Real Data Practical

(Borrelia) flagellar motors
138 particles, already cropped.
Let's consider the data in

/scratch/dynamo/data/flagellarMotorParticles

those are flagellar motors subtomograms, already cropped.

Exercise
Produce this image using
ddbrowse

You might want to use
dynamo_table_blank
The result of the alignment should be something like this
The objective of this tutorial is to show a way to create a first set of initial orientations that can be later used to feed an alignment project.

One option could be to start on a totally blind manner, creating a random average without taking into account orientations, and letting Dynamo carry several iterations on full search modus....
... it might work, but on a very inefficient way: convergence is very slow!

Another option would be to provide the particles with an initial estimate of the orientations.
This can be performed with manual alignment, but this is probably messy, and will not work for all the particles in the data set.
Suggestion:

Create a template that captures the geometry of the expected average.

Dynamo includes several prepared geometries. Let's try with `dpktomo.examples.motiveTypes.MembraneWithRod`.

```python
>> a = dpktomo.examples.motiveTypes.MembraneWithRod();
>> a

a =

MembraneWithRod with properties:

    rodRadius: 8
    rodHeight: 20
    rodShift: [0 0 0]
    thickness: 6
    radius: 200
    shifts: [0 0 0]
    eulers: [0 0 0]
    sidelength: 64
    position: []
    data: []
    mask: []
```

It creates an object with a series of attributes. We need to adapt some of them to our situation.
>> a sidelength = 128;
>> a thickness = 10;
>> a fillData();
>> template = a data;
>> dview(template);

changes the thickness of the membrane

we do get a geometrical shape looking like a membrane with an additional mass...

.... but how useful is this actually in order to align the particles?
Suggestion:

Before creating full projects, sometimes it is useful to check how good a set of parameters is on an isolated particle!

```matlab
% read an arbitrary data particle
>> datafolder = '/scratch/dynamo/data/flagellarMotorParticles/borreliaParticles';
>> myTag = 4;
>> particle = dynamo_tag2particle(myTag, datafolder);
>> dview(particle);
```

a membrane can be recognized
So we choose a set of parameters that make sense

>> coneRange = 360;
>> coneSampling = 40;
>> inplaneRange = 0;
>> inplaneSampling = 1;
>> refine = 6;
>> refineFactor = 2;
>> sizeBinned = 64;

This selection of parameters tries to design a global search, but saves computation time by:
1) using a large interval on the first multigrid level
2) relaying heavily on the local refinements
3) using binned particles (original particles have size 128)
4) do not using any azimuthal rotation, as it is not needed just to find orientations of highly axially symmetric objects.

... so, this alignment of one particle should run very fast!
sal = dynamo_align(particle,template,...
'cr',coneRange,...
'cs',coneSampling,...
'ir',inplaneRange,...
'is',inplaneSampling,...
'rf',refine,...
'rff',refineFactor,...
'dim',sizeBinned);

“....” means that it should be written in the same line.
‘cr’,cs’, etc are flags. they are described in doc align

the output of this alignment is redirected to a variable arbitrarily close “sal”

To see what is inside just type

   >> sal

without any semicolon to end the line
It includes some volumes, that express how the aligned template matches the data (or vice versa)
As a side note remember that you can always use the GUI version of the dalign command, which pops up when invoked without arguments.
so... let’s see if our selection of parameters produced a sensible alignment:
We show slice by slice how the align particle relates to the template

```dmapview({sal.aligned_particle,sal.template});```
out of the box, it looks bad... can we actually get a reasonable comparison?
First normalize equally both intensities
Then select to visualize a single slice ....

(*') note that you can select a single slice by pressing [c] after pointing with the cursor at the region of interest

... but still it does not look good
.... unless you choose to show the result of averaging along the same direction
Those were would news!

The template is powerful enough and the set of scanned angles “exhaustive” enough to drive the signal to find a real orientation in a short time.

It means that you can now create a small project that uses the same angular settings that this alignment command on all the particles!

Exercise

Create and run this project!
Hint: use some shift limiting policy
Let's take a look on how the average looks like:

```bash
>> ddb coarseBorrelia:a -v
```

looks like we recover the expected features: a membrane with a rod.

The important thing is NOT per se a membrane+rod volume but the fact that this was computed with the data, probing so that the table computed by the project is correct and can thus be used as initial table for a more serious alignment project.

EXERCISE
Compute manually the average, using ddb [item rt] and the command daverage