

Cell tracking with Optimal Transport

src/example.ipynb: an example script for using the functions in
utils_3Dtracking.py

src/utils_3Dtracking.py:

Helper functions for display and outputs

- **show_4d_with_contours(im,seg)** produces an interactive viewer of the images where the first two dimensions (time, slices) are scrollable via sliders and the segmentation labels appear as contours
 - Inputs:
 - im: 3D grayscale image normalised between 0 and 1
 - seg: 3D integer label map
- **generate_tree(track_df,split_df,merge_df)** plots a division tree based on the three input dataframes
 - Inputs:
 - track_df: dataframe containing the tracks from create_tracks (see example script for usage)
necessary columns = 'track_id', 'timepoint'
This dataframe might contain an additional column called "colours" for specifying the colours to be used to plot each cell at each time point
 - split_df: dataframe containing cell divisions from create_tracks (see example script for usage)
necessary columns = 'parent', 'child_0', 'child_1', 'timepoint'
 - merge_df: dataframe containing cell merges from create_tracks (see example script for usage)
necessary columns = 'parent_0', 'parent_1', 'child', 'timepoint'
created by *create_simplified_tessellation* into 2D
 - Outputs:
 - tree: track_ids in an order that makes the division plot tidy
 - start: time points where all track_ids first appear
 - end: time points where all track_ids last appear

Helper functions for tracking

- **fit_Gaussian_mixture(im,seg,label_file,resolution)** calculates Gaussian parameters of all segmented regions in an image
 - Inputs:
 - im: 3D grayscale image normalised between 0 and 1
 - seg: 3D integer label map

- **label_file**: file name of the label file; this is just to print out when `list(map())` is used
 - **resolution**: resolution of the image (usually in um)
 - Outputs:
 - **integrals**: list of integrals of each region
 - **centers**: list of centre coordinates of each region
 - **widths**: list of covariance matrix of each region
 - **K**: number of labels
 - **labels**: list of region labels
 - Additional helper functions called by *fit_Gaussian_mixture*:
 - **get_moments(data,resolution)** calculates ND Gaussian parameters of data
- **GW2_ak(pi0,pi1,mu0,mu1,S0,S1)** returns the GW2 discrete map and the GW2 distance between two GMM

Note from AK: this function is originally from gmmot. I modified it to use sinkhorn regularisation instead of emd, because that produces better results.
 - **create_tracks(start_track_ids,target_labels,transition_matrix,time_point)** ceating tracks based on the transition_matrix
 - Inputs:
 - **start_track_ids**: track_ids of the cells at the start (i.e. columns)
 - **target_labels**: labels of cell at the end (i.e. rows)
 - **transition_matrix**: transition probability matrix
 - **time_point**: time point
 - Outputs:
 - **track_section**: track_section dataframe of the next time point with track_ids filled out based on transition_matrix
 - **split_section**: split_section dataframe of the next time point with parent and children ids filled out based on transition_matrix
 - **merge_section**: merge_section dataframe of the next time point with parents and child ids filled out based on transition_matrix,
 - **target_track_ids**: track ids of the cells at the end time point. These will be the start_track_ids for create_tracks of the next time point.
 - Additional helper functions called by *create_tracks*:
 - **valid_transition_ver_2(transition_matrix)** turns transition_matrix into the highest probability valid transition matrix where cells may not split into more than two pieces and no more than two cells can merge at a time