bio-protocol

Published: Apr 20, 2022

Bsoft: Image Processing for Structural Biology

J. Bernard Heymann*

National Cryo-EM Program, Cancer Research Technology Program, Frederick Office of Scientific Operations, National Cancer Institute, National Institutes of Health, Frederick, MD 21702, USA
*For correspondence: heymannb@mail.nih.gov

Abstract

Bsoft is a software package primarily developed for processing electron micrographs, with the goal of determining the structures of biologically relevant molecules, molecular assemblies, and parts of cells. However, it incorporates many ways to deal with images, from the mundane to very sophisticated algorithms. This article is an introduction into its use, illustrating that it is an extensive toolbox, for manipulating and understanding images. Bsoft has over 150 programs, allowing the user an infinite number of ways to process images. These programs can be executed on the command line, or through the interactive program called brun. The main visualization program is bshow, providing numerous ways to manipulate and interpret images. The primary aim is to provide the user with powerful capabilities, including processing large numbers of images. An important additional aim is to make it as accessible as possible, making it easier to deal with image formats and features, and enhance productivity.

Keywords: Image file formats, Cross-correlation, Projection, Image filtering, Image display, Symmetry, Fourier transformation, Power spectrum, Image masking, Structural biology, Electron microscopy

This protocol was validated in: Protein Sci (2018), DOI: 10.1002/pro.3293

Cite as: Heymann, J. B. et al. (2022). Bsoft: Image Processing for Structural Biology. Bio-protocol 12(08): e4393. DOI: 10.21769/BioProtoc.4393.

Background

Bsoft resulted from an effort to deal with the plethora of image formats during a time when we were building a prototype database for microscopic images called BioImage (Carazo *et al.*, 1999). BioImage contributed to the establishment of the Electron Microscopy Databank (EMDB) at the EBI in Hinxton, UK (https://www.ebi.ac.uk/emdb/). Bsoft has been used in many projects, to determine the structures of proteins and assemblies from cryo-electron micrographs, these days called cryoEM.

There are three conceptual domains for using Bsoft: (i) dealing purely with images, (ii) producing and handling the very large number of parameters or metadata required in processing electron micrographs, and (iii) interpreting 3D reconstructions through segmentation and models (Heymann, 2001; Heymann and Belnap, 2007; Heymann *et al.*, 2008; Heymann, 2018a; Heymann, 2020; 2021). This article covers the entry point to image processing in Bsoft, providing the basis for understanding the more complicated processing involved under the last two domains.

Procedure

A. Installation

The easiest way to install Bsoft is from already compiled binaries. To compile it, several associated packages must be installed, with potential complications that require expertise beyond the typical user. The Bsoft binary and source distributions are freely available at http://bsoft.ws.

1. Linux

   On any version of Linux, the Tcl/Tk and libxml2 libraries must be installed. The binaries for Bsoft compiled on CentOS should work on other Linux distributions. Download the archive file with a name such as bsoft2_1_0_CentOS_7.8.2003.tgz (this file name will change with updates).

   The typical install location for Bsoft is at /usr/local. This usually requires root access. Without root access, installing Bsoft in any other directory should also work.
   Change to the install directory, and unpack the distribution, *e.g.*,
   tar xzvf bsoft2_1_0_CentOS_7.8.2003.tgz

   Then run the setup program in the same directory:
   ./bsetup
   Open another shell, and test the installation, by running **bimg** or **bshow**. One common problem is that the path to the Bsoft programs is not set up correctly. In the bsoft directory, there should be two text files, called bsoft.bashrc and bsoft.cshrc. They set a variable called BSOFT that should point to the bsoft installation (*e.g.*, /usr/local/bsoft).

2. Mac OSX

   Download the disk image file called Bsoft2.dmg. Double click on it, and follow instructions. This usually requires administrative rights. The mail package will install in /usr/local/bsoft, while the programs Bshow and Brun will install in /Applications. In a Terminal shell, type **bimg** and **bshow**, to make sure the programs are properly installed. If the programs cannot be found, the environmental setup (/usr/local/bsoft/bsoft.cshrc or /usr/local/bsoft/bsoft.bashrc) needs to be fixed, to point to the correct places. Make sure the BSOFT variable indicates the correct directory, *e.g.*,
   BSOFT=/usr/local/bsoft

B. Bsoft programs

Bsoft contains many programs to do image processing and molecular modeling, mainly associated with electron microscopy data, but also touching on related fields. Most of the programs are command line, with numerous options, and allowing extensive use of scripts to process large amounts of data. Most of the Bsoft programs are launched from the command line. These can be accessed through the Terminal application on Mac OSX or Linux, which opens a Unix-type shell. Two programs are interactive. The first is **brun**, a graphical interface to the command line programs, to make it easier to deal with all the options. The second is **bshow**, the main graphical program used to display images, and handle many aspects of processing and interpreting electron microscopic data.

1. Command line programs

   Typing the program name on the command line without any options gives a listing of all the options to use with the program. For instance, type:

   bhead

   It will give the output in Figure 1A, with a description and options. Note that the option names are case sensitive, and can be abbreviated if the abbreviation is unique.

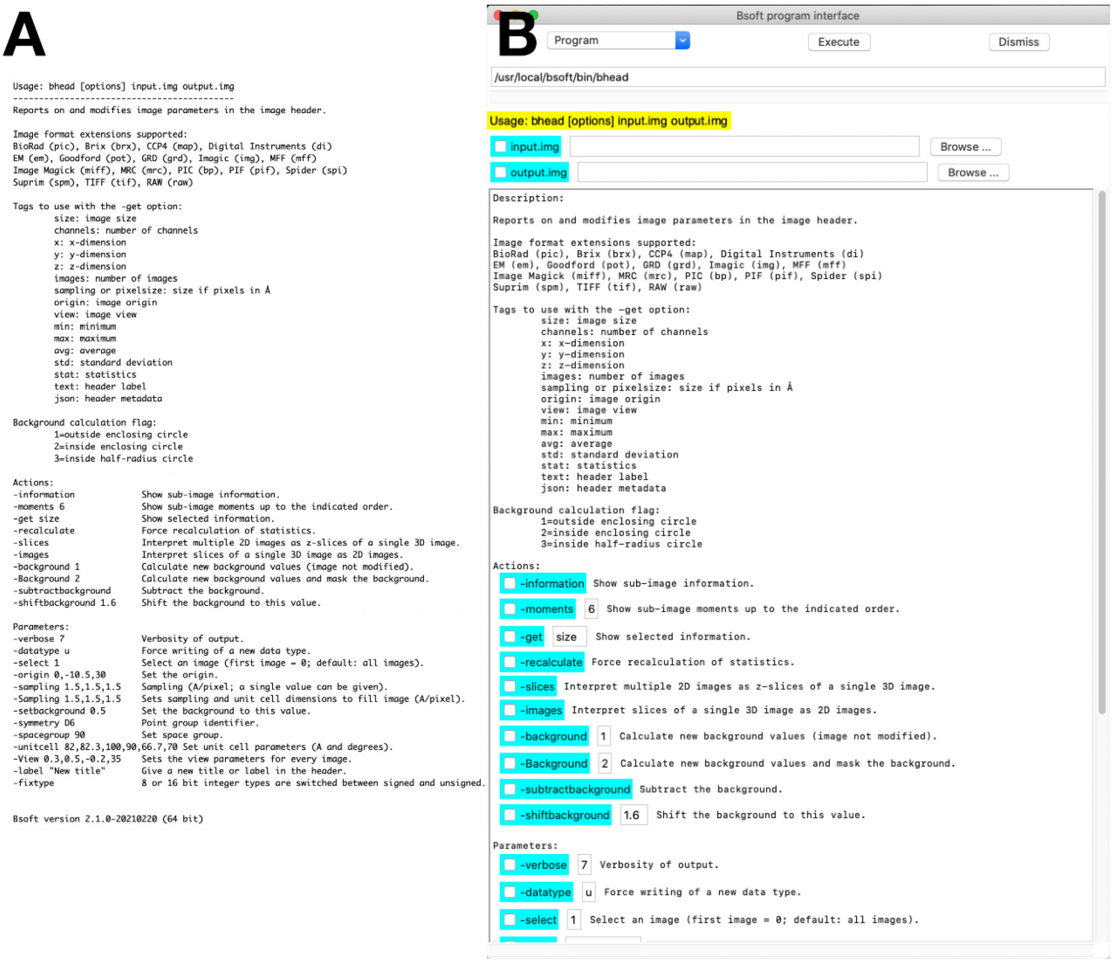
2. Interactive programs

   a. brun

   In **brun**, the same options are accessible. Type:

   brun

   And then select **bhead** from the program list (see Figure 1B). At the top is a text box with the command line. Selecting options by clicking in the radio buttons will add them to the command line. Options with arguments can be changed, to update the command line. This is one way to construct a command line without typing the options, and getting the syntax right.



```
Usage: bhead [options] input.img output.img
-------------------------------------------
Reports on and modifies image parameters in the image header.

Image format extensions supported:
BioRad (pic), Brix (brx), CCP4 (map), Digital Instruments (di)
EM (em), Goodford (pot), GRD (grd), Imagic (img), MFF (mff)
Image Magick (miff), MRC (mrc), PIC (bp), PIF (pif), Spider (spi)
Suprim (spm), TIFF (tif), RAW (raw)

Tags to use with the -get option:
        size: image size
        channels: number of channels
        x: x-dimension
        y: y-dimension
        z: z-dimension
        images: number of images
        sampling or pixelsize: size if pixels in Å
        origin: image origin
        view: image view
        min: minimum
        max: maximum
        avg: average
        std: standard deviation
        stat: statistics
        text: header label
        json: header metadata

Background calculation flag:
        1=outside enclosing circle
        2=inside enclosing circle
        3=inside half-radius circle

Actions:
-information             Show sub-image information.
-moments 6               Show sub-image moments up to the indicated order.
-get size                Show selected information.
-recalculate             Force recalculation of statistics.
-slices                  Interpret multiple 2D images as z-slices of a single 3D image.
-images                  Interpret slices of a single 3D image as 2D images.
-background 1            Calculate new background values (image not modified).
-Background 2            Calculate new background values and mask the background.
-subtractbackground      Subtract the background.
-shiftbackground 1.6     Shift the background to this value.

Parameters:
-verbose 7               Verbosity of output.
-datatype u              Force writing of a new data type.
-select 1                Select an image (first image = 0; default: all images).
-origin 0,-10.5,30       Set the origin.
-sampling 1.5,1.5,1.5    Sampling (A/pixel; a single value can be given).
-Sampling 1.5,1.5,1.5    Sets sampling and unit cell dimensions to fill image (A/pixel).
-setbackground 0.5       Set the background to this value.
-symmetry D6             Point group identifier.
-spacegroup 90           Set space group.
-unitcell 82,82.3,100,90,66.7,70 Set unit cell parameters (A and degrees).
-View 0.3,0.5,-0.2,35    Sets the view parameters for every image.
-label "New title"       Give a new title or label in the header.
-fixtype                 8 or 16 bit integer types are switched between signed and unsigned.


Bsoft version 2.1.0-20210220 (64 bit)
```

/usr/local/bsoft/bin/bhead

Usage: bhead [options] input.img output.img

□ input.img

□ output.img

```
Description:

Reports on and modifies image parameters in the image header.

Image format extensions supported:
BioRad (pic), Brix (brx), CCP4 (map), Digital Instruments (di)
EM (em), Goodford (pot), GRD (grd), Imagic (img), MFF (mff)
Image Magick (miff), MRC (mrc), PIC (bp), PIF (pif), Spider (spi)
Suprim (spm), TIFF (tif), RAW (raw)

Tags to use with the -get option:
        size: image size
        channels: number of channels
        x: x-dimension
        y: y-dimension
        z: z-dimension
        images: number of images
        sampling or pixelsize: size if pixels in Å
        origin: image origin
        view: image view
        min: minimum
        max: maximum
        avg: average
        std: standard deviation
        stat: statistics
        text: header label
        json: header metadata

Background calculation flag:
        1=outside enclosing circle
        2=inside enclosing circle
        3=inside half-radius circle

Actions:
```

□ -information Show sub-image information.

□ -moments 6 Show sub-image moments up to the indicated order.

□ -get size Show selected information.

□ -recalculate Force recalculation of statistics.

□ -slices Interpret multiple 2D images as z-slices of a single 3D image.

□ -images Interpret slices of a single 3D image as 2D images.

□ -background 1 Calculate new background values (image not modified).

□ -Background 2 Calculate new background values and mask the background.

□ -subtractbackground Subtract the background.

□ -shiftbackground 1.6 Shift the background to this value.

Parameters:

□ -verbose 7 Verbosity of output.

□ -datatype u Force writing of a new data type.

□ -select 1 Select an image (first image = 0; default: all images).

Figure 1. Bsoft program options.
(A) Options for the program **bhead** obtained by just entering its name on the command line. (B) Accessing command line programs through the program **brun**, showing the output for the program **bhead,** with options that can be entered interactively.

b. bshow

The main interactive program in Bsoft is bshow (a Tcl/Tk script-based interface to the capabilities in Bsoft). On a Linux platform, it is typically launched with an image file name:

```
bshow input.mrc &
```

On Mac OSX it can be similarly launched, or multiple instances can be launched by specifying multiple input files:

```
  bshowX in1.mrc in2.mrc …
```

It will be used in several of the following sections, to examine images.

C. Image basics

1. Examining a map

a. An example map

To start, obtain a cryoEM map from the Electron Microscopy Databank (EMDB: https://www.ebi.ac.uk/pdbe/emdb/index.html/). These maps can be very large, so it is better to choose a small map, so that processing times are not very long. I selected human methemoglobin (EMD-0407) (Herzik *et al.*, 2019), one of the smallest proteins solved by cryoEM. The complete package is approximately 208 MB, and contains multiple files. Open the map and model in UCSF Chimera (files map/emd_0407.map and fittedModels/PDB/6nbc.ent), which should give rise to a display as in Figure 2. It shows a clear C2 (cyclic two-fold) symmetry, as expected for hemoglobin. The model fits into the density, but there are pieces that may not fit well. The background also has a lot of "dust" originating from the noisy micrographs.

b. Command line examination of a map file

Navigate to the map directory and type the following:

```
bhead -verb 7 emd_0407.map
```

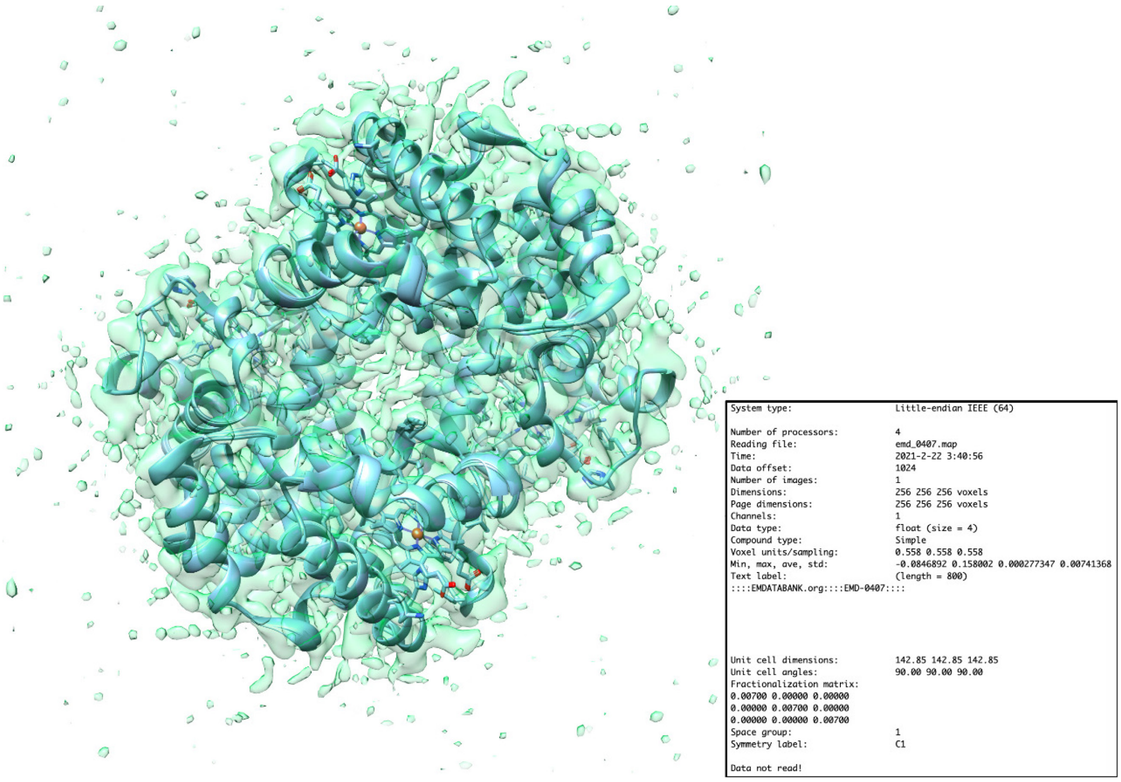
This will give rise to an output as in the Figure 2 inset. The **-verbose** option in all Bsoft programs controls how much output is generated. If it is not used, no output will be generated. This is typically used in scripts to not produce a lot of output. An argument of 1 gives minimal output, which is often desired for programs doing a lot of processing. The most important pieces in the output when reading an image file are the number of images, the dimensions given in pixels/voxels, and the number of channels. The number of channels indicate the number of data values per voxel, which is just one for typical cryoEM maps. Bsoft also understands complex and color images with two and three channels, respectively. In addition, the channels relate to the compound type, with the most common types "Simple" (grayscale or one channel), "Complex" (2 channels), and "RGB" (3-channel color). The data type is important, because it can cause some operations to yield strange or undesirable results. In most cases, a floating point data type that takes 4 bytes of storage per value is the most appropriate type. However, with very large micrographs, the data can be encoded as single byte integer values (0–255), to save space. Most programs in Bsoft have an option to change the data type.

The pixel/voxel units or sampling are crucially important in cryoEM, because they relate the image scale to the real physical scale. It is advisable to set the correct pixel/voxel size as early as possible, in any data processing protocol. Well-configured data acquisition will typically include the sampling in the image header, but sometimes that needs to be explicitly given when processing images.

Another line in the output gives the overall statistics of the image as the minimum, maximum, average, and standard deviation. This provides an easy check, to ensure the image has data with the expected distribution. This can be forced to update with:

```
bhead -verb 7 -recalc emd_0407.map
```

The text label from other software packages may contain a variety of information, some useful, but mostly not. In Bsoft, the command line that generated the file is written into the text label. So, it is almost always possible to retrieve the exact command used to generate an image.



```
System type:                    Little-endian IEEE (64)

Number of processors:           4
Reading file:                   emd_0407.map
Time:                           2021-2-22 3:40:56
Data offset:                    1024
Number of images:               1
Dimensions:                     256 256 256 voxels
Page dimensions:                256 256 256 voxels
Channels:                       1
Data type:                      float (size = 4)
Compound type:                  Simple
Voxel units/sampling:           0.558 0.558 0.558
Min, max, ave, std:             -0.0846892 0.158002 0.000277347 0.00741368
Text label:                     (length = 800)
::::EMDATABANK.org::::EMD-0407::::




Unit cell dimensions:           142.85 142.85 142.85
Unit cell angles:               90.00 90.00 90.00
Fractionalization matrix:
0.00700 0.00000 0.00000
0.00000 0.00700 0.00000
0.00000 0.00000 0.00700
Space group:                    1
Symmetry label:                 C1

Data not read!
```

Figure 2. EMDB entry 0407.
Human methemoglobin rendered in UCSF Chimera (Pettersen *et al.*, 2004), with its atomic model (PDB: 4N7P). The inset shows the information for the map, using the command line: **bhead -verb 7 emd_0407.map**.

c. Examining a map in **bshow**

Open the map in the interactive program:

bshow emd_0407.map

The initial display (Figure 3A) is disappointing, because one would expect to see some nice density. However, the origin is set to (0,0,0). To fix this, select the menu item, "Image/Set origin", and click on the "Center" button in the dialog box. The origin is now (128,128,128) (Figure 3B). Move the "Slice" slider to 128, to show slice 128, with the origin indicated by a blue circle. The "Min" and "Max" sliders can be moved, to change the display density range. Note that none of these operations change the map data, just how it is displayed. Clicking on the "Animate" button starts a loop going through the slices. There are numerous features in **bshow** to handle images, covering simple image operations, as well as more complex ones required for micrograph processing. Most of these and many more can be handled from the command line.

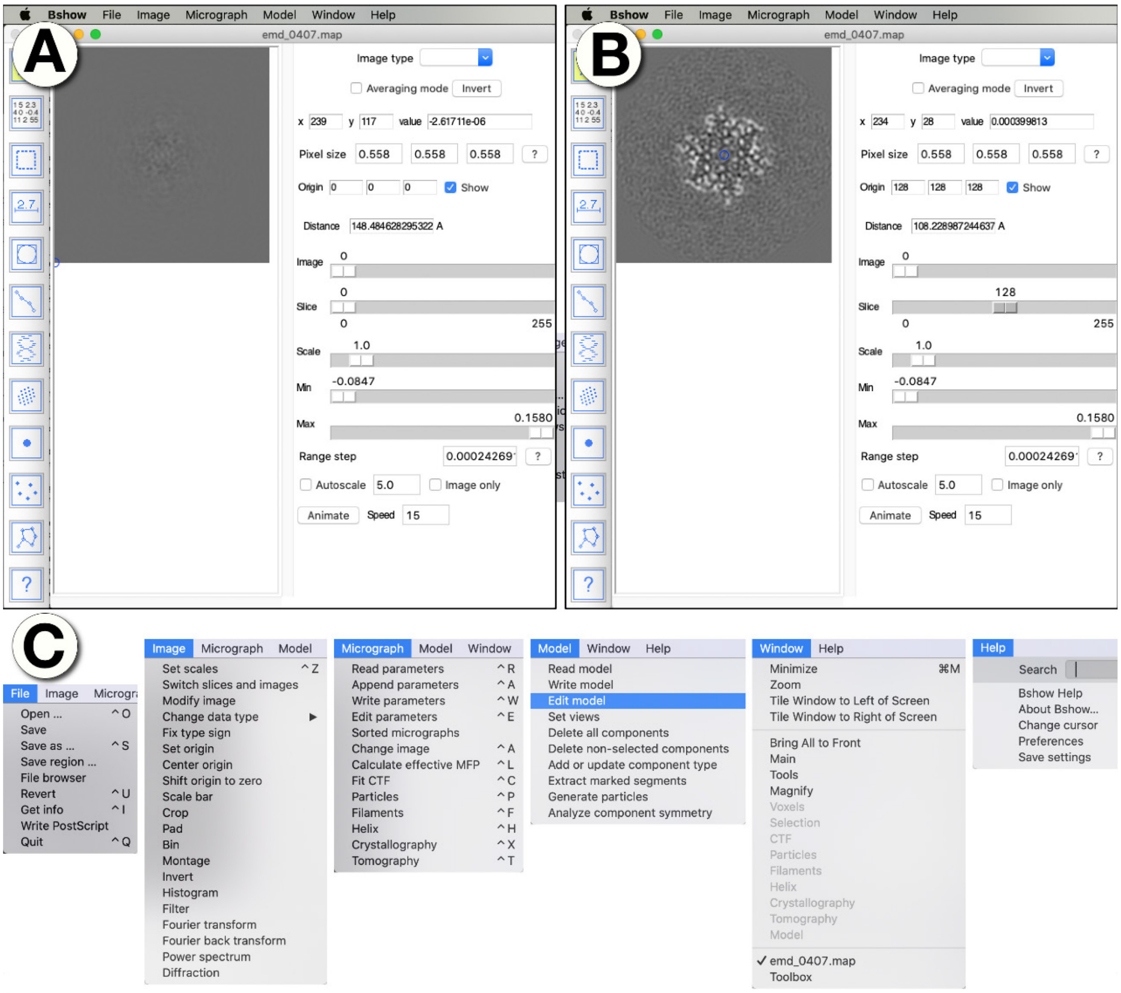
The "File/File browser" menu item opens a window for easy selection of files to open. It shows the contents of the directory where **bshow** was launched.

The menus in **bshow** deal with different aspects (Figure 4C):

1. File: Reading/writing image files and getting information.
2. Image: Manipulating the image data.
3. Micrograph: Handling the parametric metadata associated with micrographs.
4. Model: Handling models built onto 3D reconstructions.

5. Window: A list of available and open windows.
6. Help: Information and preferences.

This protocol mostly covers the use of the first two menus.



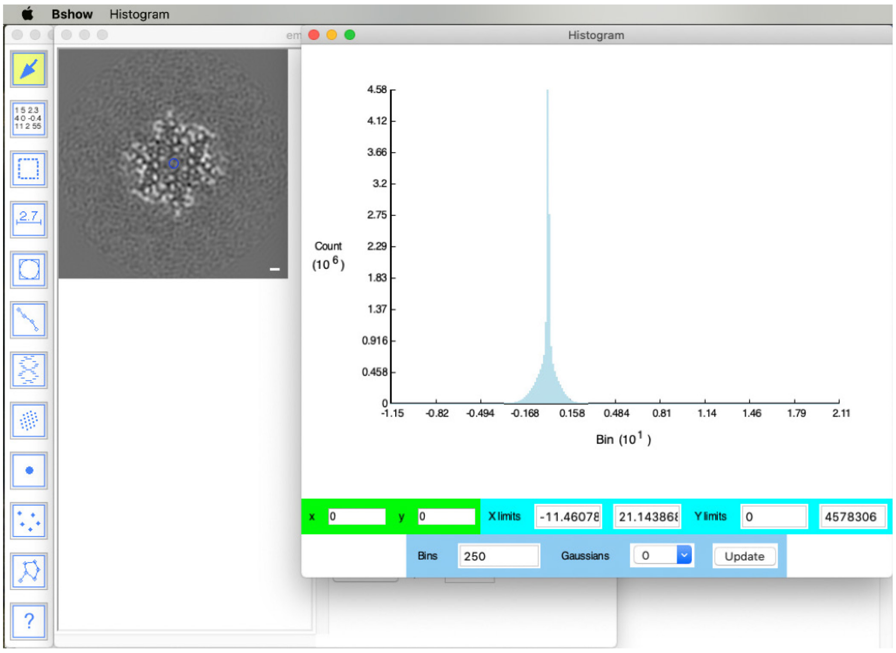
A: x 239 y 117 value -2.61711e-06
Pixel size 0.558 0.558 0.558
Origin 0 0 0 ☑ Show
Distance 148.484628295322 A
Image 0
Slice 0 (0–255)
Scale 1.0
Min -0.0847
Max 0.1580
Range step 0.00024269
□ Autoscale 5.0 □ Image only

B: x 234 y 28 value 0.000399813
Pixel size 0.558 0.558 0.558
Origin 128 128 128 ☑ Show
Distance 108.228987244637 A
Image 0
Slice 128 (0–255)
Scale 1.0
Min -0.0847
Max 0.1580
Range step 0.00024269
□ Autoscale 5.0 □ Image only

Figure 3. Display in bshow.
(A) Default display of map emd_0407 from the EMDB in **bshow.** (B) Display of the map after fixing the origin. (C) The menus in bshow, as displayed in Mac OSX.

d. Histogram

The distribution of values in an image is captured in a histogram, where the range of values is divided into a number of bins, and the values falling within each bin counted. In **bshow**, select the "Image/Histogram" menu item to open the histogram graph (Figure 4).



x 0 y 0 X limits -11.46078 21.14386 Y limits 0 4578306

Bins 250 Gaussians 0

Figure 4. Histogram of the rescaled map emd_0407.
Parts of the histogram can be examined by selecting a rectangle with the mouse.

2. Image formats
Bsoft started out many years ago, to deal with all the different image formats. The aim is to adhere as closely as possible to the specification of each format. Therefore, many issues can be resolved by fixing images that come from other software by rewriting them with Bsoft programs. The format is identified by the file extension, the part after the last period. It is therefore always important to always have an extension for a file. Also, spaces and strange characters (such as "-", "/", "&" and "$") in file names can create problems for command line programs. Please do not use spaces or special characters in names.

The most important format in cryoEM is the MRC (Medical Research Council) format (extensions: .mrc, .mrcs), that, in its latest specification, can encode most of the information (Cheng *et al.*, 2015). The closely related CCP4 format (extension: .map) should be compatible with the MRC format, but there are minor differences. Many other formats are supported, including color formats such as TIFF, PNG, and JPEG. Note that not all formats support all data types, and it may be necessary to change the data type when rewriting into a new format. If many images need to be changed to a new format, use the program **bconvert**.

3. Slices vs. images
Software packages often make no distinction between a set of 2D images and the slices of 3D images. However, from a data processing perspective, it is important to understand whether operations will be applied to 2D images or 3D volumes. In Bsoft, the distinction between 2D images and slices of 3D volumes is explicit, and indicated in the header output (Figure 1). Where the slices and images are switched, it can be rectified in several programs.

To go from slices to 2D images:

```
bhead -verb 7 -images in.mrc out.mrc
```

To go from a stack of 2D images to 3D slices:

```
bhead -verb 7 -slices in.mrc out.mrc
```

4. Montage
A file with a 3D map or multiple 2D images can be montaged, *i.e.*, displayed in a 2D array (Figure 5). This montage can be saved to an image for inclusion in documents. The same result can be obtained using the command line program, **bmontage**.

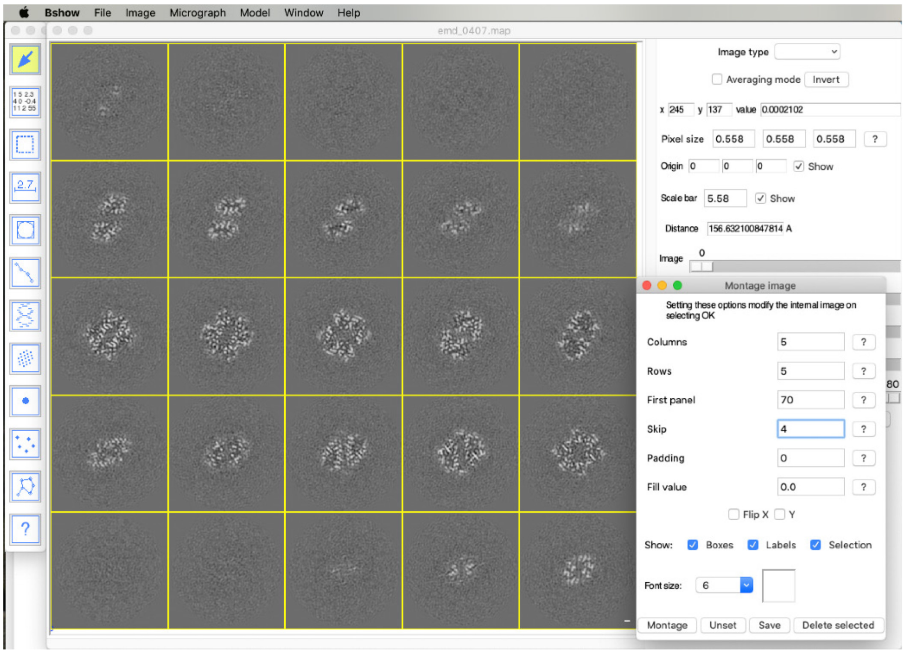


Image type

☐ Averaging mode

x 245 y 137 value 0.0002102

Pixel size 0.558 0.558 0.558

Origin 0 0 0 ☑ Show

Scale bar 5.58 ☑ Show

Distance 156.632100847814 A

Image 0

Montage image

Setting these options modify the internal image on selecting OK

Columns 5

Rows 5

First panel 70

Skip 4

Padding 0

Fill value 0.0

☐ Flip X ☐ Y

Show: ☑ Boxes ☑ Labels ☑ Selection

Font size: 6

Figure 5. Montage of slices from the map emd_0407 in bshow.

5. Splitting images
The slices of a 3D image or multiple images in a file can be separated into individual files:
bsplit -verb 7 -images -select 100,110-120 emd_0407.map split/emd_0407.mrc
Note that the -images option will reinterpret the image as 2D images for splitting. Omitting the -select option results in generating all the slices as separate 2D images.

6. Combining images
Images can be combined into one file, adjusted to the maximum size of the images:
bcat -verb 7 -slices -out emd_0407_cat.mrc split/*.mrc
The -slices option specifies that the images should be packed as the slices of a 3D image.

D. Image manipulation

1. Geometric operations
The most used operations on images involve the geometric transformations, translation, rotation and scaling. The programs in Bsoft are designed to give the user full control over all aspects, but that comes with a requirement to understand the conventions.
Because the origin is very important, the origins in a map can be examined using:
bhead -verb 7 -info emd_0407.map
The main program to do geometric operations is **bint** (interpolation).
To translate the map, type the following command:
bint -verb 7 -ori cen -trans -100,100,0 emd_0407.map emd_0407_t.mrc
The "-origin" option sets the origin to the center of the map. This is important, because most operations depend on how it is set. The "-translate" option moves the map by the indicated number of voxels. In this

case, it moves it to the upper left corner, cutting of part of the density. Also note that the origin has been adjusted to reflect this move.

The origin is particularly important for rotations, because it specifies the center of rotation.
To rotate the map, type the following command:
bint -verb 7 -ori cen -rotate 0,0,1,33 emd_0407.map emd_0407_r.mrc
The rotation is specified by a 3-value rotation axis [in this case, the z-axis (0,0,1)], and a rotation angle in degrees.

To change the map scale, type the following command:
bint -verb 7 -ori cen -scale 0.5,0.5,0.5 -size 128,128,128 -trans -64,-64,-64 emd_0407.map emd_0407_s.mrc
As with rotation, the change in scale is around the origin. This means that if we want to fit it into a smaller size, we also must translate it correctly. The result of this command line now has the origin in the center of the new map, and the sampling is twice the original. All three transformations can be done at the same time.

2. Binning
Images are often rescaled by binning, a form of rescaling by integer factors.
The most common is binning by 2, *i.e.*, averaging every 2 × 2 pixels, or 2 × 2 × 2 voxels:
bint -verb 7 -bin 2 emd_0407.map emd_0407_b2.mrc
This binning is isotropic, but anisotropic binning can also be done, by specifying the bin factor for every dimension.

3. Density operations
Image processing is about manipulating the data in images. Some of the simple operations changes the density distributions, with the goal of eliminating problems, or setting a distribution with specific features. The program **bimg** has options to truncate the data (eliminate values above and below thresholds), or set the minimum and maximum to desired values.
An operation extensively used throughout Bsoft is to set the average to zero, and the standard deviation to one:
bimg -verb 7 -ori cen -rescale 0,1 emd_0407.map emd_0407_rs.mrc
Now that the density distribution is well defined, many subsequent operations will work well.

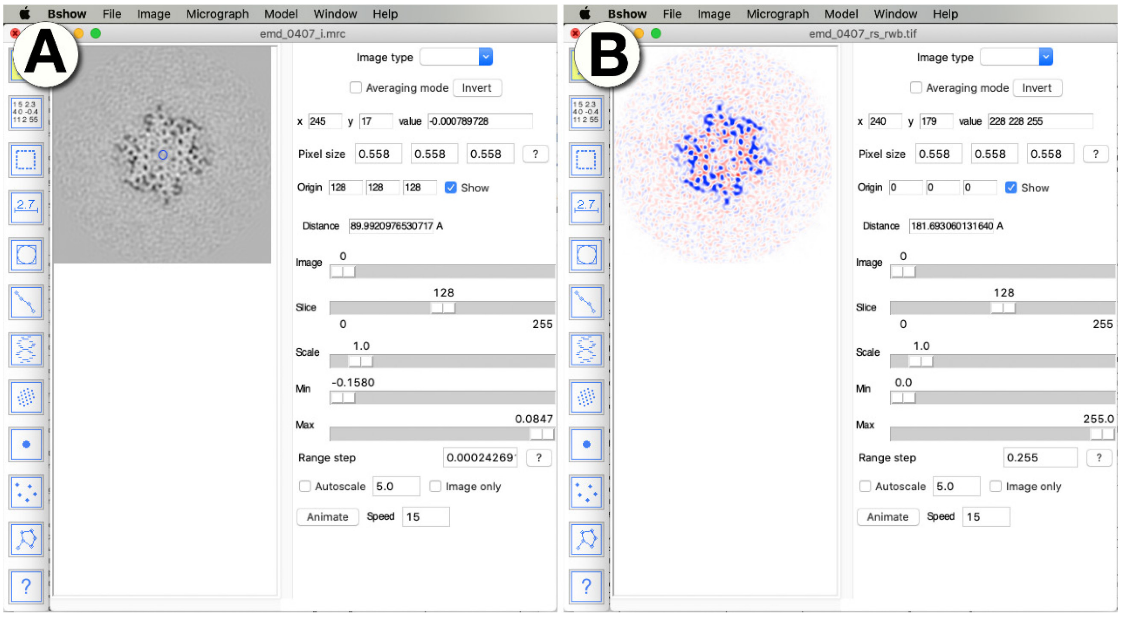
The micrographs of cryoEM show the biomolecules as dark on a slightly lighter background, *i.e.*, lower values indicate density (also referred to as negative density). When it comes to interpreting the density, the intuition is to have higher values indicating greater density (*i.e.*, positive density). The map of our example was deposited in the EMDB as a positive density map. To see what the original reconstructed map looks like corresponding to the density seen in micrographs, the map must be inverted (Figure 6A):
bimg -verb 7 -ori cen -invert emd_0407.map emd_0407_i.mrc

Sometimes, the extreme values in an image are very large and do not contribute to the information content. The extremes can be modified by truncation to more reasonable values, or to enhace the display contrast:
bimg -verb 7 -trunc -0.05,0.05 emd_0407.map emd_0407_tr.mrc

Images can also be adjusted by simple manipulations, such as shifting the density values by adding a constant, or multiplying the density values by a constant:
bar -verb 7 -add 0.1 -mult 10 emd_0407.map emd_0407_am.mrc

4. Color
Bsoft supports various image formats that can store color encoding (TIFF, JPEG, PNG, or GRD). Color is often used to better show differences in a map.
One conversion is to a red-blue transition (Figure 6B):
bcolour -verb 7 -verb 7 -rwb -10,-0.1,0.1,10 emd_0407_rs.mrc emd_0407_rs_rwb.tif

Note that the origin is (0,0,0), because the TIFF format does not include fields for specifying it.


A
Bshow File Image Micrograph Model Window Help
B
Bshow File Image Micrograph Model Window Help

A: emd_0407_i.mrc

Image type

□ Averaging mode Invert

x 245 y 17 value -0.000789728

Pixel size 0.558 0.558 0.558 ?

Origin 128 128 128 ☑ Show

Distance 89.9920976530717 A

Image 0

Slice 128 (0 – 255)

Scale 1.0

Min -0.1580

Max 0.0847

Range step 0.00024269 ?

□ Autoscale 5.0 □ Image only

Animate Speed 15

B: emd_0407_rs_rwb.tif

Image type

□ Averaging mode Invert

x 240 y 179 value 228 228 255

Pixel size 0.558 0.558 0.558 ?

Origin 0 0 0 ☑ Show

Distance 181.693060131640 A

Image 0

Slice 128 (0 – 255)

Scale 1.0

Min 0.0

Max 255.0

Range step 0.255 ?

□ Autoscale 5.0 □ Image only

Animate Speed 15

Figure 6. Manipulating map emd_0407.
(A) Inverted map. (B) A red-blue colored map.

5. Adding images
   There are various programs to add images to each other.
   If the images are within one file, these can be summed with:
   bimg -verb 7 -images -sum emd_0407.map emd_0407_sum.mrc
   Note that here we interpreted the slices of the 3D volume as individual 2D images. The result is a projection of the map in the z-direction (see later for more on projections). The -average option is the same as the sum, except that it divides the result by the number of original images.

   In cryoEM, we usually produce two "half maps", each calculated from independent halves of the original data set.
   To sum the two half maps from the example, do:
   bop -verb 7 -add 1,0 emd_0407_half_map_1.map emd_0407_half_map_2.map emd_0407_full.map

   If we want to add multiple images in different files, we can use wildcards ("?" and "*") to specify a set of files:
   badd -verb 7 -out emd_0407_sum.mrc emd_0407_half_map_?.map

6. Multiplying images
   The most common reason in cryoEM for multiplication of images is to apply a mask.
   In our example, we can multiply the map with the provided mask:
   bop -verb 7 -mult 1,0 emd_0407.map ../masks/emd_0407_msk_1.map emd_0407_masked.mrc

7. Difference images
   If we have maps from different sample compositions, we often want to see the difference in the maps.
   The simplest comparison is to subtract the one from the other:
   bop -verb 7 -add -1,0 emd_0407_half_map_1.map emd_0407_half_map_2.map emd_0407_dif.map
   Note that, in this case, the maps are at relatively high resolution, so that the difference map does not show much beyond noise.

8. Constructing images

Sometimes we want to generate a synthetic image to use as reference or mask. The simplest approach is to place fuzzy spheres into a volume.

As an example, let us place a sphere into a box:

```
beditimg -v 7 -create 100,100,100 -origin 50,50,50 -sphere 35,35,38,12 -edge 3 -fill 1 spot1.mrc
```

The sphere is now located at voxel (35,35,38), with a radius of 12 voxels, and a soft edge of 3 voxels. The "-fill" option is required, to specify the extreme value within the sphere (otherwise the image will just be empty).

We can add another sphere:

```
beditimg -v 7 -sphere 21,40,38,2 -edge 1 -fill 1 spot1.mrc spot2.mrc
```

If the intended image has symmetry, we can apply that (Figure 7):

```
bsym -verb 7 -sym D5 spot2.mrc d5.mrc
```

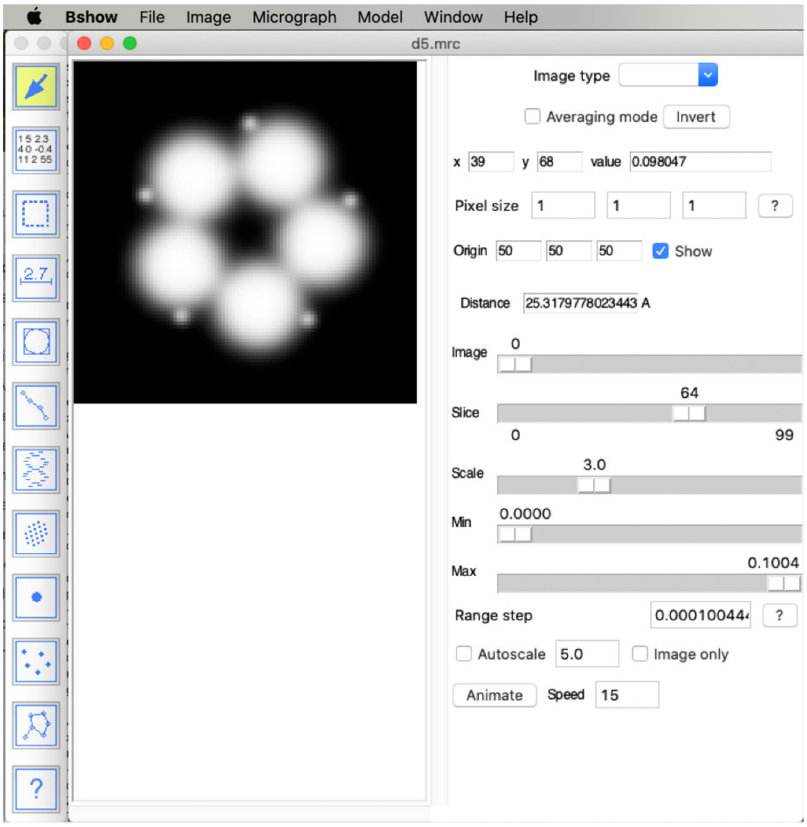


Image type

□ Averaging mode

x 39 y 68 value 0.098047

Pixel size 1 1 1

Origin 50 50 50 ☑ Show

Distance 25.3179778023443 A

Image 0

Slice 64 0 99

Scale 3.0

Min 0.0000

Max 0.1004

Range step 0.00010044

□ Autoscale 5.0 □ Image only

Speed 15

Figure 7. A synthetic map composed of two spheres with D5 symmetry imposed.

9. Images with random content and noise

Several types of random images can be generated.

The simplest is a uniform distribution (Figure 8A):

```
brandom -verb 7 -size 256,256,1 -type uni uniform.mrc
```

In cryoEM, the data mostly conforms to a Poisson distribution, also called shot or quantum noise (Figure 8B):

```
brandom -verb 7 -size 256,256,1 -type pois -avg 3 poisson.mrc
```

At larger averages, Poisson distributions approach Gaussian distirbutions.

To generate a Gaussian distribution, do (Figure 8C):

brandom -verb 7 -size 256,256,1 -type gaus -avg 9,3 gauss.mrc

A Gaussian distribution is also called white noise, because its spectral distribution approaches a constant power.

To generate a random image with a different spectral distribution, such as for brown noise, do (Figure 8E):

brandom -verb 7 -size 256,256,1 -type spec -alpha 2 brown.mrc

To generate pink noise (Figure 8D), set the alpha value to 1, and for blue noise (Figure 8F), set it to -1.

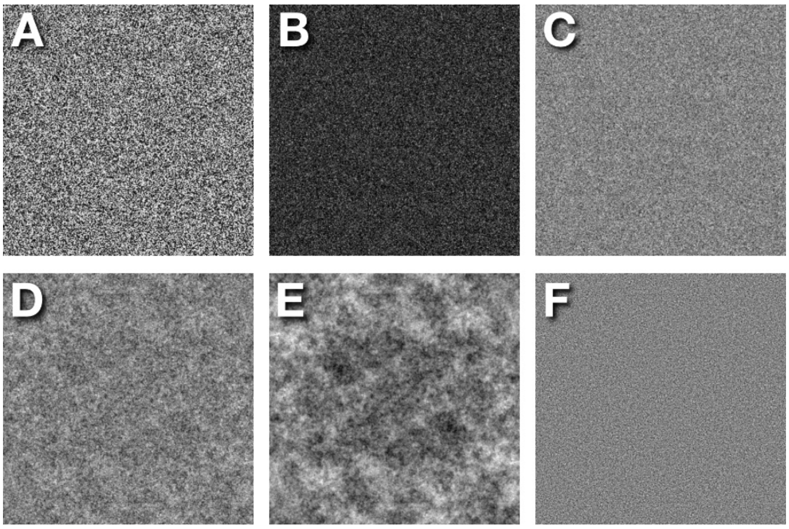


Figure 8. Random images.

(A) Uniform. (B) Poisson/Shot/Quantum noise. (C) Gaussian/White noise. (D) Pink noise. (E) Red/Brown noise. (F) Blue noise.

10. Point group symmetry

The symmetries of isolated biomolecules we deal with in cryoEM are mostly point group symmetries. The five classes of point group symmetries are shown in Figure 9A. Various symbols have been used for the symmetries, with the ones shown adopted in Bsoft.

As an example, the icosahedral map was created as follows:

beditimg -v 7 -create 100,100,100 -origin 50,50,50 -sphere 33,33,38,8 -edge 3 -fill 1 spot1.map
beditimg -v 7 -sphere 19,38,38,2 -edge 1 -fill 1 spot1.map spot2.map
bsym -verb 7 -sym I spot2.map ico.map

11. Helical symmetry

Helical filaments have their own symmetry. The fundamental symmetry of any helix can be expressed as a rise, and a rotation per subunit.

The first helix shown in Figure 9B was created as follows:

beditimg -v 7 -create 200,200,200 -ori 100,100,100 -sam 2 -sph 110,100,100,10 -fill 1 -edge 2 sph1.map
beditimg -v 7 -sph 120,110,100,2 -fill 1 -edge 2 sph1.map sph2.map
bhelix -v 7 -helix 40,66 -zlim 180,220 sph2l2.map hel2.map
bimg -v 7 -size 200,200,400 -trans 0,0,100 sph2.map sph2l2.map
bhelix -v 7 -helix 40,66 -zlim 180,220 sph2l2.map hel2.map

In addition, a helix can have further symmetries or features. The first is a dyad axis, which indicates that the two directions of the helix are 2-fold symmetry-related. The second is that it can have cyclic symmetry, typically with multiple protofilaments contributing to a composite helix. Figure 9B shows such a helix with five protofilaments on the right.

It was generated from the sph2l2.map created above, as follows:

bint -verb 7 -trans 10,0,0 -newori center sph2l2.map sph2l2t.mrc

```
bhelix -v 7 -helix 40,66,1,5 -zlim 180,220 sph2l2t.mrc hel2c5.mrc
```

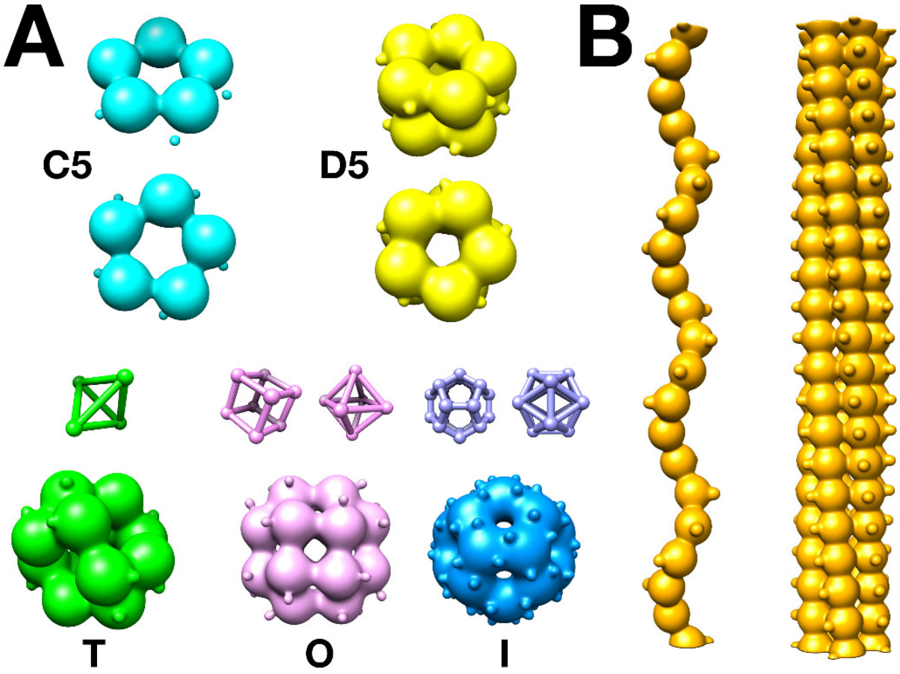


Figure 9. Synthetic maps with different symmetries.
(A) Point group symmetry: C5: 5-fold cyclic; D5: 5-fold dihedral; T: tetrahedral; O: octahedral; I: icosahedral. The models show the corresponding platonic solids: tetrahedron, cube, octahedron, dodecahedron, and icosahedron. (B) Helical symmetry: Left: H40,66 (rise of 40Å, rotation of 66°); Right: H40.66,1,5 (rise of 40Å, rotation of 66°, no dyad, cyclic 5-fold).

E. Fourier transforms and frequency space

1. Fourier transforms
   One of the most important operations in image processing is Fourier transformation. A Fourier transform of an image is a representation of the data in frequency space. This is analogous to representing an audio signal by its frequency spectrum. The transformation is linear, which means that we can backtransform it to recover the original image exactly.
   A forward transform is calculated with **bfft**:
   bfft -v 7 emd_0407_rs.mrc emd_0407_rs_ft.mrc
   The resultant image is complex, which requires an image format that can store complex values (such as MRC, Spider, Imagic, etc.).
   The backtransform should recover the original map:
   bfft -v 7 -inv emd_0407_rs_ft.mrc emd_0407_rs_bft.mrc
2. Power spectrum
   A power spectrum is the intensities of the Fourier transform (*i.e.*, without phases), and approximates the diffraction pattern obtained in the back focal plane of an electron microscope (technically, it approximates Fraunhofer diffraction). The power spectrum tells us a lot about the quality of an image, and how it was formed. In cryoEM, the resolution of a map is loosely defined as corresponding to the maximum spatial frequency with reliable data.

   A power spectrum can be calculated using **bfft** (with the -powerspectrum option), or after opening the image in **bshow**. In **bshow**, select the “Image/Power spectrum” menu item to bring up a dialog box (Figure 10). The “Logarithm” option calculates the logarithm of the power spectrum, to decrease the dynamic range, and make it more interpretable. Click on "OK" to calculate the power spectrum. Note that the "Image type" in the top right changes to "ps", to indicate it is a power spectrum. In this example, the power spectrum is clearly truncated. Hovering the mouse over the edge where it is truncated gives the corresponding resolution value in the "Distance" field on the right.

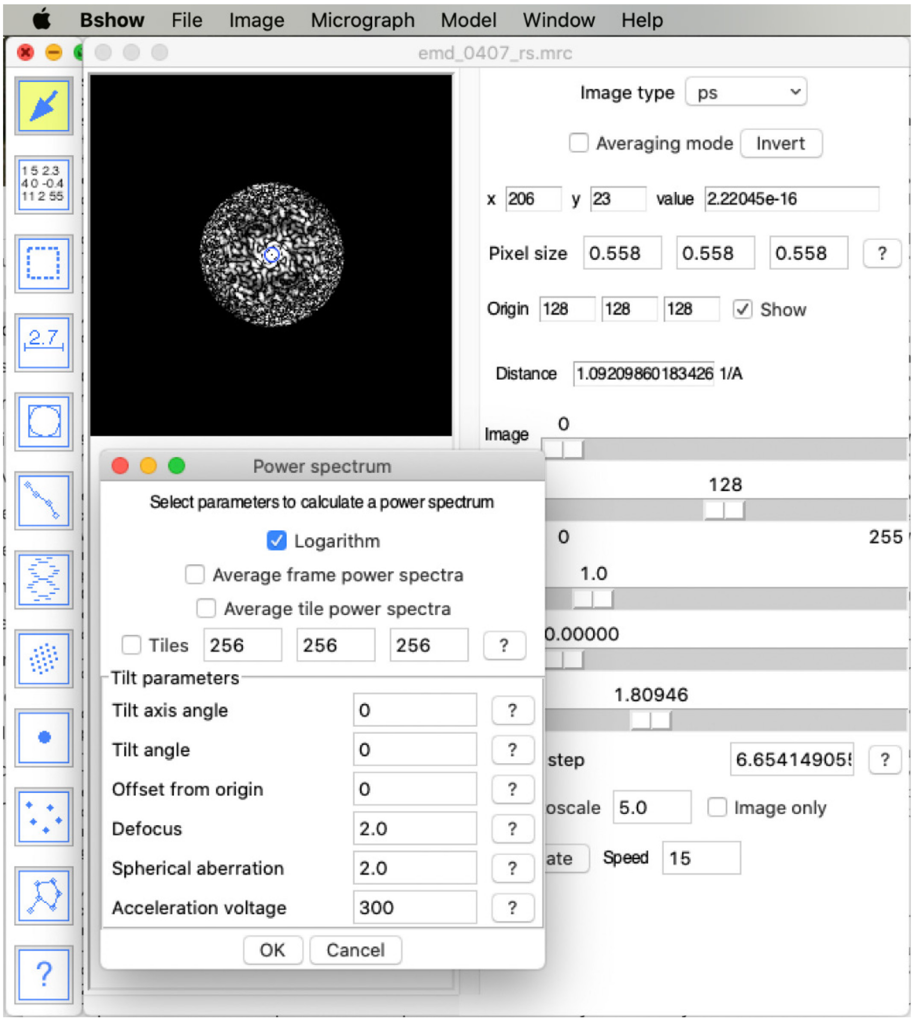


Image type ps
□ Averaging mode Invert
x 206 y 23 value 2.22045e-16
Pixel size 0.558 0.558 0.558 ?
Origin 128 128 128 ☑ Show
Distance 1.09209860183426 1/A
Image 0
128
0 255
1.0
0.00000
1.80946
step 6.65414905! ?
oscale 5.0 □ Image only
ate Speed 15

Power spectrum
Select parameters to calculate a power spectrum
☑ Logarithm
□ Average frame power spectra
□ Average tile power spectra
□ Tiles 256 256 256 ?
Tilt parameters
Tilt axis angle 0 ?
Tilt angle 0 ?
Offset from origin 0 ?
Defocus 2.0 ?
Spherical aberration 2.0 ?
Acceleration voltage 300 ?
OK Cancel

Figure 10. Power spectrum of map emd_0407 (logarithm).
The cutoff at 2.8 Å is the result of a low-pass filter that zeroed all the high frequency shells.

F. Image filters

Filters change the distribution of intensities in an image, and are typically used to better interpret image content. Care must be taken not to change the image so drastically that artifacts are produced. Filters can be applied either in real space or frequency space. In real space, this typically takes the form of a kernel of a specific size that is applied at every voxel (*i.e.*, a convolution). In frequency space, the filter has a Fourier transform, that is multiplied with the Fourier transform of the image.

1. Band-pass/Low-pass/High-pass filters
   Probably the most important filter in image processing is the band-pass filter. It selects a certain frequency range in the Fourier transform between two resolution limits, and sets everything else to zero. Two special

cases of the band-pass filter are the low-pass filter (everything below a high resolution limit), and the high-pass filter (everything above a low resolution limit).

Use **bfilter** to impose a band-pass filter (Figure 11B):

```
bfilter -verb 7 -band 5,20 emd_0407_rs.mrc emd_0407_rs_bp.mrc
```

The effect of the filter is to reduce the large-scale intensity variation across the image (set by the larger value), and to smooth or blur the local variation (set by the smaller value). The main intent of the latter is to remove the unwanted noise that dominates the high frequencies. Note that filtering results in oscillations or fringes in the background, which increase with larger values for the high resolution limit.

2. Averaging filter

The simplest and fastest real space kernel-based filter is the averaging filter. For every voxel, it sums all the values in the image within the size of the kernel and assigns the average to the central voxel of the new image. In **bshow**, select the "Image/Filter" menu item. Select the "Averaging" type, set the kernel size to 7, and click on filter (Figure 11C). The result is similar to the smoothing of the band-pass filter, but the background is now more even.

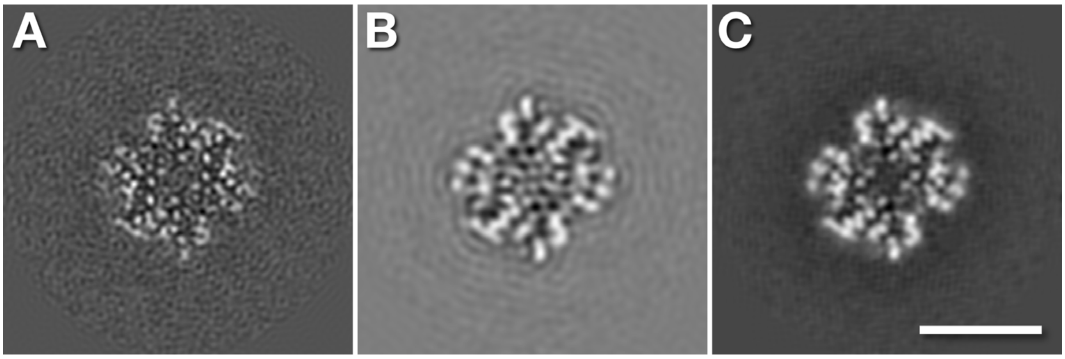


Figure 11. Filtering.
(A) Original map of emd_0407. (B) The map band-pass filtered with resolution limits of 5 Å and 20 Å. (C) The map filtered with a 7 x 7 x 7 voxel averaging kernel. Scale bar: 50 Å.

G. Cross-correlation

One of the most important operations in image processing is the cross-correlation of two images, used to align one image to another. This is fundamental to many parts of the processing of electron micrographs in cryoEM. It is typically done in frequency space, to speed up the calculation. In Bsoft, the use of frequency space allows the specification of resolution limits, to avoid the undue influence of noise on the result. This is effectively combining a band-pass filter with correlation analysis, to emphasize those frequencies with a high signal-to-noise ratio.

To illustrate it, we can shift the map by some arbitrary vector:

```
bint -verb 7 -trans -2,3.5,-1.5 emd_0407_rs.mrc emd_0407_t3.mrc
```

Then we cross-correlate it with the original, using conservative resolution limits

```
 bcc -verb 7 -resol 5,50 -Cross emd_0407_rs.mrc -Map emd_0407_cc.mrc emd_0407_t3.mrc
```

The output to the shell shows the resultant vector and correlation coefficient (CC):

```
 Image          dx             dy             dz             CC             P
 1              2.0010         -3.5865        1.4239         0.9828         0.9973
```

Note that the result vector is the negative of the applied vector, because it represents a shift back to the original position. Figure 12A shows the cross-correlation map, where the peak in the lower right corresponds to the determined shift vector. The peak in this case is very sharp indicating highly accurate alignment. To see the background in the map it can be rescaled with the min/max sliders in bshow (Figure 12B). The origin of this map can be placed in the center of the image with the "Image/Center origin" menu item (Figure 12C).

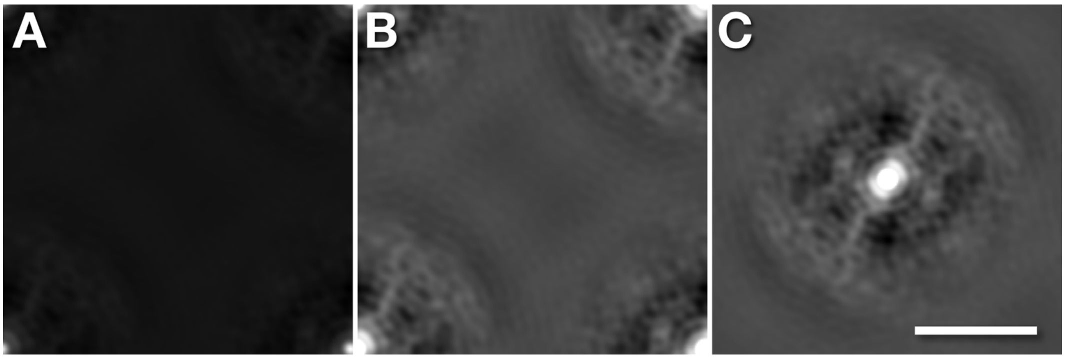


Figure 12. Cross-correlation maps.
(A) The cross-correlation map shows a peak in the lower right corner that corresponds to the offset vector. (B) A rescaled version of the cross-correlation map, to better show areas with low correlation. (C) The same map, but with the peak centered. Scale bar: 50 Å.

H. Masks

We use masks to indicate specific regions within an image, to eliminate parts we are not interested in, or to limit calculations. Masks can come in different forms depending on how we want to use them (Table 1).
To make a binary mask, we pick a threshold and generate it with:

```
bmask -verb 7 -threshold 2.2 emd_0407_rs.mrc emd_0407_mask.mrc
```

The binary mask in Figure 13A has a lot of background noise above the threshold.
To remove it, we can apply an opening operation (Figure 13B):

```
bmask -verb 7 -open 1 emd_0407_mask.mrc emd_0407_mask_o1.mrc
```

The basic operations on binary masks are dilation and erosion. Dilation adds voxels to the edges of a mask to enlarge it, while erosion deletes voxels at the edges to reduce it. Opening operations apply one or more iterations of erosion, followed by dilation. Closing operations will apply one or more iterations of dilation, followed by erosion. The intent of these operations is to better define the extent of a mask, so we can more appropriately cover the structure in a map, and eliminate noise.

In some cases, we want to indicate different regions with different values within a mask. A simple way to generate multi-level masks is to specify thresholds (Figure 13C):

```
bmultimask -verb 7 -threshold -4,-2,0,2,4 emd_0407_rs.mrc emd_0407_mlm.mrc
```

Multi-level masks are very useful where we want to segment a map, to indicate different parts. Although Bsoft has several tools for segmentation, this is a very complicated topic and it will not be covered here.

Table 1. Masks

Mask type	Range of values	Typical intended use
Binary	0 and 1	Foreground-background (e.g., part of a structure)
Multi-level	Integers: 0…	Segmentation result, mutually exclusive regions
Bit-level	Integers: 0…	Segmentation result, overlapping regions
Fuzzy	0-1	Applying a mask with soft edges

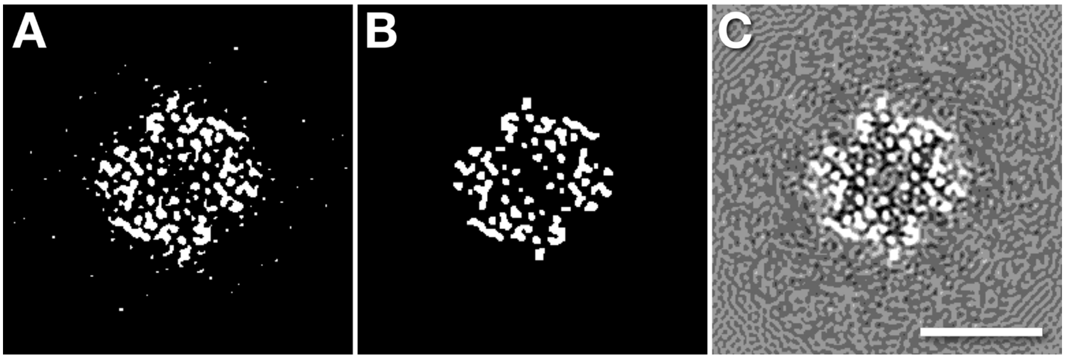


Figure 13. Masks.
(A) A binary mask generated from emd_0407. (B) The same mask, but with an additional opening operation to eliminate noise. (C) A multi-level mask. Scale bar: 50 Å.

I. Radial functions

1. Real space radial profile
   For many particles with a spherical shape (such as virus capsids), the radial profile is instructive, to demarcate the layers or shells of the particle.
   This is calculated with an output in Postscript format (note that the origin and sampling must be correctly specified in the map header):
   bradial -verb 7 -radial -Post emd_22333_rad.ps emd_22333.map
   Figure 14A shows the radial profile for the "mottled" capsid of the Salmonella phage, SPN3US (Heymann *et al.*, 2020). The three peaks on the right side correspond to structured protein shells, while the high density on the left indicates a less structured mass in the center of the capsid.

2. Radial power spectrum
   The radial power spectrum of an image or map gives the strength or power at different frequencies or resolution shells, indicating how much detail is in the image:
   bradial -verb 7 -Radialps 7 -log -Post emd_22333_rps.ps emd_22333.map
   Here, the power spectrum is calculated to 7 Å, because there is not much information beyond that. The power is also converted to its logarithm, to better display the high dynamic range (Figure 14B). The Postscript file is a text file with a table containing the data. This can be read in a graphing program (here, I used Kaleidagraph), to replot it with different axes (Figure 14B inset).

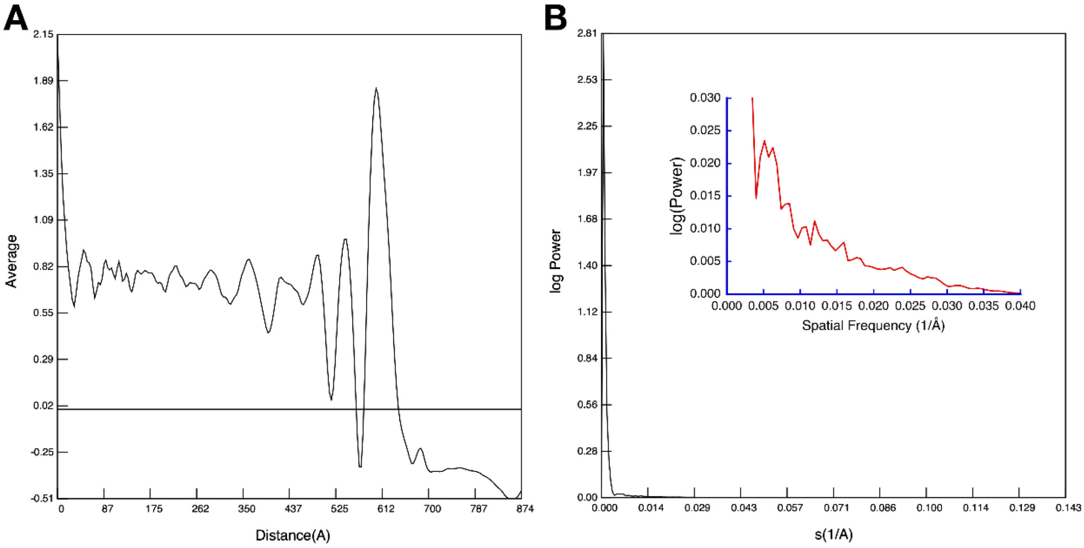


Figure 14. Radial profiles of map emd_22333.
(A) Radial average profile in real space, showing the capsid shell as the highest peak, with other peaks and density inside. (B) Radial power spectrum (logarithm) with a high dynamic range. The data in the text Postscript file can be extracted and replotted with different ranges (inset).

3. Radial sections
   While the radial profiles provide information, radial sections aim to present the lateral information at each radius.
   For simple spherical shells, the operation is straightforward:
   bradial -verb 7 -shells emd_22333.map emd_22333_sh.mrc
   Figure 15A shows two radial sections of the SPN3US capsid (Heymann *et al.*, 2020). Because the capsid is not exactly spherical, but icosahedral, the vertices look different from the rest.
   To compensate for the icosahedral shape, a more complicated calculation can be done:
   bradsec -verb 7 -sym I-3 -frac 0.3 emd_22333.map emd_22333_radsec.mrc
   Here, the symmetry is specified to indicate that the 3-fold faces of the icosahedron are somewhat flattened, with the fraction of sphericity at 0.3 (Figure 15B).

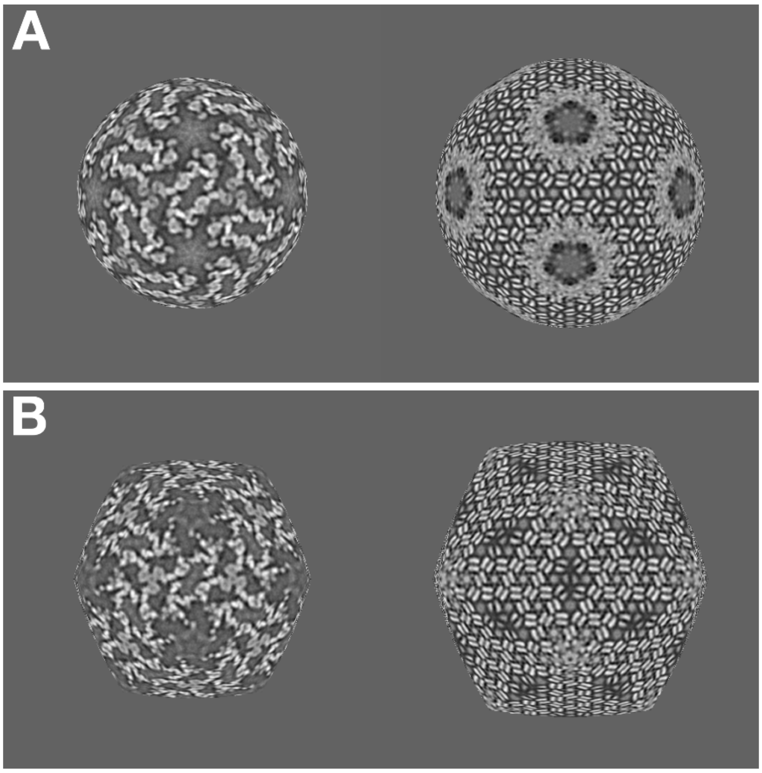
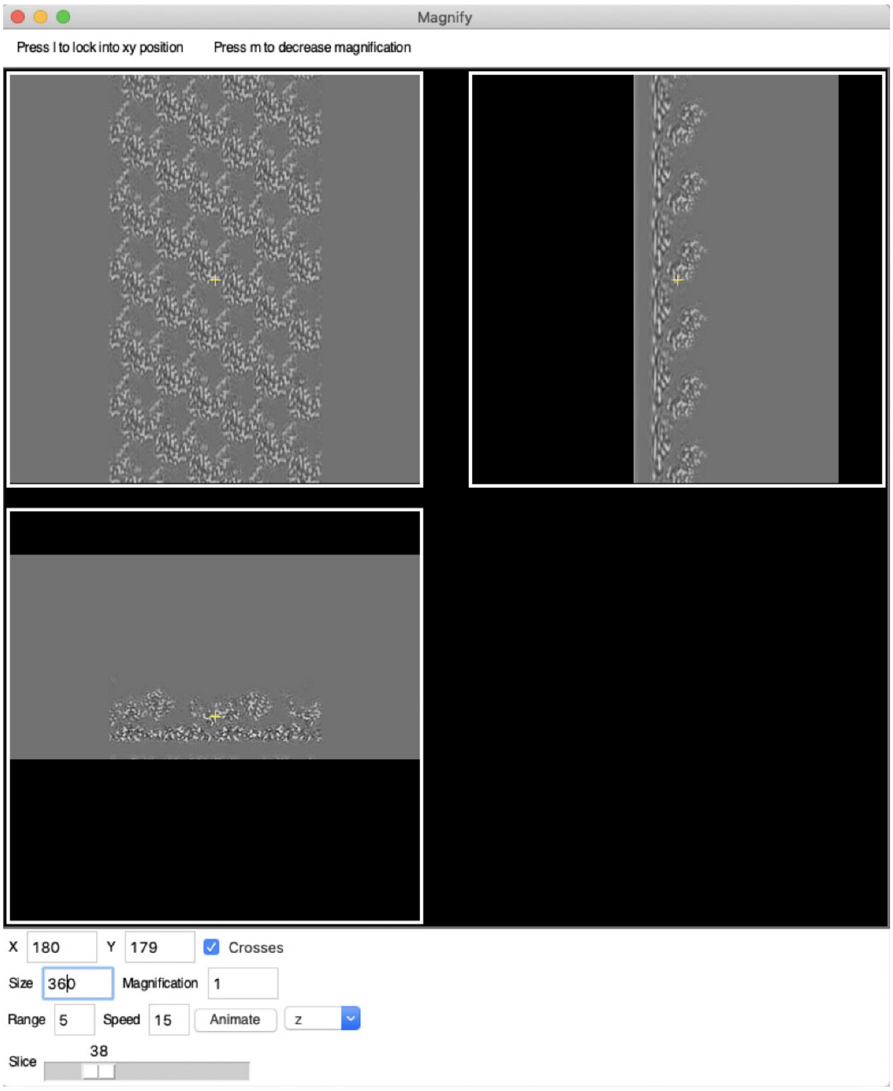


Figure 15. Radial sections of map emd_22333.
(A) Isotropic radial sections at 537 Å and 625 Å. (B) Symmetry-adjusted radial sections for icosahedral symmetry, excluding the 3-fold axes at similar radii as shown in (A).

4. Cylindrical unwrapping
   With filamentous particles, the different layers around the cylinder can be better displayed when it is unwrapped:
   bcyl -verb 7 -unwrap 1 -reslice z-yx emd_4802.map emd_4802_uw.mrc
   The reslicing option switches the axes, and can be used to obtain the desired display direction after unwrapping. Figure 21 shows three orthogonal views of the unwrapped map of the antifeeding prophage

of *Serratia entomophila* (Desfosses *et al.*, 2019). The top left panel gives a nice impression of the helical repeat pattern.



Magnify

Press l to lock into xy position   Press m to decrease magnification

X 180   Y 179   ☑ Crosses

Size 360   Magnification 1

Range 5   Speed 15   Animate   z

Slice 38

Figure 16. Unwrapping a helical filament, map emd_4802.
Three orthogonal sections, as viewed with the "Window/Magnify" menu item in bshow.

J. Projections

While the maps in structural biology are 3D, the original data is 2D. We therefore want to be able compare the projections from 3D maps with the 2D images.

1. Real space projections
   The simplest approach is to calculate the average of the densities in the 3D map, in a direction for each projection. Typically, we are interested in a set of unique views within the asymmetric unit of a map. The projection directions are chosen to form a regular distribution with given step size in degrees:
   bproject -verb 7 -angles 30 -sym C2 emd_0407_rs.mrc emd_0407_rs_proj.mrc
   Figure 17A shows a set of projections generated in real space limited to the 2-fold asymmetric unit of the particle.

2. Frequency space projection

A better way to generate projections is to calculate them in frequency space using the central section theorem (Figure 17B):

bproject -verb 7 -kernel 8,2 -angles 30 -sym C2 emd_0407_rs.mrc emd_0407_rs_proj_fs.mrc

The kernel is for frequency space interpolation (Lanzavecchia and Bellon, 1995).

We can compare the two methods:

bop -verb 7 -fit 0 emd_0407_rs_proj.mrc emd_0407_rs_proj_fs.mrc emd_0407_rs_proj_dif.mrc

While the real space projection produces mostly good projections, there are some with artifacts resulting from a Moire pattern, between the original and final samplings (*e.g.*, projection 28 in Figure 17C).

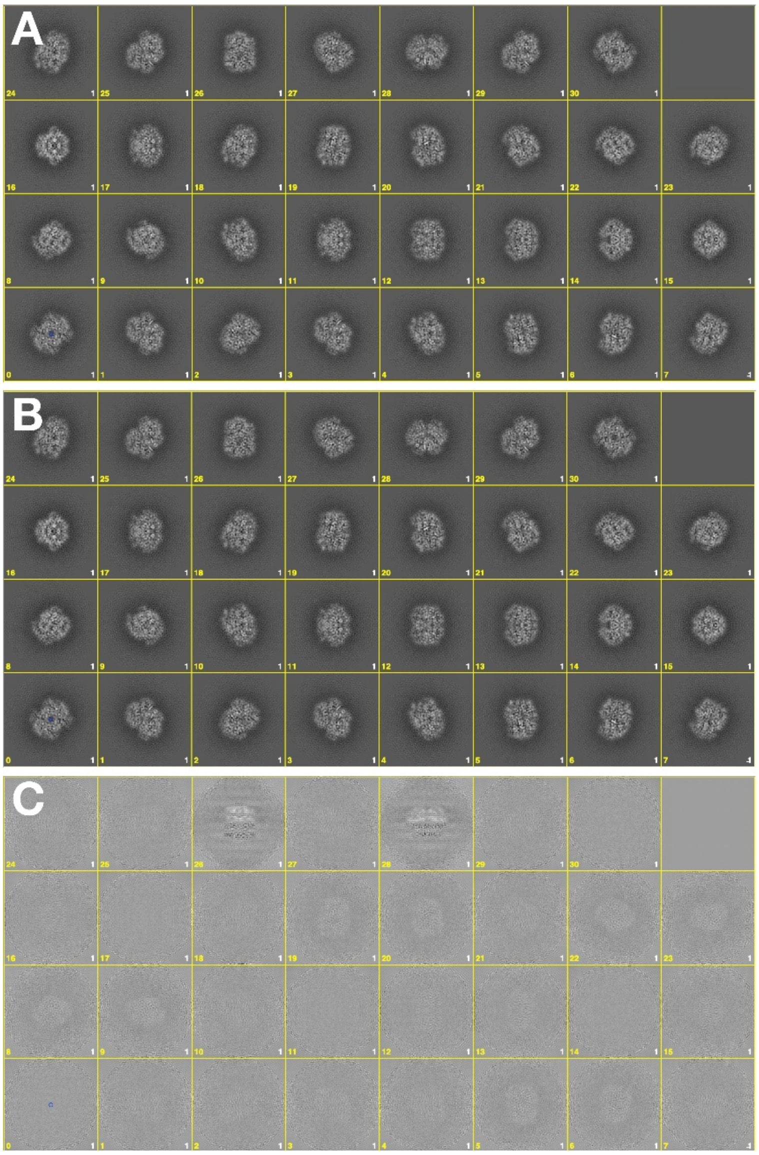


Figure 17. Projections of map emd_0407.

(A) Real space projections. (B) Frequency space projections. (C) Difference images between the real and frequency space projections show significant artifacts that may affect processing such as used in cryoEM.

K. Other capabilities in Bsoft

The main purpose behind Bsoft is to provide tools for structural biology, as done with electron microscopy. The two main areas are (i) the high resolution structure determination of biomolecules by single particle analysis and (ii) electron tomography, to visualize biocomplexes and cellular substructure. For the first, please see a general description of single particle analysis (Heymann, 2018a), a way to validate results (Heymann, 2015), a demonstration of validate reconstruction (Heymann, 2018b), and detail protocols for a specific case (Heymann, 2020). For tomography, please see the original paper (Heymann *et al.*, 2008), and the current state in Bsoft (Heymann, 2021). The future vision for Bsoft is to continue development for single particle analysis in cryoEM, and to integrate the molecular details within the context of cells, using electron tomography.

Notes

1. A common problem after installing Bsoft is the path to the executables (programs) is not properly formed, resulting in an inability to find them. These are typically set in the /etc/profile, or /etc/csh.cshrc files, depending on the operating system and shell used.
   In them, they should refer to the bsoft.bashrc or bsoft.cshrc files, *e.g.*,
   source /usr/local/bsoft/bsoft.bashrc
2. The interactive programs in Bsoft, **bshow** and **brun**, depend on the system Tcl/Tk installation. If any other software package has been installed that redirects to a different version of Tcl/Tk, it could prevent the Bsoft programs from running. The only remedy is to reset the links to the system Tcl/Tk. This in turn depends on how the system is configured, and is beyond the scope here.
3. The recognition of image formats depends on the extension of the file name, such as .mrc or .tiff (*i.e.*, the characters after the last period).
4. Many image formats do not support a wide range of data types. Therefore, it is important to understand at least some of the details about image formats. For instance, the TIFF specification includes a wide range of data types that are supported in Bsoft. However, many other programs can only read TIFF files with data types of byte or short. A file in floating point format can be converted to byte with many Bsoft programs, but it must be understood that this rescales the values in the image to fit into bytes. It may have consequences that degrade the image considerably. To prevent this, use the -rescale and -truncate options in **bimg,** to convert an image into an acceptable form.
5. The interpretation of images, especially in cryoEM, depends on the correct specification of the pixel or voxel size. This is often absent, or not correctly specified in images derived from other software packages. In Bsoft programs, set it correctly using the -sampling option. It can take three values to indicate anisotropic sampling.
6. Any geometric operations depend on the correct specification of the origin. In Bsoft programs, set it with the -origin option. An argument of "center" puts the origin in the center of the image.
7. Comparing any two or more images requires that they are the same size and scaling. They must also have the same intensity distribution, such that white corresponds to white, and black to black.

Acknowledgments

This work was supported by the Intramural Research Program of the National Institute for Arthritis, Musculoskeletal and Skin Diseases, NIH.

Competing interests

The author has no competing interests.

References

Carazo, J. M., Stelzer, E. H., Engel, A., Fita, I., Henn, C., Machtynger, J., McNeil, P., Shotton, D. M., Chagoyen, M., de Alarcon, P. A., *et al*. (1999). Organising multi-dimensional biological image information: the BioImage Database. *Nucleic Acids Res* 27(1): 280-283.

Cheng, A., Henderson, R., Mastronarde, D., Ludtke, S. J., Schoenmakers, R. H., Short, J., Marabini, R., Dallakyan, S., Agard, D. and Winn, M. (2015). MRC2014: Extensions to the MRC format header for electron cryo-microscopy and tomography. *J Struct Biol* 192(2): 146-150.

Desfosses, A., Venugopal, H., Joshi, T., Felix, J., Jessop, M., Jeong, H., Hyun, J., Heymann, J. B., Hurst, M. R. H., *et al*. (2019). Atomic structures of an entire contractile injection system in both the extended and contracted states. *Nat Microbiol* 4(11): 1885-1894.

Herzik, M. A., Jr., Wu, M. and Lander, G. C. (2019). High-resolution structure determination of sub-100 kDa complexes using conventional cryo-EM. *Nat Commun* 10(1): 1032.

Heymann, J. B. (2001). Bsoft: image and molecular processing in electron microscopy. *J Struct Biol* 133(2-3): 156-169.

Heymann, J. B. (2015). Validation of 3D EM Reconstructions: The Phantom in the Noise. *AIMS Biophys* 2(1): 21-35.

Heymann, J. B. (2018a). Guidelines for using Bsoft for high resolution reconstruction and validation of biomolecular structures from electron micrographs. *Protein Sci* 27(1): 159-171.

Heymann, J. B. (2018b). Single particle reconstruction and validation using Bsoft for the map challenge. *J Struct Biol* 204(1): 90-95.

Heymann, J. B. (2020). Protocols for Processing and Interpreting cryoEM Data Using Bsoft: A Case Study of the Retinal Adhesion Protein, Retinoschisin. *Bio-protocol* 10(2): e3491.

Heymann, J. B. (2021). High resolution electron tomography and segmentation-by-modeling interpretation in Bsoft. *Protein Sci* 30(1): 44-59.

Heymann, J. B. and Belnap, D. M. (2007). Bsoft: image processing and molecular modeling for electron microscopy. *J Struct Biol* 157(1): 3-18.

Heymann, J. B., Cardone, G., Winkler, D. C. and Steven, A. C. (2008). Computational resources for cryo-electron tomography in Bsoft. *J Struct Biol* 161(3): 232-242.

Heymann, J. B., Wang, B., Newcomb, W. W., Wu, W., Winkler, D. C., Cheng, N., Reilly, E. R., Hsia, R. C., Thomas, J. A. and Steven, A. C. (2020). The Mottled Capsid of the Salmonella Giant Phage SPN3US, a Likely Maturation Intermediate with a Novel Internal Shell. *Viruses* 12(9): 910.

Lanzavecchia, S. and Bellon, P. L. (1995). A Bevy of Novel Interpolating Kernels for the Shannon Reconstruction of High-Bandpass Images. *J Vis Commun Image Represent* 6(2): 122-131.

Pettersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C. and Ferrin, T. E. (2004). UCSF Chimera--a visualization system for exploratory research and analysis. *J Comput Chem* 25(13): 1605-1612.