# **Python Documentation**

## version

May 13, 2020

# **Contents**

Welcome to moseq2-pca's documentation!	1
moseq2_pca package	1
CLI Module	1
moseq2-pca	1
apply-pca	1
clip-scores	2
compute-changepoints	2
train-pca	3
GUI Module	5
Utilities Module	5
Visualization Module	8
Subpackages	9
moseq2_pca.helpers package	9
Helpers - Data Module	9
Helpers - Wrapper Module	9
moseq2_pca.pca package	10
PCA - Utilties Module	10
Indices and tables	13
Index	15
Python Module Index	21

## Welcome to moseq2-pca's documentation!

### moseq2\_pca package

#### **CLI Module**

```
moseq2-pca
```

```
moseq2-pca [OPTIONS] COMMAND [ARGS]...
```

-d, --dask-cache-path <dask\_cache\_path>

Cluster queue/partition for submitting jobs [default: debug]

-q, --queue <queue>

-n, --nworkers <nworkers>
Number of workers [default: 10]

#### apply-pca

```
moseq2-pca apply-pca [OPTIONS]
                                                 Options
-i, --input-dir <input_dir>
 Directory to find h5 files [default: /Users/aymanzeine/Desktop/moseq/moseq2-pca/docs]
--cluster-type <cluster_type>
 Cluster type [default: local]
         Options: local|slurm|nodask
-o, --output-dir <output_dir>
 Directory to store results [default: /Users/aymanzeine/Desktop/moseq/moseq2-pca/docs/_pca]
--output-file <output_file>
 Name of h5 file for storing pca results [default: pca_scores]
--h5-path <h5_path>
 Path to data in h5 files [default: /frames]
--h5-mask-path <h5_mask_path>
 Path to log-likelihood mask in h5 files [default: /frames_mask]
--pca-path <pca_path>
 Path to pca components [default: /components]
--pca-file <pca_file>
 Path to PCA results
--chunk-size <chunk_size>
 Number of frames per chunk [default: 4000]
--fill-gaps <fill_gaps>
 Fill dropped frames with nans [default: True]
--fps <fps>
 Fps (only used if no timestamps found) [default: 30]
--detrend-window <detrend_window>
 Length of detrend window (in seconds, 0 for no detrending) [default: 0]
--config-file <config_file>
 Path to configuration file
```

Path to spill data to disk for dask local scheduler [default: /Users/aymanzeine/moseq2\_pca]

```
-c, --cores <cores>
 Number of cores per worker [default: 1]
-p, --processes cesses>
 Number of processes to run on each worker [default: 1]
-m, --memory <memory>
 RAM usage per workers [default: 15GB]
-w, --wall-time <wall_time>
 Wall time for workers [default: 06:00:00]
--timeout <timeout>
 Time to wait for workers to initialize before proceeding (minutes) [default: 5]
clip-scores
```

Clips PCA scores from the beginning or end

#### Args:

pca file (string): Path to PCA scores clip samples (int): number of samples to clip from beginning or end from\_end (bool): if true clip from end rather than beginning

Note that scores are modified in place.

```
moseq2-pca clip-scores [OPTIONS] PCA_FILE CLIP_SAMPLES
```

**Options** 

--from-end [default: False]

Arguments

PCA\_FILE

Required argument

CLIP\_SAMPLES

Required argument

#### compute-changepoints

```
moseq2-pca compute-changepoints [OPTIONS]
```

#### **Options**

```
-i, --input-dir <input_dir>
 Directory to find h5 files [default: /Users/aymanzeine/Desktop/moseq/moseq2-pca/docs]
-o, --output-dir <output_dir>
 Directory to store results [default: /Users/aymanzeine/Desktop/moseq/moseq2-pca/docs/_pca/]
--output-file <output_file>
 Name of h5 file for storing pca results [default: changepoints]
--cluster-type <cluster_type>
 Cluster type [default: local]
         Options: local|slurm
--pca-file-components <pca_file_components>
 Path to PCA components
--pca-file-scores <pca_file_scores>
 Path to PCA results
--pca-path <pca_path>
 Path to pca components [default: /components]
--neighbors <neighbors>
```

```
Neighbors to use for peak identification [default: 1]
--threshold <threshold>
  Peak threshold to use for changepoints [default: 0.5]
-k, --klags <klags>
 Lag to use for derivative calculation [default: 6]
-s, --sigma <sigma>
  Standard deviation of gaussian smoothing filter [default: 3.5]
-d, --dims <dims>
 Number of random projections to use [default: 300]
--fps <fps>
  Fps (only used if no timestamps found) [default: 30]
--h5-path <h5_path>
  Path to data in h5 files [default: /frames]
--h5-mask-path <h5_mask_path>
  Path to log-likelihood mask in h5 files [default: /frames_mask]
--chunk-size <chunk_size>
 Number of frames per chunk [default: 4000]
--config-file <config_file>
  Path to configuration file
--dask-cache-path <dask_cache_path>
  Path to spill data to disk for dask local scheduler [default: /Users/aymanzeine/moseq2_pca]
--visualize-results <visualize_results>
 Visualize results [default: True]
-q, --queue <queue>
 Cluster queue/partition for submitting jobs [default: debug]
-n, --nworkers <nworkers>
 Number of workers [default: 10]
-c, --cores <cores>
 Number of cores per worker [default: 1]
-p, --processes cesses>
 Number of processes to run on each worker [default: 1]
-m, --memory <memory>
  RAM usage per workers [default: 15GB]
-w, --wall-time <wall_time>
 Wall time for workers [default: 06:00:00]
--timeout <timeout>
  Time to wait for workers to initialize before proceeding (minutes) [default: 5]
train-pca
moseq2-pca train-pca [OPTIONS]
                                                 Options
-i, --input-dir <input_dir>
  Directory to find h5 files [default: /Users/aymanzeine/Desktop/moseq/moseq2-pca/docs]
--cluster-type <cluster_type>
  Cluster type [default: local]
         Options: local|slurm
-o, --output-dir <output_dir>
```

```
Directory to store results [default: /Users/aymanzeine/Desktop/moseq/moseq2-pca/docs/ pca]
--gaussfilter-space <gaussfilter_space>
 Spatial filter for data (Gaussian) [default: 1.5, 1]
--gaussfilter-time <gaussfilter_time>
 Temporal filter for data (Gaussian) [default: 0]
--medfilter-space <medfilter_space>
 Median spatial filter [default: 0]
--medfilter-time <medfilter_time>
 Median temporal filter [default: 0]
--missing-data
 Use missing data PCA [default: False]
--missing-data-iters <missing_data_iters>
 Missing data PCA iterations [default: 10]
--mask-threshold <mask_threshold>
 Threshold for mask (missing data only) [default: -16]
--mask-height-threshold <mask_height_threshold>
 Threshold for mask based on floor height [default: 5]
--min-height <min_height>
 Min mouse height from floor (mm) [default: 10]
--max-height <max_height>
 Max mouse height from floor (mm) [default: 100]
--tailfilter-size <tailfilter_size>
 Tail filter size [default: 9, 9]
--tailfilter-shape <tailfilter_shape>
 Tail filter shape [default: ellipse]
--use-fft
 Use 2D fft [default: False]
--recon-pcs <recon_pcs>
 Number of PCs to use for missing data reconstruction [default: 10]
--rank <rank>
 Rank for compressed SVD (generally>>nPCS) [default: 50]
--output-file <output_file>
 Name of h5 file for storing pca results [default: pca]
--chunk-size <chunk_size>
 Number of frames per chunk [default: 4000]
--visualize-results <visualize_results>
 Visualize results [default: True]
--config-file <config file>
 Path to configuration file
-d, --dask-cache-path <dask_cache_path>
 Path to spill data to disk for dask local scheduler [default: /Users/aymanzeine/moseq2_pca]
--local-processes <local_processes>
 Use processes with local scheduler [default: True]
-q, --queue <queue>
 Cluster queue/partition for submitting jobs [default: debug]
-n, --nworkers <nworkers>
 Number of workers [default: 10]
-c, --cores <cores>
 Number of cores per worker [default: 1]
```

#### -p, --processes cesses>

Number of processes to run on each worker [default: 1]

#### -m, --memory <memory>

Total RAM usage per worker [default: 15GB]

#### -w, --wall-time <wall\_time>

Wall time for workers [default: 06:00:00]

#### --timeout <timeout>

Time to wait for workers to initialize before proceeding (minutes) [default: 5]

#### **GUI Module**

moseq2\_pca.gui.apply\_pca\_command (input\_dir, index\_file, config\_file, output\_dir,
output\_file, output\_directory=None)

Compute PCA Scores given trained PCA using Jupyter Notebook.

#### Parameters:

- input\_dir (str) (path to directory containing training data)
- index\_file (str) (path to index file.)
- config\_file (str) (path to config file)
- output\_dir (str) (path to output pca directory)
- output\_file (str) (name of output pca file.)
- output\_directory (str) (alternative output directory path)

Returns: (str)

Return type: success string.

moseq2\_pca.gui.compute\_changepoints\_command (input\_dir, config\_file, output\_dir, output\_file, output\_directory=None)

Compute Changepoint distribution using Jupyter Notebook.

#### Parameters:

- input\_dir (str) (path to directory containing training data)
- config\_file (str) (path to config file)
- output\_dir (str) (path to output pca directory)
- output\_file (str) (name of output pca file.)
- output\_directory (str) (alternative output directory path)

Returns: (str)

Return type: success string.

moseq2\_pca.gui.train\_pca\_command (input\_dir, config\_file, output\_dir, output\_file, output\_directory=None)

Train PCA through Jupyter notebook, and updates config file.

#### **Parameters:**

- input\_dir (str) (path to directory containing training data)
- config\_file (str) (path to config file)
- output\_dir (str) (path to output pca directory)
- output\_file (str) (name of output pca file.)
- output\_directory (str) (alternative output directory path)

Returns:

Return type: None

#### **Utilities Module**

moseq2\_pca.util.clean\_frames (frames, medfilter\_space=None, gaussfilter\_space=None,
medfilter\_time=None, gaussfilter\_time=None, detrend\_time=None, tailfilter=None,
tail\_threshold=5)

Filters spatial/temporal noise from frames using Median and Gaussian filters, given kernel sizes for each respective requested filter.

#### Parameters:

- frames (3D numpy array) (frames to filter.)
- medfilter\_space (list) (median spatial filter kernel.)
- gaussfilter\_space (list) (gaussian spatial filter kernel.)
- medfilter\_time (list) (median temporal filter.)
- gaussfilter\_time (list) (gaussian temporal filter.)
- detrend\_time (int) (number of frames to lag for.)
- tailfilter (int) (size of tail-filter kernel.)
- tail\_threshold (int) (threshold value to use for tail filtering)

Returns: out (3D numpy array)

Return type: filtered frames.

moseq2\_pca.util.command\_with\_config (config\_file\_param\_name)

moseq2\_pca.util.gauss\_smooth (signal, win\_length=None, sig=1.5, kernel=None)
Perform Gaussian Smoothing on a 1D signal.

#### Parameters:

- signal (1d numpy array) (signal to perform smoothing)
- win\_length (int) (window\_size for gaussian kernel filter)
- sig (float) (variance of 1d gaussian kernel.)
- kernel (tuple) (kernel size to use for smoothing)

Returns: result (1d numpy array)

Return type: smoothed signal

moseq2\_pca.util.gaussian\_kernelld (n=None, sig=3) Get 1D gaussian kernel.

#### Parameters:

- n (int) (number of points to use.)
- sig (int) (variance of kernel to use.)

Returns: kernel (1d array)
Return type: 1D numpy kernel.

moseq2\_pca.util.get\_changepoints (scores, k=5, sigma=3, peak\_height=0.5, peak\_neighbors=1, baseline=True, timestamps=None)

Compute changepoints distribution and CP Curve.

#### Parameters:

- scores (3D numpy array) (nframes \* r \* c)
- k (int) (klags Lag to use for derivative calculation.)
- sigma (int) (Standard deviation of gaussian smoothing filter.)
- peak\_height (float) (user-defined peak Changepoint length.)
- peak neighbors (int) (number of peaks in the CP curve.)
- baseline (bool) (normalize data.)
- timestamps (array) (loaded timestamps.)

Returns: cps (2D numpy array) (array of values for CP curve) normed\_df (1D numpy array) (array of values for bar plot)

moseq2\_pca.util.get\_metadata\_path (h5file)

Return path within h5 file that contains the kinect extraction metadata.

Parameters: h5file (str) (path to h5 file.)

Returns: (str)

**Return type:** path to acquistion metadata within h5 file.

moseq2\_pca.util.get\_rps (frames, rps=600, normalize=True) Get random projections of frames.

#### Parameters:

- frames (2D or 3D numpy array) (Frames to get dimensions from.)
- rps (int) (Number of random projections.)
- normalize (bool) (indicates whether to normalize frames.)

Returns: rproj (2D or 3D numpy array)

**Return type:** Computed random projections with same shape as frames

moseq2\_pca.util.get\_timestamp\_path(h5file)

Return path within h5 file that contains the kinect timestamps

Parameters: h5file (str) (path to h5 file.)

Returns: (str)

Return type: path to metadata timestamps within h5 file

moseq2\_pca.util.initialize\_dask (nworkers=50, processes=1, memory='4GB', cores=1,
wall\_time='01:00:00', queue='debug', local\_processes=False, cluster\_type='local',
scheduler='distributed', timeout=10, cache\_path='/Users/aymanzeine/moseq2\_pca', \*\*kwargs)
Initialize dask client, cluster, workers, etc.

#### Parameters:

- nworkers (int) (number of dask workers to initialize)
- processes (int) (number of processes per worker)
- memory (str) (amount of memory to allocate to dask cluster)
- cores (int) (number of cores to use.)
- wall\_time (str) (amount of time to allow program to run)
- queue (str) (logging mode)
- local\_processes (bool) (indicate whether the processes are local)
- cluster type (str) (indicate what cluster to use)
- scheduler (str) (indicate what scheduler to use)
- timeout (int) (number of worker timeouts to allow)
- cache\_path (str or Pathlike) (path to store cached data)
- kwargs (extra keyward arguments)

Returns:

client (dask Client) (initialized Client) cluster (dask Cluster) (initialized Cluster) workers (dask Workers) (intialized workers or None if cluster\_type = 'local') cache (dask Chest) (initialized Chest (cache) object pointing to given cache path)

moseq2\_pca.util.insert\_nans (timestamps, data, fps=30)
Fills NaN values with 0 in timestamps.

#### Parameters:

- timestamps (1D array) (timestamp time-strs)
- data (1D array) (timestamp values)
- fps (int) (frames per second)

Returns: filled\_data (1D array) (filled missing timestamp values.) data\_idx (1D array) (indices of inserted 0s) filled timestamps (1D array) (filled timestamp-strs)

```
moseq2_pca.util.read_yaml (yaml_file)
```

Reads yaml file and returns dictionary representation of file contents.

Parameters: yaml\_file (str) (path to yaml file)

Returns: return\_dict (dict)
Return type: dict of yaml file contents

moseq2\_pca.util.recursive\_find\_h5s

(root\_dir='/Users/aymanzeine/Desktop/moseq/moseq2-pca/docs', ext='.h5',
yaml\_string='{}.yaml')

Recursively find h5 files, along with yaml files with the same basename

#### Parameters:

- root\_dir (str or os.Pathlike) (path to directory to start recursive search)
- ext (str) (extension to search for, e.g. .h5)
- yaml\_string (str) (a format to use to name yaml files)

Returns: h5s (list) (list of h5 file paths) dicts (list) (list of dicts containing metadata file contents)

yamls (list) (list of yaml file paths)

moseq2\_pca.util.recursively\_load\_dict\_contents\_from\_group (h5file, path)

Reads all contents from h5 and returns them in a nested dict object.

Parameters:

- h5file (str) (path to h5 file)
- path (str) (path to group within h5 file)

Returns: ans (dict)

Return type: dictionary of all h5 group contents

moseq2\_pca.util.select\_strel(string='e', size=(10, 10))

Selects Structuring Element Shape

Parameters:

- string (str) (e for Ellipse, r for Rectangle)
- size (tuple) (size of StructuringElement)

Returns: strel (cv2.StructuringElement)

**Return type:** returned StructuringElement with specified size.

moseq2\_pca.util.shutdown\_dask (scheduler)

Graceful shutdown dask scheduler. source:

https://github.com/dask/distributed/issues/1703#issuecomment-361291492

Parameters: scheduler (dask Scheduler) (scheduler to shutdown.)

Returns:

Return type: None

#### Visualization Module

moseq2\_pca.viz.changepoint\_dist (cps, headless=False)
Creates bar plot describing computed Changepoint Distribution.

Parameters:

- cps (np.ndarray) (changepoints to graph)
- headless (bool) (trim first element in PC list)

Returns: plt (plt.figure) (figure to save/graph) ax (plt.ax) (figure axis variable)

moseq2\_pca.viz.display\_components (components, cmap='gray', headless=False)
Creates grid of computed Principal Components.

Parameters:

- components (2D np.ndarray) (components to graph)
- cmap (str) (color map to use)
- headless (bool) (trim first element in PC list)

Returns: plt (plt.figure) (figure to save/graph) ax (plt.ax) (figure axis variable)

moseq2\_pca.viz.scree\_plot (explained\_variance\_ratio, headless=False) Creates Scree plot describing principal components.

Parameters:

 explained\_variance\_ratio (1D np.array) (explained variance ratio of each principal component)

headless (bool) (trim first element in PC list)

Returns: plt (plt.figure) figure to save/graph Return type:

#### Subpackages

#### moseq2\_pca.helpers package

#### Helpers - Data Module

moseq2\_pca.helpers.data.get\_pca\_yaml\_data (pca\_yaml) Reads PCA yaml file and returns metadata

> Parameters: pca\_yaml (str) (path to pca.yaml)

Returns: use\_fft (bool) (indicates whether to use FFT) clean\_params (dict) (dict of image filtering

parameters) mask\_params (dict) (dict of mask parameters)) missing\_data (bool)

(indicates whether to use mask params)

moseq2\_pca.helpers.data.load\_pcs\_for\_cp (pca\_file\_components, config\_data) Load computed Principal Components for Model-free Changepoint Analysis.

Parameters:

- pca file components (str) (path to pca h5 file to read PCs)
- config\_data (dict) (config parameters)

Returns:

pca components (str) (path to pca components) changepoint params (dict) (dict of relevant changepoint parameters) cluster (dask Cluster) (Dask Cluster object.) client (dask Client) (Dask Client Object) missing\_data (bool) (Indicates whether to use mask\_params) mask\_params (dict) (Mask parameters to use when computing CPs)

moseq2\_pca.helpers.data.setup\_cp\_command(input\_dir, config\_data, output\_dir, output\_file, output\_directory=None)

Helper function for changepoints\_wrapper to perform data-path existence checks.

#### Parameters:

- input dir (int) (path to directory containing all h5+yaml files)
- config\_data (dict) (dict of relevant PCA parameters (image filtering etc.))
- output\_dir (str) (path to directory to store PCA data)
- output file (str) (pca model filename)
- output directory (str) (alternative output dir)

config\_data (dict) (updated config\_data dict with the proper paths) pca\_file\_components (str) (path to trained pca file) pca\_file\_scores (str) (path to pca\_scores file) h5s (list) (list of relevant pca h5 files) yamls (list) (list of relevant pca metadata yaml files) save\_file (str) (path to save changepoints)

#### Helpers - Wrapper Module

moseq2\_pca.helpers.wrappers.apply\_pca\_wrapper (input\_dir, config\_data, output\_dir, output\_file, output\_directory=None, gui=False)

Wrapper function to obtain PCA Scores.

#### Parameters:

- input\_dir (int) (path to directory containing all h5+yaml files)
- config\_data (dict) (dict of relevant PCA parameters (image filtering etc.))
- output\_dir (str) (path to directory to store PCA data)
- output\_file (str) (pca model filename)
- output\_directory (str) (alternative output\_dir)
- qui (bool) (indicate GUI is running)

Returns: config\_data (dict)

Return type: updated config\_data variable to write back in GUI API

moseq2\_pca.helpers.wrappers.compute\_changepoints\_wrapper (input\_dir, config\_data, output\_dir, output\_file, gui=False, output\_directory=None)
Wrapper function to compute model-free (PCA based) Changepoints.

#### Parameters:

- input\_dir (int) (path to directory containing all h5+yaml files)
- config\_data (dict) (dict of relevant PCA parameters (image filtering etc.))
- output\_dir (str) (path to directory to store PCA data)
- output\_file (str) (pca model filename)
- output\_directory (str) (alternative output\_dir)
- gui (bool) (indicate GUI is running)

Returns: config\_data (dict)

Return type: updated config\_data variable to write back in GUI API

moseq2\_pca.helpers.wrappers.train\_pca\_wrapper (input\_dir, config\_data, output\_dir,
output\_file, output\_directory=None, gui=False)
Wrapper function to train PCA.

#### Parameters:

- input\_dir (int) (path to directory containing all h5+yaml files)
- config\_data (dict) (dict of relevant PCA parameters (image filtering etc.))
- output\_dir (str) (path to directory to store PCA data)
- output\_file (str) (pca model filename)
- output directory (str) (alternative output dir)
- gui (bool) (indicate GUI is running)

Returns: config\_data (dict)

Return type: updated config\_data variable to write back in GUI API

#### moseg2 pca.pca package

#### PCA - Utilties Module

moseq2\_pca.pca.util.apply\_pca\_dask (pca\_components, h5s, yamls, use\_fft, clean\_params, save\_file, chunk\_size, mask\_params, missing\_data, client, fps=30, gui=False)

"Apply" trained PCA on input frame data to obtain PCA Scores using Distributed Dask cluster.

#### Parameters:

- pca\_components (np.array) (array of computed Principal Components)
- h5s (list) (list of h5 files)
- yamls (list) (list of yaml files)
- use\_fft (bool) (indicate whether to use 2D-FFT)
- clean\_params (dict) (dictionary containing filtering options)
- save\_file (str) (path to pca\_scores filename to save)
- chunk\_size (int) (size of chunks to process)
- mask\_params (dict) (dictionary of masking parameters (if missing data))
- missing\_data (bool) (indicates whether to use mask arrays.)
- fps (int) (frames per second)

#### Returns:

#### Return type: None

moseq2\_pca.pca.util.apply\_pca\_local (pca\_components, h5s, yamls, use\_fft, clean\_params, save\_file, chunk\_size, mask\_params, missing\_data, fps=30)

"Apply" trained PCA on input frame data to obtain PCA Scores using local cluster/platform.

#### Parameters:

- pca\_components (np.array) (array of computed Principal Components)
- h5s (list) (list of h5 files)
- yamls (list) (list of yaml files)
- use\_fft (bool) (indicate whether to use 2D-FFT)
- clean\_params (dict) (dictionary containing filtering options)
- save\_file (str) (path to pca\_scores filename to save)
- chunk\_size (int) (size of chunks to process)
- mask\_params (dict) (dictionary of masking parameters (if missing data))
- missing\_data (bool) (indicates whether to use mask arrays.)
- fps (int) (frames per second)

#### Returns:

#### Return type: None

moseq2\_pca.pca.util.get\_changepoints\_dask (changepoint\_params, pca\_components, h5s, yamls, save\_file, chunk\_size, mask\_params, missing\_data, client, fps=30, pca\_scores=None, progress\_bar=False, gui=False)

Computes model-free changepoints using PCs and PC Scores on distributed dask cluster.

#### Parameters:

- changepoint\_params (dict) (dict of changepoint parameters)
- pca\_components (np.array) (computed principal components)
- h5s (list) (list of h5 files)
- yamls (list) (list of yaml files)
- save\_file (str) (path to save changepoint files)
- chunk\_size (int) (size of chunks to process in dask.)
- mask\_params (dict) (dict of missing\_data mask parameters.)
- missing\_data (bool) (indicate whether to use mask\_params)
- client (dask Client) (initialized Dask Client object)
- fps (int) (frames per second)
- pca\_scores (np.array) (computed principal component scores)
- progress\_bar (bool) (display progress bar)
- gui (bool) (indicate GUI use)

#### Returns:

Return type: None

moseq2\_pca.pca.util.mask\_data (original\_data, mask, new\_data)
Create a mask subregion given a boolean mask if missing data flag is used.

#### Parameters:

- original\_data (3d np.ndarray) (input frames)
- mask (3d boolean np.ndarray) (mask array)
- new\_data (3d np.ndarray) (frames to use)

Returns: output (3d np.ndarray)

Return type: masked data array

moseq2\_pca.util.train\_pca\_dask (dask\_array, clean\_params, use\_fft, rank, cluster\_type, client, workers, cache, mask=None, iters=10, recon\_pcs=10, min\_height=10, max\_height=100) Train PCA using dask arrays.

#### Parameters:

- dask\_array (dask array) (chunked frames to train PCA)
- clean\_params (dict) (dictionary containing filtering parameters)
- use\_fft (bool) (indicates whether to use 2d-FFT on images.)
- rank (int) (Matrix rank to use)
- cluster type (str) (indicates which cluster to use.)
- client (Dask.Client) (client object to execute dask operations)
- workers (int) (number of dask workers)
- cache (str) (path to cache directory)
- mask (dask array) (dask array of masked data if missing data parameter==True)
- iters (int) (number of SVD iterations)
- recon\_pcs (int) (number of PCs to reconstruct. (if missing\_data = True))
- min\_height (int) (minimum mouse height from floor in (mm))
- max\_height (int) (maximum mouse height from floor in (mm))

Returns: output\_dict (dict)

**Return type:** dictionary containing PCA training results.

# **Indices and tables**

- genindex
- modindex
- search

Index

--h5-mask-path <h5\_mask\_path>

moseq2-pca-apply-pca

line option

moseq2-pca-compute-changepoir command line option

comma

command

#### **Symbols**

-cluster-type <cluster\_type>

commhan@ath <h5\_path> --chunk-size <chunk size> moseq2-pca-apply-pca

line option

command

command

--klags <klags>

moseg2-pca-compute-changepoints command line option

--input-dir <input\_dir> moseq2-pca-train-pca line option

moseq2-pca-apply-pca line option

moseq2-pca-compute-changepoints

command line option moseq2-pca-train-pca

line option

--config-file <config\_file>

line option

moseq2-pca-train-pca

moseq2-pca-compute-changepoints command line option

moseq2-pca-train-pca line option

moseq2-pca-apply-pca --cores <cores> line option

moseg2-pca-compute-changepoints command line option

line option sk-cache-path <dask\_cache\_path>

--detrend-window <detrend\_window>

--dims <dims>

command line option --fill-gaps <fill\_gaps>

--fps <fps> moseq2-pca-apply-pca line option

--from-end

-gaussfilter-space <gaussfilter\_space>

--gaussfilter-time <gaussfilter\_time>

moseq2-pca-train-pca command line option

command line option

moseq2-pca-compute-changepoints command line option moseq2-pca-apply-pca command

line option moseq2-pca-compute-changepoints

moseq2-pca-apply-pca

line option

command line option moseq2-pca-train-pca command line option

moseq2-pca-compute-changepoints command line option moseq2-pca-apply-pca --local-processes <local\_processes>

moseq2-pca-train-pca command line option

moseq2-pca-tr --mask-height-threshold <mask\_height\_threshold> command line --mask-threshold <mask\_threshold> moseq2-pca-train-pca

command line option moseq2-pca-train-pca

command line option command --medfilter-space <medfilter\_space>

moseq2-pca-train-pca --medfilter-time <medfilter time> command

--max-height <max height>

--memory <memory> line option moseq2-pca-apply-pca command line option

moseg2-pca-compute-changepoints command line option

moseq2-pca-train-pca command

line option moseq2-pca-apply-pca command line option --missing-data

moseq2-pca-compute-changepoints 

> moseq2-pca-apply-pca command line option -- neighbors < neighbors >

command --nworkers <nworkers>

moseq2-pca-compute-changepoints command line option

moseq2-pca-clip-scores command line option moseq2-pca-train-pca

moseq2-pca-train-pca command line option

command line option moseq2-pca-apply-pca command

moseq2-pca-compute-changepoints command line option

moseq2-pca-train-pca command line option

command line option

moseq2-pca-train-pca -min-height <min\_height> command line option moseq2-pca-train-pca

> command line option moseq2-pca-compute-changepoints command line option

moseq2-pca-apply-pca command line option

moseg2-pca-compute-changepoints command line option

moseq2-pca-train-pca command line option

<output_d< td=""><td>dir&gt; moseq2-pca- line option</td><td>apply-pca c</td><td>command-use-</td><td>-fft</td><td></td><td></td><td></td><td></td></output_d<>	dir> moseq2-pca- line option	apply-pca c	command-use-	-fft				
			<b>igeponis</b> lts	<visua< td=""><td>lize_re</td><td>sults&gt; n</td><td></td><td>•</td></visua<>	lize_re	sults> n		•
	moseq2-pca- line option	train-pca d	command					rain-pca c
<output_f< td=""><td></td><td>a-apply-pca -</td><td><del>comh</del>atá<sup>me</sup></td><td><wall_< td=""><td>time&gt;</td><td></td><td>a-apply-pca</td><td>command</td></wall_<></td></output_f<>		a-apply-pca -	<del>comh</del> atá <sup>me</sup>	<wall_< td=""><td>time&gt;</td><td></td><td>a-apply-pca</td><td>command</td></wall_<>	time>		a-apply-pca	command
		•	angepoints					angepoints
	moseq2-pca	•	command				a-train-pca	command
file <pca< td=""><td>_file&gt; moseq2-</td><td></td><td>-c</td><td></td><td></td><td></td><td>ca comma</td><td>nd</td></pca<>	_file> moseq2-		-c				ca comma	nd
<pca_file< td=""><td></td><td></td><td></td><td>epoints</td><td></td><td></td><td>•</td><td>nts</td></pca_file<>				epoints			•	nts
<pca_file< td=""><td></td><td></td><td>• •</td><td>ts</td><td></td><td></td><td>a comma</td><td>nd</td></pca_file<>			• •	ts			a comma	nd
<pca_path< td=""><td>n&gt; moseq2-pca-ap line option</td><td>ply-pca con</td><td>nmand <sub>-d</sub></td><td></td><td></td><td></td><td>ca comma</td><td>nd</td></pca_path<>	n> moseq2-pca-ap line option	ply-pca con	nmand <sub>-d</sub>				ca comma	nd
			epoints				•	nts
<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	es> moseq2-pca-a line option	pply-pca co	ommand				a comma	nd
			gepoints_i				ca comma	nd
	moseq2-pca-tr line option	ain-pca co	ommand					nts
-	moseq2-pca-apply-pline option	oca comma	nd				a comma	nd
			nts <sub>-k</sub>				• •	nts
	moseq2-pca-train-p line option	ca comma	nd -m				ca comma	nd
k <rank></rank>	command li	ne option						nts
_	comma	nd line option					a comma	nd
	command line optio	n	-n				ca comma	nd
		command li	ne option				•	nts
		command line	e option				a comma	nd
	command line	option	-0		moseq2-	-pca-apply-p	ca comma	nd
· CIMEOUL/	line option				moseq2-	-pca-comput	•	nts
		command line option		moseq2-pca-train-p		-pca-train-pc		nd
	moseq2-pca-trair line option	n-pca comr	mand		ine optio	ווע		
	<pre><output_f <pca="" <pca_file="" <pca_path="" <pre="" file=""><pre><pre><pre>&lt; <rank></rank></pre> <coutput_f <pca_file="" <pca_path="" <pre=""><pre><pre><pre>&lt; <pre>&lt; <pre>&lt; <ipse><coutput <pre=""><coutput <pre=""><pre><coutput <pre=""><pre><coutput <pre=""><pre></pre> <pre><coutput <pre=""><coutput <pre=""><pre><coutput <pre=""><pre></pre> <pre><coutput <pre=""><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></coutput></pre></coutput></pre></coutput></coutput></pre></coutput></pre></coutput></pre></coutput></coutput></ipse></pre></pre></pre></pre></pre></coutput_f></pre></pre></output_f></pre>	line option moseq2-pca- command line moseq2-pca- line option <pre></pre>	line option   moseq2-pca-comptient   moseq2-pca-train-pca   command line option   moseq2-pca-train-pca   line option   moseq2-pca-train-pca   line option   moseq2-pca-compute-cha   command line option   moseq2-pca-compute-cha   command line option   moseq2-pca-train-pca   line option   moseq2-pca-compute   command line option   moseq2-pca-train-pca   command line option   command line option   moseq2-pca-train-pca   command line option   command line option   moseq2-pca-train-pca   command line option   command line option   sigma   moseq2-pca-compute-changepoin   command line option   shape   cailfilter_shape   moseq2-pca-command line   command   command line   command   command line   command   com	line option   moseq2-pca-compide   command   moseq2-pca-train-pca   command   moseq2-pca-compute-changepoints   command   moseq2-pca-compute-changepoint   command   moseq2-pca-compute-changepoint   command   moseq2-pca-compute-changepoint   command   moseq2-pca-compute-changepoint   command   moseq2-pca-compute-changepoint   command   moseq2-pca-compute-changepoints   command   moseq2-pca-compute-changepoints   command   moseq2-pca-compute-changepoints   command   moseq2-pca-compute-changepoints   command   moseq2-pca-train-pca   command   com	line option   moseq2-pca-compute-valaringepoints  time option   moseq2-pca-compute-changepoints   command line option   moseq2-pca-train-pca   command line option   comma	line option	Ine option	Ine option

-p	moseq2-pca-app	oly-pca	command	get_metadata_path() (in module moseq2_pca.util)				
	line option moseg2-pca-cor	nnute-ch	angenoints	get_pca_yaml_data() (in module moseq2_pca.helpers.data)				
	moseq2-pca-compute-changepoints command line option		angopomio	get_rps() (in module moseq2_pca.util)				
	moseq2-pca-trai	n-pca	command	get_timestamp_path() (in module moseq2_pca.util)				
-d	moseq2-pca-appline option	oly-pca	command	I				
	moseq2-pca-cor	npute-changepoints		initialize_dask() (in module moseq2_pca.util)				
	moseq2-pca-trai		command	insert_nans() (in module moseq2_pca.util)				
-s	line option moseq2-pca-cor	•	angepoints	L load_pcs_for_cp() (in module				
-w	command line opmoseq2-pca-app		command	moseq2_pca.helpers.data)				
	line option			M				
	moseq2-pca-cor command line op		angepoints	mask_data() (in module moseq2_pca.pca.util)				
	moseq2-pca-trai	n-pca	command	moseq2-pca-apply-pca command line optionchunk-size <chunk size=""></chunk>				
	што ориот			cluster-type <cluster_type></cluster_type>				
A				config-file <config_file></config_file>				
apply_pca_comm	nand() (in module	moseq2_	_pca.gui)	cores <cores></cores>				
apply_pca_dask(	) (in module mose	eq2_pca.	pca.util)	dask-cache-path <dask_cache_path></dask_cache_path>				
apply_pca_local(	) (in module mose	eq2_pca. <sub> </sub>	pca.util)	detrend-window <detrend_window></detrend_window>				
apply_pca_wrapp		in	module	fill-gaps <fill_gaps></fill_gaps>				
moseq2_pca.help	bers.wrappers)			fps <fps></fps>				
С				h5-mask-path <h5_mask_path></h5_mask_path>				
changepoint_dist	() (in module mos	eq2_pca	.viz)	h5-path <h5_path></h5_path>				
clean_frames() (in module moseq2_pca.util)			)	input-dir <input_dir></input_dir>				
CLIP_SAMPLES				memory <memory></memory>				
	-clip-scores comn		•	nworkers <nworkers></nworkers>				
command_with_c		•	_, ,	output-dir <output_dir></output_dir>				
compute_change moseq2_pca.gui)	•	() (in	module	<pre>output-file <output_file>pca-file <pca_file></pca_file></output_file></pre>				
compute_change moseq2_pca.help		(in	module	pca-path <pca_path></pca_path>				
5				processes <pre><pre>cesses&gt;</pre></pre>				
D		_		queue <queue></queue>				
display_compone	ents() (in module r	moseq2_	pca.viz)	timeout <timeout></timeout>				
G				wall-time <wall_time> -c</wall_time>				
gauss_smooth()	(in module moseq	∣2_pca.ut	il)	-d				
gaussian_kernel1	Id() (in module mo	oseq2_po	ca.util)	-i				
get_changepoints	s() (in module mo	seq2_pca	a.util)	-m				
get_changepoints moseq2_pca.pca	•	(in	module	-n				

-0

-p -q -q -s moseq2-pca-clip-scores command line option moseq2-pca-train-pca command line option --from-end --chunk-size <chunk size> CLIP SAMPLES --cluster-type <cluster type> PCA FILE --config-file <config\_file> moseg2-pca-compute-changepoints command line --cores <cores> option --dask-cache-path <dask\_cache\_path> --chunk-size <chunk\_size> --gaussfilter-space <gaussfilter\_space> --cluster-type <cluster\_type> --gaussfilter-time <gaussfilter time> --config-file <config\_file> --input-dir <input\_dir> --cores <cores> --local-processes < local processes> --dask-cache-path <dask\_cache\_path> --mask-height-threshold < mask height threshold> --dims <dims> --mask-threshold <mask threshold> --fps <fps> --max-height <max\_height> --h5-mask-path <h5\_mask\_path> --medfilter-space <medfilter space> --h5-path <h5\_path> --medfilter-time <medfilter time> --input-dir <input\_dir> --memory <memory> --klags <klags> --min-height <min height> --memory <memory> --missing-data --neighbors <neighbors> --missing-data-iters <missing\_data\_iters> --nworkers <nworkers> --nworkers < nworkers> --output-dir <output\_dir> --output-dir <output dir> --output-file <output\_file> --output-file <output file> --pca-file-components <pca\_file\_components> --processes cesses> --pca-file-scores <pca\_file\_scores> --queue <queue> --pca-path <pca\_path> --rank <rank> --processes cesses> --recon-pcs <recon\_pcs> --queue <queue> --tailfilter-shape <tailfilter shape> --sigma <sigma> --tailfilter-size <tailfilter size> --threshold <threshold> --timeout <timeout> --timeout <timeout> --use-fft --visualize-results <visualize results> --visualize-results <visualize results> --wall-time <wall\_time> --wall-time <wall time> -C -C -d -d -i -i -k -m -m -n -n -0 -0 -p -p

```
-q
-w
moseq2_pca.gui (module)
moseq2_pca.helpers.data (module)
moseq2_pca.helpers.wrappers (module)
moseq2_pca.pca.util (module)
moseq2_pca.util (module)
moseq2_pca.viz (module)

P
PCA_FILE
moseq2-pca-clip-scores command line option
```

# R read\_yaml() (in module moseq2\_pca.util) recursive\_find\_h5s() (in module moseq2\_pca.util) recursively\_load\_dict\_contents\_from\_group() (in module moseq2\_pca.util)

scree\_plot() (in module moseq2\_pca.viz)
select\_strel() (in module moseq2\_pca.util)
setup\_cp\_command() (in module moseq2\_pca.helpers.data)

## shutdown\_dask() (in module moseq2\_pca.util)

S

T

train\_pca\_command() (in module moseq2\_pca.gui)
train\_pca\_dask() (in module moseq2\_pca.pca.util)
train\_pca\_wrapper() (in module
moseq2\_pca.helpers.wrappers)

# **Python Module Index**

#### m

moseq2\_pca.gui
moseq2\_pca.helpers.data
moseq2\_pca.helpers.wrappers
moseq2\_pca.pca.util
moseq2\_pca.util
moseq2\_pca.viz