

IHRSR++ tutorial (1.5)

Robert Sinkovits and Kristin Parent
Department of Chemistry and Biochemistry
University of California, San Diego

Introduction

This document describes how to perform a helical reconstruction, start to finish, using the IHRSR++ software package (Parent et al, Phys Biol 2010). IHRSR++ builds on Ed Egelman's original IHRSR program (Egelman, Ultramicroscopy 2006), but makes the image reconstruction software easier to use and adds scripts to handle much of the data pre- and post-processing.

Please note that IHRSR++ is not actively supported at this time and the usual caveats regarding freely distributed software apply. Also keep in mind that the steps described below worked for us, but may not be strictly applicable in all cases. Every helical reconstruction project is unique and you may find it necessary to modify our protocol. Finally, one of the most challenging aspects of solving a helical structure involves the indexing of the diffraction patterns to determine the rotational symmetry and axial rise and rotation per asymmetric unit. We make no attempt here to describe this process and instead refer the reader to the many excellent published descriptions (e.g. Glaeser et al in Electron Crystallography of Biological Macromolecules) of this topic.

Software

To run IHRSR++ and work through the test problem, you will need to download the following software

IHRSR++	http://cryoem.ucsd.edu/programs.shtm
SPIDER	http://www.wadsworth.org/spider_doc/spider/docs/spider.html
EMAN	http://blake.bcm.tmc.edu/eman/

The image reconstruction core is built on SPIDER. While not strictly required, EMAN does provide very useful programs for boxing helices (helixboxer), viewing images (v2), and performing file format conversions (proc2d).

Optional software that you may find useful includes ctffind3 for defocus estimation and Chimera for visualizing 3D maps.

ctffind3	http://emlab.rose2.brandeis.edu/ctf
Chimera	http://www.cgl.ucsf.edu/chimera/

Notes on tutorial

Throughout the tutorial, **input specific to the accompanying test problem is shown in red**. For simplicity three micrograph in TIFF format were provided, but a real reconstruction project typically involves many micrographs. Although you should be able to run through the test problem using just the micrograph files, we have also provided the output files that are generated along the way.

The Perl and SPIDER scripts provided with IHRSR++ are shown in **bold**. Full file names are required for the Perl scripts, whereas file names without extensions are used at SPIDER prompts (in SPIDER, file extensions are specified on the command line, e.g. spider spd/dat)

Overview of helical reconstruction process

The full image reconstruction process is outlined below. For low-resolution reconstructions (i.e. at resolutions below the first phase reversal in the CTF curve), the CTF correction is not necessary and steps 1e and 4 can be omitted. Step 5 is not required but is strongly encouraged since it provides a quick and easy sanity check on the final structure.

- 1) Process micrographs
 - a. Box particles
 - b. Calculate power spectra
 - c. Rename files and set byte ordering
 - d. View power spectra
 - e. Apply CTF correction (optional)
- 2) Create image stack
 - a. Cut tubes into segments
 - b. Center segments
 - c. Sort segments by diameter
 - d. Invert, pad, low-pass filter, and decimate
- 3) Run helical reconstruction
- 4) Divide result by $\langle \text{CTF}^2 \rangle$ (optional)
- 5) Generating simulated power spectrum (optional)

Step 1 – Process the micrographs

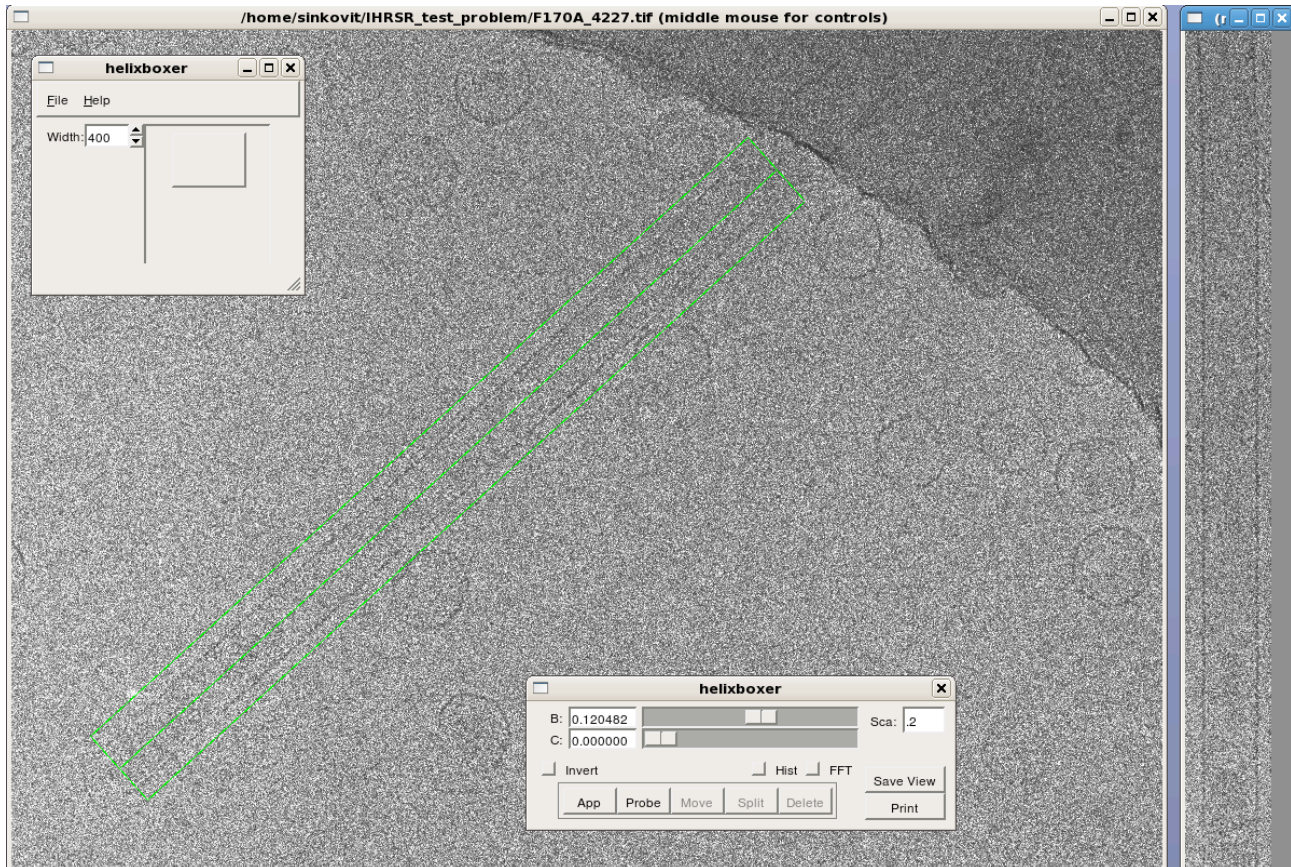
Box particles

After micrographs are available, in either TIFF or MRC format, the first step is to box out the helical particles. Three micrographs are included in the test problem, but the example below is worked out for a single micrograph.

1. Type heliboxer to start program
2. File > Open Micrograph, then select image (**F170A_4227.tif**)
3. Middle-click on image to open a new window that allows you to scale micrograph and adjust contrast. A scale factor of **0.2** was used for this screen shot and particle is located in upper right of micrograph. Use right mouse button to drag image. Moving the slider next to B may make it easier to see the helix.
4. Set box width (pixels). In this case, **400** pixels turned out to be optimal.
5. Left-click and drag mouse to box particle. To delete a box, Control key + left-click. For some mouse configurations, Shift + left-click may perform this operation.
6. When satisfied with boxing, save box coordinates and boxed images. Be sure to use 'spi' extension so that images are saved in spider format. Heliboxer will automatically add box number to file name (e.g. name.0.spi, name.1.spi)
 - a. File > Save DB, then enter box file name (**4227.box**)
 - b. File > Save particles, then enter file name (**4227.spi**)

7. When opening subsequent micrographs in a helixboxer session, the existing boxes will remain.
Be sure to delete these before proceeding
8. File > Exit to end session

The figure below illustrates the use of helixboxer and is preceded by step-by-step instructions



Calculate power spectra

The calculation of power spectra is most easily performed using the **helix_automation** script

1. Type **helix_automation** to start script. At prompts enter:
 - a. List of boxed image files to be processed: ***.*.spi**
 - b. Dimension for power spectrum calculation: **8192**
 - c. Size of windowed power spectrum to be written out: **2048**
 - d. Threshold intensity in power spectrum – default normally used: **0**

The power spectra of the boxed images are not necessary for the reconstruction, but rather are used to evaluate or classify the data. Note that dimension of power spectrum must be at least as large as length of longest particle. Helices will often be discarded because they fail to diffract. In other cases, the sample may contain a collection of structures with different helical symmetries and the power spectra provide a convenient way to sort the boxed images into classes (Parent et al, Phys Biol 2010). The average power spectra from particles that produce the same diffraction pattern can be generated as follows:

1. Execute **avg_pw2**. At prompts enter:
 - a. Enter power spectra to be averaged: **pow***
 - b. Enter name of avg power spectrum: **average**
 - c. Enter box dimension: **2048**

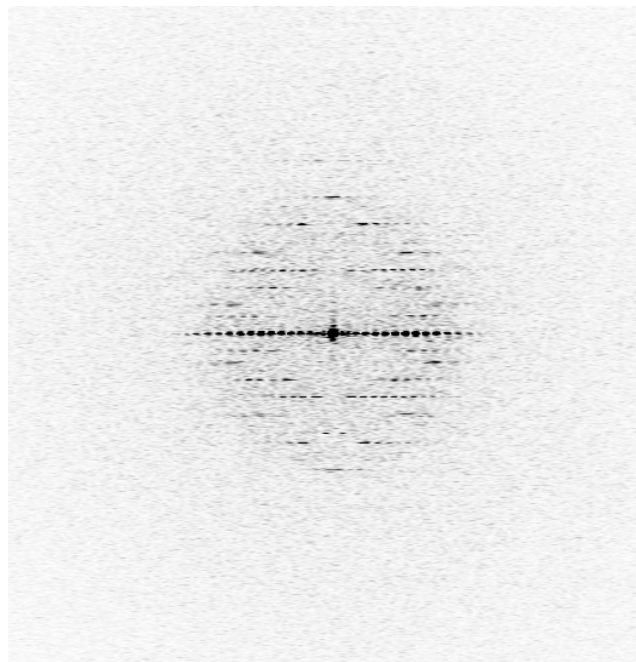
Rename files and set byte ordering

helix_automation performs two important tasks in addition to calculating power spectra.

1. Convert boxed image file to correct byte ordering for use by SPIDER
2. Rename the byte-swapped file to add the 'new' prefix, replace 'spi' extension with 'dat', and convert all dots in the file name except for the last one to underscores (SPIDER treats everything after the first dot as the file name extension)

View power spectra

The power spectra can be viewed using EMAN's v2 viewer (e.g. v2 **average.dat**). Clicking the center mouse button brings up a new window, similar to that in helixboxer, which allows one to change the contrast and scaling. In this instance setting the scaling to **0.5** and moving the C slider to **0.9** made the power spectrum easy to visualize. At this scaling, the image will have to be dragged from the upper right corner of the viewing area while holding down the right mouse button.



Apply CTF correction (optional)

This section can be skipped and revisited later after working through the test problem. It is provided here since this is the point at which the CTF correction would be applied. Note that the test problem does not contain CTF-corrected files.

To reach the highest possible resolution, it is necessary to account for the effect of the contrast transfer function (CTF). Although this can be done in several ways, we find that multiplying the micrographs

by the CTF, boxing the helices from the corrected micrographs, and dividing the final reconstruction by sum over the CTF² works best.

In addition to the microscope accelerating voltage, the key parameter required for applying the CTF correction is the defocus level at which the image was collected. Other parameters, such as the spherical aberration (Cs) and amplitude contrast (Famp) are hardcoded for an FEI Polara microscope. The defocus can be estimated in several ways, but we find that the ctffind3 program (see introduction) is particularly easy to use. Once the defocus level is known, the CTF correction is applied and newly boxed images are obtained as follows

1. Use the proc2d command to convert the micrographs into SPIDER format files with the correct byte ordering: `proc2d F170A_4227.tif F170A_4227.dat spiderswap`
2. Launch the CTF correction script (`spider spd/dat @ctf_corr`). At prompts enter:
 - a. Input image name: `F170A_4227`
 - b. Output image name: `F170A_4227_corr`
 - c. Pixel size: `1.62`
 - d. Defocus: `1.41`
 - e. Voltage: `200`
 - f. Correction: `2`
 - g. Sign: `-1`
3. Box particles from corrected micrographs
 - a. Launch helixboxer
 - b. File > Open Micrograph, then select image (`F170A_4227_corr.tif`)
 - c. File > Read DB, then select (`4227.box`)
 - d. File > Save Particles, then enter filename (`4227_corr.spi`)

If both phase and amplitude corrections were applied, the particles and other features will not be visible. For this reason, the boxing must be done as a two-step process where the box coordinates are obtained for the uncorrected micrographs and then applied to the corrected micrographs. This procedure is analogous to that used for focal pairs, where boxing is done on the far-from-focus micrographs and then applied to the near-to-focus micrographs. If just phase corrections were applied, particles will still be visible but the two-step process aids in bookkeeping. In our experience using phase corrections only works best.

At this point we suggest running the `helix_automation` script again on these boxed image files to ensure that byte ordering and naming conventions are correct.

Step 2 – Create image stack

Before proceeding, the selected helices must be cut into overlapping segments and combined into a single stack file. The quality of the reconstruction is improved if the segments are centered in their respective boxes and those segments with diameters outside a fairly narrow range of values are excluded. Before launching the reconstruction, the segment images must also be inverted, padded out to square dimensions, and optionally low-pass filtered and decimated (binned).

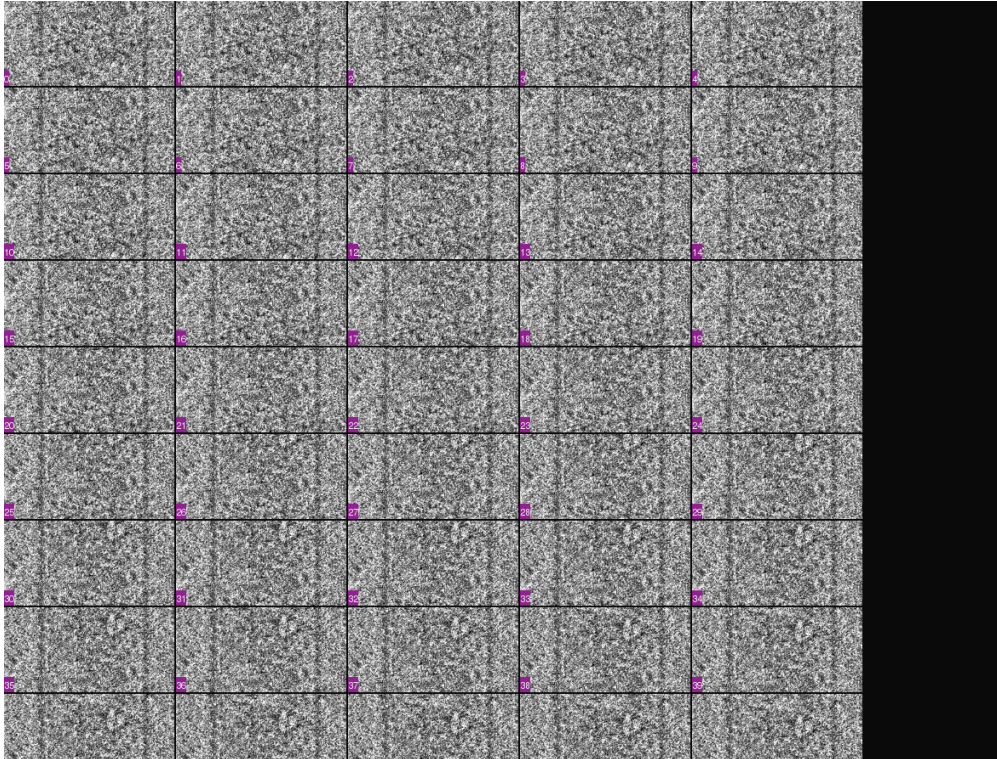
Cut tubes into segments

1. Launch `helix_cutstk`. At the prompts enter:
 - a. Files to be processed: `4???_0.dat*`

- b. Size of cut boxes: 400
- c. Shift distance: 8
- d. Stacked image file to be created: stack

The box size is determined empirically and changing can have a large impact on resolution. Once you find a value that works, stick with it. As a rule of thumb, start with a value greater than the axial rise.

Using the boxed image file provided with the test problem will result in an image stack containing 1920 segments. The number of segments will likely be different if you used your own boxed image files. A screen shot displaying a portion of this stack file in v2 is shown below



Center segments

1. Launch centering script (spider spd/dat @center_tubes). At the prompts enter:
 - a. Input image stack: stack
 - b. Output image stack: cent
 - c. Pixel size (in angstroms): 1.62
 - d. Low pass filter resolution (in angstroms): 20
 - e. Number images (0 to use all): 0

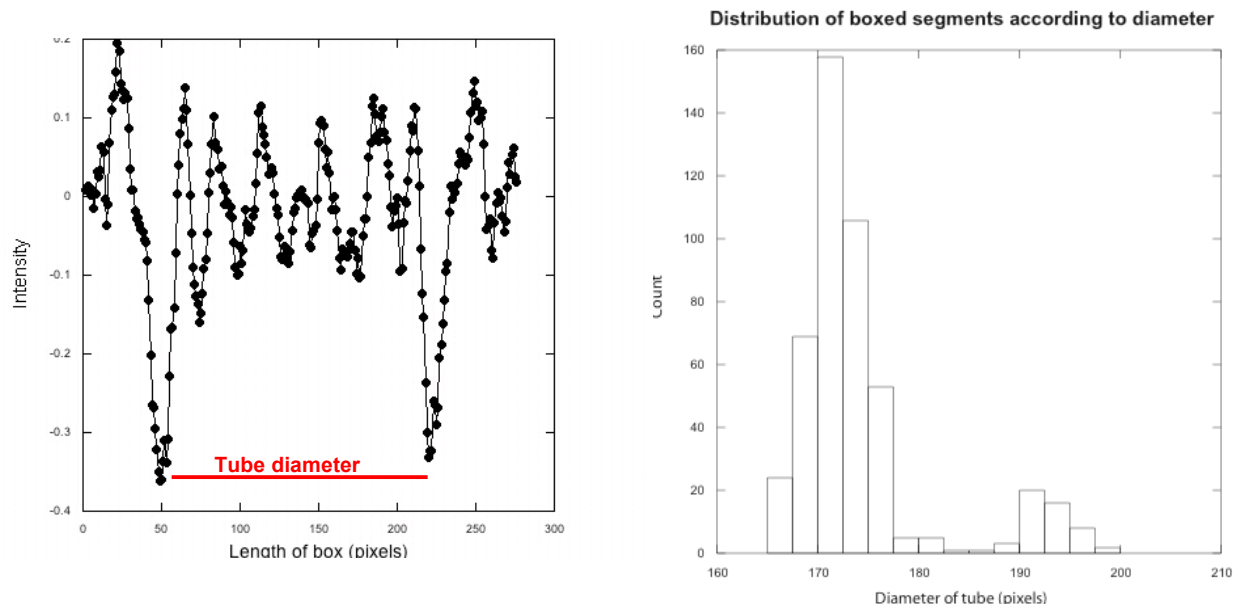
Sort segments by diameter

It is very common both for particles that produce the same diffraction pattern to have different diameters and for there to be significant variations in the diameter along the length of a particle. Therefore, it is essential to determine the diameters of all segments and then restrict the reconstruction to use just those segments that fall within a specified range of sizes. Unfortunately, there are no hard and fast rules for choosing this range. An overly narrow distribution can lead to a lower-resolution

reconstruction since too much data will be discarded, while too broad of a distribution may degrade the final result because of segment heterogeneity. Below we describe how to determine the segment diameters and generate a new stack file containing segments of the desired size.

1. Launch diameter determination script: spider spd/dat @get_tube_diameters. At prompts enter
 - a. Input image stack: **cent**
 - b. Pixel size: **1.62**
 - c. Low pass filter resolution: **20**
 - d. Number of images (0 to use all): **0**
2. Plot a histogram of the data in the doc_peaks.dat file. Use last column only since first two columns contain data used by SPIDER to keep track of record number and number of fields in each record. Decide on cutoffs, keeping in mind that the distribution of diameters may be a little wider than expected owing to noise in the 1d collapse of the data.
3. Launch sorting script: spider spd/dat @sort_by_diameter. At prompts enter
 - a. Input image stack: **cent**
 - b. Output stack: **new_246_253**
 - c. Minimum tube diameter: **246**
 - d. Maximum tube diameter: **253**
 - e. Number of images (0 to use all): **0**

For this test case, the particle has a sufficiently consistent diameter that all segments will be used, but the figures below (taken from Parent et al, Phys Biol 2010) illustrate the diameter determination for a single segment (left panel) and the range of segment sizes (right panel). After working through this tutorial, you may want to try running the reconstruction again using a subset of the segments.



4. Launch invert, pad, low-pass, decimate script: spider spd/dat @ipdlp. At prompts enter:
 - a. Input image stack: **cent**
 - b. Output image stack: **fin**
 - c. Pixel size: **1.62**

- d. Low pass filter resolution: 20
- e. Padded box size: 440
- f. Binning/decimation factor: 2
- g. Number of images (0 to use all): 0

To reach the highest resolutions, no low-pass filtering or decimation of data would be applied. Low-pass filtering and decimation of data that had not been CTF corrected allows one to do a fast reconstruction to see if we are on the right track. This test problem took approximately 20 minutes to run on compute node with two Intel quad-core Nehalem processors (2.66 GHz).

Step 3 – Run helical reconstruction

Once the image stack is available, copy the **bob_helix.spd** script to the directory containing the data. Be sure to rename this file so that you don't accidentally access the script containing the default parameters. In this case we changed its name to **IHRSR_input.spd**.

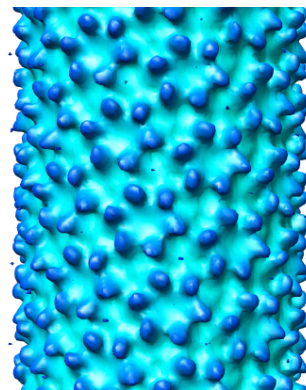
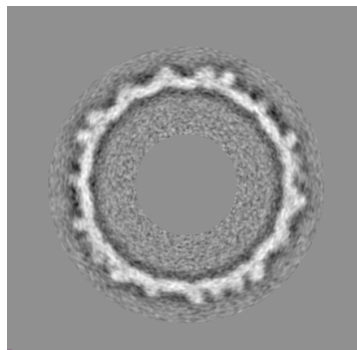
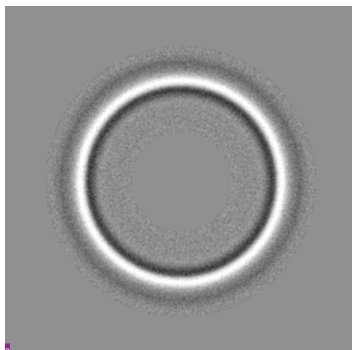
The file **IHRSR_input.spd** present in the test data has already been configured for this test problem, but you will need to edit the following records

- Line 4: The number of Open MP threads must be set to a value appropriate for your system. The value must be no larger than the number of available shared memory processors or cores
- Line 10: Record must contain the full path to the location of the image stack

The values for the symmetry parameters ([Cn], [rise], [deltaphi]) were derived from the diffraction pattern (derivation not shown). Other parameters are either set to reasonable defaults or assigned values reasonable for this structure. To launch job, simply type

`spider spd/dat @IHRSR_input &`

The results are shown below. Left and center panels are axial views from iterations 1 and 10 generated using v2; right panel is surface-shaded rendering of final result generated using Chimera.



Step 4 – Divide result by <CTF> (optional)

After obtaining a reconstruction from multiple CTF-corrected images, the final result should be divided by the averaged CTF².

1. Create a file called **defval.dat** that lists the defocus levels and number of segments
2. Launch the script that calculates corrections: `spider spd/dat @ctf2inv`. At the prompts enter:
 - a. Input file containing defocus info: **defval**
 - b. Output file containing avg(ctf2): **ctf2_func**
 - c. Output file containing 1/avg(ctf2): **ctf2inv_func**
 - d. Pixel size: **1.62**
 - e. Voltage: **200**
 - f. Image dimension: **440**
3. Launch script that applies correction: `spider spd/dat @apply_function`. At the prompts enter:
 - a. Input image name: **4227_010**
 - b. Output image name: **4227_010_corr**
 - c. File containing tabulated function: **ctf2inv_func**

An example defval.dat file is shown below

1	2	297	3.25	#3563.0
2	2	534	2.56	#3734.0
3	2	222	2.76	#3755.0
4	2	45	3.53	#3728.2
5	2	101	3.53	#3728.3

Column 1: Record number (required by SPIDER)
Column 2: Number of fields in record (required by SPIDER)
Column 3: Number of segments in tube
Column 4: Defocus level (μm)
Column 5: Optional comment used to keep track of particles

Step 5 – Generate simulated power spectrum (optional)

While it's possible for a solved structure to be incorrect and still yield a power spectrum that resembles that obtained from the original data, it is highly unlikely that an incorrect structure will produce a power spectrum that is correct. In other words, being able to generate a correct power spectrum can be thought of as a necessary, but not sufficient, condition for the solved structure to be correct.

The program **himpose**, which is normally called from within IHRSR to generate a symmetrized volume, can also be used as a standalone program to write out an arbitrarily long tube or its projection. The power spectrum of this projection can then be calculated and compared to the original spectrum. Using the longest tube possible, subject to memory and computational restrictions, gives the best results.

1. Generate projection of long tube from un-symmetrized volume. Execute **himpose** without command line arguments to see explanation of command line arguments.

himpose **ftzdisk010.dat** **syndoc.dat** **long.dat** **3.24** **100.0** **275.0** **4096** **1**

ftzdsk010.dat	un-symmetrized volume
syndoc.dat	file listing helical symmetry parameters (generated by IHRSR)
long.dat	name of output file
3.24	pixel size (note that this is for binned data)
100.0	inner tube radius
275.0	outer tube radius
4096	length of reconstruction
1	flag indicating that 2D projection rather than 3D structure be generated

2. Calculate power spectrum: spider spd/dat @mappw2. At prompts enter
 - a. Input map or projection: **long**
 - b. Padded transform size: **4096**
 - c. Threshold value: **0**
3. View power spectrum (e.g. **v2 pow_long.dat**)

The power spectrum obtained from this reconstruction is show below

