

IMAGIC-5

Hands-On Tutorial



Contents

1	Test dataset: House Centipede (<i>Scutigera coleoptrata</i>) Hemocyanin	3
2	Hands-On: Contrast Transfer Function (CTF)	4
2.1	Playing around with EM parameters and their influence on the CTF	
2.2	Interactive CTF correction	
3	Hands-On: Pre-treatment	6
4	Reference-Free Alignment	6
5	Hands-On: Centering	7
6	Hands-On: Multivariate Statistical Analysis (MSA) Classification	8
7	Hands-On: Multi-Reference Alignment (MRA)	9
8	Hands-On: Angular Reconstitution – Initial Angular Assignment	10
9	Hands-On: Initial 3D reconstruction	11
9.1	3D Reconstruction	
9.2	3D Masking	
10	Hands-On: Angular Reconstitution – Refinement with an Anchor Set	13
11	Hands-On: Refinement	14
12	Hands-On: Fourier Shell Correlation (FSC)	14

Acknowledgement

This tutorial was prepared by Andreas Martin with material from the 3rd Brazil School for Single Particle Cryo Electron Microscopy (<http://www.singleparticles.org/school/>), held in 2008.



1 Test dataset: House Centipede (*Scutigera coleoptrata*) Hemocyanin

Hemocyanin (Hc) is the copper-containing oxygen-transport metalloprotein present in the blood (hemolymph) of arthropods and crustaceans. Hc is extracellular, running free in the blood as ~720 kDa hexamers and multiple hexamers (up to 8 in the horseshoe crab *Limulus polyphemus*). The Hc of *S. coleoptrata* is a hexamer of hexamers. Data was collected under the following conditions:

- Sample preparation method: freeze-plunging into liquid-ethane with a C3PO cryo-plunger (Gatan) under controlled temperatur (24°C) and humidity (100%)
- Data collection: 57 micrographs collected with a Philips CM12 TEM with a LaB₆ operated at 120 kV
- Magnication: 60k
- Pixel Size: 1.2 Å (7 μm stepsize)
- Pointgroup symmetry: Hexamer assembly (D₃)



2 Hands-On: Contrast Transfer Function (CTF)

An electron microscope unfortunately does not image all frequencies in the same way. This hands-on deals with IMAGIC command TRANSFER, which is an interactive program to calculate the CTF according to chosen microscope parameters. You will be supplied with a micrograph that has been coarsened by factor 4 and its respective picked particles to perform a phase flip. Due to time issues we won't be estimating the CTF of all 57 micrographs. The picked and CTF corrected particles of all micrographs will be supplied later as a single stack.

2.1 Playing around with EM parameters and their influence on the CTF

First open a command window and call command TRANSFER. Like PLOT and DISPLAY this command is an interactive command with many parameters. You can use keywords written in capitals to change important parameters. TRANSFER shows the parameter settings until you give CR/ENTER, which means NO CHANGES, i.e. go ahead and display the CTF curve.

```
Current TRANSFER settings:
=====
Desired TRANSFER function      : Phase CTF
Acceleration VOLTAGE          : 120.0          kV
Relativistic WAVE length in Angstrom: 0.0334462  Angstrom
CHROMATIC aberration          : None
SPHERICAL aberration constant  : 2.59999999  mm
FOCAL length of objective     : 1.6          mm
APERTURE of objective lens    : 50.0        micro meter
Coherent illumination SOURCE/ANGLE : 0.0
DEFOCUS value                 : 20000.0     Angstrom
GENERAL defocus value         : 21.4471512  Scherzer
OBJECT size defocus envelope   : 0ff
LENGTH of transfer function    : 512         pixel
PIXEL size in curve           : 6.0         Angstrom
-----
MODE of operation              : Calculation of CTF
Output destination for plot(s) is : IMAGIC-5 plot
Change options (VOLT,DESTIN.,MODE, etc. ...) [NO] : █
```

1. Change the pixel size to 4.8 Å:

Change options (VOLT,DESTIN.,MODE, etc. ...) : PIXEL
Pixel size measured in Angstroms :4.8

Note The pixel size of the original images is 1.2 Å. When coarsened by 4, the pixel size changes to 4.8Å.

2. Next change the GENERAL defocus value to 1 Scherzer:
Change options (VOLT, DESTIN.,MODE, etc. ...) : GENERAL
General defocus value: 1
3. Now give CR/ENTER to display the CTF curve.
4. Now you can play around with other GENERAL defocus values and notice their influence on the CTF.

NOTE: In Scherzer focus (GENERAL defocus value = 1.0 Scherzer) we have good image contrast over a large range of frequencies but, unfortunately, we have very little image contrast in the low frequencies and, as a result, we cannot recognize our particles. Using higher Scherzer foci lower frequencies are better transferred, thus, enabling us to see the particles. But, unfortunately, we now get more frequencies which are not imaged at all (the “zeroes”) and even worse, some frequencies are imaged with reversed contrast.

5. Also play around with other parameters (VOLT etc.) and examine the related CTF curves.
6. Also play with CHROMATIC aberration. Examine the related CTF curves.

NOTE: A large amount of chromatic aberration creates an envelope function which is imposed onto the CTF so that the very high frequencies are not transferred any more. Even CTF correction cannot restore these higher frequencies.



2.2 Interactive CTF correction

Although there are IMAGIC commands to automatically correct for CTF it is a good idea to use command **TRANSFER** for an interactive determination of the CTF. As before use command **TRANSFER**.

1. First specify all important EM parameters. In this case:
 - VOLTAGE: **120**
 - SPHERICAL aberration: **2.0**
 - PIXEL size: **4.8**
2. To estimate the CTF call option FIND_CTF:
 - Change options (VOLT,DESTIN.,MODE, etc. ...) [NO] : **MODE**
 - Dimension of the data set: **2D**
 - Choose mode of operation: **FIND**
 - Input file, image loc#: **6_2-c**
 - Default filter/mask parameters : **YES**
3. Play around with various defocus values until the zeroes in the CTF curve (lower curve) and the zeroes/Thon rings in the micrograph images (images above) are the same.
4. Once you are satisfied with the results, **TRANSFER** will flip the phases in the stack of picked particles:
 - Change options (VOLT,DESTIN.,MODE, etc. ...) [NO] : **MODE**
 - Choose mode of operation: **FLIP**
 - Input file, image loc#: **6_2-bxd-c**
 - Output file, image loc#: **6_2-bxd-c-ctf**
 - Remember to hit CR/ENTER to actually apply the phase flips.



3 Hands-On: Pre-treatment

1. After picking, the stacks of particles of the individual micrographs were separately CTF corrected and combined in a single file (`raw`). This file contains all the selected particles. Coarsen the picked and CTF corrected particles by factor 4 using the command `COARSE` as this will speed up the rest of the processing. The pixel size of the coarsened particles will be 4.8 Å. You can always go back to the uncoarsened data to achieve a higher resolution at a later stage. Please call the output file `raw-c`.
2. Then, we will continue with filtering and normalization. Let's determine the low-frequency and high-frequency cut-offs:
 - Low-frequency cut-off: to remove all those low frequencies that contain information larger than the size of our particle. These could be density ramps or other low-frequency information coming from the background. To calculate this value use: $2 \times \text{pixel size (Nyquist frequency)} / \text{particle size} = (2 \times 4.8 \text{ (Å/pixel)}) / 240 \text{ Å} = 0.04$.
 - High frequency cut-off: to remove high-frequencies containing mostly noise and little signal, thus increasing the overall SNR (signal to noise ratio) of the images. To calculate this value use: $2 \times \text{pixelsize} / \text{expected resolution} = (2 \times 4.8 \text{ (Å/pix)}) / 15 \text{ Å} = 0.64$.

NOTE: $2 \times \text{pixel size}$ is the Nyquist frequency which is the theoretical limit to the resolution that can be achieved.

3. Using these filter values, run the command `INCORE-PREP` to pre-treat `raw-c`. It will also normalize and zero float the dataset. Finally, the images will be masked with a large mask. Use the previously calculated values for the filtering parameters.
 - **NO** parallelisation
 - Please call the output file `flt`.
 - Use a radius of `0.99` and a drop-off of `0.2` for masking. These values will mask the images using a soft circle. The output images will be normalized inside this circular mask. This radius was chosen such that no part of the molecules are masked. The drop-off parameter determines the width of the soft edge. As usual, the radius and drop-off may be given as a fraction of the inner radius of the image or in pixels.
 - Set the new sigma to `1`
 - Invert the image densities: **YES** (particles are expected to be white)
4. What is the total sum of your dataset? Calculate it running the command `INCORE-SUM`.

4 Reference-Free Alignment

Reference-free alignment is a method by which we can obtain classes by aligning the particles to references generated from the dataset itself. Neither external references nor references generated from 3D reconstructions are used at this stage. It consists of the following steps:

1. Centering
2. Multivariate Statistical Analysis (MSA)
3. Selection of classes
4. Multi-reference alignment of the particles against the selected classes.



5 Hands-On: Centering

Once you have pre-treated the dataset, the first thing to do is centering. This means all the images will be aligned translationally (horizontally and vertically) but not rotationally, to the total sum of the dataset. In IMAGIC, the command to do this is **CENTER-IMAGE**. In this command, we specify how many rounds (iterations) of summing and aligning against the total sum we would like to perform. Usually, after 4 rounds the alignment stops changing (converges). Try the following settings:

- Input file, image loc#: **flt**
- Output file, image loc#: **ali0**
- Options for centering:: **TOTAL-SUM**
- Correlation functions available: **CCF**
- Max shift (pixels or as fraction): **0.2**
- Number of centering iterations: **4**

What is the total sum of your dataset now? Compare it to the total sum after pre-treatment.



6 Hands-On: Multivariate Statistical Analysis (MSA) Classification

After you have centered your dataset, the particles can be treated by MSA.

1. First, we need to create a mask for MSA that will specify the area of the image used to calculate the statistics. The classification will be based on this area alone. As this is the first classification and we do not know much about our object yet, we will use a large circular mask taking almost the entire image area into account for classification. In later runs you might want to use a different mask. Use the command **TEST-IMAGE** to create this mask using the **DISC** option, with a radius of **0.9**. Remember that the box size of the mask should be the same box size of the images. Call the output **msa-msk**.
2. Now, run **MSA-RUN** on the centered dataset using this mask. Bear in mind that this will take some time to run. You should display the eigenimages every few iterations to have a feeling for how they evolve throughout the run. Leave the default values to all the options but the iterations: **24**. Look for symmetry components in the eigenimages. What symmetry eigenimages can you observe? What information can you extract from them about the symmetry of the molecule being studied?

TIP: You can use the option **WATCHDOG** of the **DISPLAY** command to continuously **DISPLAY** an **IMAGIC** file as it is being updated.

3. Once **MSA-RUN** has finished we can classify the dataset. The number of classes you choose to create is related to the average number of images per class you would like. You can play with this value to see how the quality of the classes is affected. Ideally, you would have as few members per class as possible whilst still obtaining high contrast class averages. The classification is performed with the **IMAGIC** command **MSA-CLASSIFY**. Specify between 50 and 100 classes and look at the **lis** file to see how the eigenvalues are distributed.
4. Once the classification has finished, average all the particles that belong to the same class using the command **MSA-SUM**. This should improve the signal-to-noise ratio. Call the output **sum0**.
5. Play around with different numbers of classes (**MSA-CLASSIFY** / **MSA-SUM**) to find the optimal number for this dataset.
6. You need to select a number of your “best” class averages which you believe to be characteristic of your dataset. You will hopefully be able to select a number of different views, and so it is a good idea to take the “best” images from each of the different views to use as references for your first alignment. The term “best” is obviously subjective but “by eye” is the best way to select the class averages. Try and select views which contain the most internal detail, also avoid class averages which are very round. Generally if a class average contains artifacts or looks “strange” to you, then you shouldn't take it. It can cause some problems if you do not have a reference corresponding to each type of view. Almost all datasets have some views which are very common, and some views which are much rarer. If you miss references for the rare views at this stage, it can be difficult to regain them at a later stage, and so it is important to bear this in mind when choosing references. As you shouldn't be selecting too many references at this stage. Create a **PLT** file (text file with one number per line with the suffix “**plt**”) containing the locations of the best class averages. Call the **plt** file **extr.plt**.
7. Extract them using the command **EX-COPY** option **EXTRACT**. These will be used as references in the next Multi-Reference alignment. Call the output **jnk0**.



7 Hands-On: Multi-Reference Alignment (MRA)

TIP: You can type “?” at any time for help on a question.

1. It can be a good idea at this stage to use the command **PREPARE-MRAREFS**. This will centre and normalize the references for this initial alignment. Call the command with the following parameters and leave the rest as default:
 - Alignment modes available: **ALL**
 - Correlation functions available: **CCF**
 - Reference input file: **jnk0**
 - Use contours on the reference images?: **NO**
 - Give RADIUS for circular mask: **0.9**
 - MRA references output file: **mra-ref1**
2. Once this command has finished we are ready to run multi-reference alignment. In order to avoid interpolation artifacts, IMAGIC keeps the original filtered data and continually calculates the equivalent rotation necessary to reach the latest alignment. Run **M-R-A** with the following parameters (use defaults for whatever is not listed):
 - MRA options: **FRESH**
 - Alignment modes available: **ALL**
 - Options available: **ROTATION_FIRST**
 - Correlation functions available: **CCF**
 - Input file, image loc#: **ali0**
 - Output file, loc#: **ali1**
 - Original (pretreated) file, loc#: **flt**
 - Reference file, loc#: **mra-ref1**
 - Option to filter the reference(s): **NO_FILTER**
 - Also mirror alignment: **NO**
 - Max shift (pixels or fract. of inner radius): **0.1**
 - Min, max rotation angle: **-180,180**
 - Precision for rotational alignment: **LOW**
 - Min,max radius for rot alignment: **0.0,0.9**
 - Number of alignment iterations: **3**
 - Full output of all parameters: **NO**
3. After the alignment has completed, you should run a new round of MSA (**MSA-RUN** / **MSA-CLASSIFY** / **MSA-SUM**) on the aligned dataset and generate new class averages. You can iterate this MSA / MRA cycle until you feel your class averages are of sufficient quality. Due to time constraints however, we will continue straight on with these class averages.
4. Select the best class averages and extract them to a new file using the command **EX-COPY**. Call the output **jnk1**.



8 Hands-On: Angular Reconstitution - Initial Angular Assignment

Once you have selected the best class averages you need to find their relative orientation (Euler angles). This hemocyanin is a molecule with D3 symmetry. This rather high degree of symmetry makes the initial angular assignment much easier than with lower degrees of symmetry. Can you explain why?

1. Filter the class averages using **INCORE-PREP**. This can help with the accuracy of the angular assignment. Use the following values:
 - Output: **jnk1-bp**
 - Low-freq cut-off (pixels or fraction): 0.05
 - Transmission below cut-off (0.0-1.0): 0
 - High-freq cut-off (pixels or fraction): 0.6
 - Mask radius, drop-off: 0.9, 0.2
 - Invert the image densities: **NO** (particles are expected to be white)
2. Select three class averages that look very different from each other. Write those numbers down.
3. To assign angles by angular reconstitution we will use the IMAGIC command **EULER** option NEW with following values:
 - Pointgroup symmetry: **D3**
 - Minimal stay-away from equator (Beta=90): **0**
 - Option for angular reconstitution: **NEW**
 - Option of NEW: **FRESH**
 - Input (classum) images, NO loc#s: **jnk1-bp**
 - Location number(s) wanted: the three selected class numbers (separate the numbers by a semicolon)
 - Output (ordered) image file: **my_ordered**
 - Sinogram file, NO loc#s: **my_sino**
 - ASQ filter the sinogram lines: **YES**
 - Linear mask radius for sinograms: **0.9**
 - Output sinecorr file, NO loc#s: **my_sine**
 - Wanted angular increment in search: **5**
4. After the program has finished you should check that the angles obtained for the three class averages are reasonably apart from each other. If this is not the case, you should select different class averages and do it again.

NOTE: Angular reconstitution updates the header of the file containing the class averages with the Euler angles found. In this case, and only in this case (option NEW), the class averages file is not updated but another file is generated instead. Here we called it "**my_ordered**" and this will now contain the class averages with the Euler angles in the header.



9 Hands-On: Initial 3D reconstruction

9.1 3D Reconstruction

1. Once you have assigned angles to your class averages you are ready to build your first 3D using the IMAGIC command **THREED-REC**:
 - Use option **ALL_IN_ONE_3D**
 - Pointgroup symmetry to be used: **D3**
 - Use default 3D reconstruction options: **YES**
 - Input 2D (classsum) images, loc#: **my_ordered**
 - Source of Euler angles: **ANGREC_HEADER_VALUES**
 - Output 3D rec. filename, loc#: **my_ordered-map**
 - Output file for reprojections, NO loc#: **my_ordered-rep**
 - Output file for error projections, NO loc#: **err**
 - Mask the reconstruction: **NO**
 - Hamming window factor: **0.5**
 - Object size as fraction of image size: **0.8**

Here we use a relatively low hamming window factor due to the low number of class averages used to build the 3D. This is a filtering factor and affects the resolution. It will be increased in future rounds.

2. It is important to check “by eye” how well the reprojections match the class averages. To do this, display the class averages “**my_ordered**” you used to create the 3D in one window and the reprojections, “**my_ordered-rep**”, in another. The **DISPLAY** settings such as **SIZE** and **SCALE** should be the same in both windows. By flicking back and forth between the two overlaid windows compare how well these two match. If a class average does not match the re-projection you should remove it using **EX-COPY** and build the 3D again.

You can visualize the 3D you have created with the command **DISPLAY**. The display will show slices through the 3D from bottom to top.



9.2 3D Masking

We will use the **THREED-AUTO-MASK** command to automatically generate a mask for our reconstruction. We use a mask to remove the influence of the noise when generating references for alignments or anchor sets for angular reconstitution. **THREED-AUTO-MASK** follows a two step procedure: firstly, a modulation volume is calculated. Secondly, this is binarised.

1. Select the option **DO_IT_ALL**.
2. We will use **MODULATION**, as opposed to **VARIANCE** (check the help for this question if you are curious why we do this).
3. In this program, the filter parameters for the modulation calculation need to be specified manually. Here are the suggest values:
 - for the bandpass parameters: **0.12, 0.9**
 - for the low-pass parameter: **0.2**

A good starting point for the lower bound of the bandpass parameter is the smallest dimension of our object. The other bandpass parameter is a noise-reducing low-pass filter, but in our case, the pixel size is so large (4.8 Å) that noise levels are already pretty low, so 0.9 should be sufficient in this case. Lastly, the low-pass parameter needs to be \geq the low bound of the bandpass. The higher this filter is set, the "finer/sharper" the mask will be.

4. Finish running **THREED-AUTO-MASK** (just use the defaults for the last two answers for now). Display the output modulation volume. Then, start **THREED-AUTO-MASK** again, change the filter parameters (for example, change the low-pass filter to **0.1**), and observe how the modulation volume is affected.
5. Now, let's play with the last two questions. They deal with the binarisation step of the **THREED-AUTO-MASK** procedure. The easiest way is to use option **AUTOMATIC**. All you then need to set is the percentage of the reconstruction volume which you expect to belong to the object. In our case, **15%** (i.e. 150,000 out of 1,000,000 voxels in our volume) is a good value.
6. How happy are you with your mask? An ideal mask removes noise outside of the object, but leaves our object completely intact.

These next few steps you will create a soft version of the mask. These steps are optional.

7. A soft mask is one where there is a smooth "transition" between the object voxels and the solvent voxels.
8. Filter the output mask file from **THREED-AUTO-MASK** (not the masked 3D). Use the **THREED-FILTER** command, option **GAUSS**, halfwidth of **0.3**.
9. **DISPLAY** the original mask from **THREED-AUTO-MASK** and the "softened" mask from the filter operation. You should notice that the mask appears to "start" closer to the object, and may even "eat into" it. Check carefully whether this is the case. If so, to avoid this, we need to dilate our mask before we soften it.
10. Run **THREED-DILATE**, radius **2** and threshold **0.001** (or any value between 0 and 1). Filter the output as in step 8, and compare with the original mask from **THREED-AUTO-MASK**.



10 Hands-On: Angular Reconstitution – Refinement with an Anchor Set

1. Once you have a masked 3D project it to create a set of references for angular refinement using the IMAGIC command **THREED-FORWARD-PROJECT**. Remember to reproject in the asymmetric triangle of **D3**. An angular increment of **18°** will give you enough references at this stage. Normalize the set of references using **NORM-VAR**. The set of references used for angular refinement is called an anchor-set. Call the output **eul-ref1**.
2. We will assign angles to all the class averages using the anchor set with the **EULER** command and the **ANCHORSET** option:
 - Pointgroup symmetry: **D3**
 - Option for angular reconstitution: **ANCHOR_SET**
 - Please specify option: **FRESH**
 - How are the input (classum) images available: **IMAGES**
 - Input (classum) images, image loc#s: **jnk1-bp**
 - Sinogram file, image loc#s: **my_sino**
 - ASQ filter the sinogram lines: **YES**
 - Linear mask radius for sinograms: **0.9**
 - How is the anchor set available: **IMAGES**
 - Input anchor set IMAGES: **eul-ref1**
 - Output anchor set sinograms: **my_arsino**
 - Output sinecorr file, NO loc#s: **my_sine**
 - Delete output sinecorr file(s): **YES**
 - Wanted angular increment in search: **2.0**
 - Use **PEAK_OVER_STDV** criterion: **YES**
3. In general, class averages with a low angular error tend to be better. In this case we will use the angular error to select **200** of the “best” class averages using the command **EX-COPY**, option **SORT**, with criterion **ANGULAR_ERROR** and option **UP** and select **200** class averages, so that the class averages with the lowest angular reconstitution error will be at the beginning of the output file. You should always check “by eye” as those class averages are not necessarily the best ones. Call the output **bst1**.
4. Use these 200 class averages to perform a 3D reconstruction as before. In this case, use a hamming window of **0.9**.
5. When **THREED-REC** has completed it will re-project the 3D in the same direction as the input class averages. It will then calculate an error based on the correlation between the re-projections and the input class averages. If one re-projection has a much higher error than the rest, i.e. if the error suddenly jumps then you could exclude that class average using **EX-COPY** and recalculate the 3D without it.



- Once you have a satisfactory 3D, filter it to remove unwanted frequencies using the command **THREED-FILT**. The filter values that you choose will depend on what stage you are at in refinement. Typically it is used to correct the dampening of the power in your original data due to the envelope terms of the CTF. We want to filter out dominant low frequencies to achieve this. In addition, we want to remove high frequencies beyond the current resolution to remove noise. Use:
 - Low-freq cut-off (pixels or fraction): **0.25**
 - Transmission below cut-off: **0**
 - High-freq cut-off (pixels or fraction): **0.9**
- Use **THREED-AUTO-MASK** to mask the filtered 3D.

11 Hands-On: Refinement

- Repeat Section 10 iteratively until the angles stabilize.
- Project the last masked 3D with an angular increment of **10°** with the command **THREED-FORWARD**.
- Normalize and zero float these projections with the command **NORM-VAR** and use them as references for a new round of **M-R-A**. The input file is the aligned output of your last **M-R-A**. The original pretreated file does not change.
- Run a new MSA cycle as in Section 6 and generate class averages.
- Use your last anchor set to assign angles to these class averages and refine the angles as before. This refinement loop is repeated many times until you reach your desired resolution or convergence.

This iterative refinement is the most time consuming process. As the quality of your reconstruction increases you can use a finer angular increment for projecting your **M-R-A** references. Also one can go back to the un-coarsened dataset if the limit of the current sampling is reached.

12 Hands-On: Fourier Shell Correlation (FSC)

You can assess the resolution of your structure by Fourier Shell Correlation (FSC) following these steps:

- Split the class averages using the **EX-COPY** command, option **EXTRACT, INTERACTIVE**. Select all the locations in your file and then use option **ODD**. Repeat this process giving a separate output file with option **EVEN**.
- Then, build two 3D structures with the two subsets created above. In this case we do not need the re-projections.
- Using command **F-S-C** and the two 3Ds created above to create a PLT file containing the Fourier shell correlation curve.
- PLOT** this curve and estimate your resolution based on where the 1/2 bit curve crosses the FSC curve.

Whilst the 1/2 bit curve provides a single figure for your resolution it is important to always take into account the curve as a whole when judging the quality of the reconstruction.

