Tutorial ManifoldEM 1.0.0

Ghoncheh Mashayekhi

2020

Table of Contents

[Introduction 2](#_Toc46220665)

[Demo Data 2](#_Toc46220666)

[GUI 3](#_Toc46220667)

[Input 3](#_Toc46220668)

[Tessellation 4](#_Toc46220669)

[Running on a Cluster or a Single machine 5](#_Toc46220670)

[Calculations 5](#_Toc46220671)

[1. Distance Calculation 5](#_Toc46220672)

[2. Embedding 6](#_Toc46220673)

[3. NLSA 6](#_Toc46220674)

[Make 2D Movies 6](#_Toc46220675)

[Propagate the conformational coordinate (CC) among prDs 6](#_Toc46220676)

[Extract data for 3D reconstructions along a Conformational Coordinate 7](#_Toc46220677)

[Helpful Notes 8](#_Toc46220678)

[Checking the manifolds 8](#_Toc46220679)

[Checking the manifolds after NLSA 8](#_Toc46220680)

[References 9](#_Toc46220681)

# Introduction

ManifoldEM is a stand-alone MATLAB program to extract continuous conformational changes from single-particle cryo-electron microscopy data. It is developed in the Ourmazd research group at the University of Wisconsin Milwaukee (UWM) (https://sites.uwm.edu/ourmazd/). The underlying methodology with application to different experimental datasets is described in the following publications.

1. Dashti, A. et al. Trajectories of the ribosome as a Brownian nanomachine. *Proc Natl Acad Sci U S A* **111**, 17492-7 (2014).
2. Dashti, A. & Mashayekhi, G. et al. Retrieving Functional Pathways of Biomolecules

from Single-particle Snapshots. *Nature Communication,* (2020). (Under review)

If ManifoldEM is useful in your work, please cite the papers 1 & 2.

The software needs the Cryo-EM snapshots (aligned particle images) with the assigned orientations and defocus values (alignment file). For the preprocessing of the data and extracting the orientations and related parameters any software could be used. In this repository, a MATLAB function (getStarFile.m) is provided to extract the information from the .star file and put it in the appropriate format of ManifoldEM. As for the image stack the default setting is for .mrcs files. But such files/settings could be developed for any other data format to make the software compatible with.

Over the years, the MATLAB codes were developed in different MATLAB versions. The current distribution has been tested with MATLAB 2015b and 2019a 64 bit. Some functions used in ManifoldEM were part of standard MATLAB 2015, but moved to special toolboxes in the later versions.[[1]](#footnote-1)

All the main functions and scripts are located in the Codes/MainScriptsnFunctions folder. All other functions and codes which are called in the main function and scripts are located in the Codes/Utilities folder. A GUI (ManifoldEM-Gui.fig/m) is provided to help with organizing and running these functions. Some additional codes for data representation or preprocessing is added in the ExtraCodes folder just in case they would be helpful.

# Demo Data

A [Demo Dataset](https://panthers-my.sharepoint.com/:f:/g/personal/mashayek_uwm_edu/Er5pr36Rz01IgUxHSFa_eeQBNyF01hRGc6n4vVL7R0uMrQ?e=kzP6Da), which is a subset of the experimental data from the functional pathway analysis of ryanodine receptor type 1 (RyR1)[1], is provided. This is only a small subset to help with understanding the software and for training purposes. Otherwise we do not expect to have the best performance with this limited dataset.

# GUI

A GUI is provided for the users to make the workflow easier. You can run it by opening the ManifoldEM-Gui.fig in GUIDE. It looks like the following figure. We explain each part of the GUI and integrated functions in the next sections.

A screenshot of a cell phone

Description automatically generated

Software GUI

# Input

At first, the alignment and image stack files should be fed into the algorithm. Currently, the getStarFile.m function is provided to extract the orientations, defocus and other essential parameters form the star file and convert them to the acceptable format for ManifoldEM. This code can be revised to accommodate other formats as well.

In ManifoldEM, first the 2D conformational changes are extracted for each viewing angle (what we call projection direction (prD)), then they are mapped to 3D conformational changes. To do so, first the S2 sphere is divided into small apertures (prDs) depending on the size of the particle and resolution. In the ideal case, it is better to have the Aperture Size equal to one Shannon Angle which is equal to resolution over the diameter of the object. But because we are always dealing with limited number of Snapshots, we may increase the Aperture Size by some factor (Aperture Index). The ideal dataset would be the one with uniformly distributed prDs all over the S2, with enough number of snapshots in each prD with the aperture size equal to one Shannon.

The assumption in ManifoldEM is that, the effect of orientational changes is less than the conformational changes in each prD, therefore diffusion map extracts the conformational changes. But we know that, by increasing the aperture size we are increasing the effect of orientations and from some point on our dominant eigenfunctions would pick orientations. So, in selecting the aperture size, we need to do a trade-off to have the most number of snapshots possible without having a notable orientational effect.

In GUI, in the input panel, you would need to spot where the .star and .mrcs file are located and also set the parameters which explained above. For the demo data, we go with following settings:

Resolution: 5 A

Diameter: 360 A

Aperture Index: 4

Aperture Size=4\*5/360

Please note that, the diameter (+10 pixel size) is used to create a circular mask (in the center of image) around the object in the calculation. So be sure to enter the maximum diameter of the object here. Otherwise some part of the object might be masked out from calculation.

Please note that depending on the data size, reading data from the .star file and saving it might take some time. After pushing the button wait until the button is enabled again to proceed.

# Tessellation

As explained in the input section and our previous publications, we first divide the orientational space to small patches called prDs and then use diffusion map and NLSA to extract the conformational coordinates. The S2 tessellation panel in the GUI does that for us based on the provided Aperture Size by the user. In this panel, two options are provided:

* Great Circle
* S2

If you select the great circle option, the algorithm would look for the mostly populated great circle on the S2 and select that for further analysis. In the current settings it looks for one great circle. But you can change that setting (GCnum) in the callback function of the start. By selecting more great circles, the algorithm tries to find the GCnum highly populated great circles which are further apart. In cases, that you do not care for the resolution and want to have an overview of what is going on in the dataset, you can select a great circle.

If you select the S2 option, the algorithm would divide the whole S2 to patches. There is a threshold set in the S2tesselation function in the variable called PDsizeTh. The prDs with smaller than PDsizeTh number of snapshots are not included in the next calculations.

At the end of both methods, a graph is represented which shows the prDs on S2. The colorbar represents the number of snapshots in each prD. Please note that, these two tessellation methods are using two different approaches; in the great circle one, the great circle is divided into prDs with the width of an aperture size (square shape patches), whereas in S2 tessellation the patches’ shapes depend on the triangulation. We are reporting the results of demo data with S2 tessellation method.

Please note that, the more snapshots we have in each prD, the better would the quality of conformational movies be and vice versa.

Please note that, the better coverage of S2 (uniformly), would end up in better resolution of final 2D conformational movies.

A close up of a map

Description automatically generated

S2 tessellation result for demo data

# Running on a Cluster or a Single machine

The calculations in this software could either be done on a single machine or on a cluster using Matlab Parallel Server. To run on a cluster, you need to check the Par Cluster option in the calculations panel in the GUI and specify in which directory on the cluster you want the results to be stored. For cluster computing, the default parallel computing cluster profile, with the cluster object properties set to the values defined in that profile would be used.

Please note that, to be able to run the calculations on a cluster, you also need to copy the original .mrcs file (not the mrcsfile.mat!), starFile.mat, S2tessellationResult.mat files to the Data folder in the specified directory on the cluster.

Please note that you must take care of appropriate files/directories access settings on the cluster.

# Calculations

Instead of putting all the calculations in one huge script, the calculations are divided into sub- categories to help with understanding the methodology.

The main calculations before getting the continuous conformational motions consist of the following steps [1,2]:

1. Distance Calculation
2. Embedding
3. NLSA

These steps should be done in the specified order. The results of one step is needed for the other one and are stored either on the local machine or the cluster based on the user’s selection (checking/unchecking the Par Cluster option in the GUI). Following we would explain each step briefly:

## Distance Calculation

As explained before, diffusion map embedding is used to embed the snapshots from high dimensional space to a much lower dimensional space. The first step in embedding is the distance calculation. Please specify the number of jobs you want to submit at a time to the cluster through the Number of Par jobs edit box in the GUI.

The results of distance calculation will be saved in a folder called Dist.

## Embedding

The next step would be the embedding. In the embedding there are two parameters tune and rad which are set to default values in the GUI. Tune is used to change the sigma value by multiplying it with the optimum calculated sigma. In the provided algorithm, embedding and trimming are being done together until all the points lie within a hypersphere spanned by the first 3 eigenfunctions with a radius specified by rad.

The results of embedding will be saved in a folder called EM.

## NLSA

This step visualizes the conformational changes along each diffusion map eigenfunction. This information can be used to identify biologically interesting conformational coordinates. For more information about Non-Linear Spectral Analysis please refer to [3]. The user can set the Number of Eigenfunctions of diffusion map for which the NLSA would be applied. The results of NLSA calculation will be saved in a folder called NLSA.

After NLSA along each eigenfunction, we expect to have a 1 dimensional manifold (parabola) defined by √2cos(*k* 𝜋 τ). Fitting a parabola to this 1D manifold, will give us the parameter τ . The mapping from a space of unknown metric to one characterized by a parameter τ allows a consistent description of the conformational changes among the projection directions.

Please note that, if in any prDs along any eigenfunction, we do not get any parabola (you can scatter plot the psirec in NLSA generated mat files), it means that with the assigned setting the NLSA did not converge. So, either you should change the settings or remove that prD from further analysis. Normally in these cases, the 2D movies which are generated in the next step would be flashy and not smooth.

# Make 2D Movies

After NLSA, to visualize the conformational changes, we need to make the 2D movies along each eigenfunctions. This pushbutton on the GUI would do so. The movies consist of 50 frames along each eigenfunctions.

Please note that, movie making is being done on the local machine and need the results of NLSA step. But, if you have done the calculations in the cluster, you should check the par cluster option in the calculation panel, so that the function would fetch the results of NLSA from the cluster. Please set the Number of Par Jobs cautiously to not fill up the /var space in your cluster.

# Propagate the conformational coordinate (CC) among prDs

In this step we need to find out the conformational changes that we are looking for in all prDs to map the 3D conformational changes. If there is one dominant and large conformational motion going on, this motion is most likely to be captured at the first eigenfunction of the manifold. But if there are subtle changes happening and we do not have enough number of snapshots and/or there are more than one conformational coordinate, or other artifacts the order of the eigenfunctions ranked according to the eigenvalue may be different in different projection directions. Also, the sense of the conformational changes may change due to sense-neutrality in eigen-decomposition.

In order to establish a consistent set of conformational coordinates across different projection directions, we need to have an automated software. We are still working on such a software. We have developed a couple of methods to propagate the conformational coordinates across the S2, but there is none which could be used for all the datasets in general and compete with the manual selection with acceptable accuracy.

The user can do this, by looking at the 2D movies generated by the algorithm and select the eigenfunctions which shows the specific changes with a specific sense. The results must be saved in a file called slcPsinums.mat, which is a # of prD by 2-dimensional matrix. The first column is representing the selected eigenfunction number and the second column is representing the sense (it is either 1 or -1). If for any prD the conformational coordinate could not be selected, then the selected eigenfunction should be put as zero.

By pushing the propagate the CC button in GUI, the user needs to select the slcPsinums.mat file and then the selected eigenfunctions are mapped with the correct sense.

The slcPsinums.mat for the demo data is stored in the Demo Data folder.

# Extract data for 3D reconstructions along a Conformational Coordinate

Now that we have the 50 bins (frames) 2D movies for each prDs, we can make a 3D movie of 50 frames along that conformational coordinate. We would have 50 volumes representing each bin. By pushing the Extract 3D Reconst Data button in the GUI, the processed snapshots and corresponding orientations are extracted from ManifoldEM files and generated for 3D reconstruction. At the end we would have 50 .star files and 50 .mrcs files which can be used by Relion [4] to reconstruct 3D volumes.

Please note that depending on the number of snapshots, this procedure might take long. The button would be activated after the procedure being complete. At the end of this part the image stack (imgsSPIDER\_\*\_of\_50.dat) and their coresponding orientations (align\_\*.dat) for each of the 50 bins would be ready in .dat files.

Please note that this step also can be run on the cluster. You need to check the Par Cluster option in the GUI like the calculations section and specify the directory and copy the necessary files to the cluster (starFile.mat, S2tessellationResult.mat files to the Data folder in the specified directory on the cluster).

To use Relion to make the 3D volumes, some addition scripts are provided. You need to copy them to the Dat folder and run the following scripts in the Dat folder path:

1. ./setup\_data.perl max\_number start\_number increment

Please change the --bg\_radius size to the mask size, which is (floor(Dimater/PixSize/2)+5). If this radius does not match the masksize, you would receive a warning about std being zero by Relion.

If you want to make all the 50 volumes, you should use the following command:

./setup\_data.perl 50 1 1

Or if you want to make every other volumes from 1 to 50, you should use the following:

./setup\_data.perl 50 1 2

1. ./create\_star.perl max\_number start\_number
2. rename \_0 \_ norm\_psi\_Class50\_\*.mrcs
3. ./reconst.perl max\_number start\_number increment

Please note the current setting in the reconst.perl is to impose a C4 symmetry. You can remove it depending on the dataset.

You can delete all the .dat files after being done with this step.

Please note that, after having all the 50 volumes, SVD can be applied on these volumes to have an additional step of noise removal and have smoother motions. The SVDonVolumes.m function is developed to do so and is included in the ExtraCodes folder.

Please note that, any further polishing like beta factor correction could be applied to these volumes.

# Helpful Notes

## Checking the manifolds

After embedding section is done, it is good to review the manifolds to see if they have any specific shape, or any clusters. As the conformational changes would be continuous, we do not expect to see clusters in the conformational manifold. These clusters could be representative of some artifacts. If this is the case, it is better to remove the snapshots coming from the artifact cluster from your analysis. You can investigate to see what these clusters are representing.

As for an example please take a look at the SI Fig.7 of our paper2 which shows such clustering happening in RyR experimental dataset. That cluster is showing the images with low contrast than others. In the provided demo data we have removed those clusters. But you can see that in some prDs, that artifact removal is not perfectly done and we have the tace of it on our 2D movies.

The provided codes in Extra codes folder might be helpful in this regard.

## Checking the manifolds after NLSA

As mentioned in the text, after applying NLSA, we expect to have a one-dimensional manifold for each of the eigenfunctions as shown below. If it is not the case, the NLSA could not converge with the specified settings. So, the results are not reliable. When looking at the 2D movies by eye, you would see this being reflected in flashy and not smooth movies.

A picture containing lot

Description automatically generated

# References

1. Dashti, A. et al. Trajectories of the ribosome as a Brownian nanomachine. *Proc Natl Acad Sci U S A* **111**, 17492-7 (2014).
2. Dashti, A. & Mashayekhi, G. et al. Retrieving Functional Pathways of Biomolecules

from Single-particle Snapshots. *Nature Communication,* (2020). (Under review)

1. Giannakis, D. & Majda, A.J. Nonlinear Laplacian spectral analysis for time series with intermittency and low-frequency variability. *Proc Natl Acad Sci U S A* **109**, 2222-2227 (2012).
2. Scheres, S.H. RELION: implementation of a Bayesian approach to cryo-EM structure determination. *J Struct Biol* **180**, 519-30 (2012).

1. You might encounter this issue when running great circle tessellation (selectGCs.m). This function uses kmeans which requires the Statistics and Machine Learning Toolbox in Matlab 2019a and higher. But also, this can be replaced by substitutes. [↑](#footnote-ref-1)