
plumitas Documentation

Release 0.0.1.dev2

uwprg

Apr 09, 2018

Contents:

1	plumitas	1
1.1	API	1
1.2	Testing	5
2	Indices and tables	7
	Python Module Index	9

1.1 API

1.1.1 plumitas.core

```
class plumitas.core.GridParameters(sigma, grid_min, grid_max)
```

Bases: tuple

grid_max

Alias for field number 2

grid_min

Alias for field number 1

sigma

Alias for field number 0

```
class plumitas.core.MetaDProject(colvar, hills, input_file=None, bias_type='MetaD',  
                                multi=False)
```

Bases: *plumitas.core.SamplingProject*

```
potential_of_mean_force(collective_variables, mintozero=True, xlabel='CV', xlim=None,  
                        ylim=None)
```

Create PMF plot for one or several collective variables.

Parameters

- **collective_variables** (*list*) – List of CVs you’d like to plot. These should be supplied in the form of a list of column names, or an instance of `pd.Index` using `df.columns`
- **mintozero** (*bool*, *True*) – Determines whether or not to shift PMF so that the minimum is at zero.
- **xlabel** (*string*) – Label for the x axis.
- **xlim** (*tuple/list*) – Limits for x axis in plot (i.e. `[x_min, x_max]`).
- **ylim** (*tuple/list*) – Limits for y axis in plot (i.e. `[y_min, y_max]`).

Returns axes

Return type matplotlib.AxesSubplot

reconstruct_bias_potential ()

weight_frames (*temp=None*)

Assign frame weights using the Torrie and Valleau reweighting method from a quasi-static bias potential.
Adds a 'weight' column to self.colvar.

Parameters *temp* (*float, None*) – If self.temp exists, the user does not need to supply a temp because self.temp will take it's place anyway. If self.temp does not exist, temp must be supplied in the method call or an error will be printed with no further action.

Returns

Return type None

class plumitas.core.PBMetaDProject (*colvar, hills, input_file=None, bias_type='PBMetaD', multi=False*)

Bases: *plumitas.core.SamplingProject*

potential_of_mean_force (*collective_variables, mintozero=True, xlabel='CV', xlim=None, ylim=None*)

Create PMF plot for one or several collective variables.

Parameters

- **collective_variables** (*list*) – List of CVs you'd like to plot. These should be supplied in the form of a list of column names, or an instance of pd.Index using df.columns
- **mintozero** (*bool, True*) – Determines whether or not to shift PMF so that the minimum is at zero.
- **xlabel** (*string*) – Label for the x axis.
- **xlim** (*tuple/list*) – Limits for x axis in plot (i.e. [x_min, x_max]).
- **ylim** (*tuple/list*) – Limits for y axis in plot (i.e. [y_min, y_max]).

Returns axes

Return type matplotlib.AxesSubplot

reconstruct_bias_potential ()

weight_frames (*temp=None*)

Assign frame weights using the Torrie and Valleau reweighting method from a quasi-static bias potential.
Adds a 'weight' column to self.colvar.

Parameters *temp* (*float, None*) – If self.temp exists, the user does not need to supply a temp because self.temp will take it's place anyway. If self.temp does not exist, temp must be supplied in the method call or an error will be printed with no further action.

Returns

Return type None

class plumitas.core.SamplingProject (*colvar, hills, input_file=None, bias_type=None, multi=False*)

Bases: object

Base class for management and analysis of an enhanced sampling project.

free_energy_surface (*x, y, weight=None, bins=50, clim=None, xlim=None, ylim=None, energy_cut=50*)

Create a 2D FES from a COLVAR file with frame weights.

Parameters

- **x** (*string*) – Name of one of the CVs (column name from df).
- **y** (*string*) – Name of one of the CVs (column name from df).
- **bins** (*int*) – Number of bins in each dimension to segment histogram.
- **temp** (*float*) – Temperature of simulation which generated Plumed file.
- **weight** (*str*) – Name of static bias column.
- **clim** (*int*) – Maximum free energy (in kJ/mol) for color bar.
- **xlim** (*tuple/list*) – Limits for x axis in plot (i.e. [x_min, x_max]).
- **ylim** (*tuple/list*) – Limits for y axis in plot (i.e. [y_min, y_max]).
- **energy_cut** (*float*) – Cut off to exclude very high free energy values from histogram to help visualization.

Returns axes**Return type** matplotlib.AxesSubplot**get_bias_params** (*input_file, bias_type*)

Method to grab bias parameters incase user forgot to supply plumed.dat or input file wasn't automatically identified in the working directory.

Parameters

- **input_file** (*string*) – Relative path to PLUMED input file. Most commonly called plumed.dat.
- **bias_type** (*string*) – String associated with biasing method used for enhanced sampling. Currently only “MetaD” and “PBMetaD” supported (case insensitive).

Returns**Return type** None**plumitas.core.get_float** (*string*)

Helper function in case grid boundaries are pi.

Parameters **string** (*string*) – Parameter string.**Returns** **number****Return type** float**plumitas.core.load_project** (*colvar='COLVAR', hills='HILLS', method=None, **kwargs*)

High-level function to read in all files associated with a Plumed enhanced sampling project. ****kwargs** supplied since different project types will be instantiated with different arguments.

Parameters

- **colvar** (*string*) – Name of the COLVAR file to read in.
- **hills** (*string*) – Name of the HILLS file to read in.
- **method** (*string*) – Name of enhanced sampling method used to bias the simulation. Supported methods will include “MetaD”, “PBMetaD”, and others. If the default None value is passed, plumitas will try to create

Returns **project** – Project base class, or subclass if ‘method’ is specified.**Return type** plumitas.SamplingProject

`plumitas.core.parse_bias(filename='plumed.dat', bias_type=None)`

Function that takes experimental data and gives us the dependent/independent variables for analysis.

Parameters

- **filename** (*string*) – Name of the plumed input file used for enhanced sampling run.
- **bias_type** (*string*) – Name of bias method used during

Returns **bias_args** – Dictionary of key: value pairs from the plumed.dat file. Will facilitate automatic reading of parameter reading once core.SamplingProject class is implemented.

Return type dict

`plumitas.core.read_colvar(filename='COLVAR', multi=0, unbiased=False)`

Function that takes experimental data and gives us the dependent/independent variables for analysis.

Parameters

- **filename** (*string*) – Name of the COLVAR file to read in.
- **multi** (*int*) – Tells the method to read 1 or more COLVAR files. Default falsy value (0) means read only 1 file.
- **unbiased** (*bool*) – If True, adds a 'weight' column of all 1s.

Returns **df** – CVs and bias as columns, time as index.

Return type Pandas DataFrame

`plumitas.core.read_hills(filename='HILLS')`

Function that takes experimental data and gives us the dependent/independent variables for analysis.

Parameters **filename** (*string*) – Name of the COLVAR file to read in.

Returns **df** – CVs and bias as columns, time as index.

Return type Pandas DataFrame

`plumitas.core.sum_hills(grid_points, hill_centers, sigma, periodic=False)`

Helper function for building static bias functions for SamplingProject and derived classes.

Parameters

- **grid_points** (*ndarray*) – Array of grid values at which bias potential should be calculated.
- **hill_centers** (*ndarray*) – Array of hill centers deposited at each bias stride.
- **sigma** (*float*) – Hill width for CV of interest.
- **periodic** (*bool*) – True if CV is periodic, otherwise False.

Returns **bias_grid** – Value of bias contributed by each hill at each grid point.

Return type ndarray

1.1.2 plumitas.analysis

1.1.3 plumitas.visualization

1.2 Testing

1.2.1 plumitas.tests

plumitas.tests.test_plumitas

```
plumitas.tests.test_plumitas.test_metad()  
    Testing function to convert of COLVAR file to pandas DataFrame.  
plumitas.tests.test_plumitas.test_no_bias()  
plumitas.tests.test_plumitas.test_pbmetad()  
    Testing function to convert of COLVAR file to pandas DataFrame.
```


CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

p

`plumitas.core`, [1](#)

`plumitas.tests.test_plumitas`, [5](#)

F

`free_energy_surface()` (plumitas.core.SamplingProject method), 2

G

`get_bias_params()` (plumitas.core.SamplingProject method), 3

`get_float()` (in module plumitas.core), 3

`grid_max` (plumitas.core.GridParameters attribute), 1

`grid_min` (plumitas.core.GridParameters attribute), 1

`GridParameters` (class in plumitas.core), 1

L

`load_project()` (in module plumitas.core), 3

M

`MetaDProject` (class in plumitas.core), 1

P

`parse_bias()` (in module plumitas.core), 3

`PBMetaDProject` (class in plumitas.core), 2

`plumitas.core` (module), 1

`plumitas.tests.test_plumitas` (module), 5

`potential_of_mean_force()` (plumitas.core.MetaDProject method), 1

`potential_of_mean_force()` (plumitas.core.PBMetaDProject method), 2

R

`read_colvar()` (in module plumitas.core), 4

`read_hills()` (in module plumitas.core), 4

`reconstruct_bias_potential()` (plumitas.core.MetaDProject method), 2

`reconstruct_bias_potential()` (plumitas.core.PBMetaDProject method), 2

S

`SamplingProject` (class in plumitas.core), 2

`sigma` (plumitas.core.GridParameters attribute), 1

`sum_hills()` (in module plumitas.core), 4

T

`test_metad()` (in module plumitas.tests.test_plumitas), 5

`test_no_bias()` (in module plumitas.tests.test_plumitas), 5

`test_pbmetad()` (in module plumitas.tests.test_plumitas), 5

W

`weight_frames()` (plumitas.core.MetaDProject method), 2

`weight_frames()` (plumitas.core.PBMetaDProject method), 2