Reference

class PHETS.PH.**Filtration**(*sig*, *params*[, *filename='none'*])

Parameters:
- **sig** (*ndarray or str*) – Input trajectory as numpy array or path to .txt file.
- **params** (*dict*) – Options for computation of filtration.
- **filename** (*str*) – If sig is an ndarray, a filename may be provided for labelling in visualizations.

PHETS.PH.**make_movie**(*filtration*, *out_filename*[, *color_scheme='none'*, *camera_angle=(135, 55)*, *alpha=1*, *dpi=150*, *save_frames=False*, *framerate=1*])

Make filtration movie.

Parameters:
- **filtration** (*Filtration*) –
- **out_filename** (*str*) – Path/filename for movie. Should end in ".mp4" or other video format.
- **color_scheme** (*str*) – 'none', 'highlight new', or 'birth_time gradient'
- **camera_angle** (*tuple*) – for 3D mode. (azimuthal, elevation) in degrees.
- **alpha** (*float*) – opacity of simplexes. 0…1 : transparent…opaque
- **dpi** (*int*) – dots per inch (resolution)
- **save_frames** (*bool*) – save frames to PH/frames/frame*.png for debugging
- **framerate** (*int*) – frames per second

PHETS.PH.**make_PD**(*filtration*, *out_filename*)

Plot persistence diagram.

Parameters:
- **filtration** (*Filtration*) –
- **out_filename** (*str*) – Image path and filename. Should end in ".png" or other image format.

PHETS.PH.**make_PRF_plot**(*filt*, *out_filename*[, *PRF_res=50*])

Plot persistent rank function.

Parameters:
- **filtration** (*Filtration*) –
- **out_filename** (*str*) – Image path and filename. Should end in ".png" or other image format.
- **PRF_res** (*int*) – number of divisions per epsilon axis

Todo:

a function for a single frame of the filtration (i.e. fixed epsilon) as an image. (For now, a single frame movie can be used.)

PHETS.DCE.Movies.**slide_window**(*in_filename*, *out_filename*, *window_size=.5*, *step_size=.1*, *tau=10*, *ds_rate=1*, *max_frames=0*, *save_worms=True*, *save_movie=True*)

Show embedding of in_filename with window start point varied over time.

Parameters:
- **window_size** (*float*) – seconds
- **step_size** – window start point step size (seconds)
- **tau** (*float*) – seconds ?

PHETS.DCE.Movies.**vary_tau**(*in_filename*, *out_filename*, *tau_lims=(1, 15)*, *tau_inc=1*, *embed_crop=(1, 2)*, *ds_rate=1*, *save_worms=True*, *save_movie=True*, *m=2*)

Show embedding of in_filename with tau varied over time.

Parameters:

- **in_filename** (*str*) – Path/filename for text file time series.
- **out_filename** (*str*) – Path/filename for movie. Should end in ".mp4" or other video format.
- **tau_lims** (*tuple*) – tau range (seconds)
- **tau_inc** (*int*) – tau stepsize (seconds)
- **embed_crop** (*tuple*) – Limits for window from input time series (seconds)
- **ds_rate** (*int*) – time series downsample rate
- **save_worms** (*bool*) – save embeddings to text files in output/DCE /saved_worms
- **save_movie** (*bool*) – If False, no movie will be created. Useful for saving embeddings quickly.
- **m** (*int*) – target embedding dimension

PHETS.DCE.Movies.**compare_vary_tau**(*in_filename_1*, *in_filename_2*, *out_filename*, *tau_lims*, *tau_inc=1*, *embed_crop=(1, 2)*, *ds_rate=1*, *m=2*, *save_worms=True*, *save_movie=True*)

Like vary_tau(), but shows embeddings for two time series side by side.

PHETS.DCE.Movies.**compare_multi**(*dir1*, *dir1_base*, *dir2*, *dir2_base*, *out_filename*, *i_lims=(1, 89)*, *embed_crop_1='auto'*, *embed_crop_2='auto'*, *auto_crop_length=.3*, *tau_1='auto ideal'*, *tau_2='auto ideal'*, *tau_T=1/np.pi*, *save_worms=True*, *save_movie=True*, *normalize_volume=True*, *waveform_zoom=None*, *ds_rate=1*, *dpi=200*, *m=2*)

Takes two directories of (eg one with piano notes, another with range of viol notes), and generates a movie over a range note indexes (pitch). Tau and crop may be set explicity or automatically.

Parameters:

- **dir1** (*str*) – Path of first directory to be iterated over
- **dir1_base** (*str*) – Base filename for files in dir1
- **dir2** (*str*) –
- **dir2_base** (*str*) –
- **out_filename** (*str*) –
- **i_lims** (*tuple*) – (start, stop) index. Default is (1, 89).
- **embed_crop_1** (*tuple or str*) – (start, stop) in seconds or 'auto'
- **embed_crop_2** (*tuple or str*) –
- **auto_crop_length=.3** (*float*) – seconds
- **tau_1** (*str*) – explicit (seconds) or 'auto detect' or 'auto ideal'
- **tau_2** (*str*) –
- **tau_T** (*float*) – For use with auto tau: tau = period * tau_T
- **save_worms** (*bool*) –
- **save_movie** (*bool*) –
- **normalize_volume** (*bool*) –
- **waveform_zoom** –
- **ds_rate** (*int*) –
- **dpi** (*int*) –

> Todo:
>
> function for plotting embeddings without varying a parameter or input, as an image. (For now, a single frame movies can be used.)

PHETS.PRFCompare.**plot_dists_vs_ref**(*dir*, *base_filename*, *fname_format*, *out_filename*, *filt_params*, *i_ref=15*, *i_arr=np.arange(10, 20, 1)*, *weight_func=lambda i, j: 1*, *metric='L2'*, *dist_scale='none'*, *PRF_res=50*, *load_saved_PRFs=False*, *see_samples=5*)

Takes range of time-series files and a reference file. Generates PRF for each, and finds distances to reference PRF, plots distance vs index.

Parameters:

- **dir** (*str*) – input directory
- **base_filename** (*str*) – input base filename
- **fname_format** (*str*) – input filename format: 'base i or 'i base'
- **out_filename** (*str*) – output filename
- **filt_params** –
- **i_ref** (*int*) –
- **i_arr** (*arr*) –
- **weight_func** (*lambda*) – Default is lambda i, j: 1
- **metric** (*str*) – 'L1' (abs) or 'L2' (euclidean). Default is 'L2'.
- **dist_scale** (*str*) – 'none', 'a', or 'a + b'. Default is 'none'.
- **PRF_res** (*int*) – number of divisions used for PRF. Default is 50.
- **load_saved_PRFs** (*bool*) – reuse previously computed PRF set. Default is False.
- **see_samples** (*int*) – interval to generate PRF plots, PDs, and filtration movies when generating PRF set. 0 is none, 1 is all samples, 2 is every other sample, etc.

`PHETS.PRFCompare.`**`plot_dists_vs_mean`**(*filename_1, filename_2, out_filename, filt_params, load_saved_PRFs=False, time_units='seconds', crop_1='auto', crop_2='auto', auto_crop_length=.3, window_size=.05, num_windows=10, mean_samp_num=5, tau_1=.001, tau_2=.001, tau_T=np.pi, note_index=None, normalize_volume=True, PRF_res=50, dist_scale='none', metric='L2', weight_func=lambda i, j: 1, see_samples=5*)

Takes two time-series or 2D trajectory files. For each input, slices each into a number of windows. If inputs are time-series, embeds each window. Generates PRF for each window. selects subset of window PRFs, computes their mean, plots distance to mean PRF vs time.

Parameters:

- **filename_1** (*str*) –
- **filename_2** (*str*) –
- **out_filename** (*str*) –
- **filt_params** (*dict*) –
- **load_saved_PRFs** (*bool*) –
- **time_units** (*str*) –
- **crop_1** (*str or tuple*) –
- **crop_2** (*str or tuple*) –
- **auto_crop_length** (*float*) –
- **num_windows** (*int*) – per file
- **mean_samp_num** (*int*) – per file
- **tau_1** (*str or float*) –
- **tau_2** (*str or float*) –
- **tau_T** (*float*) –
- **note_index** (*int*) –
- **normalize_volume** (*bool*) –
- **PRF_res** (*int*) – number of divisions used for PRF
- **dist_scale** (*str*) – 'none', 'a', or 'a + b'
- **metric** (*str*) – 'L1' (abs) or 'L2' (euclidean)
- **weight_func** (*lambda*) –
- **see_samples** (*int*) –

`PHETS.PRFCompare.`**`plot_clusters`**(**args, **kwargs*)

See plot_dists_vs_mean for call signature.

PHETS.PRFCompare.**plot_variances**(*filename*, *out_filename*, *filt_params*, *vary_param_1*, *vary_param_2*, *load_saved_PRFs=False*, *time_units='seconds'*, *crop=(100, 1100)*, *auto_crop_length=.3*, *window_size=1000*, *num_windows=5*, *tau=.001*, *tau_T=np.pi*, *note_index=None*, *normalize_volume=True*, *PRF_res=50*, *dist_scale='none'*, *metric='L2'*, *weight_func=lambda i, j: 1*, *see_samples=5*)

Parameters:

- **filename** (*str*) –
- **out_filename** (*str*) –
- **filt_params** (*dict*) –
- **vary_param_1** (*(str, tuple)*) – filtration parameter to vary over x axis
- **vary_param_2** (*None or (str, tuple)*) – filtration parameter to vary over line colors
- **load_saved_PRFs** (*bool*) – reuse saved
- **time_units='seconds'** (*str*) –
- **crop=(100, 1100)** (*tuple*) – (start, stop) in time units
- **num_windows** (*int*) – Number of windows to select from crop, evenly spaced. Window length is chosen with the ‘worm_length’ filtration parameter. Windows may or may not overlap
- **tau** (*int or float*) – time units
- **normalize_volume** (*bool*) – normalize volume (per crop)
- **normalize_sub_volume** (*bool*) – normalize volume (per window) [coming soon]
- **PRF_res** (*int*) –
- **dist_scale** (*str*) – ‘none’, ‘a’, or ‘a + b’
- **metric** (*str*) – ‘L1’ (abs) or ‘L2’ (euclidean)
- **weight_func** (*lambda*) –
- **see_samples** (*bool*) –