Real Data Practical

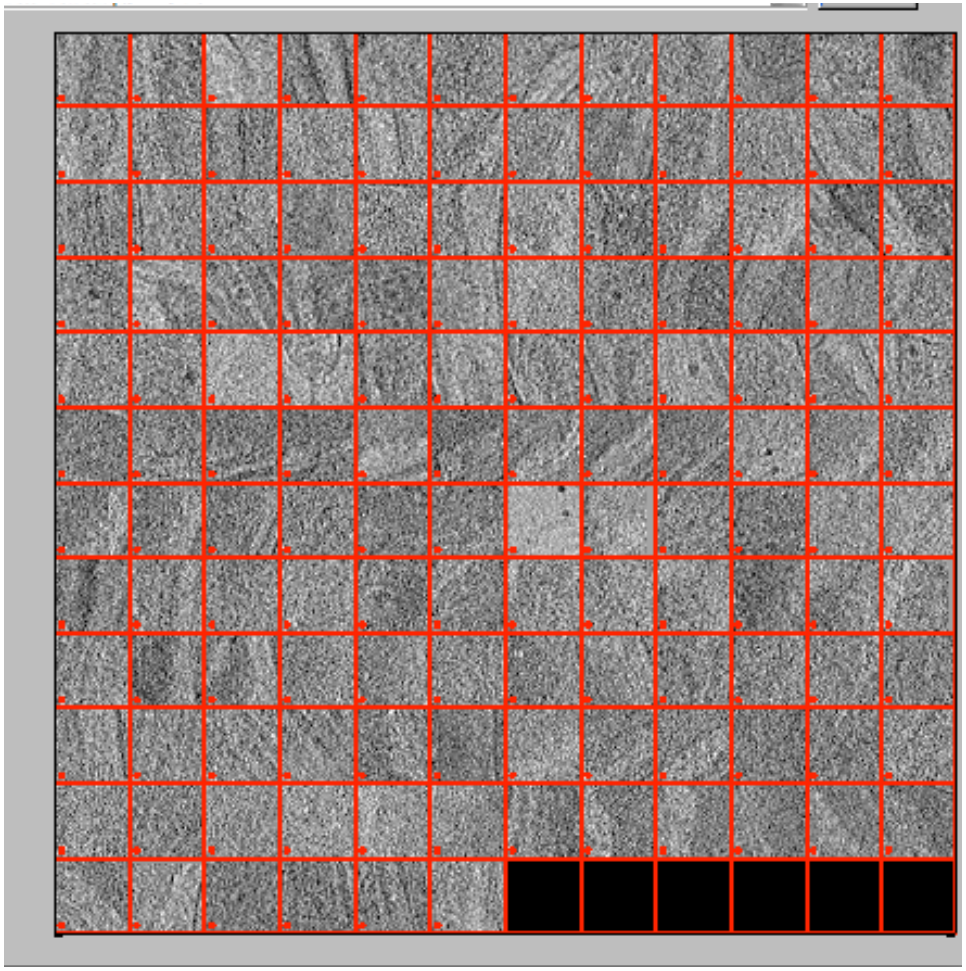
(Borrelia) flagelar motors

138 particles, already cropped.

Lets considere the data in

`/scratch/dynamo/data/flagelarMotorParticles`

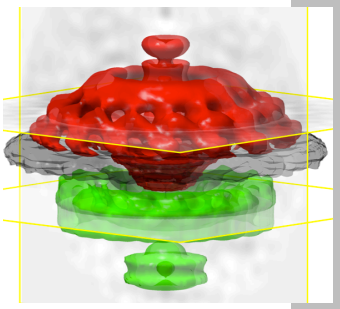
those are flagelar motors subtomograms, already cropped.

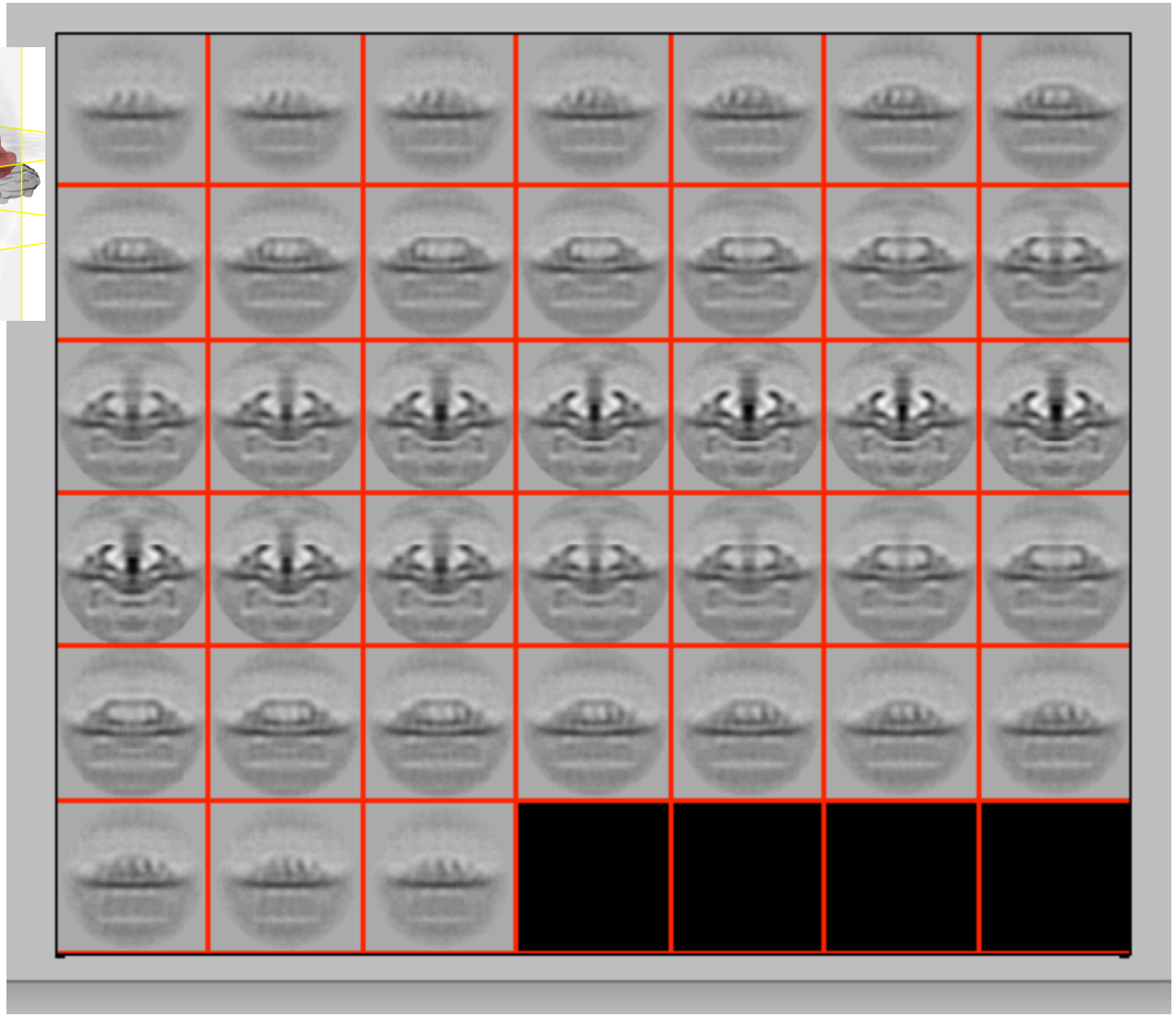


Exercise
Produce this image using
`ddbrowse`

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

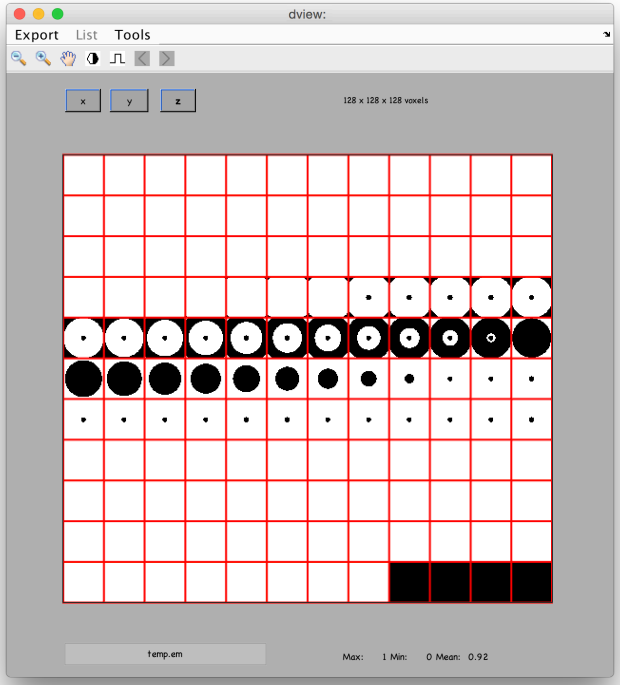
Dynamo includes several prepared geometries.
Lets try with `dpktomo.examples.motiveTypes.MembraneWithRod`

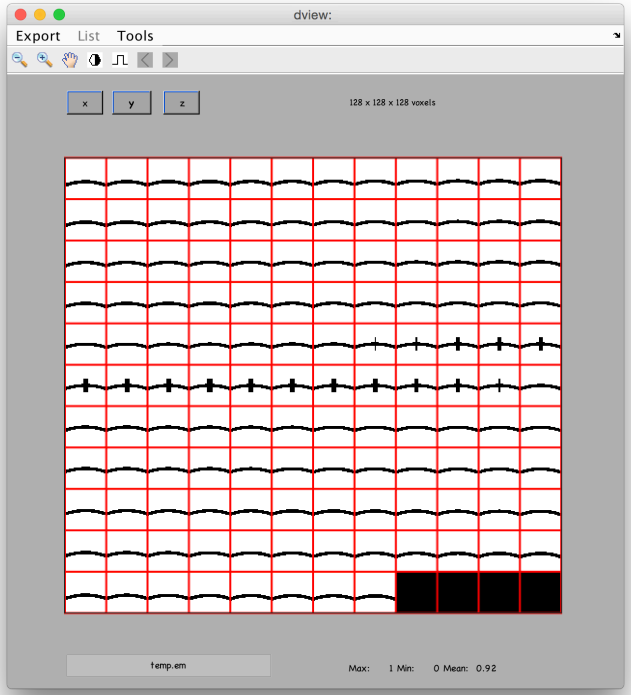
```
>>
>> a = dpktomo.examples.motiveTypes.MembraneWithRod();
>> a

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 a 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…

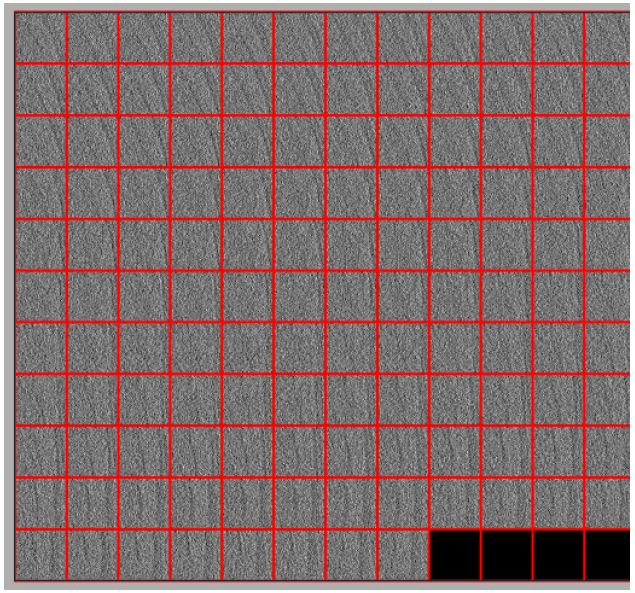
…. but how useful is this actually in order to align the particles?

Suggestion:

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

```
% 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 azymuthal 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.

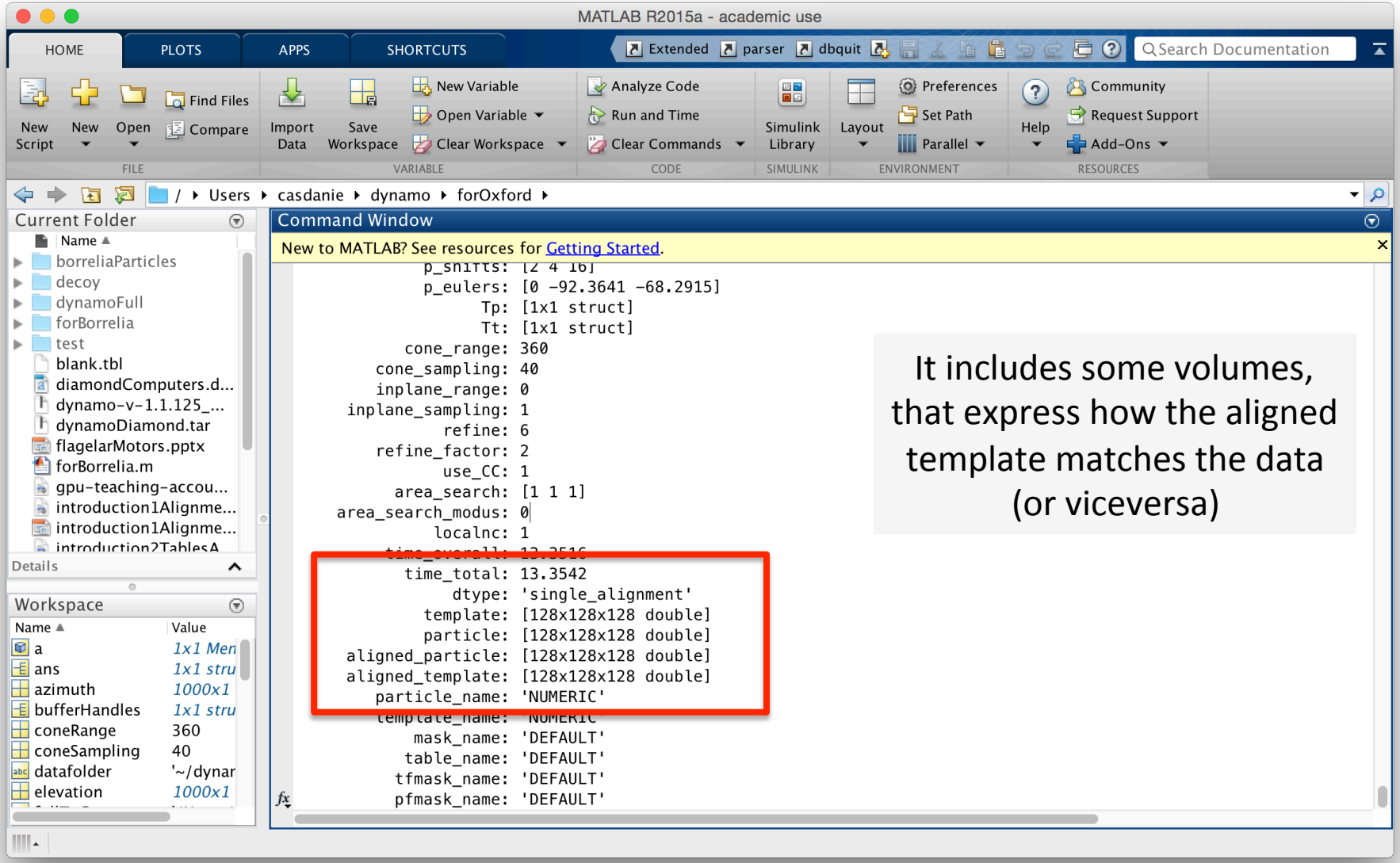
'cr',cs', etc are flags. they are described in doc align

the output of this alignment is redirected to a variable arbitrarily close "`sal`"

To see what is inside just type

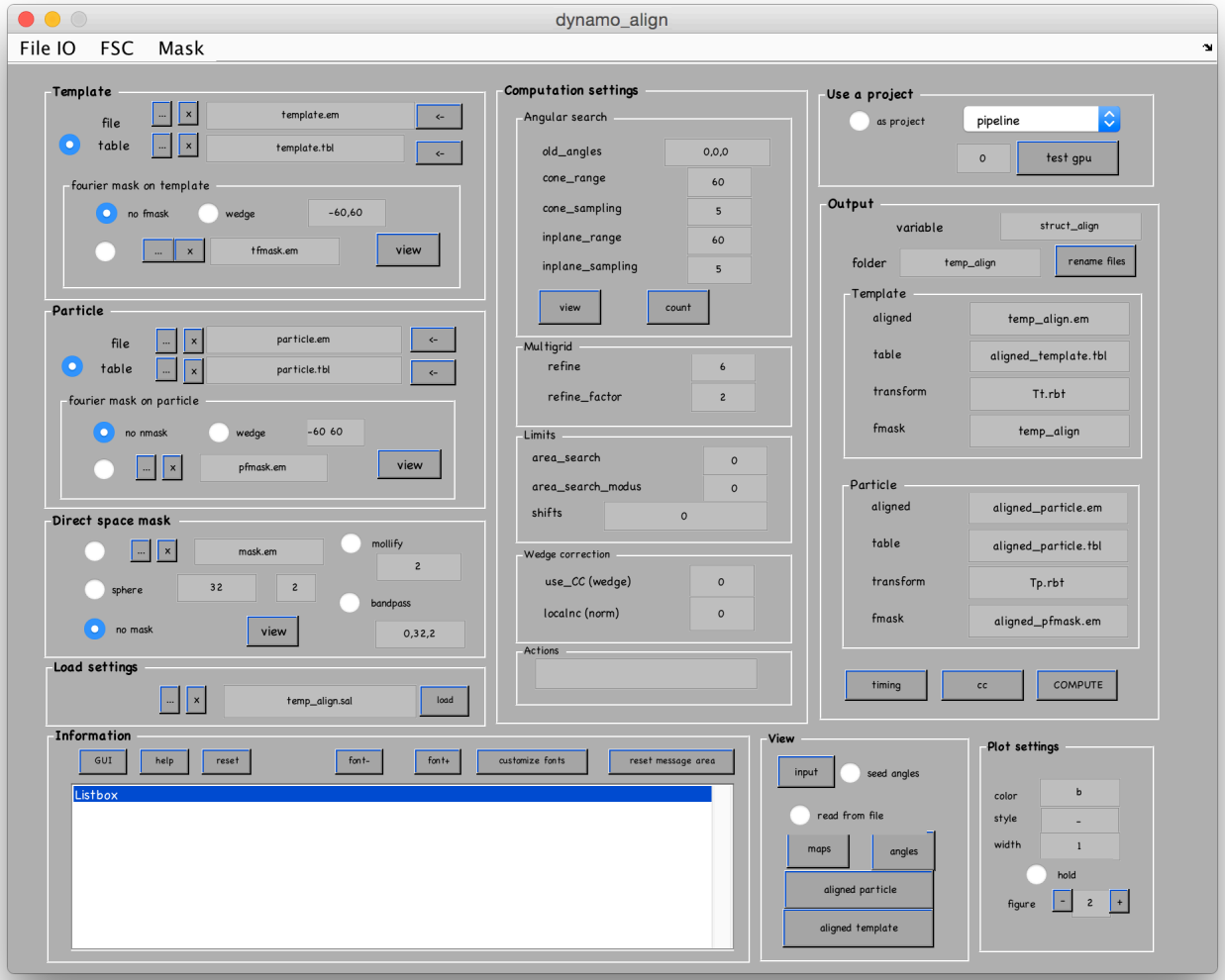
```
>> sal
```

without any semicolon to end the line


MATLAB R2015a - academic use
HOME
PLOTS
APPS
SHORTCUTS
Extended
parser
dbquit
Search Documentation
New Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Analyze Code
Run and Time
Clear Commands
Simulink Library
Layout
Preferences
Set Path
Parallel
Help
Community
Request Support
Add-Ons
FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
RESOURCES
/ ▸ Users ▸ casdanie ▸ dynamo ▸ forOxford ▸
Current Folder
Name
borreliaParticles
decoy
dynamoFull
forBorrelia
test
blank.tbl
diamondComputers.d...
dynamo-v-1.1.125_...
dynamoDiamond.tar
flagelarMotors.pptx
forBorrelia.m
gpu-teaching-accou...
introduction1Alignme...
introduction1Alignme...
Details
Workspace
Name
Value
a
ans
azimuth
bufferHandles
coneRange
coneSampling
datafolder
elevation
Command Window
New to MATLAB? See resources for Getting Started.
p_shifts: [2 4 16]
p_eulers: [0 -92.3641 -68.2915]
Tp: [1x1 struct]
Tt: [1x1 struct]
cone_range: 360
cone_sampling: 40
inplane_range: 0
inplane_sampling: 1
refine: 6
refine_factor: 2
use_CC: 1
area_search: [1 1 1]
area_search_modus: 0
localnc: 1
time_total: 13.3542
dtype: 'single_alignment'
template: [128x128x128 double]
particle: [128x128x128 double]
aligned_particle: [128x128x128 double]
aligned_template: [128x128x128 double]
particle_name: 'NUMERIC'
mask_name: 'DEFAULT'
table_name: 'DEFAULT'
tfmask_name: 'DEFAULT'
pfmask_name: 'DEFAULT'
It includes some volumes, that express how the aligned template matches the data (or viceversa)

As a side note remember that you can always use the GUI version of the `dalign` command, which pops up when invoked without arguments.

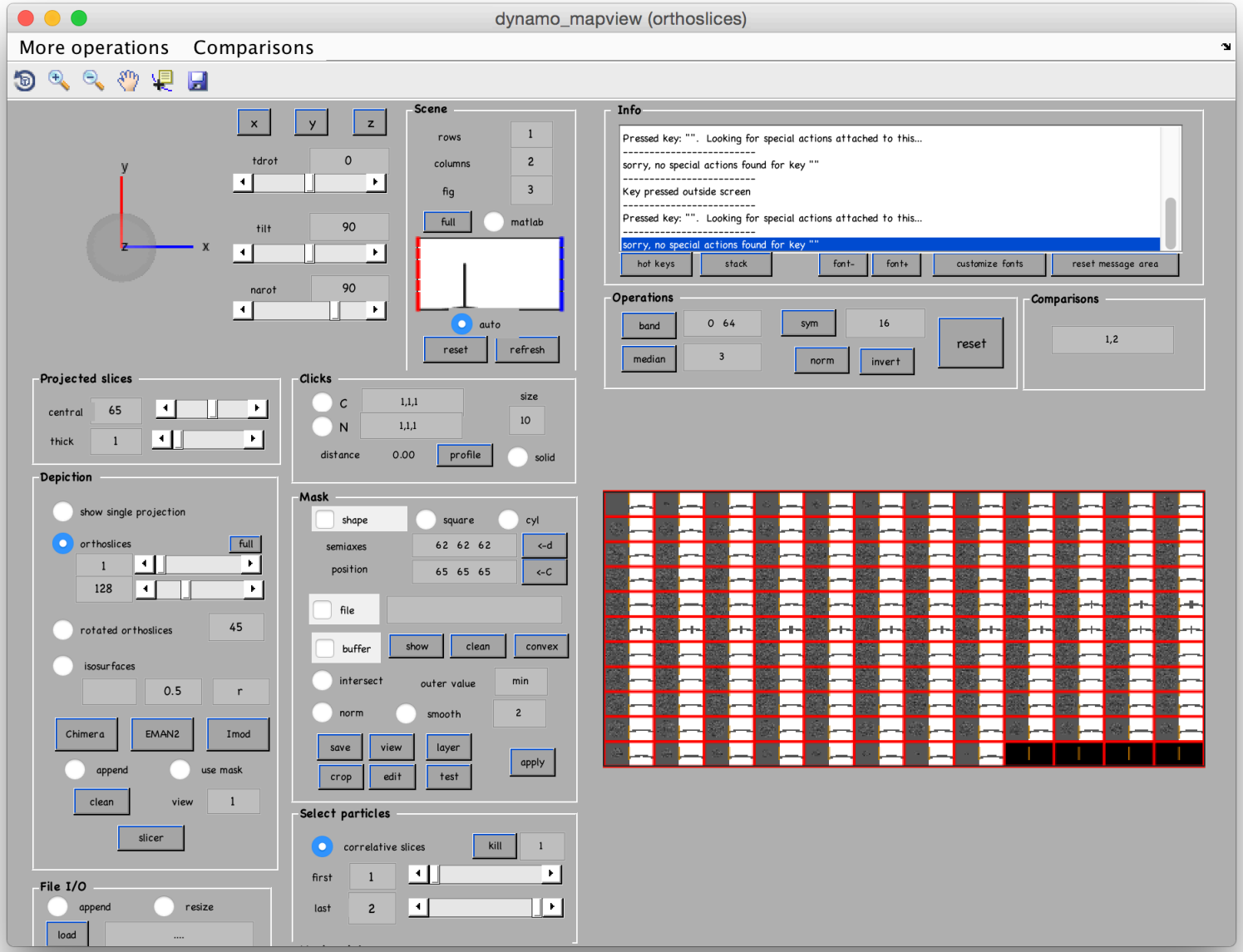
```
>> dalign
```

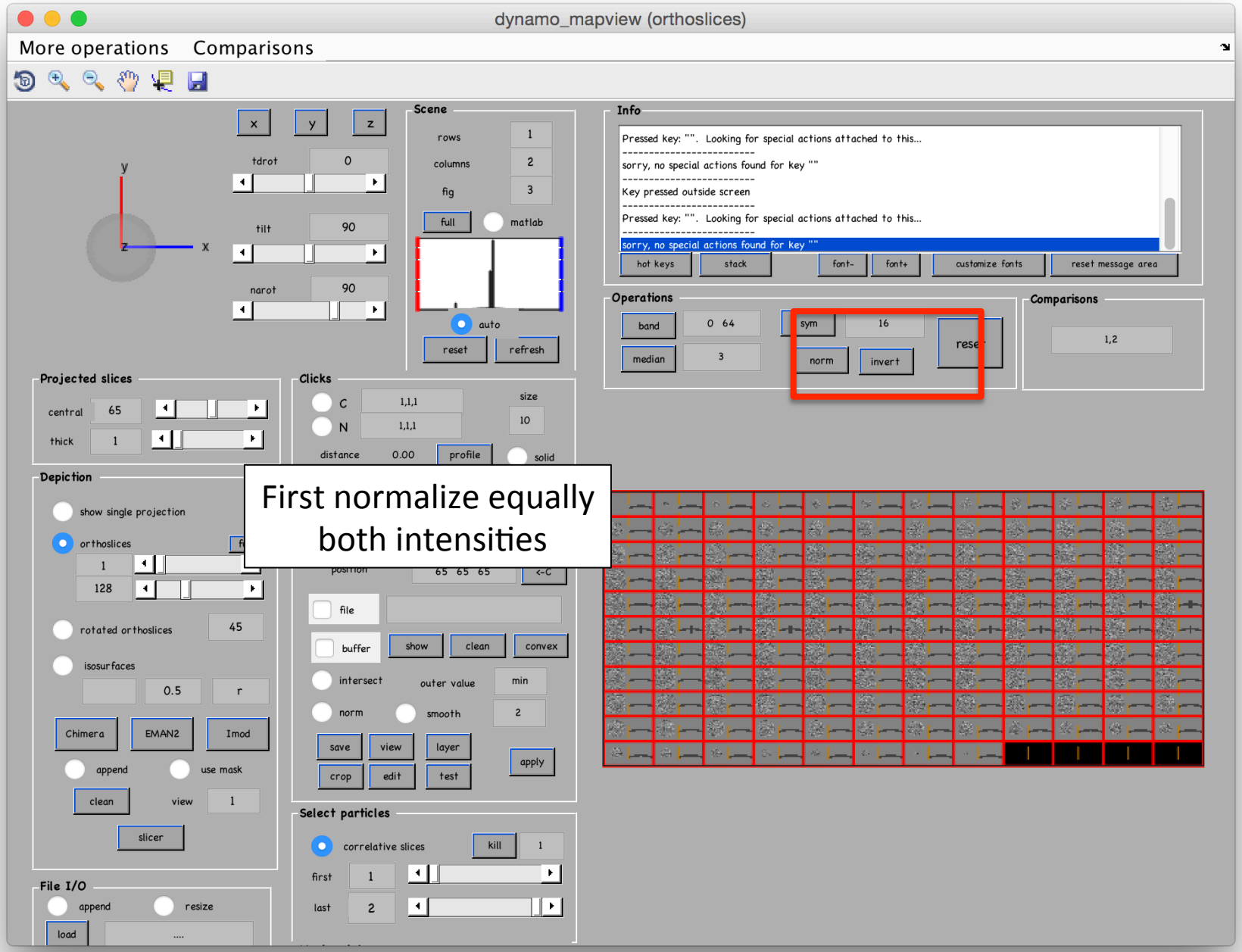


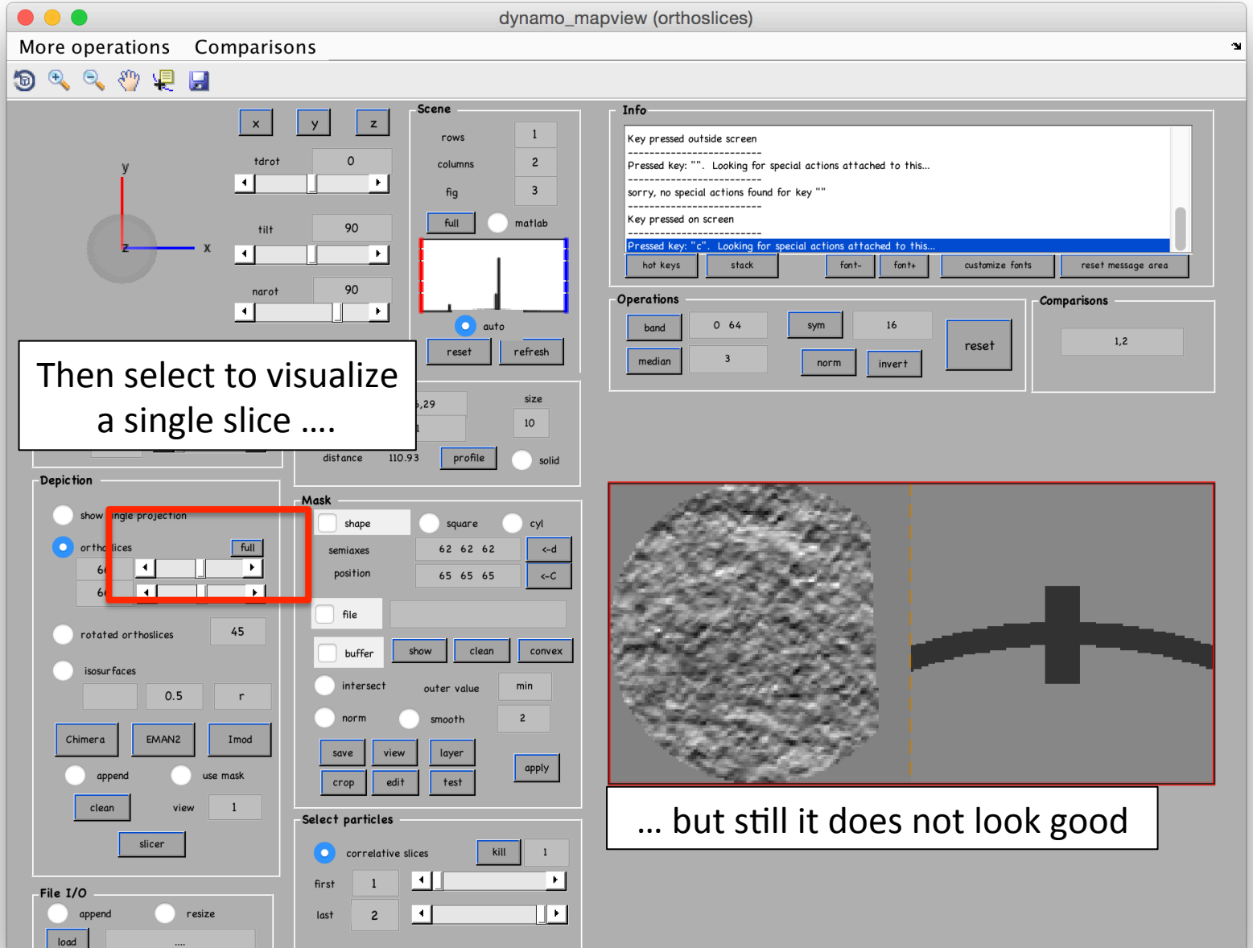
so… let's see if our selection of parameters produced a sensible alignment:
We show slice by slice how the align particle relates to the template

```
>> dmapview({sal.aligned_particle,sal.template});
```

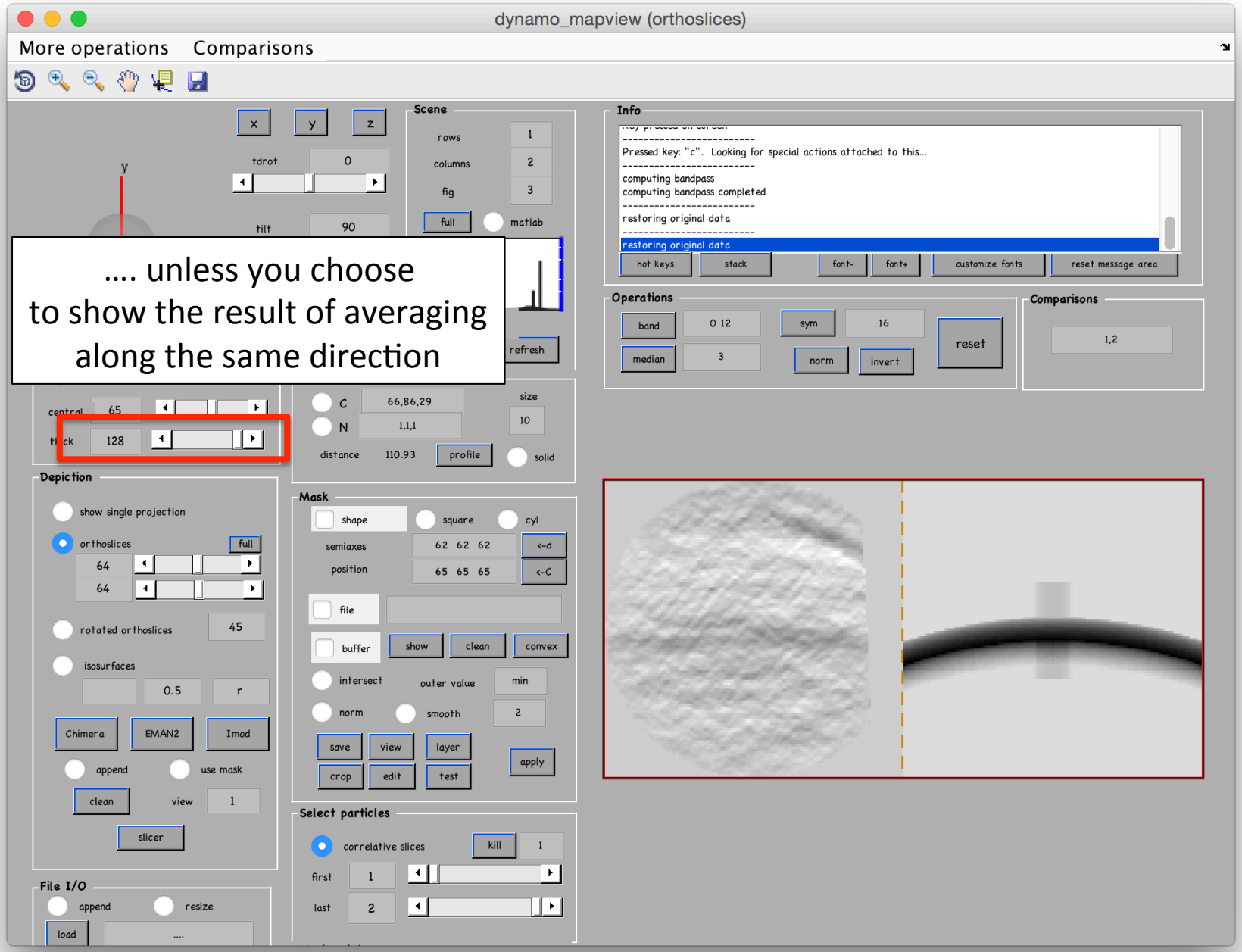
out of the box, it looks bad… can we actually get a reasonable comparison?




dynamo_mapview (orthoslices)
More operations
Comparisons
x
y
z
tdrot
0
tilt
90
narot
90
Scene
rows
1
columns
2
fig
3
full
matlab
auto
reset
refresh
Info
Pressed key: "". Looking for special actions attached to this...
sorry, no special actions found for key ""
Key pressed outside screen
hot keys
stack
font-
font+
customize fonts
reset message area
Operations
band
0 64
median
3
sym
16
norm
invert
Comparisons
1,2
Projected slices
central
65
thick
1
Clicks
C
N
1,1,1
size
10
distance
0.00
profile
solid
First normalize equally both intensities
Depiction
show single projection
orthoslices
1
128
rotated orthoslices
45
isosurfaces
0.5
r
Chimera
EMAN2
Imod
append
use mask
clean
view
1
slicer
file
buffer
show
clean
convex
intersect
outer value
min
norm
smooth
2
save
view
layer
crop
edit
test
apply
Select particles
correlative slices
kill
1
first
1
last
2
File I/O
append
resize
load



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


dynamo_mapview (orthoslices)
More operations
Comparisons
x
y
z
tdrot
0
tilt
90
Scene
rows
1
columns
2
fig
3
full
matlab
refresh
.... unless you choose
to show the result of averaging
along the same direction
central
65
thick
128
C
66,86,29
N
1,1,1
distance
110.93
profile
size
10
solid
Info
Pressed key: "c". Looking for special actions attached to this...
computing bandpass
computing bandpass completed
restoring original data
restoring original data
hot keys
stack
font-
font+
customize fonts
reset message area
Operations
band
0 12
sym
16
median
3
norm
invert
reset
Comparisons
1,2
Depiction
show single projection
orthoslices
full
64
64
rotated orthoslices
45
isosurfaces
0.5
r
Chimera
EMAN2
Imod
append
use mask
clean
view
1
slicer
Mask
shape
square
cyl
semiaxes
62 62 62
<-d
position
65 65 65
<-C
file
buffer
show
clean
convex
intersect
outer value
min
norm
smooth
2
save
view
layer
crop
edit
test
apply
Select particles
correlative slices
kill
1
first
1
last
2
File I/O
append
resize
load
....

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.

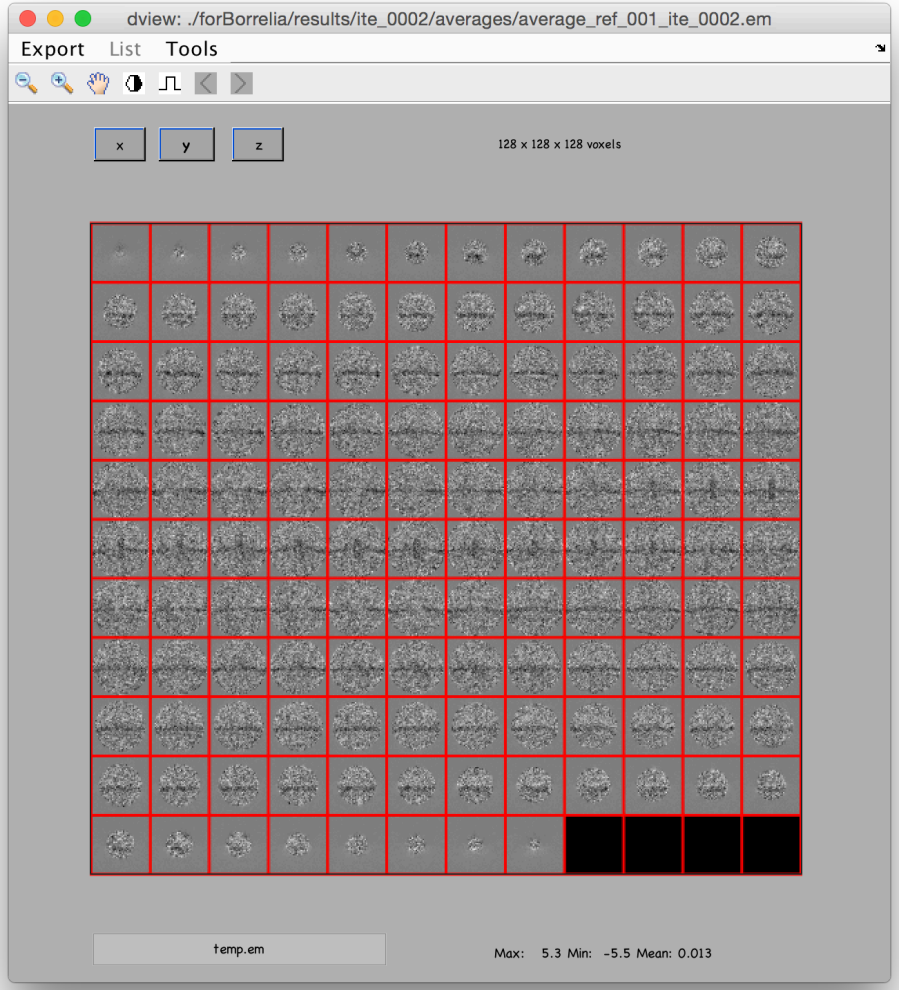
It means that you can now create a small project that uses the same angular settings that this alignment command on all the particles!

Exercise

Create and run this project!
Hint: use some shift limiting policy

Let's take a look on how the average looks like:

```
>>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`