the nine jobs actually ran without a fatal error: box size 32 px with overlap 6 px, box size 48 px with overlap 6 px, and (the parameters from all previous DeePiCt runs) box size 64 px with overlap 12 px. This will require further investigation in the future.

I also tried another experiment to fix the problem with representation of side views. Due to the shape of the imaged *M. pneumoniae* cells, there were more top and bottom views than side views of the cage complex, and this orientation bias leads to anisotropic data and therefore reduced overall resolution. This will be discussed in detail in Section 2.10, but I wanted to see if it was possible to equalize the distribution. I trained two DeePiCt models with all the standard parameters but using subsets of the particle lists: ones with tilt angle 60°–120° and ones with tilt angle 40°–140°. The output from both had plenty of top and bottom views again, with the occasional extra side view, so I tried this same retraining technique as earlier, each time filtering for high-scoring predictions with appropriate tilt angle. After two rounds of retraining, the DeePiCt models hadn't improved sufficiently to pick side views, and so this issue seems unavoidable for now.

After refinement in RELION, which will be discussed in detail in Section 2.6, Rasmus Kjeldsen Jensen ran a round of classification in RELION, which subjects the particles to unsupervised clustering into a specified number of classes, and found that the refined average of one of the classes looked very different from the cage-like structure of our particle of interest. Different views of this seemingly heptameric protein complex are shown in Figure 2.9. As a working name from now on, it will be referred to as the heptamer. It contains seven likely identical copies of this banana-shaped protein or



(A) Heptamer viewed from the top.

(B) Heptamer viewed from the side.

(C) Heptamer viewed from the bottom.

FIGURE 2.9: Views of the surface rendered (at level 0.042) from the refined heptamer density with voxel size 3.401 Å, Gaussian-filtered with width 3.4 Å.

protein complex arranged through seven-fold rotational symmetry, as well as another rotationally symmetric cap on top, leaving a large circular hollow space in its centre. Due to how similar the cage and heptamer look from certain angles in tomograms, it's possible that DeePiCt had trouble differentiating between them. Another possibility is that heptamers were picked alongside cages as far back as template matching or even manual picking. Two important differences to point out between the cage and heptamer structures are that the cage is slightly bigger—approximately 160 Å in diameter compared to approximately 140 Å—and that their symmetries clearly differ—three-fold for the cage compared to seven-fold for the heptamer.

To test DeePiCt's ability to distinguish between the cage and the heptamer, I trained a new DeePiCt model on the approximately 1500 heptamer particles, both training and evaluating on the same 297 tomograms (00033–00051, 00067–00208, 00240–00251, and 00447–00677). The binary masks were generated using a binarized stencil derived from the average in Figure 2.9, and of course using the particle coordinates and orientations from the heptamer refinement. With deconvolved tomograms as the image data, a box size of 64 px, a box overlap of 12 px, the model with a depth of 2 and 32 initial features was trained with a batch size of 5 for 150 epochs with a training–validation split of 0.8, batch normalization active, no encoder dropout, decoder dropout of 0.2, no data augmentation, and no cross-validation. The resulting probability maps were thresholded at probability 0.7, and voxels were clustered with a clustering connectivity of 3 to a minimum cluster size of 300 and no maximum cluster size. In total, after this instance of retraining, 8632 particles were predicted across all 297 tomograms. Upon manual inspection of these predicted particles, it was clear that many cages had been misidentified as heptamers. I also tried making use of the fact that DeePiCt supports multiple semantic classes and trained cages and heptamers at once as independent classes. DeePiCt, however, simply trains one model per semantic class using only the labels for that semantic class, and therefore doesn't learn via implicit negative examples (i.e. that a cage is not a heptamer, and vice versa). Without some improvements, or at least much more experimentation, DeePiCt isn't suitable for maintaining a purely cage (or heptamer) particle list.

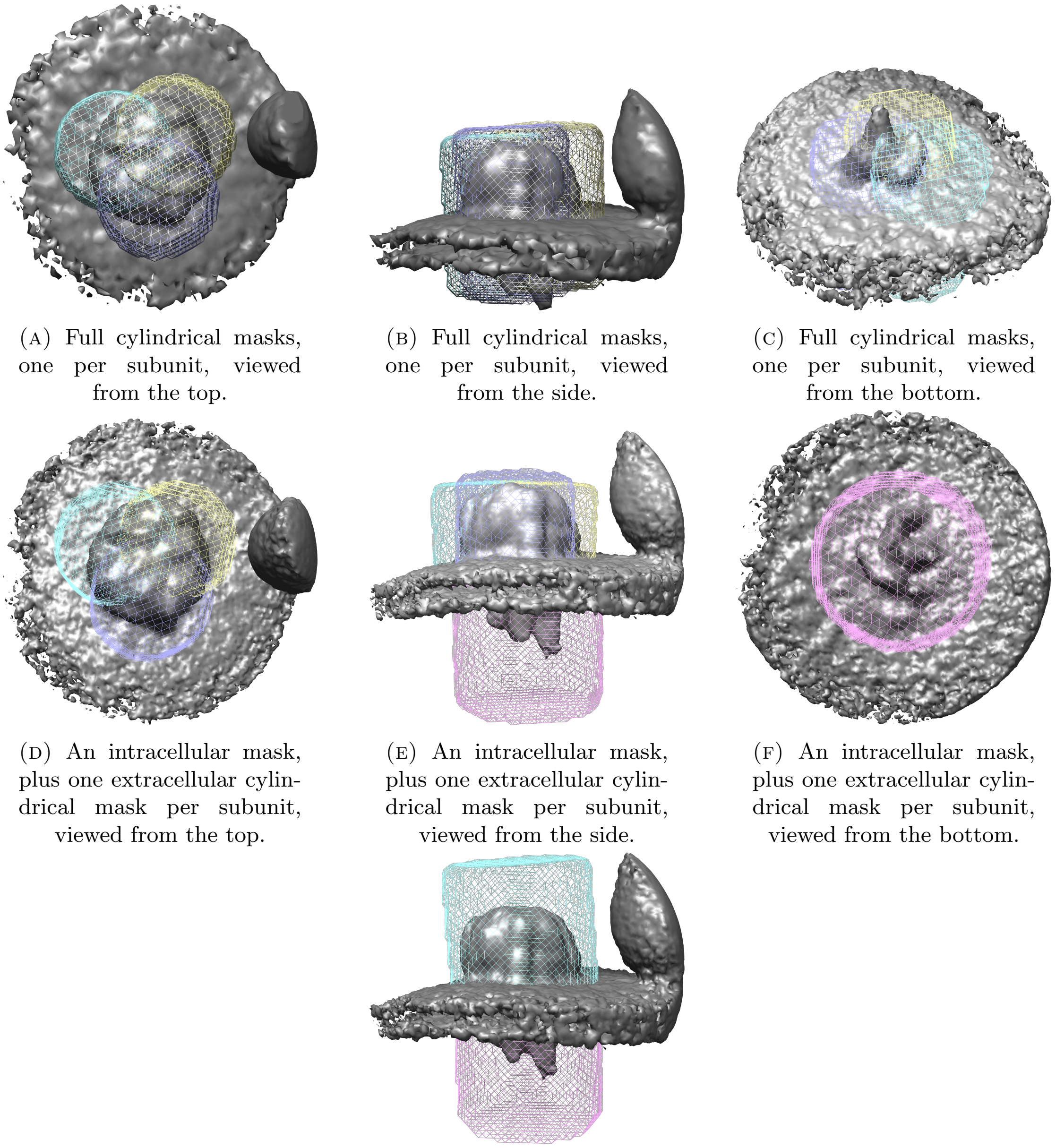
Using a tool called DeepFinder [104], another software package for particle picking in tomograms using a 3D U-Net [93], Rasmus Kjeldsen Jensen trained a model on the curated outputs from the second round of DeePiCt retraining. With DeepFinder, the model is truly multi-class, which means that a single model learns all classes at once, for each training example simultaneously encouraging the target class and discouraging the other classes. Additionally, Rasmus Kjeldsen Jensen manually picked instances of the Nap complex, a known transmembrane adhesion complex in *M. pneumoniae* [121] and *M. genitalium* [122]. Since we also had the ribosome particle lists from the previous work of Liang Xue, the model was trained on four classes: cages, heptamers, Naps, and ribosomes. The masks for training were generated, in much the same way as for DeePiCt, by pasting the appropriate binarized-average stencil using the coordinates and orientations for each particle into an empty volume for each tomogram. The masks are now not explicitly binary, but rather use 0 for the background and 1, . . . , n for the n non-overlapping semantic classes. Using deconvolved tomograms as the image data and patch size 60 px, the model was trained with a batch size of 25 for 100 epochs with 100 training steps per epoch and 10 validation steps per epoch. Patches were randomly sampled from the training and validation data where there was at least one annotated semantic class, with resampling applied to normalize class representation, random jitter within ±13 px in each dimension added to each training patch, and data augmentation by 180° rotation about the tilt axis with a likelihood of 0.5. Training was performed using 5386 cage particles, 3882 heptamer particles, 406 Nap particles, and 74 959 ribosome particles across 356 tomograms. Validation was performed using 233 cage particles, 185 heptamer particles, 27 Nap particles, and 1411 ribosome particles across 8 tomograms. Training the model took longer than DeePiCt, but the results were much better, with very little confusion between the four classes. Another advantage of DeepFinder was that it had no problem finding centroids in particle predictions, even with two or more particles very close to one another. In total, 25 431 cage particles and 20 005 heptamer particles were predicted across all tomograms.

2.6 Particle refinement and classification

From a list of predicted particle coordinates—from manual picking or template matching, as discussed in Section 2.3, or from a particle-picking CNN like DeePiCt or DeepFinder, as discussed in Section 2.5—the next step is to extract appropriate tomogram subvolumes (subtomograms) and align and refine them to produce a density map. Due to the experimental nature of the procedure, many classification and refinement attempts were made, including multibody refinement. In this section, I describe some general methods as well as some specific methods. Subtomogram extraction was always performed using Warp [94] using, until intermediate steps in the protocol, a box size of 64 px at pixel size 6.802 Å and particle normalization diameter of 180 Å. Once enough particles had been picked to benefit from increased resolution and more captured delocalized signal, a box size of 88 px at pixel size 3.401 Å was used. The subtomograms were averaged without alignment to generate an initial reference for running a consensus refinement in RELION (v3.0.8/v4.0-beta-1/v4.0-beta-2) [59]. Refinements were run in RELION iteratively using various particle masks and sampling strategies. All RELION work was performed with help from Rasmus Kjeldsen Jensen.

Before switching from DeePiCt to DeepFinder and solving the problems of contamination with heptamers and picking closely neighbouring cage particles, it didn't seem that increased numbers of cage particles did much to increase the resolution of the refined average. One potential explanation for this could be that the cage complex can adopt a number of different conformations. To assess this, multibody refinement was performed in RELION, which allows for various regions of the particle to be submasked and treated as independently moving rigid bodies [123]. The multibody masks used are shown in Figure 2.10. Having tried a few multibody jobs in RELION to see whether this would resolve detail in different areas of the average, it didn't help much except perhaps with highlighting some differences between the three proposed extracellular subunits.

Another issue could have been that the refined particle angles showed some strong tendencies. Due to the geometry of cells blotted for cryo-ET, there is more membrane at the tops and bottoms of cells than on the sides. Additionally, as discussed in Section 2.5, it also seems to be harder to pick side views in general. In Figure 2.11, the angular distribution is shown depicted as a sphere. The location on the surface of the sphere refers to a specific combination of two Euler angles (imagining the sphere as a globe, "rot" is longitude and "tilt" is latitude) required for the correct orientation of the particle, and the height and colour of the bars are based on the number of particles with that orientation. It's clear that the top and bottom views of the particle (tops and bottoms of the spheres) are more commonly represented in the dataset. Surprisingly, the few



(A) Full cylindrical masks, one per subunit, viewed from the top.

(B) Full cylindrical masks, one per subunit, viewed from the side.

(C) Full cylindrical masks, one per subunit, viewed from the bottom.

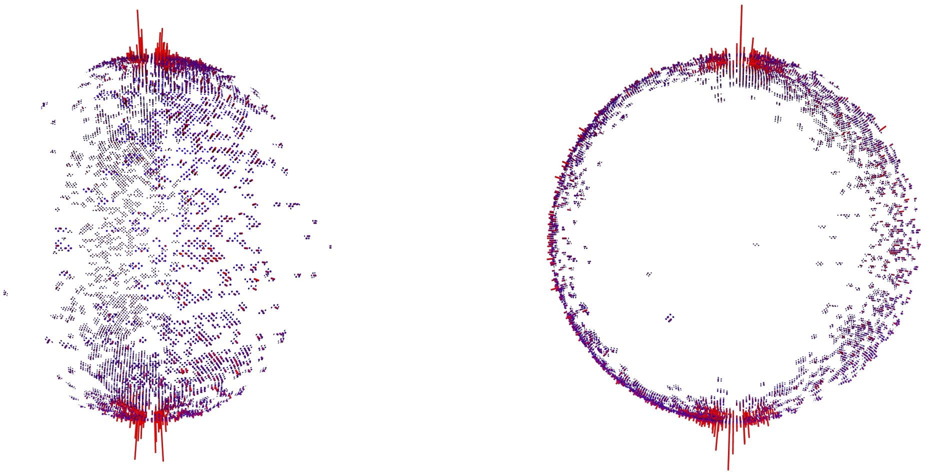
(D) An intracellular mask, plus one extracellular cylindrical mask per subunit, viewed from the top.

(E) An intracellular mask, plus one extracellular cylindrical mask per subunit, viewed from the side.

(F) An intracellular mask, plus one extracellular cylindrical mask per subunit, viewed from the bottom.

(G) An intracellular mask and an extracellular mask, viewed from the side.

FIGURE 2.10: Different views of combinations of submasks used for multibody refinements in RELION.



(A) Angular distribution sphere viewed from the front.

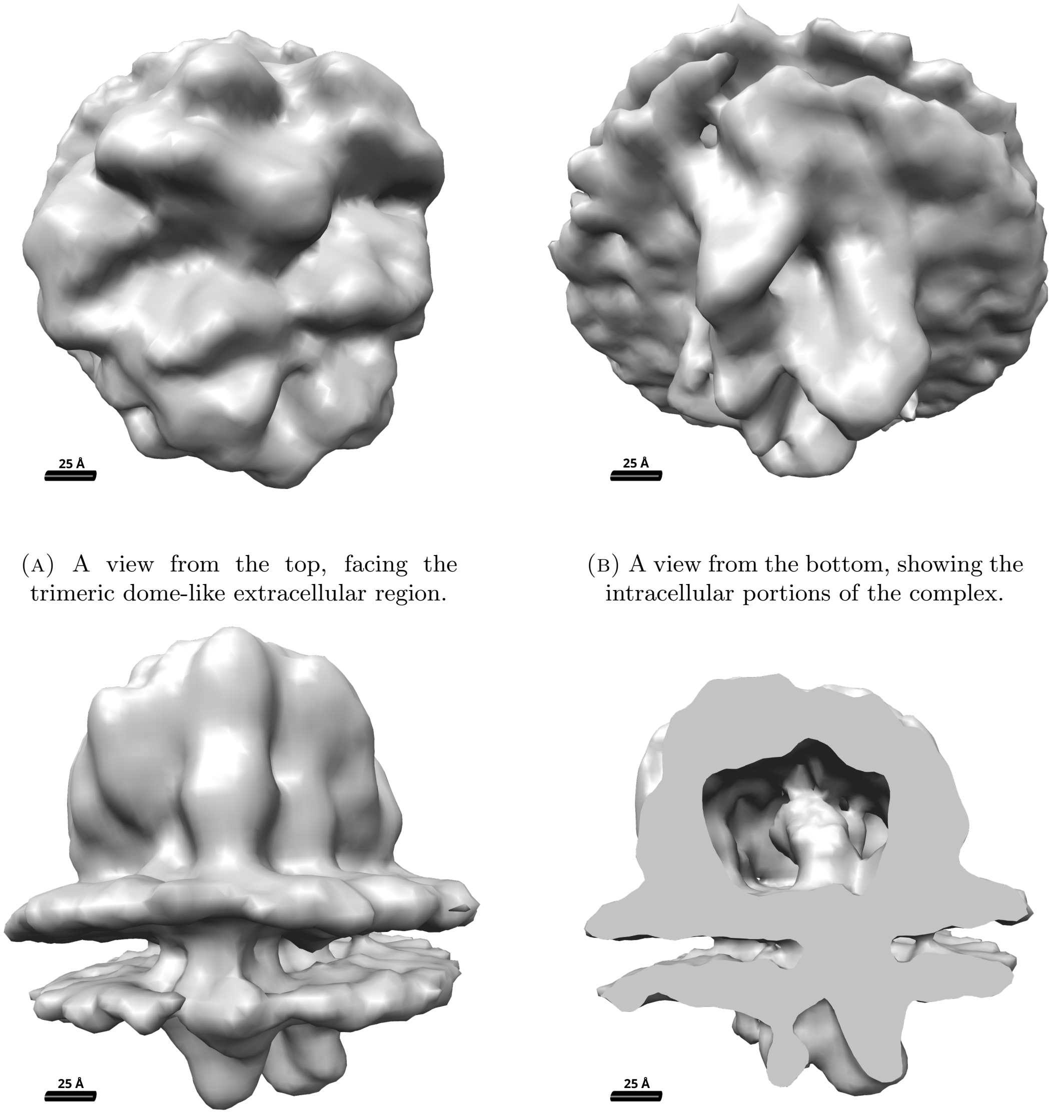
(B) Angular distribution sphere from Subfigure 2.11A rotated 90° about the y-axis.

FIGURE 2.11: Views of the angular distribution of particles after a round of refinement. The top and bottom of the spheres represent particles at the tops and bottoms of cells.

side views that do appear seem to have a “rot” bias (front and back of the sphere in Subfigure 2.11A; left and right of the sphere in Subfigure 2.11B).

With the final list of cage particles from DeepFinder, particles were averaged to obtain an initial reference for running a consensus refinement. The alignment parameters from the consensus refinement were then used to classify the particles into five classes. The best-refining class was used for a final round of refinement with a soft particle-shaped mask, giving the final alignment parameters and a density map refined to 11 Å as determined by a Fourier shell correlation (FSC) cutoff of 0.143. The resulting density map is shown in Figure 2.12.

It was also interesting to observe an additional large density approximately 65 Å away from the cage (when subtomograms had been extracted with sufficiently large box size) in some averages after classification. Since the cage particle visually does seem to appear in clusters, this extra density could be the edge of a neighbouring cage particle. An example using a box size of 296 px (503 Å at pixel size 1.7005 Å) is shown in Figure 2.13, where the cages sit averaged in the centre of the image, and the putative neighbouring cage complex appears to the right side of the box.



(A) A view from the top, facing the trimeric dome-like extracellular region.

(B) A view from the bottom, showing the intracellular portions of the complex.

(C) A view from the side, showing the lipid bilayer and transmembrane helices in the complex.

(D) A cutaway side view, showing the hollow extracellular dome.

FIGURE 2.12: Four views of the final cage density map of resolution 11 Å produced from refining the picked cage particles.

2.7 Identifying complex constituents

Efforts thus far have been for the purposes of uncovering a greater number cage particles in the tomogram data. Even with a high-resolution structure of the cage complex, it could be challenging to identify the constituent proteins without some further data to guide the search. In order to determine the identities of the proteins forming the cage complex, which sits partially outside the cell membrane, the outsides of cells can be 'shaved' with a protease like trypsin or proteinase K. After labelling the resulting

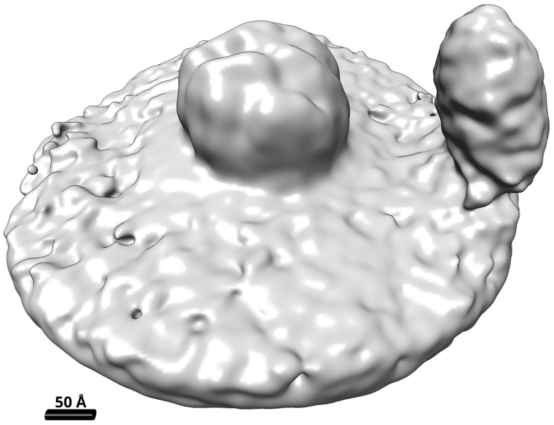


FIGURE 2.13: After refining cage subtomograms with a box size of 503 Å, an extra density at the edge of the box is resolved, appearing to be a neighbouring cage complex.

fragments of extracellular proteins with tandem mass tags (TMTs) to facilitate multiplexing, liquid chromatography followed by tandem mass spectrometry (LC–MS/MS) was used to identify the peptides present, map them back to their origin proteins, and calculate their relative abundances [124]. Combining the results from the trypsin and proteinase K experiments, there were a total of 117 proteins with detected enrichment, of which 101 were predicted to be membrane-associated (transmembrane proteins or lipoproteins).

Structures were predicted for this candidate list of proteins, first using AlphaFold2 [87, 88], and later also using the AlphaFold Protein Structure Database (AlphaFold DB) [125] once it was released for Swiss-Prot proteins in December 2021. The first relaxed model from each AlphaFold run was assessed for fit within the density map for the extracellular portion of the cage complex by rigid-body docking using a software package called PowerFit [126]. The top ten poses ranked by cross-correlation score for each protein structure were saved. The top-scoring protein was MPN643; looking through its top poses, however, the positions and orientations within the map were quite random, and the goodness of fit was likely due to its low size. In second place was MPN444; it seemed to fit much more snugly into the map and the top three poses actually corresponded to the three 120° rotations about the axis through the middle of the cage. These top three poses are shown within the context of the map in Figure 2.14.

Looking a bit more into MPN444, I found that it's an uncharacterized lipoprotein and a homologue of MG309 in *Mycoplasma genitalium*. According to Pfam [127], it contains Pfam entry PF12506, a domain of unknown function called DUF3713 that occurs only in the genus *Mycoplasma*. UniProt [128] similarly classifies MPN444 as part of the "MG307/MG309/MG338 family". There is not much information in the literature about

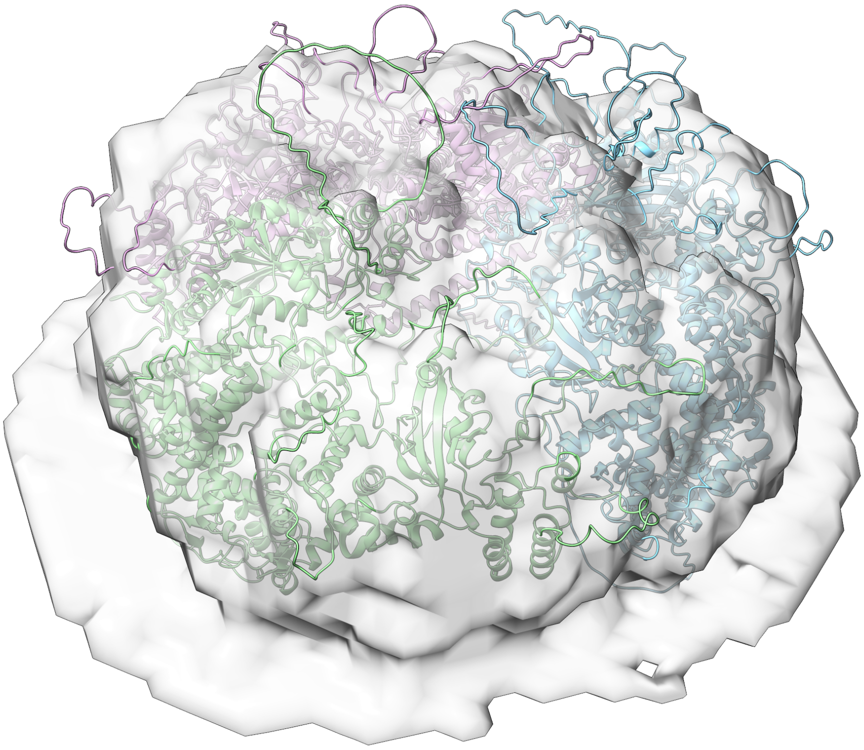


FIGURE 2.14: Superimposing the top three MPN444 fits from PowerFit on a density map of the cage complex, they correspond to 120° rotations about the particle's z-axis. Each pose is shown in a different colour.

this family, other than that the C-terminal portion of MG309 has some immunostimulatory capacity [129] and that all three can be used for strain typing of *M. genitalium* based on variable short tandem repeat (STR) sequences they contain [130].

In *M. pneumoniae*, UniProt lists nine proteins in the "MG307/MG309/MG338 family". There are three large proteins—MPN436, MPN444, and MPN489—each around 1300 amino acids, which uniquely map as homologues of the three *M. genitalium* proteins forming the name of the family. There are also six more of these proteins, much smaller than the other three and potentially fragmented versions thereof. MG307, MG309, and MG338, as well as their homologues MPN436, MPN444, and MPN489, are all lipoproteins whose first 26 or 27 residues form the signal peptide, which is cleaved off in the final protein alongside membrane anchoring via lipidation of the residue immediately after. In *M. pneumoniae*, the MPN436, MPN444, and MPN489 proteins are part of lipoprotein multigene family 3 [131] of unknown function since the genome was sequenced [1, 4]. More details on these proteins can be seen in Table 2.4.

MPN436 interestingly appears with the third-highest cross-correlation score in the PowerFit results, and MPN489 appears a bit further down the list in thirty-second place. The AlphaFold models of MPN436 and MPN489 are also very structurally similar to

M. pneumoniae Protein	Length (AA)	Mass (kDa)	*M. genitalium* Homologue	Length (AA)	Mass (kDa)
MPN436	1244	139	MG307	1177	132
MPN437	572	64	—	—	—
MPN438	345	37	—	—	—
MPN439	237	27	—	—	—
MPN440	726	81	—	—	—
MPN442	150	17	—	—	—
MPN444	1325	146	MG309	1225	138
MPN485	316	34	—	—	—
MPN489	1300	143	MG338	1270	142

Table 2.4: Proteins in *Mycoplasma pneumoniae* that are listed by UniProt [128] as part of the "MG307/MG309/MG338 family", along with the length and mass of each, and the length and mass of its homologue in *Mycoplasma genitalium*, when one exists.

that of MPN444. The largest pairwise difference in structural similarity is a root-mean-square deviation (RMSD) of 10.167 Å. All pairwise differences in structural similarity for aligned protein models, as well as the pairwise similarity of their sequences, are presented in Table 2.5.

	MPN436	MPN444	MPN489
MPN436	—	5.972 Å 39.5%	10.167 Å 35.4%
MPN444	—	—	6.536 Å 37.1%
MPN489	—	—	—

Table 2.5: Pairwise structural-similarity RMSDs of atomic positions (top line of cell) and pairwise sequence-similarity percentages (bottom line of cell) among the cage-forming candidate proteins.

Using crosslinking mass spectrometry (CLMS) [132] data originally from the *M. pneumoniae* expressome study [50], the AlphaFold models were assessed for accuracy.

For any internal (intra-protein) crosslinks identified in MPN436, MPN444, or MPN489, the distance between the involved residues was inspected in the AlphaFold model, and nearly all were verified to be within the maximum reach of the crosslinker. The totals for the best-scoring AlphaFold for each of the three proteins is shown in Table 2.6.

Using the same crosslinking dataset, but this time looking at external (inter-protein) crosslinks, we find that there are twenty such crosslinks involving MPN436, MPN444, or MPN489, of which five are directly between two of these proteins. Table 2.7 shows these twenty crosslinks with the five crosslinks between two of the cage candidates shown in bold.

Within Protein	Best Model	BS^3 Score	DSSO Score
MPN436	2	7/7	25/26
MPN444	3	4/4	32/33
MPN489	3	0/1	10/11

Table 2.6: For each of three cage candidates, the number of the best AlphaFold model (by relaxed number), based on the number of experimental crosslinks likely in the model. The maximum crosslinking distance between alpha carbons was considered 30 Å for both BS^3 and DSSO [133, 134].

From Protein	To Protein	From Residue	To Residue
AtpG	MPN489	59	881
MnuA	MPN436	178	759
MnuA	MPN436	73	576
MPN376	MPN436	206	919
MPN376	MPN444	522	490
MPN376	MPN489	561	710
MPN436	**MPN444**	**1062**	**176**
MPN436	**MPN444**	**1062**	**306**
MPN436	MPN523	98	161
MPN436	MPN523	98	170
MPN444	MgpA	490	98
MPN444	MPN400	490	158
MPN444	SecD	1058	271
MPN444	SecD	1058	73
MPN444	SecD	887	245
MPN489	**MPN436**	**1041**	**308**
MPN489	**MPN444**	**273**	**412**
MPN489	**MPN444**	**658**	**1151**
MPN489	MPN488a	1107	20
MPN489	MPN488a	43	90

Table 2.7: Inter-protein crosslinks involving the three cage candidate proteins. Shown in bold are crosslinks between two cage candidate proteins.

These five crosslinks uniquely define the MPN436–MPN444–MPN489 assembly. Each of the five crosslinks shows the proximity of one side of the structure of one of these proteins to the other side of the structure of another of these proteins. Figure 2.15 shows the only logical way to form this assembly given the crosslinking data. Protein abundance data for *M. pneumoniae* [135] on *Myco*Wiki [136] also corroborate this: with 57, 48, and 48 copies per cell on average of MPN436, MPN444, and MPN489, respectively, there are nearly equal expression levels of these proteins.

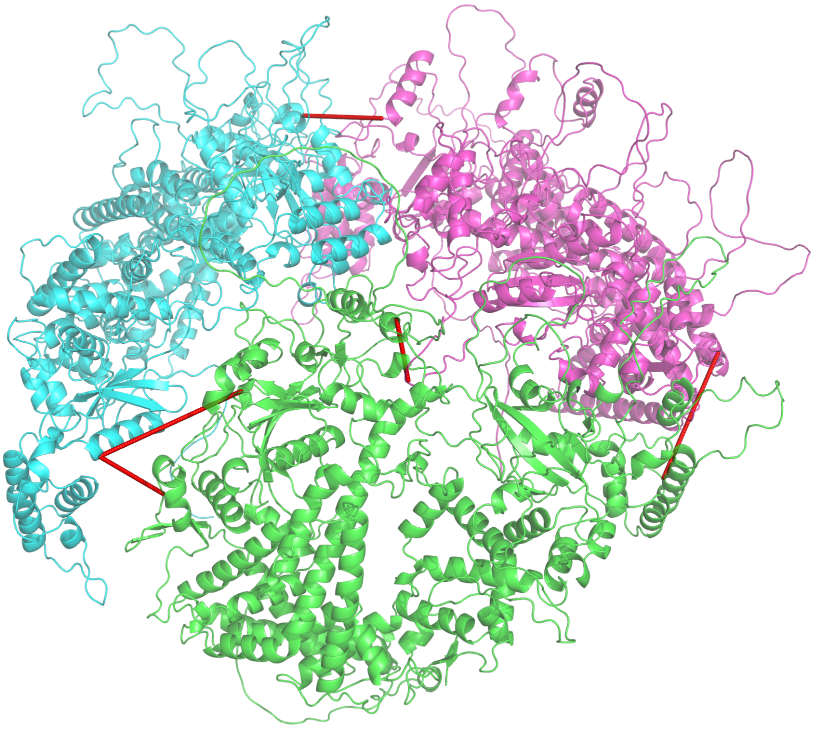
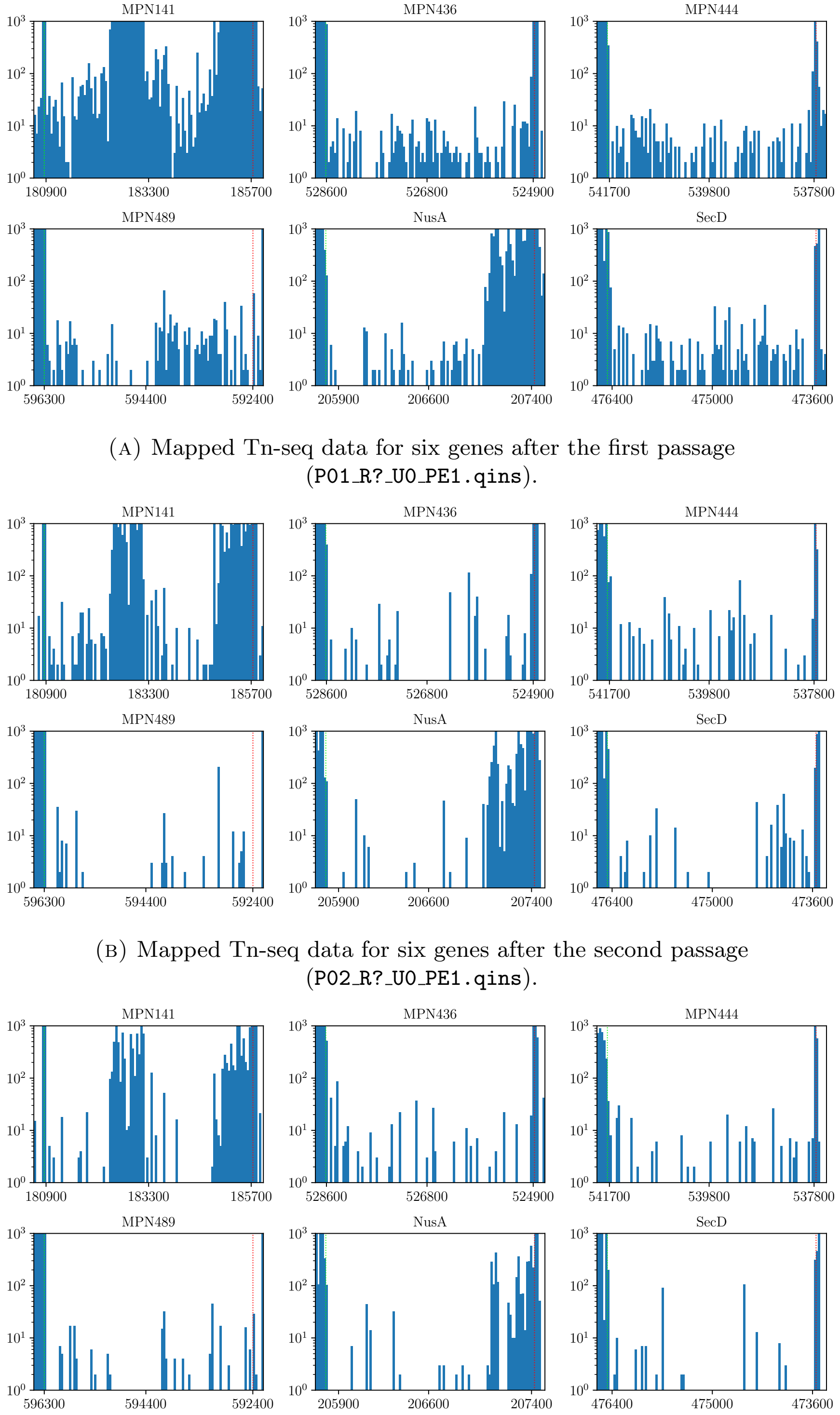


FIGURE 2.15: The unique assembly of the three cage proteins—MPN436, MPN444, and MPN489—as defined by crosslinking data. MPN436 (PowerFit pose 1) is shown in cyan, MPN444 (PowerFit pose 2) is shown in green, MPN489 (PowerFit pose 1) is shown in magenta, and the crosslinks are shown in red.

2.8 Essentiality analysis

According to *Myco*Wiki [136], MPN436, MPN444, and MPN489 are proteins essential for the survival of the *Mycoplasma pneumoniae* cell. In order to verify this, however, as well as to obtain better resolution in essentiality, I analysed data from transposon sequencing (Tn-seq). Tn-seq involves the random insertion of genetic material into the genome by transposases, thereby creating disruptions at all possible genome positions [137]. After several passages of cell growth, the surviving cells are sequenced with high coverage, and the insertion sites are identified, usually correlating well with non-essentiality [138]. The Tn-seq data had been preprocessed using the FASTQINS pipeline [139] to map the sequenced reads to the *M. pneumoniae* genome. For the cage proteins—MPN436, MPN444, and MPN489—as well as for control proteins SecD (MPN396; essential), NusA (MPN154; essential except C-terminus), and MPN141 (non-essential), I plotted histograms showing where Tn-seq reads map to in the genome, shown in Figure 2.16. The set of six histograms is shown three times: data from the first, second, and third passages. Replicates were pooled, and only the paired-end approach to mapping was used (without filtering for unique reads).



(A) Mapped Tn-seq data for six genes after the first passage (`P01_R?_U0_PE1.qins`).

(B) Mapped Tn-seq data for six genes after the second passage (`P02_R?_U0_PE1.qins`).

(C) Mapped Tn-seq data for six genes after the third passage (`P03_R?_U0_PE1.qins`).

FIGURE 2.16: MPN436, MPN444, and MPN489 are essential proteins based on Tn-seq data mapped to the *M. pneumoniae* genome. The genes (plus 5% on each end) for these proteins, as well as for MPN141, NusA, and SecD, are shown as histograms (each with 110 bins) three times: once for each of the first, second, and third passage. The x-axes of the histograms refer to the genomic position. The vertical dotted lines represent the beginning (green) and end (red) of the gene. Replicates were pooled, and only the paired-end approach to mapping was used (without filtering for unique reads).

Figure 2.16 convincingly shows that random transposon insertions in the genes for MPN436, MPN444, and MPN489 turn up rarely in viable cells, and therefore all three are essential for *M. pneumoniae*. In the first passage (Subfigure 2.16A), the effect is less extreme than in the second and third passages (Subfigures 2.16B/2.16C), likely due to residual protein from earlier generations. The controls also verify the accuracy of the approach: SecD has the same pattern as the cage proteins, NusA also does with the exception of a non-essential C-terminal domain, and MPN141 shows a large part of the protein is non-essential. It's also convincing to see that the genomic regions immediately before the N-terminus (shown as a green dotted line) and immediately after the C-terminus (shown as a red dotted line) of fully essential proteins have a sudden increase in non-essentiality. Using gold-standard data on gene essentiality from projects related to minimal cells [140, 141], I trained a model in ANUBIS [139] to recognize differences in linear density using penalized kernel change-point detection, and the model corroboratively predicts that the cage proteins are almost entirely essential.

2.9 Homology search

The cage proteins—MPN436, MPN444, and MPN489—have very few identifiable relatives. Using BLAST [142] and multiple rounds (with pruning) of PSI-BLAST [143], I found homologues almost exclusively within the *Mollicutes* class of bacteria—nine species of *Mycoplasma*, two species of *Ureaplasma*, two species of *Mycoplasmopsis*, one species of *Hepatoplasma*, and one species of *Spiroplasma*. Outside *Mollicutes*, there were two weak hits in *Firmicutes*. There were also ten hits in *Eperythrozoon* species, which are now considered haemoplasma-type species of the *Mycoplasma* genus [144, 145]. It seems that the cage proteins have a very specialized function, as they're found mainly in the *Mollicutes* class and especially the *Mycoplasma* genus, although even the *Mycoplasma* genus has many species that have lost the cage proteins (e.g. *M. imitans*).

With no success searching the sequence space, I turned my attention to two structure-based approaches. Firstly, I tried Geometricus [146, 147], an algorithm that breaks a protein structure into structural fragments in two ways (k-mer-based or radius-based) and produces a long vector of measures. Creating an embedding for these vectors allows for easier comparison of structures that share structural fragments, and it has been shown that by training a topic model (originally a method from natural language processing to cluster co-occurring 'words' into abstract 'topics') on these vectors, AlphaFold proteins cluster into distinct families with this method [148, 149]. I first tried this for all *Mollicutes* proteins, and then for all bacterial proteins, and this method simply did not work for the cage proteins, as there were very few other proteins with a sufficient feature overlap, and mostly clearly unrelated proteins. There were significant differences in the set of topics assigned to a protein even just between different AlphaFold models of it that differ only by an RMSD of 1–2 Å, even when the structural features were only extracted for high-confidence (pLDDT $\geq$ 70) parts of the models, implying the method is not robust enough to help find structural relatives in this putative low-abundance family. Also interesting to note is that larger proteins did not correlate with a larger set of assigned topics, even when more complex topic models were trained, which further suggests that this method is primarily suited to protein families with more representation.

I also tried searching structural-homology space using a tool called Foldseek [150], which extracts angles and distances between nearby elements along the protein structure and converts these features into a high-density sequence using a variational autoencoder. The sequence derived from the query structure is then searched against precomputed databases of such sequences, and high-scoring hits are aligned structurally. Although Foldseek ultimately maps to sequence space and, using an algorithm for sequence alignment (MMseqs2 [151]), also provides hits with smaller overlapping alignment length, to maximize sensitivity, I tried breaking the AlphaFold models of the cage proteins into domains, as well as removing low-complexity and/or low-confidence regions from the model, and running Foldseek queries on each substructure. For each query, I saved the results within a reasonably liberal confidence threshold. The results for each query, as well as for the multiple databases used for the searches, have lots of overlap, so I counted the proteins that appeared in the combined results most often. The three homologues (MG307, MG309, and MG338) from *Mycoplasma genitalium* are by far the most common hits, followed by MPN440 and MPN439 in *Mycoplasma pneumoniae*, which appeared in Table 2.4. Proteins in the "MG307/MG309/MG338 family" outside of *M. pneumoniae* also appear in the Foldseek results: MPN437, MAMA39_01700 from *Mycoplasma amphoriforme* A39, F537_02475 and F537_02480 from *Mycoplasma pneumoniae* 85084 (almost identical to MPN436 and MPN437), three lipoproteins from *Mycoplasma gallisepticum*, and H3143_02785 and H3143_02790 from *Mycoplasma tullyi.*

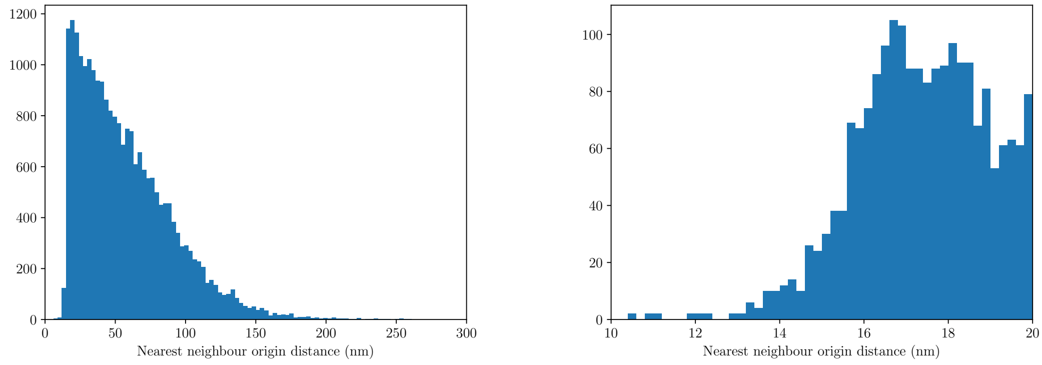
Outside of the UniProt family, there are many similar-sounding uncharacterized lipoproteins that appear as hits from likely organisms: MGA_0332 from *Mycoplasma gallisepticum*, MAMA39_01690 from *Mycoplasma amphoriforme* A39, as well as further hits in *Ureaplasma parvum*, *U. urealyticum*, *U. diversum*, *Mycoplasma genitalium*, *M. gallisepticum*, *M. penetrans*, *M. marinum*, *M. conjunctivae*, *M. haemobos*, *M. haemofelis*, *M. wenyonii*, *M. suis*, *Spiroplasma platyhelix*, *Spiroplasma alleghenense*, and *Mycoplasmopsis columbinasalis.* In the rest of the results, protein classes that tend to come up a lot include: outer-membrane lipoprotein LolB, periplasmic chaperone PpiD, membrane protein P80, foldase protein PrsA, and peptidylprolyl isomerase. One secreted protein from *Mycoplasma haemolamae* also appears.

2.10 Particle orientation and distribution

By eye, the cage complex often seems to appear in clusters, bunched into groups rather than spread evenly throughout the cell surface. In order to verify this, I calculated the origin-to-origin distance between each identified particle and its nearest neighbour in the same tomogram. While there are likely many repeated (inverted) pairs of cage particles in this analysis, this is not possible to avoid, since the nearest-neighbour function isn't involutory (i.e. if the nearest neighbour of p is q, p isn't necessarily the nearest neighbour of q).

Figure 2.17 shows a histogram of these nearest-neighbour distances, and at first glance, as shown in Subfigure 2.17A, it looks like there exists a minimum neighbouring distance between cage particles, after which the proportions rapidly diminish due to the unlikelihood of such a distance between nearest neighbours. In Subfigure 2.17B, however, after zooming in on the lowest distances in the histogram, there is clearly a small peak around 17 nm. Given that the cage particle has a similar diameter, this likely represents some interaction between closely packed cage particles.

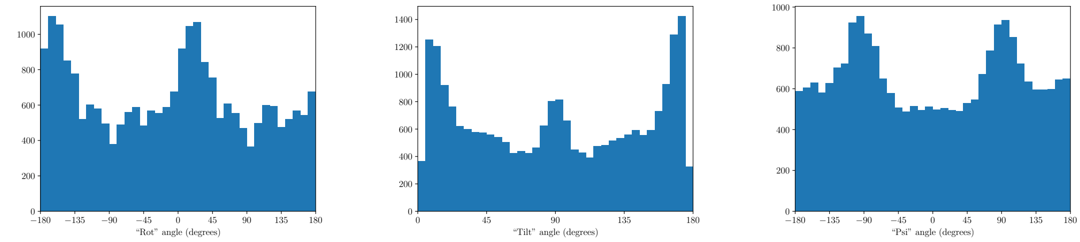
In order to investigate this potential interaction between neighbouring cage particles, it's first important to understand the orientation and distribution biases of the cage particles. In Figure 2.18, a simple histogram has been plotted for each of the three Euler angles across the refined cage particles. The distribution of “rot” angles in Subfigure 2.18A shows a reasonably uniform distribution with the exception of two peaks at approximately $-160°$ and $20°$. For the “tilt” angles whose distribution is shown in Subfigure 2.18B, we see three peaks: $10°$, $90°$, and $170°$. Since the pseudosymmetry axis for the cage particle is aligned to the z-axis in the refined density map, perpendicular



(A) Entire histogram.

(B) Zoomed in on a subrange.

FIGURE 2.17: A histogram of the distances between the origins of each cage particle and its nearest neighbour. Prior to the main distribution, there is a small peak that may represent interacting cage particles.



(A) Histogram for the "rot" angle (ϕ).

(B) Histogram for the "tilt" angle (θ).

(C) Histogram for the "psi" angle (ψ).

FIGURE 2.18: Histograms showing the distribution of each of the three Euler angles across the dataset of cage particles.

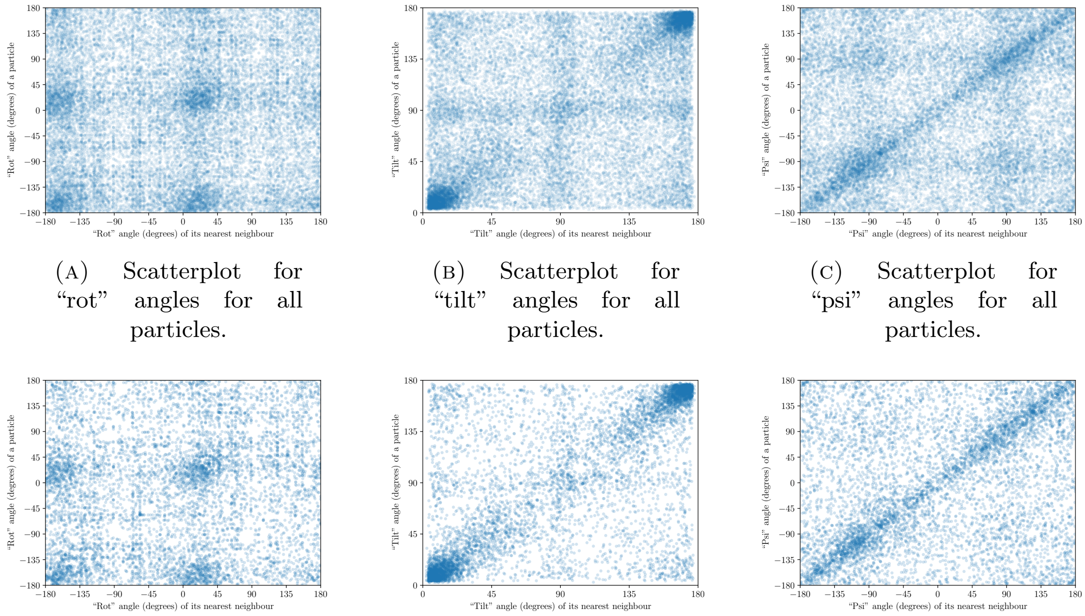
to the cell membrane, the "tilt" angle of a particle actually communicates the local cell geometry. The *M. pneumoniae* cells were slightly deformed in the blotting process prior to imaging, causing the cell to spread out more in the xy-plane (i.e. the plane of the grid). Cage particles found in the 'top' and 'bottom' of the cell (i.e. in the low and high z-slices of the tomogram, respectively) were therefore higher in number, as well as easier to locate due to the distinctive circular shape, which explains the 10° and 170° peaks in "tilt" angle. According to the 3D Image Conventions [64], the "tilt" angle cannot go below 0° or above 180°, which probably impacts the distribution. Instead of tilting, for example, by an angle of −10°, one would first rotate 180° about the z-axis, implement a tilt of 10°, and then fix the z rotation with the "psi" angle. The "tilt" peaks would probably sit right at 0° and 180° were it not for the discontinuous nature of the angle measure. The 90° "tilt" peak might also be explained by distinctiveness: the cage particle is harder to discern visually when seen from the side, but the distinctive dome shape can help when the image is clear enough. Finally, for the "psi" angles, whose distribution is shown in Subfigure 2.18C, there is a steady baseline with peaks at −100° and 90°. Just as for the two "rot" peaks, the two "psi" peaks seem to be separated by approximately 180°. Figure 2.18, as well as the figures and calculations in the rest of this section, were generated with the NumPy (v1.23.4) [152], SciPy (v1.9.3) [119], pandas (v1.5.1) [153], Matplotlib (v3.6.2) [154], `seaborn` (v0.12.1) [155], `mrcfile` (v1.4.x) from CCP-EM [120], and NetworkX (v2.8.8) [156] libraries in Python.

Assuming the cellular membrane curves smoothly around the cell, there should not be a large difference between planes tangent to the membrane at points near to one another on the cell surface. The orientations of closely neighbouring cage particles, therefore, should have similar tilt angles and tilt axes. In order to test this and investigate whether neighbouring cage particles indeed have correlated orientations, scatterplots were created, as shown in Figure 2.19, comparing each Euler angle of a particle with that Euler angle of its nearest neighbour. Subfigures 2.19A–2.19C show the data for all particles and their nearest neighbours, while Subfigures 2.19D–2.19F show the data for particles whose nearest neighbours are within 35 nm, in order to enrich for particles near enough to have some direct interaction.

Subfigures 2.19a/2.19d show that there is little "rot" angle correlation both in general and for particles within 35 nm. The four spots of higher density in each scatterplot arise naturally from sampling the bimodal "rot" distribution (Subfigure 2.18a) in general. Subfigures 2.19b/2.19e show that "tilt" angle is weakly correlated in general—except for large hotspots where the particle and its nearest neighbour have angles both near 0° or both near 180°—while the enrichment along the $y = x$ diagonal is much more pronounced when the distance between neighbours is thresholded. The (0°, 0°) and (180°, 180°) hotspots in both "tilt" scatterplots are expected based on the peaks at 0° and 180° in the "tilt" distribution (Subfigure 2.18b). Interestingly, however, despite the peaks in the "tilt" distribution, no hotspots appear at (0°, 180°) or (180°, 0°), which makes sense in the cellular context, since it would be very rare for the nearest neighbour of a particle in the 'top' membrane of the cell to be located in the 'bottom' membrane of the cell, and vice versa. Finally, for the "psi" angle correlation, shown in Subfigures 2.19c/2.19f, we see that there is an uncorrelated background for the general case—with some enrichment along the $y = x$ diagonal, and weak additional enrichment at (−100°, −100°) and (90°, 90°)—while this enrichment becomes much more pronounced for nearest neighbours within 35 nm. This "psi" correlation is surprising and warrants further investigation. True biological interaction between neighbouring cage particles could cause "psi" correlation, to be sure, but enrichment along the diagonal means that "psi" is unbounded as long as the neighbour has the same "psi", which seems contrary to the geometry of a potential lateral interaction.

One thing that's important to note is that, while the one-dimensional angle distributions (histograms) in Figure 2.18 are projections of the two-dimensional angle correlations (scatterplots) in Figure 2.19, and therefore the peaks also appear in the same locations, it's not possible to discern between a peak as a histogram artefact causing these densities in the scatterplots and something truly biological observed as density in a scatterplot causing a peak in a histogram.

Since a particle's nearest neighbour is not necessarily the only particle nearby, it was worth checking that the correlation between the angles of particles is generally enriched as a function of the distance between particles. Therefore, temporarily discarding the concept of a nearest neighbour, similarity metrics were established for each Euler angle, and these were plotted as a function of distance.



(A) Scatterplot for "rot" angles for all particles.

(B) Scatterplot for "tilt" angles for all particles.

(C) Scatterplot for "psi" angles for all particles.

(D) Scatterplot for "rot" angles for particles whose nearest neighbours are within 35 nm.

(E) Scatterplot for "tilt" angles for particles whose nearest neighbours are within 35 nm.

(F) Scatterplot for "psi" angles for particles whose nearest neighbours are within 35 nm.

Figure 2.19: For each Euler angle, a scatterplot showing the relationship between the angle of each cage particle and the angle of its nearest neighbour, plotted both for all particles ($\alpha = 0.1$) and for those with nearest neighbours within 35 nm ($\alpha = 0.2$).

The similarity metric for differences in "rot" angle was defined to be

$$\Delta_\phi(\phi_1 \to \phi_2) = \begin{cases} \phi_2 - \phi_1 - 360^\circ\,, & \text{if } \phi_2 - \phi_1 > 180^\circ\,; \\ \phi_2 - \phi_1 + 360^\circ\,, & \text{if } \phi_2 - \phi_1 < -180^\circ\,; \\ \phi_2 - \phi_1\,, & \text{otherwise}\,. \end{cases} \tag{2.1}$$

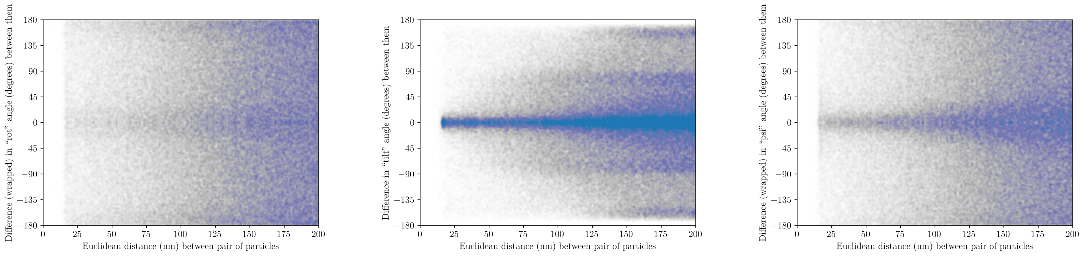
This preserves the direction of rotation to move from ϕ_1 to ϕ_2 and keeps the result in the allowed range $-180^\circ \leq \phi \leq 180^\circ$. Since there is no circular closure for the "tilt" angle, no such trick is necessary, and its similarity metric was simply defined as

$$\Delta_\theta(\theta_1 \to \theta_2) = \theta_2 - \theta_1\,. \tag{2.2}$$

For the "psi" angle, the same was done as in Equation 2.1, and the "psi" similarity metric was defined to be

$$\Delta_\psi(\psi_1 \to \psi_2) = \begin{cases} \psi_2 - \psi_1 - 360^\circ\,, & \text{if } \psi_2 - \psi_1 > 180^\circ\,; \\ \psi_2 - \psi_1 + 360^\circ\,, & \text{if } \psi_2 - \psi_1 < -180^\circ\,; \\ \psi_2 - \psi_1\,, & \text{otherwise}\,. \end{cases} \tag{2.3}$$

The 'distance' between Euler angles can now be resolved in a logical way and plotted



(A) Difference between "rot" angle, calculated according to Equation 2.1, plotted against Euclidean distance between cage particles.

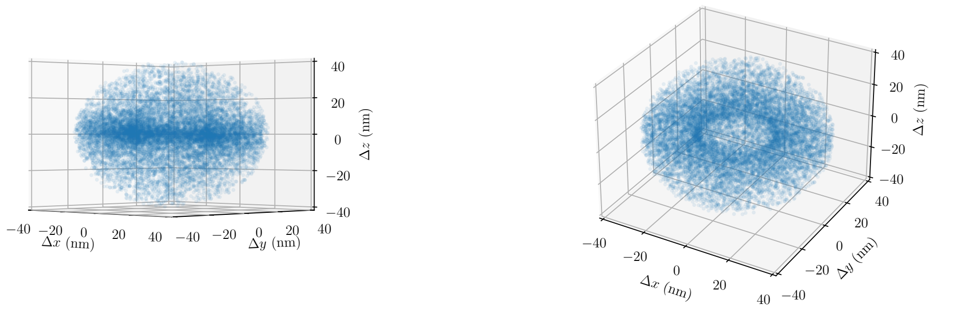
(B) Difference between "tilt" angle, calculated according to Equation 2.2, plotted against Euclidean distance between cage particles.

(C) Difference between "psi" angle, calculated according to Equation 2.3, plotted against Euclidean distance between cage particles.

FIGURE 2.20: For all pairs of cage particles that occur in the same tomogram, the difference between their Euler angles is shown plotted ($\alpha = 0.002$) against the Euclidean distance between them. For all three histograms, the x-axis is clipped at 200 nm, above which the plot would look completely filled.

as a function of spatial distance between all pairs of particles that occur in the same tomogram. For large spatial distances, this is meaningless, but the goal here is to show that some change happens as the smallest possible distance is approached. Three such scatterplots are shown in Figure 2.20, one for each Euler angle.

In all three scatterplots in Figure 2.20, the lowest observed distance is approximately 15 nm, as shown to be the case previously in Subfigure 2.17B. In Subfigure 2.20A, there is a uniform distribution of differences in "rot" angle for distances below 100 nm. For the differences in "tilt" angle, shown in Subfigure 2.20B, there is distinct enrichment around $y = 0$, with an increasing spread of noise as the distance increases, but a very clean signal until around 60 nm. Finally, in Subfigure 2.20C, there seems to be only a weak enrichment of difference in "psi" angle around $y = 0$ for all distances. The main discovery here is that the "tilt" angle is the only angle that stays similar for particles close to one another. This is likely due to some role overlap between "rot" and "psi" in certain scenarios. For side views, there is no problem: neighbouring particles have a similar "tilt" and therefore also need a similar "rot" in order not to be misaligned with the membrane, leaving the "psi" angle to correct the final in-plane particle rotation. For top (or bottom) views, however, there isn't much "tilt" applied (or nearly 180° of "tilt" applied), and so the "rot" actually doesn't matter as much. As an example, the sets of rotation angles (10°, 10°, 10°) and (0°, 10°, 20°) don't end up differing from one another too much. A more extreme example could also occur for top and bottom views: even with small membrane deviations from the tomogram's xy-plane, the tilt of the membrane (and therefore the "tilt" angle of the particles in the membrane around that location) could fluctuate around 0°, reaching −2° and 2°. Although Equation 2.2 does calculate



(A) View of the 3D plot from an elevation of 0° and an azimuth of −45°.

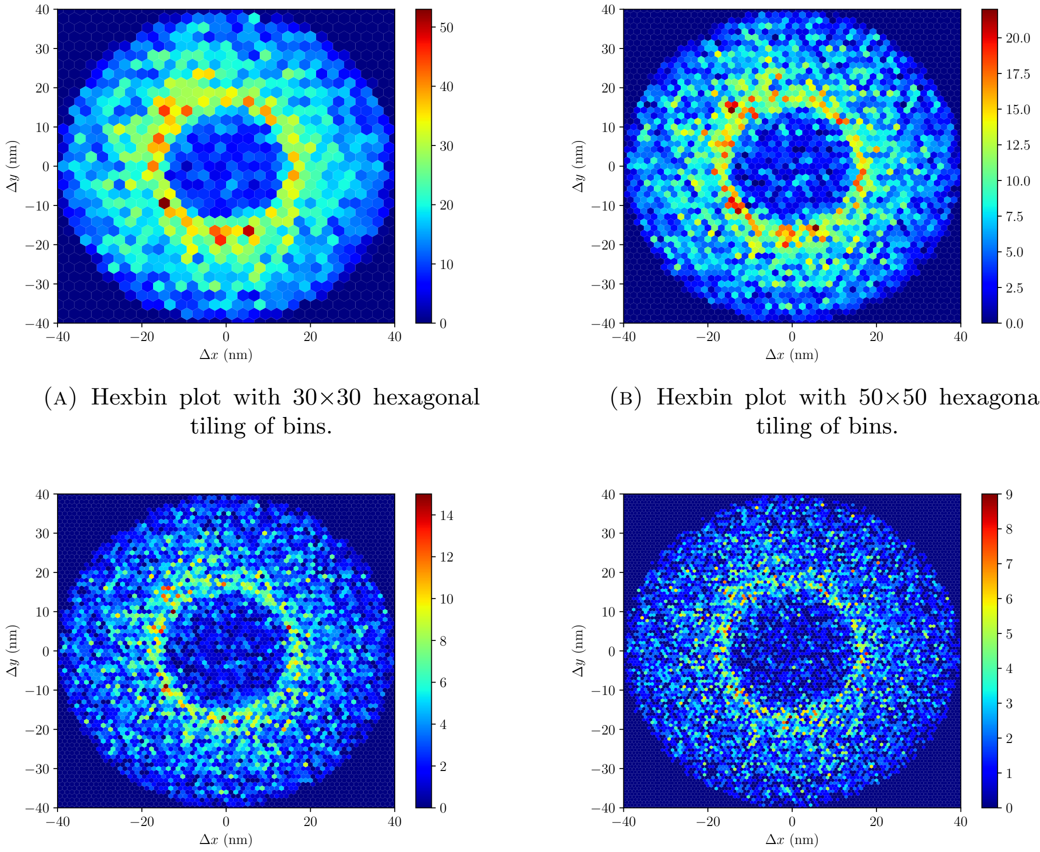
(B) View of the 3D plot from an elevation of 30° and an azimuth of −60°.

Figure 2.21: A plot of points ($\alpha = 0.1$) showing the relative position of the second particle in all pairs of cage particles, normalized such that the first particle is in a consistent orientation and located at the origin. Pairs of particles with a Euclidean distance between them exceeding 40 nm were excluded.

this difference correctly (4°), the rest of the Euler angles are very different. A negative “tilt” angle cannot be applied, and so it needs to be accessed via an approximately 180° “rot” angle, a positive “tilt” angle, and finally a corrective “psi” angle. In other words, the sets of rotation angles (10°, 5°, 5°) and (−170°, 5°, −175°) are also surprisingly similar.

Ultimately, the in-plane rotation of the cage particle is not easily determined from just a statistical analysis of Euler angles, and therefore an analysis of relative rotations between neighbouring particles is not practical for elucidating interactions between cage particles. We can instead use normalized relative positions between pairs of nearby particles by drawing the vector from one particle to another and rotating this vector using the inverted rotation matrix for the first particle. This aligns the first particle in all cases with the reference density map used in the refinement and creates a large number of vectors pointing outwards to show the relative locations of all other particles in the set of pairs. In Figure 2.21, two views of a 3D plot of the endpoints of these vectors are shown, as long as the length of the vector is at most 40 nm. As can be seen in Subfigure 2.21A, with a view parallel to the xy-plane, the bulk of the density lies on the xy-plane, which makes sense in light of the relatively continuous membrane plane that restricts the orientation of nearby particles. Better seen from the view in Subfigure 2.21B, there is also a ring devoid of datapoints, which is caused by the natural minimum distance between cage complexes. The spherical outer shape of the pointcloud is due to the distance cutoff of 40 nm.

The 3D pointcloud was a good way to ensure everything was working correctly, but it is a bit difficult to interpret due to the z-dimension and variations in point overlap. I



(A) Hexbin plot with 30×30 hexagonal tiling of bins.

(B) Hexbin plot with 50×50 hexagonal tiling of bins.

(C) Hexbin plot with 70×70 hexagonal tiling of bins.

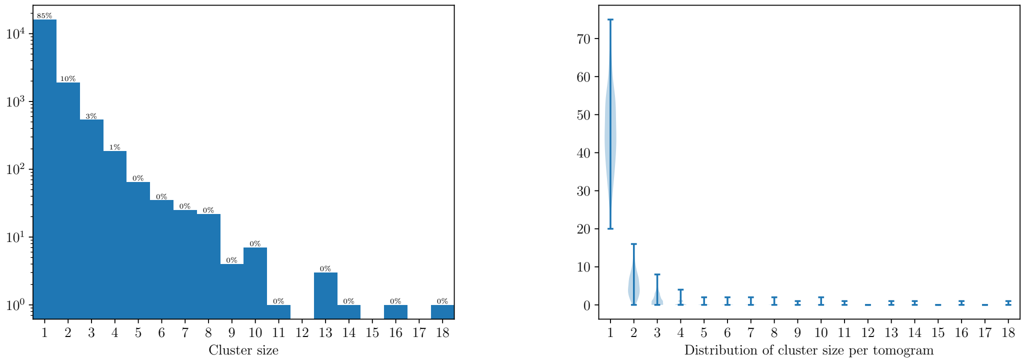
(D) Hexbin plot with 100×100 hexagonal tiling of bins.

FIGURE 2.22: Four hexbin plots, each with different binning, showing the relative position of the second particle in all pairs of cage particles, normalized such that the first particle is in a consistent orientation and located at the origin, and projected down to the xy-plane. Pairs of particles with a (3D) Euclidean distance between them exceeding 40 nm were excluded.

therefore plotted a 2D version, simply ignoring the z-dimension and projecting everything on the xy-plane, with point density represented by colour in a 2D histogram called a hexbin, which uses a tiling of hexagonal bins. In Figure 2.22, four versions of this plot, with differing binning granularity, can be seen.

Looking at the plots in Figure 2.22, we can see that cage particles have no tendency to prefer one relative position over another when in the proximity of another cage particle. Although some cells in the hexbin plots contain higher numbers than their surroundings, these are one-off cases, likely just statistical noise, rather than the dense clusters of enrichment that could be expected in the case of true biological interaction.

In spite of not finding evidence for interactions between cage particles in the form of preferred orientations between neighbouring cage particles, the clustering of cage particles remains visible to the naked eye, and the clusters should be quantified. To do



(a) The distribution of cluster sizes: for each given cluster size, how many clusters exist across the dataset?

(b) Per-tomogram cluster distributions: for each given cluster size, what's the distribution of how many there are per tomogram?

Figure 2.23: Statistics on the clustering of cage particles in the dataset. Particles within 35 nm are considered neighbouring for the purposes of cluster formation.

this, I simply gave each particle a unique name and created a vertex with that name using the NetworkX (v2.8.8) [156] library in Python, added an edge between all pairs of vertices in the same tomogram with a Euclidean distance between them below 35 nm, and calculated the connected components of the graph. Of course, the weakness of this approach is that cage particles that have gone unpicked and remain missing in the particle list might break up clusters that would otherwise exist. A maximum distance between particles of 35 nm for clustering purposes attempts to account for this, since tightly clustered cage particles tend to have a distance between them of 17 nm, and with three in a straight line (the most spaced-out arrangement possible), the cluster would still be formed even if the central particle were missing. For non-linear clustering arrangements, a higher distance between particle and missing particle would be tolerated. In Figure 2.23, some clustering statistics are presented.

Subfigure 2.23a presents how many clusters there are, for each cluster size, in the dataset of cage particles. The vast majority of cage particles (85%) are not in clusters, and 10% of cage particles are in clusters of size 2. This leaves only 5% of particles left for participation in clusters of size ≥ 3—slightly more than 1000 in absolute terms. The largest cluster size is 18, although there is only one instance of this, which is also true for cluster sizes 16, 14, and 11. Subfigure 2.23b illustrates the distribution of number of clusters per tomogram, broken down by cluster size, in the dataset of cage particles. An average tomogram has 45 unclustered particles, 5 clusters of size 2, and just a handful of larger clusters.

Chapter 3

Discussion

3.1 On optimizing deep-learning-based particle picking

Nobody wants to pick particles manually: it's time-consuming, eye-straining, and error-prone. In order to use a deep-learning-based particle-picking tool, however, we need a training mask. Even if opting to create this mask using solid spheres instead of particle-shaped stencils, particle coordinates are needed. The decision is therefore between manually picking all particles to train the CNN model and using template matching as an intermediate step. The use of template matching is also important in the applicability of the workflow, in the sense that there are now ways to enter the workflow starting from a density map (the template) or a particle list (coordinates). There is an endless number of approaches to get particle picking started when template matching is used as a jumping-off point and any density map can be used as a template. For the approach demonstrated with the cage complex, manual picking was shown as one way to generate a template for template matching; another way was by trimming it from a density map where the cage complex was associated with the ribosome. Manual picking for template generation is an obvious choice when the particle represents a visual curiosity in the tomogram dataset, something relatively easy to identify by eye and something that clearly exists. Another approach could be the use of a homologous solved structure (of a protein or a complex) from the Protein Data Bank (PDB) [157] to find something in the dataset—the homologous structure can be converted to a 3D electron density, filtered and binned, and used directly as a template. For example, in order to pick rpoA (the alpha subunit of RNA polymerase in *M. pneumoniae*), running a sequence search against the PDB shows that `6WVJ` and `7L7B`, full RNA polymerase structures for *Bacillus subtilis* and *Clostridia bacterium*, respectively, are top matches, which can be saved and converted for use in template matching. Another way of adapting this style of workflow, depending on the biological question at hand in the experiment, is using larger protein complexes as 'bait' and, after template matching and deep-learning-based particle picking, probing for interactions by classifying out baits that associate with 'prey'. One limitation of this approach, however, is that minimum sizes for both bait and prey would exist, below which it's unlikely these interactions would be well resolved. Of course, with the release of AlphaFold2 [87, 88], there is also the new possibility of generating a template directly from the sequence of interest. One could use precomputed AlphaFold DB [125] structures, or even run AlphaFold-Multimer [158] to predict the structure of a hypothesized protein complex and use that as a template.

As mentioned, a training mask is needed to train deep-learning-based particle-picking models like DeePiCt [105] or DeepFinder [104], and these masks can be created by repeatedly pasting binary stencils at the coordinates and orientations dictated by the particle list. The stencils are typically either a 3D geometric shape—the naïve approach, especially useful without any prior information—or particle-shaped. With shape-based stencils, there isn't much that can be adjusted other than the size and shape. Not knowing much about the particle of interest other than that it's cylindrical, one might opt for a cylinder-shaped stencil. With particle-shaped stencils, on the other hand, there is more to consider. Should the whole particle be used, or rather just a distinct domain/subunit? For membrane proteins, should the membrane be included? What density threshold should be used to binarize the stencil? After thresholding, does it make sense to grow the edges to include a bit more context? Should the contour of the threshold be followed even when it creates holes in certain places, or should the stencil be filled in to be completely solid? From the experiments and evaluations in Section 2.5, these questions can be answered. Using the stencils shown in Figure 2.6, with results shown in Table 2.1, it's clear that differences between the three stencils are minimal. In general, stencil volume helps with detection, although this doesn't affect the predicted probability map as much as the accuracy of clustering and assigning centroids to connected components. Although not quantified here due to a negligible difference in actual detection at the level of the probability map, the biggest usable gain in accuracy as a result of stencil use was switching to the solid stencils in Figure 2.7, which was a direct result of improving the split clusters in predicted probability maps, as shown in Figure 2.8. Including or excluding the membrane from the stencil didn't have much of an effect, although it was ultimately decided to proceed with extracellular-only stencils, so this should be decided on a case-by-case basis. The density threshold at which to binarize the stencil (i.e. the cutoff below which voxel values in the density take a value of 0 in the stencil and above which they take a value of 1 in the stencil) should be decided empirically such that there's a smooth surface (in what seems like a reasonable shape) without any holes on the particle's exterior and also as few speckles as possible floating around the particle (although those can be erased manually after the fact). When in doubt, however, erring on the side of larger stencils is probably wise; incorrectly merged particle clusters are easier to find due to their size and can be separated. Growing the stencil for the purpose of context is pointless, since the particle-picking CNN should be trained with a box size large enough to include context even when centred at the centre of the particle. One final thing to note with respect to DeePiCt is that the clustering algorithm applied in the postprocessing step (i.e. the conversion of thresholded probability maps to particle lists) does not actually do things differently depending on the minimum and maximum particle cluster sizes given as parameters. In other words, on seeing a cluster twice as large as the maximum allowed cluster size, instead of fitting two particles to that cluster, it simply discards the whole thing before outputting the particle list. It is therefore recommended not to set minimum and maximum cluster sizes and instead do this filtering manually alongside the thresholding of low-scoring clusters out of the results.

In Section 2.5, the differential effect of using raw tomograms, deconvolved tomograms, or denoised tomograms in DeePiCt was elucidated. It's clear from Table 2.2 that using deconvolved tomograms as the 3D image data for training and prediction offers a great improvement over denoised tomograms. Remembering that denoised tomograms are first deconvolved before denoising, this comparison doesn't show any deconvolution impact but rather a negative impact that denoising has. The effect is indisputable, however, with all six clear misses recovered when switching from denoised to deconvolved. This is not particularly surprising, since denoising is a bit of a shadowy operation, creating some blurriness in the tomogram and potentially also introducing artefacts. Although the data is not shown, a small sample of the effect of switching from deconvolved tomograms to raw tomograms can be seen in the montages of probability maps in Figure 2.8. Comparing Subfigures 2.8D/2.8E to Subfigures 2.8F/2.8G, respectively, shows that there is much the same result, although here it was decided to proceed with deconvolved tomograms. The deconvolution operation is effectively a filter, amplifying certain parts of the frequency spectrum and reducing the range over which the data is spread. Since CNNs of sufficient complexity can model any function, it's not unexpected that the particle-picking CNN can learn to pick in raw or deconvolved data equally well, especially considering that the box size is likely large enough to capture even the most delocalized signal.

If denoising doesn't help with such analysis, why should one even consider applying it? Although here it has been shown that denoising is counterproductive in the context of deep-learning-based particle picking, that doesn't mean it isn't very useful in other contexts. Indeed, with any kind of manual curation, or even manual inspection as a sanity check, it is often very appreciated to have a denoised version of the tomogram to reference alongside the coordinates. Not all particles, even distinctive ones such as the cage complex, are easily visible in deconvolved tomograms, and denoised tomograms surely make any visual task much easier to handle. With this dataset, it was demonstrated that Noise2Map [94] produced better output than Topaz-Denoise [95], and so the Noise2Map-denoised tomograms were used when denoised tomograms were required. It is hard to establish, however, whether to lend any credence to the generalizability of these results. With a high-quality dataset derived from motion-corrected movie frames, the best suggestion would be to train a new model if there exists the computational capacity and time to do so. The fact that Noise2Map with a pretrained model (trained on the same data but reconstructed at a different pixel size) outperformed Topaz-Denoise with a freshly trained model is evidence in favour of Noise2Map.

When it comes to training deep-learning-based particle-picking models, it has been demonstrated in Section 2.5 that iterative retraining helps a lot to improve the predictions. The fundamental principle behind this is that the predictions of a model almost certainly deviate in some way or another from the data used to train it, even when the scope (tomograms, in this case) is the same for training and prediction. The first round of training is generally on a small amount of data—the highest-confidence data from template matching—and the goal of even using software like DeePiCt is both to expand the variability and number of particles picked and to be confident in making good picks. It's therefore reasonable to expect that the first round of training will produce a model with some of the bias of the template-matching results, and its predictions will find mostly similar particles that had been missed, and occasionally some new varieties, too. This is why (at least) a second round of training is required; the curated results must now be allowed to update the model with improved information. In each round of retraining, the model predicts some new particles, and the hope is that they act as bridges, pushing the model scope in a direction that can eventually include a significant subpopulation that had yet to be picked. The downside to this approach is that the model counterproductively grows more and more confident about what it knows, and it thus becomes harder to distinguish new particles lacking this confidence (measured by score, related to cluster size) in the shadow of the 'old-boy' particles. Although not shown here, this can be observed by looking at score histograms of predictions after successive rounds of retraining, wherein the scores get progressively higher and the model no longer makes predictions that don't end up being confident ones. In a sense, this could be considered overfitting: training to the point where generalizability is lacking and only predictions that match the training perfectly, including the noise, are made. This is why it's important to benefit from rounds of retraining without going too far and reaching this state. In work with the cage particle, two to three rounds of retraining were found to suffice. This is also why DeepFinder can be suggested over DeePiCt for the majority of particle-picking tasks. Even ignoring the fact that DeepFinder trains a truly multi-class model, DeepFinder does a better job setting itself up for success by sampling particles in an intelligent way to foster robustness and also prevent overfitting.

3.2 On characterizing picked cage particles

In general, the strategy for optimizing particle refinement and classification in RELION [59] will depend drastically on the specifics of the particle of interest. The ideal case would be an abundant particle with an even distribution of orientations that doesn't display much structural flexibility; in all other cases, some testing needs to be done. In Section 2.6, there was a fundamental problem increasing the resolution of the average even when adding more particles. This could potentially have been due to flexibility between different domains/subunits, and this was tested by performing multibody refinement in RELION [159], using the masks shown in Figure 2.10 to define the subregions to be considered for focused refinements. Apart from potentially highlighting some differences between the three subunits (which are now indeed known to be three homologous but distinct subunits), multibody refinement did not help in this case. Eventually, after some classification runs in RELION, one class was distinctly defined yet unlike the cage complex, and it resulted in the discovery of the hexamer. Removing such interference from the dataset is crucial to obtaining good results from refinement, and sometimes it just takes a lot of fiddling with parameters and hierarchical jobs (refining one class of a classification, etc.) in RELION to find something of that nature.

An uneven orientation distribution was another factor that led to reduced yield in RELION. There would ideally be even coverage across all three Euler angles, as could be expected with a cytosolic protein that forms only transient interactions, which would allow for compensation of the anisotropy [160]. Due to the cage complex being restricted by the plane of the membrane, and the fact that the cell is deformed slightly outwards at the sides, there is a resulting bias for views of the cage complex from the top and bottom of the cell, with respect to the z-axis, as shown in Figure 2.11. At various times and with a few attempts at retraining, a DeePiCt model was trained on a subset of cage particles that had side-view orientations. Due to the alignment of the z-axis of the average with the axis of pseudosymmetry of the complex, the "tilt" angle alone could define the view in this sense, with tilts near 0° being top views, tilts near 90° being side views, and tilts near 180° being bottom views. Despite it being a good sign of DeePiCt's ability to generalize across anisotropic views, the model rarely found more side views and instead rather found the top and bottom views on which it had not even been trained. Another approach was also attempted, limiting the number of top and bottom views in the particle list, deleting the ones with the lowest contribution weight from RELION first, until they were balanced with the number of side views. This should have equalized the information content as a function of tilt angle and allowed for a more homogeneous refinement, but the results were only a slight improvement. A slight improvement despite removing a significant proportion of the particles, however, does mean that this strategy works; it's just that more side views would help much more than removing top and bottom views.

After finalizing the list of cage particles identified in the dataset of tomograms, the particles were averaged to obtain an initial reference for running a consensus refinement in RELION. The alignment parameters from the consensus refinement were then used to classify the particles into five classes. The best-refining class was used for a final round of refinement with a soft particle-shaped mask, giving the final alignment parameters and a density map refined to 11 Å as determined by a Fourier shell correlation (FSC) cutoff of 0.143. The resulting density map is shown in Figure 2.12, in which a number of features can be observed by eye. In the top view of the density (Subfigure 2.12A), the top of the extracellular cage component of the complex is visible, and a keen eye may see the three ridges running from the centre to the top-left, the top-right, and the bottom-middle, which demarcate where the three subunits of the trimer fit. In the bottom view (Subfigure 2.12B), the portion of the density corresponding to the intracellular components of the complex can be seen. In the side view (Subfigure 2.12C), the lipid bilayer of the cellular membrane can clearly be seen, as well as the transmembrane densities passing through it. In the cutaway side view (Subfigure 2.12D), the name of the cage complex seems very appropriate due to the dome-like extracellular structure whose hollow interior can now be seen, complete with some contained cargo.

Based on anecdotal statistics, it seems that cage particles in tomograms sometimes form clusters. When large enough, the clusters tend to seem hexagonal (i.e. in a sort of honeycomb-like arrangement), but this happens rarely. In addition, as shown in Figure 2.17, there is a peak in the histogram of distances between cage particles and their nearest neighbours that roughly corresponds to the spacing of such cluster-packing behaviour. Finally, as shown in Figure 2.13, when subtomograms of cage particles are extracted with a sufficiently large box size and subjected to classification in RELION, some classes include what appears to be a neighbouring cage particle, implying that the distance and angle of this neighbour relative to the aligned central cage particle is consistent enough in at least some subset of subtomograms. These three observations taken together are good signs that the cage particles do truly cluster somehow. In order to assess the validity of this hypothesis, as presented in Section 2.10, a detailed analysis of the positions and orientations of cage particles was performed, including the relationships between angle similarity of particles and distance between them (Figure 2.20), the trends in their relative positions (Figure 2.22), and the correlations between the orientation angles of particles and those of their nearest neighbours (Figure 2.19). Calculating the difference between respective Euler angles of all pairs of particles within the same tomogram and plotting them as a function of the spatial distance between them, as shown in Figure 2.20, it is clear that the only proper enrichment of near-zero difference in angle for particles close to one another is the "tilt" angle (Subfigure 2.20B). Since this isn't also the case for the "rot" angle (Subfigure 2.20A), likely due to the lessened impact of "rot" angle for "tilt" angles of top or bottom views (as it can be compensated by "psi" angle with little net effect), it can be concluded that an analysis of relative positions in order to establish the potential for an interaction between neighbouring cage particles cannot be conducted using the correlations of individual Euler angles. In other words, the resulting in-plane rotation (in the plane of the membrane) is not easily determined from just the individual Euler angles. Normalized relative positions are instead used between the aligned first particle and the relative second particle, for all pairs within 40 nm, temporarily ignoring the orientation of the second particle for the sake of simplicity. This is plotted in Figure 2.22 as a two-dimensional histogram (using only the xy-plane of the first particle), where colour represents the density of relative neighbours, and there appears to be no tendency for cage particles to prefer one relative position over another when in the proximity of another cage particle.

Despite this, the clustering tendencies of cage particles can also be quantified by calculating the connected components of a graph structure where nodes represent particles and edges represent a distance between two particles of at most 25 nm. Figure 2.23 plots this data, with Subfigure 2.23A showing the distribution of cluster sizes throughout all tomograms. Most cage particles (85%) are not in clusters, and 10% of cage particles are in clusters of size 2. This leaves only 5% of particles in clusters of size ≥ 3—slightly more than 1000 in absolute terms. Subfigure 2.23B illustrates the distribution of number of clusters per tomogram, broken down by cluster size, showing that an average tomogram has 45 unclustered particles but with a large variance, 5 clusters of size 2, and just a handful of larger clusters. This all seems to match what can be observed visually in tomograms, where clusters are clear and can be large when they do exist, but cannot be seen ubiquitously. With data-driven analysis through the alignment of cage particles in large subtomograms (Figure 2.13) and through this clustering analysis (Figure 2.23), it seems that cage particles do appear in clusters at least some of the time. What can be concluded if clusters appear without any evidence of interaction between them? Given the likelihood that the cage complex is some kind of chaperone involved with a secretory system (presented in Section 2.9; more on this in Section 3.3), in combination with the fact that the cage complex is seen interacting with ribosomes, it could be that cage complexes are translated in bulk at a location when there is a high need for them or that they are similarly recruited to a location. Another possibility is that they are somehow associated with polysomes, three-dimensional arrangements of ribosomes documented by Xue *et al.* [37], with each cage complex interacting with one ribosome in a polysome.

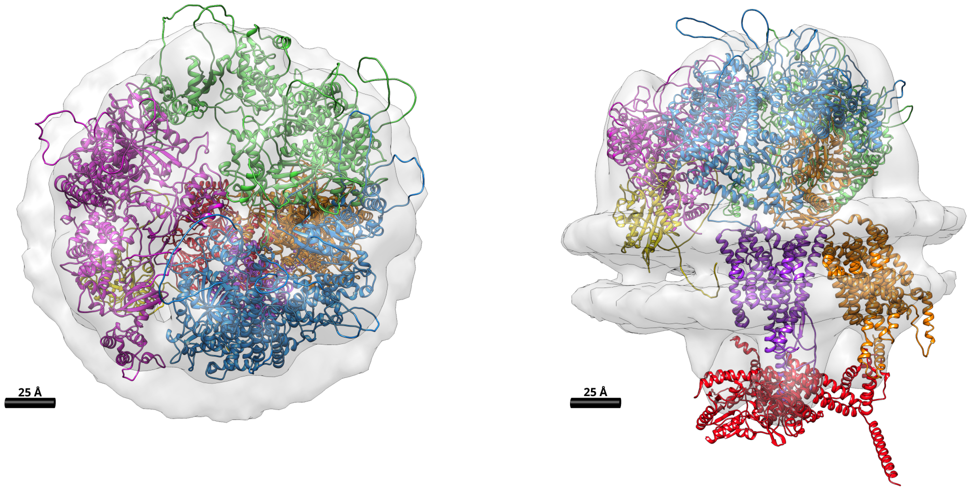
3.3 On characterizing the cage and its constituents

As presented in Section 2.7, the density map of the cage complex had been finalized and could now be used to help identify the proteins forming the complex. PowerFit [126] was used for the rigid-body docking of candidate protein structures predicted by AlphaFold2 [87, 88] into the density map. AlphaFold's timely appearance halfway through this project aided greatly in the characterization of the cage complex.

MPN444 was highly ranked in terms of cross-correlation score with the map, and the top three poses correspond to near-perfect 120° rotations about the z-axis, which implies the existence of three-fold rotational (pseudo)symmetry, although not necessarily with three instances of this same protein. MPN444 is an uncharacterized lipoprotein and a homologue of MG309 in *Mycoplasma genitalium.* Pfam [127] annotates it as containing a domain of unknown function called DUF3713 that occurs only in the genus *Mycoplasma*, while UniProt [128] classifies it as part of the "MG307/MG309/MG338 family". The C-terminal portion of MG309 has some immunostimulatory capacity [129] and all three can be used for strain typing of *M. genitalium* based on variable short tandem repeat (STR) sequences they contain [130]. There are nine proteins in this family in *M. pneumoniae*: three large ones (MPN436, MPN444, and MPN489)—each around 1300 amino acids, which uniquely map as homologues of the three *M. genitalium* proteins forming the name of the family—and six much smaller ones that appear to originate as fragments of the three large ones. MPN436, MPN444, and MPN489 are all lipoproteins whose first 26 or 27 residues form the signal peptide, which is cleaved off in the final protein product. In *M. pneumoniae*, the MPN436, MPN444, and MPN489 proteins form part of lipoprotein multigene family 3 [131] of unknown function since the genome was sequenced [1, 4]. MPN436 also appears with a top score in the PowerFit results, and MPN489 appears further down the list in thirty-second place, but the most striking thing to note is that the AlphaFold models of all three are very similar in both sequence and structure to one another, with both these measures shown pairwise in Table 2.5. The AlphaFold models of MPN436, MPN444, and MPN489 were assessed for accuracy using crosslinking mass spectrometry (CLMS) [132] data from the *M. pneumoniae* expressome study [50]: comparing the distance between intra-protein crosslinked residues in the models with the maximum reach of the crosslinker, on average 92.5% of crosslinks were validated, as shown in Table 2.6. Investigating inter-protein crosslinks in the same dataset, there are twenty such crosslinks involving MPN436, MPN444, or MPN489, of which five are directly between two of these proteins. These five crosslinks uniquely define the arrangement of the MPN436–MPN444–MPN489 assembly, shown in bold in Table 2.7. In Figure 2.15, this assembly is illustrated along with its crosslinks using the AlphaFold-modelled structures in their highest-scoring PowerFit pose with the correct relative position. Data from *Myco*Wiki [136] also support this, with nearly equal protein abundance levels of these three proteins [135].

Table 2.7 also shows that MPN444 has three crosslinks to "SecD" (MPN396), a protein involved in protein translocation in *Escherichia coli* along with SecF [161]. In fact, the MPN396 gene in *Mycoplasma pneumoniae* is known to code for a fusion of SecD and SecF [4]. Knowing that SecDF is a transmembrane protein in the vicinity of MPN444, the AlphaFold-predicted structure of SecDF was manually positioned roughly into the density below MPN444 and then optimized in Chimera [162], which fits well. SecYEG is known to associate with SecDF in *E. coli* as part of a transmembrane complex [163], and SecYEG is known to co-assemble with SecA in *E. coli* [164]. SecY (MPN184), SecE (MPN068), and SecG (MPN242) are independent proteins in *M. pneumoniae* but form a complex largely dominated by SecY. Using the AlphaFold-predicted structures for SecYEG and SecA (MPN210), fits were performed just as for SecDF. Additionally, MPN523 is shown in Table 2.7 to have crosslinks with MPN436, implying that it should be extracellularly membrane-anchored and close to MPN436. The same fitting approach was used here to fit MPN523 into the remaining empty density below MPN436. Although not presented in this thesis, an updated model is shown here in Figure 3.1, with the structure models for different proteins shown in different colours, all superimposed on a semi-transparent version of the density map from Figure 2.12.

Although *Myco*Wiki [136] labels MPN436, MPN444, and MPN489 as essential proteins, an analysis of higher-resolution essentiality data was performed to gain some insight on the essential components of these proteins. As presented in Section 2.8, a dataset from transposon sequencing (Tn-seq) was analysed. For the cage proteins—MPN436, MPN444, and MPN489—as well as for control proteins SecD (MPN396; essential), NusA (MPN154; essential except C-terminus), and MPN141 (non-essential), diagrams were plotted to show where transposons could insert without affecting cell viability, therefore suggesting non-essentiality, as shown in Figure 2.16. Insertions in the genes for MPN436, MPN444, and MPN489 turn up rarely in viable cells, and therefore all three all essential for *M. pneumoniae*. The effects become more extreme with progressive cell passages, likely due to residual protein from earlier generations. The controls also verify the accuracy of the approach: SecD has the same pattern as the cage proteins, NusA also does with the exception of a non-essential C-terminal domain, and MPN141 shows a large part of the protein is non-essential. A model in ANUBIS [139] was also trained



(A) A view of the model from the top, facing the dome-shaped trimer.

(B) A view of the model from the side, with the dome-shaped trimer above the membrane.

FIGURE 3.1: The most complete model of the cage complex at the time of writing. Shown superimposed on a semi-transparent density map of the cage complex are MPN436 (in pink), MPN444 (in blue), MPN489 (in green), SecDF (in orange), SecY (in violet), SecA (in red), and MPN523 (in yellow).

using gold-standard data on gene essentiality [140, 141], and the model predicts that the cage proteins are almost entirely essential.

The cage trimer proteins—MPN436, MPN444, and MPN489—have very few identifiable relatives. As presented in Section 2.9, using PSI-BLAST [143], homologues were found almost exclusively within the *Mollicutes* class of bacteria. Outside *Mollicutes*, there were two weak hits in *Firmicutes*. The cage complex likely has a highly specialized function, as its main components are found mainly in the *Mollicutes* class and especially the *Mycoplasma* genus, although even the *Mycoplasma* genus has many species that have lost the cage proteins (e.g. *M. imitans*). Focusing instead on structural similarity, the structural space was searched using a tool called Foldseek [150], with AlphaFold models of MPN436, MPN444, and MPN489 as the queries, both in their whole forms and in smaller domain- or subsequence-based pieces in order to maximize sensitivity. An overwhelming number of hits are produced, so the higher-scoring results were saved from each search and analysed for frequency of appearance. The three homologues from *Mycoplasma genitalium* (MG307, MG309, and MG338) are by far the most common hits, followed by MPN440 and MPN439 in *Mycoplasma pneumoniae*. Proteins in the "MG307/MG309/MG338 family" outside of *M. pneumoniae* also appear in the results: MPN437, MAMA39_01700 from *Mycoplasma amphoriforme* A39, F537_02475 and F537_02480 from *Mycoplasma pneumoniae* 85084 (almost identical to MPN436 and MPN437), three lipoproteins from *Mycoplasma gallisepticum*, and H3143_02785 and

H3143_02790 from *Mycoplasma tullyi*. Outside of the UniProt family, there are many potentially similar uncharacterized lipoproteins that appear as hits from likely organisms: MGA_0332 from *Mycoplasma gallisepticum*, MAMA39_01690 from *Mycoplasma amphoriforme* A39, as well as further hits in *Ureaplasma parvum*, *U. urealyticum*, *U. diversum*, *Mycoplasma genitalium*, *M. gallisepticum*, *M. penetrans*, *M. marinum*, *M. conjunctivae*, *M. haemobos*, *M. haemofelis*, *M. wenyonii*, *M. suis*, *Spiroplasma platyhelix*, *Spiroplasma alleghenense*, and *Mycoplasmopsis columbinasalis*. In the rest of the results, protein classes that tend to come up a lot include: outer-membrane lipoprotein LolB, periplasmic chaperone PpiD, membrane protein P80, foldase protein PrsA, and peptidylprolyl isomerase. One secreted protein from *Mycoplasma haemolamae* also appears. In light of the SecA–SecDF–SecYEG machinery, known to be associated with protein transport, forming a large part of the cage complex, and especially given that the ribosome seems to interact at least sometimes with the cage complex via this machinery, it is likely that the cage complex is also part of a novel type of translocon. This would also be a good explanation of the small chain-like density that can be seen entering the hollow dome of the cage complex in the cutaway side view of the density map (Subfigure 2.12D). Considering the protein classes that appeared most frequently in the Foldseek results, a narrower hypothesis is allowed: the cage complex is likely a chaperone or foldase [165] to aid with protein folding of nascent translocated polypeptides.

3.4 Conclusions

This thesis has accomplished its aim of describing and demonstrating a novel workflow for the systematic analysis of protein complexes from cryo-ET data. Starting from raw imaging data, a structured outline of steps has been presented to enable tomogram reconstruction, tomogram denoising, template matching, deep-learning-based particle picking, subtomogram averaging, particle refinement and classification, and constituent identification. At each step, the bottlenecks were eliminated through optimization, and orthogonal datasets were used as much as possible to whittle down the scope of imposing steps. Finally, as a way to be guided through the workflow, and also as a proof of principle, an *in situ* dataset of tomograms from *Mycoplasma pneumoniae* was used to characterize a novel membrane-associated protein complex, achieving an electron density map with a reasonable resolution, a multimeric structural model that fills the majority of the map, and a good idea of its function.

The intricacies of deep-learning-based particle picking were analysed, and optimizations were performed and discussed for tasks ranging from stencil selection for tomogram masks to iterative strategy for model training. The role of denoising was explored and two popular denoising tools were compared. As well as touching on ways to avoid bias in model training and particle refinement, a detailed analysis of particle orientation distribution and clustering was performed. While demonstrating the power of complexity reduction through the use of orthogonal datasets, structural models were examined for suitability, and ultimately nine proteins were found to be part of the complex. Using both sequence-based and structure-based tools, the phylogeny and homology of the three cage-forming proteins were explored, leading to insights that help to characterize a novel class of protein. Finally, through the analysis of another orthogonal dataset, it was shown that these cage-forming proteins are entirely essential to the viability of a *Mycoplasma pneumoniae* cell.

With respect to future research, there is still much to explore in relation to the cage complex. Evidence thus far suggests that the complex is formed of a dome component, which may act as a chaperone, as well as translocon machinery that can interact with the ribosome. Further proteomic studies are in progress to determine the identity of the remaining interaction partners. This workflow has additionally uncovered other complexes in the same dataset of *M. pneumoniae* whole-cell tomograms and should continue to be applied to uncover more.

Acknowledgements

I would like to thank my supervisor, Peer Bork, for the opportunity to pursue a PhD in his research group and the invaluable big-picture advice. I would also like to thank Julia Mahamid, the Chair of my Thesis Advisory Committee (TAC) and my unofficial second supervisor, for the opportunity to get involved in this project and all the support along the way. I further thank Anna Kreshuk, the third internal member of my TAC, for her help with everything related to machine learning, and especially for her kindness. Thanks also to Thomas Dandekar, my university adviser, for agreeing to join my TAC and providing good feedback over the course of my PhD.

I would also like to acknowledge the members of the Bork Group and the Mahamid Group, without whom these groups would not be what they are today with the support and atmosphere required for such high-calibre research. In particular, I cannot thank Rasmus enough for his time, patience, and especially enthusiasm while effectively becoming my mentor since joining EMBL two years ago. A special thanks also to Liang for his efforts acquiring the *M. pneumoniae* dataset, without which this project would not have happened.

I would also like to give my thanks to the Weekly *Mycoplasma* Meeting team—Maria, Michael, and Federico—for structured support and an approachable place for problem-solving and brainstorming, as well as to the participants of the more formal Monthly *Mycoplasma* Interest Group for critical feedback and interesting talks.

Thanks also to Luis, a former member of the Bork Group and my hosting group leader for three months in Shanghai, for the opportunity and amazing supervision.

A huge thank-you to my officemates Askarbek, Pamela, Marisa, Alice, Yeong, Moritz, and Daniel, who have provided thinking-buddy-style academic help, listened to personal problems, and just generally been great to have around for a wonderful atmosphere.

Finally, on the more personal side of things, I would like to thank my father, mother and brother, for long-distance encouragement; Maja, for love, support, and companionship through the roughest parts of the journey; and Imre, Martin, Dmitri, Denis, Omar, Ashwin, and Simon, for long-lasting friendship and reliable meme exchange.

References

1. Himmelreich R, Hubert H, Plagens H, Pirkl E, Li BC, and Herrmann R. Complete sequence analysis of the genome of the bacterium Mycoplasma pneumoniae. Nucleic Acids Res. 1996;24:4420–49.
2. He J, Liu M, Ye Z, et al. Insights into the pathogenesis of Mycoplasma pneumoniae (Review). Mol. Med. Rep. 2016;14:4030–6.
3. Nakane D, Kenri T, Matsuo L, and Miyata M. Systematic Structural Analyses of Attachment Organelle in Mycoplasma pneumoniae. PLoS Pathog. 2015;11:1–19.
4. Dandekar T, Huynen M, Regula JT, et al. Re-annotating the Mycoplasma pneumoniae genome sequence: Adding value, function and reading frames. Nucleic Acids Res. 2000;28:3278–88.
5. Hasselbring BM, Jordan JL, Krause RW, and Krause DC. Terminal organelle development in the cell wall-less bacterium Mycoplasma pneumoniae. Proc. Natl. Acad. Sci. U. S. A. 2006;103:16478–83.
6. Morowitz HJ. The completeness of molecular biology. Isr. J. Med. Sci. 1984;20:750–3.
7. Razin S. Peculiar properties of mycoplasmas: The smallest self-replicating prokaryotes. FEMS Microbiol. Lett. 1992;100:423–31.
8. Ochman H and Raghavan R. Excavating the Functional Landscape of Bacterial Cells. Science (80-.). 2009;326:1200–1.
9. Kühner S, Van Noort V, Betts MJ, et al. Proteome organization in a genome-reduced bacterium. Science (80-.). 2009;326:1235–40.
10. Yus E, Maier T, Michalodimitrakis K, et al. Impact of genome reduction on bacterial metabolism and its regulation. Science (80-.). 2009;326:1263–8.
11. Güell M, Van Noort V, Yus E, et al. Transcriptome complexity in a genome-reduced bacterium. Science (80-.). 2009;326:1268–71.
12. Razin S. Mycoplasmas. In: *Med. Microbiol. 4th Ed.* Ed. by Baron S. 4th ed. Galveston (TX): University of Texas Medical Branch at Galveston, 1996. Chap. 37.

13. Waites KB and Talkington DF. Mycoplasma pneumoniae and its role as a human pathogen. Clin. Microbiol. Rev. 2004;17:697–728.

14. Seybert A, Herrmann R, and Frangakis AS. Structural analysis of Mycoplasma pneumoniae by cryo-electron tomography. J. Struct. Biol. 2006;156:342–54.

15. Kammer GM, Pollack JD, and Klainer AS. Scanning-beam electron microscopy of Mycoplasma pneumoniae. J. Bacteriol. 1970;104:499–502.

16. Gaspari E, Malachowski A, Garcia-Morales L, et al. Model-driven design allows growth of Mycoplasma pneumoniae on serum-free media. npj Syst. Biol. Appl. 2020;6.

17. Balish MF and Krause DC. Cytadherence and the Cytoskeleton. In: *Mol. Biol. Pathog. Mycoplasmas.* 2002. Chap. 22:491–518. DOI: `10.1007/0-306-47606-1_22`.

18. Miyata M and Hamaguchi T. Integrated information and prospects for gliding mechanism of the pathogenic bacterium Mycoplasma pneumoniae. Front. Microbiol. 2016;7.

19. Henderson GP and Jensen GJ. Three-dimensional structure of Mycoplasma pneumoniae's attachment organelle and a model for its role in gliding motility. Mol. Microbiol. 2006;60:376–85.

20. Freundlich MM. Origin of the electron microscope. Science (80-.). 1963;142:185–8.

21. Gordon RE. Electron microscopy: A brief history and review of current clinical application. Methods Mol. Biol. 2014;1180:119–35.

22. Gan L and Jensen GJ. Electron tomography of cells. Q. Rev. Biophys. 2012;45:27–56.

23. Koning RI, Koster AJ, and Sharp TH. Advances in cryo-electron tomography for biology and medicine. Ann. Anat. 2018;217:82–96.

24. Neumüller J. Electron tomography—a tool for ultrastructural 3D visualization in cell biology and histology. Wiener Medizinische Wochenschrift 2018;168:322–9.

25. Wali N and Jagadeesh JM. Collection and handling of ultrathin serial sections for 3-dimensional reconstruction. J. Neurosci. Methods 1989;30:117–20.

26. Kremer JR, Mastronarde DN, and McIntosh JR. Computer visualization of three-dimensional image data using IMOD. J. Struct. Biol. 1996;116:71–6.

27. Koster AJ, Grimm R, Typke D, et al. Perspectives of molecular and cellular electron tomography. J. Struct. Biol. 1997;120:276–308.

28. Vinothkumar KR and Henderson R. Single particle electron cryomicroscopy: trends, issues and future perspective. Q. Rev. Biophys. 2016;49.

29. Wlodawer A and Dauter Z. ’Atomic resolution’: A badly abused term in structural biology. Acta Crystallogr. Sect. D Struct. Biol. 2017;73:379–80.

30. Yip KM, Fischer N, Paknia E, Chari A, and Stark H. Breaking the next cryo-EM resolution barrier – atomic resolution determination of proteins! bioRxiv 2020:2020.05.21.106740.

31. Nakane T, Kotecha A, Sente A, et al. Single-particle cryo-EM at atomic resolution. Nature 2020;587:152–6.

32. Frank J. Advances in the field of single-particle cryo-electron microscopy over the last decade. Nat. Protoc. 2017;12:209–12.

33. Hylton RK and Swulius MT. Challenges and triumphs in cryo-electron tomography. iScience 2021;24:102959.

34. Zhang P. Advances in cryo-electron tomography and subtomogram averaging and classification. Curr. Opin. Struct. Biol. 2019;58:249–58.

35. Förster F and Hegerl R. Structure Determination In Situ by Averaging of Tomograms. In: *Methods Cell Biol.* Vol. 2007. 79. 2007. Chap. 29:741–67. DOI: 10.1016/S0091-679X(06)79029-X.

36. Winkler H, Zhu P, Liu J, Ye F, Roux KH, and Taylor KA. Tomographic subvolume alignment and subvolume classification applied to myosin V and SIV envelope spikes. J. Struct. Biol. 2009;165:64–77.

37. Xue L, Lenz S, Zimmermann-Kogadeeva M, et al. Visualizing translation dynamics at atomic detail inside a bacterial cell. Nature 2022;610:205–11.

38. Galaz-Montoya JG and Ludtke SJ. The advent of structural biology in situ by single particle cryo-electron tomography. Biophys. Reports 2017;3:17–35.

39. Gierasch LM and Gershenson A. Post-reductionist protein science, or putting Humpty Dumpty back together again. Nat. Chem. Biol. 2009;5:774–7.

40. Asano S, Engel BD, and Baumeister W. In Situ Cryo-Electron Tomography: A Post-Reductionist Approach to Structural Biology. J. Mol. Biol. 2016;428:332–43.

41. Lučić V, Rigort A, and Baumeister W. Cryo-electron tomography: The challenge of doing structural biology in situ. J. Cell Biol. 2013;202:407–19.

42. Briggs JA. Structural biology in situ — the potential of subtomogram averaging. Curr. Opin. Struct. Biol. 2013;23:261–7.

43. Frangakis AS and Förster F. Computational exploration of structural information from cryo-electron tomograms. Curr. Opin. Struct. Biol. 2004;14:325–31.

44. Baumeister W. Cryo-electron tomography: The power of seeing the whole picture. Biochem. Biophys. Res. Commun. 2022;633:26–8.

45. Nickell S, Kofler C, Leis AP, and Baumeister W. A visual approach to proteomics. Nat. Rev. Mol. Cell Biol. 2006;7:225–30.

46. Robinson CV, Sali A, and Baumeister W. The molecular sociology of the cell. Nature 2007;450:973–82.

47. Förster F, Han BG, and Beck M. Visual Proteomics. In: *Methods Enzymol.* Vol. 483. Elsevier Inc., 2010. Chap. 11:215–43. DOI: `10.1016/S0076-6879(10)83011-3`.

48. Bäuerlein FJ and Baumeister W. Towards Visual Proteomics at High Resolution. J. Mol. Biol. 2021;433.

49. Baumeister W. Cryo-electron tomography: A long journey to the inner space of cells. Cell 2022;185:2649–52.

50. O'Reilly FJ, Xue L, Graziadei A, et al. In-cell architecture of an actively transcribing-translating expressome. Science (80-.). 2020;369:554–7.

51. Danev R and Baumeister W. Expanding the boundaries of cryo-EM with phase plates. Curr. Opin. Struct. Biol. 2017;46:87–94.

52. Sigworth FJ. Principles of cryo-EM single-particle image processing. Reprod. Syst. Sex. Disord. 2016;65:57–67.

53. Danev R, Buijsse B, Khoshouei M, Plitzko JM, and Baumeister W. Volta potential phase plate for in-focus phase contrast transmission electron microscopy. Proc. Natl. Acad. Sci. U. S. A. 2014;111:15635–40.

54. Danev R, Tegunov D, and Baumeister W. Using the volta phase plate with defocus for cryo-em single particle analysis. Elife 2017;6:1–9.

55. Buijsse B, Trompenaars P, Altin V, Danev R, and Glaeser RM. Spectral DQE of the Volta phase plate. Ultramicroscopy 2020;218:113079.

56. Lenz S, Sinn LR, O'Reilly FJ, Fischer L, Wegner F, and Rappsilber J. Reliable identification of protein-protein interactions by crosslinking mass spectrometry. Nat. Commun. 2021;12:1–11.

57. Beck M, Malmström JA, Lange V, Schmidt A, Deutsch EW, and Aebersold R. Visual proteomics of the human pathogen Leptospira interrogans. Nat. Methods 2009;6:817–23.

58. Ortiz JO, Förster F, Kürner J, Linaroudis AA, and Baumeister W. Mapping 70S ribosomes in intact cells by cryoelectron tomography and pattern recognition. J. Struct. Biol. 2006;156:334–41.

59. Scheres SH. RELION: Implementation of a Bayesian approach to cryo-EM structure determination. J. Struct. Biol. 2012;180:519–30.

60. Hall SR, Allen FH, and Brown ID. The crystallographic information file (CIF): a new standard archive file for crystallography. Acta Crystallogr. Sect. A 1991;47:655–85.

61. Hall SR. The STAR File: A New Format for Electronic Data Transfer and Archiving. J. Chem. Inf. Comput. Sci. 1991;31:326–33.

62. Hall SR and Spadaccini N. The STAR File: Detailed Specifications. J. Chem. Inf. Comput. Sci. 1994;34:505–8.

63. Spadaccini N and Hall SR. Extensions to the STAR file syntax. J. Chem. Inf. Model. 2012;52:1901–6.

64. Heymann JB, Chagoyen M, and Belnap DM. Common conventions for interchange and archiving of three-dimensional electron microscopy information in structural biology. J. Struct. Biol. 2005;151:196–207.

65. Diebel J. Representing attitude: Euler angles, unit quaternions, and rotation vectors. Tech. rep. 2006.

66. Mukhamediev RI, Popova Y, Kuchin Y, et al. Review of Artificial Intelligence and Machine Learning Technologies: Classification, Restrictions, Opportunities and Challenges. Mathematics 2022;10:1–25.

67. Sommer C and Gerlich DW. Machine learning in cell biology – teaching computers to recognize phenotypes. J. Cell Sci. 2013;126:5529–39.

68. Hong Y, Hou B, Jiang H, and Zhang J. Machine learning and artificial neural network accelerated computational discoveries in materials science. Wiley Interdiscip. Rev. Comput. Mol. Sci. 2020;10:1–21.

69. Belkin M, Hsu D, Ma S, and Mandal S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. Proc. Natl. Acad. Sci. U. S. A. 2019;116:15849–54.

70. Dietterich T. Overfitting and Undercomputing in Machine Learning. ACM Comput. Surv. 1995;27:326–7.

71. Abiodun OI, Jantan A, Omolara AE, et al. Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition. IEEE Access 2019;7:158820–46.

72. Rasamoelina AD, Adjailia F, and Sincak P. A Review of Activation Function for Artificial Neural Network. SAMI 2020 - IEEE 18th World Symp. Appl. Mach. Intell. Informatics, Proc. 2020:281–6.

73. Rumelhart DE, Hinton GE, and Williams RJ. Learning representations by back-propagating errors. Nature 1986;323:533–6.

74. Lecun Y, Bengio Y, and Hinton G. Deep learning. Nature 2015;521:436–44.

75. LeCun Y, Boser B, Denker JS, et al. Backpropagation Applied to Handwritten Zip Code Recognition. Neural Comput. 1989;1:541–51.

76. Lecun Y, Bottou L, Bengio Y, and Haffner P. Gradient-based learning applied to document recognition. Proc. IEEE 1998;86:2278–324.

77. Krizhevsky A, Sutskever I, and Hinton GE. ImageNet classification with deep convolutional neural networks. Commun. ACM 2017;60:84–90.

78. Simonyan K and Zisserman A. Very deep convolutional networks for large-scale image recognition. 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc. 2015:1–14.

79. He K, Zhang X, Ren S, and Sun J. Deep residual learning for image recognition. In: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* Vol. 2016-Decem. 2016: 770–8. DOI: 10.1109/CVPR.2016.90.

80. Ronneberger O, Fischer P, and Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: *Med. Image Comput. Comput. Interv. – MICCAI 2015*. Ed. by Navab N, Hornegger J, Wells WM, and Frangi AF. Cham: Springer International Publishing, 2015: 234–41.

81. Alzubaidi L, Zhang J, Humaidi AJ, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J. Big Data 2021;8.

82. Li Y, Huang C, Ding L, Li Z, Pan Y, and Gao X. Deep learning in bioinformatics: Introduction, application, and perspective in the big data era. Methods 2019;166:4–21.

83. Lipton ZC. The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability is Both Important and Slippery. Queue 2018;16:31–57.

84. Kuhlman B and Bradley P. Advances in protein structure prediction and design. Nat. Rev. Mol. Cell Biol. 2019;20:681–97.

85. Waterhouse A, Bertoni M, Bienert S, et al. SWISS-MODEL: Homology modelling of protein structures and complexes. Nucleic Acids Res. 2018;46:W296–W303.

86. Ovchinnikov S, Park H, Varghese N, et al. Protein structure determination using metagenome sequence data. Science (80-.). 2017;355:294–8.

87. Jumper J, Evans R, Pritzel A, et al. Applying and improving AlphaFold at CASP14. Proteins Struct. Funct. Bioinforma. 2021;89:1711–21.

88. Jumper J, Evans R, Pritzel A, et al. Highly accurate protein structure prediction with AlphaFold. Nature 2021;596:583–9.

89. Kryshtafovych A, Schwede T, Topf M, Fidelis K, and Moult J. Critical assessment of methods of protein structure prediction (CASP)—Round XIV. Proteins Struct. Funct. Bioinforma. 2021;89:1607–17.

90. Frangakis AS. It's noisy out there! A review of denoising techniques in cryo-electron tomography. J. Struct. Biol. 2021;213:107804.

91. Kunz M and Frangakis AS. Super-sampling SART with ordered subsets. J. Struct. Biol. 2014;188:107–15.

92. Lehtinen J, Munkberg J, Hasselgren J, et al. Noise2Noise: Learning image restoration without clean data. 35th Int. Conf. Mach. Learn. ICML 2018 2018;7:4620–31.

93. Çiçek Ö, Abdulkadir A, Lienkamp SS, Brox T, and Ronneberger O. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In: *Med. Image Comput. Comput. Interv. – MICCAI 2016*. Ed. by Ourselin S, Joskowicz L, Sabuncu MR, Unal G, and Wells W. Cham: Springer International Publishing, 2016: 424–32.

94. Tegunov D and Cramer P. Real-time cryo-electron microscopy data preprocessing with Warp. Nat. Methods 2019;16:1146–52.

95. Bepler T, Kelley K, Noble AJ, and Berger B. Topaz-Denoise: general deep denoising models for cryoEM and cryoET. Nat. Commun. 2020;11:1–12.

96. Bepler T, Morin A, Rapp M, et al. Positive-unlabeled convolutional neural networks for particle picking in cryo-electron micrographs. Nat. Methods 2019;16:1153–60.

97. Buchholz TO, Krull A, Shahidi R, Pigino G, Jékely G, and Jug F. Content-aware image restoration for electron microscopy. In: *Three-Dimensional Electron Microsc.* Ed. by Müller-Reichert T and Pigino GBTMiCB. Vol. 152. Academic Press, 2019: 277–89. DOI: `10.1016/bs.mcb.2019.05.001`.

98. Li H, Zhang H, Wan X, et al. Noise-Transfer2Clean: Denoising cryo-EM images based on noise modeling and transfer. Bioinformatics 2022;38:2022–9.

99. Chung JM, Durie CL, and Lee J. Artificial Intelligence in Cryo-Electron Microscopy. Life 2022;12:1–12.

100. Wang F, Gong H, Liu G, et al. DeepPicker: A deep learning approach for fully automated particle picking in cryo-EM. J. Struct. Biol. 2016;195:325–36.

101. Zhu Y, Ouyang Q, and Mao Y. A deep convolutional neural network approach to single-particle recognition in cryo-electron microscopy. BMC Bioinformatics 2017;18:1–10.

102. Redmon J and Farhadi A. YOLO9000: Better, Faster, Stronger. In: *2017 IEEE Conf. Comput. Vis. Pattern Recognit.* Vol. January. 2017: 6517–25. DOI: 10.1109/CVPR.2017.690.

103. Wagner T, Merino F, Stabrin M, et al. SPHIRE-crYOLO is a fast and accurate fully automated particle picker for cryo-EM. Commun. Biol. 2019;2:1–13.

104. Moebel E, Martinez-Sanchez A, Lamm L, et al. Deep learning improves macromolecule identification in 3D cellular cryo-electron tomograms. Nat. Methods 2021;18:1386–94.

105. Teresa-Trueba I de, Goetz SK, Mattausch A, et al. Convolutional networks for supervised mining of molecular patterns within cellular context. Nat. Methods 2023.

106. Xue L. In-cell structural biology of protein synthesis in Mycoplasma pneumoniae. PhD thesis. Heidelberg University, 2022.

107. Halbedel S, Hames C, and Stülke J. In vivo activity of enzymatic and regulatory components of the phosphoenolpyruvate: sugar phosphotransferase system in Mycoplasma pneumoniae. J. Bacteriol. 2004;186:7936–43.

108. Mastronarde DN. Automated electron microscope tomography using robust prediction of specimen movements. J. Struct. Biol. 2005;152:36–51.

109. Schorb M, Haberbosch I, Hagen WJ, Schwab Y, and Mastronarde DN. Software tools for automated transmission electron microscopy. Nat. Methods 2019;16:471–7.

110. Hagen WJ, Wan W, and Briggs JA. Implementation of a cryo-electron tomography tilt-scheme optimized for high resolution subtomogram averaging. J. Struct. Biol. 2017;197:191–8.

111. Fukuda Y, Laugks U, Lučić V, Baumeister W, and Danev R. Electron cryotomography of vitrified cells with a Volta phase plate. J. Struct. Biol. 2015;190:143–54.

112. Mastronarde DN and Held SR. Automated tilt series alignment and tomographic reconstruction in IMOD. J. Struct. Biol. 2017;197:102–13.

113. Tegunov D, Xue L, Dienemann C, Cramer P, and Mahamid J. Multi-particle cryo-EM refinement with M visualizes ribosome-antibiotic complex at 3.5 Å in cells. Nat. Methods 2021;18:186–93.

114. Tang G, Peng L, Baldwin PR, et al. EMAN2: An extensible image processing suite for electron microscopy. J. Struct. Biol. 2007;157:38–46.

115. Hrabe T, Chen Y, Pfeffer S, Kuhn Cuellar L, Mangold AVV, and Förster F. PyTom: A python-based toolbox for localization of macromolecules in cryo-electron tomograms and subtomogram analysis. J. Struct. Biol. 2012;178:177–88.

116. Nickell S, Förster F, Linaroudis A, et al. TOM software toolbox: Acquisition and analysis for electron tomography. J. Struct. Biol. 2005;149:227–34.

117. Böhm J, Frangakis AS, Hegerl R, Nickell S, Typke D, and Baumeister W. Toward detecting and identifying macromolecules in a cellular context: Template matching applied to electron tomograms. Proc. Natl. Acad. Sci. U. S. A. 2000;97:14245–50.

118. Lucas BA, Himes BA, Xue L, Grant T, Mahamid J, and Grigorieff N. Locating macromolecular assemblies in cells by 2d template matching with cistem. Elife 2021;10:1–25.

119. Virtanen P, Gommers R, Oliphant TE, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods 2020;17:261–72.

120. Burnley T, Palmer CM, and Winn M. Recent developments in the CCP-EM software suite. Acta Crystallogr. Sect. D Struct. Biol. 2017;73:469–77.

121. Vizarraga D, Kawamoto A, Matsumoto U, et al. Immunodominant proteins P1 and P40/P90 from human pathogen Mycoplasma pneumoniae. Nat. Commun. 2020;11:1–3.

122. Aparicio D, Scheffer MP, Marcos-Silva M, et al. Structure and mechanism of the Nap adhesion complex from the human pathogen Mycoplasma genitalium. Nat. Commun. 2020;11:1–10.

123. Nakane T and Scheres SH. Multi-body Refinement of Cryo-EM Images in RELION. In: *Methods Mol. Biol.* Vol. 2215. 2021: 145–60. DOI: 10.1007/978-1-0716-0966-8_7.

124. Siciliano RA, Lippolis R, and Mazzeo MF. Proteomics for the investigation of surface-exposed proteins in probiotics. Front. Nutr. 2019;6.

125. Varadi M, Anyango S, Deshpande M, et al. AlphaFold Protein Structure Database: Massively expanding the structural coverage of protein-sequence space with high-accuracy models. Nucleic Acids Res. 2022;50:D439–D444.

126. Zundert GC van and Bonvin AM. Fast and sensitive rigid-body fitting into cryo-EM density maps with PowerFit. AIMS Biophys. 2015;2:73–87.

127. Mistry J, Chuguransky S, Williams L, et al. Pfam: The protein families database in 2021. Nucleic Acids Res. 2021;49:D412–D419.

128. Bateman A, Martin MJ, Orchard S, et al. UniProt: the universal protein knowledgebase in 2021. Nucleic Acids Res. 2021;49:D480–D489.

129. McGowin CL, Liang M, Martin DH, and Pyles RB. Mycoplasma genitalium-encoded MG309 activates NF-κB via toll-like receptors 2 and 6 to elicit proinflammatory cytokine secretion from human genital epithelial cells. Infect. Immun. 2009;77:1175–81.

130. Ma L, Taylor S, Jensen JS, Myers L, Lillis R, and Martin DH. Short tandem repeat sequences in the Mycoplasma genitalium genome and their use in a multilocus genotyping system. BMC Microbiol. 2008;8:1–13.

131. Hallamaa KM, Browning GF, and Tang SL. Lipoprotein multigene families in Mycoplasma pneumoniae. J. Bacteriol. 2006;188:5393–9.

132. O'Reilly FJ and Rappsilber J. Cross-linking mass spectrometry: methods and applications in structural, molecular and systems biology. Nat. Struct. Mol. Biol. 2018;25:1000–8.

133. Merkley ED, Rysavy S, Kahraman A, Hafen RP, Daggett V, and Adkins JN. Distance restraints from crosslinking mass spectrometry: Mining a molecular dynamics simulation database to evaluate lysine-lysine distances. Protein Sci. 2014;23:747–59.

134. Kao A, Chiu Cl, Vellucci D, et al. Development of a Novel Cross-linking Strategy for Fast and Accurate Identification of Cross-linked Peptides of Protein Complexes. Mol. Cell. Proteomics 2011;10:M110.002170.

135. Maier T, Schmidt A, Güell M, et al. Quantification of mRNA and protein and integration with protein turnover in a bacterium. Mol. Syst. Biol. 2011;7:1–12.

136. Elfmann C, Zhu B, Pedreira T, et al. MycoWiki: Functional annotation of the minimal model organism Mycoplasma pneumoniae. Front. Microbiol. 2022;13:1–10.

137. Wong SMS, Gawronski JD, Lapointe D, and Akerley BJ. High-Throughput Insertion Tracking by Deep Sequencing for the Analysis of Bacterial Pathogens. In: *High-Throughput Next Gener. Seq. Methods Appl.* Ed. by Kwon YM and Ricke SC. Totowa, NJ: Humana Press, 2011: 209–22. DOI: 10.1007/978-1-61779-089-8_15.

138. Opijnen T van, Bodi KL, and Camilli A. Tn-seq: high-throughput parallel sequencing for fitness and genetic interaction studies in microorganisms. Nat. Methods 2009;6:767–72.

139. Miravet-Verde S, Burgos R, Delgado J, Lluch-Senar M, and Serrano L. FASTQINS and ANUBIS: two bioinformatic tools to explore facts and artifacts in transposon sequencing and essentiality studies. Nucleic Acids Res. 2020;48:e102–e102.

140. Lluch-Senar M, Delgado J, Chen WH, et al. Defining a minimal cell: essentiality of small ORFs and ncRNAs in a genome-reduced bacterium. Mol. Syst. Biol. 2015;11:780.

141. Yus E, Lloréns-Rico V, Martínez S, et al. Determination of the Gene Regulatory Network of a Genome-Reduced Bacterium Highlights Alternative Regulation Independent of Transcription Factors. Cell Syst. 2019;9:143–158.e13.

142. Altschul SF, Gish W, Miller W, Myers EW, and Lipman DJ. Basic Local Alignment Search Tool. J. Mol. Biol. 1990;215:403–10.

143. Altschul SF, Madden TL, Schäffer AA, et al. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. Nucleic Acids Res. 1997;25:3389–402.

144. Neimark H and Kocan KM. The cell wall-less rickettsia Eperythrozoon wenyonii is a Mycoplasma. FEMS Microbiol. Lett. 1997;156:287–91.

145. Neimark H, Johansson KE, Rikihisa Y, and Tully JG. Proposal to transfer some members of the genera Haemobartonella and Eperythrozoon to the genus Mycoplasma with descriptions of 'Candidatus Mycoplasma haemofelis', 'Candidatus Mycoplasma haemomuris', 'Candidatus Mycoplasma haemosuis' and 'Candidatus Mycopl. Int. J. Syst. Evol. Microbiol. 2001;51:891–9.

146. Durairaj J, Akdel M, Ridder D de, and Dijk AD van. Geometricus represents protein structures as shape-mers derived from moment invariants. Bioinformatics 2020;36:I718–I725.

147. Durairaj J, Akdel M, and De Ridder D. Fast and adaptive protein structure representations for machine learning. bioRxiv 2021:2021.04.07.438777.

148. Akdel M, Pires DE, Pardo EP, et al. A structural biology community assessment of AlphaFold2 applications. Nat. Struct. Mol. Biol. 2022:2021.09.26.461876.

149. Durairaj J, Pereira J, Akdel M, and Schwede T. What is hidden in the darkness? Characterization of AlphaFold structural space. bioRxiv 2022:2022.10.11.511548.

150. Van Kempen M, Kim SS, Tumescheit C, Mirdita M, Söding J, and Steinegger M. Foldseek: fast and accurate protein structure search. bioRxiv 2022;1:2022.02.07.479398.

151. Steinegger M and Söding J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. Nat. Biotechnol. 2017;35:1026–8.

152. Harris CR, Millman KJ, Walt SJ van der, et al. Array programming with NumPy. Nature 2020;585:357–62.

153. McKinney W. Data Structures for Statistical Computing in Python. In: *Proc. 9th Python Sci. Conf.* Ed. by Walt S van der and Millman J. SciPy. 2010: 56–61. DOI: `10.25080/majora-92bf1922-00a`.

154. Hunter JD. Matplotlib: A 2D graphics environment. Comput. Sci. Eng. 2007;9:90–5.

155. Waskom M. Seaborn: Statistical Data Visualization. J. Open Source Softw. 2021;6:3021.

156. Hagberg AA, Schult DA, and Swart PJ. Exploring network structure, dynamics, and function using NetworkX. In: *7th Python Sci. Conf. (SciPy 2008).* Ed. by Varoquaux G, Vaught T, and Millman J. SciPy. 2008: 11–6.

157. Berman HM, Westbrook J, Feng Z, et al. The Protein Data Bank. Nucleic Acids Res. 2000;28:235–42.

158. Evans R, O'Neill M, Pritzel A, et al. Protein complex prediction with AlphaFold-Multimer. bioRxiv 2021:2021.10.04.463034.

159. Nakane T, Kimanius D, Lindahl E, and Scheres SH. Characterisation of molecular motions in cryo-EM single-particle data by multi-body refinement in RELION. Elife 2018;7:1–18.

160. Lyumkis D. Challenges and opportunities in cryo-EM single-particle analysis. J. Biol. Chem. 2019;294:5181–97.

161. Pogliano JA and Beckwith J. SecD and SecF facilitate protein export in Escherichia coli. EMBO J. 1994;13:554–61.

162. Pettersen EF, Goddard TD, Huang CC, et al. UCSF Chimera - A visualization system for exploratory research and analysis. J. Comput. Chem. 2004;25:1605–12.

163. Komar J, Botte M, Collinson I, Schaffitzel C, and Berger I. Chapter Two - ACEMBLing a Multiprotein Transmembrane Complex: The Functional SecYEG-SecDF-YajC-YidC Holotranslocon Protein Secretase/Insertase. In: *Membr. Proteins—Production Funct. Charact.* Ed. by Shukla AKBTMiE. Vol. 556. Academic Press, 2015: 23–49. DOI: `https://doi.org/10.1016/bs.mie.2014.12.027`.

164. Bariya P and Randall LL. Coassembly of SecYEG and SecA Fully Restores the Properties of the Native Translocon. J. Bacteriol. 2019;201:e00493–18.

165. Fink AL. Chaperone-Mediated Protein Folding. Physiol. Rev. 1999;79:425–49.