Protomo Subvolume Processing Tutorial

Version 3.1

Hanspeter Winkler

Contents

Revision 0 – 2022-10-31

1 Getting started

1.1 Sample data set

The data set in this tutorial was obtained from a section through a sarcomere of isometrically contracting asynchronous insect flight muscle, which was stained and plastic embedded Wu et al. (2009). The thin section shows a myac layer with alternating thick filaments (myosin) and thin filaments (actin) that run vertical in the image. The tilt series were acquired on a CM300 electron microscope with the Saxton scheme and recorded on a TVIPS Tem-Cam F224 (2K×2K). Two tilt series with approximately orthogonal tilt axes were aligned with the *protomo* package and the resulting maps were merged with IMOD to produce the raw dual-axis tomogram with the dimensions 1600×1344×70 pixels. The pixel size is 0.69 nm. The the raw tomogram is the starting point for this tutorial. It can be downloaded from the EM data bank (http://www.emdatabank.org), entry EMD-1561.

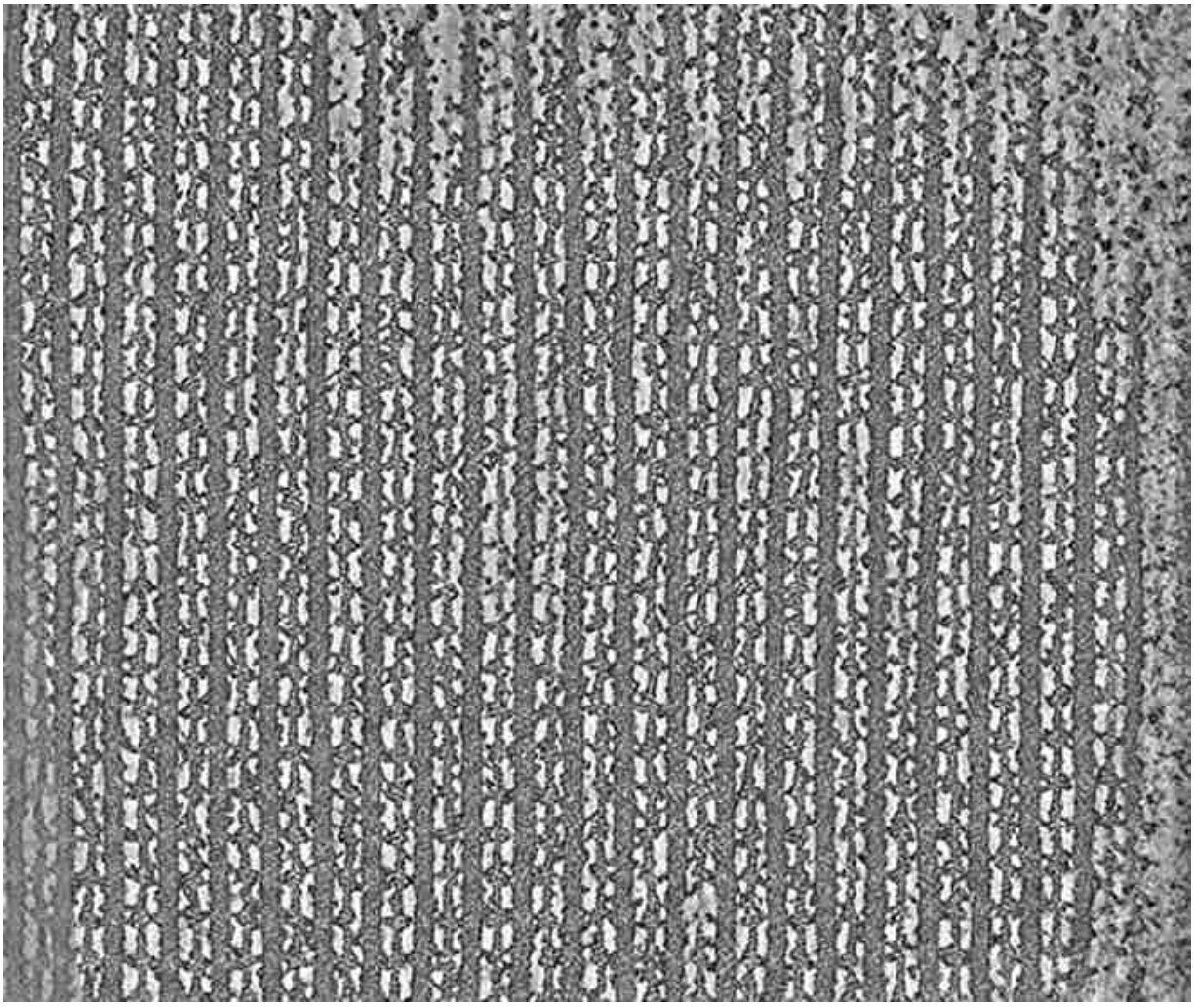

Figure 1: *Section through insect flight muscle map*

1.2 Initial setup

Make sure that the latest version of the *protomo* software is installed. Unpack the tutorial data by typing the following command at the shell prompt:

```
tar -xjf protomo-subvolume-tutorial-3.x.tar.bz2
```

where `3.x` is the release number. This will create a new subdirectory in the current working directory with the name "`protomo-subvolume-tutorial-3.x`". We will refer to it as the top directory for our data analysis. Within the top directory you will find all the required files and data, except for the raw tomogram, which needs to be downloaded from the EM data bank (http://www.emdatabank.org, entry EMD-1561), and copied into the subdirectory "`maps`".

2 Input data

2.1 General considerations

The conventions described in the *protomo* user's guide apply for units, geometry, and transformations. A linear transformation is associated with each subvolume that defines a translation and rotation from the coordinate system of the tomogram to a user defined coordinate system that is fixed with respect to the structural motif. In our case, a point centered on the actin filament axis is the origin of the motif coordinate system, and the filament axis is chosen as the z-axis. Three origin coordinates and a rotation matrix need to be stored for each subvolume. Note, that if a rotation is expressed with Euler angles, the z-x-z convention is used: the first rotation is about the z-axis, the second rotation about the new x-axis, and the third rotation about the new z-axis after the second rotation.

Before we can start with the alignment, we need to provide the positions of the subvolumes relative to the tomogram and their orientation in space. If the orientations are unknown, an attempt should be made to obtain an estimate. This will reduce the computational effort for the alignment. For example, in the alignment of Env spikes on SIV or HIV virions Winkler et al. (2009), the direction of the spike axes could be approximated by the surface normals to an ellipsoidal body, the virion, so that only a rotation about the spike axis was unknown in the beginning. The subsequent refinement of the orientation could thus be restricted to a small angular sector.

2.2 Subvolume positions and orientations

In our tutorial, the orientation determination is very simple, because the specimen is paracrystalline. Since the motifs are arranged in a regular array, all subvolume orientations are roughly the same, but they may vary to a certain extent because of imperfect lattice order. For the subvolumes we chose the actin filament axis as the z-axis, which is roughly parallel to the y-axis of the tomogram. The reason for that choice is that the rotational orientation search can only be carried out around the z-axis with the current software.

The positions of the motifs are already picked, and the files with the coordinates are located in the subdirectory "`pos`". The positions have been divided into subsets, one file per subset, with each subset referring to the subvolumes of a single filament in the tomogram. On each line in the

text files the x-, y-, and z-coordinates of the centers of the subvolumes are given. These center coordinates are specified relative to the tomogram coordinate system. Depending on the data source, the origin of this coordinate system may be at the center of the tomogram, at the bottom left voxel of the tomogram, or at some other point. In our tomogram the origin is at the bottom left and the file is a CCP4 map.

Positions can be determined automatically or manually. Automatic particle picking is beyond the scope of this tutorial and will not be discussed. For the manual picking and/or displaying positions we can use the program `i3display`" (see *protomo* user's guide for how to use it). To display all positions, we execute the following commands at the shell prompt, assuming that our working directory is the "`pos`" subdirectory:

```
cat col?.pos >all.pos
i3display -r -2200 2500 -pos all.pos ../maps/emd_1561.map
```

The Unix "`cat`" command concatenates all text files with positions into one single file.

Once we have the subvolume positions, we also need the orientations of the motifs in the form of rotation matrices. These matrices define the rotation of the tomogram coordinate axes to the coordinate axes relative to the structural motif within the subvolume. As already mentioned, we define the thin filament axis as the z-axis, so that the tomogram y-axis becomes the subvolume z-axis, while the x-axes remain the same. The matrix for this rotation is

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

2.3 Preparing a data set

First, we extract the subvolumes from the tomograms using the motif positions with the following command after we change the working directory to "`prepare`":

```
tomoprepare -log extract.prep
```

The file "`extract.prep`" contains a list of commands for the program "`tomoprepare`":

```
mapdir ../maps
map emd_1561.map
window 90 80 80
area 0.8
rotation  1 0 0  0 0 -1  0 1 0
extract ../pos/col0.pos to stacks/col0.i3i
extract ../pos/col1.pos to stacks/col1.i3i
extract ../pos/col2.pos to stacks/col2.i3i
...
...
extract ../pos/coll.pos to stacks/coll.i3i
extract ../pos/colm.pos to stacks/colm.i3i
```

"`mapdir`" specifies the location where the maps are stored, and "`map`" provides the file name. We also specify the size of the extracted window, and additionally, we indicate with the parameter "`area`" that windows at the edges must overlap the map with at least 80% of their volume in order to be selected and extracted. For all extracted subvolumes the rotation matrix specified with the parameter "`rotation`" is recorded. The extraction does not interpolate the data, which means that the extracted subvolumes have the same orientation as in the tomogram. The rotation is applied during processing, so that only a single interpolation of the data has to be carried out. The positions for the extraction are read from the files "`../pos/col?.pos`" and the subvolumes are written into the subdirectory "`stacks`" with the file names "`col?.i3i`" (the question mark indicates a digit from `0 - 9` and a letter from `a - m`).

To create the initial data set, we use the file "`dataset.prep`" as input to "`tomoprepare`":

```
tomoprepare -log dataset.prep
```

In the input file, "`search`" specifies the subdirectory containing the individual stacks which are combined into a dataset with the command "`attach`". At the end, the whole data set is saved into a new file with the name "`dataset.i3i`":

```
search stacks
attach col0.i3i
attach col1.i3i
attach col2.i3i
...
...
attach coll.i3i
attach colm.i3i
save dataset.i3i
```

The number of subvolumes of the original dataset has been reduced to 115 for faster turnaround in the tutorial. The complete set of positions for our sample data set (515) can be found in the subdirectory "`all-pos`". We do not use missing wedge compensation, because the map is a dual-axis tomogram. The missing pyramid in the tomogram is relatively small, compared to the missing wedge of a single axis tilt series, and we do not expect that the missing data has a significant effect in the alignment, classification, or the averaging. Furthermore, our subvolumes are all oriented in more or less the same way relative to the tomogram, so any effect the missing information has would affect all subvolumes in the same way.

3 The alignment and classification procedure

The general procedure is shown schematically in Figure 2. The initial references are usually derived from a global average. Cycles following the initial cycle typically use class averages. If the reference file is an image stack, a multireference alignment (MRA) is carried out in which each subvolume is cross-correlated with all references and the alignment with the highest cross-correlation coefficient is selected. The alignment is followed by multivariate statistical analysis (MSA), classification, and the generation of new references. A more detailed discussion of subvolume processing can be found in Taylor et al. (2006).

Computationally, the procedure is implemented in a slightly different way than depicted in the figure. All parameters and reference images are specified at the beginning of the procedure,

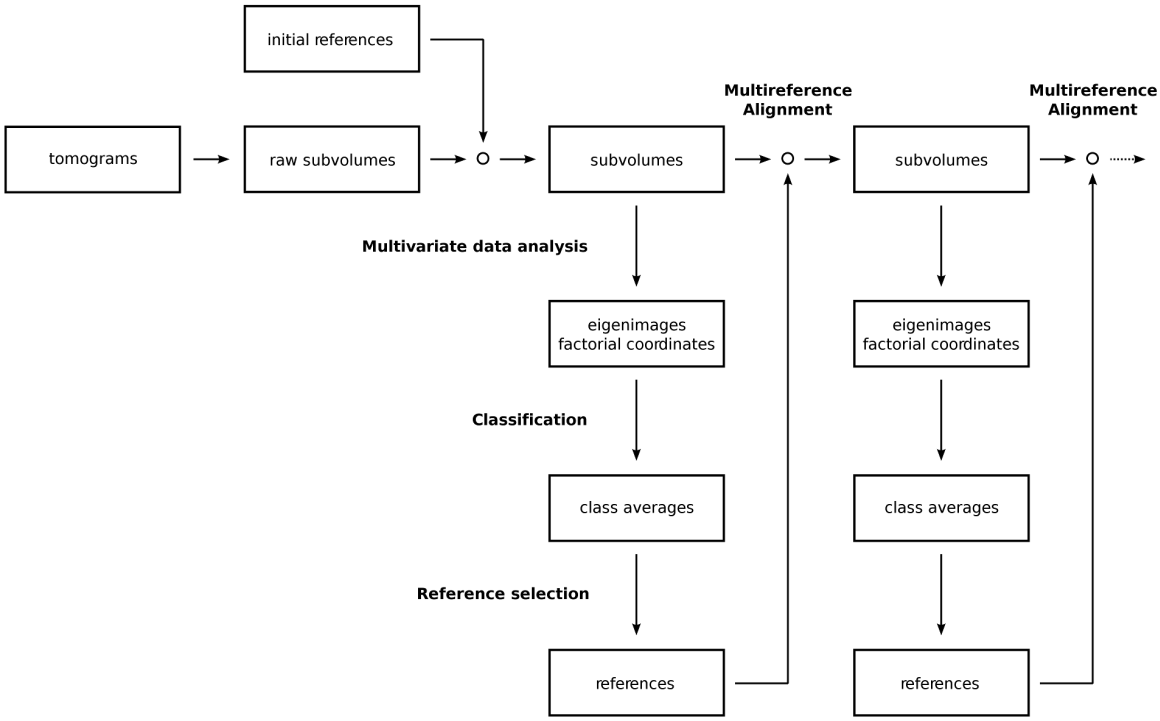

Figure 2: *Subvolume processing procedure*

before the time-consuming calculations are executed. Each cycle is split into three parts: (1) the creation of alignment references and masks for the multivariate statistical analysis (MSA-masks), (2) the alignment and classification of the subvolumes, and (3) the alignment of class averages to each other. After completion of these three steps, a new cycle is started. The results of each cycle are written to a subdirectory created within the top level directory.

3.1 Processing parameters

The processing parameters are specified in a shell script as exported environment variables. Therefore, the format must conform to the shell syntax. In particular, no blanks should be inserted between the variable name, the equal sign, and the assigned value. The default name of the file is "`param-template.sh`", and the file is not directly used during processing. A copy of it is made at the beginning of each cycle, and put in the subdirectory of the cycle. Only changes made to the copy will have any effect in the processing. The file provided with the tutorial lists all valid parameters with a short description. The assigned parameter values ensure that the tutorial will work "out of the box" without errors. In many cases, however, finding the optimal parameters is a matter of trial and error, which can be time-consuming.

3.2 Initial cycle

First we initialize the processing with the shell script "`subvolinitial.sh`". The working directory should now be "`process`". The initial parameters are stored in file "`template-initial.sh`". We copy it to the default template file and run the initialization script:

```
cp template-initial.sh param-template.sh
subvolinitial.sh ../prepare/dataset.i3i
```

This creates a subdirectory called "`cycle-000`" in which all the data and results of the first cycle are stored, e. g. a copy of the parameter file, the raw image stack, a global average, a MSA-mask, etc. At this point we can create the MSA-mask for visualization, before we run the time-consuming alignment:

```
subvolmsamask.sh 0
```

and inspect the mask and a superposition on the global average.

Next, we create the reference for the alignment:

```
subvolreference.sh 0
```

We left the parameter `REFIMG` blank, so the reference was generated from the global average. If we want to use a different image than the global average as a reference, we could specify its file name with the `REFIMG` parameter. This file can be a single image, or a stack of images, in which case a multireference alignment will be carried out. The script produces a file called "`*-reference.i3i`", which contains the Fourier transform of the image, a filtered version with the name "`*-refimg.i3i`", and montages "`*-refmont.img`" for easier visualization. The asterisk indicates the prefix of the file names, here it is "`hst-000`".

The images in the montages are ordered as shown in the figure below. The first image is placed at the bottom left of the montage and the following images to the right of it. The next row of images is placed above the first row.

20	21	22	...	29
10	11	12	...	19
0	1	2	...	9

Earlier, we have chosen the z-axis parallel to the filament axes, so when we display the output files with the command

```
i3display cycle-000/hst-000-refmont.img
```

we look down the z-axis, the y-axis is vertical in the display, and the x-axis horizontal. The image shows the cross-section of the thin filament in the middle, and two thick filaments on either side. Additional files with the file name component "`-y`" contain the same images rotated about the x-axis, so that the filament axes are now vertical as in the original tomograms. Compare these images to the global average "`cycle-000/hst-000-raw.avg`" and "`cycle-000/hst-000-raw-y.avg`" to check the applied masks and the effect of low- and high-pass filters.

In the second step, we align the subvolumes, do an eigenvalue/eigenvector decomposition, a classification, and then produce class averages:

```
subvolalign.sh 0
subvolsvd.sh 0
subvolhac.sh 0
subvolclassaverage.sh 0
```

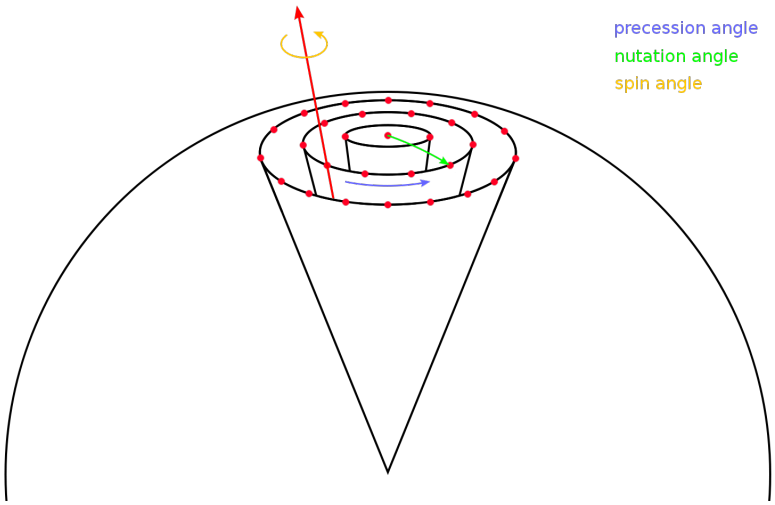

Figure 3: *Grid search for rotational alignment*

The rotational alignment is carried out with a grid search (Figure 3).

The step size for the precession angle is assumed to be the same as the step size for the nutation angle. The step size is defined as the angle between adjacent grid points, measured from the center of the unit sphere. Thus, the number of grid points on concentric cones increases with increasing nutation angle. Only four parameters need to be specified with this definition, the step size and a limit for precession/nutation, and the step size and limit for the spin angle. For details, refer to the parameters `MRALIMIT` and `MRASTEPS` in the parameter file.

In our tutorial, we only do a rotational search for the spin angle at this point. If both precession and nutation angles are zero, the spin angle rotates about the z-axis. What we want to do is to test the filament orientation for a 180° rotation about the filament axis, which we chose above to be parallel to the z-axis. The search limits are zero for the nutation angle and 180° for the spin angle: `MRALIMIT=''0 180''`, and the step size is zero and 180° for these angles: `MRASTEPS=''0 180''`. This results in only two rotations to be tested: 0° and 180°.

The subvolumes are resampled using the transformations stored in the stack file, then masked with the relevant parameters (names start with "`MRA`" for the multireference alignment). An optional molecular mask can be specified, which is an image with values from 0 to 1 with which all subvolumes are multiplied, in addition to the geometric masks. The cross-correlation function is filtered (parameters `LOWPASS` and `HIGHPASS`), and a peak search is performed (parameters `MRAPKR` and `MRACMR`). The type of correlation functions is selected with the parameter `MRACC`, "`xcf`" is the conventional cross-correlation, "`mcf`" the mutual correlation, and "`pcf`" the phases-only correlation.

For the eigenvalue/eigenvector decomposition, a similar set of parameters with names starting with "`MSA`" is used. The subvolumes are extracted with the transformations obtained by the alignment, then masked and filtered. The additional MSA-mask selects the pixels within each image that will be used in the analysis. It is essentially a binary mask, but given as a mask file with values from 0 to 1 (parameter `MSAMASK`). Pixels below a threshold value (parameter `MSAMASKTHR`) are rejected. The default for the threshold is 0.5. The output of this step are singular values (file name suffix "`.sv`"), "singular images" or eigenimages (file name suffix "`.rsv`"),

and factorial coordinates (file name suffix "`.coo`"). Optionally, a montage of the eigenimages is produced (file name suffix "`-mont.rsv`").

Hierarchical ascendant classification takes the factorial coordinates as input and produces a new data set that includes the class memberships for the specified number of classes (parameters `CLSMIN`, `CLSMAX`, `CLSINC`). The program precomputes and stores classifications with the number of classes in the range specified with these parameters, from `CLSMIN` to `CLSMAX`, with an increment of `CLSINC`. Actual class averages are generated based on this stored information. The specific classifications to generate class averages for is given by the parameter `CLASSES`, which is a space separated list of integer numbers. The numbers must be covered by the sequence defined with `CLSMIN`, `CLSMAX`, and `CLSINC`. For each classification, a stack of class averages, optional montages, and additional information is stored in a separate subdirectory. The class averages are not filtered. The montages, however, are low-pass filtered (parameter `CLSMONT`). They can be viewed with the commands:

```
i3display cycle-000/hst-000-class-004-all-y-mont.img
i3display cycle-000/hst-000-class-008-all-y-mont.img
```

Each class average stack contains an additional "junk class", which is the average of all images excluded by the classification algorithm (see parameters `CLSHVO` and `CLSHVM`).

After inspection of all classification data, we select a particular classification (parameter `SELNR`) and align the class averages with respect to each other:

```
subvolclassalign.sh 0
```

A subset of class averages can be specified with parameter `SELAVG`. Normally, we would at least exclude the junk class here. The alignment is carried out with all possible pairs of the selected class averages. From these pairs, the optimal alignments with the highest cross-correlation coefficients are determined and stored in an output file ("`hst-000-sel.i3i`") in preparation of generating a new reference for the next cycle. Montages of the results can be viewed with the commands

```
i3display cycle-000/hst-000-selali-mont.img
i3display cycle-000/hst-000-selali-mont-y.img
```

3.3 Following cycles

For the second cycle, we do a multireference alignment and select three of the four aligned class averages by changing parameters in the "`param-template.sh`" file:

```
export REFIMG=''classaverages''
export REFSEL="1-3"
```

These changes are already recorded in the file "`template-cycle-1.sh`'', so we only need to copy the file and start a new cycle by calling the script "`subvolnext.sh`":

```
cp template-cycle-1.sh param-template.sh
subvolnext.sh 0 1
```

The first parameter is the previous cycle number from which the alignment information will be copied, and the second one is the new cycle number (the initial cycle is number 0, the second cycle number is 1, etc.).

The following steps are the same as in the initial cycle: alignment, MSA and classification, alignment of class averages:

```
subvolreference.sh 1
subvolalign.sh 1
subvolsvd.sh 1
subvolhac.sh 1
subvolclassaverage.sh 1
subvolclassalign.sh 1
```

In the third cycle, we run the alignment with a different orientation search. We want to compensate for lattice and filament bending and search in a narrow angular sector, a 1° cone (limit 1°, step size 1°) and a limited range of spin angles (limits ±3°, step size 1.5°). For the new reference, we use all class averages and edit the parameter file accordingly:

```
export REFIMG=''classaverages''
export REFSEL="0-3"
export MRALIMIT="1 3"
export MRASTEPS=''1 1.5''
export SELLIMIT="1 3"
export SELSTEPS=''1 1.5''
```

Again, these changes are reflected in the file "`template-cycle-2.sh`". The following commands run the third cycle:

```
cp template-cycle-2.sh param-template.sh
subvolnext.sh 1 2
subvolreference.sh 2
subvolalign.sh 2
subvolsvd.sh 2
subvolhac.sh 2
subvolclassaverage.sh 2
subvolclassalign.sh 2
```

This concludes the third cycle. Criteria to repeat or terminate the alignment process can be based on visual inspection of class averages, estimated resolution (Fourier shell correlation), etc.

4 References

Taylor, K. A., Liu, J. and Winkler, H. (2006). Localization and classification of repetitive structures in electron tomograms of paracrystalline assemblies. In Electron Tomography. Methods for three-dimensional visualization of structures in the cell, (Frank, J., ed.), pp. 417–439. Springer New York.

Winkler, H. (2007). 3D reconstruction and processing of volumetric data in cryo-electron tomography. J. Struct. Biol. *157*, 126–137.

Winkler, H. and Taylor, K. A. (1999). Multivariate statistical analysis of three-dimensional cross-bridge motifs in insect flight muscle. Ultramicroscopy *77*, 141–152.

Winkler, H., Zhu, P., Liu, J., Ye, F., Roux, K. H. and Taylor, K. A. (2009). Tomographic subvolume alignment and subvolume classification applied to myosin V and SIV envelope spikes. J. Struct. Biol. *165*, 64–77.

Wu, S., Liu, J., Reedy, M. C., Winkler, H., Reedy, M. K. and Taylor, K. A. (2009). Methods for identifying and averaging variable molecular conformations in tomograms of actively contracting insect flight muscle. J. Struct. Biol. *168*, 485–502.