ChimeraX MorphOT User Manual

arthurecoffet

July 2020

Contents

1 Introduction

1.1 What is MorphOT ?

The main purpose of our bioinformatic tool, *MorphOT*, is to adapt and optimize the OT-based interpolation of shapes for EM maps, as a plugin for *ChimeraX*. *MorphOT* allows users to directly interpolate multiple density maps, with a significant improvement in quality, compared with the standard interpolation command provided with *ChimeraX*. Some theoretical details and motivation are available <At our paper >

1.2 Development and requirements

MorphOT has been developed using Python 3.7 for implementation as a plugin into UCSF ChimeraX. A GPU implementation of all *MorphOT* function is provided, the use of GPUs requires NVidia GPUs and the Cuda Toolkit.

1.3 Downloading and running ChimeraX-morphOT

While the plugin hasn't been reviewed by UCSF ChimeraX team, one can download the source code folder *ot-morph* on the following github.
Then type into the chimeraX command line :

```
devel build Path/To/Source/Code/ot-morph
```

and

```
devel install Path/To/Source/Code/ot-morph
```

After UCSF ChimeraX team reviewal, the tool will be available either by clicking **More tools** in the ChimeraX **Tool** menu, or on the internet here by searching *MorphOT*.

2 General Features

2.1 Input Type

MorphOT takes as input two ChimeraX density maps. Since the main morph method requires smooth inputs, it is likely that the density maps require preprocessing, see **Tutorial** for more details on how to preprocess data.

The plugin *MorphOT* is composed of four main function, which documentation is detailed below.

2.2 morphOT

- **MorphOT morphOT** *volume-spec* [**start** *start-fraction*] [**playStep** *increment*] [**frames** *N*] [**playDirection 1** | -1] [**playRange** *low-fraction, high-fraction*] [**rate 'linear'** | 'sinusoidal', 'ramp up', 'ramp down'] [**maxsize** *max*][**constantVolume** true | **false**] [**hideOriginalMaps true** | false] [**interpolateColors true** | false] [**niter** *K*] [**reg** *r*] *new-map-options*

  morphOT has the same description as previous ChimeraX "volume morph" : Morph between two or more maps. For a reasonable result, the input maps should have the same grids: dimensions, spacing, and numbers of points. Note volume resample can be used to make a copy of one map that has the same grid as another. A morphing fraction of 0.0 corresponds to the first map and a fraction of 1.0 corresponds to the last, with intermediate maps evenly spaced within that range. There is smooth interpolation between each adjacent pair of maps.

  The morph display will proceed from *start-fraction* (default **0.0**) in steps of *increment* (default **0.04**) for *N* **frames** (default **25**). If the number of frames and step increment exceed what is needed to reach the **playRange** bounds (default is the entire range : **0.0,1.0**), the morph display will "bounce" back and forth. The **rate** option (default 'linear') has the coefficients change at linear, sinusoidal, ramp up (slow at the beginning, fast at the end) or ramp down rate. The **maxsize** option (default 60) sets the maximum grid size (maxsize^3) over which structures are resized to maxsize for efficiency issues. The **constantVolume** option specifies adjusting the threshold (contour level) automatically to keep the enclosed volume constant. The **hideOriginalMaps** option specifies hiding the input maps. The **interpolateColors** option only applies when the maps have the same number of threshold (contour levels for surface/mesh

display). The **niter** option specifies (default **20**) specifies the number of iteration in the loop computing each morph frames, the greater this parameter the more accurate each frames are, yet computation scales linearly with this term and **20** is generally enough for convergence [**?**]. The **reg** option (default **max_grid_dimension/60**) defines the entropy parameter in [**?**], convergence of each frames is faster when this parameter gets bigger yet it also implies a more blurred morphing, on the other hand there tends to be numerical errors if the parameter gets too small. See here for details about *new map options*

The morph is created in a new map (volume) model. However, if the modelId of an existing morph map is given, the existing morph will be used instead of a new one being calculated.

2.3 semiMorphOT

- **MorphOT semiMorphOT** *volume-spec* [**ot-frames** N_{OT}] [**frames** N] *morph-ot-options*

  An approximation of *morphOT*, instead of computing each frame using Wasserstein Barycenters (OT theory), *semimorphOT* computes N_{OT} Wasserstein Barycenter between which it linearly interpolates. Althoug linear interpolations produces results which don't have any physical interpretation and which can be not satisfying on *big, non linear* transition pathways, the results are satisfying when linearly interpolating between close shapes computed using Wasserstein Barycenters. *semiMorphOT* allows to compute satisfying transition pathways with a good compromise on computation time for bigger structures, thus sometimes allowing to produce prompt transition movies.

  The option **ot-frames** (default 4) determines the number of optimal transport barycenters between which linear interpolation will be displayed. The option **frames** (default 25) is the *total* number of frames displayed.

2.4 oneBarycenter

- **MorphOT oneBarycenter** *volume-spec weights , w1,w2 ...* [**niter** K] [**reg** r] [**maxsize** *max*] [**interpolateColors true** | false] *new-map-options*

  oneBarycenter computes and display one weighted barycenter of *two or more* volumes given the **weights** (as many as volumes, summing to one). The other options behave as in *morphOT* and *semiMorphOT*.

2.5 BarycenterSave

- **MorphOT BarycenterSave** *volume-spec folder_path* [**frames** N] [**name1 n1**] [**name2 n2**] [**niter** K] [**reg** r] [**rate 'linear'** | 'sinusoidal', 'ramp up', 'ramp down'] *new-map-options*

  BarycenterSave computes and save locally the transition pathway displayed by *morphOT*. Options **name1** and **name2** allow the user to chose the volume names used for saving each barycenter. The file name has form :

  %frame_number%_%name1%_%name2%_weights_%current_frame_weights%.mrc

  Other options behave the same as in the previous functions.

3 Tutorial

1. First, optimal transport morphing requires "smooth" structures with little noise. Let's begin with a preprocessing structures tutorial.

   We use on this example EMDB Structure 5140. As one can see on the screenshot below, a lot of noise is present with low density values

   <Insérer screenshot >

   To smooth the structure we apply a gaussian blur by typing :

   ```
   volume gaussian #1 sd 2 %Find a way to explain that #1 is #structure_num
   ```

   <Screenshot result >

2. Once the noise has been reduced by smoothing, we can also threshold the structure density to only get the structure shape

   <Screenshot two times same structure one we can see the ring around the other one clean and highlight density value >

   We can do this by typing the following in the command line :

   ```
   volume threshold #2 min 'threshold_value'
   ```

   Finally, ChimeraX deals better with displaying the transition pathway if inputs have values between 0 and 1 so we type :

   ```
   volume scale #3 shift -"threshold_value"
   ```

   Our first structure has been preprocess and is ready. Assume the same operations have been done to the second structure

3. In the case where both structure are not on the same grid, one can use **volume resample** to correct this by typing (assuming the first and second structure have IDs 1 and 2)

   ```
   volume resample #2 onGrid #1
   ```

All structures are now preprocessed and have the same size, we can start using *MorphOT*

3.1 displaying morphOT movie

MorphOT morphOT #1 #2

4 Appendix

4.1 Running on GPU