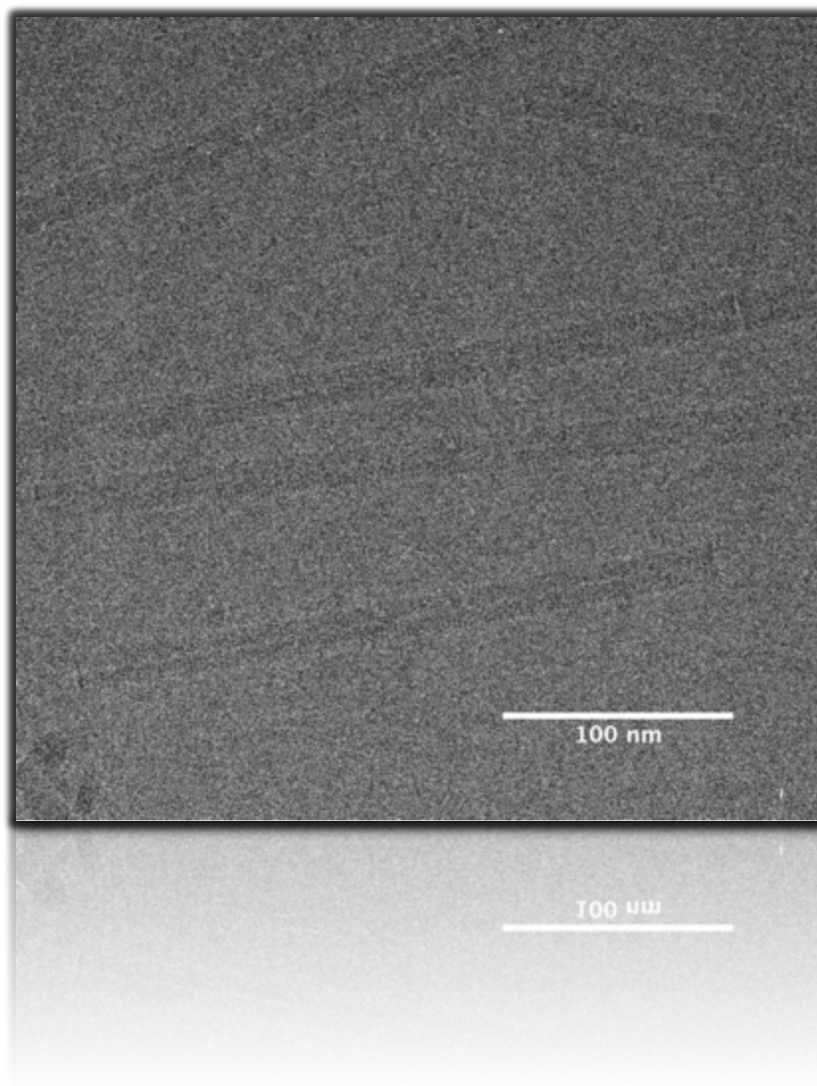


Single-Particle 3D Image Reconstruction of Tobacco Mosaic Virus using SPRING

EMBL, Meyerhofstr. 1, 69117 Heidelberg, Germany

Simon Fromm, Stefan Huber, Arjen Jakobi, Rodolfo Ciuffa, Carsten Sachse



GETTING STARTED	3
1. EXAMINATION OF MICROGRAPHS	3
2. DETERMINATION OF THE CONTRAST TRANSFER FUNCTION	4
3.1 MANUAL SELECTION OF TMV PARTICLES	6
4. SEGMENTATION OF TMV PARTICLES	10
5. ANALYSIS OF EXTRACTED SEGMENTS	11
6. CLASSIFICATION	12
7. CLASS ANALYSIS IN REAL AND FOURIER SPACE	14
8. SYMMETRY DETERMINATION	14
9. 3D STRUCTURE REFINEMENT	15
10. 3D STRUCTURE ANALYSIS	17
11. FRAME PROCESSING WITH DED DATA	18
APPENDIX	22

Tobacco Mosaic Virus (TMV) has been used as a test specimen to determine the three-dimensional (3D) structural information from biological macromolecules from the early days of structural biology (Bawden et al. 1936). In the current tutorial, we will employ single-particle based helical 3D reconstruction to obtain a high-resolution 3D structure from electron cryo-micrographs. TMV is a particularly rigid and highly-ordered virus and only four micrographs are sufficient to compute a sub-nanometer structure. More challenging samples of higher disorder and heterogeneity will require different and more time-consuming processing strategies. We will use SPRING for the entire processing workflow (Figure 1), which is a comprehensive suite of currently developed programs from the Sachse lab and is based on the EM libraries of EMAN2/SPARX (Tang et al. 2007; Hohn et al. 2007) and scientific python packages of Numpy and Scipy (<http://numpy.scipy.org>). In the current tutorial, SPRING will be addressed with a simple Graphical User Interface (GUI).

Each step of the tutorial is carried out using a minimal set of parameters and is meant to allow the user to familiarize with the basics of the software. A brief description of all the other parameters can be obtained by hovering over them with the mouse; they can be used by the user who is already acquainted with the overall architecture of SPRING.

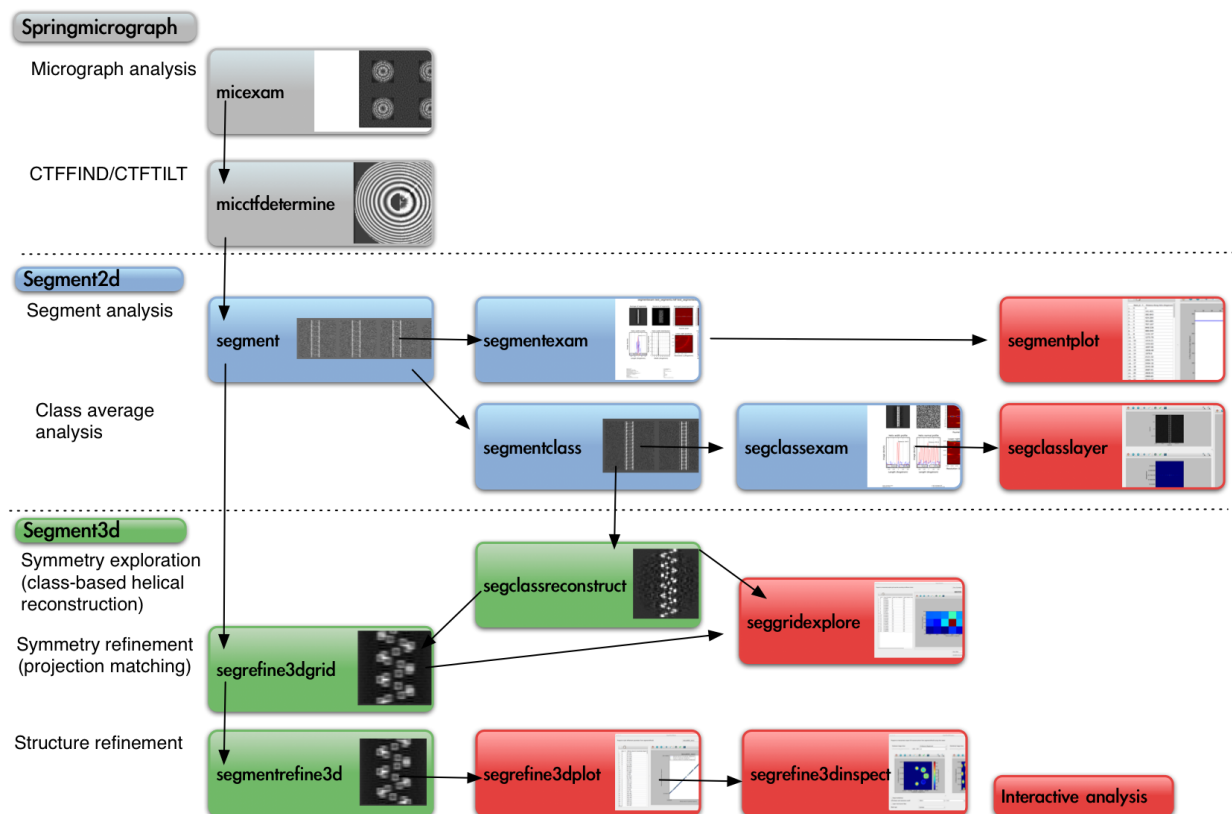


Figure 1. Flow chart of processing helical specimens including micrograph analysis, CTFIND/CTFTILT, segment and class average analysis. The class average can be used for symmetry exploration. The helical symmetry parameters can be refined by a projection-matching based symmetry grid. Finally, the structure can be refined to highest resolution. Interactive GUI programs (red) help to analyze results of the respective programs (from Desfosses et al. 2014).

Getting started

First open the terminal, and then launch the GUI of SPRING by typing:

```
% mkdir tutorial && cd tutorial
```

```
% spring
```

Spring will create new directories for every program that is executed. If you want more information about technical terms and usage of SPRING, please refer to the manual (Help→Open manual).

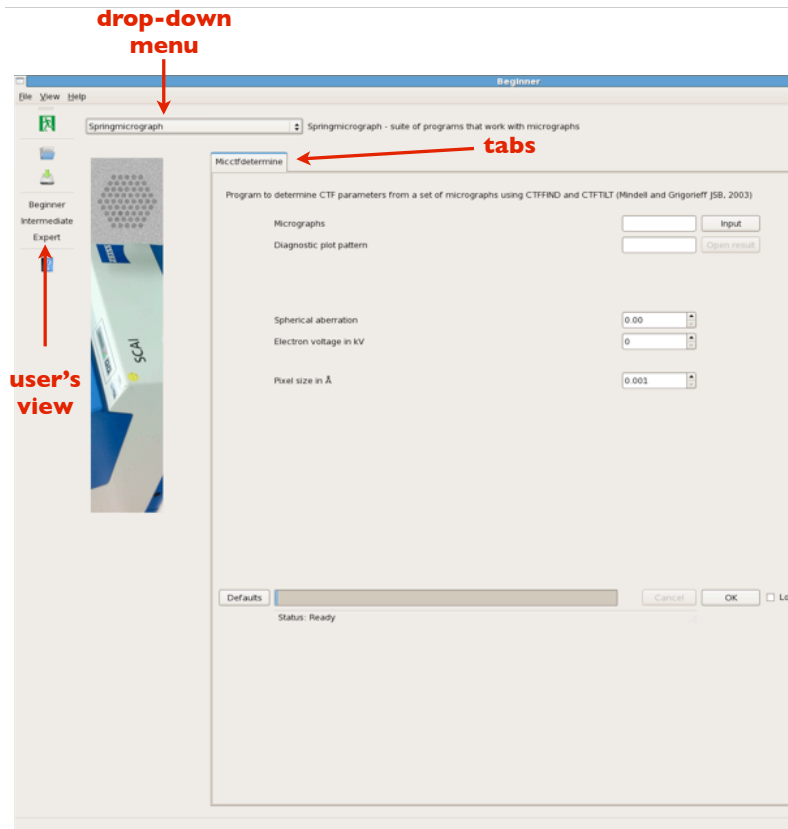


Figure 2. SPRING's Graphical User Interface

1. Examination of micrographs

MICEXAM is a useful program to analyze the quality of your micrographs by examining the Thon rings from the carbon support film.

Once the GUI appears, click on the "Intermediate" user's view. Click on the drop-down menu on the top left, select SPRINGMICROGRAPHS, and go to the tab MICEXAM. Select all files in the current directory (INPUT) and leave the other values as they are (click on the DEFAULT button next to the progress bar; if you're unsure about the parameters to insert, mouse over the parameter field and you will get instructions). Press OK to start the program.

First, the program calculates local power spectra of the micrograph, which are placed on top of the micrograph at the respective position. Second, an average power spectrum of overlapping tiles is displayed and rotationally averaged. The averaged power spectrum reveals how much information has been transferred from the sample

to your images. This step can be used to exclude images that exhibit drift, astigmatism and charging.

Correction of beam-induced specimen motion

In view of the low signal-to-noise ratio in electron cryo-micrographs, image quality is a central factor of how accurately the alignment parameters can be determined during the process of 3D reconstruction. Both instabilities of the sample stage and specimen motion induced by the illuminating electron beam can substantially deteriorate the recorded micrographs. Recently, new detector concepts have been developed that allow direct detection of individual electrons and the read-out of intermediate images at frequencies higher than the acquisition period of an individual exposure. This makes it possible to fractionate a single exposure into a number of subframes that are sufficiently short to attenuate specimen drift to an acceptable level. With such frames it is then possible to correct for drift during the full exposure by alignment of the individual frames (see appendix for an example). In this tutorial two sets of images are provided: (1) integrated micrographs from uncorrected frames and (2) integrated micrographs from frames that have been corrected for beam-induced motion with the program *motioncorr* (Li et al. 2013) <http://cryoem.ucsf.edu/software/driftcorr.html>. You can work with either of the image sets and compare the results.

Q1: How do you identify drift in your micrograph?

Q2: How do you identify astigmatism in your micrograph?

Q3: How do you identify charging in your micrograph?

2. Determination of the Contrast Transfer Function

MICCTFDETERMINE is used to determine the contrast transfer function (CTF) of your micrographs. The CTF is an oscillatory function that describes how different spatial frequencies are modulated in your images. CTFFIND optimizes the fit between the average of power spectrum tiles and the simulated power spectrum of an ideal CTF including astigmatism (Figure 3). CTFTILT detects specimen tilt from the micrograph and fits a corresponding defocus gradient across the micrograph. MICCTFDETERMINE uses CTFFIND and CTFTILT (Mindell & Grigorieff, 2003) combined into a single program.

Table 1. Summary of data acquisition information

Specimen	TMV
Microscope	FEI Titan Krios
Voltage (kV)	300
Spherical aberration	2.7
Underfocus range (μm)	1.5 – 4.5
Number of micrographs	4
Pixel size (\AA)	1.062

CTFFIND power montage

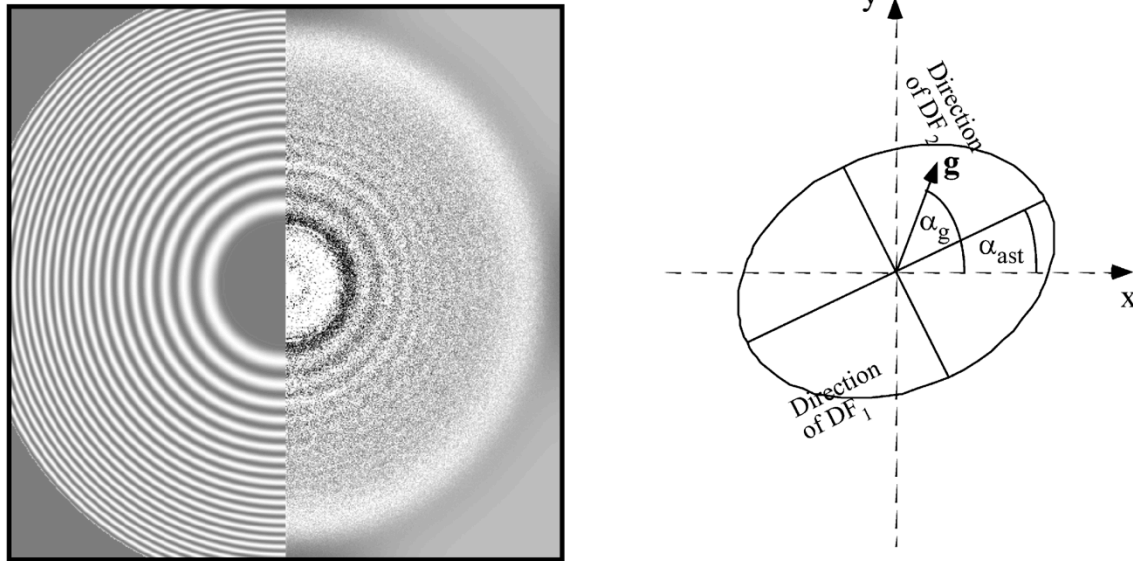


Figure 3. Left. Diagnostic output of CTFFIND. Right. Definition of the CTFFIND/CTFTILT defocus values DF1, DF2, and astigmatism angle α_{ast} (Mindell & Grigorieff 2003).

Move from MICEXAM to the adjacent window MICCTFDETERMINE. The CTF parameters will be stored internally in a database file, and used later on during the processing to correct the CTF.

The data set was collected using a FEI Titan KRIOS microscope at 300 kV and micrographs were recorded on a FEI Falcon II direct electron detector. The micrographs were taken at an underfocus from 1.5 to 4 μm . The pixel size is 1.062 \AA . More acquisition details required for input can be found in Table 1. In the interest of time, data will be binned 2 times, which will make the processing much faster (therefore set *binning factor* to 2). Now we can set the resolution range using the filters to range between 21.0 and 5.0 \AA .

Click on OK to start the MICCTFDETERMINE. Once the computation has come to an end open the diagnostic plot by clicking on the button OPEN RESULT.

Output:

A plot with three columns will appear. In the first column, CTFFIND and CTFTILT plots of the predicted and observed power spectra are shown next to each other. In the second column, two defocus (DF) values are shown for two different regions of the micrograph. If they differ significantly, it means that the image is astigmatic (astigmatism magnitude=DF1-DF2). In the third column, tilt angle and axis values are reported.

Q1. What are the essential parameters to obtain the correct CTF? What would

happen if we changed the voltage of the microscope or the defocus value?

Q2. Are astigmatic images suitable for data analysis? Is there any advantage of collecting astigmatic images for image processing?

3.1 Manual Selection of TMV Particles

We now want to crop helical particles from the micrograph. For this purpose helices can be interactively picked using external programs such as EMAN's HELIXBOXER (Tang et al. 2007) or BOXER (Ludtke et al. 1999). An example from EMAN's heliboxer.py is provided below:

Open a new terminal (you can leave the SPRING's GUI open) and type:

```
% e2heliboxer.py --gui path-to-mrc-file
```

From the new appeared GUI click on File→Open and select the four TMV micrographs (tmv_m001.mrc tmv_m002.mrc, tmv_m003.mrc and tmv_m004.mrc). The four micrographs will appear as four entries of the table in the bottom part of the GUI. If you click on one of the two entries the specified micrograph will appear (Figure 4).

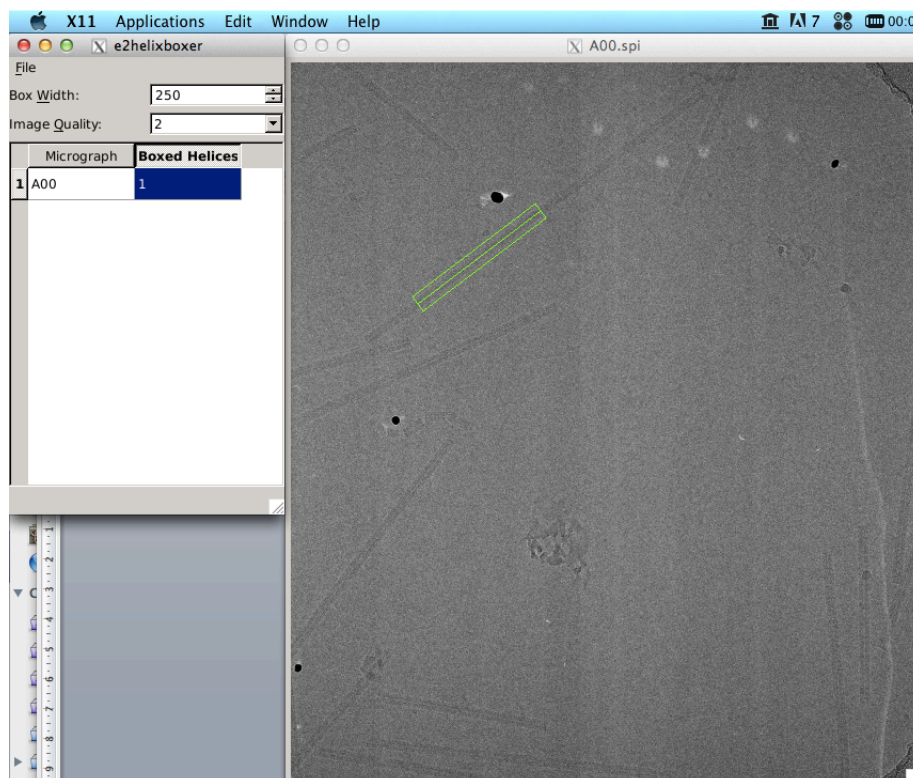


Figure 4. Cryo-micrographs of TMV displayed using E2HELIXBOXER. The green rectangle is used to determine the start and the end of the helical particles.

You can now digitally select the viruses using a draggable rectangular box to define the start and end of the helix. Before starting the selection of the helices set the *box width* value to 170. If you want to change the magnification or other parameters Alt+Click (or click with the middle mouse button) on the image and select the value you prefer in the parameters panel. Continue with the selection of all TMV particles on the micrograph. You can delete helices with Shift+Click on the central axis of the box.

For each micrograph, save the coordinates of the helices that you have boxed with *File→Save*. In the pop-up window, un-tick all the options but *Helix coordinates* and save the .txt coordinate file with the same name of the corresponding micrograph (e.g. A00.spi → A00.box). Continue with the interactive selection for the second micrograph.

3.2 Automated Tracing with MICHELIXTRACE

SPRING contains a program for automated helical tracing called MICHELIXTRACE, described in (Huber et al., 2017). The user provides a reference image of the helix and receives the coordinates of traces per micrograph. The reference helix must be vertically aligned and centered. Such a helix reference can be obtained interactively, e.g. using E2HELIXBOXER, or from a class average analysis, e.g. using SPRING's SEGMENTCLASS.

At the initial stage, the micrograph and reference can be strongly binned by specifying a *binning factor*, e.g. by a factor of 4 or 8 to achieve better speed. Due to their symmetry, helical specimens possess layer lines in the Fourier domain. As the amplitude spectrum is invariant to translation of the object in real space, we can reliably determine the rotation θ by a rotational correlation function directly from the power spectra of the tile and the reference. According to the determined helix angle, we then rotate the tile and compute a cross correlation function to determine the shift relative to the tile center Δ including a cc-score. As a result of the procedure, the θ / Δ pair and a normalized cc-score is obtained for every tile.

Based on the determined θ and Δ values of each tile, cc-scores can be mapped back on the micrograph with much higher precision than the initial coarse tiling. The resulting micrograph map exhibits high correlation for paths where the reference is present and shows small correlations where the reference is absent. In the next step, the cc-map is thresholded using a statistical approach to separate helices from background. The background null distribution of cc values is estimated, corresponding to the cc values when no helices are in the micrograph. The user specifies the significance level 'Alpha threshold' between 0 and 1, which determines the absolute cc cutoff value. The significance level corresponds to the probability of getting this or a higher cc value when there is no helix at this pixel, thus being the probability of a false positive pixel. Values around 0.001 worked well.

The resulting binary map is thinned to one pixel width and branch points where helices crossing over one another are removed. Connected pixels are extracted from the micrograph. Traces shorter than a user-defined minimum helix length parameter are discarded as they often correspond to contaminations and non-helical protein, for example aggregates. Traces longer than a user-defined maximum are cut in half to ensure good fitting of the traces by polynomials of low order.

In a final pruning step, the distribution of straightness measures of helices is analyzed and outliers are discarded. The user can specify 'Pruning cutoff bending', which is the number of standard deviations the straightness measure can deviate from the mean. We found the value 2 to be reasonable. Please note that pruning is just possible if the dataset contains at least 100 traces, as the distribution of straightness measures can be reliably analyzed for larger populations. When more than 100 traces are present, MICHELIXTRACE also provides an output about the persistence length of the traced polymer by analyzing the falloff of dot product between tangent vectors along the chain. The results of this analysis are printed in the file 'PersistenceLength.pdf'.

For usage of tracing results in other software, we provide the parameter 'Box file coordinate step', which determines the distance between coordinates along the helix in the output box file. These files can be read into other software and the coordinates treated as particles.

On the SPRING GUI, click on the drop-down menu on the top left and select SPRINGMICROGRAPH. Find the MICHELIXTRACE tab (accessible through the "Expert" mode).

Select all micrographs as well as a vertically aligned and centered reference. For this tutorial, we provide 'reference.hdf', cut using E2HELIXBOXER. A *binning factor* of 8 provides sufficient accuracy for tracing TMV. Set the *tile size* to 500 Å and the *tile overlap* to 80%. The estimated helix width is 190 Å and *Alpha threshold cc-map* is 0.0001. *Minimum and maximum helix length* is (500, 1500) and *pixelsize* is 1.062 Å.

Press OK to start a run.

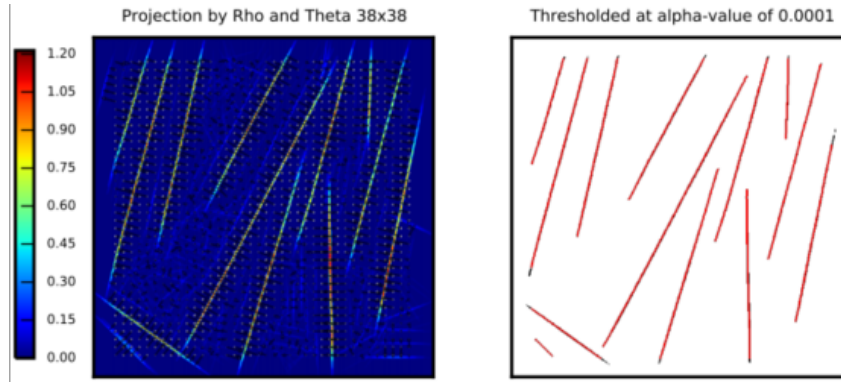


Figure 5. Diagnostic plot of MICHELIXTRACE, showing a cc-map with vectors for the contribution from every tile, as well as the thresholded cc-map.

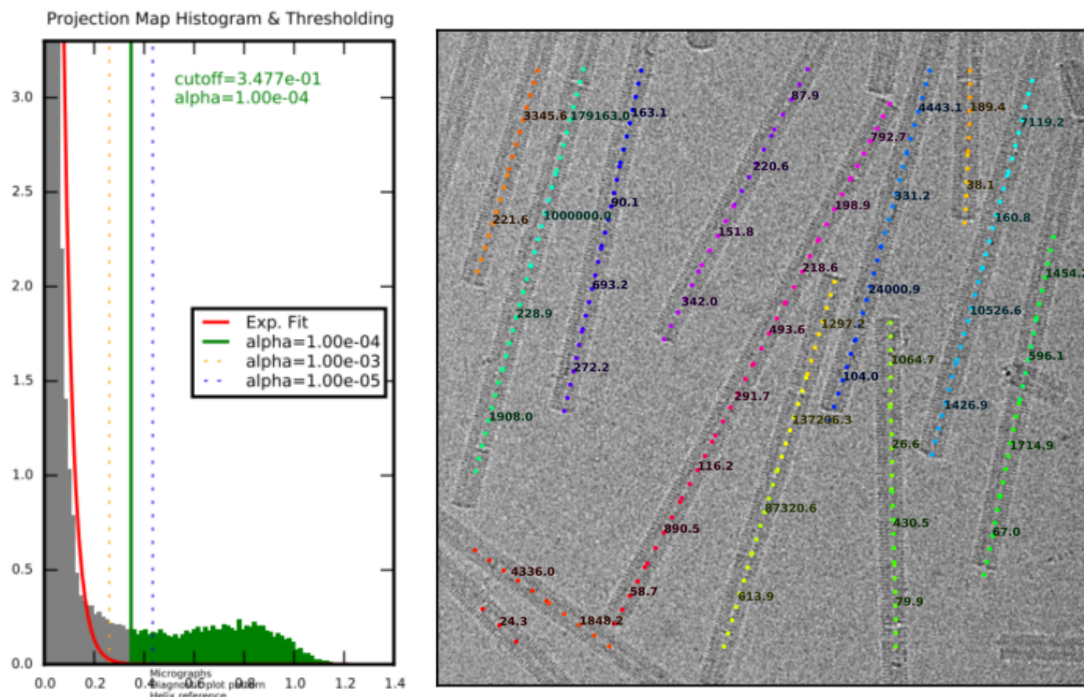


Figure 6. Diagnostic plot of MICHELIXTRACE, showing the statistical thresholding procedure using the parameter ‘Alpha threshold’. Alternative parameter choices are indicated by dotted lines. The right image shows the micrograph with an overlay of the detected traces. The numbers denote a straightness measure for every trace.

The statistical thresholding procedure assumes that the micrograph consists mainly of background, and estimates the background distribution of cc-values. In case, the user wants to have control about an absolute cutoff of the cc-values, there is the option to choose ‘Absolute threshold option cc-map’.

The tracing result depends mainly on the choice of threshold and minimum helix length cutoff. The threshold controls if traces are properly separated from background, dirt and noise in the cc-map. The minimum helix length cutoff excludes small traces from the result, since it is likely to arise from contaminations in the micrograph. We offer a systematic way of searching for the optimal parameters using the ‘Parameter search option’. For this, a small number of micrographs (~10) must

be carefully hand-traced, e.g. using e2helixboxer. The resulting box-files are given as an input to MICHELIXTRACE in the parameter search mode. The above mentioned two parameters are systematically tested and a diagnostic output plot is generated that depicts parameter spaces for precision, recall and an F-measure that corresponds to an effective combination of precision and recall.

Figure 7 shows diagnostic plots of a parameter search, as reported in (Huber et al, 2017). For a user, these plots can give a good estimate of the parameter range to use.

Recommended parameters for MICHELIXTRACE are: *Minimum helix length* from 300 to 1000 Å, *Maximum helix length* at least 2.5 times *Minimum helix length*, *Tile size* 400 to 700 Å, *Tile overlap* ~75 to 85 %, *Alpha threshold* from 0.1 to $1e-7$, *Pruning cutoff bending* from 1 to 3 and *Binning factor* 4 or 8.

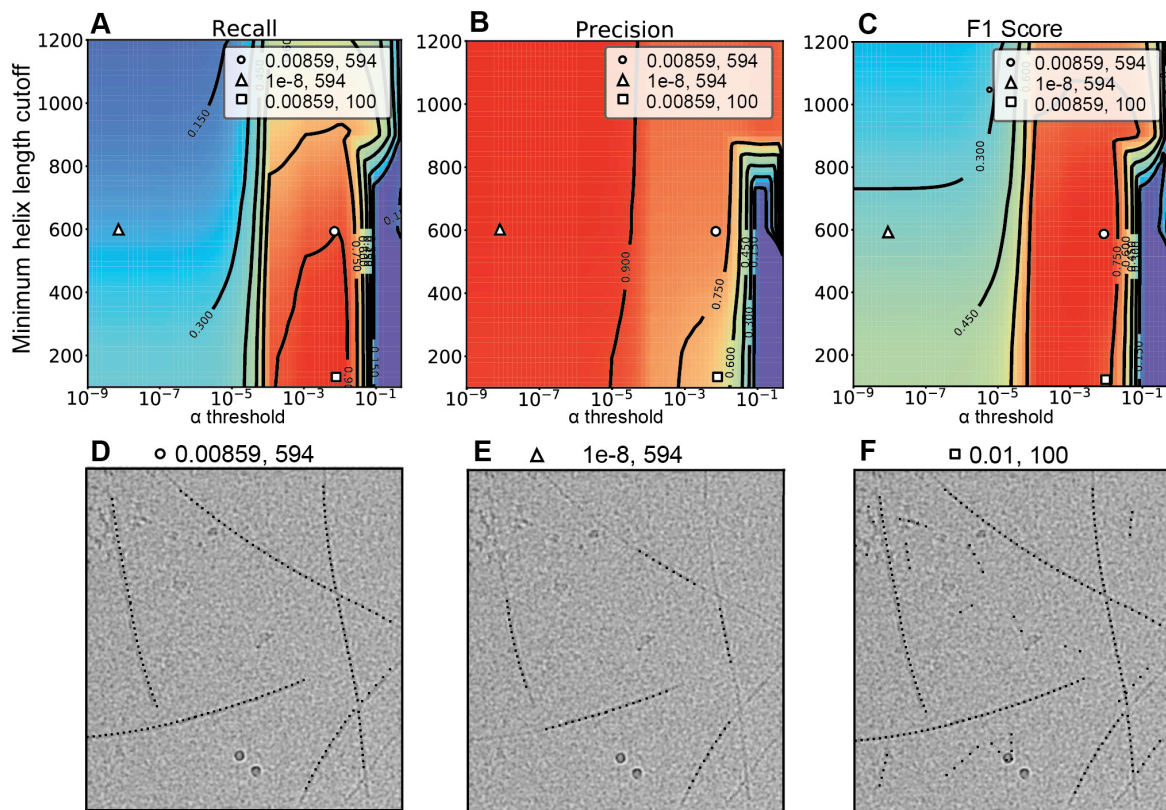


Figure 7. Extracts from diagnostic plots of MICHELIXTRACE in parameter search mode, for a dataset of 100 ParM micrographs (Huber et al., 2017). Tracing results from three different parameter choices are shown.

4. Segmentation of TMV particles

The selected TMV particles from different micrographs will be extracted into a single stack of overlapping segments. The size of the segments is a critical parameter for the alignment that will be used for 3D refinement (step 8/9). The larger the segment size the more mass and signal is present in the images but the more deviation from an ideal helical structure due to bending is expected.

In addition, the program also performs the correction of the microscope parameters from the CTF. We have two possible options: first phase-flipping that corrects the

phases at the negative values of the CTF. Second, convolution (or multiplication in Fourier Space) of the segment will be performed with the previously determined CTF. Phase inversion is recommended for low-resolution analysis such as classification while CTF multiplication is the recommended option for high-resolution analysis because of the implied weighting scheme. Since we're going to do both analyses, we need to generate two separate stacks of segments.

On the SPRING GUI, click on the drop-down menu on the top left and select SPRING2D. Find the SEGMENT tab.

Select all micrographs and corresponding segment coordinates (.box files) in the current directory. Set the *segment size* to 350 Å. The estimated helix width is 190 Å, and the *step size* of segmentation is set to 70 Å.

Stack for classification

Select *phase-flip* as a CTF correction and the *perturb step option* (the reason why we need to use this option will become apparent during the classification).

Stack for reconstruction

Tick the *CTF correct option* and select *CTFFIND*, and then *Convolve* in the next drop-down menu. Tick also the *Astigmatism correction*, and the *Invert*, *Normalization* options.

Q1. How is the step size parameter determined? How is it related to the overlap in percentage?

5. Analysis of Extracted Segments

We can quickly obtain information on the dimensions and regularity of the helix by averaging of segments using only the in-plane orientation information that was derived from the interactive selection procedure. In real space, the intensity profile reveals the width of our object. In Fourier space, the presence of layer lines in the sum of in-plane rotated power spectra from the segments indicates the existence of a helical lattice. In particular, the reduction of the power spectrum into a “collapsed” one-dimensional profile by averaging the spectrum along the layer lines is useful to assess the existence of weak layer lines. If we had to determine the helical symmetry, we could use the layer lines to derive it by indexing as it is usually done in Fourier-Bessel procedures.

SEGMENTEXAM (accessible through the “Expert” mode) generates a diagnostic plot that shows the segment average in real space and Fourier space. The intensity profile of our object can later be used to define its inner and outer radius.

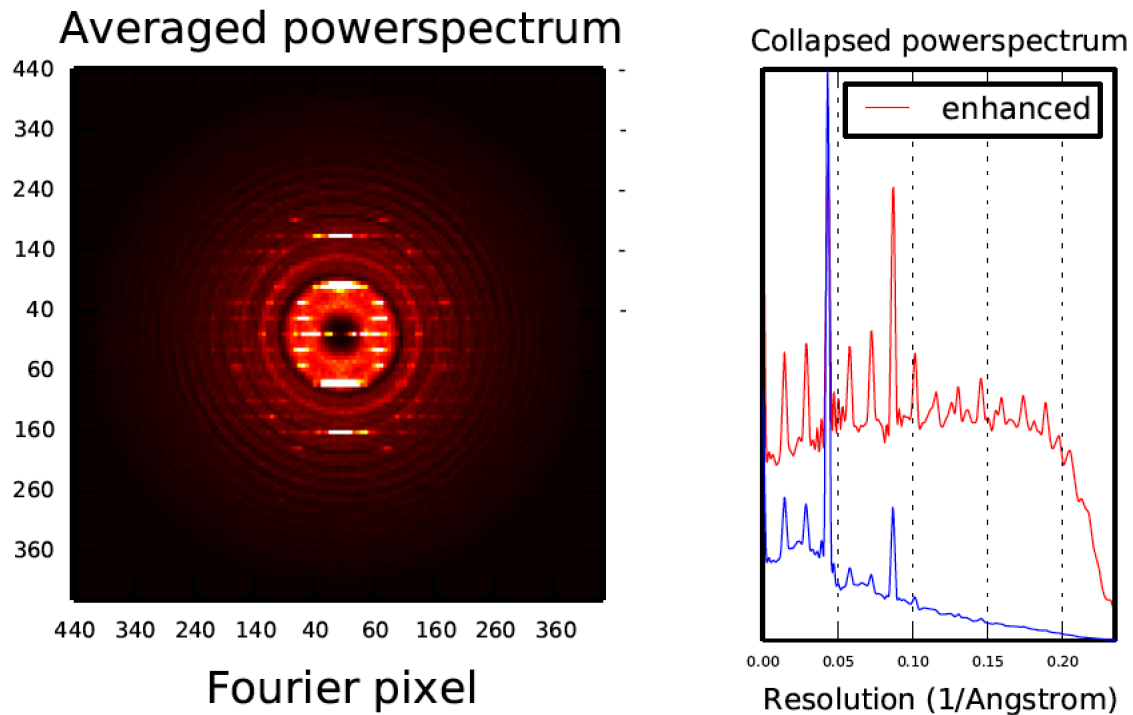


Figure 8. Diagnostic plot detail showing the average of segments in Fourier space (left) as the sum of their power spectra, and the collapsed 1-dimensional profile (right).

SEGMENTPLOT is a second tool to analyze the properties of the segments that we have extracted from the helices. Different quantities – such as the in-plane rotation, the phi angle, etc. – can be chosen and plotted against the distance along the helix. Plots for each helix or micrograph can be easily explored interactively and reveal how flexible or regular a specimen is.

Q1. What is the average width of the TMV filaments?

Q2. How many layer lines can you identify? What resolution do they match?

6. Classification

Classification of the segments produces averaged high-signal projection images of the specimen and may be useful to determine more details on the dimension and regularity of the specimen. Principal component analysis (PCA) is a routine analytical procedure for the analysis of single particles. Here, we use a k-means clustering algorithm from SPARX to compute class averages of segmented helices. Since the process of segmenting helices for classification-based averaging using a regular step size bears the risk of introducing artificial frequency components, for the classification procedure we must choose the first of the two stacks that we have generated, that was not segmented by a regular step size but by using a randomized step between 0 and the desired step size (the *perturb step option*).

Filters:

A low-pass filter is a filter that removes high frequencies *beyond* a certain threshold (i.e, the low frequencies pass). The high-pass filter is a filter that removes frequencies *lower* than a certain threshold. Since we use segmented helices that span an entire segment length we don't need to apply such a filter.

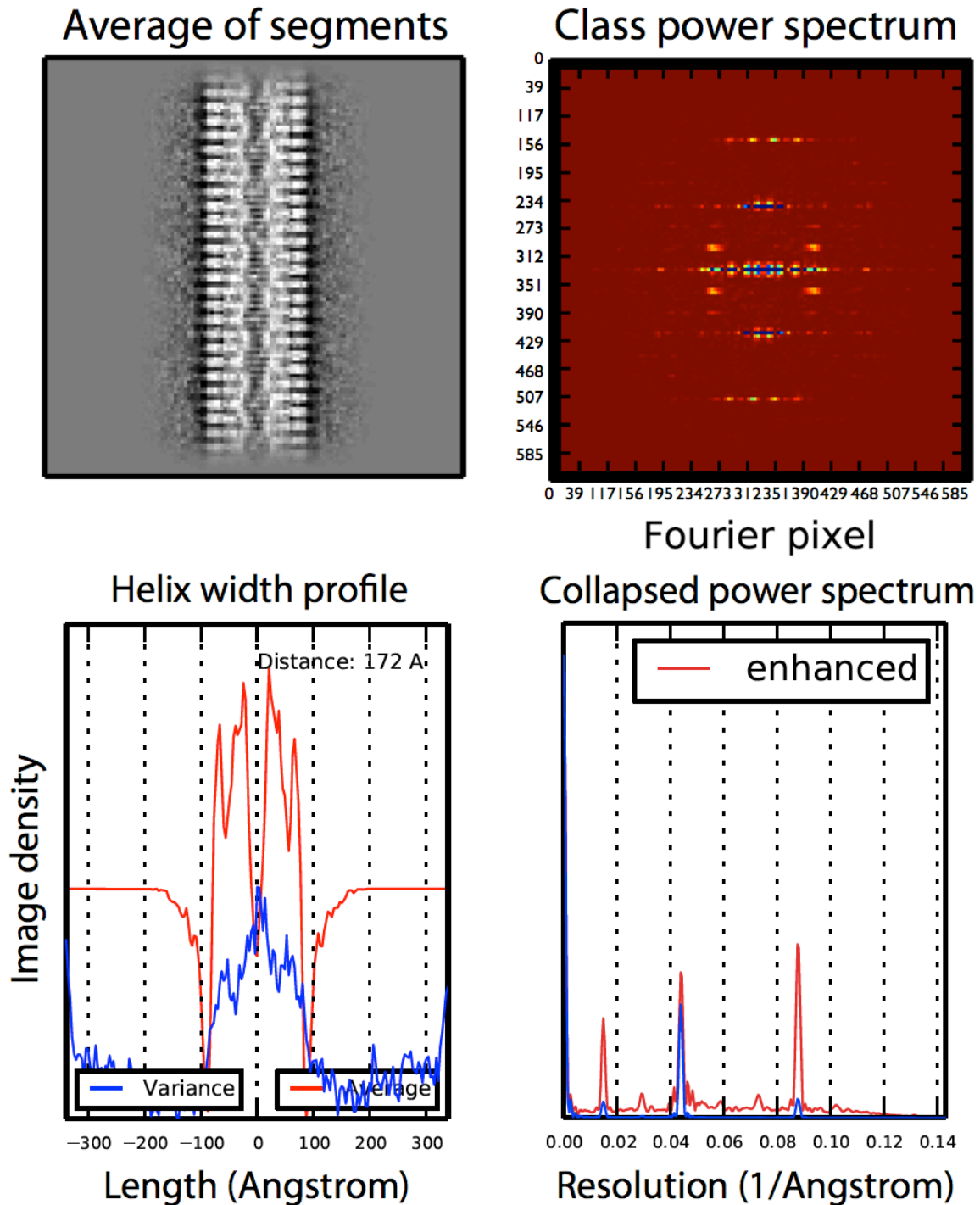


Figure 9. Diagnostic plot detail showing the average of segments in real space (top left), the helix width profile (bottom left), the Fourier transform of the average (top right) and its collapsed power spectrum (bottom right; no enhanced: blue line; enhanced: red line).

7. Class analysis in real and Fourier space

The class averages contain higher signal amplitude and phase information when compared with the raw segments. Similar to SEGMENTEXAM, a plot will be generated for each selected class that shows the class-averages in real space and Fourier space (Figure 9: left and right hand side of the panel, respectively). The images can be used for more detailed interpretation of the regular features and the determination of the symmetry parameters. First, class averages have improved signal-to-noise ratio in comparison with raw images. Second, classification can separate different symmetries from a heterogeneous data set and improve the average of interest. Third, in comparison with the added power spectra from SEGMENTEXAM class averages have the additional advantage that they contain phase information and can therefore be used to determine whether Bessel orders are odd or even. In addition to the class analyses, SEGCLASSLAYER allows the interactive exploration of layer lines from class averages.

Q1. What is the highest resolution layer line visible in the collapsed power spectrum? What resolution does it correspond to?

Q2. Given your pixel size, what is the highest achievable resolution? Would it be theoretically possible to attain this resolution with the classification only?

Q3. What is the B-factor compensating for (see the enhanced power spectrum)?

8. Symmetry determination

A. Class-based helical reconstruction and symmetry exploration

Once an initial guess about possible symmetry parameters has been formulated, we can test a large number of them by using one or more chosen class averages. The program SEGCLASSRECONSTRUCT (SPRING3D) generates a low-resolution 3D model starting from a 2D class average, that gets duplicated, shifted and rotated along and around the helical axis by a factor dictated by the specific symmetry to be tested. The agreement between the reprojections of the 3D model and the starting class average can be quantitatively evaluated using the program SEGGRIDEXPLORE that plots amplitude correlation, cross-correlation, variance and other relevant values (Figure 10).

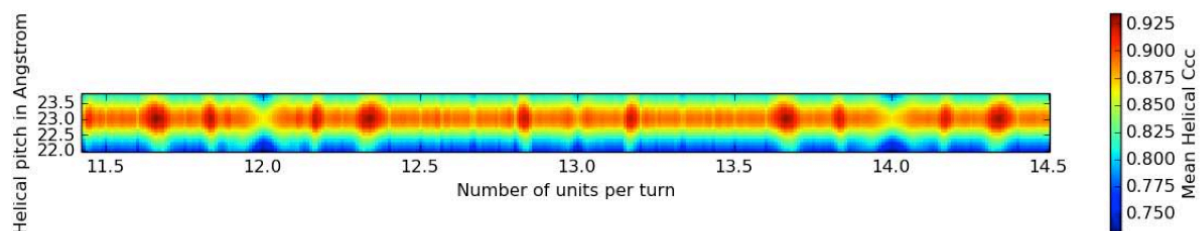


Figure 10. Diagnostic plot showing the mean helical cross correlation between a chosen class average and the reprojection of different 3D volumes generated from the class average using the symmetry parameters shown in the plot. This plot contains more than 5000 combinations. Note the characteristic periodicity of the correlation profile that repeats every $n+2$ units per turn.

If the symmetry is searched by varying the helical pitch from 22.0 to 24.0 Å in 0.1 Å steps and the number of units per turn from 10.00 to 20.00 in 0.01 steps (for a total of 20000 combinations), only few reconstructions will produce a well-matching 3D reconstructions and ‘good’ solutions. The segments can be now used to generate corresponding high-resolution, refined 3D models. Together with the layer line analysis, the quality of the EM map and ultimately the quality of the fit of an atomic model into the density will allow to decide which one is the best solution.

B. High-resolution symmetry refinement

We should now be able to generate 3D reconstructions for a number of symmetry pairs corresponding to the best solutions obtained with SEGCLASSRECONSTRUCT. The plot tells us that, while we have only one pitch maximum at ~23 Å (compatible with the class averages), we have several maxima for the number of subunits per turn, at n_1 , n_1+2 , ... and n_2 , n_2+2 ,... where $n_1=13.66$ and $n_2=14.34$. Therefore, using the CTF-corrected stack, symmetry parameters and the gathered dimension information, we can perform a 3D structure refinement imposing symmetries at a pitch of 23 Å and 13.66, 14.34, 15.66, 16.34, 17.66 and 18.34 subunits per turn (Figure 10). To generate the corresponding 3D structures, use the program SEGREFINE3DGRID (Desfosses et al. 2014). In order to visually analyze the refinements according to a number of criteria, e.g. amplitude correlation etc. use SEGGRIDEXPLORE.

9. 3D Structure refinement

The refinement procedure cycles through a sequence of projection, alignment and reconstruction to improve the 3D structure until convergence is achieved. The essential step is the assignment of five orientation parameters (3 Euler angles and translation x/y) for each segment to projections and to combine them into a 3D reconstruction. If no initial reference is available a cylinder with the specified dimension is sufficient to iteratively improve the structure until convergence.

The refinement will require the unrotated, CTF corrected raw segments. For every iteration cycle, a 3D structure and two series of diagnostic plots will be generated. First, statistics from the iterations of the refinement process can be found (Figure 11). The plot reveals information on the centering of the helices (top left), the polarity distribution, the number of asymmetric units after symmetrization (center right) and the Fourier Shell Correlation (bottom left), which is used as the achieved resolution of your model.

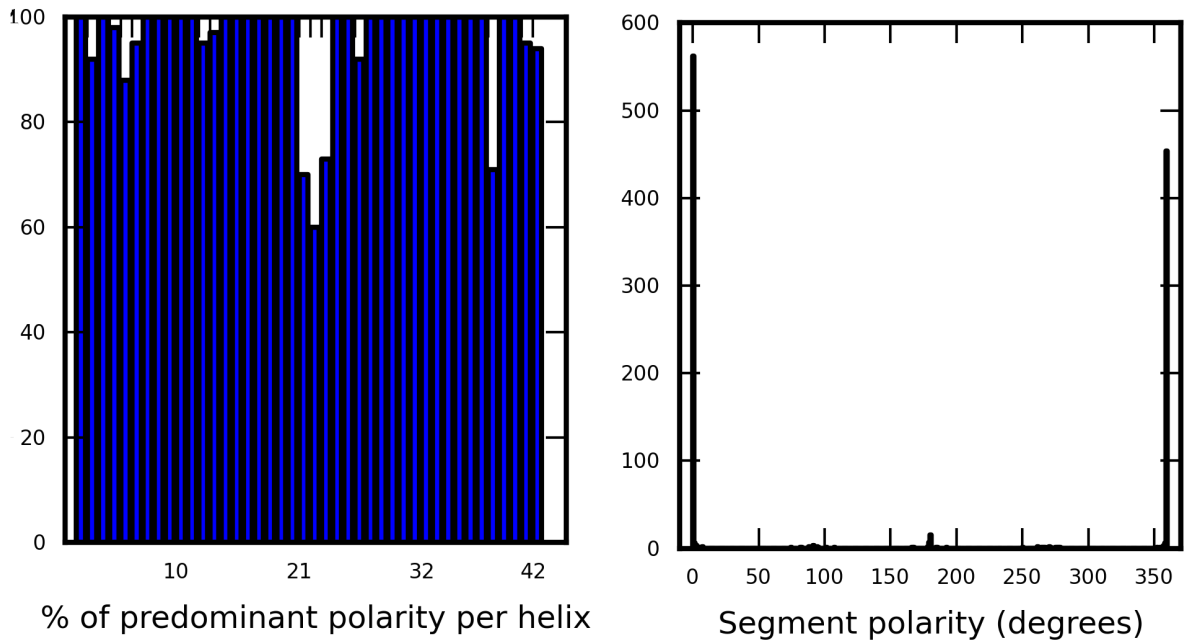


Figure 11. Polarity statistics from the final iteration cycle of the refinement procedure.

The second series of plots compares observed and simulated projections in Fourier and real space. The upper side-by-side panel displays an average power spectrum of the assigned segments next to a power spectrum from a reprojection of the 3D structure (Figure 10). The layer lines give a resolution estimate and should match up in both spectra if the symmetry was correctly entered.

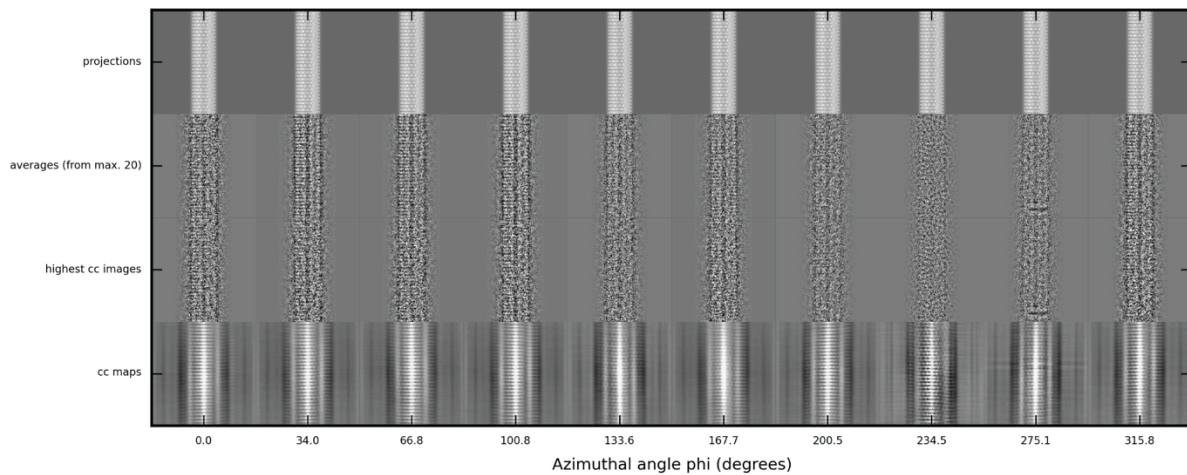


Figure 12. Correlation between the model projections and the experimental projections. The gallery displays reprojections of the refined 3D model and assigned segments as sums and individual images. The lowest row represents the cross-correlation maps of the projections and the aligned segment. The maps are a useful way to judge the accuracy of the alignment. The sharper the peaks the better the alignment can be defined.

Even though the analysis of the plots can be revealing and tells us whether a

reconstruction (and therefore a chosen symmetry pair) is wrong, it will never suffice to conclusively tell that it is right. Since an atomic structure of the four-helix bundle is available, we can try to fit it into the EM density (PDB code 4UDV, 2OM3, 2TMV) (Fromm et al. 2015). As a result of this fitting exercise, we will find out that only the 3D reconstruction with 16.34 numbers of subunits per turn imposed is compatible with the known four-helix bundle structure, whereas the others are not.

Q1. Is there a good agreement between your observed and reprojected power spectrum?

Q2. What is the initial number of segments and the one after the symmetrization? What is the relation between the two values?

Q3. Given your pixel size, which is the highest achievable resolution? Would it be theoretically possible to attain this resolution?

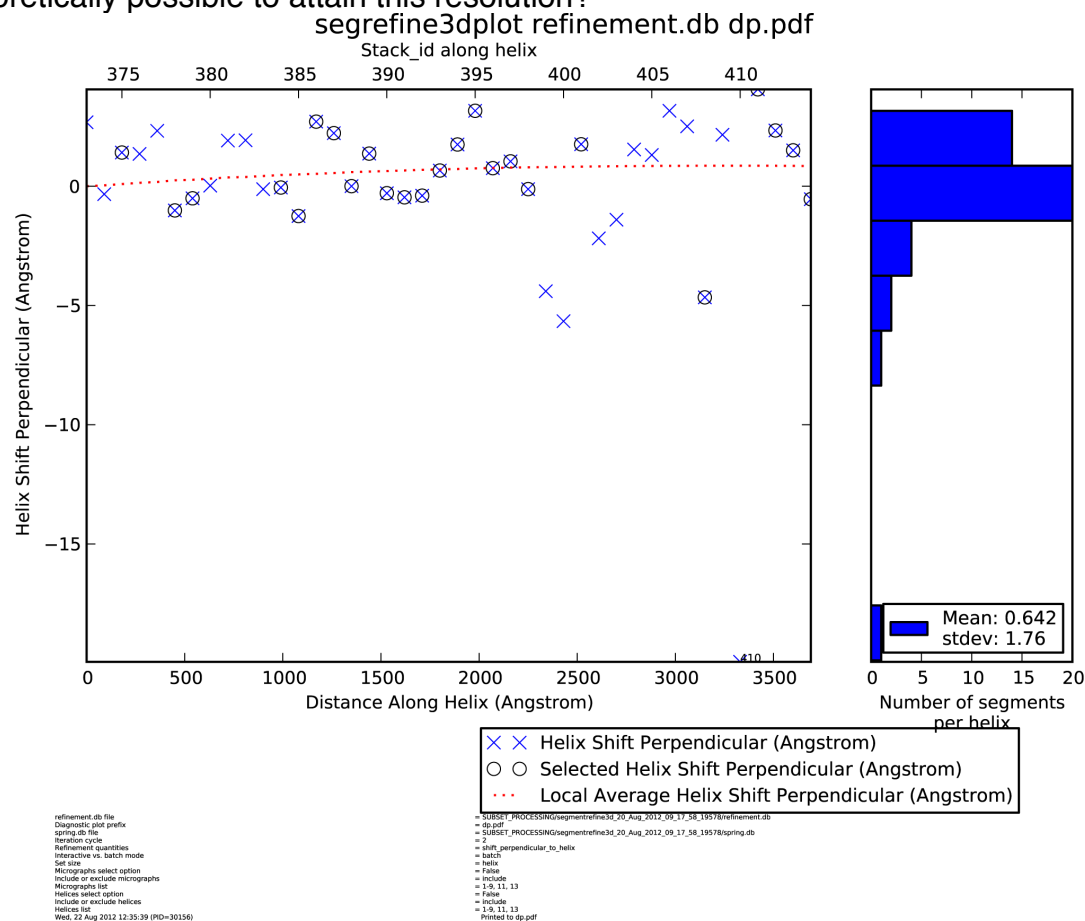


Figure 13. Plots showing the segment distribution perpendicular to the helical axis on a scatter plot (left) and on a histogram (right).

10. 3D structure analysis

We have now the opportunity to analyze the alignment in greater detail. We can interactively browse and plot the properties of the segments used for the reconstruction using the program SEGMENT3DPLOT using a similar interface like SEGMENTPLOT.

In each plot both the initial segments and the one that have been selected for the specified iteration step will be visualized (Figure 13). Load the refinement.db and the spring.db files, specify the quantity you want to look at and launch the interactive session by clicking on the “OK” button.

Using SEGREFINE3DINSPECT, we will finally explore our reconstructed structure. Once you have filled in the parameters and launched the software, a two-panel window will appear. The two panels are just two different visualization modes that you can use to analyze the structure.

On the left-hand side the structure is visualized in a Cartesian coordinate system with the helix axis aligned with the z-axis. You can switch from one axis to the other using the drop-down menu on the top right and slice through the selected axis with the slide bar next to it. On the right-hand side, the structure is visualized in cylindrical coordinates so you can inspect different radial slices with changing the radius r .

You can inspect what your structure looks like with a resolution cutoff and a B-factor applied. The B-factor is a correction factor that compensates the decay of high-resolution amplitudes. Specific features of the helix can be sharpened or blurred using a specific resolution cutoff combined with a B-factor correction. If you want to save a model to which these corrections have applied, click *Save* and visualize it in 3D either with chimera or e2display.

Q1: What do you see when you increase or decrease the B-factor?

Q2: What do you see when you change the resolution cutoff?

11. Frame processing with DED data

The advent of direct electron detectors (DED) opened up new processing strategies to retrieve more structural information from cryo-EM images. New cameras come with an improved signal-noise-ratio and a faster read-out rate, which enables the fractionation of one exposure into multiple frames (movies). These advances in hardware technology allow image-processing based correction of specimen movement (stage drift or beam induced motion) and to limit radiation damage accumulating during image acquisition. The first routine to account for the specimen movement is the abovementioned alignment of subsequent frames in one movie (Li et al. 2013).

Another strategy to address specimen movement, is to treat individual segments from frames, or running averages of consecutive frames, as individual exposures in the structure refinement procedure. This effectively shortens the exposure time and thereby limits specimen movement. At the same time, one can limit radiation damage by excluding frames after a certain accumulated dose (e.g. $20 \text{ e}^-/\text{\AA}^2$). In contrast to the whole-frame alignment strategy, this type of motion correction accounts for local differences of motion in the same movie and can also correct for segment rotations in addition to translations.

To make optimal use of movie data in SPRING the following processing strategy is suggested:

1. Complete reconstruction of a 3D model from integrated (aligned) movies as described in this tutorial (1-9).
2. Create new segment stack from individual frames with same helix coordinates used for the integrated images with the program SEGMENT (4). (Dose limitation can be performed at this stage by considering only a certain frame range for segmentation.)
3. Local alignment and 3D reconstruction with the newly created stack based on alignment of segments from the integrated images with the program SEGMENTREFINE3D (9). Segments from frames can be treated entirely independently or as running averages over frames. In addition, movement of neighboring segments within a helix can be locally averaged. Dose limitation can be applied at this stage to minimize radiation damage effects.

To follow up the reconstruction obtained during this tutorial so far, two additional steps are required to run movie processing with DED data:

A. Segmentation of helices from movies

In principle, segmentation is the same as the one described at (4), with the difference of using individual frames instead of the integrated images. Movies containing individual frames need to be available as image stacks in the .mrscs format. To distinguish .mrc files with only on image from stacked .mrc files, the movies (image stacks) need to have the .mrscs extension with the exact same prefix name as the integrated image used for the first segmentation. For example:

Name of integrated image:	tmv_m001.mrc
Name of movie (image stack):	tmv_m001.mrscs

Other movie related options in SEGMENT are:

<p>For the segmentation of helices from movies using SEGMENT:</p> <ol style="list-style-type: none"> 1. Tick the <i>Frame processing option</i>. 2. Specify the <i>First and last frame</i> of the movies to be considered for segmentation (the stack starts with 0 as first frame). 3. Provide the last <i>spring.db</i> and <i>refinement.db</i> databases from the SEGMENTREFINE3D run with the integrated images. As this segmentation builds on the previous reconstruction made with the integrated images, the <i>spring.db</i> and <i>refinement.db</i> databases need to be updated. 4. Provide the same <i>spring.db</i> as <i>Segment coordinates</i>.
--

SEGMENT will generate a new segment stack from the segmentation of movies in addition to updated *spring.db* and *refinement.db* databases (will be named *_x-yframes.db*). All segment attributes from the integrated images are transferred to the

respective frames, which are needed for local alignments and subsequent 3D structure refinement.

B. Local alignment and 3D structure refinement

The generated segment stack from the movies can now be used for 3D structure refinement. As the alignment and 3D structure refinement has already been completed with helices from integrated images and only small changes in alignment are expected between the integrated and the frame images. Therefore, it is essential to base 3D structure refinement with movie helix stack on previous refinement parameters by continuing the refinement from previous runs and activating only the *Maximum resolution aim* in the SPRING 3D refinement routine (while skipping the lower resolution aims).

1. Specify the segment stack generated from the movies in SEGMENTREFINE3D.
2. Tick the *Reference structure option*, specify the updated *spring.db* and *refinement.db* databases generated during segmentation of the movies and tick the *Continue refinement option*. Use the latest volume from the 3D refinement with the integrated images as a *Reference volume*. (If you don't provide a reference volume here, SPRING will re-generate one according to the determined orientation parameters provided in the *refinement.db*.)
3. Only activate the *MaxR – Maximum resolution aim* and lower the number of iterations accordingly (e.g. 2-4 iterations are sufficient to perform local refinement).

In SEGMENTREFINE3D there are several ways to process frames from helices. First, one can treat all helix segments from individual frames as independent segments and perform a local alignment based on the attributes assigned to the respective segment from the integrated image. However, this might lead to poor alignments due to the low contrast present in segments from individual frames (due to low electron doses in each individual frame). Second, to improve the effective signal, first SPRING can generate running averages of consecutive frames, which will then be used for the alignment. Third, SPRING can take advantage of the fact that neighboring segments from the same helix most likely undergo nearly the same motion during one exposure. In this option, one can specify a distance within which the movement of segments of a helix will be averaged. As the projected helix density decreases for example for thinner filaments improving the signal in frame images is recommended. For the used test data set of TMV including 7 frames distributed over $31 \text{ e}^-/\text{\AA}^2$, we found that individual frames could be aligned to sufficiently high accuracy to improve the resolution (Fromm et al. 2015).

Optional:

Tick the *Frame motion correction* option in SEGMENTREFINE3D and specify the *Frame average window size* and *Frame local averaging distance*.

Q1: Do you improve your refinement?

Q2: What happens if you make use of additional frame running averaging and local averaging?

REFERENCES

- Bawden, F.C. et al., 1936. Liquid crystalline substances from virus-infected plants. *Nature*, 138, pp.1051–1052.
- Desfosses, A. et al., 2014. SPRING - an image processing package for single-particle based helical reconstruction from electron cryomicrographs. *Journal of structural biology*, 185(1), pp.15–26.
- Fromm, S.A. et al., 2015. Seeing tobacco mosaic virus through direct electron detectors. *Journal of structural biology*, 189(2), pp.87–97.
- Hohn, M. et al., 2007. SPARX, a new environment for Cryo-EM image processing. *Journal of structural biology*, 157(1), pp.47–55.
- Huber, S.T. et al., 2017. Automated tracing of helical assemblies from electron cryomicrographs. *Journal of Structural Biology*, (September), 0–1.
- Li, X. et al., 2013. Electron counting and beam-induced motion correction enable near-atomic-resolution single-particle cryo-EM. *Nature methods*, 10(6), pp.584–590.
- Ludtke, S., Baldwin, P. & Chiu, W., 1999. EMAN: semiautomated software for high-resolution single-particle reconstructions. *Journal of structural biology*, 128(1), pp.82–97.
- Mindell, J.A. & Grigorieff, N., 2003. Accurate determination of local defocus and specimen tilt in electron microscopy. *Journal of structural biology*, 142(3), pp.334–347.
- Tang, G. et al., 2007. EMAN2: an extensible image processing suite for electron microscopy. *Journal of structural biology*, 157(1), pp.38–46.

Appendix

Tables of the input parameters and expected computation times

Parameters not specified here can be left default.

1. EXAMINATION OF MICROGRAPHS

MICEXAM

Micrographs	= tmv_m001.mrc, tmv_m002.mrc, tmv_m003.mrc, tmv_m004.mrc
Diagnostic plot pattern	= scanexam_diag.pdf
Pixel size in Angstrom	= 1.062

Exp. computation time: 43" on 1 cpus

2. DETERMINATION OF THE CONTRAST TRANSFER FUNCTION

MICCTFDETERMINE

Micrographs	= tmv_m001.mrc, tmv_m002.mrc, tmv_m003.mrc, tmv_m004.mrc
Diagnostic plot pattern	= micctfdetermine_diag.pdf
Spherical aberration	= 2.7
Electron voltage in kV	= 300
Pixel size in Angstrom	= 1.062
Range of defocus in Angstrom	= (10000, 45000)
Defocus search step size	= 300
Astigmatism search restraint in Angstrom	= 2000
Tile size power spectrum in Angstrom	= 544
Resolution search range in Angstrom	= (20.0, 6.0)
CTFILT refine option	= False
Binning option	= True
Binning factor	= 2
MPI option	= True
Number of CPUs	= 4
Temporary directory	= /g/embo2016/tmp

Exp. computation time: 5'07" on 4 cpus
--

3 AUTOMATED TRACING OF TMV PARTICLES

MICHELIXTRACE

Micrographs	= tmv_m001.mrc, tmv_m002.mrc, tmv_m003.mrc, tmv_m004.mrc
Diagnostic plot pattern	= michelixtrace_diag.pdf
Helix reference	= reference.hdf
Estimated helix width in Angstrom	= 190
Pixel size in Angstrom	= 1.062
Binning option	= True
Binning factor	= 8
Tile size power spectrum in Angstrom	= 500
Tile overlap in percent	= 80
Alpha threshold cc-map	= 0.0001
Order fit	= 2
Minimum and maximum helix length	= (500, 1500)
Pruning cutoff bending	= 2.0
MPI option	= True
Number of CPUs	= 4
Temporary directory	= /g/embo2016/tmp

Exp. computation time: ~20" on 4 cpus

4. SEGMENTATION OF TMV PARTICLES

A. Segmentation for classification

SEGMENT

```
Micrographs                = tmv_m001.mrc, tmv_m002.mrc, tmv_m003.mrc, tmv_m004.mrc
Image output stack         = tmv_4class.hdf
Spring database option     = True
spring.db file             = spring.db          #from previous CTF determination
Segment coordinates        = tmv_m001.box, tmv_m002.box, tmv_m003.box, tmv_m004.box
Segment size in Angstrom   = 350
Estimated helix width in Angstrom = 200
Step size of segmentation in Angstrom = 70
Perturb step option        = True
Pixel size in Angstrom     = 1.062
CTF correct option         = True
CTFFIND or CTFTILT        = Ctffind
convolve or phase-flip    = Phase-flip
Astigmatism correction     = True
Binning option             = True
Binning factor             = 2
Invert option              = True
Normalization option       = True
Remove helix ends option   = True
Rotation option            = True
MPI option                 = True
Number of CPUs             = 4
Temporary directory        = /g/embo2016/tmp
```

Exp. computation time: 2'22" on 4 cpus

B. Segmentation for 3D refinement

SEGMENT

```
Micrographs                = tmv_m001.mrc, tmv_m002.mrc, tmv_m003.mrc, tmv_m004.mrc
Image output stack         = tmv_4class.hdf
Spring database option     = True
spring.db file             = spring.db
Segment coordinates        = tmv_m001.box, tmv_m002.box, tmv_m003.box, tmv_m004.box
Segment size in Angstrom   = 350
Estimated helix width in Angstrom = 200
Step size of segmentation in Angstrom = 70
Perturb step option        = False
Pixel size in Angstrom     = 1.062
CTF correct option         = True
CTFFIND or CTFTILT        = Ctffind
convolve or phase-flip    = convolve
Astigmatism correction     = True
Binning option             = False
Binning factor             = 2
Invert option              = True
Normalization option       = True
Remove helix ends option   = True
Rotation option            = False
MPI option                 = True
Number of CPUs             = 4
Temporary directory        = /g/embo2016/tmp
```

Exp. computation time: 2'20" on 4 cpus

5. ANALYSIS OF EXTRACTED SEGMENTS

SEGMENTEXAM

```
Image input stack           = tmv_4class.hdf
Diagnostic plot             = segmentexam_diag.pdf
Power spectrum output image = power.hdf
Pixel size in Angstrom     = 1.062
Estimated helix width in Angstrom = 200
Power spectrum resolution cutoff in 1/Angstrom = 0.47
spring.db file             = spring.db      #from segmentation
MPI option                 = False
Temporary directory        = /g/embo2016/tmp
```

Exp. computation time: 3' on 1cpu

6. CLASSIFICATION

SEGMENTCLASS

```
Image input stack           = tmv_4class-2xbn.hdf
Class average stack         = averages.hdf
Reference image option      = False
Spring database option      = False
Class variance stack        = variances.hdf
Eigenimage stack           = eigenimages.hdf
Pixel size in Angstrom     = 2.124
Estimated helix width and height in Angstrom = (200, 350)
Number of classes          = 10
Number of iterations        = 3
Limit in-plane rotation     = True
Delta in-plane rotation angle = 10.0
X and Y translation range in Angstrom = (50, 50)
High-pass filter option     = False
Low-pass filter option      = True
High and low-pass filter cutoffs in 1/Angstrom = (0.001, 0.09)
Binning option             = False
Binning factor             = 2
MPI option                 = True
Number of CPUs             = 16
Temporary directory        = /g/embo2016/tmp
```

Exp. computation time: 8'27" on 16cpus on a HPC cluster

7. CLASS ANALYSIS IN REAL AND FOURIER SPACE

SEGCLASSEXAM

```
Class average stack         = multi_ref_avg.hdf
Diagnostic plot             = segclassexam_diag.pdf
Power spectrum output image = power.hdf
Pixel size in Angstrom     = 2.124
Estimated helix width in Angstrom = 200
Power spectrum resolution cutoff in 1/Angstrom = 0.23
Class number range to be analyzed = (0, 9)
MPI option                 = False
Temporary directory        = /g/embo2016/tmp
```

Exp. computation time: 1' on 1cpu

8. SYMMETRY DETERMINATION

A. Class-based helical reconstruction

SEGCLASSRECONSTRUCT

```
Class average stack         = multi_ref_avg.hdf
Volume name                 = recvol.hdf
Class number to be analyzed = 3
Estimated helix inner and outer diameter in Angstrom = (0, 190)
```

```

Pixel size in Angstrom                = 2.124
Helical rise/rotation or pitch/number of units per turn choice = pitch/unit_number
Range of helical rise or pitch search in Angstrom = (22.0, 24.0)
Range of helical rotation in degrees or number of units per turn search = (10.0, 20.0)
Increments of helical symmetry steps in Angstrom or degrees = (0.1, 0.01)
Keep intermediate files                = False
Montage stack of class average vs. reprojection images = mont_reprj.hdf
Montage stack of class average vs. reprojection power spectra = mont_power.hdf
Rotational symmetry                   = 1
Helix polarity                        = Polar
Out-of-plane tilt angle               = 0
Percent of image reconstruction size  = 50
Minimum number of images for 3D reconstruction = 8
Local symmetry grid search            = False
Center option                         = True
MPI option                           = True
Number of CPUs                       = 64
Temporary directory                   = /g/embo2016/tmp

```

Exp. computation time: 8hrs51' on 64 cpus on a HPC cluster (20000 symmetry pairs)

B. High-resolution reconstruction

SEGREFINE3DGRID

```

Image input stack refinement          = tmv_4rec.hdf
Output volume name                    = recvol.hdf
First parameter                      = helical_rise_or_pitch
Second parameter                     = helical_rotation_or_number_of_units_per_turn
Lower and upper limit first parameter = (23.1, 23.1)
Lower and upper limit second parameter = (13.66, 17.66)
First and second parameter increment  = (0.0, 2.0)
Subgrid option                       = True
Part and number of subgrids           = (1, 2)
Diagnostic plot prefix                = diagnostic_plot.pdf
Number of iterations                  = 9
spring.db file                       = spring.db
Assemble refinement strategy          = True
LR - Low resolution aim               = True
MR - Medium resolution aim            = True
HR - High resolution aim              = True
MaxR - Maximum resolution aim         = False
Estimated helix inner and outer diameter in Angstrom = (0, 190)
Pixel size in Angstrom                = 1.062
Helical rise/rotation or pitch/number of units per turn choice = pitch/unit_number
Helical symmetry in Angstrom or degrees = (23.1, 13.66)
Rotational symmetry                   = 1
Helix polarity                        = Polar
Image alignment size in Angstrom      = 350
Step size of segmentation in Angstrom = 70
MPI option                           = True
Number of CPUs                       = 32
Temporary directory                   = /g/embo2016/tmp

```

~3hrs30' on 32 cpus on a HPC cluster; this is 1/2 of the final grid

9. 3D STRUCTURE REFINEMENT

SEGMENTREFINE3D

```

Image input stack refinement          = tmv_4rec.hdf
Output volume name                    = recvol.hdf
Diagnostic plot prefix                = diagnostic_plot.pdf
Number of iterations                  = 16

```

spring.db file	= spring.db
Assemble refinement strategy	= True
LR - Low resolution aim	= True
MR - Medium resolution aim	= True
HR - High resolution aim	= True
MaxR - Maximum resolution aim	= True
Estimated helix inner and outer diameter	in Angstrom = (0, 190)
Pixel size in Angstrom	= 1.062
Helical rise/rotation or pitch/number of	units per turn choice = rise/rotation
Helical symmetry in Angstrom or degrees	= (1.408, 22.03)
Rotational symmetry	= 1
Helix polarity	= Polar
Image alignment size in Angstrom	= 350
Step size of segmentation in Angstrom	= 70
MPI option	= True
Number of CPUs	= 32
Temporary directory	= /g/embo2016/tmp

~4hrs22' on 32 cpus on a HPC cluster

11. FRAME PROCESSING

A. SEGMENTATION

SEGMENT

Micrographs	= tmv_m001.mrcs, tmv_m002.mrcs, tmv_m003.mrcs, tmv_m004.mrcs
Image output stack	= tmv_4rec_frames.hdf
Spring database option	= True
spring.db file	= spring.db
Segment coordinates	= spring.db
Segment size in Angstrom	= 350
Estimated helix width in Angstrom	= 200
Step size of segmentation in Angstrom	= 70
Pixel size in Angstrom	= 1.062
CTF correct option	= True
CTFFIND or CTFTILT	= Ctffind
convolve or phase-flip	= Convolve
Astigmatism correction	= True
Invert option	= True
Normalization option	= True
Remove helix ends option	= True
Rotation option	= False
Frame processing option	= True
First and last frame	= (0, 3)
Refinement.db to process	= refinement016.db
MPI option	= True
Number of CPUs	= 4
Temporary directory	= /g/embo2016/tmp

~8' on 4 cpus

B. 3D STRUCTURE REFINEMENT

SEGMENTREFINE3D

Image input stack refinement	= tmv_4rec_frames.hdf
Output volume name	= recvol.hdf
Diagnostic plot prefix	= diagnostic_plot.pdf
Number of iterations	= 2
Reference structure option	= True
Reference volume	= recvol_1062apix_016.hdf
spring.db file	= spring_0-3frames.db
Continue refinement option	= True
refinement.db file	= refinement_0-3frames.db

```

Assemble refinement strategy      = True
LR - Low resolution aim          = False
MR - Medium resolution aim       = False
HR - High resolution aim         = False
MaxR - Maximum resolution aim    = True
Absolute X and Y translation limit in Angstrom = (100, 100)
Frame motion correction           = False
Frame average window size        = 1
Frame local averaging distance   = 700
Estimated helix inner and outer diameter in Angstrom = (0, 190)
Pixel size in Angstrom           = 1.062
Helical rise/rotation or pitch/number of units per turn choice = rise/rotation
Helical symmetry in Angstrom or degrees = (1.408, 22.03)
Rotational symmetry              = 1
Helix polarity                   = Polar
Image alignment size in Angstrom = 350
Step size of segmentation in Angstrom = 70
MPI option                       = True
Number of CPUs                   = 64
Temporary directory              = /g/embo2016/tmp
  
```

~4hrs on 64 cpus on a HPC cluster

Micrograph-based motion correction

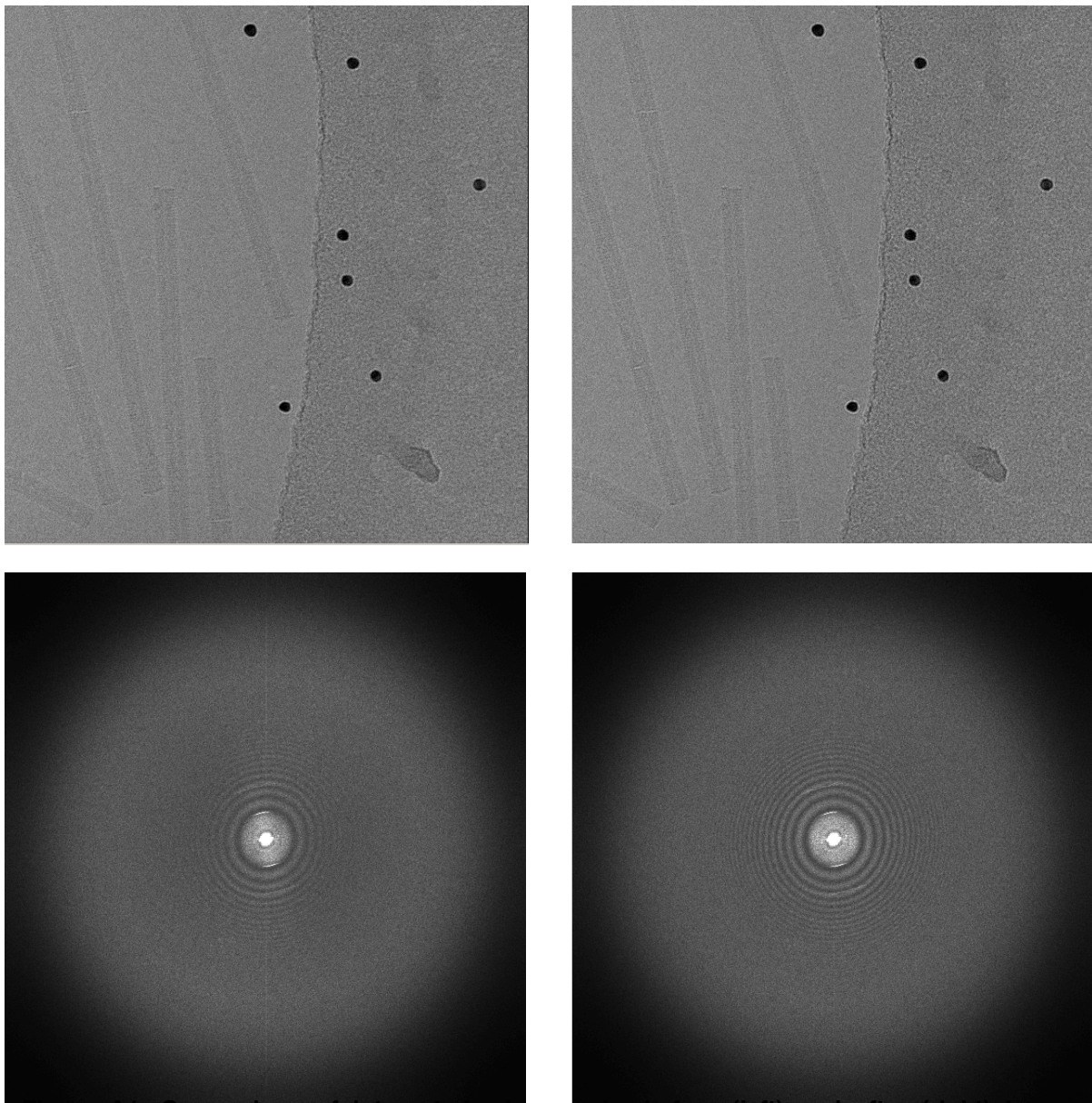


Figure A1. Comparison of integrated micrographs before (left) and after (right) beam-induced motion correction using *motioncorr* (Li et al.). The effect of specimen drift and drift correction after frame alignment is clearly visible in the power spectra of the images.