technische universität
dortmund

MAX-PLANCK-INSTITUT
FÜR MOLEKULARE PHYSIOLOGIE

Fully automated processing and optimization of single particle and filamentous transmission electron cryomicroscopy samples

Markus Stabrin
September 2022

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor rerum naturalium
at
Fakultät Physik, Technische Universität Dortmund
Abteilung für Strukturbiochemie, Max-Planck-Institut für molekulare Physiologie

Supervised by
Prof. Dr. Metin Tolan and Prof. Dr. Stefan Raunser

This thesis is set in Libertinus (Serif, Sans and Math) and Fira Code,
typeset using LaTeX with LuaTeX from TeX-Live 2020.

First referee: **Prof. Dr. Metin Tolan**

Lehrstuhl für Experimentelle Physik I
Fakultät Physik
Technische Universität Dortmund

Second referee: **Prof. Dr. Stefan Raunser**

Abteilung für Strukturbiochemie
Max-Planck-Institut für molekulare Physiologie

Fakultät für Chemie und Chemische Biologie
Technische Universität Dortmund

Work here has been carried out at the department of Structural Biochemistry at the Max-Planck-Institut of Molekular Physiology.

Abstract

Single particle cryo-EM is getting increasingly accessible to researchers from diverse research areas. Furthermore, recent innovations in cryo-EM hardware development and data acquisition strategies have led to an increase in data quality as well as quantity. However, the overall quality of a data set does not directly correlate with the quality of a single image or the number of images collected, but also strongly depends on the sample itself. Nevertheless, with great data comes great responsibility and just having a large data set is not necessarily advantageous. To get the most out of a data collection, the researcher needs to carefully monitor and curate all data, ideally while its still being collected.

Therefore, automated data processing and the analysis of the collected data live during acquisition becomes increasingly important. To get the most information in the shortest amount of time, ideally, all major pre-processing steps would be executed while the data are still collected. In this way, the researcher gets direct feedback about the sample quality and has the chance to make necessary adjustments to the data collection. While there are several tools available to execute the processing pipeline, they all use a static set of input settings for each individual task, limiting their applicability.

In this thesis, I present TranSPHIRE, a tool for fully automated on-the-fly data processing. It executes all the important pre-processing steps required for the processing of single particle projects, as well as filamentous samples, in a parallel manner. Additionally, the important metrics of each processing step are presented via the TranSPHIRE GUI, allowing a fast evaluation of all parameters and the data collection itself. TranSPHIRE also includes a machine learning based feedback loop, which enables the optimization of particle picking for any given sample. Specifically, the loop performs iterations of particle picking, 2D classification, and 2D class selection, followed by training of a new model for particle picking. The curated particles can subsequently be subjected to a 3D refinement within TranSPHIRE. For further analysis, the output particles and volumes can eventually be transferred to other software packages, such as SPHIRE.

I demonstrate the capabilities of the TranSPHIRE pipeline based on three different scenarios: A previously unknown data set; a data set consisting of two sub-populations, where only one is targeted for particle picking; and a filamentous sample. All three scenarios lead to a high-resolution 3D reconstruction of the target protein in a fully automated manner. Therefore, fully automated data processing and optimization could pave the way for high-throughput screenings of unknown samples without user intervention.

Despite decades of research, processing of filamentous cryo-EM samples remains challenging. One of the reasons is that most 3D refinement approaches require prior knowledge about the helical symmetry parameters. However, for most samples the helical symmetry differs locally, leading to model bias, low-resolution results, or even incorrectly reconstructed structures. Here, I present *sp_meridien_alpha.py*, a modification of the single particle 3D refinement program *sp_meridien.py*, to allow filamentous processing in the SPHIRE package. My refinement approach utilizes filamentous constraints to help convergence and does not require previous knowledge about the helical symmetry. Based on two examples, a tobacco mosaic virus and an actomyosin data set, I show that the final resolution and overall map quality achieved by *sp_meridien_alpha.py* surpasses the one achieved by *sp_meridien.py*.

In summary, the software tool TranSPHIRE and the filamentous 3D refinement program *sp_meridien_alpha.py* combined simplify the cryo-EM data collection and processing and thereby present a valuable contribution to the field.

Zusammenfassung

Die Einzelpartikel-Kryo-EM wird immer zugänglicher für Forschende aus den verschiedensten Forschungsgebieten. Weiterhin haben neueste Innovationen im Bereich der verwendeten Hardware und neuartige Strategien bei der Datenakquise zu einem Anstieg der Qualität, sowie der Quantität von Daten geführt. Allerdings korreliert die Qualität des Datensatzes nicht direkt mit der Qualität eines einzelnen Bildes oder der Anzahl an aufgenommenen Bildern, sondern ist von der verwendeten Probe abhängig. Doch aus großen Daten folgt große Verantwortung und nur einen großen Datensatz aufzunehmen ist nicht immer vorteilhaft. Um das meiste aus der Datenakquise herauszuholen, muss der Forschende diese genaustens überwachen und die Daten durchgängig begutachten und bewerten. Dies geschieht idealerweise bereits während der Datenakquise.

Daher bekommt die automatische Prozessierung und Analyse der aufgenommenen Daten während der Datenakquise eine immer größer werdende Bedeutung. Um alle nötigen Informationen in möglichst kurzer Zeit zusammenzutragen, sollten idealerweise alle wichtigen Pre-Processing Schritte noch während der Akquise durchgeführt werden. Auf diese Weise ist es möglich aus den Informationen Rückschlüsse auf die Qualität der Probe zu ziehen und in die Datenakquise einzugreifen. Es gibt bereits verschiedenste Tools zum Ausführen einer solchen Pipeline, doch arbeiten diese mit statischen Settings, wodurch ihr Anwendungsgebiet eingeschränkt ist.

In dieser Doktorarbeit präsentiere ich TranSPHIRE; Ein Tool zur vollautomatischen Datenprozessierung während der Datenaufnahme. Dabei führt es alle wichtigen Pre-Processing Schritte parallel für Projekte der Einzelpartikelanalyse, sowie filamentöse Proben, aus. Weiterhin präsentiert es alle Informationen über die einzelnen Schritte der Datenprozessierung innerhalb der TranSPHIRE GUI, wodurch eine schnelle Evaluation der Datenakquise ermöglicht wird. TranSPHIRE beinhaltet zusätzlich die auf maschinellem Lernen basierte „Feedback loop“, welche das „Particle picking“ so optimiert, dass es sich an jede Probe anpasst. Genauer gesagt besteht diese aus einem iterativen Prozess bestehend aus „Particle picking“, „2D classification“, „2D class selection“ und dem Trainieren eines neuen Modells für das „Particle picking“. Die auf diese Weise gereinigten und optimierten Partikel können daraufhin in ein anschließendes „3D refinement“ gegeben werden. Für die weitere Analyse der Daten können die erzeugten Partikel und Volumen in andere Softwarepakete, wie beispielsweise SPHIRE, überführt werden.

Die Möglichkeiten der TranSPHIRE Pipeline habe ich anhand von drei Szenarios demonstriert: Eine zuvor unbekannte Probe, einem Datensatz mit zwei Populationen und eine filamentöse Probe. In allen drei Fällen konnte vollautomatisch eine hochaufgelöste Struktur des gewünschten Proteins erreicht werden. Daher könnte die vollautomatische Prozessierung und Optimierung den Weg ebnen, um High-Throughput-Screening von unbekannten Proben für die Kryo-EM zu ermöglichen.

Trotz jahrelanger Forschung ist die Prozessierung von filamentösen Proben weiterhin eine Herausforderung. Dies liegt unter anderem daran, dass die meisten „3D refinement“ Algorithmen Vorwissen über die helikale Symmetry des Proteins benötigen. Allerdings ist die Symmetry in den meisten Fällen lokal begrenzt, wodurch es zu niedrigen Auflösungen oder sogar inkorrekten Strukturen kommen kann. Hier präsentiere ich *sp_meridien_alpha.py*, eine Modifikation des „3D refinement“ Programms *sp_meridien.py*, welches die Prozessierung von filamentösen Proben im SPHIRE Software Paket ermöglicht. Dabei erfordert mein angepasster Algorithmus kein Wissen über die helikale Symmetry, sondern es werden Einschränkungen vom Filament selbst genutzt. In dieser Arbeit zeige ich anhand von zwei Beispielen, einem Datensatz des Tabakmosaikvirus und einem Actomyosin Datensatz, dass die erreichte Auflösung, sowie die Qualität der Rekonstruktion, welche von *sp_meridien_alpha.py* erzeugt wurden, die von *sp_meridien.py* übertreffen.

Zusammenfassend lässt sich sagen, dass die Kombination aus TranSPHIRE und dem „3D refinement“ Programm *sp_meridien_alpha.py* die Datenakquise und die Prozessierung in der Kryo-EM vereinfacht und stellt daher einen wertvollen Beitrag für das Feld dar.

Contents

List of Figures

List of Tables

1 Introduction

Biological mechanisms depend on the interactions between various participators such as proteins, deoxyribonucleic acid (DNA), viruses, single molecules, membranes, and toxins and their environment. In order to analyze the underlying mechanisms it is crucial to investigate the three dimensional (3D) structures of the whole system, i.e., all participating components. Therefore, high-resolution structural biology methods, such as transmission electron cryomicroscopy (cryo-EM), X-ray crystallography, or nuclear magnetic resonance spectroscopy (NMR), are required to shed light on the underlying mechanisms and pave the way for drug discovery research to find possible treatments.

1.1 Structural biology

Proteins are macromolecules that are composed of at least one chain of amino acids (monomer). If a protein consists of more than one monomer, the protein is referred to as a multimer. The 3D structure of a protein can be characterized by four structural categories: the primary structure, the secondary structure, the tertiary structure, and the quaternary structure (Figure 1.1a). The primary structure is the amino acid sequence of which each chain of the protein is composed. Directly neighboring amino acids form secondary structural elements like α helices, β sheets, β turns, and Ω loops. Interactions between amino acids that are farther away from each other fold the secondary structural elements to the tertiary structure. A quaternary structure is only present in multimers and describes the interaction between the individual monomers to form the final 3D structure of the protein [99].

On the one hand, the protein structure can be simply described by the four structural categories. On the other hand, predicting the final 3D structure is far from trivial, because environmental factors like the pH-value, electric charges, proteins that assist with protein folding (such as chaperones), and many others can have a huge influence on the final appearance [42] (Figure 1.1b).

Consequently, methods of high-resolution structural biology are required to analyze and identify the 3D structure and shed light on the underlying mechanisms of protein interactions.

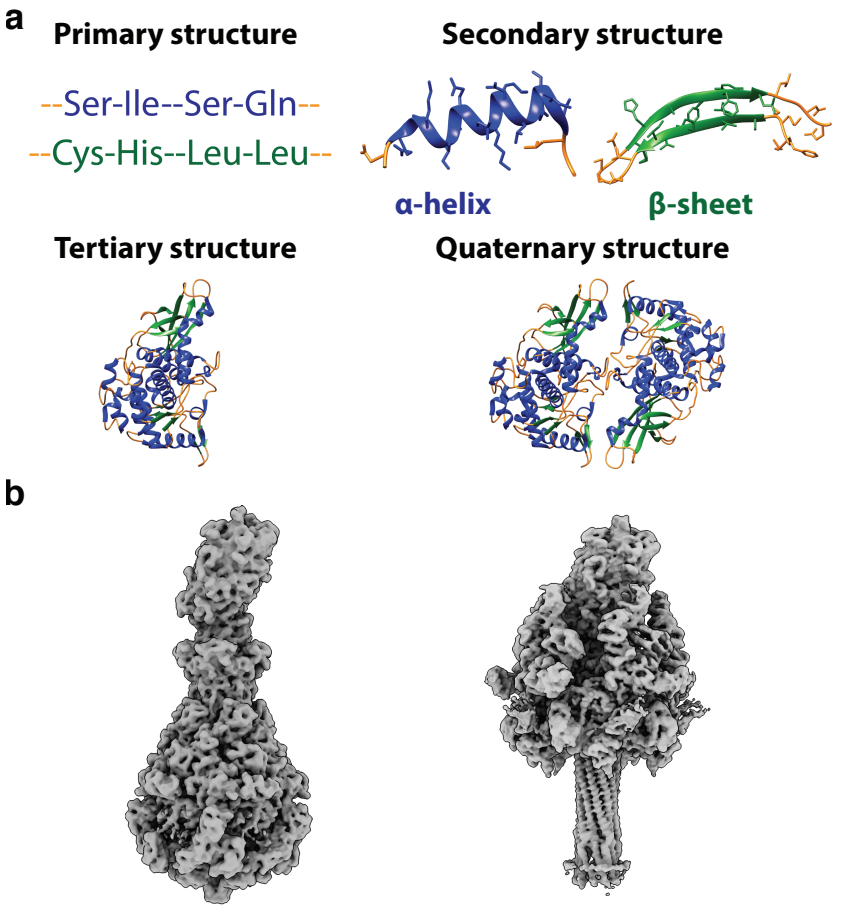

Figure 1.1: Structural elements of a protein. **a** Exemplary diagram of the structural elements of a protein. Figure adapted from [117]. **b** Even though the right and the left structure share the same primary structure, their secondary, tertiary, and quaternary structural elements differ significantly, emphasizing the need of high-resolution structural biology methods. Figure taken and adapted from [118].

1.2 Methods of high-resolution structural biology

While there exist several methods to analyze the presence and interactions of biological samples, three methods are commonly used to resolve the structure with close-to-atomic resolution: NMR, X-ray crystallography, and cryo-EM.

1.2.1 Nuclear magnetic resonance spectroscopy

With NMR the interactions between electromagnetic radiation and atoms and their direct neighbours is analyzed. When exposed to a magnetic field, atomic nuclei absorb electromagnetic radiation at a nucleus specific frequency. The strength of the absorption is not only dependent on the excited nuclei, but also on the atoms in the nearby environment. It is possible to calculate

the 3D structure of the sample at atomic resolution based on the information about the excited atoms and the distance to their neighbours [10]. However, the number of contacts increases with the total number of atoms in the sample and the method is mostly applicable to samples with a molecular weight of about 30 kDa or less [74].

1.2.2 X-ray crystallography

Another very popular method is X-ray crystallography, where a biological sample is crystallized. The resulting crystal is illuminated by an X-ray beam which results in a diffraction pattern based on the crystal lattice. By rotating the crystal during the illumination the information from the diffractions patterns at different angles can be used to calculate the 3D structure of the sample [116]. This method used to be the most popular one for high-resolution structural biology, due to the achievable resolution below 1 Å, which is about the size of a single atom [59]. The downside of the method is that the sample needs to arrange in a crystal lattice in the first place, which is highly dependent on the size, complexity, and symmetry of the sample as well as external factors like temperature and pressure. Additionally, the formation of a crystal by itself can change the native 3D structure of the observed sample [76].

1.2.3 Transmission electron cryomicroscopy

Cryo-EM is the newest of the high-resolution structural biology methods and received tremendous attention when the Nobel Prize in chemistry was awarded "for developing cryo-electron microscopy for the high-resolution structure determination of biomolecules in solution" in 2017 [125]. In cryo-EM, the sample is fixated in a thin, vitrified ice layer within the microscope and illuminated with a beam of accelerated electrons [84, 55]. Recent advances in hard- and software led to the so called "Resolution Revolution" [67] which enabled the reconstruction of 3D structures up to 1.2 Å resolution [81, 137]. Even though the highest resolutions are reserved for very rigid samples, the average regularly achieved resolution is nowadays in the range of 3.0 Å to 7.5 Å [84]. The molecular weight of the protein should ideally be about 100 kDa or higher, because biological samples are very fragile when they interact with the electrons of the beam. Consequently, only low dose rates can be used during image acquisition to preserve the 3D structure, leading to exceptionally low signal-to-noise ratios (SNRs) in the acquired images. Nevertheless, the Wang lab recently resolved a protein of about 50 kDa to near-atomic-resolution [33], highlighting the potential of the new technological advances.

The two main branches of cryo-EM are single particle analysis (SPA) and transmission electron cryotomography (cryo-ET). In SPA hundreds to millions of projections of the same sample from different, but unknown, angles are collected. With the help of statistical methods, the unknown angles are determined and the 3D structure can be calculated [2]. In the case of cryo-ET, the sample holder is rotated with a known angle and for each one image is acquired. On the one hand, in cryo-ET the 3D structure can be directly calculated without statistical methods. On the other hand, the same position is illuminated over and over again limiting the achievable resolution, as the sample gets destroyed over time. Additionally, due to the geometry of the sample holder it is not possible to acquire images for every rotation angle which leads to a so called missing wedge effect. In order to compensate for the additional beam damage and the missing wedge, a method called "subtomogram averaging" can be used. For this technique, those areas of the resulting tomogram

that contain the same sample are aligned to each other, and the correctly rotated maps are averaged [83]. Recently, with this strategy resolutions of about 3 Å of in-vivo samples could be reached [123].

1.3 High resolution transmission electron cryomicroscopy

Cryo-EM is becoming increasingly attractive for studies where the analysis of a native high-resolution 3D structure is required, such as drug discovery research. In the following, the functional principles of a transmission electron microscope (TEM) and the SPA workflow will be addressed.

1.3.1 Structure of the electron microscope

The transmission electron microscope consists of a column which can be roughly classified into five parts (Figure 1.2).

I. Electron gun
II. Condenser system
III. Objective system
IV. Projection system
V. Detector

The column itself is under vacuum to reduce the amount of possible non-specimen particles that could otherwise interact with the electrons [100].

I. Electron gun The electrons used for specimen illumination are extracted from the electron source and accelerated towards the specimen. Firstly, the donor material is exposed to an external energy source, such as heat or an electric field. Thus, the work function of the electron is reduced and electrons can transition from the material into the evacuated column.

Even though there are several electron sources known, only one category is relevant in the context of high-resolution cryo-EM: field emission guns. They can be categorized into Schottky field emitters and cold field emitters, which both use a tapered material with a strong external field applied to utilize the Schottky effect for electron extraction. Additionally, the field can be adjusted to such an extend that the electrons leave only from the tip of the material with a very narrow energy range. Therefore, the resulting electron beam has a high temporal and spatial coherence.

The cold field emitter uses only an electric field for electron extraction while the Schottky field emitter is actively heated, resulting in a reduced need for high field strengths. On the one hand, the coherence of the former is overall superior. On the other hand, the surface is prone to contamination resulting in an overall unstable emission characteristic and demands even higher vacuum. Therefore, the Schottky field emitter is currently the standard in the field due to its superior stability and lifespan, despite the coherence being slightly inferior. However, recent technological advances in field emission gun research led to a new generation of cold emission field emitters installed in the latest versions of available high-end microscopes [15, 66].

After the electrons leave the electron source, a Wehnelt cylinder is used to focus the beam and accelerate the electrons towards the condenser system [100].

II. Condenser system The main purpose of the condenser system is to guarantee a parallel electron beam for the illumination of the specimen. It consists of a few electromagnetic pole piece lenses [110], i.e., electromagnetic coils with a metal core and a hole in the center for the electrons to pass through, and a condenser aperture. While the lenses can be used to tilt, focus, and rotate the beam, the condensor aperture filters out the electrons which are far away from the beam axis. The electrons move with very high speed inside the column and, therefore, the electromagnetic field interaction is very brief, resulting in deflection angles in the range of milliradian [133]. Due to the geometry of the magnetic field, the electrons are deflected in a spiral shaped trajectory. Compared to optical lenses, electromagnetic lenses cannot be produced with such high precision, and the electromagnetic field can be easily disturbed by external sources [133].

III. Objective system The objective system is equipped with objective lenses to focus the beam, the specimen stage where the specimen holder is placed during data acquisition, an objective aperture to filter electrons with exceptionally high scattering angles, and a mechanism to insert the specimen holder. The latter is located between two objective lenses to allow for the alignment of the electron beam on the specimen.

Since lens aberrations degrade the signal, it is most important that the objective lens is exceptionally well manufactured, because the subsequent lenses of the projection system not only magnify the image, but also the introduced error [133].

IV. Projection system The projection system consists of several lenses to magnify the image and focus the beam on the detector. Magnifications ranging from 10^2 to 10^6 times can be achieved by chaining multiple lenses together [133].

V. Detector In order to visualize the electrons after the projection system, a detector and/or a fluorescent screen is installed at the bottom of the column. Prior to the ”Resolution Revolution”, charge-coupled device (CCD) based detectors were used which detected the signal with the help of a photodiode. Therefore, the incoming electrons needed to be translated into photons first for detection before being converting back into electrons, which has a negative influence on the detection quantum efficiency (DQE) [20]. For this reason, complementary metal-oxide-semiconductor (CMOS) based direct detecting devices (DDDs) are nowadays used for high-resolution cryo-EM, where the incoming electrons are directly detected. The development of DDDs is a major part of the ”Resolution Revolution” achievements and resulted in previously-unimaginable high DQEs and the possibility to collect movies instead of single images per acquisition position [67]. As a consequence, single electron events can be detected and the movies allow for the correction of beam induced motion [8]. Nowadays, frame rates of more than 1 500 frames/s, a large field of view of 4 092 pixel ×5 760 pixel, and super resolution approaches to increase the DQE for high frequencies are commonly used [61].

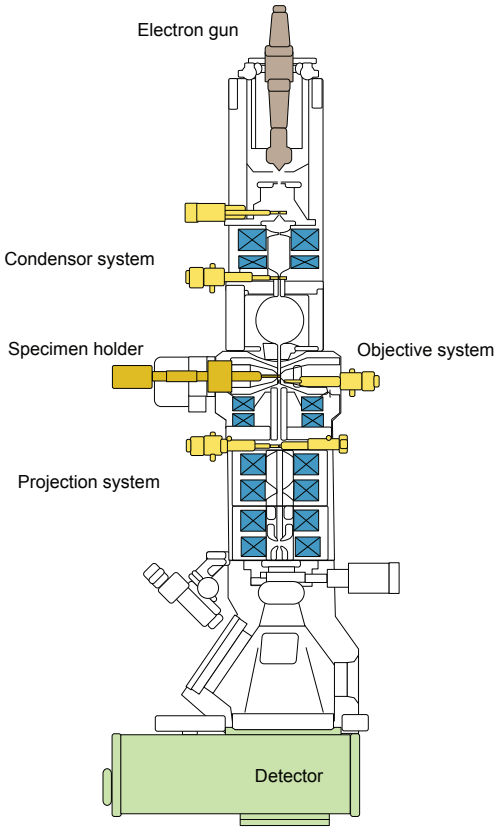

Figure 1.2: Schematic illustration of an TEM. Lenses are highlighted in blue, apertures in yellow, the specimen holder in dark yellow, the electron gun in light brown, and the detector in green. Figure adapted from [47].

1.3.2 Specimen preparation

Since the protein is already in solution and the native conformation should ideally be preserved, rapid freezing of the sample on a specimen holder, from now on referred to as ”grid”, is the fixation method of choice for cryo-EM [55, 84, 126]. The grid is typically a round and thin piece of metal, often copper or gold, with a radius of 1.525 mm [126]. While the outer ring is stable to avoid distortions the inner part is divided into empty squares arranged in a grid pattern.

On top of the squares is a thin support layer made of carbon, cellulose, or gold, which is specific to the experiment at hand and can *inter alia* be a continuous layer, a layer with holes arranged in a random pattern with a random size and shape, or the most commonly used holey layer, where the holes are arranged in a regular pattern with pre-defined shape and size. When the sample in solution is applied to the specimen holder, the surface tension of the solution is allowing the specimen to be located within the empty holes. Ideally, the resulting solution layer is thin enough to allow for a single layer of proteins, but the solution layer needs to be thick enough to avoid any deformations or distortions. The ice thickness can be controlled by removing parts of the liquid solution by blotting for a project dependent amount of time [126].

To avoid ice crystals during the freezing process, which would destroy the protein, the sample needs to be vitrified. Therefore, a method called ”plunge freezing” is commonly used where tweezers with the prepared specimen holder attached are plunged into liquid ethane. Compared to liquid nitrogen, liquid ethane is not subject to the Leidenfrost-effect, which describes the creation of an insulating layer of gas when a material comes in contact with the liquid [70]. Hence, the sample is frozen almost instantaneously keeping the specimen in its native state and leaving no time to form ice crystals [126].

1.3.3 Image formation

For image acquisition, the prepared specimen holder is inserted in the objective system of the microscope. Since the obtained image is the result of the interaction of the electrons with the specimen, it is important to understand the underlying principles of image formation for a correct interpretation.

Electron scattering The electrons of the beam and the electrons of the atomic shell of an atom can interact via elastic and inelastic scattering (Figure 1.3a). Elastically scattered electrons are deflected, but their energy is unchanged. Therefore, elastically scattered electrons carry structural information encoded in their deflection angle and phase. On the other hand, inelastically scattered electrons are also deflected, but energy is additionally transferred to the atoms they interact with. Not only does the inelastically scattered electron lose part of its energy and can no longer contribute to the structural information, but the transferred energy can also severely damage the structure of the protein [100]. The ratio of elastic-to-inelastic scattering events is about $1/3$ for biological specimens and the probability is proportional to $z^{\frac{4}{3}}$ and $z^{\frac{1}{3}}$ for elastic and inelastic scattering, respectively [104]. Additionally, the scattering angles of heavy atoms are in general larger than the scattering angles of light atoms due to the increased size and charge of the atom [100]. The resulting image in cryo-EM can therefore be considered as a projection of the specimen with the structural information being encoded in its image contrast (Figure 1.3b).

Image contrast The resulting contrast of the image is a superposition of amplitude and phase contrast and samples can be categorized into mainly amplitude or mainly phase objects depending on the prominent contrast present [100, 133]. Pure amplitude objects only change the amplitude of the incoming beam by changing the beams intensity. In contrast, pure phase objects only change the phase of the beam due to differences in the density of the specimen. Weak-phase objects, such as biological samples with small atomic numbers fixated in a thin ice layer, have a phase difference of about 90° between the scattered and the unscattered beam. Therefore, the amplitude of the resulting beam is almost identical to the amplitude of the incident beam leading to a homogeneous grey image. Nonetheless, it is possible to introduce aberrations or use a phase plate to introduce an additional phase shift of ideally +90° or -90° to the scattered beam and transfer the phase shift variations into more pronounced amplitude variations. The difference between the signal present directly after the specimen and the signal detected at the detector is described by the contrast transfer function (CTF) (Section 1.3.4, Figure 1.3c).

By defocusing the specimen, the traveled distance of the electrons is artificially modified leading to a phase shift and enhanced image contrast. However, this leads to a delocalisation of the

structural information in the image as further discussed in section 1.3.4 [100, 133]. The most commonly used phase plate, the Volta phase plate [16], consists of a piece of thin carbon to be inserted in the back focal plane of the microscope. When the electrons are passing through the carbon the charging effects introduce a phase shift to the scattered electrons. However, the amount of introduced phase shift is dependent on the number of electrons that have already contributed to the charging effect, requiring the position on the phase plate to be changed every few hours. The introduced phase shift $\Delta\phi$ can be described by

$$\Delta\phi(\vec{k}, \Delta z) = \frac{\pi}{2}\left(\lambda^3 C_s |\vec{k}|^4 + 2\lambda\Delta z|\vec{k}|^2\right) + \text{phase shift}\,, \tag{1.1}$$

with the wavelength of the electrons λ, the spatial frequency $\vec{k}$, the spherical aberration C_s, the phase shift of the phase plate, and the difference in defocus Δz [130, 31].

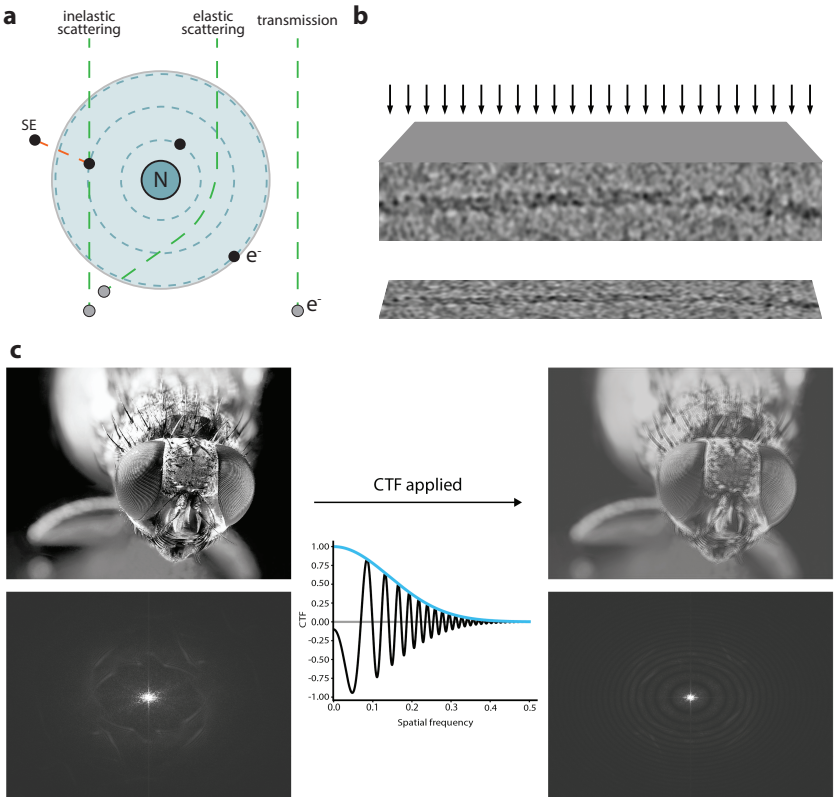

Figure 1.3: a Schematic of the scattering events important for the image formation in cryo-EM between the electrons of the beam (gray with green path) and an atom of the specimen (electrons in black with blue path and the nucleus N). Inelastically scattered electrons transmit parts of their energy to the electron of the shell leading to an ejected secondary electron (SE). **b** The incoming electron beam interacts with the specimen and leads to a two dimensional (2D) projection image with encoded structural information. **c** Influence of the CTF on an image. (**Left**) Photograph of a fly (top) and the corresponding power spectrum (bottom). Applying a CTF function with a defocus of 0.5 µm and a B-factor of 100 Å^2 results in a frequency dependent contrast inversion and a dampening of high-resolution details (top). The information at the zero-crossings of the CTF is lost, as visualized by the Thon rings [127] in the power spectrum (bottom). Figure and caption adapted from [90].

1.3.4 Resolution limitations

The ultimate aim of a high-resolution structure determination method is an atomic model representing atomic resolution. However, resolution limiting factors, introduced by the imaging system, or the specimen itself prevent this goal. Therefore, it is important to get an understanding for the underlying problems and how they can be accounted for.

Abbe diffraction limit Ernst Abbe figured out that the maximum achievable resolution $d_{\text{diffraction limit}}$ of a microscope is dependent on the imaging wavelength and that in vacuum it is

defined as

$$d_{\text{diffraction limit}} = \frac{\lambda}{2 \cdot \sin(\theta)}\,, \tag{1.2}$$

with the diffraction limit d, the wavelength λ, and the angular aperture θ of the aperture [19]. For electrons accelerated with common voltages U_{B} of 200 kV and 300 kV, the limit $d_{\text{diffraction limit}}$ is about 0.025 pm and 0.019 pm, respectively. Hence, with the assumption of a small angular aperture of $\theta \geq 1°$, the achievable resolution can be calculated to be about $d_{\text{diffraction limit}} \leq 0.75$ Å and is therefore smaller than the desired resolution and not a limiting factor in practical applications [133].

Sampling theorem The Nyquist-Shannon sampling theorem describes the resolution limitation introduced by the detector itself. A signal can only be lossless recovered when the condition

$$f_{\text{sampling}} > 2 \cdot f_{\text{signal}} \tag{1.3}$$

with the sampling frequency f_{sampling} and the signal frequency f_{signal} is met [82]. Therefore, the maximum lossless recoverable signal is $f_{\text{nyquist}} = {}^{f_{\text{sampling}}}/_{2}$, which is also known as the Nyquist frequency.

For an imaging system using a DDD, which is composed of multiple discrete pixel, every electron event inside one pixel is assigned to the pixels central coordinates. Hence, the sampling frequency is $f_{\text{sampling}} = {}^{1}/_{\text{pixel size}}$, where pixel size is defined by pixel size $= {}^{\text{physical pixel size}}/_{\text{magnification}}$ [100]. Thus, the maximum achievable resolution $d_{\text{sampling theorem}}$ at a given magnification is

$$d_{\text{sampling theorem}} = \frac{1}{f_{\text{nyquist}}} = \frac{2}{f_{\text{sampling}}} = 2 \cdot \text{pixel size}\,. \tag{1.4}$$

Typically, pixel size values between 0.2 Å and 1.5 Å are used for high-resolution data sets, limiting the resolution $d_{\text{sampling theorem}}$ to 0.4 Å and 3.0 Å, respectively. If a data set is able to reach the maximum resolution $d_{\text{sampling theorem}}$, and the sampling theorem is therefore the limiting factor, it is possible to collect a new data set with a smaller pixel size, i.e., higher magnification.

Specimen motion The sample in high-resolution cryo-EM is typically embedded in a vitreous ice layer on a grid. In terms of movement, motion is generally distinguished between two separate types: stage-induced motion and beam-induced motion [8].

The stage-induced motion is introduced by the specimen stage itself. To collect a data set, the area of interest on the specimen stage needs to be physically moved into the electron beam with atomic precision in order to get imaged. However, the stage is still moving after the desired coordinates are reached, due to relaxation effects and instabilities. Therefore, the stage-induced motion has the character of linear motion.

The beam-induced motion is introduced by the electron beam itself. When the specimen is illuminated by the electron beam the transferred energy excite the atoms of the specimen and, hence, leads to motion. While ice melting effects show characteristics of linear motion, the motion of excited atoms of the sample can be approximated with Brownian motion.

Even though the origin of specimen movement can be due to a multitude of reasons, the effect on the resulting image is identical: image blurring and hence lost high-resolution information. In order to compensate for the motion effects, modern DDD collect movies instead of single images

with very short exposure times in the millisecond range. After acquisition, an alignment, in other fields also known as image registration, of the frames to each other can compensate and mitigate the negative influence in the resulting summed image [139, 45].

Beam-induced damage Beam-induced motion is not the only negative side effect induced by the electron beam. Biological samples are highly sensitive to radiation damage, and therefore the structure of the sample suffers after illuminations. Hence, to keep the structural information as intact as possible, electron doses of about 40 electrons/Å^2 to 60 electrons/Å^2 are used for each image. On the one hand, this preserves more information about the structure, on the other hand it is not low enough to compensate for the damage in the high-resolution range. However, reducing the overall electron dose used even more would reduce the already low SNR even more, rendering the collected images useless for later alignment procedures. As a consequence, different dose weighting schemes have emerged to reduce the influence of damaged structures on the high-frequency information of the summed image by introducing frequency dependent weighting factors per frame [139, 45, 8, 140].

Contrast transfer function When the electron beam passes through the imaging system, different artifacts are introduced. The ones with the most impact are *inter alia* spherical aberration (C_s), chromatic aberration (C_c), coherence of the electron beam, high-tension fluctuations, and a misalignment of the optical system itself. Additionally, altering the defocus to increase the phase shift, as described in 1.3.3, of the unscattered electron beam also plays a major role.

All those artifacts are described by the CTF, which is defined as

$$\mathrm{CTF}\,(\vec{k}) = \frac{\mathrm{imagecontrast}\,(\vec{k}, \Delta Z)}{\mathrm{objectcontrast}\,(\vec{k}, \Delta Z)}\,,$$

with $\vec{k}$ being the spatial frequency within the image [100]. As a result, the CTF can be approximated to

$$\mathrm{CTF}\,(\vec{k}, \Delta z) = e^{-|\vec{k}|^2 \frac{B}{4}} \cdot \left(\sqrt{1 - A^2}\sin \Delta\phi(\vec{k}, \Delta z) - \sqrt{A}\cos \Delta\phi(\vec{k}, \Delta z)\right)\,, \tag{1.5}$$

with $\vec{k}$ being the spatial frequency, Δz the additional defocus value, B describing all negative effects such as imperfections of the optical system and the coherence of the beam, A the amplitude contrast of the specimen, and the additional phase shift $\Delta\phi$ described in equation 1.1.

The CTF is, therefore, describing how much information is present at a certain frequency in the resulting image. Problems arise when the sign of the CTF is changing the first time, leading to an inversion of the contrast and making the information past this point not directly interpretable (Figure 1.3c). Hence, without further processing the resolution is limited to the frequency of the first zero crossing.

CTF estimation programs have been developed to estimate the parameters of the CTF and allow to correct for it. However, even if the contrast inversion can be accounted for by knowing the shape of the CTF, information at frequencies where the CTF is zero cannot be recovered. Therefore, the SPA approach tries to fill the gaps with information contributed by projections of the same protein that were acquired at a slightly different defocus value.

The influence of the C_s can be reduced by inserting a C_s-corrector in the condenser system of the microscope [133].

The protein itself The most important factor for the resolution limitation is the protein itself, because SPA assumes that the protein projections are exact copies of each other [86]. However, there are several aspects like protein flexibility, air-water interface interactions, and aggregation/degradation of the sample that break the assumption.

Therefore, digital purification is necessary to reach high resolution by either removing the affected data from the data set or by creating homogeneous subsets.

1.3.5 Data processing

In order to tackle the challenges described in the previous sections, several open-source and proprietary software packages like *SPHIRE* [79], *EMAN2* [121], *RELION* [113], *cisTEM* [12], *cryoSPARC* [96], *SPIDER* [38], *IMAGIC* [53], and *SPARX* [57] have been developed. Those cover most of the SPA processing steps needed to obtain a 3D reconstruction of the protein at hand. However, packages with a maximum-likelihood based 3D refinement approach, e.g., *SPHIRE* [79], *RELION* [113], *cisTEM* [12], and *cryoSPARC* [96], perform significantly better for high-resolution cryo-EM.

Additionally, tools that fill niches missing in previously mentioned pipelines like *crYOLO* [131], *Cinderella* [11], *MotionCor2* [139], *gCTF* [138], *WARP* [123], *LAFTER* [98], *ROME* [134], and many more have been contributed from all over the world.

During the course of this doctoral thesis, the software package *SPHIRE* [79] has been actively developed and maintained and the outcome has been highly dependent on the tools included. Therefore, an overview of the *SPHIRE* [79] pipeline and the included tools and methods is provided in the following sections.

Theoretical background

A large variety of methods used in the SPA workflow have been developed over the past decades [103]. Within the *SPHIRE* [79] pipeline, the core functionality can be reduced to five main methods: Equal size K-means, maximum likelihood, projection matching, weighted back projection, and convolutional neural networks.

Euclidean distance In euclidean space the euclidean distance is the length of a line between two given points $\vec{A}$ and $\vec{B}$. Therefore, it is also known as the L^2 norm of the difference vector of those two points which is calculated by the Pythagorean theorem

$$||\vec{B} - \vec{A}|| = \sqrt{(B_1 - A_1)^2 + ... + (B_n - A_n)^2}\,.$$

Maximum a posteriori probability estimation Nowadays, several high-resolution 2D and 3D processing strategies utilize a maximum a posteriori based strategy to identify the projection or alignment parameters of the particles [113, 96, 134, 79]. This method allows the approximation of a best estimate of the unknown parameters based on empirical data, i.e., the particles collected at the microscope. For this purpose, the likelihood for every combination of parameters and particles is calculated and the set of parameters yielding the maximum likelihood value is assigned to the particles

$$P(\Theta|X,Y) \approx P(X|\Theta,Y) \cdot P(\Theta|Y)\,,$$

with X being the observed data, Y being the prior information, Θ being the best parameter set considering X and Y, $P(\Theta|X,Y)$ being the maximum a posteriori estimate, the likelihood $P(X|\Theta,Y)$ quantifying the probability of observing the data given the model, and the prior $P(\Theta|Y)$ expressing how likely the model is given the prior information [113]. It should be noted that these parameter assignments highly depend on the provided starting model and incorrect models are prone to lead to incorrect results.

Equal size K-means Conceptionally, the K-means algorithm is a very easy to understand, yet powerful clustering approach and is mainly used for the 2D classification step in the cryo-EM pipeline [73]. The advantages are a guaranteed convergence and a simple implementation. However, that the outcome is highly dependent on the initial cluster assignments and the clusters are assumed to be clearly separated and of a circular shape are disadvantages of the algorithm.

In its most simple form it can be described by:

Algorithm 1 Description of the K-means algorithm

Require: Data is split randomly into K clusters
Require: E_d be the minimum squared Euclidean distance
  while Cluster K assignments change **do**
    Average cluster K assignments to get cluster representatives
    Calculate the E_d between every data point and every cluster representative
    Re-assign cluster members based on the shortest E_d
  end while

Since the input data set in cryo-EM is dominated by noise, those disadvantages have an even greater impact on the outcome. Therefore, the *SPARX* [57] package developed an alternative approach called Equal size K-means [136] which modifies the original workflow:

Algorithm 2 Description of the equal size K-means algorithm

Require: E_d be the least squared Euclidean distance
Require: N be the maximum number of members per cluster
  while Particles available or no additional cluster found **do**
Require: Data set is split randomly into K clusters
    while Validated cluster K is found **do**
      for X iterations **do**
        Average cluster K assignments to get cluster representatives
        Calculate the E_d between every data point and every cluster representative
        Re-assign cluster members based on the shortest E_d considering N
      end for
      Validate the heterogeneity of the cluster
      Remove the members of heterogeneous clusters from the data set
    end while
  end while

Originally, the K-means algorithm is clustering the data set by area, but the Equal size K-

means approach is additionally limiting the total number of particles N per cluster. This has the advantage that potentially large clusters are split into multiple smaller clusters allowing for a higher assignment accuracy and overall more robust clustering results in noisy environments.

Fourier shell correlation Unlike other structure determination methods, the resolution of a structure determined by cryo-EM cannot be directly determined and the resolution is typically estimated with the help of the Fourier shell correlation (FSC) [51]. However, the FSC is not a measure of resolution, but a measure of similarity between two 3D volumes, where the cross correlation is calculated between each shell of those volumes in Fourier space. Afterwards is the resolution of the volumes defined as the value of the first shell that has a correlation value smaller than a certain threshold.

Nonetheless, the exact value of the threshold is under constant debate and several values have been proposed [52]. The most commonly used ones are the $FSC_{0.143}$ and the $FSC_{0.5}$ criterion, where the threshold is defined as 0.143 and 0.5, respectively.

Projection slice theorem The fundamental basis that high-resolution cryo-EM is build upon is the projection slice theorem that enables the 3D reconstruction of an object based on 2D projection images of the same object [7] (Figure 1.4). To do so, some requirements need to be fulfilled, e.g., a parallel illumination and a specimen that is thin enough to transmit the beam: the assumptions of the SPA process.

Assuming an image I_o, the theorem states that the 2D Fourier transform of an image I_p, which is the projection along a known projection direction P of the image I_o, is equivalent to the slice of the 3D Fourier transform of I_o normal to the same projection direction P. Therefore, in the core of the SPA pipeline, the 3D refinement, it is the aim to identify the projection directions P of the individual projection images collected at the microscope

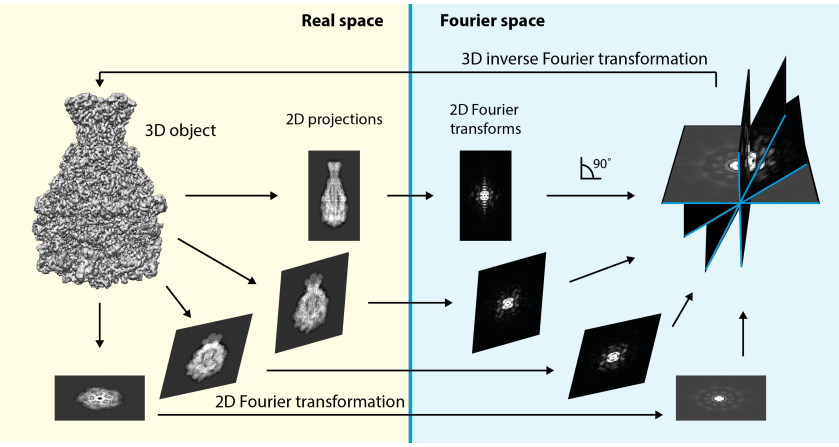

Figure 1.4: Illustration of the projection slice theorem using the 3D density of a bacterial Tc Toxin (EMDB 10034, [71]). The initial 3D object can be reconstructed using 2D projections of the object with known projection angles using a 3D Fourier transformation, because the Fourier transform of each 2D represent a slice in the 3D Fourier space. Figure and caption adapted from [90].

Convolutional neural networks Neural network based applications have been sprouting from the ground almost everywhere [18]. Therefore, it is not surprising that it is also applicable to problems in cryo-EM and is already well-established in the processing step of particle picking [5, 131, 122].

For image recognition and classification, convolutional neural networks (CNNs) have proven to be very effective [114]. Those consist of a feature extractor possibly followed by a fully connected neural network.

In general, the feature extractor part consists of two types of layers: convolutional layers and pooling layers. Convolution layers in a CNN learn kernels instead of weights, and instead of multiplying the input with the associated weight, the output of the previous node is convoluted with the kernel followed by a pixel-wise application of the activation function to introduce non-linearity. The pooling layer reduces the dimensions, i.e., the parameter space of the output layer. Most common are the max pooling, which summarizes the most activated presence of a feature, and average pooling, which summarizes the average presence of a feature.

The fully connected neural network in its basic form consists of an input layer and an output layer and every layer can additionally contain multiple nodes [114]. Between those two layers, there can be additional 0 to N so called hidden layers. While the input layer serves as the entry point to the network architecture and receives the input data set, the output layers task is to provide the result of the networks calculations. The optional hidden layers of the network modify the data in a way that is interpretable by the output layer. The number of hidden layers N should be proportional to the complexity of the problem and the amount of training data at hand. While a network consisting of just an input and an output layer can only learn linear functions to describe the data set, every additional hidden layer allows for more complex relationships. However, the demand for more training data increases with an increasing number of hidden layers to prevent overfitting.

Each layer contains a certain number of nodes which are made of several inputs, associated weights, an activation function, and one output. The inputs are the outputs of the nodes from the previous layer connected to the node and the output is determined by the activation function. Each input is multiplied with a weight associated with it and which is the target of the training procedure. Afterwards, all the inputs are summed, and the sum is provided to the activation function, which should introduce non-linearity to represent the character of real world data. During the training of the network with sophisticated methods like back-propagation the weights of the network are optimized so that the output of the output layer matches the labels of the input data set as accurate as possible.

With these building blocks in place, a CNN is able to learn visual patterns from the input data set and can interpret its content

Processing pipeline

The ultimate aim of the SPA pipeline is the high-resolution 3D reconstruction of the sample. Therefore, the workflow requires all the steps that benefit a successful 3D refinement which does the assignment of projection parameters to the projection images obtained from the micrographs collected at the TEM, i.e., rotation inside the thin ice layer and centering relative to the reference structure.

A majority of the results produced in this thesis are based on the SPA software package *SPHIRE* [79], which has been in parts developed in the Raunser lab. To classify the results in the context, the basic SPA workflow is described in the following sections (Figure 1.5a).

Data acquisition First, the specimen holder is inserted into the TEM and typically followed by the microscope alignment procedure. The data collection itself used to be a tedious manual process that yielded about 30 micrographs/h to 100 micrographs/h. Nowadays, automated data collection software packages like *EPU* [30], *Serial EM* [75], and *Leginon* [120] perform this procedure in a fully automated manner. However, the yield is highly dependent on the acquisition strategy, i.e., how many images are acquired by shifting the beam without moving the stage, magnification, and the camera used. In combination with hardware advances like aberration free image shift (AFIS) or fringe-free illumination (FFI) the yield varies between 50 micrographs/h to 600 micrographs/h [9, 1, 32].

As described in section 1.3.4, it is crucial to collect the data set at different defocus values to account for the effects of the CTF. Therefore, the defocus values needs to be as large as necessary to have enough contrast, but on the other hand as small as possible to minimize artifacts. Additionally, biological specimens suffer from beam damage requiring a total useable electron dose of 20 $e^-/Å^2$ to 60 $e^-/Å^2$ [45].

At every acquisition position a movie with a suitable total electron dose is collected, but the number of frames per movie is dependent on the strategy and the stability of the TEM. Assuming a fixed total dose, increasing the number of frames decreases the electron dose per frame, hence reducing the contrast due to a smaller SNR. In contrast, decreasing the number of frames increases the illumination time per frame, hence increasing the motion blur, while increasing the SNR. Therefore, the number of frames is a trade-off between the information loss due to a low SNR and the information loss due to motion blur.

The resulting movies typically contain between 40 frames to 300 frames with a dimension between 4 000 pixel ×4 000 pixel and 12 000 pixel ×8 000 pixel per frame [61]. In order to safe disk space, lossless compressions algorithms like *TIFF LZW* are used, reducing the file size by up to a tenth of the original size.

Motion correction The reason for collecting movies consisting of frames with a small illumination time instead of single image with a large illumination time is twofold. Firstly, the fixed specimen is moving due to the induced energy from the electrons. Secondly, the specimen in the sample is suffering from beam damage, rendering the information from later times of exposure useless for high-resolution reconstruction. To account for those issues, the first step after data acquisition is typically to perform motion correction and dose weighting [45, 139, 140].

Motion correction algorithms perform an image alignment of the single frames with respect to each other. The drift can either be calculated on the whole micrograph or in patches. To increase the robustness of the procedure, a software-specific polynomial fitted to the estimated shifts for smoothing and the resulting shifts are applied to the individual frames.

To account for the beam-induced damage, the frames are weighted according to their cumulative dose. Since the damage is least pronounced at the beginning of the exposure, i.e., high-resolution information is most intact, the high-resolution information of the first frames is up-weighted while it is down-weighted for later frames.

CTF estimation The individual frames of the movie or the aligned average can be used to estimate the influence of the CTF at the acquisition position [138, 106, 89]. To estimate the parameters of equation 1.5, a fit against the oscillation of the values of the Fourier transform (FT) of the image is performed. Therefore, to simplify the procedure, the values for the amplitude contrast A and the spherical aberration C_s are kept constant so that only the defocus Δz and a possible phase shift $\Delta\phi$ is estimated. Additionally, the astigmatism can be estimated by determining the parameters of the CTF for one dimensional (1D) central sections at different angles.

Particle picking The later stages of the processing pipeline involve thousands to millions of small regions extracted from the aligned micrographs. However, the positions of interest need to be determined before extraction. Multiple options are available, like manual selection, template matching, and CNN based approaches [131, 122, 5, 43, 121].

While manual particle selection is a tedious procedure involving labeling millions of locations in noise dominated images, template matching has the advantage of being a semi-automated technique. The provided reference image is compared with every position of the micrograph and those regions with the highest local maximum are considered locations of interest. However, drawbacks of this method are the inaccuracy, requiring sophisticated cleaning procedures, and the possibly introduced model bias in case high-resolution images of the particle of interest are provided as reference images [54].

Recently, CNN based particle pickers like *crYOLO* [131], *Topaz* [5], and *WARP picker* [123] have revolutionized particle picking. With provided pre-trained models based on large training data sets, previously unknown data sets can be picked with human level accuracy and with a speed of multiple micrographs per second. Additionally, by picking a few micrographs manually and training a model based on the provided data set allows for tailor-made models for specific samples without the bias of template based procedures.

Particle extraction After the successful identification of potential locations of interest, they are cropped out of the micrograph with a pre-defined box size to yield the so called ”particles”. On the one hand, particle extraction has the advantage that the dimensions of the processed images is reduced and subsequent processing steps are less computationally expensive. On the other hand, neighboring particles might be included in the boxed out area influencing the alignment calculations of subsequent processing steps. Additionally, the introduction of aberrations like defocusing leads to a delocalization of structural information in real-space image, which is described by the CTF. Hence, the choice of the box size is a trade-off between computational speed, including necessary delocalized information, and the influence of neighboring particles. Additionally, the centering of the particles, especially for elongated samples, turns out to be challenging, and a small box size might remove parts of the sample's density.

2D classification Due to the exceptionally low SNR of the micrographs, even the most sophisticated particle picking procedures are prone to false-positive picks, i.e., structures falsely identified as the intact sample of interest. This can either mean that contamination has been picked, but also different conformations or aggregates of the sample. Therefore, 2D classification is used to clean the data set from the falsely picked particles. The most commonly used classification algorithms used in the field are either modifications of a maximum likelihood based approach or K-means.

Maximum likelihood based implementations are fast and yield high-resolution classes. However, the drawback is that classes with such a high resolution tend to attract particles of even different conformations. This method is able to achieve high-resolution classes, however, at the cost of homogeneity as they typically consist of multiple thousand members. K-means based algorithms like *ISAC* [136] try to perform better in terms of homogeneous classes. However, this requires more sophisticated sorting approaches, convergence criteria, and stability testing which makes it more computationally demanding.

Additionally, deep-learning based classifiers like *Cinderella* [11] and *2DAssess* [72] have been developed, which are able to separate classes into "kept" and "discarded" based on the model provided by the user.

Initial 3D reference Before a high-resolution 3D reconstruction of the sample can be calculated, the refinement algorithm needs an initial reference as a starting point. Even though noise filled spheres or crystal structures of the sample are in principle possible to use, the refinement algorithms profit from an initial reference that resembles a low-resolution version of the actual target structure. Therefore, a multitude of initial 3D reconstruction algorithms are available that calculate a low-resolution version from the data set itself [37].

One method is based on random conical tilt [36, 97], where two images of the same acquisition position are taken, but the stage itself is tilted by a known amount between the acquisitions so that the relative angle between the two projection images is known. Combining the information of those projection pairs, the orientations of all molecules can be readily determined and the result can be verified. Additionally, the handedness of the 3D structure can be correctly determined due to the known angle between the pairs. The drawbacks of this method are the requirement to collect pairs of images, since it is difficult to automate, and additional challenges regarding CTF correction and stage stability due to the tilt.

The second common method that also results in a 3D reconstruction with the correct handedness is cryo-ET. However, the resulting reconstruction has the limitations as described in section 1.2.3 regarding SNR and the missing wedge effects. Additionally, the collection of a tomographic reconstruction requires different alignment parameters of the microscope than a SPA project and is therefore rarely used for the initial reference creation.

Another method is the common lines approach [14]. As described in section 1.3.5, when all the particles are projections of an exact copy of the same protein at different orientations the FTs of the projections are central sections of the corresponding 3D structure. Therefore, all slices share a common line in 3D Fourier space, and the common-lines algorithm tries to identify those common lines and assign the correct orientations relative to each other. Since the homogeneity of the sample cannot be guaranteed, and the low SNR of the sample also influences the SNR of the FT, the success of this method is very dependent on the quality of the input.

Finally, the most commonly used methods are based on stochastic optimization procedures using the raw extracted particles or previously obtained 2D class averages [26, 79, 96]. They require an initial reference of some sort, e.g., a heuristic or random initialization to avoid model bias, and use optimization methods like stochastic gradient descent and stochastic hill climbing to calculate an initial reference. This has the advantage that no additional data acquisition strategies need to be utilized, and the initial reference is based on the data set.

If an initial model is created from the data set itself without prior knowledge about its structure,

the mechanism is referred to as *ab initio* 3D reconstruction. However, it should be noted that mathematically speaking the outcome of a 3D *ab initio* reconstruction often can have two valid outcomes with opposite handedness. This is due to the fact that projection information does not have depth information, and therefore they are equivalent. To identify the correct result, either a high-resolution 3D structure of the protein is necessary, as it is possible to identify the correct handedness by the turn of an alpha helix, prior knowledge about the structure with, e.g., a homologous 3D structure for comparison, or use of random conical tilt/cryo-ET.

3D refinement Undoubtedly, the heart of every structure determination project is the 3D refinement, which tries to assign the correct projection parameters to the extracted particles to obtain a high-resolution 3D structure. Nowadays, every software package has its own implementation that works in a fully automated manner and is taking care of the adjustment of parameters like filter frequency based on the current resolution, information limit of the current images to prevent overfitting, and the angular search strategy [113, 79, 119, 56].

One major prerequisite is an initial 3D model that ideally resembles a low-resolution version of the expected final result and can be obtained by different *ab initio* 3D reconstruction methods. A multitude of automated iterative 3D refinement strategies utilizing different heuristics have emerged from the field which, at their core, consist of an alternation between projection parameter assignment and 3D reconstruction:

Algorithm 3 3D refinement

Require: S be a measure of similarity, e.g., likelihood, cross correlation coefficient, or Euclidean distance.
Require: N be the number of real data objects contributing to the 3D reconstruction. This value is based on the heuristic used.
Require: Assign the provided initial model as the reference model
  while Projection parameter assignment changes **do**
    Generate reference projection images from the reference model
    Calculate S between the generated reference projections and the input data set
    Take the N highest similarities to reconstruct a 3D volume
    Calculate the resolution of the 3D volume and perform adjustments like masking
    Assign the reconstructed 3D volume as the reference model
  end while

Due to the low SNR of the input data set, the refinement procedure is prone to overfitting and, therefore, several safety measures have been developed. Firstly, most programs nowadays split the particles randomly into two groups and each group is refined almost independently from each other, which is commonly referred to as a "gold standard" [46] refinement. Furthermore, the information present in the particles is artificially limited based on the achieved resolution by the 3D reconstruction of the previous refinement iteration either by low-pass filtering or downsampling the particles and reference projections prior to comparison [56].

Additionally, the angular sampling that of the reference projections created is gradually adjusted based on the performance of the refinement, i.e., the accuracy of projection parameter assignments over multiple iterations. In the early stages of the refinement, the particles are compared with every

created reference projection (global refinement mode), however, in later stages of the refinement the comparison happens only between the reference projections that are in a close vicinity of the previously assigned projection parameter (local refinement mode). Otherwise, the procedure would not be computationally feasible. Since the number of comparisons approximately grows exponentially with every increase of the angular sampling, often the mode of the refinement changes when the angular sampling reaches an angular distance of 1.875° [79, 113].

The achieved resolution after each refinement iteration is assessed with the help of the FSC as described in section 1.3.5. The volume used in the next iteration step is afterwards filtered accordingly to prevent overfitting. Additionally, often a real-space 3D mask, which encapsulates the structure, is applied to reduce background and improve the alignment accuracy in the subsequent iteration.

The final 3D reconstruction is always calculated with full information present and no masking is applied.

Resolution estimation and sharpening After the 3D refinement is finished, the resolution is again assessed [56, 79, 113]. Furthermore, the SNR can be increased by applying a mask to the volumes and focusing the resolution estimation on the region that contains the sample.

Afterwards, the output is masked, low-pass filtered and sharpened to improve its usability for structural analysis and model building. As the low-pass filter threshold, the estimated resolution is typically used to prevent over-interpretation and increase the SNR. Sharpening the volume corrects for the loss of contrast in the high-frequency range introduced by, e.g., radiation damage and errors in the reconstruction procedure. While done correctly, sharpening can help to visualize high-resolution features for model building, but the also enhanced noise can make the volume look fragmented and harder to interpret. Therefore, algorithms that analyze the structure factor of the volume are used to identify the optimal threshold [109].

Often samples are not rigid bodies, but consist of some rigid and some flexible parts, resulting in a resolution gradient within the 3D reconstruction. Therefore, it is possible to filter each region of the volumes by a local resolution value instead of the overall resolution. Such a local filter has the advantage that worse resolved regions show a higher SNR, compared to a filter according to the overall resolution, while better resolved regions allow for a more detailed analysis of the structure.

Additional processing steps To further improve the quality of the structure and account for flexibility within the sample, additional processing steps can be executed. The most common ones are *inter alia* 3D sorting, heterogeneity analysis, particle polishing, beam-tilt estimation, CTF refinement, signal subtraction, and multi-body-refinement [87, 94, 56, 112, 80, 141].

3D sorting and heterogeneity analysis try to identify groups of particles originating from identical copies of the sample. Afterwards, those homogenous subsets run separately through the SPA pipeline resulting in a 3D volume per conformation. However, the resolution and SNR of the resulting reconstruction is not necessarily increased, because the number of particles per reconstruction is decreased. On the other hand, particle polishing, beam-tilt estimation, and CTF refinement re-assess the parameters initially estimated for dose weighting and CTF correction on a per-particle basis. This results in a higher SNR for higher frequencies, hence an improved resolution based on the new parameters. Finally, signal subtraction and multi-body-refinement computationally remove the rigid parts from the particles itself so that they can be

refined individually to reduce the influence on another flexible part of the sample on the alignment.

The introduction of the additional processing steps allows for the design of a tailored processing strategy for every project. However, the initial processing steps up to the first 3D reconstruction are often so identical to each other that first approaches for automation have been taken.

Helical processing A helical filament consists of a single subunit that is repeated with its specific helical symmetry parameters $\Delta\phi$ and Δz. The parameter Δz describes the translational difference along the helical axis between two adjacent subunits, while $\Delta\phi$ is the difference in rotation between those. When it comes to helical processing, the general SPA workflow remains identical, but certain steps are slightly modified and require different assumptions due to the rod-like character of the sample [39]. Now, unlike in SPA, the particle extraction process works in a way that each extracted particle image has one unique subunit in its center plus several additional subunits, which themselves are the central subunits of the neighboring particle images. Therefore, the informational overlap between individual particles can be up to 95 %.

Additionally, the projections originating from filaments only show so-called side-views of the filament due to physical limitations such as their elongated shape, which often extends over several holes or even grid squares, with an ice thickness of a few tens of nano meter. This has the effect that the filament cannot tilt appreciable within the ice layer. However, the embedded filaments show a repeating pattern that is small compared to the field of view. Therefore, all projections around the equator are present in a single micrograph allowing for high-resolution structure determination.

The most common algorithms for the 3D reconstruction of helical samples were for a long time Fourier-Bessel methods [64] and then single-particle based methods like Iterative Helical Realspace Reconstruction (IHRSR) [23], while nowadays Bayesian based single-particle methods built on top of IHRSR are heavily used [27, 50]. Fourier-Bessel methods take advantage of the fact that the FT of a helical sample is a composition of different Bessel functions, which can be identified by the pattern of diffraction maxima, the so-called layer lines, in the FT of the filament. Based on the Bessel function information in combination with the knowledge about the symmetry parameters, the 3D structure can be calculated. However, in case of a low SNR, protein flexibility, or helical symmetry parameters diverging from the assumed symmetry, Fourier-Bessel methods are very limited in their application [24].

Single-particle based methods try to overcome the limitations of Fourier-Bessel like Bessel overlap for certain symmetries, the need to computationally straighten flexible filaments, weak Bessel layers caused by weakly diffracting filaments based algorithms by treating the extracted particles independently from each other during 3D refinement, as if they were part of a SPA project. Additionally, the amount of particles within a data set is much larger in SPA based approaches than Fourier-Bessel based approaches allowing to account for variability within the structure itself. Most helical refinement programs like *Helical RELION* [50], *SPIDER* [38], *SPARX* [57], *FREALIGN* [119], *FREALIX* [107], *SPRING* [17], and *cryoSPARC* [96] are at their core based on the IHRSR algorithm which decouples the 3D reconstruction from the helical symmetry parameters:

However, calculating the helical symmetry based on potentially low-resolution 3D reconstructions and applying the best match to the volume used as a reference for the subsequent iteration can lead to model bias and wrongly estimated structures and symmetry parameters. Therefore, while useful for many data sets, the higher the flexibility of the filament at hand, the higher the risk of model bias [91, 25].

Algorithm 4 Iterative helical real space reconstruction

Require: P be the assigned pose (Projection angles Φ, Θ, and Ψ and shifts sx, sy) of a particle.
Require: Sym be the helical symmetry parameters $\Delta\phi$ and Δz.
Require: An initial volume serving as the reference volume V for the first iteration.
Require: CCC be the cross-correlation coefficient calculated between a reference projection and a particle.
while The user defined number of iterations is not reached **do**
Calculate 2D projections from the reference volume V along the azimuth.
Perform a multi-reference alignment with the particles from the data set.
Assign P of the reference projections that led to the highest CCC for each particle.
Calculate a 3D reconstruction based on the assignments to each particle.
Apply different combinations of Sym to the reconstructed volume and determine the best fit by least squares fit with the original volume.
Use the best fitting symmetrized volume as reference volume V for the next iteration.
end while

Recent modifications of existing SPA workflows established in *RELION* [113] in the Raunser lab showed that it is possible to achieve high resolution for flexible helical filaments like actin without imposing any helical symmetry and hence reducing the risk of model bias [78].

1.3.6 Automated on-the-fly data processing

Automation is a tool to make processing tasks more standardized, efficient, and accessible for the user, and its true potential can be utilized in combination with on-the-fly processing, i.e., processing the data live during data acquisition. Nowadays, hundreds of images per hour are collected at the microscope resulting in thousands of images per data set within a few days. However, possible problems with the microscope alignment, data acquisition strategy, or sample quality cannot be directly deduced from the visual appearance of the raw data itself. Therefore, possible problems might show up days, weeks, or even months after the actual data collection, rendering the whole data set useless. Since time for data acquisition is typically rare and expensive, collecting the data set under the most optimal conditions is of utmost importance.

To assist with the evaluation of the data and to rule out the most obvious problems with the microscope alignment, the incoming data can be immediately processed. The processing steps can be separated into the three categories basic pre-processing, advanced pre-processing and 3D processing. Basic pre-processing includes motion correction, CTF estimation, particle picking, and particle extraction and is executed on a per micrograph level. Those steps provide insight into the stability and accuracy of the microscope and the overall quality of the sample preparation. On the other hand, advanced pre-processing consists of 2D classification and 2D class selection and is built upon the results of the basic pre-processing but the processing itself is on a 2D basis. With the help of those tasks it is possible to assess the quality of the sample itself and identify possible problems like preferred orientation or heterogeneity. Finally, 3D processing includes all the remaining steps of the SPA workflow and also builds upon the results of the basic pre-processing steps. However,

the 3D processing can additionally be influenced by the results of the advanced pre-processing steps. For example, particles could have been removed based on 2D classification results or 2D classes are used instead of particles as an input for the initial 3D reference estimation. All in all, the results of the live automated data processing allow the user to make educated decisions about the status of their data acquisition settings depending on the different tasks executed and effectively reduce the amount of unusable images.

Software like *WARP* [123], *Focus* [6], *RELION* [113], *Scipion* [108], *Appion* [68], *cryoSPARC* [96], and *cryoFlare* [111] already tackle those tasks and are able to automate the processing up to a certain point by chaining the individual steps of the SPA workflow together while collecting data at the microscope. However, they lack the ability to optimize the input settings for the data itself based on the behavior of the data set. First attempts of automated data optimization have been done by the Cianfrocco lab that recently published a deep-learning based tool that is able to select 2D class averages of good quality after successful 2D classification, and therefore remove those particles that are members of the class averages of worse quality or showing contamination [72]. However, it has not been designed for on-the-fly processing, and hence works best in a scenario where the data set is already collected, and each individual step is executed one after the other on the complete data set. An alternative approach has been implemented by the Liu lab, where the $\%_{\text{Res}}$, which is based on provided values by *RELION* [113] 2D classification, is evaluated and particles belonging to 2D classes with a value below a certain threshold are removed [77]. Additionally, the remaining particles are then used to re-train the particle picking model to yield better picking results in a subsequent picking process.

In the course of this thesis, the program TranSPHIRE [118] has been developed that enables efficient on-the-fly data processing and data optimization utilizing established tools in the field combined with new inventions from the *SPHIRE* [79] project. TranSPHIRE covers all of the initial pre-processing steps of the SPA workflow and additionally implements the TranSPHIRE feedback loop that automatically optimizes the particle picking performance in an iterative way based on 2D classification results (Figure 1.5).

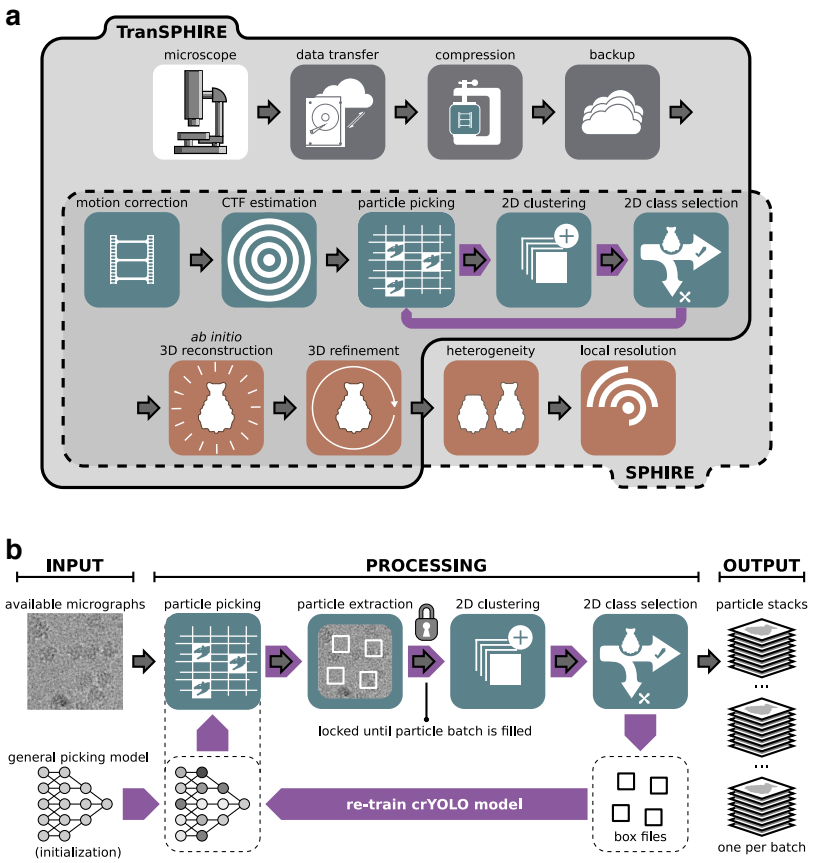

Figure 1.5: a Upper register (solid line): Overview of the integrated TranSPHIRE pipeline and all automated processing steps. The pipeline includes file management tasks, i.e., parallelized data transfer, file compression, and file backup (gray); 2D processing, i.e., motion correction, CTF estimation, particle picking, 2D clustering, and 2D class selection (turquoise); and 3D processing, i.e., *ab initio* 3D reconstruction and 3D refinement (red). Additionally, the pipeline includes an automated feedback loop optimization to adapt picking to the current data set during runtime (purple). Lower register (dotted line): The *SPHIRE* software package forms the backend for TranSPHIRE and offers the tools used for 2D and 3D processing. *SPHIRE* includes additional tools for advanced processing, such as heterogeneity analysis and local resolution determination. **b** The TranSPHIRE feedback loop. Gray arrows indicate the flow of data processing. Purple arrows indicate the flow of the feedback loop. Left (input): Micrographs are initially picked using the *crYOLO* general model. Center (processing): Particles are picked and extracted. Once a pre-defined number of particles have been accumulated, the pipeline performs 2D classification; the resulting 2D class averages are labeled as either ”kept” or ”discarded” by *Cinderella*. Class labels and *crYOLO* box files are then used to re-train *crYOLO* and adapt its internal model to the processed data. In the next feedback round this updated model is used to re-pick the data. Right (output): After five feedback rounds, the complete data set is picked with the final optimized picking model and 2D classified in batches. For every batch a particles stack of ”kept” particles is created and available for 3D processing. Figure and caption adapted from [118].

1.4 Aim of this thesis

In the past few years, cryo-EM established itself as a key method for high-resolution structure determination. While cryo-ET is still developing, the SPA approach leads to consistent structures with a resolution of better than 4 Å. However, the achievable resolution of a project is dependent on factors like specimen preparation, microscope alignment, and the quality and purity of the sample itself. Nonetheless, the resolution cannot be determined based on the visual appearance of the raw data. Rather, computational demanding processing is required to obtain a 3D structure which needs to be interpreted. Hence, it can take days, weeks, or even months to identify possible problems during data acquisition, which would render the data useless. Therefore, it is of utmost importance to not only increase the rate of the data collection, but additionally improve the quality of the data.

The aim of this thesis was to develop an automated on-the-fly processing pipeline for the SPA workflow with a focus on computational efficiency and adaption of the parameters to the data at hand to provide crucial information about the quality of the data. Therefore, the pipeline should cover all the necessary SPA steps up to the first 3D reconstruction to be able to detect the most resolution limiting factors. To yield close-to-optimal initial results, certain steps of the data processing would be optimized by a feedback-driven approach between particle picking and 2D classification. Additionally, those metrics should be presented via a graphical user interface (GUI) interface to provide as much information as possible and to allow for a fast analysis of the results to be able to correct for them as soon as they occur. Furthermore, the pipeline should not only be able to handle globular samples, but should work with filamentous samples as well.

Ultimately, the results of this thesis project would help to decrease the influence of resolution limiting factors in a data set that prevents them to culminate in a high-resolution 3D reconstruction. Combined with an efficient handling of computational resources, this approach can lead to high-resolution 3D reconstructions during data acquisition paving the way for cryo-EM to become a method for high-throughput structure determination.

2 Material and Methods

2.1 Computational resources

The presented results related of the TranSPHIRE [118] project were produced on a single machine equipped with two Intel(R) Xeon(R) Gold 6128 central processing units (CPUs)(3.4 GHz), three NVIDIA GeForce 1080 TI graphics processing units (GPUs), and 192 GB random access memory (RAM). Computational demanding processing like the 3D refinement or the *ab initio* 3D reconstruction as well as the results of the refinement of filaments were performed on the local high-performance computing (HPC) cluster CLEM. The nodes of the HPC cluster are equipped with two Intel(R) Xeon(R) Gold 6134 CPUs (3.2 GHz) and 382 GB RAM.

2.2 Data sets

2.2.1 Tobacco Mosaic Virus

A Tobacco Mosaic Virus (TMV) [40] data set was used for the evaluation of the refinement of filaments. This data set consists of 14 micrograph movies and their respective helical box coordinates and is available on EMPIAR entry EMPIAR-10020 [28]. The micrograph movies were collected at a Titan Krios (FEI Thermo Fisher) microscope equipped with an X-FEG and operated at 300 kV using *Serial EM* [75] for data acquisition. One collected movie contains 22 frames with an equal electron dose of 1.95 e/Å^2/frame and a pixel size of 1.126 Å/pixel collected with a K2 Summit (Gatan, Inc) direct electron detector.

For motion correction including dose weighting *MotionCor2* [139] version 1.3.2 without patch alignment was used. CTF estimation was performed using *CTER* [89, 79] with a C_s value of 2.7 between 4 Å and 30 Å. Segments were created based on the provided box files on EMPIAR with an overlap of 80 pixel and a box size of 300 using *e2helixboxer.py* from the *EMAN2* [121] package. For *sp_meridien.py* and *sp_meridien_alpha.py* the settings *--delta=3.75*, *--radius=132*, *--xr=5*, *--ts=1*, *--inires=15*, *--ccfpercentage=90* were used along with a soft-edge mask covering 85 % of the protein along the helical axis. Additionally, for *sp_meridien_alpha.py* the settings *--angle_method=M*, *--theta_min=80*, *--theta_max=100*, *--howmany=16*, *--helical_rise=1.41*, and *--filament_width=130* were used. The used reference was created based on an biological assembly (PDB:4UDV [40]) with a pixel size of 1.126 Å/pixel and a box size of 300.

The provided soft-edge mask was created using the program *sp_mask.py* based on the provided reference with the settings *--threshold*=0.003 8, *--fill_mask*, *--low_pass_filter_resolution*=15, *--pixel_size*=1.126, *--ndilation*=3, *--edge_width*=8, *--second_mask_shape=cylinder*, *--s_threshold*=1, *--s_nx*=300, *--s_ny*=300, *--s_nz*=255, *--s_ndilation*=0, *--s_edge_width*=15, and *--s_radius*=150.

For the SPHIRE sharpening program *sp_process.py --combinemaps* the same soft-edge mask

covering 60 % of the protein along the helical axis and *--B_enhance=50* was used. The provided soft-edge mask was created using the program *sp_mask.py* based on the unfiltered result of the first meridien run with the settings *--threshold=0.015*, *--ndilation=3*, *--edge_width=8*, *--second_mask_shape=cylinder*, *--s_edge_width=8*, *--s_radius=140*, *--s_nz=180*, *--s_nx=300*, and *--s_ny=300*

2.2.2 Actomyosin

An actomyosin data set was used for the evaluation of the refinement of filaments and the TranSPHIRE pipeline. This data was kindly provided by my colleague Dr. Sabrina Pospich [93]. The data set was collected at a C_s-corrected Titan Krios (FEI Thermo Fisher) microscope equipped with an X-FEG and operated at 300 kV using *EPU* [30] for data acquisition. The collected movie contains 40 frames with an equal electron dose of 2.03 e/Å^2/frame and a pixel size of 0.55 Å/pixel collected with a K2 Summit (Super resolution mode; Gatan, Inc) direct electron detector. Additionally, a GIF quantum-energy filter with a slit width of 20 e V was used.

For the processing in TranSPHIRE the number of feedback loop iterations were set to five. Within the TranSPHIRE pipeline the movies were drift corrected, dose weighted, and binned to a pixel size of 1.10 Å/pixel using *MotionCor2* [139] without patches to avoid filament distortions. CTF estimation was performed using *CTFFIND4* [106] with a C_s value of 0.001 between 4 Å and 30 Å. Since there is no general model for filaments available, a general model on data sets of actin filaments, but not the actomyosin or other actin complexes, was trained. Due to the fact that actin and actomyosin substantially differ in their visual appearance, this simulates the picking of an yet unknown filament with the help of a general model.

During the feedback loop iterations, the *crYOLO* picking confidence threshold is set to 0.1 and the anchor size to the estimated box size of 320 pixel. Additionally, the filament width parameters was set to 100 pixel and the box distance to 25 pixel, which is about the distance of the helical rise of 27.5 Å. A filament needs to consist of at least six segments to be considered. Once the fifth feedback round was finished, the picking confidence threshold was set to be 0.3, because the threshold evaluation script is not available for filamentous trainings. The resulting picked regions on the micrographs were extracted using the *sp_window.py* [79] program with a box size of 320 pixel and the expected filament width of 100 pixel. 2D classification was executed in batches of 20 000 particles with a provided particle radius of 160 pixel, a group size of 50, and a minimum group size of 30. For the 2D class selection with *Cinderella* [11] the general model from the website was used combined with a conservative confidence threshold of 0.1. Since filamentous samples differ strongly from the training data of the single particle based general *Cinderella* model, the *TranSPHIRE* pipeline was stopped after the 2D classification of the first feedback round to train a new *Cinderella* model. In addition to the afterwards manually labeled 2D class averages, as ”kept” labeled class averages of pure actin and as ”discarded” labeled class averages of contamination was added to the data prior training. The *crYOLO* training was performed on a maximum of 50 micrographs that contained ”kept” labeled particles by *Cinderella* [11].

The evaluation of the resulting models from every feedback loop iteration was done using *sp_auto.py* [79] on a fixed subset of 100 micrographs using identical input settings as described above. For the evaluation of the refinement of filaments the cleaned stack of the final *TranSPHIRE* feedback loop iteration with a *crYOLO* picking threshold of 0.3 containing 45 297 particles from 97

different micrographs was used.

For *sp_meridien.py* and *sp_meridien_alpha.py* the settings *--delta=1.875*, *--radius=132*, *--xr=5*, *--ts=1*, *--inires=15*, *--ccfpercentage=90* were used along with a soft-edge mask covering 85 % of the protein along the helical axis. Additionally, for *sp_meridien_alpha.py* the settings *--angle_method=M*, *--theta_min=90*, *--theta_max=90*, *--howmany=16*, *--helical_rise=27.5*, and *--filament_width=20* were used. The used reference was created based on an biological assembly (PDB:5JLH [22]) with a pixel size of 1.1 Å/pixel and a box size of 320.

The provided soft-edge mask was created using the program *sp_mask.py* based on the provided reference with the settings *--threshold*=0.008, *--fill_mask*, *--low_pass_filter_resolution*=15, *--pixel_size*=1.1, *--ndilation*=3, *--edge_width*=8, *--second_mask_shape=cylinder*, *--s_threshold*=1, *--s_nx*=320, *--s_ny*=320, *--s_nz*=272, *--s_ndilation*=0, *--s_edge_width*=15, and *--s_radius*=160.

For the SPHIRE sharpening program *sp_process.py --combinemaps* the same soft-edge mask covering 60 % of the protein along the helical axis and *--B_enhance=50* was used. The provided soft-edge mask was created using the program *sp_mask.py* based on the unfiltered result of the first meridien run with the settings *--threshold=0.01*, *--ndilation=3*, *--edge_width=8*, *--second_mask_shape=cylinder*, *--s_edge_width=8*, *--s_radius=160*, *--s_nz=192*, *--s_nx=320*, and *--s_ny=320*

2.2.3 Tc holotoxin

A Tc holotoxin data set was used for the evaluation of the *TranSPHIRE* pipeline. The sample is the ABC holotoxin from *Photorhabdus luminescens* in lipid nanodisc [105] and is available on EMPIAR entry EMPIAR-10313 [29]. This data set consists of the pre-pore and pore state of the holotoxin, i.e., of a mixture of conformational states. The data set was collected at a C_s-corrected Titan Krios (FEI Thermo Fisher) microscope equipped with an X-FEG and operated at 300 kV using *EPU* [30] for data acquisition. The collected movie contains 40 frames with an equal electron dose of 1.52 e/Å^2/frame and a pixel size of 0.525 Å/pixel collected with a K2 Summit (Super resolution mode; Gatan, Inc) direct electron detector. Additionally, a GIF quantum-energy filter with a slit width of 20 e V was used.

For the processing in TranSPHIRE the number of feedback loop iterations were set to five. Within the TranSPHIRE pipeline the movies were drift corrected, dose weighted, and binned to a pixel size of 1.05 Å/pixel using *MotionCor2* [139] with a patch value of `"3 3 0"`. CTF estimation was performed using *CTFFIND4* [106] with a C_s value of 0.001 between 4 Å and 30 Å. For particle picking, the general model of *crYOLO* was used.

During the feedback loop iterations, the *crYOLO* picking confidence threshold is set to 0.1 and the anchor size to the estimated particle diameter of 205 pixel. Once the fifth feedback round was finished, the optimal threshold was evaluated by the *cryolo_evaluation.py* to be 0.194 based on the input training data. The resulting picked regions on the micrographs were extracted using the *sp_window.py* [79] program with a box size of 420 pixel. 2D classification was executed in batches of 20 000 particles with a provided particle radius of 160 pixel, a group size of 100, and a minimum group size of 50. For the 2D class selection with *Cinderella* [11] the general model from the website was used combined with a conservative confidence threshold of 0.1. To demonstrate the ability of the TranSPHIRE feedback loop to learn how to distinguish sub-populations within the data, the *Cinderella* model was trained on existing 2D class averages of the pore state as instances of ”kept” classes (318) and 2D class averages of the pre-pore state mixed with contaminations as instances of

”discarded” classes (644). The *crYOLO* training was performed on a maximum of 50 micrographs that contained ”kept” labeled particles by *Cinderella* [11].

The evaluation of the resulting models from every feedback loop iteration was done using *sp_auto.py* [79] on a fixed subset of 500 micrographs using identical input settings as described above. For the 3D refinement, *sp_meridien.py* [79] was used with an initial model created by *sp_rviper.py* [79] based on classes labeled as ”kept” by *Cinderella* [11] of each *sp_auto.py* [79] run is used. Additionally, no symmetry was imposed and the refinement ran without a mask to prevent model bias.

2.2.4 Transient receptor channel 4

A transient receptor channel 4 (TRPC4) data set was used for the evaluation of the TranSPHIRE pipeline. This data set was received upon request from my colleague Dr. Deivanayagabarathy Vinayagam and the sample is TRPC4 from zebra fish in lauryl maltose neopentyl glycol (LMNG) detergent [129]. The data set was collected at a C_s-corrected Titan Krios (FEI Thermo Fisher) microscope equipped with an X-FEG and operated at 300 kV using *EPU* [30] for data acquisition. The collected movie contains 50 frames with an equal electron dose of 1.77 e/Å^2/frame and a pixel size of 0.85 Å/pixel collected with a K2 Summit (Super resolution mode; Gatan, Inc) direct electron detector. Additionally, a GIF quantum-energy filter with a slit width of 20 e V was used.

For the processing in TranSPHIRE the number of feedback loop iterations were set to five. Within the TranSPHIRE pipeline the movies were drift corrected and dose weighted using *MotionCor2* [139] with a patch value of `"5 5 20"`. CTF estimation was performed using *CTFFIND4* [106] with a C_s value of 0.001 between 4 Å and 30 Å. To avoid bias of the *crYOLO* model TRPC related data sets were removed from the training data prior training a new *crYOLO* model. Additionally, the coordinates of 90 % of the resulting picks were were randomized to simulate a worst-case-scenario.

During the feedback loop iterations, the *crYOLO* picking confidence threshold is set to 0.1 and the anchor size to the estimated particle diameter of 240 pixel. Once the fifth feedback round was finished, the optimal threshold was evaluated by the *cryolo_evaluation.py* to be 0.357 based on the input training data. The resulting picked regions on the micrographs were extracted using the *sp_window.py* [79] program with a box size of 288 pixel. 2D classification was executed in batches of 20 000 particles with a provided particle radius of 120 pixel, a group size of 100, and a minimum group size of 50. For the gls2D class selection with *Cinderella* [11] the general model from the website was used combined with a conservative confidence threshold of 0.1. The *crYOLO* training was performed on a maximum of 50 micrographs that contained ”kept” labeled particles by *Cinderella* [11].

The evaluation of the resulting models from every feedback loop iteration was done using *sp_auto.py* [79] on a fixed subset of 500 micrographs using identical input settings as described above. For the 3D refinement, *sp_meridien.py* [79] was used with an initial model created by *sp_rviper.py* [79] based on classes labeled as ”kept” by *Cinderella* [11] in every feedback loop iteration was used. Additionally, a c4 symmetry was imposed and the refinement ran without a mask to prevent model bias.

2.3 Refinement of filaments

For the adjustments made to the 3D refinement the program *sp_meridien.py* [79] from the *SPHIRE* [79] package was used as a starting point (Algorithm 5). The *SPHIRE* [79] package is codistributed with the cryo-EM software package *EMAN2* [121] and the versions used for modifications were 1.3 and 2.31, respectively.

First, the particle stack is randomly split into two groups with about the same defocus distribution. Meridien has four operating modes: INITIAL, PRIMARY, EXHAUSTIVE, and RESTRICTED. Each mode consists of two major steps: The coarse grid search and the fine grid search. The coarse grid has an angular distance of twice the current angular distance (default current angular starting distance `--delta`=7.5°), which results in a lower computational demand. The fine grid has an angular distance of the current angular distance. Those projection parameters in the close proximity of the best matches of the coarse grid search are evaluated for the final projection parameter assignment. Finally, a 3D reconstruction is calculated from the particles with their respective projection parameter assignments weighted by their likelihood. The resulting 3D reconstruction is filtered to the current resolution and used as an input to the subsequent refinement iteration. Once the procedure converges, i.e., the resolution does not increase and the assigned projection parameters stay about the same, a final unfiltered 3D reconstruction is calculated.

During the INITIAL mode, only the very best reference match of each particle is taken into account for the 3D reconstruction, which makes this step behave very similarly to the *projection matching* strategy [88]. In the PRIMARY mode all reference matches of each particle are taken into consideration for the fine grid searches that have negative squared Euclidean distance values above −10. At the end of each step, only the best projection parameters whose accumulated likelihood is smaller than a user defined threshold are taken into account for the 3D reconstruction (default `--ccfpercentage`=99.9 %). The number of projections taken into account for each particle after the final filtering define the *smear* value of the respective particle. Additionally, the background noise is estimated from the data. The EXHAUSTIVE mode uses the same reference match method as in PRIMARY, and uses the estimated background noise values of the PRIMARY step. During the RESTRICTED mode, each particle is not compared to every possible reference projection, but only to those within a small region around the best match from the previous iteration.

Algorithm 5 Description of the Meridien algorithm

```
Center the input particles
Split the input particles into two independent groups
Filter the input reference to the user-defined resolution
Filter particles by reducing the image size
Set the filtered input reference as the reference volume
while Refinement did not converge do
  if Iteration == 1 then
    Set Mode to INITIAL
  else if Mode == INITIAL then
    Set mode to PRIMARY
  else if Angular projection parameters did not change significantly and Resolution
  did not improve then
    if Angular accuracy is high then
      Refinement converged
    end if
    Reduce the shift search range
    Reduce the angular distance by half
    Increase particle box size
    if Mode == PRIMARY then
      Set mode to EXHAUSTIVE
    else if Mode == EXHAUSTIVE and delta <= 1.875 then
      Set mode to RESTRICTED
    end if
  end if
  Generate coarse and fine search grid for shifts and angles
  Shake the search grids to prevent overfitting

  Generate coarse grid reference projections from the reference volume
  Calculate the Euclidean distance for each particle with each reference projection
  Keep the best matches
  Identify nearest neighbors on the fine grid to the best matches

  Generate nearest neighbors reference projections from the reference volume
  Calculate the Euclidean distance for each particle with each reference projection
  Keep the best matches

  Perform filtered 3D reconstruction
  Set the reconstruction as the new reference volume
end while
Perform final unfiltered 3D reconstruction
```

2.4 TranSPHIRE

The TranSPHIRE [118] implementation consists of three parts: the TranSPHIRE GUI, the TranSPHIRE worker, and the embedded external programs. TranSPHIRE is an open-source software for the Linux operating system utilizing *Python*[35] version 3.6 and is freely available online (https://github.com/MPI-Dortmund/transphire). The package can be installed with the *Python* package manager PIP and a manual is available from the TranSPHIRE wiki (https://transphire.readthedocs.io)

2.4.1 TranSPHIRE GUI

The TranSPHIRE GUI serves as the entry point for the user and helps setting up the TranSPHIRE session, manages the communication with the TranSPHIRE worker, and visualizes the results of the individual worker processes. The GUI utilizes the wrapper *PyQt* [34] version 5.9.2 for the *QT* [13] framework version 5.9.2. For data visualization, the data is imported with *numpy* [49] version 1.19.4 and visualized with *matplotlib* [58] version 3.3.3. Notifications can be sent via GUI pop-ups, *Telegram* [124] bots with the help of the *telepot* [69] module, or e-mail.

2.4.2 TranSPHIRE worker

While the TranSPHIRE GUI focuses on the interaction with the user, the TranSPHIRE worker is running the TranSPHIRE pipeline in several independent processes. The program utilizes the multiprocessor module and the queue module of the shipped standard library for sub-process spawning and for inter-process communication, respectively.

2.4.3 External software

Within the TranSPHIRE workflow several well established software packages and applications from external sources are wrapped and available through the TranSPHIRE GUI. For motion correction the software packages *MotionCor2* [139] and *Unblur* [45], for CTF estimation the software packages *CTER* [89, 79], *CTFFIND4* [106], and *gCTF* [138], for particle picking *crYOLO* [131], for particle extraction *sp_window.py* [79], for 2D classification *GPU ISAC* [44], for 2D class selection *Cinderella* [11], for 3D ab-initio reconstruction and 3D refinement *sp_auto.py* [79], which uses the programs *sp_rviper.py* [79] for *ab initio* 3D reconstruction and *sp_meridien.py* [79] or *sp_meridien_alpha.py* for the 3D refinement of single particles or filaments, respectively, and utility tools *EMAN2* [121] and *IMOD* [65].

The presented results were produced with TranSPHIRE [118] v1.4.50 and *SPHIRE* [79] v1.4. Specifically, the software versions used were: Cuda 10.2.86 version of *MotionCor2* [139] v1.3.0, *CTFFIND4* [106] v4.1.13 for CTF estimation, *crYOLO* [131] v1.6 for particle picking, SPHIRE *sp_window.py* [79] for particle extraction, *GPU ISAC* [44] v1.0 of SPHIRE *ISAC* [136] for 2D classification, SPHIRE *Cinderella* [11] v0.5 for 2D class selection, SPHIRE *sp_rviper.py* [79] for *ab initio* 3D reconstruction, and SPHIRE *sp_meridien.py* [79] or *sp_meridien_alpha.py* executed via the SPHIRE *sp_auto.py* [79] for the 3D refinement of single particles or filaments, respectively.

3 Results

The aim of this thesis was to invent an automated cryo-EM workflow for single particles and filamentous samples. As a starting point, the internally developed SPA software package *SPHIRE* [79] was used. To add support for filamentous samples, modifications to the programs were necessary, and those will be presented in section 3.1. Afterwards, the newly developed on-the-fly processing tool TranSPHIRE [118] is presented in section 3.2. The results related to the TranSPHIRE workflow have been published in Nature Communications [118].

3.1 Processing of filaments

To allow the processing of filamentous samples within the SPA based software package *SPHIRE* [79], different modifications were necessary. The focus was on the 3D refinement, since it is the most crucial step of every cryo-EM structure determination project. Additionally, minor adjustments to the pre-processing programs, utility programs, and the *SPHIRE* [79] GUI were made.

3.1.1 Adjustments to pre-processing and utility programs

sp_window.py

The *sp_window.py* program extracts, i.e., crops out, the provided regions of interest from the micrographs. Since cryo-EM micrographs have a low SNR, the identification of protein signal is error prone and often leads to false-positive picks, i.e., regions marked as protein signal which actually contain noise or contamination. Therefore, in the following the term ”particle” will be used to refer to the content of the cropped area rather than a potential protein signal within the cropped area. In SPA, the coordinates for the square regions of interest are chosen in a way that each particle contains the center of the signal identified as protein in its center. Thus, one particle contains potentially one unique asymmetric unit. On the other hand, in filamentous processing rectangular regions of interest are used, whose coordinates are chosen in a way that contain the center of the straight elongated signal identified as a filament in its center. Since helical filaments are composed of the same repeating subunit, particles are extracted with an overlap so that each particle contains a unique asymmetric unit in its center. Therefore, the protein within the particle is commonly referred to as segment.

After the extraction of the particles, in SPA a circular mask is used to calculate the mean and the standard deviation of the signal inside the central region, i.e., the area where the protein is located. Those values are then used to normalize the image by subtracting the mean of the central region and dividing by the standard deviation. Filamentous samples have a rod-like shape spanning over the entire box along the helical axis, while often being much thinner perpendicular to it. Therefore, a circular mask could include a lot of noise and can lead to distorted pixel statistics.

To overcome this issue, a rectangular mask along the helical axis was introduced for the calculation of the image statistic. However, this requires knowledge about the orientation of the segment inside the particle and the approximate width of the filament. Therefore, the orientation angle is calculated from the orientation of the filament, since the filament is identified as a whole and the individual particles are extracted from it afterwards. The orientation angle is saved in the metadata for each particle so that it can be used in later stages of the processing. Additionally, the filament id and the segment id are stored. The width of the filaments is provided by the user.

sp_isac2.py

The *sp_isac2.py* program performs 2D clustering to clean the data set of false-positive particles, i.e., contamination or noise falsely identified as protein signal. Therefore, it provides an initial overview of the 2D views present in the data. To yield the best clustering results, an initial global 2D alignment is performed which puts all particles into global register, i.e., find the orientation which maximizes the overlap of similar features. Thus, all available particles are averaged to form a first initial reference image. Afterwards, each particle is compared to the reference image to identify the optimal particle orientation which overlaps best. A new reference image for the subsequent iteration is created by averaging all particles after the respective 2D alignment parameters are applied. Therefore, the particles are centered by their center of mass within their respective boxes in an iterative way.

However, for filamentous samples, the segments always show a rod-like shape. Therefore, an initial reference image containing only signal within a rectangle parallel to the X-axis and with a width of the filament is used as a reference image for the first five out of overall 14 alignment iterations. Thus, the alignment is guided to properly center the filaments in the center of the box. Afterwards, the resulting shifts parallel to the helical axis are set to 0 for each particle prior clustering. That is, because filamentous particles are extracted with an overlap of up to 95 % based on the helical rise of the filament and the alignment procedure tends to overlap neighboring particles by shifting along the helical axis. This would lead to identical central regions of the particles for neighboring particles, and hence not only a distorted clustering result, but also blurry class averages due to information from redundant members and the resulting missing noise reduction. Therefore, it can be beneficial to disallow any particle shift during the main iterations of *sp_isac2.py* by setting the parameter `--xr=0`, and allow the clustering can focus on the central subunit of each individual particle.

Additionally, the particle normalization strategy using a rectangular mask instead of a circular mask is adopted from *sp_window.py*.

sp_pipe.py

After the successful execution of *sp_isac2.py*, the utility tool *sp_pipe.py* is used to remove those particles from the data set that were marked as outliers by the *ISAC* [136] algorithm. However, this can break the continuity of particles originating from one filament. Therefore, the program was modified to split fragmented filaments into sub-filaments that contain contiguous particles and remove those sub-filaments that contain less than the user-defined number of particles. Since contiguous filaments with only a few associated particles in the data set can lead to distorted filament projection parameter statistics, it can be beneficial to remove those from the data set.

3.1.2 Adjustments to the 3D refinement

The *sp_meridien.py* program performs the 3D refinement to calculate a high-resolution reconstruction by identifying and assigning 3D projection parameters to the 2D particles. To work with filamentous samples, a SPA based refinement strategy was implemented that utilizes additional knowledge about the particles stemming from their filamentous character [78]. The modified version is available as *sp_meridien_alpha.py* in the *SPHIRE* [79] package.

Filamentous constraints Firstly, the filaments formed by the individual subunits are typically long compared to the size of the holes of the grid, and therefore the filaments are oriented almost parallel to the thin ice layer. This results in an out-of-plane rotation angle Θ to values close to 90°, i.e., all angle combinations that are about perpendicular to the helical axis. Secondly, the particles of a filament are extracted in a consecutive way. Therefore, neighboring particles inside a filament should have similar shifts perpendicular to the helical axis, out-of-plane rotation angles Θ, and in-plane rotation angles Ψ. Thirdly, each particle of a filament is boxed with a unique central subunit while neighboring subunits are unique central subunits of a neighboring particle. Hence, the allowed shift along the helical axis per particle should not exceed about half the helical rise Δz of the filamentous structure to prevent identical assignments of projection directions for the same central subunit.

The projection directions in *sp_meridien_alpha.py* are described by the Euler angles Φ, Θ, and Ψ. To work properly, the 3D reference is expected to be orientated so that the helical axis is parallel to the z-axis of the box. Therefore, a rotation around the helical axis is represented by the angle Φ in the range from 0° to 360°, the out-of-plane rotation in respect to the thin ice layer is reflected by the angle Θ in the range from 0° to 180°, and the in-plane-rotation angle in respect to the thin ice layer is reflected by the angle Ψ in the range from 0° to 360°, respectively. To account for the first constrain the existing angular assignment creation method P has been modified to always contain the out-of-plane rotation angle Θ of 90°. Adapted from the P method, the newly created method M starts with a Θ value of 90° and continues by subtracting and adding the current angular distance value $\Delta\Theta$ to identify all Θ angles within the user-defined range. To approximate an even distribution of angles within the whole range of possibilities, the angular distance $\Delta\Phi$ between Φ values for each Θ value is defined with the help of the formula $\Delta\Phi = {}^{\Delta\Theta}/_{\sin\Theta}$.

The second constraint can be used to analyze the behavior of a data set after each iteration of the 3D refinement. By comparing the projection parameters of particles within one filament, inconsistency of those particles can be identified for possible data set optimizations at a later point in the processing pipeline. To identify outliers in the out-of-plane rotation angle Θ the Θ_{mean} value of all particles within one filament is determined and all values outside the range $\Theta_{\text{mean}} \pm 15°$ (default value) are considered outliers. On the other hand, the identification of outlier particles within one filament for the in-plane rotation angle Φ is more complex due to the periodic boundary conditions of the angle at 360°, i.e., the angle 350° has the same angular distance of 10° from 0° and 360° as the angle 10°.

To account for the wrapping, the original Ψ range from 0° to 360° is first translated to the range from −180° to 180° by subtracting values larger than 180° by 360°. Afterwards, the median is calculated based on the absolute values of the angles and the median gets a negative sign if the number of negative angles is larger than the number of positive angles. The influence of the

wrapping effect is reduced by adding the median value to all angles and adjust the angle ranges again to −180° to 180°. Finally, the median is iteratively determined based on the rotated angles until convergence, i.e., the median being 0 or is alternating between two values. Following the determination of the median, the mean value is iteratively calculated based on the angle values inside the range −30° to 30° (default value). Particles are then labeled Ψ outliers if they are not within the range −15° to 15° (default value) around the determined mean value. Additionally, whole filaments are considered outliers if more than 20 % of their particles are considered outliers. Those statistics can be used to either remove filaments from the data set or interpolate the expected angle values and start a subsequent local refinement.

Lastly, to avoid large shifts along the helical axis the respective 2D shift value is reduced to the range $^{-\Delta z}/_{2}$ to $^{\Delta z}/_{2}$ before each iteration. However, the filaments are arbitrarily oriented and therefore the component of the shift along the helical axis needs to be identified first with the help of the known filament orientation from the particle extraction step. First, the shift vector $\vec{s}$ is rotated by the filament orientation angle ψ by the rotation matrix M

$$M = \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix}$$

to the rotated shift vector $\vec{s}_{\text{rot}} = M\vec{s}$, whose X-axis represents the helical axis and the Y-axis the shift perpendicular to the helical axis. Afterwards $\vec{s}_{\text{rot,x}}$ is reduced to the valid range by

$$\vec{s}_{\text{rot,x}} = \left(\left(\vec{s}_{\text{rot,x}} + \frac{\Delta z}{2}\right) \mod \Delta z\right) - \frac{\Delta z}{2},$$

before rotating the shift vector $\vec{s}_{\text{rot}}$ back to its original coordinate system $\vec{s}_{\text{new}} = M^{\text{T}}\,\vec{s}_{\text{rot}}$.

Those changes are complemented by extra changes to the overall *sp_meridien.py* processing strategy.

General processing strategy In addition to the changes introduced based on the filamentous character, the processing strategy described in section 2.3 is modified to yield better results for filamentous samples. Since the expected out-of-plane rotation angle Θ value is about 90° the coarse search grid only contains Θ values of 90° (default `--theta_min=90` and `--theta_max=90`) and is not shaken for the initial search step. On the other hand, the fine search grid is not altered. However, to include out-of-plane rotation angle values further away from 90° the `howmany` parameter, i.e., the number of neighboring points on the fine search grid to the best matches on the coarse search grid, is increased from 4 to 10 (default value).

Limiting the number of comparisons on the coarse grid to values along the equator allows for the usage of smaller initial angular distance values $\Delta\Theta$ like 3.75° or even 1.875° to account for filamentous samples with small helical twist and rise.

Finally, the initial 2D pre-alignment strategy is adopted from *sp_isac2.py* to only center the filaments perpendicular to the helical axis and the particle normalization strategy using a rectangular mask instead of a circular mask is adopted from *sp_window.py*.

The adapted strategy for helical specimen was tested on a TMV and an actomyosin data set.

Refinement of Tobacco Mosaic Virus To show the capabilities of the new refinement strategy a TMV data set with 30 968 extracted filament particles from 14 micrographs were used.

The initial reference was filtered to 15 Å. The programs *sp_meridien.py*, i.e., without filament related modifications, and *sp_meridien_alpha.py*, i.e., with filament related modifications, were run five times each with identical settings on the same data set.

Running *sp_meridien.py*, the particles were divided into chunks of 16 066.20 ± 1 512.85 and 14 901.80 ± 1 512.85 for chunk 0 and chunk 1, respectively (Table 3.1, Table 5.2). The number of iterations it took for the refinement to finish was 24.20 ± 3.35 with iteration 3.00 ± 0.00 being the best iteration resulting in a final nominal resolution of (10.92 ± 2.89) Å. The FSC curves of the individual runs show a drop at about 20 Å followed by noise dominated values until about 9 Å. Afterwards, the values decrease slowly to a FSC value of about 0 (Figure 3.1d, Table 5.20-5.24). A visual inspection of the resulting 3D reconstructions confirm the reported resolutions (Figure 3.1b).

Running *sp_meridien_alpha.py*, the particles were divided into chunks of 15 470.60 ± 1 098.54 and 15 497.40 ± 1 098.54 for chunk 0 and chunk 1, respectively (Table 3.1, Table 5.1). The number of iterations it took for the refinement to finish was 22.00 ± 2.00 with iteration 22.00 ± 2.00 being the best iteration resulting in a final nominal $FSC_{0.143}$ resolution of (4.37 ± 0.08) Å. Calculations of outliers resulted in 31.80 ± 42.63 and 47.20 ± 42.63 outliers for chunk 0 and chunk 1, respectively. The FSC curves of the individual runs show a decreasing behavior from 1 starting at about 11 Å to 0 at about 4 Å (Figure 3.1c, Table 5.15-5.19). A visual inspection of the resulting 3D reconstructions confirm the reported $FSC_{0.143}$ resolutions (Figure 3.1a).

All in all, the modifications for filamentous samples present in *sp_meridien_alpha.py* led to a major improvement of the achieved nominal $FSC_{0.143}$ resolution from (10.92 ± 2.89) Å to (4.37 ± 0.08) Å. In the runs using *sp_meridien.py*, the best iteration was 3.00 ± 0.00, which is internally the first iteration taking into consideration for a resolution estimation. Therefore, the refinement could no longer improve from that point on for the subsequent 21 iterations. On the other hand, running *sp_meridien_alpha.py* the achieved high $FSC_{0.143}$ resolution of (4.37 ± 0.08) Å can be confirmed by a visual inspection of the 3D reconstruction and the healthy appearance of the FSC. The jitter in the values of the FSC can be explained because only 14 micrographs were used, hence only 14 defocus groups, leading to missing information at certain spatial frequencies.

Table 3.1: 3D refinement results of the TMV data set running the helical version *sp_meridien_alpha.py* and the SPA version *sp_meridien.py* with the same input settings five times each.

Parameter	*sp_meridien_alpha.py*	*sp_meridien.py*
#Particles Chunk 0	15 470.60 ± 1 098.54	16 066.20 ± 1 512.85
#Outliers Chunk 0	31.80 ± 42.63	0.00 ± 0.00
#Particles Chunk 1	15 497.40 ± 1 098.54	14 901.80 ± 1 512.85
#Outliers Chunk 1	47.20 ± 42.63	0.00 ± 0.00
#Iterations	22.00 ± 2.00	24.20 ± 3.35
Best iteration	22.00 ± 2.00	3.00 ± 0.00
$FSC_{0.143}$ resolution / Å	4.37 ± 0.08	10.92 ± 2.89
$FSC_{0.5}$ resolution / Å	5.30 ± 0.21	18.39 ± 2.14

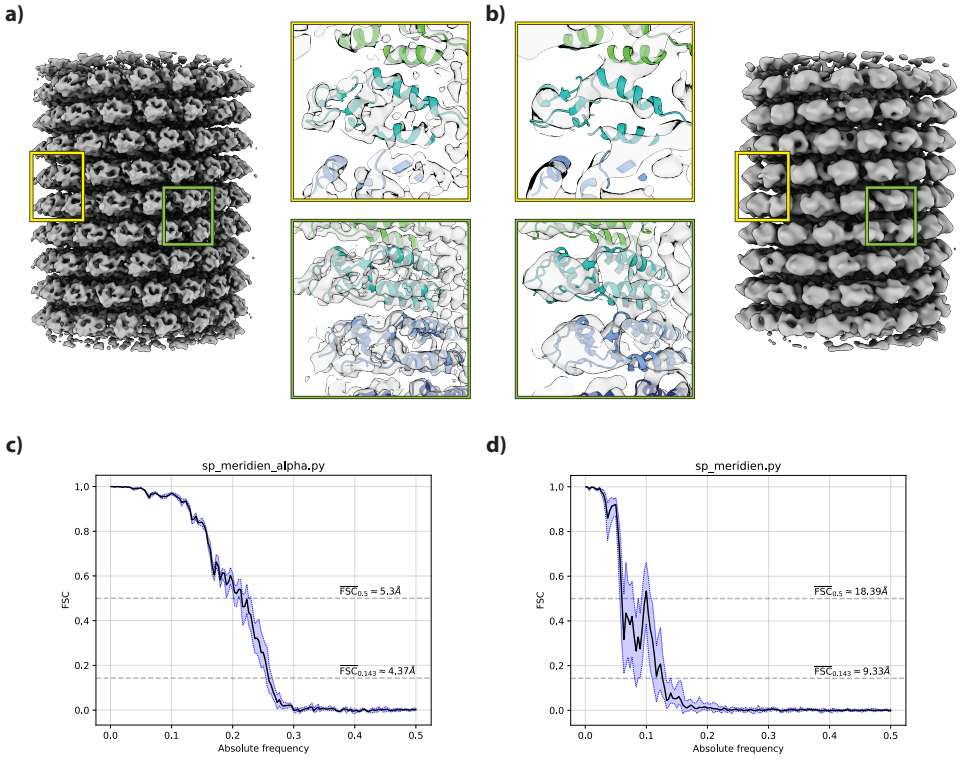

Figure 3.1: Representative sharpened 3D reconstruction of TMV computed from 30 968 particles using **a** *sp_meridien_alpha.py* and **b** *sp_meridien.py*. The yellow and green boxes indicate regions that are shown as close-up views in the center. Here, the atomic model of TMV (PDB:6R7M, colored by monomer) is shown in addition to the density map (transparent gray). While most of the secondary structure elements, including the twist of the α-helices, were resolved by *sp_meridien_alpha.py* **a**, *sp_meridien.py* was only able to reconstruct the overall shape of the protein **b**. **c-d** Masked FSC curves of the reconstructions computed by *sp_meridien_alpha.py* **c** and *sp_meridien.py* **d**, respectively. The black curve represents the average FSC curve, whereas the blue area illustrates the spread of results, i.e., the worst and best FSC curve gained within a total of five independent runs. Resolution values according to the $FSC_{0.5}$ and $FSC_{0.143}$ (gold standard) criterion are given for the average FSC curve (black). The significant difference in the map quality (see **a-b**) is in perfect agreement with the achieved resolutions, which is significantly higher in case of *sp_meridien_alpha.py* **c**.

Refining actomyosin To show the capabilities of the new refinement strategy, a actomyosin data set with 45 297 extracted filament particles from 97 micrographs was used. The initial reference was filtered to 15 Å. The programs *sp_meridien.py*, i.e., without filament related modifications, and *sp_meridien_alpha.py*, .i.e., with filament related modifications, were run five times each with identical base settings.

Running *sp_meridien.py*, the particles were divided into chunks of 22 891.80 ± 1 979.52 and 22 405.20 ± 1 979.52 for chunk 0 and chunk 1, respectively (Table 3.2, Table 5.4). The number of iterations it took for the refinement to finish was 17.20 ± 2.77 with iteration 17.20 ± 2.77 being

the best iteration resulting in a final nominal $FSC_{0.143}$ resolution of (4.47 ± 0.02) Å. The FSC curve show a slight plateau in the range of 7 Å to 5 Å (Figure 3.2d, Table 5.10-5.14). A visual inspection of the resulting 3D reconstructions confirm the reported nominal $FSC_{0.143}$ resolutions (Figure 3.2b). However, it is noticeable that the periphery of the 3D reconstruction is less well resolved compared to the inner area.

Running *sp_meridien_alpha.py*, the particles were divided into chunks of 22 706.00 ± 514.94 and 22 591.00 ± 514.94 for chunk 0 and chunk 1, respectively (Table 3.2, Table 5.3). The number of iterations it took for the refinement to finish was 23.60 ± 1.52 with iteration 23.40 ± 1.52 being the best iteration resulting in a final nominal $FSC_{0.143}$ resolution of (4.40 ± 0.20) Å. Calculations of outliers resulted in 7 570.60 ± 148.22 and 7 631.00 ± 263.95 outliers for chunk 0 and chunk 1, respectively. The FSC curve show a slight plateau in the range of 6 Å to 5 Å (Figure 3.2c, Table 5.5-5.9). A visual inspection of the resulting 3D reconstructions confirm the reported nominal $FSC_{0.143}$ resolutions (Figure 3.2a). However, it is noticeable that the periphery of the 3D reconstructions is less well resolved compared to the inner area.

All in all, the modifications for filamentous samples present in *sp_meridien_alpha.py* did not improve the nominal $FSC_{0.143}$ resolution of the 3D reconstructions compared to the unmodified version *sp_meridien.py*. However, analyzing the overall FSC values of both programs the values for *sp_meridien_alpha.py* are higher than those of the *sp_meridien.py* in the range from about 8 Å to 4.5 Å which is also reflected by the better $FSC_{0.5}$ resolution of (6.83 ± 0.28) Å compared to (7.49 ± 0.00) Å. A more detailed inspection of structural details confirms the presence of higher resolved structural details in the 3D reconstructions of the modified version *sp_meridien_alpha.py* (Figure 3.2e-f).

Table 3.2: 3D refinement results of the actomyosin data set.

Parameter	*sp_meridien_alpha.py*	*sp_meridien.py*
#Particles Chunk 0	22 706.00 ± 514.94	22 891.80 ± 1 979.52
#Outliers Chunk 0	7 570.60 ± 148.22	0.00 ± 0.00
#Particles Chunk 1	22 591.00 ± 514.94	22 405.20 ± 1 979.52
#Outliers Chunk 1	7 631.00 ± 263.95	0.00 ± 0.00
#Iterations	23.60 ± 1.52	17.20 ± 2.77
Best iteration	23.40 ± 1.52	17.20 ± 2.77
$FSC_{0.143}$ resolution / Å	4.40 ± 0.20	4.47 ± 0.02
$FSC_{0.5}$ resolution / Å	6.83 ± 0.28	7.49 ± 0.00

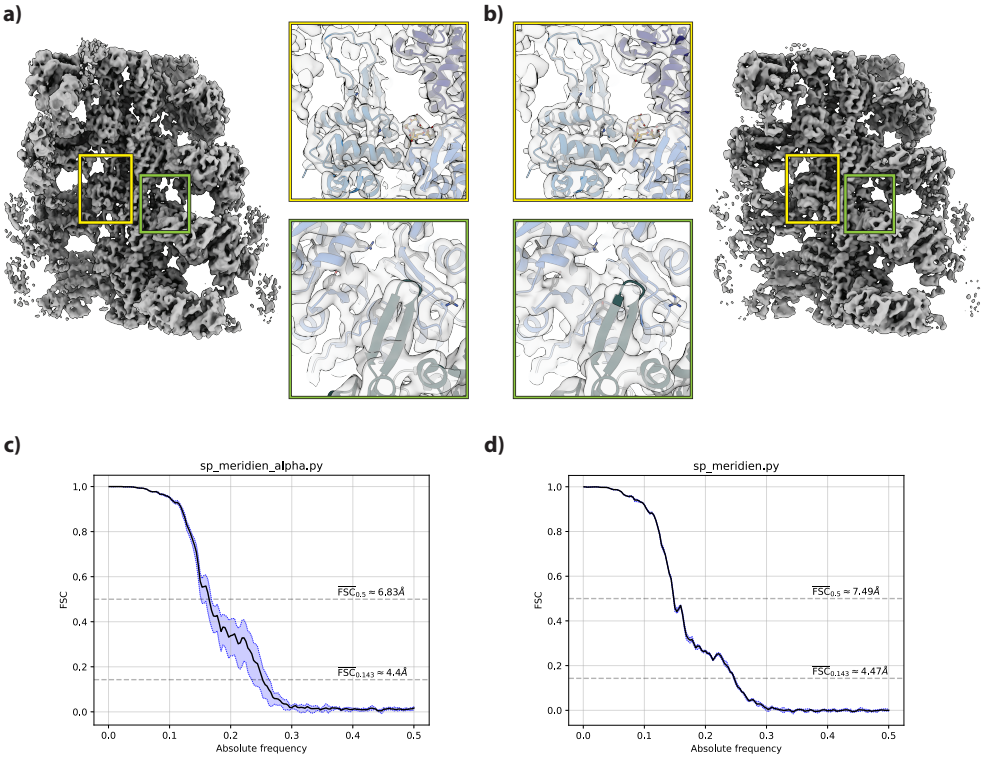

Figure 3.2: Representative sharpened 3D reconstruction of actomyosin computed from 45 297 particles using **a** *sp_meridien_alpha.py* and **b** *sp_meridien.py*. The yellow and green boxes indicate regions that are shown as close-up views in the center. Here, the atomic model of actomyosin (PDB:7PLU, colored by monomer) is shown in addition to the density map (transparent gray). While both programs were able to resolve most of the secondary structure elements, including the twist of the α-helices, the density computed by *sp_meridien_alpha.py* shows more details, such as the bound ligand (depicted in yellow) and the β-hairpin. **c-d** Masked FSC curves of the reconstructions computed by *sp_meridien_alpha.py* **c** and *sp_meridien.py* **d**, respectively. The black curve represents the average FSC curve, whereas the blue area illustrates the spread of results, i.e., the worst and best FSC curve gained within a total of five independent runs. Resolution values according to the $FSC_{0.5}$ and $FSC_{0.143}$ (gold standard) criterion are given for the average FSC curve (black). The significant difference in the map quality (see **a-b**) is in perfect agreement with the achieved resolutions, which is significantly higher in case of *sp_meridien_alpha.py* **c**.

3.1.3 Discussion

To enable filamentous processing in the *SPHIRE* [79] package modifications to the pre-processing programs *sp_window.py*, *sp_isac2.py*, and *sp_pipe.py* were introduced to utilize information about the filamentous character of the particles. The in-plane rotation, the filament id, and the segment id are stored in the metadata of each particle. These parameters are used for normalization, limiting the shift along the helical axis, and the calculation of filamentous consistency of the 3D projection

parameters. Other modern software packages for filamentous processing, such as *Helical RELION* [50] and *cryoSPARC* [96], follow a similar strategy in their pre-processing pipeline. However, their maximum likelihood based 2D classification approaches tend to produce large inhomogeneous classes. This can be problematic, because filamentous particles typically look very similar due to their overlap during extraction and continuous repeat of the same asymmetric unit. The *ISAC* [136] algorithm on the other hand is known for its capability to produce homogeneous classes with few members. Therefore, *ISAC* [136] is very suitable to successfully classify filamentous particles. *RELION* recently announced an improved version of its 2D classification algorithm which could produce more homogeneous and smaller classes [101]. However, the usability for filamentous particles needs to be investigated.

The 2D classification step assigns particles into groups of similar projection angles for a fast cleaning and quality assessment of the data set. However, the continuity of filaments can be destroyed due to the removal of particles assigned to "discarded" classes. In the presented 3D refinement program *sp_meridien_alpha.py* the consistency of the assigned projection parameters for each particle within the same filament is analyzed. To create proper statistics, a minimum number of contiguous particles per filament is required. Thus, filaments are split into contiguous sub-filaments, and particles from sub-filaments that are too short are removed from the data set. Other filamentous refinement methods do not perform consistency checks within one filament and therefore do not require contiguous filaments.

Most of modern filamentous refinement programs, such as *Helical RELION* [50], *SPIDER* [38], *SPARX* [57], *FREALIGN* [119], *FREALIX* [107], *SPRING* [17], and *cryoSPARC* [96], follow the IHRSR approach, which combines 3D refinement with helical symmetry estimation. However, estimating the helical symmetry parameters especially on low resolution structures can lead to model bias and falsely estimated structures [91, 25]. To circumvent model bias due to assumptions about the helical symmetry the 3D refinement program *sp_meridien_alpha.py* does implement a processing strategy free from enforcement of helical symmetry. Enforcing a symmetry during processing has the advantage that only a few particles are needed to achieve a high-resolution 3D reconstruction. On the other hand, applying a symmetry assumes that each particle inside a filament has the exact same information. Therefore, using a symmetry is especially useful for very rigid filaments with little variation. However, the helical symmetry parameters differ locally in most specimens [24].

As an alternative to implying a helical symmetry, a filamentous processing strategy has been developed in the Raunser lab which applies constraints to the processing stemming from the geometry of filaments [78]. The original workflow is implemented in *RELION* and involves file conversions, manual metadata adjustments, and the usage of advanced parameters in the 3D refinement. The 3D refinement itself is executed in two steps, the first is a global refinement which is limited to a search range for the out-of-plane rotation angle Θ of 90° and the second is a local refinement with a narrow search range close to the results of the first refinement. In between those two refinements, the processing is stopped and the consistency of projection parameters within a filament is analyzed. Filaments with many outliers are removed from the data set, while the projection parameters of outliers in filaments with only a few outliers are adjusted to match the filamentous constraints. In the implementation of *sp_meridien_alpha.py*, this strategy is available natively in the refinement itself and no manual execution of the strategy is required. Additionally, the consistency of projection parameters is not only calculated between the global and local refinement, but in every refinement iteration. This consistency check allows for the monitoring of

the refinements quality and might provide indications about possible problems with the data set.

The 3D refinement program *sp_meridien.py* has another very useful unique feature called "user functions". With the help of user-provided functionality, the reference map for each refinement iteration can be modified. Since *sp_meridien_alpha.py* has this functionality available as well, it is possible to introduce for example helical symmetrization to the reference map to improve the SNR of the map in early refinement iterations. Therefore, it is possible to combine the advantages of the symmetry-free approach with some of the advantages of helical symmetry if applicable to the project.

The next steps for the 3D refinement could be to utilize the consistency checks even more. For example, it could be possible to calculate the helical symmetry parameters from the assignments of the individual particles within a filament. Since hundreds to thousands of filaments are available within one data set it is possible to get statistics about the helical symmetry to get an in-depth understanding of the data set. Furthermore, the way outliers are determined and consistent parameters are calculated could be optimized. The parameters currently used are optimized especially for actin filaments, as it is the most used filamentous sample in the Raunser lab. Therefore, it could be interesting to test multiple parameters such as the minimum filament length or thresholds for the outlier determination on a multitude of data sets to identify sets of parameters that work best for most of them.

The programs *sp_meridien.py* and *sp_meridien_alpha.py* are developed for HPC systems, i.e., the performance increases as more compute processes are available. However, in the field of cryo-EM the workflow switches increasingly from expensive and large computing clusters to single workstations with at least one GPU. On the one hand, especially small research groups can run GPU accelerated processing software, on the other hand, the system is maintainable even for inexperienced users. *RELION* for example moved their algorithms to the GPU a few years ago and gained an acceleration of more than an order-of-magnitude [62]. *cryoSPARC* [96] entered the field one year later with the aim to solve the speed issue in cryo-EM workflows of these days. Their algorithms and implementation manages to solve 3D structures within minutes or hours compared to hours or days, while being able to result in a high resolution [95]. Therefore, rewriting the *sp_meridien_alpha.py* refinement to run on the GPU can make it accessible to research groups unable to afford an HPC system. A first step has already been made when the 2D classification program *ISAC* [136] was ported to the GPU and made available as *GPU ISAC* [44].

Another possibility for performance optimization is the usage of a different file format for the input images. Currently the Berkeley database (bdb) file format [85] is used, which slows down input/output operations on many systems. An alternative approach is the self-defining text archive and retrieval (STAR) file format used in *RELION* [48]. Reading the data from the STAR format is not only faster, but due to the popularity of the *RELION* workflow the STAR format is also a well established input and output format for most of the available software. Therefore, not only can the performance of *sp_meridien_alpha.py* be further improved, but the program would additionally be more accessible to more users. Hence, the *SPHIRE* [79] project will release soon a new version which allows the STAR file format to be used for all input/output operations.

In terms of variety, the presented actomyosin data set and TMV data set are different in many aspects such as visual appearance, rigidity, and helical symmetry parameters. Nevertheless, in both cases the filamentous refinement yielded a high-resolution structure. A comparison with the results of the unmodified SPA refinement revealed that both cases reached a higher resolution.

This indicates that the filamentous refinement works for a multitude of data sets. When comparing the achieved resolutions to the literature, the resolution is slightly worse than the published results [50, 93]. However, this is expected due to the limited number of particles used for the refinement combined with the absence of an applied helical symmetry. A limited number of particles was chosen to avoid resolution saturation effects. In addition, the Raunser lab successfully published high-resolution structures of filamentous samples [93, 92, 3, 41, 4].

All in all, the filamentous refinement strategy implemented in *sp_meridien_alpha.py* offers a symmetry unbiased approach to reach high-resolution for filamentous samples, which is especially beneficial for curved filaments and those with a flexible helical symmetry.

3.2 Automated processing with TranSPHIRE

Automated on-the-fly data processing which automatically adjusts to the data at hand is a crucial step towards high-throughput structure determination. Therefore, I developed TranSPHIRE [118]: an automated on-the-fly processing pipeline including a deep-learning based feedback loop to optimize to the data at hand. Additionally, it provides a GUI for user input and visualization of results. The source code of the program is available on Github [128].

3.2.1 Graphical User Interface

The interaction with the TranSPHIRE pipeline process and the visualization of the results is the main purpose of the TranSPHIRE GUI 3.3. To keep the GUI responsive, tasks like the TranSPHIRE pipeline, which runs the actual programs, and the TranSPHIRE data import, which imports the results of the pipeline for visualization, are outsourced in separate processes. The administrative area in the upper left part of the GUI allows the setup of notification receivers, import of previously created setting templates, and to start or monitor the TranSPHIRE pipeline. Notifications can either be received via E-Mail or a *Telegram* [124] bot and it is possible to specify multiple receivers. Templates allow modification of the TranSPHIRE default values for all present settings and therefore minimizes required user input. Upon pressing the TranSPHIRE **Start** button, first all TranSPHIRE inputs are checked for validity as far as possible and the provided settings are passed to the TranSPHIRE pipeline process. Alternatively, the results of an already running TranSPHIRE pipeline or a previous TranSPHIRE pipeline run can be visualized with the **Monitor** option.

The processing status area in the right part of the GUI informs about the available disk space on the different systems involved in processing and the overall progress of the TranSPHIRE pipeline process. The user is informed about possible problems like not having enough disk space or failing processes of the TranSPHIRE pipeline. Additionally, a text area showing the log of the TranSPHIRE pipeline is available and the pipeline log file, the error file, and the GUI log file can be opened via the click of the respective button. The pipeline log file contains the information and errors coming from the TranSPHIRE GUI regarding the setup of the TranSPHIRE pipeline. The error file contains the information about failing individual programs running within the TranSPHIRE pipeline processes. The GUI log stores the timestamps of the individual TranSPHIRE pipeline processes. The **notes** button can be used to enter notes related to the data set.

The setup and visualization area in the lower left part of the GUI is used to set up the TranSPHIRE pipeline and visualize its results. The first horizontal layer of tabs consists of **Mount**, **Settings**,

Retrain, and **Visualization**.

Under the **Mount** tab, the user can mount previously specified file systems on a per-user basis. Once mounted, its available and total disk space can be monitored in the processing status area of the GUI.

The **Settings** tab is used to setup the TranSPHIRE pipeline. A vertical layer of tabs guides the user through the setup process so that all required entries for the TranSPHIRE pipeline are filled. If multiple programs are available for one tab entry, an additional vertical layer of tabs is present. Additionally, each setting area under the vertical tabs has an additional horizontal layer of tabs named **Main**, **Advanced**, and **Rare** which allows the assignment of priorities for each setting. Hence, every possible setting of the process is exposed to the user to cover special needs for specific use cases, but those settings which are changed on a regular basis are easy to access.

The **Retrain** area is used to fine-tune the results of the TranSPHIRE feedback loop. Here, the results of the *Cinderella* [11] classification in ”kept” and ”discarded” 2D class averages is shown. Additionally, adjustments can be made and the *Cinderella* [11] model can be trained to improve the classification results.

The last horizontal tab **Visualization** provides an overview over all results of the TranSPHIRE pipeline. Each processing program contains a horizontal tab bar comprising **Overview**, **Show images**, **Plot per micrograph**, and **Plot histogram**. To allow for the monitoring of several of the plots at the same time, each of the tab areas can be disconnected from the GUI and positioned on the monitor as needed. The **Plot per micrograph** tab shows the result of the program for each output parameter, which can be selected from a horizontal tab bar at the bottom, on a per micrograph basis. To adjust the display, input areas for masking are available and additionally statistics like the minimum, maximum, mean, or median are presented. Additionally, the **Plot histogram** contains the same information as the **Plot per micrograph** area, but instead of presenting the output parameters on a per micrograph bases they are shown in the form of a histogram. In the **Overview** tab area, all plots available in the **Plot per micrograph** and **Plot histogram** tabs are available at the same time. To check or adjust one plot in detail, left clicking it will navigate to the respective plot area, while right clicking the plot will hide it from the overview. Therefore, a fast analysis of the main parameters of the data acquisition is possible. Finally, the **Show images** tab contains images on a per micrograph basis and shows information, such as the image, the power spectrum, coordinates for particle picking, and class averages. The information presented in the **Visualization** tab is prepared by the TranSPHIRE data import process, which checks for new available data every 20 s, imports the data in the expected internal format, and is started during the initialization of the TranSPHIRE pipeline process.

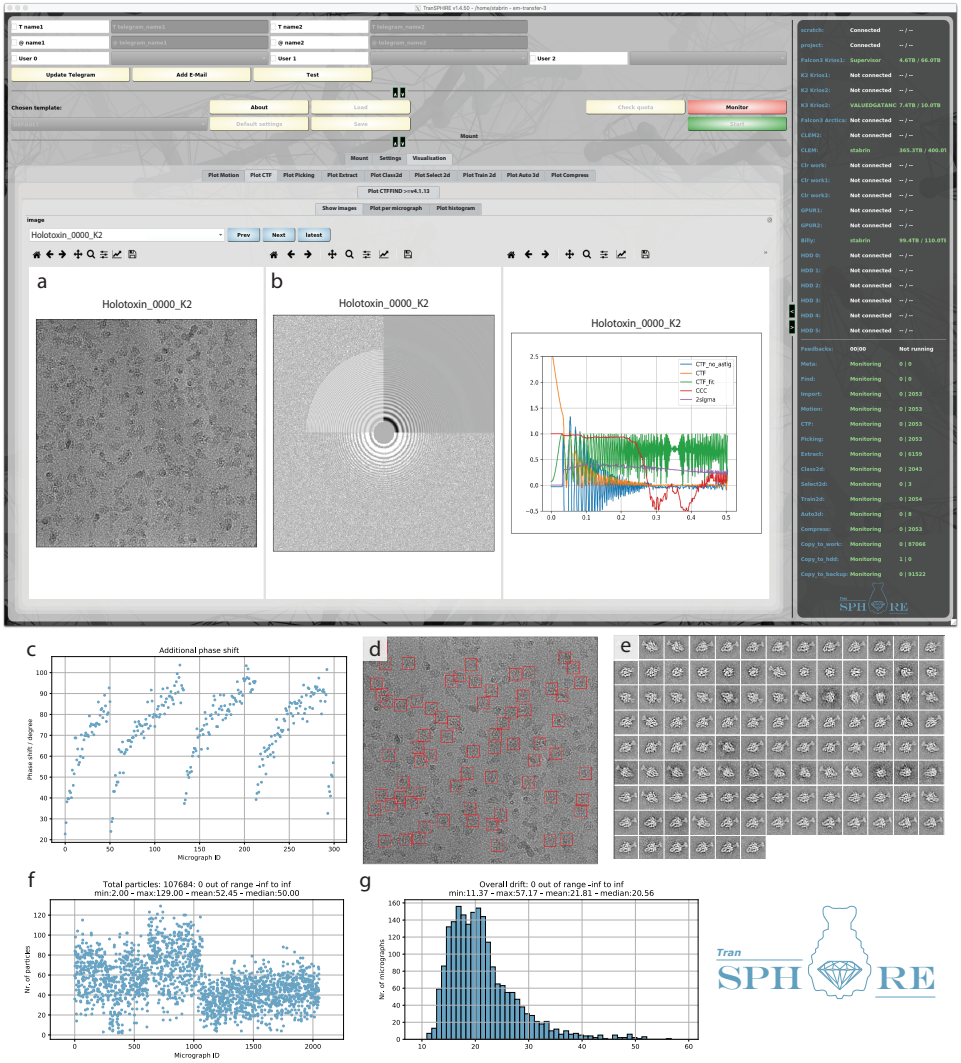

Figure 3.3: The TranSPHIRE GUI is organized in tabs for initial setup, data processing settings, and live visualization of data acquisition and processing results. Visualization options include incoming micrographs (a), CTF fitting results (b), and phase shift (c), picking results (d), and 2D class averages (e). The shown phase shift development follows a logarithmic curve and helps experimentalists to decide when to switch to a new phase plate position (c). The shown picking result depicts the use of an optimized picking model, trained during runtime to only pick pore state particles (d). Output values can be plotted as either a scatter plots or histograms, as shown here for the total number of particles (f), and the overall drift per micrograph (g), respectively. This live monitoring enables an early evaluation of data quality during data acquisition and provides initial information about the protein structure. Figure and caption adapted from [118].

3.2.2 TranSPHIRE pipeline

The TranSPHIRE pipeline process is started as a sub-process of the TranSPHIRE GUI when the **Start** button is pressed (Figure 3.4). First, the provided settings in the GUI are passed to the process, the required output folders are created, and the TranSPHIRE data import process is started. Afterwards, the data queues used for communication between the processes are initialized. Each queue exists in memory as well as files on the file system and the content is always synchronised. Therefore, the status of the processing can be easily recovered even after a crash of the computer. The user defined number of sub-processes, which run the actual processing of the TranSPHIRE pipeline in a parallel manner, are started before an event loop handles the communication between the TranSPHIRE GUI and the output of the TranSPHIRE pipeline sub-processes. Each sub-process handles one individual task, and to utilize available hardware resources in an optimal way there can be multiple processes of the same sub-process type. The event loop is exited when the **Stop** button is pressed.

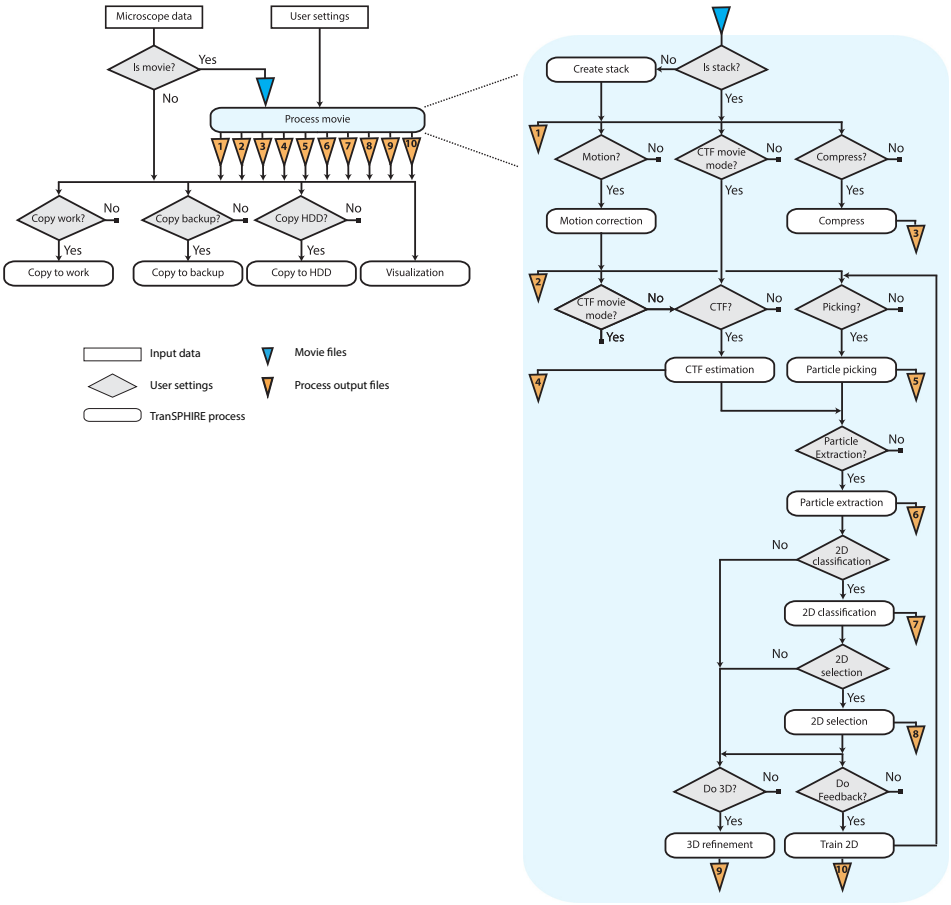

Figure 3.4: TranSPHIRE workflow. Flow chart of the TranSPHIRE pipeline depicting sequentially executed processes below each other and parallel running processes next to each other. The workflow is highly adaptable allowing, for example, the binning of super resolution data during motion correction and CTF estimation. All inputs from the microscope, outputs from the involved processes, and additional statistics produced by TranSPHIRE are monitored and presented live in the GUI. If specified, the processes "Copy to work", "Copy to backup" and "Copy to HDD" create a copy of the results of each individual step to a workstation or cluster, a backup server, or an external hard drive, respectively. Figure and caption adapted from [118].

Firstly, the *Find* task is constantly crawling the provided input directory for new incoming micrographs. Once a new micrograph arrives, the *Import* task copies the related data to the local machine for further processing. Afterwards, the *Motion* task performs motion correction on the copied stack. Additionally, the stack is compressed by the *Compress* task if necessary and CTF estimation is performed by the *CTF* task if the movie mode of the program is requested. On the other hand, if the movie mode of the CTF estimation program should not be used the output of the motion correction is used to perform CTF estimation.

The *Picking* task runs every 30 s and uses all the output of the *Motion* task that was produced

during that time as input to limit program initialization overhead and therefore speedup the pipeline. Once the *Picking* task, the *CTF* task, and the *Picking* task finished the processing of the same micrograph the *Extract* task uses the outputs to perform particle extraction. Those extracted particles are then used as an input for the *Class2d* task which performs 2D classification once a specified amount of particles is accumulated. Afterwards, the *Select2d* task classifies the resulting 2D class averages into ”kept” and ”discarded” and selects those particles belonging to classes in the ”kept” category. Finally, the *Auto3d* task performs *ab initio* 3D reconstruction, if no initial 3D reference is provided, or directly performs 3D refinement once a specified amount of ”kept” classes (200 by default) and particles (40 000 by default) is accumulated, respectively. However, *ab initio* 3D reconstruction is performed only once at the beginning and every subsequent 3D refinement uses the resulting volume as input. The *Auto3d* process can be executed on a remote machine available via secure shell (SSH) such as a HPC cluster, because the 3D refinement and the *ab initio* 3D reconstruction are computational expensive.

Each process within TranSPHIRE is responsible for a single process type such as data import, motion correction, and CTF estimation. The process monitors its respective queue, starts as soon as new data arrives, prepares the command to run, runs the command, prepares the outputs for the subsequent tasks and visualization, and puts the micrograph into the queues of the subsequent processes. Additionally, the process checks for conditions that needs to be met before running the command like a certain amount of particles need to be extracted from several micrographs before running 2D classification or 3D refinement. After the command has finished, the output is analyzed for errors and values outside the user specified range. While oversubscribing the CPU is typically not an issue due to plenty of RAM available, the RAM on the GPU is limited allowing to run only one single task at at time. Therefore, an additional queue management system for commands running on the GPU was developed to avoid crashes while scheduling the execution of commands running on the CPU is left to the operating system.

While the basic TranSPHIRE pipeline is of linear character, the TranSPHIRE feedback loop allows information from later parts of the pipeline influence the settings of tasks earlier in the pipeline to perform data optimization. Specifically, the output of the 2D class selection is used to improve the performance of the particle picking task utilizing the deep learning particle picker *crYOLO* [131]. Therefore, during the feedback loop the *Train2d* process trains a new *crYOLO* model based on the ”kept” particles after the *Select2d* process and the resulting model is used as the model for *Picking*. Additionally, the queues for *Extract*, *Class2d*, *Select2d*, and *Train2d* are emptied at the end of a feedback loop iteration, while the queue for the *Picking* process is reset to apply the new model to all micrographs collected up to this point.

For the evaluation of the TranSPHIRE processing speed a Tc holotoxin data set consisting of 2 053 micrographs, each containing 36 particles on average, collected at a speed of 188 micrographs/h (Fig. 3.5). All TranSPHIRE processes were run on the same machine with 12 physical cores, two GeForce GTX 1080 Ti GPU and 128 GB of RAM. Shortly after the data collection stopped after about 11 h, the processes *Import*, *Compress*, *Motion*, and *CTF* finished processing as well. Additionally, the processes for *Picking*, *Extract*, *Class2d*, and *Select2d* also finished after about 11 h. All five feedback loop iterations finished after about 7.3 h allowing the first *ab initio* 3D reconstruction to start after about 8 h and finish after about 9.1 h. The first feedback loop iteration finished after about 2 h with first 2D class averages being available after about 1.4 h. Only the 3D refinement, which started after about 9.1 h and were running for about 6.4 h, finished 4.5 h later than the data collection. All in all,

the TranSPHIRE was able to process the incoming data as fast as new images were acquired. Only the 3D refinement and *ab initio* 3D reconstruction, which took about 1.1 h and 6.4 h, respectively, caused a delay which could be resolved by running them on a more powerful remote HPC system.

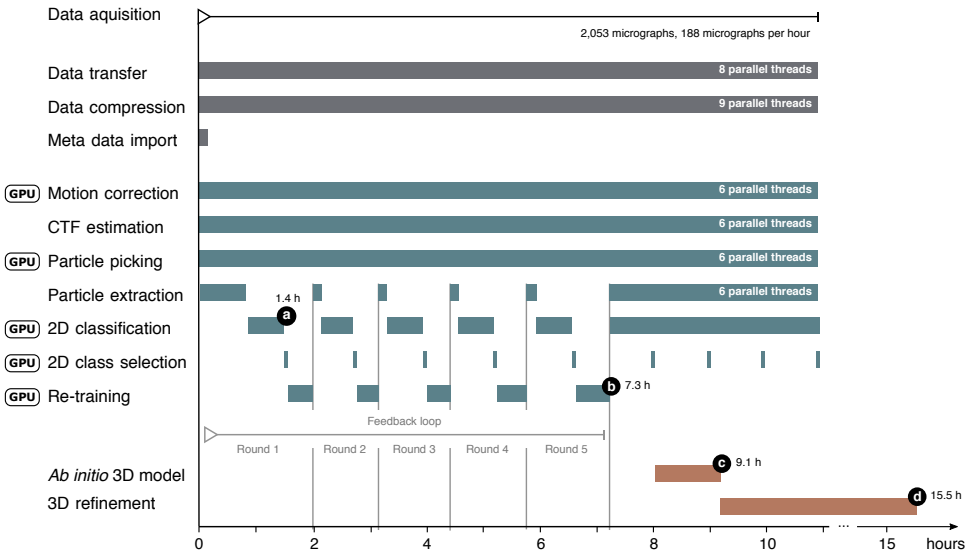

Figure 3.5: Timeline depicting the parallel execution of the processes of the TranSPHIRE pipeline. Timings are based on a Tc holotoxin data set consisting of 2 053 micrographs, each containing 36 particles on average, collected at a speed of 188 micrographs/h (K2 super-resolution, 40 frames). TranSPHIRE ran on-the-fly up to the creation of an *ab initio* 3D reconstruction using default settings. Important milestones are denoted in black: **a** first 2D class averages produced after 1.4 h; **b** end of the feedback loop after 7.3 h; **c** *ab initio* 3D reconstruction after 9.1 h; and **d** final 3D reconstruction of the first batch of particles after 15.5 h. Due to the internal scheduling of modern operating systems, and because not every TranSPHIRE thread is always working to capacity, the number of available CPUs (12 physical cores) and assigned TranSPHIRE threads (45) is not identical, and does not limit the speed of the computations. Figure and caption adapted from [118].

3.2.3 TranSPHIRE feedback loop

Machine learning based particle pickers, such as *crYOLO* [131], train an internal model to learn how to pick the particles optimally. On the one hand, if trained on a multitude of different data sets it is possible to generalize to unseen data sets. On the other hand, this ability is limited by the number and variety of available training data. Therefore, if the features present in the data are too different from the features present in the training data the model is not guaranteed to yield an optimal picking result. However, solving this issue requires different manual user intervention steps, namely detection of the insufficient picking performance, manual picking of a small representative subset of the data, and training of a new model based on the manually picked data. The TranSPHIRE feedback loop resolves this issue of a non-optimal picking performance without user intervention.

Initially, the user provides a generic *crYOLO* model, which is typically not trained on the collected

data set or behaves poorly, and specifies a low picking confidence threshold to decrease the false-negative picking rate, i.e., fewer particles will be missed. Afterwards, those identified particles are extracted and provided to the 2D classification program *GPU ISAC* [44], which is known for its ability to identify small groups of similar views in the data and is also able to discard noise by its internal stability checks. Therefore, it effectively takes care of the increased false-positive picks introduced by the low initial picking confidence threshold. Finally, those class averages are classified by the deep-learning based binary classification tool *Cinderella* [11] into ”kept” and ”discarded” classes. The coordinates of those particles contributing to the ”kept” classes are then used to train a new *crYOLO* picking model. Additionally, for single particle projects the optimal picking confidence threshold for the model is evaluated by the *cryolo_evaluation.py* program of the *crYOLO* package after the last feedback iteration. During the feedback iterations, however, a conservative picking confidence threshold of 0.1 is used by default.

Once the new model and threshold is available they are used in subsequent processing and replace the initially provided settings after the queues for particle picking, particle extraction, 2D classification, and 2D class selection are reset. The loop runs by default five times to optimize the *crYOLO* picking model to the data, and therefore improve the initial picking result which is crucial for a fast high-resolution 3D refinement result.

In the following, three different scenarios of common cryo-EM use cases are presented: Particle picking optimization of unknown data without user intervention, analysis of sub-populations within the data, and the processing of helical specimen. For each case the TranSPHIRE pipeline with the TranSPHIRE feedback loop enabled is run once. Every feedback iteration starts after about 20 000 particles are extracted, hence every feedback iteration operates on a different subset of the data. However, the amount of extracted particles is often below 20 000, because particles that clip the image border are rejected from being extracted. Therefore, the results of the feedback iterations only serve as indications for the behavior of the respective *crYOLO* models and an additional evaluation on a fixed number of micrographs was performed.

First, the results of the feedback loop and the evaluation of the models are presented. Their results are discussed at the end of each subsection.

Optimize particle picking without user intervention

To demonstrate the ability of the TranSPHIRE feedback loop to optimize the particle picking performance to yet unknown data, a data set of a TRPC4 membrane protein channel is used.

Feedback loop As an initial *crYOLO* model, a general model for *crYOLO* and *Cinderella* were provided which had been trained based on training data which do not contain any TRPC data sets. A liberal picking threshold value of 0.1 was used while 2D class selection was performed with a liberal threshold of 0.1 to keep all particles and classes that might represent a protein. Additionally, an objectively bad picking performance has been simulated to worsen the initial picking result by the *crYOLO* general model even more by randomizing the coordinates of 90 % of the picks on each micrograph before 2D classification in the first iteration of the feedback loop.

While the number of micrographs required to reach about 20 000 particles increased from 106 in the first feedback iteration to 951 in the second, the number dropped to 182 in the third and gradually decreased further to 129 in the fifth feedback iteration (Table 3.3). The number of total

particles picked per micrograph decreased from 184.58 to 19.01 from the first to the second feedback iteration and afterwards gradually increased to 149.47 in the fifth feedback iteration. The yield of "kept" particles increased from 2.54 % to 60.89 % from the first to the fourth feedback iteration and decreased to 49.94 % in the fifth. Similarly, the achieved resolution after the 3D refinement improved from 13.60 Å to 4.01 Å from the first to the fourth feedback iteration and decreased to 4.08 Å in the fifth.

The optimal picking confidence threshold for the model of the final fifth iteration was evaluated to be 0.357 by the *crYOLO* evaluation tool.

Table 3.3: Results of the TranSPHIRE feedback loop of the TRPC4 data set. Each feedback iteration started after about 20 000 particles were collected.

Feedback iteration	#Micrographs	#Particles total	#Particles total / Micrograph	#Particles "kept"
1 + To.1	106	19 566	184.58	496
2 + To.1	951	18 079	19.01	8 657
3 + To.1	182	18 929	104.01	10 491
4 + To.1	145	19 303	133.12	11 753
5 + To.1	129	19 281	149.47	9 629

Feedback iteration	#Particles "kept" / Micrograph	Particles "kept" / %	Resolution / Å
1 + To.1	4.68	2.54	13.60
2 + To.1	9.10	47.88	4.22
3 + To.1	57.64	55.42	4.15
4 + To.1	81.06	60.89	4.01
5 + To.1	74.64	49.94	4.08

Evaluation To evaluate the picking performance, the resulting *crYOLO* models after each feedback iteration are used to perform particle picking on a fixed subset of 500 micrographs from the data set (Table 3.4). Since the picking result of the first feedback iteration has been additionally sabotaged, the performance of the general *crYOLO* model has been additionally evaluated for a comparison with the final picking result.

Analyzing the particle picking performance, the absolute number of identified "kept" classes monotonically increased from 23.00 ± 1.18 to 360.00 ± 9.09 throughout the feedback loop. Additionally, the amount of "kept" particles per micrograph increased from 4.17 ± 0.18 to 62.93 ± 1.12. The achieved resolution increased from (5.51 ± 0.33) Å to (3.54 ± 0.04) Å, while the relative amount of "kept" particles stays between (44.83 ± 1.08) % and (49.25 ± 0.99) % after the second feedback loop iteration.

Using the optimal confidence threshold of 0.357, the relative amount of "kept" picks increased from (47.86 ± 1.08) % to (56.16 ± 1.09) % while the number of "kept" picks per micrograph slightly decreased to 62.93 ± 1.22 from 67.73 ± 1.53. Additionally, the number of total picks per micrograph

decreased from 141.52 ± 0.00 to 112.05 ± 0.00 reducing the total amount of picked particles by about 26 %. However, the achieved resolution stays similar at about 3.5 Å.

The reference run using the *crYOLO* general model with a confidence threshold of 0.1 as input yielded 83 319.00 ± 0.00 particles, i.e., 166.64 ± 0.00 particles per micrograph. After 2D classification and 2D class selection, 36 504.70 ± 881.47 "kept" particles are left which translates into 73.01 ± 1.76 "kept" particles per micrograph and a yield of (43.81 ± 1.06) %. The final resolution was (3.50 ± 0.03) Å.

Table 3.4: Results of the evaluation of the TranSPHIRE feedback loop on a subset of 500 micrographs of the TRPC4 data set. The listed values are the mean and standard deviation based on repeating the evaluation runs 10 times.

Feedback iteration	#Particles total	#Particles total / Micrograph	#Particles "kept"
ref + To.1	83 319.00 ± 0.00	166.64 ± 0.00	36 503.70 ± 881.47
1 + To.1	8 850.00 ± 0.00	17.70 ± 0.00	2 087.30 ± 89.18
2 + To.1	48 857.00 ± 0.00	97.71 ± 0.00	21 903.50 ± 529.75
3 + To.1	62 984.00 ± 0.00	125.97 ± 0.00	31 016.70 ± 622.11
4 + To.1	73 175.00 ± 0.00	146.35 ± 0.00	33 228.10 ± 57.98
5 + To.1	70 758.00 ± 0.00	141.52 ± 0.00	33 864.20 ± 766.45
5 + To.375	56 026.00 ± 0.00	112.05 ± 0.00	31 467.00 ± 611.07

Feedback iteration	#Particles "kept" / Micrograph	Particles "kept" / %	Resolution / Å
ref + To.1	73.01 ± 1.76	43.81 ± 1.06	3.50 ± 0.03
1 + To.1	4.17 ± 0.18	23.59 ± 1.01	5.51 ± 0.33
2 + To.1	43.81 ± 1.06	44.83 ± 1.08	3.64 ± 0.02
3 + To.1	62.03 ± 1.24	49.25 ± 0.99	3.55 ± 0.03
4 + To.1	66.46 ± 1.16	45.41 ± 0.79	3.55 ± 0.03
5 + To.1	67.73 ± 1.53	47.86 ± 1.08	3.54 ± 0.04
5 + To.375	62.93 ± 1.22	56.16 ± 1.09	3.56 ± 0.03

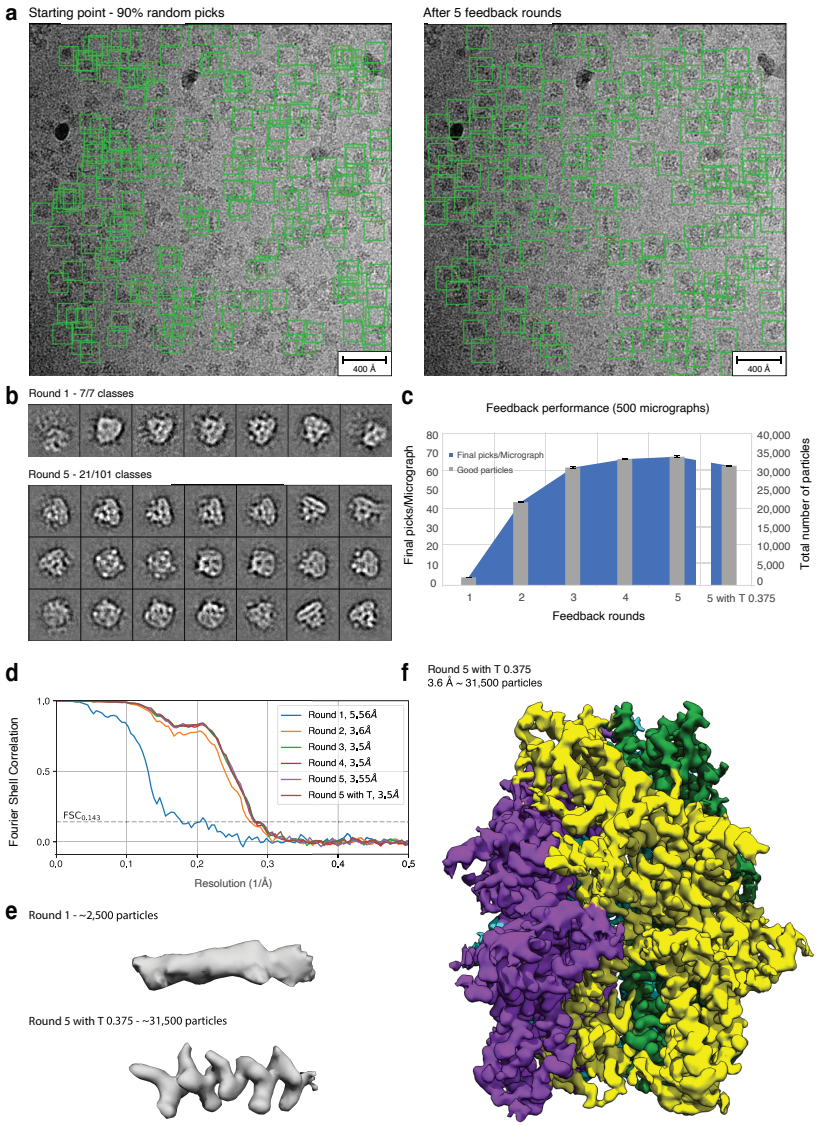

Figure 3.6: a To simulate low quality picking, only 10 % of the initial *crYOLO* picks were used while the remaining 90 % were re-positioned randomly (left). After the feedback loop *crYOLO* reliably picks the TRPC4 particles (right). Figure from [118]. **b** Total amount of 2D class averages produced in the first iteration of the feedback loop (top) and 21 representative averages produced in the final iteration of the feedback loop (bottom). **c** Progression of the number of particles labeled "kept" when applying the intermediate picking models of the feedback loop to a fixed subset of 500 micrographs. The curve flattens out in the last iterations, indicating the convergence of the feedback loop optimization. The values and errorbars represent the mean and standard deviation of the values from 10 independent runs (Table 5.25-5.31) **d** FSC curves of the individual 3D reconstructions from a representative run computed from particles labeled "kept" (also see **c**). **e** Representative α-helix (amino acids 518–535) illustrating the improvement of the density when using the final (bottom) compared to the initial (top) picking model. **f** 3D reconstruction of TRPC4 computed from 500 micrographs using the optimized picking model. Figures part **a** and **b** and caption taken and adapted from [118].

Discussion Overall, the picking performance improved during the course of the feedback loop without human intervention. While using the sabotaged *crYOLO* model merely 4.68 ”kept” particles per micrograph were identified. However, the number monotonically increased until in the fourth iteration 81.06 ”kept” particles per micrograph were selected. At the same time, the achieved resolution improved from 13.60 Å to 4.01 Å which can be explained by the increased absolute amount of ”kept” particles from 496 in the first iteration to 11 753 in the fourth. This indicates that the extracted particles represent real projections of the protein, since the resolution would not improve to about 4 Å.

In the fifth iteration, only 9 629 ”kept” particles were extracted, i.e., 74.64 ”kept” particles per micrograph, and a resolution of 4.08 Å was achieved. However, since the start of the feedback process is determined by the number of extracted particles and not by a fixed number of micrographs, the particles were extracted from 129 instead of 145 micrographs, from which the 129 micrographs are identical in both data sets. It can also be noted that the resolution only dropped slightly from 4.01 Å to 4.08 Å even though about 2 000 less particles, i.e., about 20 %, were used for the 3D refinement. This indicates, that overall more ”discarded” particles are extracted, but the ”kept” particles are of higher quality compared to those of the fourth iteration.

The evaluation of the individual models support the findings of the feedback loop iterations. After an increase in every value until the fourth iteration, the fifth iteration leads to the same resolution of (3.55 ± 0.03) Å. Therefore, the increase from 62 984 to 73 175 extracted particles did not lead to an improvement of the resolution, which can also be seen by the similar numbers of ”kept” particles of 31 016.70 ± 622.11 and 33 228.10 ± 57.98. Additionally, the resulting *crYOLO* model of the fifth feedback iteration resulted in comparable values as the input model to the fifth iteration indicating that no further optimization has happened.

Using the optimal picking confidence threshold of 0.357 , only 56 026.00 ± 0.00 instead of about 70 000 particles were extracted. However, the amount of ”kept” particles only slightly decreased to 31 467.00 ± 611.07 while yielding a similar resolution of (3.56 ± 0.03) Å. However, in total almost 15 000 particles less are extracted from the same 500 micrographs. Therefore, a result of the same quality could be obtained with about 80 % of the particles, and therefore only 80 % of the computational cost is required.

Using the confidence value of 0.1 after the fifth feedback iteration, the achieved resolution of (3.54 ± 0.04) Å is similar to the results of the general *crYOLO* model, (3.50 ± 0.03) Å. The smaller resolution value could be explained by the amount of ”kept” particles which is about 3 000 higher, however the total amount of extracted particles is about 85 % smaller.

Identifying sub-populations within a data set

Above, it was shown that the TranSPHIRE feedback loop is able to train a *crYOLO* model which learned to pick the TRPC4 sample present in the images to near-completion starting from a very poor picking performance. Therefore, a poorly behaving general model for *crYOLO* particle picking and a general model unaware of TRPC4 was used for *Cinderella* 2D class selection. However, it is also possible to utilize the TranSPHIRE feedback loop mechanism to select only a known sub-population of the data by providing a *Cinderella* model which is specifically trained to label only classes representing the ”kept” part of the data instead of a general one which can only distinguish between particle and contamination or noise. Therefore, a data set of a Tc holotoxin

was used which consists of the pore and the pre-pore state conformation of the protein, with a proportion of about 20 % and 80 %, respectively (Figure 3.7a). The aim of the feedback loop in this experiment was to target the pore state sub-population.

Feedback loop To target the pore state of the Tc holotoxin, a *Cinderella* model was trained with pre-existing classes of the pore state labeled as "kept" (318) and classes of the pre-pore state and contamination labeled as "discarded" (664). A general model was used as an initial model for *crYOLO*. As a picking threshold a liberal value of 0.1 was used while 2D class selection was performed with a liberal threshold of 0.1 to keep all particles and classes that might represent a protein.

The number of micrographs required to reach about 20 000 particles gradually increased from 128 in the first feedback iteration to 254 in the fifth (Table 3.5). Similarly, the number of total particles picked per micrograph continuously decreased from 139.83 to 74.93 from the first to the fifth feedback iteration. While, the yield of "kept" particles gradually increased from 13.73 % to 36.44 % from the first to the third feedback iteration, the number slightly decreased to 36.02 % in the fourth before increasing to 37.62 % in the fifth feedback iteration. The achieved resolution after the 3D refinement was 7.97 Å, 8.40 Å, 4.80 Å, 6.36 Å, and 4.75 Å in the first, second, third, fourth, and fifth feedback iteration, respectively.

The optimal picking confidence threshold for the model of the final fifth iteration was evaluated to be 0.194 by the *crYOLO* evaluation tool.

Table 3.5: Results of the TranSPHIRE feedback loop for the Tc holotoxin data set. Each feedback iteration started after about 20 000 particles were collected.

Feedback iteration	#Micrographs	#Particles total	#Particles total / Micrograph	#Particles "kept"
1 + To.1	128	17 898	139.83	2 458
2 + To.1	169	18 353	108.60	3 627
3 + To.1	224	18 656	83.29	6 798
4 + To.1	237	18 965	80.02	6 831
5 + To.1	254	19 032	74.93	7 160

Feedback iteration	#Particles "kept" / Micrograph	Particles "kept" / %	Resolution / Å
1 + To.1	19.20	13.73	7.97
2 + To.1	21.46	19.76	8.40
3 + To.1	30.35	36.44	4.80
4 + To.1	28.82	36.02	6.36
5 + To.1	28.19	37.62	4.75

Evaluation To evaluate the performance of the TranSPHIRE feedback loop a fixed subset of 500 micrographs was used from the data set (Table 3.6). The number of "kept" classes, "kept" particles,

"kept" picks per micrograph, and the resolution of the 3D reconstruction remained about the same at about 130, 13 000, 26, and 4.2 Å respectively. However, the total picks per micrograph monotonically decreased throughout the feedback loop from 124.71 ± 0.00 to 65.20 ± 0.00 which corresponds to an increase in the relative number of "kept" picks from (18.04 ± 0.79) % to (39.86 ± 1.56) %. Using the optimized picking confidence threshold evaluated to be 0.194, the relative amount of "kept" picks further increased to (45.89 ± 2.09) %.

Table 3.6: Results of the evaluation of the TranSPHIRE feedback loop on a subset of 500 micrographs of the Tc holotoxin data set for the individual feedback iterations. The listed values are the mean and standard deviation based on repeating the evaluation runs 10 times.

Feedback iteration	#Particles total	#Particles total / Micrograph	#Particles "kept"
ini + To.1	62 353.00 ± 0.00	124.71 ± 0.00	11 246.20 ± 492.47
1 + To.1	47 602.00 ± 0.00	95.20 ± 0.00	12 192.60 ± 718.05
2 + To.1	35 364.00 ± 0.00	70.73 ± 0.00	12 790.70 ± 584.05
3 + To.1	33 823.00 ± 0.00	67.65 ± 0.00	13 007.20 ± 415.72
4 + To.1	31 903.00 ± 0.00	63.81 ± 0.00	13 048.70 ± 73.90
5 + To.1	32 598.00 ± 0.00	65.20 ± 0.00	12 994.90 ± 509.31
5 + To.194	26 152.00 ± 0.00	52.30 ± 0.00	12 002.10 ± 545.64

Feedback iteration	#Particles "kept" / Micrograph	Particles "kept" / %	Resolution / Å
ini + To.1	22.49 ± 0.98	18.04 ± 0.79	4.26 ± 0.03
1 + To.1	24.39 ± 1.44	25.61 ± 1.51	4.23 ± 0.03
2 + To.1	25.58 ± 1.17	36.17 ± 1.65	4.20 ± 0.04
3 + To.1	26.01 ± 0.83	38.46 ± 1.23	4.25 ± 0.11
4 + To.1	26.10 ± 1.48	40.90 ± 2.32	4.22 ± 0.04
5 + To.1	25.99 ± 1.02	39.86 ± 1.56	4.19 ± 0.03
5 + To.194	24.00 ± 1.09	45.89 ± 2.09	4.25 ± 0.07

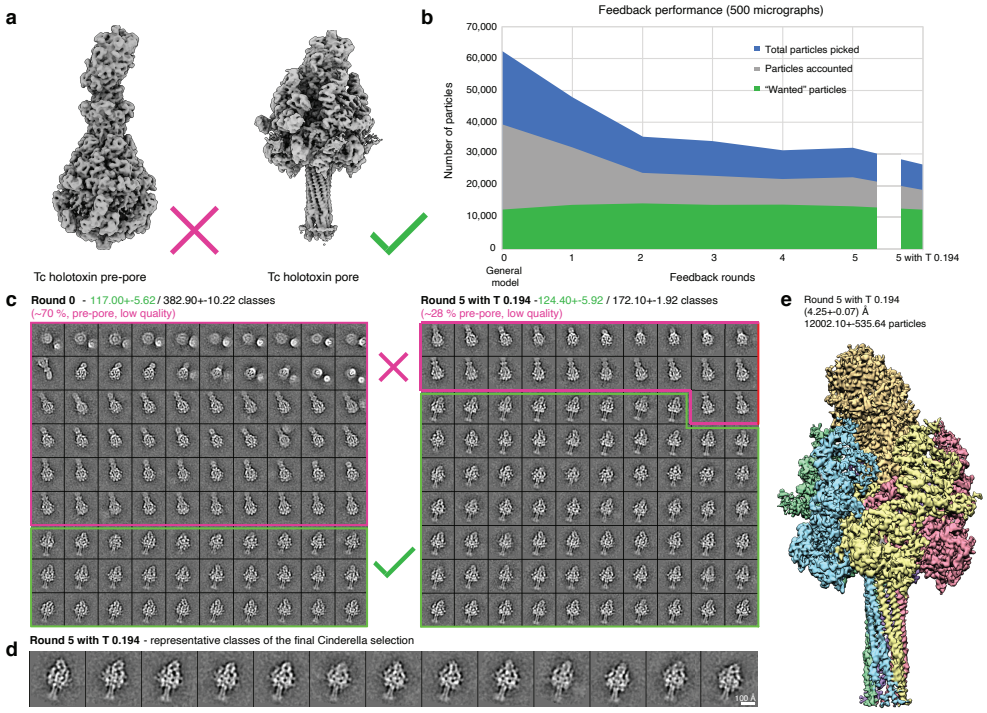

Figure 3.7: a The processed data set contains the Tc holotoxin in both the pre-pore state (left) and the more rare pore state (right). In this experiment, the pore state was specifically targeted. **b** Indication of the progression of the number of picked particles (blue), those accounted during 2D classification (gray) and particles labeled "kept", i.e., representing the pore state (green) when applying the intermediate picking models of the feedback loop to a fixed subset of 500 micrographs. Initial picking is dominated by pre-pore state particles. This overhead is reduced with each iteration, while the amount of picked pore state particle remains stable. **c** Representative 2D class averages depicting the decrease of "discarded" classes (pore state or low quality; marked magenta) from an initial 68 % in the first feedback iteration (left) to 26 % after the last feedback iteration (right). **d** Representative 2D class averages depicting the pore state as selected by *Cinderella* in the final iteration of the feedback loop. **e** 3D reconstruction of the Tc holotoxin pore state computed from 500 micrographs using the final optimized picking model. Figure parts **a**, **b**, **c**, and **d** and caption adapted from [118].

Discussion Within the feedback iterations, the picking performance improved from 19.20 "kept" particles per micrograph to about 30 "kept" particles per micrograph while decreasing the total number of picked particles per micrograph from 139.83 to about 80. At the same time, the yield of "kept" particles in relation to the total picked particles increased from 13.73 % to about 37 %. This indicates that the *crYOLO* model has successfully trained to focus on the pore state of the target protein while decreasing the picks of the pre-pore state. There is no obvious difference in numbers of "kept" particles between the thirds, fourth, and fifth iteration. Therefore, the decreased resolution of 6.36 Å could be a statistical outlier due to the overall small number of 6 831 "kept" particles used within the refinement.

Evaluating the performance on a fixed data set of 500 micrographs, similar observations are made compared to the results of the TranSPHIRE feedback loop. The number of ”kept” particles per micrograph is about 25 while the total number of particles per micrograph decreased from 124.71 ± 0.00 to about 65. This translates into a yield of about 40 % ”kept” particles from (18.04 ± 0.79) %. Since the ratio of pore to pre-pore particles is about 20 % to 80 %, initially the general *crYOLO* model picked the target pore state already almost to completion, but improved throughout the curse course of the feedback loop to focus more on the pore state. Additionally, the achieved resolution in every feedback round is at about 4.25 Å.

Using the optimized picking threshold of 0.194, the ratio of ”kept” particles further increased to (45.89 ± 2.09) %. In comparison with the results of the general model only 26 152.00 ± 0.00 total particles were extracted instead of 62 353.00 ± 0.00, reducing the computational cost of subsequent processing steps to about 40 %.

Automated data optimization of actomyosin

Filamentous proteins are of continuous character often span over the whole field of view of the micrograph. Therefore, filaments are traced instead of picked and the particles are extracted along the helical axis while filament crossings and contamination are avoided. TranSPHIRE can not only perform on-the-fly automated processing for filamentous samples with the help of the *crYOLO* filament mode, but is also able to learn how to pick yet unknown data with the help of the TranSPHIRE feedback loop. To demonstrate the processing of as yet unknown filamentous data, an actomyosin data set is used.

Feedback loop Since there is no general *crYOLO* model for filaments available, a new model was trained based on previously collected bare filamentous actin data sets, which look different from the actomyosin complex. The same holds true for a general model used in Cinderella, and therefore a new *Cinderella* model was trained based on the 2D classification results of the first TranSPHIRE feedback loop iteration. As a picking threshold a liberal value of 0.1 was used while 2D class selection was performed with a liberal threshold of 0.1 to keep all particles and classes that might represent a protein.

The number of micrographs required to reach about 20 000 particles gradually decreased from 121 in the first feedback iteration to 18 in the fifth (Table 3.7). Similarly, the number of total particles picked per micrograph continuously increased from 165.52 to 1 189.89 from the first to the fifth feedback iteration. The yield of ”kept” particles gradually decreased from 82.12 % to 32.36 % from the first to the fifth feedback iteration. The achieved resolution after the 3D refinement was 5.88 Å, 5.12 Å, 5.88 Å, 4.98 Å, and 8.74 Å in the first, second, third, fourth, and fifth feedback iteration, respectively.

Table 3.7: Results of the TranSPHIRE feedback loop for the actomyosin data set. Each feedback iteration started after about 20 000 particles were collected.

Feedback iteration	#Micrographs	#Particles total	#Particles total / Micrograph	#Particles "kept"
1 + To.1	121	20 028	165.52	16 447
2 + To.1	55	20 480	372.36	16 504
3 + To.1	41	20 029	488.51	12 410
4 + To.1	36	20 479	568.86	15 129
5 + To.1	18	21 418	1 189.89	6 930

Feedback iteration	#Particles "kept" / Micrograph	Particles "kept" / %	Resolution / Å
1 + To.1	135.93	82.12	5.88
2 + To.1	300.07	80.59	5.12
3 + To.1	302.68	61.96	5.88
4 + To.1	420.25	73.88	4.98
5 + To.1	385.00	32.36	8.74

Evaluation To evaluate the performance of the TranSPHIRE feedback loop a fixed subset of 100 micrographs was used from the data set (Table 3.8). Initially, the general *crYOLO* model for actin filaments identified 14 307.00 ± 0.00 particles, i.e., 143.07 ± 0.00 particles per micrograph, from which 12 288.20 ± 352.82 were marked as "kept", i.e., 122.88 ± 3.53 "kept" particles per micrograph. Throughout the feedback loop the amount of "kept" particles monotonically increases to reach 53 735.50 ± 1 006.56 "kept"particles, i.e., 537.36 ± 10.07 "kept" particles per micrograph, in the fifth feedback iteration and this values stays about the same with 53 356.90 ± 907.09 in the sixth feedback iteration. On the other hand, the total amount of extracted particles increased from 14 307.00 ± 0.00 in the first to 109 973.00 ± 0.00 in the fifth iteration and also this value is in the similar range with 107 343.00 ± 0.00 in the sixth iteration. Therefore, the overall yield decreased from (85.89 ± 2.47) % to about 50 %. The resolution of the first feedback iteration was (6.91 ± 0.58) Å which improved to (4.55 ± 0.18) Å and (4.52 ± 0.12) Å in the fifth and sixth feedback iteration, respectively.

Using the more conservative picking confidence threshold of 0.3, 38 686.50 ± 454.78 "kept" particles from originally 51 483.00 ± 0.00 total particles were identified. Hence, the yield increased to (75.14 ± 0.88) % resulting in a resolution of (4.78 ± 0.22) Å.

Table 3.8: Results of the evaluation of the TranSPHIRE feedback loop on a subset of 100 micrographs of the actomyosin data set for the individual feedback iterations. The listed values are the mean and standard deviation based on repeating the evaluation runs 10 times.

Feedback iteration	#Particles total	#Particles total / Micrograph	#Particles "kept"
ini + To.1	14 307.00 ± 0.00	143.07 ± 0.00	12 288.20 ± 352.82
1 + To.1	35 792.00 ± 0.00	357.92 ± 0.00	30 008.20 ± 235.02
2 + To.1	55 145.00 ± 0.00	551.45 ± 0.00	43 208.90 ± 33.00
3 + To.1	63 917.00 ± 0.00	639.17 ± 0.00	46 089.40 ± 481.48
4 + To.1	109 973.00 ± 0.00	1 099.73 ± 0.00	53 735.50 ± 1 006.56
5 + To.1	107 343.00 ± 0.00	1 073.43 ± 0.00	53 356.90 ± 907.09
5 + To.3	51 483.00 ± 0.00	514.83 ± 0.00	38 686.50 ± 454.78

Feedback iteration	#Particles "kept" / Micrograph	Particles "kept" / %	Resolution / Å
ini + To.1	122.88 ± 3.53	85.89 ± 2.47	6.91 ± 0.58
1 + To.1	300.08 ± 2.35	83.84 ± 0.66	4.76 ± 0.34
2 + To.1	432.09 ± 0.33	78.36 ± 0.06	4.49 ± 0.15
3 + To.1	460.89 ± 4.81	72.11 ± 0.75	4.43 ± 0.11
4 + To.1	537.36 ± 10.07	48.86 ± 0.92	4.55 ± 0.18
5 + To.1	533.57 ± 9.07	49.71 ± 0.85	4.52 ± 0.12
5 + To.3	386.86 ± 4.55	75.14 ± 0.88	4.78 ± 0.22

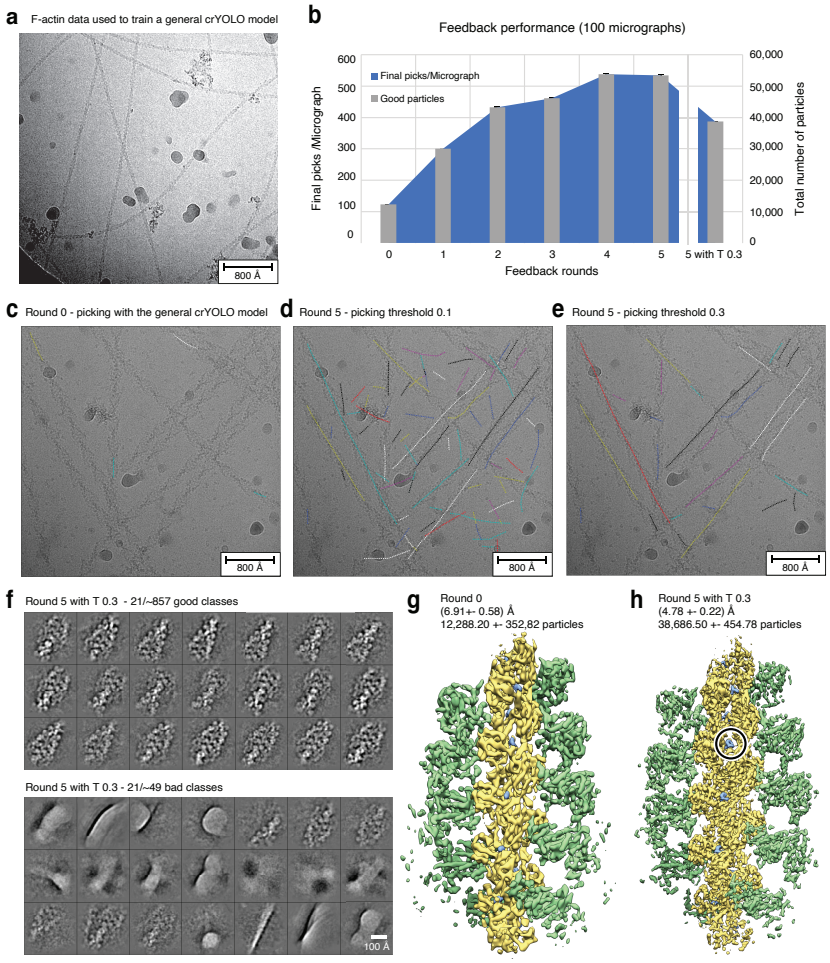

Figure 3.8: a Representative micrograph of the F-actin data used to train *crYOLO*. **b** Progression of the number of "kept" particles per micrograph (blue) and in total (gray) when applying the intermediate picking models of the feedback loop to a fixed subset of 100 micrographs. The dipping curve at the end indicates the desired loss of low-quality picks that are excluded when a higher picking threshold (0.3) is used. **c** Representative micrograph of the actomyosin complex highlighting the weak initial picking results when using the *crYOLO* model trained on F-actin data (see a). **d** Particle picking performance on the same micrograph using the final picking model. While filaments are now traced much more effectively, the model also picks unwanted filament crossings and contamination. **e** Increasing the picking threshold from 0.1 to the default value of 0.3 minimizes the amount of false positive picks, while maintaining the desired filament traces. **f** Representative 2D class averages labeled "kept" (top) and "discarded" (bottom) by *Cinderella* based on 100 micrographs and using the final model for picking. **g** 3D reconstruction of the actomyosin complex computed from 100 micrographs using the initial picking model. **h** 3D reconstruction computed from the same 100 micrographs using the final optimized picking model. The resolution is sufficient to verify the binding of a ligand (circled). Figure parts **a**, **c**, **d**, **e**, and **f** and caption adapted from [118].

Discussion The increased numbers in ”kept” particles per micrograph in combination with the increased resolution throughout the feedback loop until the fourth iteration indicates that the *crYOLO* model learns to pick the particles of interest. In the fifth iteration, the picking performance of ”kept” particles per micrograph did decrease by about 10 %, but the total number of picked particles per micrograph almost doubled. Therefore, the number of ”kept” particles and used micrographs decreased to about 50 % which results in the comparably worse resolution of 8.74 Å. The behavior of the fifth iteration on the other hand indicates that the model learns to pick overall more particles on the micrographs, but mainly increases the picking of ”discarded” particles.

Evaluation of the resulting models show the same behavior in numbers as the TranSPHIRE feedback loop iterations. The number of ”kept” particles per micrograph, as well as the achieved resolution, only changes slightly after the second feedback iteration. In agreement with the feedback loop iterations, the number of picked particles per micrograph almost doubles in the fourth feedback iteration while the number of ”kept” particles per micrograph further increases. This strengthens the hypothesis that the model picks the micrographs more to completion with the drawback of an increased false-positive rate, i.e., picking particles which actually do not represent particles.

Changing the picking confidence threshold to a more conservative value of 0.3, the rate of false-positive picks greatly decreases from almost 50 % to about 25 %. Additionally, a visual inspection of the traced filaments shows that the amount of false positive picks minimizes while maintaining the desired filament trace. However, the achieved resolution as well as the amount of ”kept” particles per micrograph decreased at the same time. In combination with the resolution peak in the third feedback iteration it could be concluded that the *crYOLO* model did not further improve in later iterations. This, however, would need a larger data set to be analyzed on to rule out resolution limitations due to the number of ”kept” particles used.

3.2.4 Discussion

TranSPHIRE automates the initial processing of the data set live during data acquisition by providing common interfaces for different available processing tools. The provided GUI not only offers access to the individual settings, but allows the user to choose between alternative programs for the individual tasks. Due to the recent development of reliable software in combination with user-focused interfaces, cryo-EM became more accessible and interesting for a larger amount of research groups. Therefore, other software for automated data processing such as *RELION* [113], *Focus* [6], *Scipion* [108], *WARP* [123], *cryoSPARC* [96], *cryoFlare* [111] and *Appion* [68] also provide GUI based user input. However, each program has its own strategy how and which settings are exposed to the user. The individual programs in *WARP* [123] and *cryoFlare* [111] have only limited options and those are exposed to the user in the GUI. *RELION* [113] offers basic settings in its *relion_it.py* GUI to setup the pipeline [63], but requires the user to change their *Python* options files in order to include advanced options of the individual programs [102]. The tools *Focus* [6], *Scipion* [108], *cryoSPARC* [96], and *Appion* [68] categorize the options into basic and advanced to help beginner users identify at which options to look at. In TranSPHIRE the three categories **Main**, **Advanced**, and **Rare** are used for the parameters and each category can be manually assigned by an experienced user based on the needs of the data acquisition environment. This allows beginner and intermediate users to setup a session without being overwhelmed by potentially many available options. However,

expert users can access every option possible to tailor the processing pipeline to their individual needs. All of the presented solutions group their settings by their individual tasks and arrange them according to the order of their execution.

Which programs are available for execution and how those are integrated into the individual pipelines and GUIs is also dependent on the software. *WARP* [123] integrates only self-developed tools which are only available from within the *WARP* pipeline. *RELION* [113] and *cryoSPARC* [96] mainly integrate their self-developed tools but also provide wrappers to some selected externally developed tools. Additionally, in the *RELION* [113] pipeline own tools can be integrated by adapting the *relion_it.py Python* file before execution. *Focus* [6], *Scipion* [108], *cryoFlare* [111], and *Appion* [68] require to write own modules for the individual externally developed tools, but provide templates as a starting point. TranSPHIRE follows the idea of including externally developed and well established software into its pipeline, and it is possible to include own programs by adapting the *Python* source code. Since the knowledge of *Python* can be a hurdle to fully adapt to the needs of the users. Therefore, in the future it could be beneficial to make external software available to TranSPHIRE via human-readable data-serialization file formats such as YAML [135] or JSON [60]. In this way, new software dependencies could be bundled to ship them with the TranSPHIRE installation or could be shared within the TranSPHIRE community.

Another important purpose next to guiding the user through the setup of the pipeline is the visualization of the results to allow for rapid adjustments of the data acquisition. *RELION*, *Scipion*, and *Appion* focus on the execution of the pipeline and require the user to manually decide which data are worth looking at and to generate the respective plots. *cryoFlare* and *Focus* are explicitly designed to run live at the microscope immediately present statistics about the data set front-and-center in their GUI. *WARP* and *cryoSPARC* go one step further and additionally allow to set thresholds for ”kept” and ”discarded” micrographs right within the plots itself. *Scipion*, *Appion*, *cryoFlare*, *Focus*, *WARP*, and *cryoSPARC* additionally present the individual results in a table format. Therefore, a combination between manual and automatic micrograph selection is possible and allows for the export of specific subsets of the data. TranSPHIRE falls into the category of *cryoFlare* and *Focus* and presents the important metrics to the user once the **Start** button is pressed. The data are presented in a timely manner, as a histogram over the entire data set, or in a per micrograph summary. However, TranSPHIRE is not yet designed to select or deselect micrographs interactively, but rather to provide a fast overview over the ongoing data collection. Therefore, there is no table of micrographs focusing on the individual results, and also no interactive threshold selection within the plots is available. Nevertheless, the advanced micrograph selection functionality could be added to further increase the use cases of the TranSPHIRE GUI.

To which extend the processing pipeline processes the individual micrographs is highly dependent on the software used. For example, *WARP* executes drift correction, CTF estimation and particle picking. *cryoFlare*, *Focus*, *RELION*, *Scipion*, and *Appion* offer a completely program independent linear pipeline creation and are therefore not limited to specific pipeline steps. *cryoSPARC* offers the processing steps of drift correction, CTF estimation, 2D classification, and 3D refinement. Similar to *cryoSPARC*, TranSPHIRE offers the processing steps from drift correction to 3D refinement. However, more flexibility is offered to the user due to exchangeable programs in the drift correction and CTF estimation step.

The processing of incoming micrographs can be categorized into linear and non-linear workflows, and all the presented programs apart from *cryoFlare* [111] implement a linear workflow. Linear

workflows process the tasks one after the other, while each task in non-linear workflows can either run in sequence or in parallel relative to other tasks. Therefore, non-linear workflows have the potential to use the limited available resources more efficiently, because the different tasks need different amount of resources and time to complete. In TranSPHIRE the workflow is dynamic and of non-linear character. Especially utility tasks such as copying data to different locations or file compression are often limited by the bandwidth between the involved machines. To avoid a delay of the actual processing *inter alia* those utility tasks run in parallel to the data collection with limited assigned resources.

The most important non-linear building block of the TranSPHIRE pipeline is the TranSPHIRE feedback loop. After 2D class selection the ”kept” particles are used to train a new *crYOLO* [131] model. Afterwards, the particle picking and all subsequent tasks are re-run on all the data using the new model. Previously, the Cianfrocco lab showed that automated processing of cryo-EM data sets can greatly benefit from deep-learning based 2D class selection [72] Additionally, using the results of the 2D class selection to re-train a new model for particle picking was performed by the Liu lab [77]. For those approaches to work, it is essential that the 2D classification step results in a homogeneous class assignment. Since the *ISAC* [136] algorithm is known for its robustness and homogeneous class assignments at the cost of computational expense compared to other modern maximum likelihood based approaches, the less resource demanding GPU version *GPU ISAC* [44] was chosen as the 2D classification program. To show the capabilities of the TranSPHIRE feedback loop, its performance was evaluated with three different real-world applications: Processing of a previously unknown data set, selection of a known subset within the data, and automated processing of a filamentous sample.

The first scenario demonstrated the processing of previously unknown data based on a TRPC4 data set. The micrographs were almost picked to completion even though the initial particle picking had merely a yield of 2.54 % ”kept” particles. Data sets in cryo-EM of the same sample show not only variations across different microscopes and used detectors, but also at the same microscope and detectors due to different sample preparation techniques and microscope optical settings. Therefore, even a specifically trained model of the sample based on previously collected data sets cannot guarantee a picking result with an optimal yield of ”kept” particles. However, those picking issues could be automatically resolved with the help of the TranSPHIRE feedback loop and enable a more reliable analysis of the data set based on 2D classifications and 3D refinements. The possibility that the TranSPHIRE feedback loop could adapt to various, even as yet unknown, different data sets could pave the way towards fully automated high-throughput cryo-EM processing.

The second scenario illustrated the automatic identification of a known subset within the data set based on a Tc holotoxin data set consisting of particles representing the pore and pre-pore state. Throughout the TranSPHIRE feedback loop the relative amount of ”kept” pore state particles increased from (18.04 ± 0.79) % to (45.89 ± 2.09) %, reducing the amount of false-positive picks, and speeding up subsequent processing steps. Therefore, the TranSPHIRE feedback loop could be used to target different sub populations of the data.

Historically, proteins like TMV were added to the target sample to improve the homogeneity of the ice layer, and therefore the distribution of the target sample within the ice layer. Additionally, the added protein can be processed independently to verify the overall quality of the data collection and rule out related issues during processing of the actual sample [21, 115].

The third scenario demonstrated the automatic processing of filamentous samples based on an

actomyosin data set. Filamentous processing remains a challenging task for automated particle picking, and therefore also automated 3D processing. However, the *crYOLO* [131] particle picker released a filament mode which works with specifically trained *crYOLO* models, but no general model for filaments is yet available [132]. Therefore, the provided *crYOLO* model for the TranSPHIRE feedback loop was specifically trained on a F-actin data set, but the TranSPHIRE feedback loop was able to correctly identify the actomyosin protein as the sample of interest. However, it was necessary to train a specific *Cinderella* [11] model for 2D class selection after the very first 2D classification during the TranSPHIRE feedback loop. That the TranSPHIRE feedback loop was able to identify the actomyosin based on a picking model that does not know about actomyosin is of special importance, as it shows the potential of a general model for filamentous samples for *crYOLO* and *Cinderella*. To my knowledge, a general approach for automated processing of filamentous samples did previously not exist. Therefore, TranSPHIRE not only shows the potential for high-throughput target screenings of known filamentous samples, but also for the initial processing of unknown filamentous samples during data acquisition.

To improve even further TranSPHIRE and the TranSPHIRE feedback loop the re-training procedure of the *crYOLO* model could be optimized. Currently, the model is trained from scratch without any prior knowledge injected. However, typical contamination caused by the sample preparation could be ignored and lead to impure picking results, and it could be beneficial to add examples of pure contamination to the training data. Additionally, *crYOLO* offers an experimental fine-tune mode that uses less computational resources, runs faster, and should reduce the risk of overfitting, as only the last few layers of the network are trained. It would be interesting to test this mode for the TranSPHIRE feedback loop to further boost the picking performance, especially for sparse-picking situations.

TranSPHIRE streamlines all major pre-processing steps live during data acquisition and the TranSPHIRE feedback loop is able to improve particle picking in various situations in a fully automated manner. Since more scientist are entering the field of cryo-EM to complement their research, rather than having the structure of the sample as their primary aim, automated data processing that does not require expert knowledge to reach high-resolution reconstructions gains importance. Streamlining the pre-processing to run in an automated way has been available for many years, but manual input was necessary at key points such as particle picking and 2D class selection. With the help of the TranSPHIRE feedback loop the optimization of those two steps are additionally automated and, depending on the use-case, require very little to no user input. Therefore, TranSPHIRE helps beginners to reach high-resolution 3D reconstructions in an automated manner, but due to its transparent input-settings design allows expert users to tackle even the most challenging samples.

After the automation and optimization of the processing of the data collected at the microscope, the next step would be to automatically optimize the data collection itself by feeding information into the data collection software such as *EPU* [30], *Serial EM* [75], or *Leginon* [120]. For example, information about high drift values in the last micrographs could be used to automatically increase the waiting time between movement of the stage and acquiring an image. Additionally, the desired defocus values set could be compared to the results of the actual data to automatically optimize the distribution of defocus values. Furthermore, grids or holes can be skipped if the particle distribution is very low to get the most out of the data acquisition. In this way, the collection of unwanted data would be avoided and more useful data be collected.

4 Conclusion

4.1 Automated processing with TranSPHIRE

In the course of this thesis, I presented the TranSPHIRE [118] pipeline. The TranSPHIRE pipeline executes the pre-processing steps of the SPA cryo-EM pipeline in an automated way with the focus on the parallel execution of the individual programs. Recent advancements in detectors in combination with advanced automated data collection strategies such as AFIS or FFI allows for the collection of up to 600 micrographs/h [1, 9, 32]. However, how many micrographs are exactly needed to reach a specific resolution to answer the underlying biological question is dependent *inter alia* on the number of protein projections per micrograph, the stability of the microscope, the microscope settings used during data collection, the homogeneity of the sample, and the thickness of the thin ice layer on the grid. Therefore, automatic processing is of increasing importance to get immediate feedback during data collection about the behavior and quality of the data set. The feedback not only can potentially save a lot of wasted disk space, hence computational resources, but additionally optimize the time spent at the microscope.

Over the last years, software like *WARP* [123], *Focus* [6], *RELION* [113], *Scipion* [108], *Appion* [68], *cryoSPARC* [96], and *cryoFlare* [111] started to provide an easy to access interface to chain together different parts of the SPA pipeline for sequential or parallel execution. They present the intermediate processing results to allow for an optimized data collection. While *Focus* [6], *Scipion* [108], *Appion* [68], and *cryoFlare* [111] focus on the interaction of arbitrary software available in the field, *WARP* [123], *RELION* [113], and *cryoSPARC* [96] mainly provide tools specifically designed from their developers. TranSPHIRE is in between both categories providing interfaces to allow arbitrary software from the field to interact with each other, but on the other hand implementing unique features such as the presented novel TranSPHIRE feedback loop. Additionally, the focus is on the usability for beginner users to get the data processing started, as well as the advanced user that needs to tweak certain settings to get the most out of the data set from the very beginning. Therefore, the TranSPHIRE interface allows for the setup of setting templates and the assignment of each setting to one of the three categories *Main*, *Advanced*, and *Rare* for easy accessibility.

The TranSPHIRE feedback loop is able to influence earlier steps of the pipeline with knowledge from later steps to improve the results of the individual steps; specifically the particle picking result is optimized based on the results of the 2D classification. The Cianfrocco lab already showed that the resolution of a data set can be improved automatically by using a deep-learning based 2D class selection tool [72]. However, they did not use the ”kept” subset of particles to improve particle picking. For the feedback loop to succeed, the two crucial steps are the 2D classification, which needs to reliably cluster the provided particles, and the 2D class selection, which splits the resulting 2D class averages into ”kept” and ”discarded”. Since the *ISAC* [136] algorithm is known for its ability to form stable and reliable class averages with homogeneous members, its GPU version

GPU ISAC [44] was chosen for 2D classification. For 2D class selection, the deep-learning based tool *Cinderella* [11] was used, which can be trained in various ways to select "kept" and "discarded" subsets of the data.

The Liu lab implemented a different strategy for the feedback loop approach using 2D classification in *RELION* [113] in combination with their metric $\%_{\text{Res}}$ to get the desired subset of "kept" 2D classes [77]. The metric $\%_{\text{Res}}$ is defined as the ratio of the percentage class distribution, i.e., the ratio of assigned particles of a class and the total number of particles, and the respective resolution of the class. However, those two metrics are highly correlated, because the resolution of a class is dependent on the number of members [101]. Additionally, not only can large classes representing protein result in high resolution, but also classes representing noise and contamination. Therefore, it is possible that a deep-learning based classifier like *Cinderella* [11] or *2DAssess* [72] could yield more consistent and homogeneous results due to them focusing on features in the class averages belonging to protein rather than statistics of the outcome.

I demonstrated three difference scenarios of the TranSPHIRE feedback loop that represent common use cases in cryo-EM.

The first scenario showed the ability of the feedback loop to improve the particle picking performance on an as yet unknown data set of a TRPC4 sample [129]. Furthermore, the initial picks were sabotaged to only yield a very small number number of particles per micrograph for training a new *crYOLO* [131] model. Nevertheless, the picking performance of *crYOLO* [131] could be improved to a point where the target protein has been picked to completion on the micrographs. This led to a resolution of about 3.5 Å based on a data set containing 500 micrographs (Figure 3.6).

In the second scenario the particle picking was trained to pick only a specific subset of a Tc holotoxin [105] data set. The data set consists of the common pre-pore state and the rare pore state in a 5:1 ratio. To redirect the particle picking to specifically target the pore state, a *Cinderella* [11] model had been trained prior execution which targets classes representing the pore state and discards not only noise and contamination, but also high-resolution classes showing the pre-pore state. By injecting this knowledge into the feedback loop the resulting *crYOLO* [131] model showed a slight increase of pore state particles, and the pre-pore state particles were greatly reduced. Therefore, the overall computational cost could be reduced to about 40 % while a resolution of 4.2 Å could be obtained from 500 micrographs (Figure 3.7).

Thirdly, an actomyosin sample [93] was used to demonstrate the ability of the TranSPHIRE feedback loop to also improve the picking performance of filamentous samples. A general model for *crYOLO* [131] specifically trained on only pure F-actin data sets was used as input for the initial particle picking. The visual appearance of F-actin and actomyosin is fundamentally different, and therefore the actomyosin data can be considered unknown to the trained *crYOLO* [131] model. Since there was also no specific *Cinderella* [11] model available the feedback loop was stopped after the first 2D classification to train a specific model based on the obtained class averages. Even though the initial *crYOLO* [131] model did not know about actomyosin and behaved accordingly, the model improved throughout the course of the feedback loop so that a final resolution of about 4.5 Å could be obtained from 100 micrographs (Figure 3.8).

In summary, TranSPHIRE offers an automatic processing pipeline that allows beginner users as well as advanced users to monitor the progress and behavior of an ongoing data collection. Therefore, upcoming issues can be tackled as soon as they arise enabling the acquisition of high quality data sets. Additionally, the TranSPHIRE feedback loop is a tool to automatically improve

the particle picking performance on a small subset of the data to allow for an efficient later use of human and computational resources. In combination with an on-the-fly 3D refinement it is possible to identify the conformational state of the protein or the binding of small molecules, if a small molecule free reference structure is already available. Since this is performed live during data collection, TranSPHIRE enables high-throughput screenings for different buffer conditions or the binding of ligands.

The combination of *crYOLO* [131] for particle picking, *GPU ISAC* [44] for 2D classification, and *Cinderella* [11] for 2D class selection is working due to their robustness, speed, and overall usability. Recently, the authors of the software package *RELION* [113] presented their new approach for 2D classification which they claim to result in more homogeneous and smaller classes compared to its predecessor [63]. Therefore, it could be interesting to check its performance within the TranSPHIRE feedback loop.

The TranSPHIRE pipeline currently automates the data processing, optimizes the yield of "kept" particles from the collected data, and reports possible problems with the data acquisition parameters, e.g., defocus value discrepancies, high specimen drift, few particles per micrograph, or phase shift values diverging from 90°, live to the user. In the future it would be ideal if the knowledge about potential issues could be directly communicated with the data acquisition software. Thus, the collection of unusable data could be reduced to a minimum further optimizing the time spent for each data set at the microscope. While open-source software like *Serial EM* [75] and *Leginon* [120] could be adapted to provide an interface for the communication of specific problems, commercial products like *EPU* [30] would need to provide an interface for this kind of communication. Optimization of the data collection setup based on the results of the collected data could lead to a fully automated data acquisition workflow accessible to every user independent of their experience and training.

4.2 Processing of filaments

While the protein in the SPA has a distinct shape with defined borders, a helical filament consists of a continuous repeat of the same subunit. To enable the processing of filaments in the *SPHIRE* [79] package, I presented a filamentous SPA 3D refinement approach available as *sp_meridien_alpha.py* without the usage of a helical symmetry based on processing strategies developed in the Raunser lab [78].

Typically, modern 3D refinement strategies such as *Helical RELION* [50], *SPIDER* [38], *SPARX* [57], *FREALIGN* [119], *FREALIX* [107], *SPRING* [17], and *cryoSPARC* [96] are based on the IHRSR [23] approach, which estimates and applies the helical symmetry parameters of the specimen to the result of each refinement iteration. However, calculating the helical symmetry from low-resolution structures can lead to model bias, wrongly estimated 3D structures, and incorrectly estimated helical symmetry parameters [91, 25]. In the *sp_meridien_alpha.py* implementation, instead of guiding the 3D refinement by applying an estimated helical symmetry, constraints stemming from the filamentous character of the sample are used. These constraints include similar out-of-plane rotation angles and in-plane rotation angles for neighboring particles, because particles are segmented and extracted with a certain overlap from the same filament. Therefore, information such as the direct neighbors of the particles, the in-plane rotation angle within the micrograph, and

the parent filament affiliation are saved in the metadata of each particle. During the 3D refinement, this information is used to identify particles which do not follow the filamentous character of the specimen. Additionally, modifications to the reference angle generation method, the background noise estimation, and shift limitations parallel to the helical axis of the specimen were implemented in the 3D refinement as well as the 2D classification programs *sp_isac2.py* [79] and *GPU ISAC* [44].

To show the capability of *sp_meridien_alpha.py* to reach high-resolution 3D reconstructions an actomyosin data set [93] and a TMV data set [40] were processed. The actomyosin data set reached a nominal resolution of (4.40 ± 0.20) Å using 45 297 from 97 micrographs. Using the unmodified version of *sp_meridien.py* [79], the nominal resolution was about the same at (4.47 ± 0.02) Å. However a visual inspection of the inner area of the protein showed lower resolved features compared to *sp_meridien_alpha.py*. For the TMV data set a nominal resolution of (4.37 ± 0.08) Å could be achieved using 30 000 particles, i.e., 30 000 asymmetric units, from 14 micrographs. *Helical RELION* [50] achieved a resolution of 4.1 Å from the same data set using 1 761 particles and utilizing the helical symmetry of the data set in the 3D refinement. The difference in resolution is expected, as TMV is a very rigid protein and its helical symmetry leat to effectively using about $1\,761 \times 30 \approx 60\,000$ asymmetric units in the refinement, as each box contained 30 asymmetric units.

All in all, *sp_meridien_alpha.py* allows for the processing of high-resolution structures within the *SPHIRE* [79] software package. On the one hand, using a SPA approach requires overall more data to be collected, because each particle contributes only one asymmetric unit. On the other hand, the implemented strategy does not require information about helical symmetry parameters and hence reduces the risk of model bias. This can be especially advantageous for flexible filaments that are bent or show local differences in their global helical symmetry.

Bibliography

1. *Advances in Single Particle Analysis Data Acquisition*. 2019.
   https://assets.thermofisher.com/TFS-Assets/MSD/posters/MM2019-poster-advances-SPA-data-acquisition.pdf visited on 2021-04-28
2. J.-P. Armache and Y. Cheng. "Single-particle cryo-EM: beyond the resolution". *National Science Review* 6:5, 2019, pages 864–866.
   DOI: 10.1093/nsr/nwz127. https://doi.org/10.1093/nsr/nwz127
3. A. Belyy et al. "Mechanism of actin-dependent activation of nucleotidyl cyclase toxins from bacterial human pathogens". *Nature Communications* 12:1, 2021.
   DOI: 10.1038/s41467-021-26889-2. https://doi.org/10.1038/s41467-021-26889-2
4. A. Belyy et al. "Structure of the Lifeact–F-actin complex". *PLOS Biology* 18:11, 2020. Ed. by C. A. Parent, e3000925.
   DOI: 10.1371/journal.pbio.3000925. https://doi.org/10.1371/journal.pbio.3000925
5. T. Bepler et al. "Positive-unlabeled convolutional neural networks for particle picking in cryo-electron micrographs". *Nature Methods* 16:11, 2019, pages 1153–1160.
   DOI: 10.1038/s41592-019-0575-8. https://doi.org/10.1038/s41592-019-0575-8
6. N. Biyani et al. "Focus: The interface between data collection and data processing in cryo-EM". *Journal of Structural Biology* 198:2, 2017, pages 124–133.
   DOI: 10.1016/j.jsb.2017.03.007. https://doi.org/10.1016/j.jsb.2017.03.007
7. R. Bracewell. "Strip Integration in Radio Astronomy". *Australian Journal of Physics* 9:2, 1956, page 198.
   DOI: 10.1071/ph560198. https://doi.org/10.1071/ph560198
8. A. F. Brilot et al. "Beam-induced motion of vitrified specimen on holey carbon film". *Journal of Structural Biology* 177:3, 2012, pages 630–637.
   DOI: 10.1016/j.jsb.2012.02.003. https://doi.org/10.1016/j.jsb.2012.02.003
9. J. N. Cash et al. "High-resolution cryo-EM using beam-image shift at 200 keV", 2020.
   DOI: 10.1101/2020.01.21.914507. https://doi.org/10.1101/2020.01.21.914507
10. A. Cavalli et al. "Protein structure determination from NMR chemical shifts". *Proceedings of the National Academy of Sciences* 104:23, 2007, pages 9615–9620.
    DOI: 10.1073/pnas.0610313104. https://doi.org/10.1073/pnas.0610313104
11. *Cinderella*. 2021.
    https://sphire.mpg.de/wiki/doku.php?id=auto_2d_class_selection visited on 2021-04-18
12. *cisTEM*. 2021.
    https://cistem.org/ visited on 2021-04-18

13. T. Q. Company. *QT*. Version 5.9.7. 2021.
https://www.qt.io/

14. R. A. Crowther, D. J. DeRosier, and A. Klug. "The reconstruction of a three-dimensional structure from projections and its application to electron microscopy". *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 317:1530, 1970, pages 319–340.
DOI: 10.1098/rspa.1970.0119. https://doi.org/10.1098/rspa.1970.0119

15. *CRYO ARM™ 300 II (JEM-3300) Field Emission Cryo-Electron Microscope*. 2022.
https://www.jeol.co.jp/en/products/detail/JEM-3300.html visited on 2022-05-06

16. R. Danev and W. Baumeister. "Cryo-EM single particle analysis with the Volta phase plate". *eLife* 5, 2016.
DOI: 10.7554/elife.13046. https://doi.org/10.7554/elife.13046

17. A. Desfosses et al. "SPRING – An image processing package for single-particle based helical reconstruction from electron cryomicrographs". *Journal of Structural Biology* 185:1, 2014, pages 15–26.
DOI: 10.1016/j.jsb.2013.11.003. https://doi.org/10.1016/j.jsb.2013.11.003

18. A. Dhillon and G. K. Verma. "Convolutional neural network: a review of models, methodologies and applications to object detection". *Progress in Artificial Intelligence* 9:2, 2019, pages 85–112.
DOI: 10.1007/s13748-019-00203-0. https://doi.org/10.1007/s13748-019-00203-0

19. C. Ding et al. "Extending resolution of scanning optical microscopy beyond the Abbe limit through the assistance of InSb thin layers". *Optics letters* 41, 7 2016, pages 1550–1553

20. Direct Electron. *DE-Series*. 2010.
http://www.directelectron.com/products/de-series visited on 2021-04-18

21. I. Drulyte et al. "Approaches to altering particle distributions in cryo-electron microscopy sample preparation". *Acta Crystallographica Section D Structural Biology* 74:6, 2018, pages 560–571.
DOI: 10.1107/s2059798318006496. https://doi.org/10.1107%2Fs2059798318006496

22. J. von der Ecken et al. "Cryo-EM structure of a human cytoplasmic actomyosin complex at near-atomic resolution". *Nature* 534:7609, 2016, pages 724–728.
DOI: 10.1038/nature18295. https://doi.org/10.1038/nature18295

23. E. H. Egelman. "A robust algorithm for the reconstruction of helical filaments using single-particle methods". *Ultramicroscopy* 85:4, 2000, pages 225–234.
DOI: 10.1016/s0304-3991(00)00062-0. https://doi.org/10.1016/s0304-3991(00)00062-0

24. E. H. Egelman. "Reconstruction of Helical Filaments and Tubes". In: *Methods in Enzymology*. Elsevier, 2010, pages 167–183.
DOI: 10.1016/s0076-6879(10)82006-3. https://doi.org/10.1016/s0076-6879(10)82006-3

25. E. H. Egelman and F. Wang. "Cryo-EM is a powerful tool, but helical applications can have pitfalls". *Soft Matter* 17:12, 2021, pages 3291–3293.
DOI: 10.1039/d1sm00282a. https://doi.org/10.1039/d1sm00282a

26. D. Elmlund and H. Elmlund. "SIMPLE: Software for ab initio reconstruction of heterogeneous single-particles". *Journal of Structural Biology* 180:3, 2012, pages 420–427.
DOI: 10.1016/j.jsb.2012.07.010. https://doi.org/10.1016/j.jsb.2012.07.010

27. *EMPIAR*. 2021.
https://www.ebi.ac.uk/pdbe/emdb/empiar/ visited on 2021-05-08

28. *EMPIAR-10020*. 2021.
https://www.ebi.ac.uk/pdbe/emdb/empiar/entry/10020/ visited on 2021-06-03

29. *EMPIAR-10313*. 2021.
https://www.ebi.ac.uk/pdbe/emdb/empiar/entry/10313/ visited on 2021-06-03

30. *EPU (FEI Thermo Fisher)*. 2021.
https://www.thermofisher.com/de/de/home/electron-microscopy/products/software-em-3d-vis/epu-software.html visited on 2021-04-18

31. H. P. Erickson and A. Klug. "Measurement and compensation of defocusing and aberrations by Fourier processing of electron micrographs". *Philosophical Transactions of the Royal Society of London. B, Biological Sciences* 261:837, 1971, pages 105–118.
DOI: 10.1098/rstb.1971.0040. https://doi.org/10.1098/rstb.1971.0040

32. *Falcon 4 Direct Electron Detector*. 2021.
https://www.thermofisher.com/document-connect/document-connect.html?url=https%5C%3A%5C%2F%5C%2Fassets.thermofisher.com%5C%2FTFS-Assets%5C%2FMSD%5C%2FProduct-Information%5C%2FFalcon-4-Detector-datasheet.pdf visited on 2021-12-02

33. X. Fan et al. "Single particle cryo-EM reconstruction of 52 kDa streptavidin at 3.2 Angstrom resolution". *Nature Communications* 10:1, 2019.
DOI: 10.1038/s41467-019-10368-w. https://doi.org/10.1038/s41467-019-10368-w

34. P. S. Foundation. *PyQt*. Version 5.9.2. 2021.
https://riverbankcomputing.com

35. P. S. Foundation. *Python*. Version 3.7. 2021.
https://www.python.org

36. J. Frank et al. "Reconstruction of glutamine synthetase using computer averaging". *Ultramicroscopy* 3, 1978, pages 283–290.
DOI: 10.1016/s0304-3991(78)80038-2. https://doi.org/10.1016/s0304-3991(78)80038-2

37. J. Frank. *Three-Dimensional Electron Microscopy of Macromolecular Assemblies*. Oxford University Press, 2006.
DOI: 10.1093/acprof:oso/9780195182187.001.0001. https://doi.org/10.1093/acprof:oso/9780195182187.001.0001

38. J. Frank et al. “SPIDER and WEB: Processing and Visualization of Images in 3D Electron Microscopy and Related Fields”. *Journal of Structural Biology* 116:1, 1996, pages 190–199.
DOI: 10.1006/jsbi.1996.0030. https://doi.org/10.1006/jsbi.1996.0030

39. S. Fromm and C. Sachse. “Cryo-EM Structure Determination Using Segmented Helical Image Reconstruction”. In: *Methods in Enzymology*. Elsevier, 2016, pages 307–328.
DOI: 10.1016/bs.mie.2016.05.034. https://doi.org/10.1016/bs.mie.2016.05.034

40. S. A. Fromm et al. “Seeing tobacco mosaic virus through direct electron detectors”. *Journal of Structural Biology* 189:2, 2015, pages 87–97.
DOI: 10.1016/j.jsb.2014.12.002. https://doi.org/10.1016/j.jsb.2014.12.002

41. J. Funk et al. “A barbed end interference mechanism reveals how capping protein promotes nucleation in branched actin networks”. *Nature Communications* 12:1, 2021.
DOI: 10.1038/s41467-021-25682-5. https://doi.org/10.1038/s41467-021-25682-5

42. C. Gatsogiannis et al. “A syringe-like injection mechanism in Photorhabdus luminescens toxins”. *Nature* 495:7442, 2013, pages 520–523.
DOI: 10.1038/nature11987. https://doi.org/10.1038/nature11987

43. *GAutomatch*. 2019.
https://www2.mrc-lmb.cam.ac.uk/research/locally-developed-software/zhang-software/ visited on 2021-04-28

44. *GPU ISAC*. 2021.
https://sphire.mpg.de/wiki/doku.php?id=gpu_isac visited on 2021-04-18

45. T. Grant and N. Grigorieff. “Measuring the optimal exposure for single particle cryo-EM using a 2.6 Å reconstruction of rotavirus VP6”. *eLife* 4, 2015.
DOI: 10.7554/elife.06980. https://doi.org/10.7554/elife.06980

46. N. Grigorieff. “Resolution measurement in structures derived from single particles”. *Acta Crystallographica Section D Biological Crystallography* 56:10, 2000, pages 1270–1277.
DOI: 10.1107/s0907444900009549. https://doi.org/10.1107/s0907444900009549

47. Gringer. *TEM scheme*. Wikipedia Deutschland. 2022.
https://upload.wikimedia.org/wikipedia/commons/2/25/Scheme_TEM_en.svg

48. S. R. Hall. “The STAR file: a new format for electronic data transfer and archiving”. *Journal of Chemical Information and Computer Sciences* 31:2, 1991, pages 326–333.
DOI: 10.1021/ci00002a020. https://doi.org/10.1021/ci00002a020

49. C. R. Harris et al. “Array programming with NumPy”. *Nature* 585:7825, 2020, pages 357–362.
DOI: 10.1038/s41586-020-2649-2. https://doi.org/10.1038/s41586-020-2649-2

50. S. He and S. H. Scheres. "Helical reconstruction in RELION". *Journal of Structural Biology* 198:3, 2017, pages 163–176.
DOI: 10.1016/j.jsb.2017.02.003. https://doi.org/10.1016/j.jsb.2017.02.003

51. M. van Heel et al. *Structure and Function of Invertebrate Respiratory Proteins: EMBO Workshop*. Routledge, 1982, pages 69–73

52. M. van Heel and M. Schatz. "Reassessing the Revolution's Resolutions", 2017.
DOI: 10.1101/224402. https://doi.org/10.1101/224402

53. M. van Heel et al. "A New Generation of the IMAGIC Image Processing System". *Journal of Structural Biology* 116:1, 1996, pages 17–24.
DOI: 10.1006/jsbi.1996.0004. https://doi.org/10.1006/jsbi.1996.0004

54. R. Henderson. "Avoiding the pitfalls of single particle cryo-electron microscopy: Einstein from noise". *Proceedings of the National Academy of Sciences* 110:45, 2013, pages 18037–18041.
DOI: 10.1073/pnas.1314449110. https://doi.org/10.1073/pnas.1314449110

55. R. Henderson. "Overview and future of single particle electron cryomicroscopy". *Archives of Biochemistry and Biophysics* 581, 2015, pages 19–24

56. G. T. Herman and J. Frank, eds. *Computational Methods for Three-Dimensional Microscopy Reconstruction*. Springer New York, 2014.
DOI: 10.1007/978-1-4614-9521-5. https://doi.org/10.1007/978-1-4614-9521-5

57. M. Hohn et al. "SPARX, a new environment for Cryo-EM image processing". *Journal of Structural Biology* 157:1, 2007, pages 47–55.
DOI: 10.1016/j.jsb.2006.07.003. https://doi.org/10.1016/j.jsb.2006.07.003

58. J. D. Hunter. "Matplotlib: A 2D graphics environment". *Computing in Science & Engineering* 9:3, 2007, pages 90–95.
DOI: 10.1109/MCSE.2007.55

59. C. Jelsch et al. "Accurate protein crystallography at ultra-high resolution: Valence electron distribution in crambin". *Proceedings of the National Academy of Sciences* 97:7, 2000, pages 3171–3176.
DOI: 10.1073/pnas.97.7.3171. https://doi.org/10.1073/pnas.97.7.3171

60. *JSON*. 2022.
https://www.json.org visited on 2022-01-30

61. *K3 detector data sheet*. 2021.
https://info.gatan.com/acton/attachment/11413/f-067b/1/-/-/-/-/K3_Datasheet_FL4.pdf.pdf visited on 2021-04-25

62. D. Kimanius et al. "Accelerated cryo-EM structure determination with parallelisation using GPUs in RELION-2". *eLife* 5, 2016.
DOI: 10.7554/elife.18722. https://doi.org/10.7554/elife.18722

63. D. Kimanius et al. "New tools for automated cryo-EM single-particle analysis in RELION-4.0". *Biochemical Journal* 478:24, 2021, pages 4169–4185.
DOI: 10.1042/bcj20210708. https://doi.org/10.1042/bcj20210708

64. A. Klug, F. H. C. Crick, and H. W. Wyckoff. "Diffraction by helical structures". *Acta Crystallographica* 11:3, 1958, pages 199–213.
DOI: 10.1107/s0365110x58000517. https://doi.org/10.1107/s0365110x58000517

65. J. R. Kremer, D. N. Mastronarde, and J. McIntosh. "Computer Visualization of Three-Dimensional Image Data Using IMOD". *Journal of Structural Biology* 116:1, 1996, pages 71–76.
DOI: 10.1006/jsbi.1996.0013. https://doi.org/10.1006/jsbi.1996.0013

66. *Krios G4 Datasheet*. 2022.
https://assets.thermofisher.com/TFS-Assets/MSD/Datasheets/krios-g4-datasheet-ds0363.pdf visited on 2022-05-06

67. W. Kuhlbrandt. "The Resolution Revolution". *Science* 343:6178, 2014, pages 1443–1444.
DOI: 10.1126/science.1251652. https://doi.org/10.1126/science.1251652

68. G. C. Lander et al. "Appion: An integrated, database-driven pipeline to facilitate EM image processing". *Journal of Structural Biology* 166:1, 2009, pages 95–102.
DOI: 10.1016/j.jsb.2009.01.002. https://doi.org/10.1016/j.jsb.2009.01.002

69. N. Lee. *Telepot*. Version 12.7. 2021.
https://telepot.readthedocs.io

70. J. G. Leidenfrost. *De aquae communis nonnullis qualitatibus tractatus*. Ovenius, 1756

71. F. Leidreiter et al. "Common architecture of Tc toxins from human and insect pathogenic bacteria". *Science Advances* 5:10, 2019.
DOI: 10.1126/sciadv.aax6497. https://doi.org/10.1126/sciadv.aax6497

72. Y. Li et al. "High-Throughput Cryo-EM Enabled by User-Free Preprocessing Routines". *Structure* 28:7, 2020, 858–869.e3.
DOI: 10.1016/j.str.2020.03.008. https://doi.org/10.1016/j.str.2020.03.008

73. J. B. MacQueen. "Some Methods for Classification and Analysis of MultiVariate Observations". In: *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*. Ed. by L. M. L. Cam and J. Neyman. Vol. 1. University of California Press, 1967, pages 281–297

74. D. Marion. "An Introduction to Biological NMR Spectroscopy". *Molecular & Cellular Proteomics* 12:11, 2013, pages 3006–3025.
DOI: 10.1074/mcp.o113.030239. https://doi.org/10.1074/mcp.o113.030239

75. D. N. Mastronarde. "Automated electron microscope tomography using robust prediction of specimen movements". *Journal of Structural Biology* 152:1, 2005, pages 36–51.
DOI: 10.1016/j.jsb.2005.07.007. https://doi.org/10.1016/j.jsb.2005.07.007

76. A. McPherson and J. A. Gavira. "Introduction to protein crystallization". *Acta Crystallographica Section F Structural Biology Communications* 70:1, 2013, pages 2–20.
DOI: 10.1107/s2053230x13033141. https://doi.org/10.1107/s2053230x13033141

77. D. M. McSweeney, S. M. McSweeney, and Q. Liu. “A self-supervised workflow for particle picking in cryo-EM”. *IUCrJ* 7:4, 2020, pages 719–727.
DOI: 10.1107/s2052252520007241. https://doi.org/10.1107/s2052252520007241

78. F. Merino et al. “Structural transitions of F-actin upon ATP hydrolysis at near-atomic resolution revealed by cryo-EM”. *Nature Structural & Molecular Biology* 25:6, 2018, pages 528–537.
DOI: 10.1038/s41594-018-0074-0. https://doi.org/10.1038/s41594-018-0074-0

79. T. Moriya et al. “High-resolution Single Particle Analysis from Electron Cryomicroscopy Images Using SPHIRE”. *JoVE* 123, 2017, e55448. ISSN: 1940-087X.
DOI: 10.3791/55448. https://doi.org/10.3791/55448

80. T. Nakane et al. “Characterisation of molecular motions in cryo-EM single-particle data by multi-body refinement in RELION”. *eLife* 7, 2018.
DOI: 10.7554/elife.36861. https://doi.org/10.7554/elife.36861

81. T. Nakane et al. “Single-particle cryo-EM at atomic resolution”. *Nature* 587:7832, 2020, pages 152–156.
DOI: 10.1038/s41586-020-2829-0. https://doi.org/10.1038/s41586-020-2829-0

82. H. Nyquist. “Certain Topics in Telegraph Transmission Theory”. *Transactions of the American Institute of Electrical Engineers* 47:2, 1928, pages 617–644.
DOI: 10.1109/t-aiee.1928.5055024. https://doi.org/10.1109/t-aiee.1928.5055024

83. J. Ognjenović, R. Grisshammer, and S. Subramaniam. “Frontiers in Cryo Electron Microscopy of Complex Macromolecular Assemblies”. *Annual Review of Biomedical Engineering* 21:1, 2019, pages 395–415.
DOI: 10.1146/annurev-bioeng-060418-052453. https://doi.org/10.1146/annurev-bioeng-060418-052453

84. T. M. de Oliveira et al. “Cryo-EM: The Resolution Revolution and Drug Discovery”. *SLAS DISCOVERY: Advancing the Science of Drug Discovery* 26:1, 2020, pages 17–31.
DOI: 10.1177/2472555220960401. https://doi.org/10.1177/2472555220960401

85. *Oracle Berkeley database*. 2022.
https://www.oracle.com/database/berkeley-db/ visited on 2022-01-20

86. A. Paredes. “MICROSCOPY | Transmission Electron Microscopy”. In: *Encyclopedia of Food Microbiology*. Elsevier, 2014, pages 711–720.
DOI: 10.1016/b978-0-12-384730-0.00216-0. https://doi.org/10.1016/b978-0-12-384730-0.00216-0

87. P. A. Penczek, J. Frank, and C. M. Spahn. “A method of focused classification, based on the bootstrap 3D variance analysis, and its application to EF-G-dependent translocation”. *Journal of Structural Biology* 154:2, 2006, pages 184–194.
DOI: 10.1016/j.jsb.2005.12.013. https://doi.org/10.1016/j.jsb.2005.12.013

88. P. A. Penczek, R. A. Grassucci, and J. Frank. “The ribosome at improved resolution: New techniques for merging and orientation refinement in 3D cryo-electron microscopy of biological particles”. *Ultramicroscopy* 53:3, 1994, pages 251–270.
DOI: 10.1016/0304-3991(94)90038-8. https://doi.org/10.1016/0304-3991(94)90038-8

89. P. A. Penczek et al. “CTER—Rapid estimation of CTF parameters with error assessment”. *Ultramicroscopy* 140, 2014, pages 9–19.
DOI: 10.1016/j.ultramic.2014.01.009. https://doi.org/10.1016/j.ultramic.2014.01.009

90. S. Pospich. “Deciphering the structural effect of nucleotide hydrolysis and small molecule binding on actin and myosin”, 2021.
DOI: 10.17877/DE290R-22372. https://eldorado.tu-dortmund.de/handle/2003/40500

91. S. Pospich and S. Raunser. “Single particle cryo-EM — an optimal tool to study cytoskeletal proteins”. *Current Opinion in Structural Biology* 52, 2018, pages 16–24.
DOI: 10.1016/j.sbi.2018.07.006. https://doi.org/10.1016/j.sbi.2018.07.006

92. S. Pospich et al. “Cryo-EM Resolves Molecular Recognition Of An Optojasp Photoswitch Bound To Actin Filaments In Both Switch States”. *Angewandte Chemie International Edition* 60:16, 2021, pages 8678–8682.
DOI: 10.1002/anie.202013193. https://doi.org/10.1002/anie.202013193

93. S. Pospich et al. “High-resolution structures of the actomyosin-V complex in three nucleotide states provide insights into the force generation mechanism”. *eLife* 10, 2021.
DOI: 10.7554/elife.73724. https://doi.org/10.7554/elife.73724

94. A. Punjani and D. J. Fleet. “3D variability analysis: Resolving continuous flexibility and discrete heterogeneity from single particle cryo-EM”. *Journal of Structural Biology* 213:2, 2021, page 107702.
DOI: 10.1016/j.jsb.2021.107702. https://doi.org/10.1016/j.jsb.2021.107702

95. A. Punjani, H. Zhang, and D. J. Fleet. “Non-uniform refinement: adaptive regularization improves single-particle cryo-EM reconstruction”. *Nature Methods* 17:12, 2020, pages 1214–1221.
DOI: 10.1038/s41592-020-00990-8. https://doi.org/10.1038/s41592-020-00990-8

96. A. Punjani et al. “cryoSPARC: algorithms for rapid unsupervised cryo-EM structure determination”. *Nature Methods* 14:3, 2017, pages 290–296.
DOI: 10.1038/nmeth.4169. https://doi.org/10.1038/nmeth.4169

97. M. Radermacher et al. “A NEW 3-D RECONSTRUCTION SCHEME APPLIED TO THE 50S RIBOSOMAL SUBUNIT OF E. COLI”. *Journal of Microscopy* 141:1, 1986, RP1–RP2.
DOI: 10.1111/j.1365-2818.1986.tb02693.x. https://doi.org/10.1111/j.1365-2818.1986.tb02693.x

98. K. Ramlaul, C. M. Palmer, and C. H. Aylett. “A Local Agreement Filtering Algorithm for Transmission EM Reconstructions”. *Journal of Structural Biology* 205:1, 2019, pages 30–40.
DOI: 10.1016/j.jsb.2018.11.011. https://doi.org/10.1016/j.jsb.2018.11.011

99. J. B. Reece et al. *Campbell. Biology.* 9th ed. Pearson, 2011. ISBN: 978-0-321-55823-7

100. L. Reimer and H. Kohl. *Transmission Electron Microscopy.* 5th ed. Springer Series in optical Science 36. Springer, 2007. ISBN: 9780387400938

101. *Relion manual 2D classification*. 2021.
https://relion.readthedocs.io/en/latest/SPA_tutorial/Class2D.html visited on 2021-12-05

102. *relion_it.py source code*. 2022.
https://github.com/3dem/relion/blob/f0b2941a908e0d3848e2d9808bb36770c601c089/scripts/relion_it.py visited on 2022-01-30

103. J.-P. Renaud et al. "Cryo-EM in drug discovery: achievements, limitations and prospects". *Nature Reviews Drug Discovery* 17:7, 2018, pages 471–492.
DOI: 10.1038/nrd.2018.77. https://doi.org/10.1038/nrd.2018.77

104. H. Robenek et al. *Mikroskopie in Forschungund Praxis*. Git Verlag GmbH, 1995

105. D. Roderer et al. "Structure of a Tc holotoxin pore provides insights into the translocation mechanism". *Proceedings of the National Academy of Sciences* 116:46, 2019, pages 23083–23090.
DOI: 10.1073/pnas.1909821116. https://doi.org/10.1073/pnas.1909821116

106. A. Rohou and N. Grigorieff. "CTFFIND4: Fast and accurate defocus estimation from electron micrographs". *Journal of Structural Biology* 192:2, 2015, pages 216–221.
DOI: 10.1016/j.jsb.2015.08.008. https://doi.org/10.1016/j.jsb.2015.08.008

107. A. Rohou and N. Grigorieff. "Frealix: Model-based refinement of helical filament structures from electron micrographs". *Journal of Structural Biology* 186:2, 2014, pages 234–244.
DOI: 10.1016/j.jsb.2014.03.012. https://doi.org/10.1016/j.jsb.2014.03.012

108. J. de la Rosa-Trevín et al. "Scipion: A software framework toward integration, reproducibility and validation in 3D electron microscopy". *Journal of Structural Biology* 195:1, 2016, pages 93–99.
DOI: 10.1016/j.jsb.2016.04.010. https://doi.org/10.1016/j.jsb.2016.04.010

109. P. B. Rosenthal and R. Henderson. "Optimal Determination of Particle Orientation, Absolute Hand, and Contrast Loss in Single-particle Electron Cryomicroscopy". *Journal of Molecular Biology* 333:4, 2003, pages 721–745.
DOI: 10.1016/j.jmb.2003.07.013. https://doi.org/10.1016/j.jmb.2003.07.013

110. E. Ruska. "Über den Bau und die Bemessung von Polschuhlinsen für hochauflösende Elektronenmikroskope". *Archiv für Elektrotechniker*. Heft 3/4 Band 38, 1944.
http://ernstruska.digilibrary.de/bibliographie/q053/q053.pdf

111. A. D. Schenk et al. "Live Analysis and Reconstruction of Single-Particle Cryo-Electron Microscopy Data with CryoFLARE". *Journal of Chemical Information and Modeling* 60:5, 2020, pages 2561–2569.
DOI: 10.1021/acs.jcim.9b01102. https://doi.org/10.1021/acs.jcim.9b01102

112. S. Scheres. "Processing of Structurally Heterogeneous Cryo-EM Data in RELION". In: *Methods in Enzymology*. Elsevier, 2016, pages 125–157.
DOI: 10.1016/bs.mie.2016.04.012. https://doi.org/10.1016/bs.mie.2016.04.012

113. S. H. Scheres. "RELION: Implementation of a Bayesian approach to cryo-EM structure determination". *Journal of Structural Biology* 180:3, 2012, pages 519–530.
DOI: 10.1016/j.jsb.2012.09.006. https://doi.org/10.1016/j.jsb.2012.09.006

114. J. Schmidhuber. "Deep learning in neural networks: An overview". *Neural Networks* 61, 2015, pages 85–117.
DOI: 10.1016/j.neunet.2014.09.003. https://doi.org/10.1016/j.neunet.2014.09.003

115. C. Schmidli et al. "Microfluidic protein isolation and sample preparation for high-resolution cryo-EM". *Proceedings of the National Academy of Sciences* 116:30, 2019, pages 15007–15012.
DOI: 10.1073/pnas.1907214116. https://doi.org/10.1073%2Fpnas.1907214116

116. Y. Shi. "A Glimpse of Structural Biology through X-Ray Crystallography". *Cell* 159:5, 2014, pages 995–1014.
DOI: 10.1016/j.cell.2014.10.051. https://doi.org/10.1016/j.cell.2014.10.051

117. M. Stabrin. "Analyse der Auswirkungen von Helium- und Stickstoffkühlung auf die Probenbewegung in der Transmissionselektronenmikroskopie". German. Master's thesis. TU Dortmund, 2016

118. M. Stabrin et al. "TranSPHIRE: automated and feedback-optimized on-the-fly processing for cryo-EM". *Nature Communications* 11:1, 2020.
DOI: 10.1038/s41467-020-19513-2. https://doi.org/10.1038/s41467-020-19513-2

119. A. Stewart and N. Grigorieff. "Noise bias in the refinement of structures derived from single particles". *Ultramicroscopy* 102:1, 2004, pages 67–84.
DOI: 10.1016/j.ultramic.2004.08.008. https://doi.org/10.1016/j.ultramic.2004.08.008

120. C. Suloway et al. "Automated molecular microscopy: The new Leginon system". *Journal of Structural Biology* 151:1, 2005, pages 41–60.
DOI: 10.1016/j.jsb.2005.03.010. https://doi.org/10.1016/j.jsb.2005.03.010

121. G. Tang et al. "EMAN2: An extensible image processing suite for electron microscopy". *Journal of Structural Biology* 157:1, 2007, pages 38–46.
DOI: 10.1016/j.jsb.2006.05.009. https://doi.org/10.1016/j.jsb.2006.05.009

122. D. Tegunov and P. Cramer. "Real-time cryo-electron microscopy data preprocessing with Warp". *Nature Methods* 16:11, 2019, pages 1146–1152.
DOI: 10.1038/s41592-019-0580-y. https://doi.org/10.1038/s41592-019-0580-y

123. D. Tegunov et al. "Multi-particle cryo-EM refinement with M visualizes ribosome-antibiotic complex at 3.5 Å in cells". *Nature Methods* 18:2, 2021, pages 186–193.
DOI: 10.1038/s41592-020-01054-7. https://doi.org/10.1038/s41592-020-01054-7

124. *Telegram*. 2021.
https://telegram.org visited on 2021-05-08

125. *The Nobel Prize in Chemistry 2017*. 2017.
https://www.nobelprize.org/prizes/chemistry/2017/press-release/ visited on 2021-04-18

126. R. F. Thompson et al. “An introduction to sample preparation and imaging by cryo-electron microscopy for structural biology”. *Methods* 100, 2016, pages 3–15.
DOI: 10.1016/j.ymeth.2016.02.017. https://doi.org/10.1016/j.ymeth.2016.02.017

127. F. Thon. “Notizen: Zur Defokussierungsabhängigkeit des Phasenkontrastes bei der elektronenmikroskopischen Abbildung”. *Zeitschrift für Naturforschung A* 21:4, 1966, pages 476–478.
DOI: 10.1515/zna-1966-0417. https://doi.org/10.1515/zna-1966-0417

128. *TranSPHIRE source code*. 2021.
https://github.com/MPI-Dortmund/transphire visited on 2021-06-27

129. D. Vinayagam et al. “Structural basis of TRPC4 regulation by calmodulin and pharmacological agents”. *eLife* 9, 2020.
DOI: 10.7554/elife.60603. https://doi.org/10.7554/elife.60603

130. R. Wade. “A brief look at imaging and contrast transfer”. *Ultramicroscopy* 46:1-4, 1992, pages 145–156.
DOI: 10.1016/0304-3991(92)90011-8. https://doi.org/10.1016/0304-3991(92)90011-8

131. T. Wagner et al. “SPHIRE-crYOLO is a fast and accurate fully automated particle picker for cryo-EM”. *Communications Biology* 2:1, 2019.
DOI: 10.1038/s42003-019-0437-z. https://doi.org/10.1038/s42003-019-0437-z

132. T. Wagner et al. “Two particle-picking procedures for filamentous proteins: SPHIRE-crYOLO filament mode and SPHIRE-STRIPER”. *Acta Crystallographica Section D Structural Biology* 76:7, 2020, pages 613–620.
DOI: 10.1107/s2059798320007342. https://doi.org/10.1107%2Fs2059798320007342

133. D. B. Williams and C. B. Carter. *Transmission Electron Microscopy*. 2nd ed. Springer Series in optical Science 36. Springer, 2009.
DOI: 10.1007/978-0-387-76501-3

134. J. Wu et al. “Massively parallel unsupervised single-particle cryo-EM data clustering via statistical manifold learning”. *PLOS ONE* 12:8, 2017. Ed. by Z. Zhang, e0182130.
DOI: 10.1371/journal.pone.0182130. https://doi.org/10.1371/journal.pone.0182130

135. *YAML*. 2022.
https://yaml.org/ visited on 2022-01-30

136. Z. Yang et al. “Iterative Stable Alignment and Clustering of 2D Transmission Electron Microscope Images”. *Structure* 20:2, 2012, pages 237–247.
DOI: 10.1016/j.str.2011.12.007. https://doi.org/10.1016/j.str.2011.12.007

137. K. M. Yip et al. “Atomic-resolution protein structure determination by cryo-EM”. *Nature* 587:7832, 2020, pages 157–161.
DOI: 10.1038/s41586-020-2833-4. https://doi.org/10.1038/s41586-020-2833-4

138. K. Zhang. “Gctf: Real-time CTF determination and correction”. *Journal of Structural Biology* 193:1, 2016, pages 1–12.
DOI: 10.1016/j.jsb.2015.11.003. https://doi.org/10.1016/j.jsb.2015.11.003

139. S. Q. Zheng et al. “MotionCor2: anisotropic correction of beam-induced motion for improved cryo-electron microscopy”. *Nature Methods* 14:4, 2017, pages 331–332. DOI: 10.1038/nmeth.4193. https://doi.org/10.1038/nmeth.4193

140. J. Zivanov, T. Nakane, and S. H. W. Scheres. “A Bayesian approach to beam-induced motion correction in cryo-EM single-particle analysis”. *IUCrJ* 6:1, 2019, pages 5–17. DOI: 10.1107/s205225251801463x. https://doi.org/10.1107/s205225251801463x

141. J. Zivanov, T. Nakane, and S. H. W. Scheres. “Estimation of high-order aberrations and anisotropic magnification from cryo-EM data sets in RELION-3.1”. *IUCrJ* 7:2, 2020, pages 253–267. DOI: 10.1107/s2052252520000081. https://doi.org/10.1107/s2052252520000081

Glossary

1D one dimensional. 17

2D two dimensional. iv, v, 9, 12–14, 17, 18, 22–25, 28–30, 33, 36–38, 43, 44, 46, 47, 50–52, 54–57, 59, 60, 63, 65–67, 69–72

3D three dimensional. iv–vi, 1, 3, 4, 12, 14, 15, 18–25, 27, 30–33, 35, 37, 39–44, 50–53, 55–60, 63, 65–67, 71, 72, 87–89

AFIS aberration free image shift. 16, 69

bdb Berkeley database. 44

C_c chromatic aberration. 11

C_s spherical aberration. 8, 11, 17, 27–30

CCD charge-coupled device. 5

CMOS complementary metal-oxide-semiconductor. 5

CNN convolutional neural network. 15, 17

CPU central processing unit. 27, 50

cryo-EM transmission electron cryomicroscopy. 1–7, 9, 10, 12–15, 25, 31, 35, 44, 52, 64, 66, 67, 69, 70

cryo-ET transmission electron cryotomography. 3, 18, 19, 25

CTF contrast transfer function. 7, 9, 11, 16–18, 20, 22, 24, 33, 47, 49, 50, 65

DDD direct detecting device. 5, 10

DNA deoxyribonucleic acid. 1

DQE detection quantum efficiency. 5

FFI fringe-free illumination. 16, 69

FSC Fourier shell correlation. 14, 20, 39–42, 55, 90–109

FT Fourier transform. 17, 18, 21

GPU graphics processing unit. 27, 44, 50, 66, 69

GUI graphical user interface. 25, 33, 35, 45, 46, 48, 49, 64, 65

HPC high-performance computing. 27, 44, 50, 51

IHRSR Iterative Helical Realspace Reconstruction. 21, 43, 71

LMNG lauryl maltose neopentyl glycol. 30

NMR nuclear magnetic resonance spectroscopy. 1, 2

RAM random access memory. 27, 50

SNR signal-to-noise ratio. 3, 11, 16–21, 35, 44

SPA single particle analysis. 3, 4, 11, 12, 14–16, 18, 20–23, 25, 35, 37, 39, 44, 69, 71, 72

SSH secure shell. 50

STAR self-defining text archive and retrieval. 44

TEM transmission electron microscope. 4, 6, 15, 16

TMV Tobacco Mosaic Virus. 27, 38–40, 44, 66, 72, 87, 88, 100–109

TRPC4 transient receptor channel 4. 30, 52–56, 66, 70, 110–116

5 Appendix

5.1 Filament results data

Table 5.1: 3D refinement results of the TMV data set running *sp_meridien_alpha.py*.

Run	#Particles Chunk 0	#Outliers Chunk 0	#Particles Chunk 1	#Outliers Chunk 1
1	15 807	79	15 161	0
2	16 591	0	14 377	78
3	16 099	1	14 869	78
4	15 095	1	15 873	79
5	13 761	78	17 207	1

Run	#Iterations	Best iteration	$FSC_{0.143}$ resolution / Å	$FSC_{0.5}$ resolution / Å
1	21	21	4.39	5.28
2	24	24	4.44	5.63
3	23	23	4.44	5.28
4	23	23	4.28	5.28
5	19	19	4.28	5.04

Table 5.2: 3D refinement results of the TMV data set running *sp_meridien.py*.

Run	#Particles Chunk 0	#Outliers Chunk 0	#Particles Chunk 1	#Outliers Chunk 1
1	15 638	0	15 330	0
2	18 354	0	12 614	0
3	14 400	0	16 568	0
4	15 283	0	15 685	0
5	16 656	0	14 312	0

Run	#Iterations	Best iteration	$FSC_{0.143}$ resolution / Å	$FSC_{0.5}$ resolution / Å
1	27	3	8.66	18.77
2	19	3	14.08	19.87
3	25	3	14.08	19.87
4	27	3	8.66	14.69
5	23	3	9.13	18.77

Table 5.3: 3D refinement results of the actomyosin data set running *sp_meridien_alpha.py*.

Run	#Particles Chunk 0	#Outliers Chunk 0	#Particles Chunk 1	#Outliers Chunk 1
1	22 825	7 444	22 472	7 787
2	23 418	7 647	21 879	7 374
3	22 874	7 419	22 423	7 611
4	22 285	7 565	23 012	7 992
5	22 128	7 778	23 169	7 391

Run	#Iterations	Best iteration	$FSC_{0.143}$ resolution / Å	$FSC_{0.5}$ resolution / Å
1	26	26	4.40	6.77
2	23	23	4.46	6.77
3	23	23	4.69	7.33
4	22	22	4.24	6.64
5	24	23	4.19	6.64

Table 5.4: 3D refinement results of the actomyosin data set running *sp_meridien.py*.

Run	#Particles Chunk 0	#Outliers Chunk 0	#Particles Chunk 1	#Outliers Chunk 1
1	24 925	0	20 372	0
2	22 189	0	23 108	0
3	24 546	0	20 751	0
4	20 011	0	25 286	0
5	22 788	0	22 509	0

Run	#Iterations	Best iteration	$FSC_{0.143}$ resolution / Å	$FSC_{0.5}$ resolution / Å
1	14	14	4.46	7.49
2	15	15	4.46	7.49
3	17	17	4.46	7.49
4	20	20	4.51	7.49
5	20	20	4.46	7.49

5.2 Filament FSC data

Table 5.5: Values of the FSCs from the five different runs executed by sp_meridien_alpha.py for the actomyosin data set.

Resolution / $^1/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.001	1.000	1.000	1.000	1.000	1.000
0.003	1.000	1.000	1.000	1.000	1.000
0.006	1.000	1.000	0.999	1.000	1.000
0.009	1.000	1.000	0.999	1.000	0.999
0.012	1.000	1.000	1.000	1.000	1.000
0.016	1.000	1.000	1.000	1.000	1.000
0.019	1.000	1.000	1.000	1.000	1.000
0.022	1.000	1.000	1.000	1.000	1.000
0.025	0.999	0.999	0.999	0.999	0.999
0.028	0.999	0.999	0.999	0.999	0.999
0.031	0.999	0.999	0.999	0.999	0.999
0.034	0.999	0.999	0.999	0.999	0.999
0.038	0.998	0.998	0.999	0.999	0.998
0.041	0.998	0.998	0.998	0.998	0.998
0.044	0.996	0.997	0.996	0.996	0.997
0.047	0.994	0.995	0.994	0.995	0.995
0.050	0.993	0.994	0.993	0.993	0.994
0.053	0.993	0.993	0.993	0.993	0.993
0.056	0.993	0.992	0.993	0.993	0.993
0.059	0.991	0.990	0.990	0.991	0.991
0.062	0.988	0.989	0.988	0.988	0.989
0.066	0.982	0.984	0.982	0.982	0.984
0.069	0.980	0.979	0.979	0.977	0.979
0.072	0.977	0.977	0.978	0.976	0.977
0.075	0.977	0.977	0.977	0.978	0.977
0.078	0.977	0.977	0.975	0.979	0.979
0.081	0.972	0.974	0.970	0.976	0.976
0.084	0.966	0.966	0.963	0.968	0.967
0.088	0.966	0.966	0.963	0.968	0.968
0.091	0.965	0.963	0.960	0.966	0.966
0.094	0.961	0.957	0.955	0.962	0.962
0.097	0.957	0.954	0.955	0.958	0.960
0.100	0.951	0.949	0.949	0.954	0.955

Table 5.6: Values of the FSCs from the five different runs executed by sp_meridien_alpha.py for the actomyosin data set.

Resolution / $^{1}/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.103	0.942	0.937	0.939	0.947	0.946
0.106	0.935	0.929	0.930	0.939	0.939
0.109	0.926	0.922	0.923	0.931	0.936
0.112	0.928	0.926	0.924	0.933	0.938
0.116	0.921	0.918	0.914	0.930	0.929
0.119	0.909	0.904	0.895	0.915	0.914
0.122	0.886	0.884	0.864	0.900	0.890
0.125	0.864	0.857	0.844	0.877	0.867
0.128	0.837	0.823	0.806	0.844	0.843
0.131	0.818	0.802	0.787	0.827	0.821
0.134	0.799	0.780	0.765	0.806	0.804
0.138	0.767	0.753	0.739	0.782	0.784
0.141	0.736	0.736	0.702	0.768	0.760
0.144	0.699	0.707	0.659	0.742	0.738
0.147	0.649	0.644	0.593	0.690	0.692
0.150	0.589	0.569	0.517	0.616	0.609
0.153	0.564	0.541	0.481	0.593	0.589
0.156	0.554	0.532	0.482	0.605	0.606
0.159	0.551	0.538	0.490	0.608	0.605
0.162	0.518	0.514	0.445	0.590	0.582
0.166	0.485	0.464	0.404	0.547	0.539
0.169	0.454	0.437	0.374	0.496	0.490
0.172	0.437	0.406	0.351	0.471	0.463
0.175	0.435	0.399	0.352	0.469	0.465
0.178	0.427	0.408	0.359	0.466	0.478
0.181	0.400	0.378	0.305	0.436	0.453
0.184	0.356	0.332	0.269	0.403	0.425
0.187	0.374	0.347	0.297	0.432	0.433
0.191	0.374	0.350	0.289	0.429	0.429
0.194	0.347	0.326	0.264	0.423	0.416
0.197	0.326	0.295	0.238	0.408	0.392
0.200	0.333	0.304	0.249	0.411	0.394
0.203	0.327	0.312	0.249	0.415	0.399

Table 5.7: Values of the FSCs from the five different runs executed by sp_meridien_alpha.py for the actomyosin data set.

Resolution / $^{1}/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.206	0.328	0.330	0.253	0.414	0.407
0.210	0.324	0.300	0.226	0.381	0.384
0.212	0.306	0.272	0.197	0.363	0.372
0.216	0.303	0.277	0.190	0.381	0.381
0.219	0.325	0.303	0.204	0.407	0.403
0.222	0.329	0.285	0.226	0.394	0.409
0.225	0.307	0.267	0.218	0.381	0.392
0.228	0.274	0.253	0.176	0.357	0.356
0.231	0.273	0.234	0.158	0.354	0.350
0.235	0.261	0.229	0.143	0.334	0.321
0.238	0.215	0.205	0.121	0.296	0.305
0.241	0.204	0.185	0.115	0.293	0.302
0.244	0.199	0.177	0.113	0.290	0.291
0.247	0.177	0.155	0.109	0.257	0.260
0.250	0.156	0.129	0.089	0.214	0.227
0.253	0.126	0.119	0.061	0.194	0.194
0.256	0.103	0.101	0.060	0.178	0.162
0.259	0.097	0.099	0.056	0.159	0.162
0.263	0.096	0.086	0.030	0.137	0.145
0.266	0.083	0.072	0.024	0.131	0.126
0.269	0.059	0.056	0.027	0.095	0.089
0.272	0.059	0.042	0.034	0.079	0.082
0.275	0.053	0.047	0.030	0.084	0.079
0.278	0.051	0.049	0.029	0.084	0.070
0.281	0.040	0.039	0.030	0.067	0.069
0.284	0.029	0.036	0.023	0.046	0.062
0.287	0.029	0.044	0.010	0.041	0.053
0.291	0.027	0.034	0.007	0.050	0.055
0.294	0.031	0.020	0.011	0.039	0.045
0.296	0.019	0.006	0.014	0.029	0.032
0.300	0.014	0.014	0.016	0.021	0.027
0.303	0.026	0.026	0.013	0.034	0.025
0.306	0.035	0.026	0.010	0.037	0.032

Table 5.8: Values of the FSCs from the five different runs executed by sp_meridien_alpha.py for the actomyosin data set.

Resolution / $^{1}/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.309	0.014	0.029	0.006	0.026	0.026
0.312	0.021	0.029	−0.001	0.021	0.024
0.315	0.024	0.019	−0.004	0.028	0.034
0.319	0.017	0.022	−0.001	0.029	0.026
0.322	0.019	0.029	−0.003	0.020	0.016
0.325	0.027	0.026	0.009	0.037	0.019
0.328	0.028	0.014	0.003	0.030	0.019
0.331	0.015	0.015	−0.005	0.027	0.021
0.334	0.018	0.018	−0.005	0.024	0.016
0.337	0.013	0.024	−0.002	0.027	0.012
0.341	0.015	0.015	0.006	0.028	0.011
0.344	0.019	0.017	0.006	0.023	0.012
0.347	0.011	0.007	0.001	0.012	0.011
0.350	0.012	0.015	−0.003	0.021	0.021
0.353	0.007	0.019	0.003	0.027	0.016
0.356	0.009	0.023	0.008	0.036	0.012
0.359	0.017	0.018	0.009	0.029	−0.006
0.363	0.015	0.021	0.010	0.014	0.002
0.365	0.024	0.022	0.008	0.018	0.018
0.369	0.018	0.020	0.007	0.018	0.021
0.372	0.015	0.015	0.018	0.013	0.024
0.375	0.013	0.015	0.014	0.016	0.018
0.378	0.014	0.011	0.009	0.018	0.022
0.381	0.008	0.012	0.003	0.008	0.004
0.385	0.009	0.017	0.010	0.010	0.014
0.387	0.003	0.017	0.015	0.019	0.016
0.390	0.006	0.020	0.010	0.020	0.016
0.394	0.007	0.020	0.000	0.018	0.009
0.397	0.006	0.014	0.001	0.017	0.012
0.400	0.009	0.017	0.008	0.009	0.014
0.403	0.016	0.016	0.007	0.015	0.014
0.406	0.008	0.006	0.011	0.017	0.016
0.409	0.004	0.009	0.013	0.016	0.014

Table 5.9: Values of the FSCs from the five different runs executed by sp_meridien_alpha.py for the actomyosin data set.

Resolution / $^{1}/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.412	0.002	0.012	0.012	0.012	0.012
0.415	0.003	0.008	0.015	0.008	0.012
0.418	0.003	0.011	0.013	0.006	0.018
0.421	0.006	0.009	0.005	0.013	0.013
0.425	0.004	0.012	0.007	0.016	0.017
0.428	0.009	0.020	0.013	0.022	0.015
0.431	0.007	0.016	0.013	0.028	0.018
0.435	0.011	0.015	0.010	0.022	0.013
0.438	0.010	0.014	0.015	0.009	0.012
0.440	0.005	0.005	0.003	0.007	0.007
0.444	−0.001	0.008	−0.001	0.011	0.007
0.447	0.000	0.011	−0.008	0.011	0.011
0.451	0.006	0.018	0.005	0.013	0.020
0.453	0.006	0.013	0.010	0.016	0.013
0.456	0.016	0.009	0.018	0.009	0.018
0.460	0.016	0.012	0.022	0.009	0.021
0.462	0.009	0.011	0.011	0.019	0.015
0.466	0.011	0.007	0.014	0.015	0.018
0.468	0.009	0.004	0.010	0.011	0.022
0.472	0.009	0.010	0.007	0.009	0.016
0.474	0.013	0.012	0.003	0.020	0.015
0.478	0.012	0.016	0.008	0.014	0.012
0.480	0.012	0.012	0.005	0.007	0.014
0.485	0.010	0.015	0.004	0.010	0.007
0.487	0.010	0.014	0.007	0.019	0.006
0.491	0.014	0.009	0.008	0.014	0.013
0.493	0.017	0.013	0.018	0.012	0.005
0.498	0.007	0.019	0.011	0.018	0.015
0.500	0.007	0.024	0.012	0.020	0.023

Table 5.10: Values of the FSCs from the five different runs executed by sp_meridien.py for the actomyosin data set.

Resolution / $^1/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.001	1.000	1.000	1.000	1.000	1.000
0.003	1.000	1.000	0.999	1.000	1.000
0.006	0.999	0.999	0.997	1.000	0.997
0.009	0.999	1.000	0.998	1.000	0.997
0.012	1.000	1.000	0.998	1.000	0.998
0.016	1.000	1.000	0.999	1.000	0.999
0.019	1.000	1.000	0.999	1.000	0.999
0.022	1.000	1.000	0.999	1.000	0.999
0.025	0.999	0.999	0.999	0.999	0.999
0.028	0.998	0.998	0.997	0.998	0.997
0.031	0.998	0.998	0.997	0.998	0.997
0.034	0.998	0.998	0.997	0.998	0.997
0.038	0.997	0.998	0.997	0.997	0.997
0.041	0.997	0.996	0.995	0.996	0.996
0.044	0.993	0.993	0.992	0.992	0.992
0.047	0.989	0.990	0.989	0.991	0.988
0.050	0.987	0.988	0.988	0.987	0.986
0.053	0.988	0.988	0.988	0.986	0.987
0.056	0.987	0.987	0.986	0.985	0.986
0.059	0.983	0.983	0.983	0.982	0.983
0.062	0.978	0.977	0.978	0.976	0.978
0.066	0.968	0.965	0.967	0.967	0.967
0.069	0.961	0.960	0.961	0.963	0.962
0.072	0.959	0.957	0.959	0.959	0.957
0.075	0.956	0.958	0.954	0.959	0.955
0.078	0.959	0.961	0.955	0.959	0.958
0.081	0.954	0.956	0.952	0.955	0.952
0.084	0.944	0.942	0.939	0.944	0.933
0.088	0.944	0.943	0.939	0.945	0.934
0.091	0.940	0.940	0.939	0.939	0.932
0.094	0.932	0.934	0.933	0.932	0.929
0.097	0.929	0.929	0.931	0.929	0.928
0.100	0.921	0.919	0.919	0.919	0.917

Table 5.11: Values of the FSCs from the five different runs executed by sp_meridien.py for the actomyosin data set.

Resolution / $^1/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.103	0.906	0.901	0.904	0.902	0.901
0.106	0.894	0.888	0.889	0.887	0.889
0.109	0.886	0.881	0.879	0.881	0.877
0.112	0.887	0.883	0.883	0.884	0.881
0.116	0.870	0.872	0.873	0.874	0.874
0.119	0.849	0.848	0.854	0.851	0.851
0.122	0.821	0.819	0.823	0.819	0.826
0.125	0.789	0.795	0.789	0.781	0.789
0.128	0.741	0.748	0.733	0.741	0.736
0.131	0.719	0.729	0.711	0.724	0.716
0.134	0.697	0.695	0.693	0.699	0.694
0.138	0.658	0.657	0.654	0.653	0.654
0.141	0.613	0.618	0.618	0.610	0.609
0.144	0.586	0.589	0.598	0.586	0.583
0.147	0.519	0.528	0.537	0.519	0.528
0.150	0.449	0.460	0.472	0.443	0.461
0.153	0.440	0.444	0.452	0.434	0.440
0.156	0.448	0.456	0.465	0.455	0.444
0.159	0.467	0.470	0.466	0.470	0.472
0.162	0.431	0.427	0.422	0.434	0.441
0.166	0.377	0.380	0.366	0.361	0.386
0.169	0.343	0.340	0.327	0.317	0.346
0.172	0.329	0.325	0.310	0.308	0.331
0.175	0.313	0.313	0.301	0.318	0.323
0.178	0.310	0.315	0.310	0.320	0.331
0.181	0.289	0.306	0.295	0.297	0.304
0.184	0.279	0.284	0.283	0.272	0.277
0.187	0.295	0.289	0.293	0.273	0.293
0.191	0.290	0.285	0.283	0.281	0.287
0.194	0.268	0.264	0.263	0.275	0.273
0.197	0.263	0.257	0.266	0.265	0.261
0.200	0.271	0.261	0.266	0.270	0.263
0.203	0.261	0.257	0.258	0.251	0.250

Table 5.12: Values of the FSCs from the five different runs executed by sp_meridien.py for the actomyosin data set.

Resolution / $^{1}/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.206	0.254	0.259	0.255	0.253	0.250
0.210	0.235	0.244	0.246	0.239	0.243
0.212	0.226	0.226	0.225	0.223	0.221
0.216	0.243	0.240	0.238	0.235	0.233
0.219	0.253	0.256	0.249	0.245	0.251
0.222	0.257	0.252	0.258	0.246	0.253
0.225	0.244	0.252	0.248	0.243	0.235
0.228	0.233	0.236	0.226	0.222	0.220
0.231	0.219	0.221	0.207	0.200	0.205
0.235	0.200	0.219	0.195	0.198	0.194
0.238	0.192	0.199	0.195	0.183	0.187
0.241	0.167	0.176	0.175	0.166	0.175
0.244	0.166	0.170	0.167	0.165	0.161
0.247	0.155	0.150	0.151	0.137	0.143
0.250	0.132	0.125	0.115	0.119	0.114
0.253	0.104	0.114	0.105	0.100	0.100
0.256	0.089	0.101	0.102	0.100	0.094
0.259	0.082	0.087	0.083	0.081	0.083
0.263	0.079	0.081	0.071	0.078	0.081
0.266	0.068	0.067	0.055	0.068	0.062
0.269	0.051	0.054	0.041	0.053	0.053
0.272	0.044	0.039	0.044	0.044	0.039
0.275	0.043	0.046	0.052	0.047	0.042
0.278	0.043	0.052	0.053	0.049	0.045
0.281	0.037	0.043	0.037	0.033	0.039
0.284	0.025	0.040	0.027	0.026	0.020
0.287	0.017	0.040	0.033	0.028	0.026
0.291	0.024	0.031	0.032	0.023	0.029
0.294	0.021	0.024	0.021	0.011	0.026
0.296	0.014	0.021	0.013	0.007	0.017
0.300	0.006	0.015	0.016	−0.002	0.013
0.303	0.003	0.010	0.005	−0.006	0.004
0.306	0.014	0.015	0.007	0.002	0.008

Table 5.13: Values of the FSCs from the five different runs executed by sp_meridien.py for the actomyosin data set.

Resolution / $^{1}/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.309	0.017	0.016	0.010	0.009	0.019
0.312	0.010	0.004	0.010	0.006	0.011
0.315	0.013	0.006	0.004	0.007	0.005
0.319	0.001	0.003	0.007	0.002	0.010
0.322	0.003	0.001	0.004	−0.001	0.004
0.325	−0.007	0.003	0.011	0.003	−0.003
0.328	−0.011	−0.006	0.013	0.001	−0.002
0.331	−0.004	0.002	0.001	0.002	0.000
0.334	0.007	−0.004	0.002	0.006	−0.005
0.337	0.003	−0.005	0.003	0.005	0.006
0.341	−0.004	−0.009	−0.004	0.001	−0.003
0.344	−0.004	−0.016	−0.003	−0.002	−0.010
0.347	−0.002	−0.004	−0.005	0.002	0.002
0.350	0.006	−0.001	0.001	0.005	0.003
0.353	0.005	0.002	0.003	0.006	0.002
0.356	0.003	0.006	−0.001	0.002	0.007
0.359	0.001	0.004	0.003	−0.002	0.000
0.363	0.005	0.001	−0.009	−0.010	−0.009
0.365	0.006	0.003	−0.003	−0.010	−0.007
0.369	0.007	−0.005	0.002	−0.007	−0.004
0.372	0.005	0.005	0.000	0.003	0.001
0.375	0.000	0.001	−0.003	−0.002	0.002
0.378	−0.006	−0.007	0.001	−0.008	−0.003
0.381	−0.007	−0.010	0.001	−0.004	−0.006
0.385	−0.007	−0.002	0.002	0.000	−0.012
0.387	−0.001	−0.002	−0.006	−0.003	−0.005
0.390	0.002	0.002	0.002	−0.003	0.002
0.394	0.007	0.005	0.000	−0.003	0.004
0.397	0.003	0.003	−0.002	−0.004	0.003
0.400	−0.005	0.003	0.003	−0.008	−0.005
0.403	−0.005	0.005	0.005	−0.003	−0.006
0.406	−0.008	0.003	−0.001	−0.001	−0.004
0.409	0.007	0.001	−0.001	0.005	0.001

Table 5.14: Values of the FSCs from the five different runs executed by sp_meridien.py for the actomyosin data set.

Resolution / $^{1}/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.412	0.006	−0.005	−0.005	0.007	0.000
0.415	0.000	−0.008	−0.006	0.000	−0.007
0.418	−0.005	−0.005	−0.001	0.000	−0.001
0.421	−0.004	−0.005	−0.003	0.004	0.001
0.425	0.000	0.000	0.000	0.004	−0.002
0.428	0.000	0.008	−0.003	0.005	0.000
0.431	0.002	0.009	−0.004	0.003	0.002
0.435	−0.002	0.003	0.000	0.001	−0.001
0.438	−0.004	−0.007	0.001	−0.009	0.005
0.440	−0.003	−0.008	−0.005	−0.008	−0.006
0.444	−0.003	−0.003	−0.002	−0.008	−0.009
0.447	0.002	−0.002	−0.004	−0.006	−0.005
0.451	0.005	0.001	0.006	−0.005	0.005
0.453	−0.005	0.005	−0.001	0.000	0.012
0.456	−0.002	−0.001	−0.002	0.002	0.011
0.460	0.001	0.000	−0.003	0.001	0.004
0.462	−0.004	0.005	−0.003	−0.007	0.001
0.466	−0.006	0.003	−0.010	−0.004	0.003
0.468	−0.005	0.002	−0.001	0.002	0.002
0.472	−0.008	−0.002	−0.003	−0.005	−0.006
0.474	−0.004	−0.002	0.000	−0.001	−0.007
0.478	−0.002	0.002	−0.007	0.000	−0.001
0.480	0.002	0.000	−0.001	0.000	−0.010
0.485	0.004	−0.001	−0.001	−0.001	−0.002
0.487	0.001	0.003	0.005	0.000	0.000
0.491	0.004	0.001	0.006	−0.001	−0.004
0.493	−0.001	−0.001	−0.001	0.007	−0.003
0.498	−0.001	−0.002	−0.002	0.006	−0.003
0.500	0.002	−0.004	0.002	0.003	−0.002

Table 5.15: Values of the FSCs from the five different runs executed by sp_meridien_alpha.py for the TMV data set.

Resolution / $^{1}/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.001	1.000	1.000	1.000	1.000	1.000
0.003	1.000	1.000	1.000	1.000	1.000
0.007	0.999	0.999	0.999	0.997	0.999
0.010	1.000	0.999	1.000	1.000	1.000
0.013	1.000	0.999	1.000	1.000	0.999
0.017	0.999	0.998	0.999	0.999	0.999
0.020	0.997	0.997	0.999	0.996	0.995
0.023	1.000	0.999	1.000	0.999	0.999
0.027	0.998	0.998	0.998	0.997	0.998
0.030	0.998	0.997	0.998	0.997	0.998
0.033	0.995	0.996	0.997	0.994	0.996
0.037	0.989	0.992	0.992	0.984	0.993
0.040	0.989	0.993	0.994	0.985	0.994
0.043	0.989	0.993	0.995	0.986	0.994
0.047	0.991	0.992	0.993	0.991	0.994
0.050	0.993	0.993	0.995	0.994	0.994
0.053	0.986	0.987	0.990	0.989	0.987
0.057	0.979	0.983	0.981	0.984	0.981
0.060	0.959	0.971	0.970	0.963	0.962
0.063	0.944	0.960	0.954	0.953	0.945
0.067	0.956	0.973	0.968	0.969	0.966
0.070	0.960	0.974	0.965	0.969	0.969
0.073	0.969	0.978	0.971	0.972	0.968
0.077	0.967	0.971	0.966	0.965	0.963
0.080	0.958	0.961	0.958	0.960	0.959
0.083	0.944	0.949	0.956	0.953	0.961
0.087	0.950	0.957	0.962	0.953	0.961
0.090	0.955	0.953	0.966	0.946	0.955
0.093	0.957	0.959	0.967	0.954	0.954
0.097	0.960	0.966	0.970	0.965	0.961
0.100	0.964	0.973	0.974	0.970	0.971

Table 5.16: Values of the FSCs from the five different runs executed by sp_meridien_alpha.py for the TMV data set.

Resolution / $^1/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.103	0.960	0.971	0.967	0.964	0.966
0.107	0.954	0.961	0.969	0.955	0.958
0.110	0.944	0.953	0.963	0.945	0.954
0.113	0.943	0.950	0.958	0.935	0.948
0.117	0.928	0.931	0.944	0.912	0.928
0.120	0.931	0.931	0.944	0.928	0.932
0.123	0.934	0.934	0.945	0.934	0.927
0.127	0.921	0.913	0.931	0.915	0.903
0.130	0.902	0.887	0.913	0.896	0.898
0.133	0.832	0.846	0.863	0.858	0.858
0.137	0.839	0.862	0.867	0.860	0.842
0.140	0.867	0.872	0.881	0.862	0.854
0.143	0.846	0.837	0.847	0.831	0.834
0.146	0.835	0.850	0.848	0.827	0.837
0.150	0.842	0.854	0.860	0.822	0.827
0.153	0.830	0.821	0.843	0.822	0.806
0.156	0.808	0.794	0.813	0.795	0.804
0.160	0.775	0.725	0.752	0.728	0.758
0.163	0.750	0.690	0.724	0.687	0.741
0.166	0.651	0.607	0.674	0.631	0.664
0.170	0.582	0.608	0.637	0.581	0.614
0.173	0.638	0.660	0.683	0.638	0.696
0.177	0.626	0.616	0.642	0.644	0.684
0.180	0.595	0.549	0.574	0.585	0.607
0.183	0.633	0.587	0.612	0.614	0.625
0.187	0.628	0.571	0.619	0.615	0.627
0.190	0.551	0.504	0.561	0.581	0.604
0.193	0.563	0.528	0.568	0.594	0.610
0.196	0.599	0.564	0.587	0.625	0.638
0.200	0.571	0.537	0.558	0.606	0.615
0.203	0.545	0.467	0.519	0.561	0.545

Table 5.17: Values of the FSCs from the five different runs executed by sp_meridien_alpha.py for the TMV data set.

Resolution / $^{1}/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.206	0.531	0.463	0.515	0.543	0.556
0.210	0.518	0.510	0.532	0.556	0.586
0.213	0.533	0.484	0.527	0.570	0.583
0.216	0.480	0.380	0.444	0.495	0.518
0.220	0.443	0.396	0.451	0.503	0.520
0.223	0.464	0.433	0.467	0.547	0.563
0.226	0.401	0.378	0.416	0.484	0.499
0.230	0.402	0.353	0.404	0.464	0.507
0.233	0.348	0.303	0.341	0.423	0.447
0.236	0.303	0.256	0.280	0.387	0.374
0.240	0.323	0.252	0.269	0.396	0.372
0.243	0.298	0.230	0.254	0.408	0.353
0.247	0.225	0.188	0.231	0.349	0.292
0.250	0.237	0.197	0.217	0.319	0.323
0.253	0.205	0.165	0.161	0.290	0.295
0.256	0.144	0.109	0.127	0.282	0.241
0.260	0.086	0.077	0.092	0.199	0.190
0.263	0.072	0.075	0.093	0.160	0.164
0.267	0.047	0.053	0.054	0.119	0.125
0.270	0.031	0.035	0.048	0.093	0.088
0.273	0.007	0.004	0.028	0.069	0.063
0.276	0.020	0.017	0.051	0.079	0.066
0.280	0.007	0.016	0.014	0.048	0.037
0.283	0.020	0.019	0.004	0.035	0.010
0.286	0.020	0.019	0.010	0.035	0.017
0.290	0.008	0.016	0.028	0.030	0.022
0.293	0.014	0.018	0.032	0.025	0.023
0.296	0.013	0.026	0.028	0.022	0.015
0.300	−0.009	0.006	−0.001	0.007	−0.005
0.303	−0.002	0.011	−0.001	0.001	−0.013
0.307	0.007	0.012	−0.003	0.004	0.002

Table 5.18: Values of the FSCs from the five different runs executed by sp_meridien_alpha.py for the TMV data set.

Resolution / $^{1}/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.310	0.006	−0.014	−0.015	−0.002	−0.002
0.313	0.011	−0.005	−0.003	−0.003	−0.004
0.316	0.003	0.010	0.000	−0.004	0.004
0.320	0.009	0.017	−0.001	−0.004	0.013
0.323	0.004	0.007	0.002	0.006	0.011
0.326	0.002	0.004	0.013	0.022	0.022
0.330	0.001	0.002	0.014	0.020	0.031
0.333	−0.010	−0.008	−0.004	−0.003	0.009
0.337	0.012	0.009	0.008	−0.009	0.020
0.340	0.006	0.008	0.019	0.005	0.027
0.343	−0.006	0.001	0.008	0.000	0.015
0.346	−0.001	0.007	0.014	0.008	0.011
0.349	0.001	0.005	0.013	0.005	0.010
0.353	0.002	0.004	0.007	0.006	0.005
0.356	0.006	0.001	0.013	0.005	−0.001
0.359	0.010	−0.003	0.020	0.005	0.003
0.363	0.010	−0.002	0.007	0.006	0.010
0.366	0.000	0.006	−0.001	0.005	0.015
0.370	−0.003	−0.007	0.001	−0.002	0.002
0.373	−0.007	−0.007	−0.003	−0.007	−0.002
0.376	−0.010	−0.007	−0.004	−0.003	−0.005
0.380	0.006	0.010	−0.003	0.004	−0.001
0.383	0.000	0.002	0.006	0.006	0.011
0.387	−0.013	0.003	0.015	0.003	0.008
0.389	−0.005	0.001	−0.004	−0.003	−0.002
0.393	0.008	−0.003	−0.002	0.000	0.010
0.396	0.003	−0.012	−0.002	−0.013	0.000
0.400	0.006	0.000	−0.002	−0.016	−0.003
0.403	0.004	0.001	−0.002	−0.007	−0.003
0.406	−0.010	−0.005	−0.007	−0.004	−0.003
0.409	0.004	−0.004	−0.008	−0.002	−0.008

Table 5.19: Values of the FSCs from the five different runs executed by sp_meridien_alpha.py for the TMV data set.

Resolution / $^{1}/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.414	0.007	−0.002	0.010	0.004	−0.002
0.417	0.003	−0.001	0.002	0.004	0.006
0.420	−0.001	−0.004	−0.009	−0.008	−0.001
0.423	−0.008	−0.007	−0.010	−0.003	−0.003
0.426	−0.004	−0.013	0.001	0.005	0.002
0.429	0.007	−0.012	0.007	0.010	−0.003
0.433	0.001	−0.008	0.000	0.006	−0.006
0.436	−0.003	−0.010	−0.001	0.003	0.010
0.439	0.004	−0.005	−0.003	0.009	0.016
0.443	0.002	0.001	−0.001	0.001	0.005
0.446	−0.009	0.004	0.000	0.010	0.003
0.450	−0.008	−0.001	−0.002	0.016	0.007
0.454	−0.003	−0.002	0.010	0.004	0.010
0.455	−0.005	0.001	0.006	0.001	−0.003
0.459	0.003	−0.004	−0.006	0.003	−0.009
0.463	0.005	−0.002	−0.004	−0.001	−0.002
0.467	0.007	−0.006	−0.001	0.007	−0.008
0.469	0.008	−0.001	0.000	0.008	−0.001
0.473	0.003	0.007	−0.008	0.000	0.005
0.477	0.001	0.008	−0.003	−0.002	0.006
0.479	0.000	0.000	0.001	0.003	0.002
0.483	−0.001	−0.002	0.001	0.007	0.007
0.487	−0.001	−0.004	0.001	0.005	0.005
0.489	0.001	−0.006	−0.001	0.001	0.002
0.493	0.003	−0.003	0.003	0.006	0.005
0.496	0.008	0.000	0.006	−0.002	−0.003
0.500	0.010	−0.002	0.007	−0.001	0.000

Table 5.20: Values of the FSCs from the five different runs executed by sp_meridien.py for the TMV data set.

Resolution / $^1/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.001	1.000	1.000	1.000	1.000	1.000
0.003	0.999	0.999	0.999	1.000	1.000
0.007	0.991	0.982	0.994	0.994	0.992
0.010	0.998	0.996	0.998	0.999	0.998
0.013	0.999	0.997	0.997	0.999	0.999
0.017	0.993	0.986	0.986	0.994	0.994
0.020	0.992	0.982	0.981	0.992	0.991
0.023	0.997	0.991	0.990	0.997	0.996
0.027	0.986	0.965	0.966	0.990	0.982
0.030	0.979	0.951	0.939	0.986	0.977
0.033	0.958	0.902	0.878	0.967	0.955
0.037	0.928	0.771	0.761	0.920	0.920
0.040	0.946	0.812	0.830	0.933	0.934
0.043	0.945	0.844	0.881	0.951	0.943
0.047	0.944	0.866	0.891	0.942	0.942
0.050	0.952	0.869	0.896	0.945	0.943
0.053	0.894	0.710	0.748	0.880	0.866
0.057	0.813	0.556	0.505	0.787	0.769
0.060	0.655	0.318	0.294	0.669	0.563
0.063	0.468	0.175	0.165	0.522	0.255
0.067	0.619	0.201	0.311	0.661	0.381
0.070	0.547	0.236	0.279	0.573	0.423
0.073	0.512	0.203	0.189	0.558	0.456
0.077	0.534	0.239	0.245	0.577	0.503
0.080	0.482	0.176	0.160	0.456	0.402
0.083	0.416	0.125	0.105	0.371	0.310
0.087	0.481	0.194	0.145	0.501	0.316
0.090	0.411	0.147	0.163	0.436	0.220
0.093	0.533	0.209	0.317	0.534	0.233
0.097	0.588	0.318	0.404	0.623	0.439
0.100	0.662	0.405	0.387	0.651	0.562

Table 5.21: Values of the FSCs from the five different runs executed by sp_meridien.py for the TMV data set.

Resolution / $^1/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.103	0.596	0.269	0.263	0.561	0.482
0.107	0.516	0.179	0.224	0.484	0.414
0.110	0.483	0.139	0.193	0.447	0.331
0.113	0.394	0.090	0.129	0.363	0.256
0.117	0.289	0.038	0.068	0.205	0.156
0.120	0.341	0.077	0.099	0.235	0.201
0.123	0.370	0.103	0.092	0.248	0.222
0.127	0.257	0.060	0.070	0.188	0.132
0.130	0.190	0.047	0.048	0.145	0.066
0.133	0.127	0.008	0.010	0.059	0.015
0.137	0.123	0.018	0.019	0.050	0.042
0.140	0.146	0.054	0.034	0.081	0.070
0.143	0.103	0.035	0.031	0.072	0.034
0.146	0.092	0.029	0.038	0.057	0.041
0.150	0.113	0.009	0.034	0.066	0.044
0.153	0.131	0.029	0.028	0.053	0.068
0.156	0.132	0.020	0.020	0.032	0.028
0.160	0.093	0.019	−0.015	0.014	0.010
0.163	0.044	0.024	−0.007	0.014	0.002
0.166	0.043	0.000	−0.004	0.006	0.009
0.170	0.048	0.009	−0.003	0.023	0.009
0.173	0.050	0.007	0.008	0.028	0.025
0.177	0.036	0.008	−0.001	0.013	0.023
0.180	0.033	−0.004	0.001	0.009	0.002
0.183	0.039	−0.014	0.001	0.018	−0.006
0.187	0.060	−0.003	0.003	0.006	0.002
0.190	0.042	0.001	0.010	0.010	−0.002
0.193	0.031	0.006	0.004	0.022	0.006
0.196	0.031	0.005	0.014	0.021	0.008
0.200	0.035	−0.003	0.002	0.002	0.014
0.203	0.020	0.006	0.004	0.002	0.027

Table 5.22: Values of the FSCs from the five different runs executed by sp_meridien.py for the TMV data set.

Resolution / $^{1}/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.206	0.030	0.001	0.004	0.001	0.017
0.210	0.036	0.007	−0.005	0.009	−0.012
0.213	0.032	0.000	−0.006	0.013	−0.005
0.216	0.024	−0.001	−0.005	0.010	0.003
0.220	0.014	0.003	0.004	−0.004	−0.003
0.223	−0.002	0.017	0.004	0.008	−0.014
0.226	0.012	0.011	0.011	0.008	0.001
0.230	0.017	0.002	0.022	0.006	−0.007
0.233	−0.009	0.004	0.004	0.008	−0.010
0.236	−0.005	0.006	0.005	0.008	−0.003
0.240	−0.016	0.004	0.000	0.016	0.000
0.243	0.000	0.009	0.000	0.013	−0.005
0.247	0.003	0.012	0.005	0.010	−0.001
0.250	−0.007	0.005	0.004	0.010	−0.005
0.253	0.008	0.005	0.000	−0.004	−0.003
0.256	0.012	0.003	−0.001	−0.001	−0.001
0.260	0.005	0.008	−0.009	−0.002	−0.010
0.263	−0.001	0.014	−0.009	0.001	−0.001
0.267	−0.005	0.004	−0.002	−0.002	0.004
0.270	−0.007	−0.001	0.011	0.002	0.009
0.273	0.004	−0.012	0.004	0.011	0.000
0.276	0.008	−0.016	0.005	−0.007	0.003
0.280	−0.004	−0.003	0.007	−0.001	0.000
0.283	−0.007	0.004	0.022	0.006	−0.010
0.286	−0.004	0.007	0.005	0.006	−0.012
0.290	−0.002	0.002	−0.006	0.013	−0.005
0.293	−0.005	0.002	0.004	0.003	−0.004
0.296	0.001	−0.008	0.010	0.001	0.000
0.300	0.002	0.001	0.008	0.010	0.004
0.303	−0.006	0.007	0.004	0.009	0.005
0.307	0.004	0.001	0.010	0.016	0.004

Table 5.23: Values of the FSCs from the five different runs executed by sp_meridien.py for the TMV data set.

Resolution / $^{1}/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.310	0.001	0.008	0.000	0.012	0.002
0.313	−0.003	0.007	0.006	0.010	−0.002
0.316	0.000	0.001	0.004	0.005	0.002
0.320	−0.002	−0.007	0.002	0.002	−0.011
0.323	−0.006	0.006	0.006	−0.009	−0.010
0.326	−0.001	0.004	0.000	−0.003	0.002
0.330	−0.001	−0.008	−0.001	0.004	−0.001
0.333	0.010	−0.008	−0.005	0.006	0.008
0.337	0.007	0.000	−0.007	0.011	0.007
0.340	−0.002	0.004	−0.002	0.007	0.001
0.343	0.012	0.000	0.000	0.003	0.005
0.346	−0.004	0.001	0.006	−0.005	−0.004
0.349	−0.006	0.005	0.006	−0.009	0.003
0.353	0.002	0.014	0.009	−0.004	0.003
0.356	−0.003	0.009	0.001	0.002	0.005
0.359	0.001	0.003	0.000	0.004	0.003
0.363	0.002	0.005	−0.004	0.008	−0.002
0.366	0.009	0.006	0.002	0.005	0.004
0.370	0.005	0.006	0.000	0.007	−0.002
0.373	−0.002	0.007	−0.004	0.002	0.002
0.376	0.001	0.007	0.001	0.000	0.002
0.380	0.001	0.006	0.002	0.001	−0.002
0.383	0.005	0.001	−0.006	−0.003	−0.003
0.387	0.005	0.004	−0.005	−0.002	−0.013
0.389	−0.003	0.001	−0.009	0.010	−0.005
0.393	0.000	0.002	−0.004	0.006	0.004
0.396	−0.006	−0.001	0.000	0.002	−0.001
0.400	−0.003	0.000	−0.006	0.004	−0.006
0.403	−0.001	0.005	−0.002	0.007	−0.012
0.406	−0.006	0.005	0.000	0.010	−0.003
0.409	−0.001	0.004	−0.001	−0.003	−0.004

Table 5.24: Values of the FSCs from the five different runs executed by sp_meridien.py for the TMV data set.

Resolution / $^{1}/_{pixel}$	FSC run 1	FSC run 2	FSC run 3	FSC run 4	FSC run 5
0.414	0.004	0.003	0.003	−0.008	−0.001
0.417	0.004	0.000	0.000	−0.003	0.000
0.420	0.003	−0.003	−0.005	−0.008	0.000
0.423	0.002	0.000	−0.004	0.002	0.002
0.426	0.004	0.000	−0.001	−0.002	0.001
0.429	0.006	−0.001	−0.001	0.010	0.005
0.433	−0.006	−0.005	0.006	0.005	0.003
0.436	−0.005	−0.005	0.002	0.000	0.000
0.439	0.006	−0.001	−0.005	0.003	−0.002
0.443	0.004	0.002	−0.002	−0.003	0.004
0.446	−0.002	0.002	0.000	−0.004	−0.004
0.450	0.005	0.001	−0.002	−0.004	−0.005
0.454	0.002	0.002	0.005	−0.009	0.003
0.455	0.002	−0.002	−0.001	−0.001	−0.005
0.459	0.000	0.000	0.001	0.006	−0.006
0.463	0.005	−0.001	0.004	0.006	0.000
0.467	0.003	0.000	−0.002	0.003	−0.001
0.469	−0.003	−0.002	0.003	0.009	−0.001
0.473	−0.001	−0.004	0.002	0.007	−0.007
0.477	0.000	−0.005	0.007	0.003	−0.004
0.479	0.002	0.001	0.000	0.001	−0.002
0.483	0.004	−0.002	−0.004	0.003	0.000
0.487	0.002	−0.004	−0.001	0.002	0.001
0.489	−0.001	0.001	0.001	0.000	0.004
0.493	0.002	0.002	−0.001	0.006	0.003
0.496	−0.002	−0.004	−0.003	0.000	0.002
0.500	0.006	−0.006	−0.003	−0.002	0.002

5.3 TranSPHIRE feedback loop results data TRPC4

Table 5.25: *TranSPHIRE* feedback loop evaluation results of the TRPC4 data set of iteration zero using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles "kept"	#Particles "kept" / Micrograph
0 + To.1	01	83 319	166.64	36 751	73.50
0 + To.1	02	83 319	166.64	36 223	72.45
0 + To.1	03	83 319	166.64	37 671	75.34
0 + To.1	04	83 319	166.64	35 602	71.20
0 + To.1	05	83 319	166.64	36 507	73.01
0 + To.1	06	83 319	166.64	35 801	71.60
0 + To.1	07	83 319	166.64	37 639	75.28
0 + To.1	08	83 319	166.64	36 922	73.84
0 + To.1	09	83 319	166.64	34 755	69.51
0 + To.1	10	83 319	166.64	37 166	74.33

Feedback iteration	Run	Particles "kept" / %	Resolution / Å
0 + To.1	01	0.44	3.50
0 + To.1	02	0.43	3.45
0 + To.1	03	0.45	3.45
0 + To.1	04	0.43	3.50
0 + To.1	05	0.44	3.50
0 + To.1	06	0.43	3.55
0 + To.1	07	0.45	3.50
0 + To.1	08	0.44	3.50
0 + To.1	09	0.42	3.45
0 + To.1	10	0.45	3.55

Table 5.26: *TranSPHIRE* feedback loop evaluation results of the TRPC4 data set of iteration one using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles ”kept”	#Particles ”kept” / Micrograph
1 + To.1	01	8 850	17.70	2 293	4.59
1 + To.1	02	8 850	17.70	1 931	3.86
1 + To.1	03	8 850	17.70	2 019	4.04
1 + To.1	04	8 850	17.70	2 114	4.23
1 + To.1	05	8 850	17.70	2 058	4.12
1 + To.1	06	8 850	17.70	2 067	4.13
1 + To.1	07	8 850	17.70	2 102	4.20
1 + To.1	08	8 850	17.70	2 112	4.22
1 + To.1	09	8 850	17.70	2 037	4.07
1 + To.1	10	8 850	17.70	2 140	4.28

Feedback iteration	Run	Particles ”kept” / %	Resolution / Å
1 + To.1	01	0.26	5.44
1 + To.1	02	0.22	5.56
1 + To.1	03	0.23	5.83
1 + To.1	04	0.24	5.21
1 + To.1	05	0.23	5.97
1 + To.1	06	0.23	5.69
1 + To.1	07	0.24	5.32
1 + To.1	08	0.24	4.80
1 + To.1	09	0.23	5.44
1 + To.1	10	0.24	5.83

Table 5.27: *TranSPHIRE* feedback loop evaluation results of the TRPC4 data set of iteration two using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles "kept"	#Particles "kept" / Micrograph
2 + To.1	01	48 857	97.71	22 486	44.97
2 + To.1	02	48 857	97.71	21 603	43.21
2 + To.1	03	48 857	97.71	21 674	43.35
2 + To.1	04	48 857	97.71	22 233	44.47
2 + To.1	05	48 857	97.71	21 795	43.59
2 + To.1	06	48 857	97.71	21 367	42.73
2 + To.1	07	48 857	97.71	21 547	43.09
2 + To.1	08	48 857	97.71	22 425	44.85
2 + To.1	09	48 857	97.71	22 822	45.64
2 + To.1	10	48 857	97.71	21 083	42.17

Feedback iteration	Run	Particles "kept" / %	Resolution / Å
2 + To.1	01	0.46	3.65
2 + To.1	02	0.44	3.65
2 + To.1	03	0.44	3.60
2 + To.1	04	0.46	3.65
2 + To.1	05	0.45	3.65
2 + To.1	06	0.44	3.65
2 + To.1	07	0.44	3.65
2 + To.1	08	0.46	3.65
2 + To.1	09	0.47	3.60
2 + To.1	10	0.43	3.65

Table 5.28: *TranSPHIRE* feedback loop evaluation results of the TRPC4 data set of iteration three using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles ”kept”	#Particles ”kept” / Micrograph
3 + To.1	01	62 984	125.97	31 489	62.98
3 + To.1	02	62 984	125.97	30 175	60.35
3 + To.1	03	62 984	125.97	30 976	61.95
3 + To.1	04	62 984	125.97	30 948	61.90
3 + To.1	05	62 984	125.97	30 593	61.19
3 + To.1	06	62 984	125.97	31 936	63.87
3 + To.1	07	62 984	125.97	31 445	62.89
3 + To.1	08	62 984	125.97	29 915	59.83
3 + To.1	09	62 984	125.97	30 961	61.92
3 + To.1	10	62 984	125.97	31 729	63.46

Feedback iteration	Run	Particles ”kept” / %	Resolution / Å
3 + To.1	01	0.50	3.55
3 + To.1	02	0.48	3.60
3 + To.1	03	0.49	3.60
3 + To.1	04	0.49	3.55
3 + To.1	05	0.49	3.55
3 + To.1	06	0.51	3.55
3 + To.1	07	0.50	3.55
3 + To.1	08	0.47	3.55
3 + To.1	09	0.49	3.50
3 + To.1	10	0.50	3.55

Table 5.29: *TranSPHIRE* feedback loop evaluation results of the TRPC4 data set of iteration four using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles "kept"	#Particles "kept" / Micrograph
4 + To.1	01	73 175	146.35	32 757	65.51
4 + To.1	02	73 175	146.35	34 141	68.28
4 + To.1	03	73 175	146.35	33 676	67.35
4 + To.1	04	73 175	146.35	33 464	66.93
4 + To.1	05	73 175	146.35	33 019	66.04
4 + To.1	06	73 175	146.35	33 209	66.42
4 + To.1	07	73 175	146.35	32 541	65.08
4 + To.1	08	73 175	146.35	33 637	67.27
4 + To.1	09	73 175	146.35	33 685	67.37
4 + To.1	10	73 175	146.35	32 152	64.30

Feedback iteration	Run	Particles "kept" / %	Resolution / Å
4 + To.1	01	0.45	3.55
4 + To.1	02	0.47	3.55
4 + To.1	03	0.46	3.50
4 + To.1	04	0.46	3.55
4 + To.1	05	0.45	3.55
4 + To.1	06	0.45	3.55
4 + To.1	07	0.44	3.55
4 + To.1	08	0.46	3.60
4 + To.1	09	0.46	3.50
4 + To.1	10	0.44	3.60

Table 5.30: *TranSPHIRE* feedback loop evaluation results of the TRPC4 data set of iteration five using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles "kept"	#Particles "kept" / Micrograph
5 + To.1	01	70 758	141.52	33 704	67.41
5 + To.1	02	70 758	141.52	33 935	67.87
5 + To.1	03	70 758	141.52	34 584	69.17
5 + To.1	04	70 758	141.52	32 127	64.25
5 + To.1	05	70 758	141.52	34 118	68.24
5 + To.1	06	70 758	141.52	33 628	67.26
5 + To.1	07	70 758	141.52	34 032	68.06
5 + To.1	08	70 758	141.52	34 409	68.82
5 + To.1	09	70 758	141.52	33 106	66.21
5 + To.1	10	70 758	141.52	34 999	70.00

Feedback iteration	Run	Particles "kept" / %	Resolution / Å
5 + To.1	01	0.48	3.50
5 + To.1	02	0.48	3.55
5 + To.1	03	0.49	3.55
5 + To.1	04	0.45	3.55
5 + To.1	05	0.48	3.60
5 + To.1	06	0.48	3.60
5 + To.1	07	0.48	3.50
5 + To.1	08	0.49	3.50
5 + To.1	09	0.47	3.50
5 + To.1	10	0.49	3.55

Table 5.31: *TranSPHIRE* feedback loop evaluation results of the TRPC4 data set of iteration five using a threshold of 0.375.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles "kept"	#Particles "kept" / Micrograph
5 + To.375	01	56 026	112.05	31 513	63.03
5 + To.375	02	56 026	112.05	32 320	64.64
5 + To.375	03	56 026	112.05	32 066	64.13
5 + To.375	04	56 026	112.05	31 234	62.47
5 + To.375	05	56 026	112.05	30 244	60.49
5 + To.375	06	56 026	112.05	32 176	64.35
5 + To.375	07	56 026	112.05	31 587	63.17
5 + To.375	08	56 026	112.05	31 131	62.26
5 + To.375	09	56 026	112.05	30 798	61.60
5 + To.375	10	56 026	112.05	31 601	63.20

Feedback iteration	Run	Particles "kept" / %	Resolution / Å
5 + To.375	01	0.56	3.55
5 + To.375	02	0.58	3.60
5 + To.375	03	0.57	3.50
5 + To.375	04	0.56	3.55
5 + To.375	05	0.54	3.60
5 + To.375	06	0.57	3.60
5 + To.375	07	0.56	3.55
5 + To.375	08	0.56	3.60
5 + To.375	09	0.55	3.55
5 + To.375	10	0.56	3.55

5.4 TranSPHIRE feedback loop results data holotoxin

Table 5.32: *TranSPHIRE* feedback loop evaluation results of the holotoxin data set of iteration zero using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles "kept"	#Particles "kept" / Micrograph
0 + To.1	01	62 353	124.71	10 806	21.61
0 + To.1	02	62 353	124.71	11 903	23.81
0 + To.1	03	62 353	124.71	11 126	22.25
0 + To.1	04	62 353	124.71	11 717	23.43
0 + To.1	05	62 353	124.71	11 664	23.33
0 + To.1	06	62 353	124.71	10 329	20.66
0 + To.1	07	62 353	124.71	11 170	22.34
0 + To.1	08	62 353	124.71	11 684	23.37
0 + To.1	09	62 353	124.71	11 391	22.78
0 + To.1	10	62 353	124.71	10 672	21.34

Feedback iteration	Run	Particles "kept" / %	Resolution / Å
0 + To.1	01	0.17	4.28
0 + To.1	02	0.19	4.20
0 + To.1	03	0.18	4.28
0 + To.1	04	0.19	4.28
0 + To.1	05	0.19	4.28
0 + To.1	06	0.17	4.28
0 + To.1	07	0.18	4.28
0 + To.1	08	0.19	4.28
0 + To.1	09	0.18	4.24
0 + To.1	10	0.17	4.24

Table 5.33: *TranSPHIRE* feedback loop evaluation results of the holotoxin data set of iteration one using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles "kept"	#Particles "kept" / Micrograph
1 + To.1	01	47 602	95.20	12 592	25.18
1 + To.1	02	47 602	95.20	12 807	25.61
1 + To.1	03	47 602	95.20	12 008	24.02
1 + To.1	04	47 602	95.20	12 503	25.01
1 + To.1	05	47 602	95.20	11 360	22.72
1 + To.1	06	47 602	95.20	12 127	24.25
1 + To.1	07	47 602	95.20	10 622	21.24
1 + To.1	08	47 602	95.20	12 041	24.08
1 + To.1	09	47 602	95.20	12 613	25.23
1 + To.1	10	47 602	95.20	13 253	26.51

Feedback iteration	Run	Particles "kept" / %	Resolution / Å
1 + To.1	01	0.26	4.20
1 + To.1	02	0.27	4.28
1 + To.1	03	0.25	4.24
1 + To.1	04	0.26	4.20
1 + To.1	05	0.24	4.24
1 + To.1	06	0.25	4.20
1 + To.1	07	0.22	4.28
1 + To.1	08	0.25	4.20
1 + To.1	09	0.26	4.24
1 + To.1	10	0.28	4.24

Table 5.34: *TranSPHIRE* feedback loop evaluation results of the holotoxin data set of iteration two using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles "kept"	#Particles "kept" / Micrograph
2 + To.1	01	35 364	70.73	12 791	25.58
2 + To.1	02	35 364	70.73	12 630	25.26
2 + To.1	03	35 364	70.73	12 832	25.66
2 + To.1	04	35 364	70.73	13 477	26.95
2 + To.1	05	35 364	70.73	11 206	22.41
2 + To.1	06	35 364	70.73	13 118	26.24
2 + To.1	07	35 364	70.73	13 146	26.29
2 + To.1	08	35 364	70.73	13 176	26.35
2 + To.1	09	35 364	70.73	12 670	25.34
2 + To.1	10	35 364	70.73	12 861	25.72

Feedback iteration	Run	Particles "kept" / %	Resolution / Å
2 + To.1	01	0.36	4.20
2 + To.1	02	0.36	4.20
2 + To.1	03	0.36	4.24
2 + To.1	04	0.38	4.20
2 + To.1	05	0.32	4.28
2 + To.1	06	0.37	4.16
2 + To.1	07	0.37	4.20
2 + To.1	08	0.37	4.20
2 + To.1	09	0.36	4.16
2 + To.1	10	0.36	4.16

Table 5.35: *TranSPHIRE* feedback loop evaluation results of the holotoxin data set of iteration three using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles "kept"	#Particles "kept" / Micrograph
3 + To.1	01	33 823	67.65	12 707	25.41
3 + To.1	02	33 823	67.65	12 684	25.37
3 + To.1	03	33 823	67.65	13 616	27.23
3 + To.1	04	33 823	67.65	13 514	27.03
3 + To.1	05	33 823	67.65	12 788	25.58
3 + To.1	06	33 823	67.65	13 431	26.86
3 + To.1	07	33 823	67.65	12 761	25.52
3 + To.1	08	33 823	67.65	12 416	24.83
3 + To.1	09	33 823	67.65	13 437	26.87
3 + To.1	10	33 823	67.65	12 718	25.44

Feedback iteration	Run	Particles "kept" / %	Resolution / Å
3 + To.1	01	0.38	4.57
3 + To.1	02	0.38	4.16
3 + To.1	03	0.40	4.24
3 + To.1	04	0.40	4.20
3 + To.1	05	0.38	4.20
3 + To.1	06	0.40	4.20
3 + To.1	07	0.38	4.24
3 + To.1	08	0.37	4.24
3 + To.1	09	0.40	4.16
3 + To.1	10	0.38	4.24

Table 5.36: *TranSPHIRE* feedback loop evaluation results of the holotoxin data set of iteration four using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles ”kept”	#Particles ”kept” / Micrograph
4 + To.1	01	31 903	63.81	12 687	25.37
4 + To.1	02	31 903	63.81	12 580	25.16
4 + To.1	03	31 903	63.81	13 851	27.70
4 + To.1	04	31 903	63.81	13 760	27.52
4 + To.1	05	31 903	63.81	12 847	25.69
4 + To.1	06	31 903	63.81	12 304	24.61
4 + To.1	07	31 903	63.81	11 789	23.58
4 + To.1	08	31 903	63.81	14 150	28.30
4 + To.1	09	31 903	63.81	12 771	25.54
4 + To.1	10	31 903	63.81	13 748	27.50

Feedback iteration	Run	Particles ”kept” / %	Resolution / Å
4 + To.1	01	0.40	4.24
4 + To.1	02	0.39	4.24
4 + To.1	03	0.43	4.16
4 + To.1	04	0.43	4.20
4 + To.1	05	0.40	4.20
4 + To.1	06	0.39	4.28
4 + To.1	07	0.37	4.28
4 + To.1	08	0.44	4.20
4 + To.1	09	0.40	4.20
4 + To.1	10	0.43	4.16

Table 5.37: *TranSPHIRE* feedback loop evaluation results of the holotoxin data set of iteration five using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles "kept"	#Particles "kept" / Micrograph
5 + To.1	01	32 598	65.20	13 142	26.28
5 + To.1	02	32 598	65.20	13 128	26.26
5 + To.1	03	32 598	65.20	12 843	25.69
5 + To.1	04	32 598	65.20	12 889	25.78
5 + To.1	05	32 598	65.20	13 303	26.61
5 + To.1	06	32 598	65.20	11 579	23.16
5 + To.1	07	32 598	65.20	13 038	26.08
5 + To.1	08	32 598	65.20	13 483	26.97
5 + To.1	09	32 598	65.20	13 388	26.78
5 + To.1	10	32 598	65.20	13 156	26.31

Feedback iteration	Run	Particles "kept" / %	Resolution / Å
5 + To.1	01	0.40	4.20
5 + To.1	02	0.40	4.16
5 + To.1	03	0.39	4.20
5 + To.1	04	0.40	4.20
5 + To.1	05	0.41	4.20
5 + To.1	06	0.36	4.24
5 + To.1	07	0.40	4.16
5 + To.1	08	0.41	4.16
5 + To.1	09	0.41	4.13
5 + To.1	10	0.40	4.20

Table 5.38: *TranSPHIRE* feedback loop evaluation results of the holotoxin data set of iteration five using a threshold of 0.194.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles ”kept”	#Particles ”kept” / Micrograph
5 + To.194	01	26 152	52.30	10 981	21.96
5 + To.194	02	26 152	52.30	10 999	22.00
5 + To.194	03	26 152	52.30	12 302	24.60
5 + To.194	04	26 152	52.30	12 338	24.68
5 + To.194	05	26 152	52.30	11 991	23.98
5 + To.194	06	26 152	52.30	12 687	25.37
5 + To.194	07	26 152	52.30	11 879	23.76
5 + To.194	08	26 152	52.30	12 296	24.59
5 + To.194	09	26 152	52.30	12 333	24.67
5 + To.194	10	26 152	52.30	12 215	24.43

Feedback iteration	Run	Particles ”kept” / %	Resolution / Å
5 + To.194	01	0.42	4.20
5 + To.194	02	0.42	4.36
5 + To.194	03	0.47	4.20
5 + To.194	04	0.47	4.20
5 + To.194	05	0.46	4.20
5 + To.194	06	0.49	4.24
5 + To.194	07	0.45	4.40
5 + To.194	08	0.47	4.24
5 + To.194	09	0.47	4.20
5 + To.194	10	0.47	4.24

5.5 TranSPHIRE feedback loop results data actomyosin

Table 5.39: *TranSPHIRE* feedback loop evaluation results of the actomyosin data set of iteration zero using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles "kept"	#Particles "kept" / Micrograph
0 + To.1	01	14 307	143.07	12 461	124.61
0 + To.1	02	14 307	143.07	12 353	123.53
0 + To.1	03	14 307	143.07	12 173	121.73
0 + To.1	04	14 307	143.07	12 423	124.23
0 + To.1	05	14 307	143.07	11 282	112.82
0 + To.1	06	14 307	143.07	12 454	124.54
0 + To.1	07	14 307	143.07	12 251	122.51
0 + To.1	08	14 307	143.07	12 562	125.62
0 + To.1	09	14 307	143.07	12 423	124.23
0 + To.1	10	14 307	143.07	12 500	125.00

Feedback iteration	Run	Particles "kept" / %	Resolution / Å
0 + To.1	01	0.87	6.89
0 + To.1	02	0.86	6.29
0 + To.1	03	0.85	6.89
0 + To.1	04	0.87	6.29
0 + To.1	05	0.79	6.40
0 + To.1	06	0.87	6.89
0 + To.1	07	0.86	7.79
0 + To.1	08	0.88	6.89
0 + To.1	09	0.87	8.15
0 + To.1	10	0.87	6.64

Table 5.40: *TranSPHIRE* feedback loop evaluation results of the actomyosin data set of iteration one using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles "kept"	#Particles "kept" / Micrograph
1 + To.1	01	35 792	357.92	29 649	296.49
1 + To.1	02	35 792	357.92	29 662	296.62
1 + To.1	03	35 792	357.92	29 963	299.63
1 + To.1	04	35 792	357.92	30 177	301.77
1 + To.1	05	35 792	357.92	30 179	301.79
1 + To.1	06	35 792	357.92	30 136	301.36
1 + To.1	07	35 792	357.92	29 692	296.92
1 + To.1	08	35 792	357.92	30 259	302.59
1 + To.1	09	35 792	357.92	30 139	301.39
1 + To.1	10	35 792	357.92	30 226	302.26

Feedback iteration	Run	Particles "kept" / %	Resolution / Å
1 + To.1	01	0.83	5.35
1 + To.1	02	0.83	4.54
1 + To.1	03	0.84	4.48
1 + To.1	04	0.84	4.54
1 + To.1	05	0.84	4.54
1 + To.1	06	0.84	4.54
1 + To.1	07	0.83	4.78
1 + To.1	08	0.85	5.35
1 + To.1	09	0.84	5.05
1 + To.1	10	0.84	4.48

Table 5.41: *TranSPHIRE* feedback loop evaluation results of the actomyosin data set of iteration two using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles "kept"	#Particles "kept" / Micrograph
2 + To.1	01	55 145	551.45	42 594	425.94
2 + To.1	02	55 145	551.45	43 657	436.57
2 + To.1	03	55 145	551.45	43 078	430.78
2 + To.1	04	55 145	551.45	42 915	429.15
2 + To.1	05	55 145	551.45	43 274	432.74
2 + To.1	06	55 145	551.45	43 748	437.48
2 + To.1	07	55 145	551.45	42 959	429.59
2 + To.1	08	55 145	551.45	43 276	432.76
2 + To.1	09	55 145	551.45	43 417	434.17
2 + To.1	10	55 145	551.45	43 171	431.71

Feedback iteration	Run	Particles "kept" / %	Resolution / Å
2 + To.1	01	0.77	4.42
2 + To.1	02	0.79	4.48
2 + To.1	03	0.78	4.42
2 + To.1	04	0.78	4.37
2 + To.1	05	0.78	4.72
2 + To.1	06	0.79	4.32
2 + To.1	07	0.78	4.32
2 + To.1	08	0.78	4.54
2 + To.1	09	0.79	4.78
2 + To.1	10	0.78	4.54

Table 5.42: *TranSPHIRE* feedback loop evaluation results of the actomyosin data set of iteration three using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles ”kept”	#Particles ”kept” / Micrograph
3 + To.1	01	63 917	639.17	46 493	464.93
3 + To.1	02	63 917	639.17	46 290	462.90
3 + To.1	03	63 917	639.17	46 197	461.97
3 + To.1	04	63 917	639.17	46 837	468.37
3 + To.1	05	63 917	639.17	46 196	461.96
3 + To.1	06	63 917	639.17	45 703	457.03
3 + To.1	07	63 917	639.17	46 214	462.14
3 + To.1	08	63 917	639.17	45 650	456.50
3 + To.1	09	63 917	639.17	45 024	450.24
3 + To.1	10	63 917	639.17	46 290	462.90

Feedback iteration	Run	Particles ”kept” / %	Resolution / Å
3 + To.1	01	0.73	4.32
3 + To.1	02	0.72	4.42
3 + To.1	03	0.72	4.54
3 + To.1	04	0.73	4.48
3 + To.1	05	0.72	4.32
3 + To.1	06	0.72	4.48
3 + To.1	07	0.72	4.65
3 + To.1	08	0.71	4.48
3 + To.1	09	0.70	4.32
3 + To.1	10	0.72	4.27

Table 5.43: *TranSPHIRE* feedback loop evaluation results of the actomyosin data set of iteration four using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles "kept"	#Particles "kept" / Micrograph
4 + To.1	01	109 973	1 099.73	54 061	540.61
4 + To.1	02	109 973	1 099.73	52 187	521.87
4 + To.1	03	109 973	1 099.73	53 855	538.55
4 + To.1	04	109 973	1 099.73	55 430	554.30
4 + To.1	05	109 973	1 099.73	52 348	523.48
4 + To.1	06	109 973	1 099.73	52 860	528.60
4 + To.1	07	109 973	1 099.73	54 363	543.63
4 + To.1	08	109 973	1 099.73	54 414	544.14
4 + To.1	09	109 973	1 099.73	53 167	531.67
4 + To.1	10	109 973	1 099.73	54 670	546.70

Feedback iteration	Run	Particles "kept" / %	Resolution / Å
4 + To.1	01	0.49	4.42
4 + To.1	02	0.47	4.78
4 + To.1	03	0.49	4.37
4 + To.1	04	0.50	4.59
4 + To.1	05	0.48	4.42
4 + To.1	06	0.48	4.42
4 + To.1	07	0.49	4.78
4 + To.1	08	0.49	4.72
4 + To.1	09	0.48	4.27
4 + To.1	10	0.50	4.72

Table 5.44: *TranSPHIRE* feedback loop evaluation results of the actomyosin data set of iteration five using a threshold of 0.1.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles ”kept”	#Particles ”kept” / Micrograph
5 + To.1	01	107 343	1 073.43	53 228	532.28
5 + To.1	02	107 343	1 073.43	52 331	523.31
5 + To.1	03	107 343	1 073.43	54 340	543.40
5 + To.1	04	107 343	1 073.43	53 041	530.41
5 + To.1	05	107 343	1 073.43	51 780	517.80
5 + To.1	06	107 343	1 073.43	55 097	550.97
5 + To.1	07	107 343	1 073.43	53 774	537.74
5 + To.1	08	107 343	1 073.43	53 615	536.15
5 + To.1	09	107 343	1 073.43	52 810	528.10
5 + To.1	10	107 343	1 073.43	53 553	535.53

Feedback iteration	Run	Particles ”kept” / %	Resolution / Å
5 + To.1	01	0.50	4.54
5 + To.1	02	0.49	4.65
5 + To.1	03	0.51	4.37
5 + To.1	04	0.49	4.42
5 + To.1	05	0.48	4.59
5 + To.1	06	0.51	4.59
5 + To.1	07	0.50	4.72
5 + To.1	08	0.50	4.32
5 + To.1	09	0.49	4.42
5 + To.1	10	0.50	4.59

Table 5.45: *TranSPHIRE* feedback loop evaluation results of the actomyosin data set of iteration five using a threshold of 0.3.

Feedback iteration	Run	#Particles total	#Particles total / Micrograph	#Particles "kept"	#Particles "kept" / Micrograph
5 + To.3	01	51 483	514.83	38 478	384.78
5 + To.3	02	51 483	514.83	39 325	393.25
5 + To.3	03	51 483	514.83	38 333	383.33
5 + To.3	04	51 483	514.83	38 486	384.86
5 + To.3	05	51 483	514.83	37 722	377.22
5 + To.3	06	51 483	514.83	38 999	389.99
5 + To.3	07	51 483	514.83	38 816	388.16
5 + To.3	08	51 483	514.83	38 733	387.33
5 + To.3	09	51 483	514.83	38 651	386.51
5 + To.3	10	51 483	514.83	39 322	393.22

Feedback iteration	Run	Particles "kept" / %	Resolution / Å
5 + To.3	01	0.75	4.84
5 + To.3	02	0.76	4.98
5 + To.3	03	0.74	4.54
5 + To.3	04	0.75	4.84
5 + To.3	05	0.73	4.98
5 + To.3	06	0.76	5.05
5 + To.3	07	0.75	4.84
5 + To.3	08	0.75	4.42
5 + To.3	09	0.75	4.84
5 + To.3	10	0.76	4.42

Acknowledgements

First of all and most importantly, I want to thank Prof. Dr. Stefan Raunser and Prof. Dr. Metin Tolan for the possibility to accomplish this doctoral dissertation at the Max Planck Institute of Molecular Physiology. Starting from my bachelor thesis, they allowed me to identify and work on the things I love doing. I want to especially thank Stefan for his continuous trust in my work. He tolerated to some extent that I distracted myself from my own assignments to help out with other people's research ;). A big "Vielen Dank" goes to Prof. Metin Tolan for enabling all of my theses in the past and his continuous support and interest in my research. I also want to thank Prof. Dr. Stefan Kast for his support and valuable input during my TAC meetings and the IMPRS organization team Christa and Lucia.

I want to thank everybody in Department 3 for being the best colleagues one could wish for and many I consider friends by now! Although I had my own assignments, I will always cherish the huge smile that appeared on their faces after I solved a problem for them. Your appreciation for me just being me and the friendly atmosphere encouraged me to always give 100 % and to look forward everyday to a new day at the institute. A special thanks goes to the members of the software team: Thorsten (Thomas Ringer), Tapu, Fabian, Adnan, and Luca. It was great fun interacting with you, and I learned a lot about software development, english, project management, and Godzilla (BZZT). Additionally, I want to thank Christos, Toshio, Sabrina, Felipe, Julian, Björn, Daniel P., and Oliver for introducing me to the topics of software development, project management, software testing, data processing, and filamentous proteins.

Another special thanks goes to the running and painful exercising group: Dennis, Eric (Zhexin), and Claudia! It was a lot of fun going through a lot of struggles with you :) Another thanks goes to Eric for helping me understand the basics of the Chinese language and the most memorable ice skating sessions.

Last but not least, I want to thank my family and friends for their continuous support throughout my whole studies! You made it possible for Dortmund to become my new home :) Thank you so much, you are really the best!

Eidesstattliche Versicherung

Zur Person

Name, Vorname

geboren am/in

Matrikelnummer

Belehrung

Die Abgabe einer eidesstattlichen Versicherung ist eine nach §§ 156, 161 Strafgesetzbuch (StGB) strafbewehrte Bestätigung der Richtigkeit einer Erklärung. Die Abgabe einer falschen oder unvollständigen Versicherung an Eides statt ist strafbar.

Wer vorsätzlich eine falsche Versicherung an Eides statt abgibt, kann mit einer Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft werden, § 156 StGB. Die fahrlässige Abgabe einer falschen Versicherung an Eides statt kann mit einer Freiheitsstrafe bis zu einem Jahr oder Geldstrafe bestraft werden, § 161 StGB.

Die oben stehende Belehrung habe ich zur Kenntnis genommen.

Ort, Datum

Unterschrift

Eidesstattliche Versicherung

In Kenntnis der Bedeutung einer eidesstattlichen Versicherung und der Strafbarkeit der Abgabe einer falschen eidesstattlichen Versicherung versichere ich hiermit an Eides statt, dass ich die vorliegende Dissertation mit dem Titel “Fully automated processing and optimization of single particle and filamentous transmission electron cryomicroscopy samples”selbstständig und ohne unzulässige fremde Hilfe angefertigt habe. Ich habe keine anderen als die angegebenen Quellen benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht.

Ort, Datum

Unterschrift