

---

**MD DaVis**

*Release 0.4.1*

**Dibyajyoti Maity**

**Apr 03, 2024**



# CONTENTS

<b>1</b>	<b>Documentation</b>	<b>3</b>
1.1	Introduction . . . . .	3
1.2	Install . . . . .	7
1.3	Uninstall . . . . .	11
1.4	User Guide . . . . .	11
1.5	Tutorials . . . . .	32
1.6	Graphical User Interface . . . . .	38
1.7	Command Line Interface . . . . .	46
1.8	Reference . . . . .	47
<b>2</b>	<b>Indices and tables</b>	<b>59</b>
	<b>Index</b>	<b>61</b>



MD DaVis is a tool and Python 3 package to easily create helpful interactive visualizations to analyze and compare multiple molecular dynamics simulations of proteins.



## DOCUMENTATION

## 1.1 Introduction

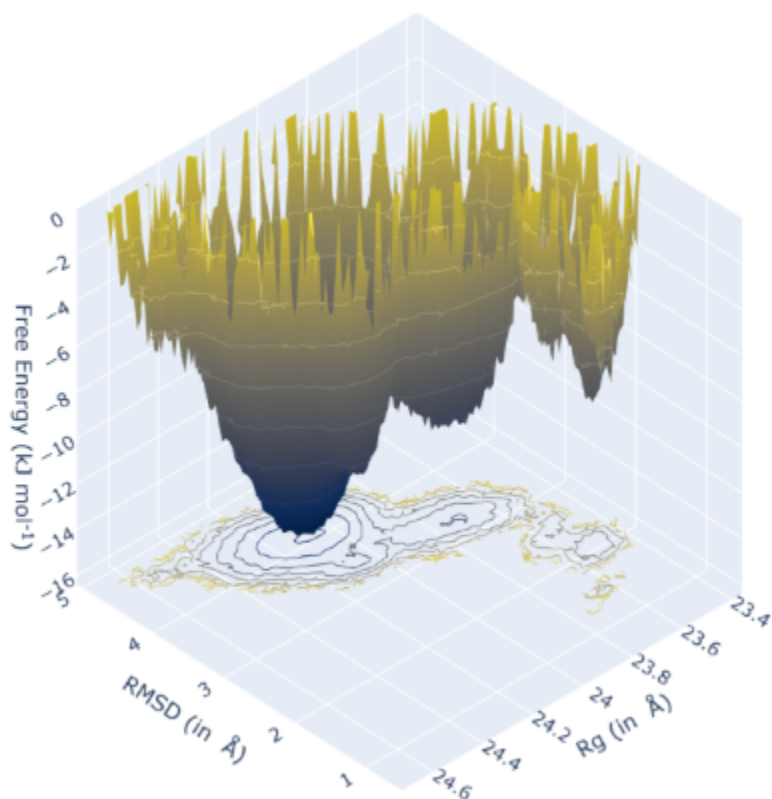
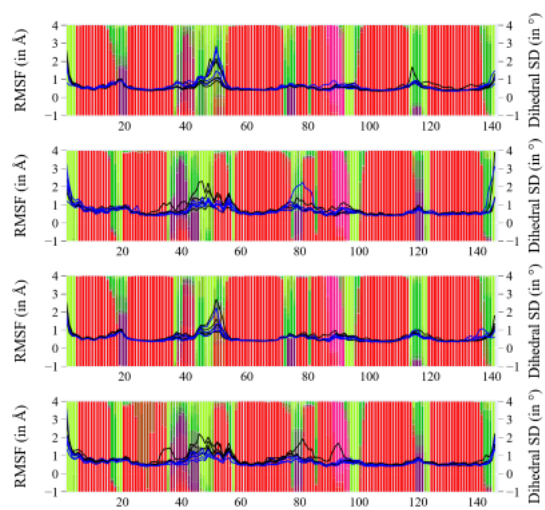
Molecular dynamics (MD) simulations are indispensable for gaining atomistic insights into the biomolecular function. MD of increasingly larger proteins has become accessible to researchers with recent advancements in computing and MD algorithms. However, the analysis of MD trajectories remains tedious. MD DaVis (Molecular Dynamics Data Visualizer) is a tool and Python 3 package to perform comparative data analysis of MD trajectories of similar proteins or the same protein under different conditions.

There are many *MD analysis tools*. However, the output from most has to be visualized using another plotting library requiring a significant amount of coding. The MD DaVis package provides the *md-davis* command-line tool and *md-davis-gui* GUI tool to create helpful interactive visualizations easily. The tool can increase the productivity of researchers simulating protein using MD and make the analysis of such simulations accessible to everyone.

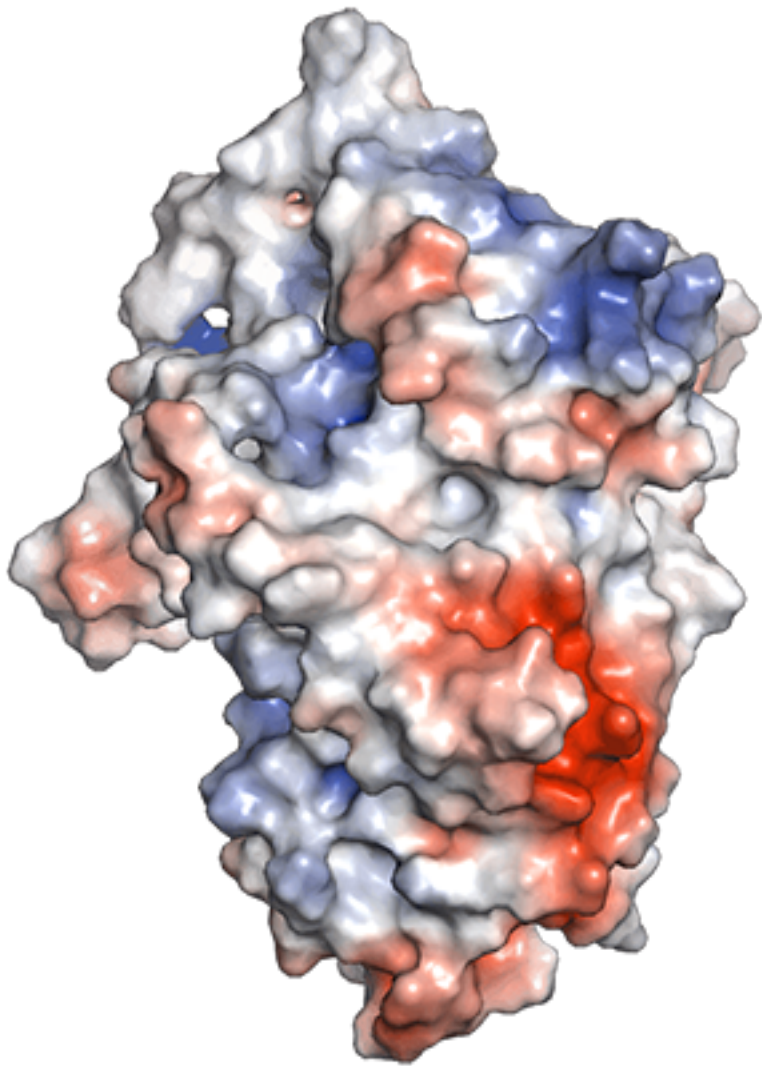
### 1.1.1 Features of MD DaVis

Click on each panel for more details.

*Free Energy Landscapes*

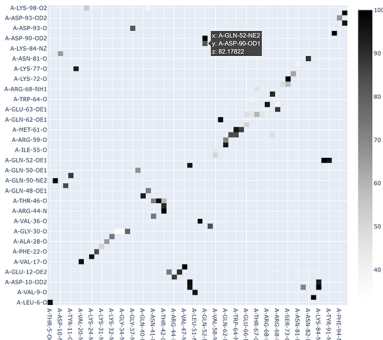
*Residue Properties Plot**Surface Electrostatics*





Electric Field Dynamics

H-bond / Contact Matrix



### 1.1.2 Interfacing MD DaVis to Other Analysis Tools

MD DaVis does not implement analysis tools and functions available elsewhere. Presently, MD DaVis natively supports analysis performed using GROMACS tools. However, other analysis tools may be used as well, depending on the format of the MD trajectory. The input to most MD DaVis commands is text files with two columns; the first is time or residue number, and the second is the value, e.g., RMSD, Rg, RMSF, SASA.

```
# gmx rms is part of G R O M A C S:
#
# Gromacs Runs On Most of All Computer Systems
#
@ title "RMSD"
@ xaxis label "Time (ps)"
@ yaxis label "RMSD (nm)"
@TYPE xy
@ subtitle "System after lsq fit to System"
  0.0000000  0.0000031
 10.0000000  0.0677038
 20.0000000  0.0837483
 30.0000000  0.0894995
```

So, interfacing other analysis tools to MD DaVis is as simple as creating these text files, easily accomplished with the Python packages `mdtraj` or `mdanalysis`. However, using Python to calculate these properties is significantly slower than using binary tools like those bundled with MD software. Therefore, we recommend using such binary tools whenever possible. A package that can read the trajectory files may be used for the remaining calculations, like `mdtraj`, which natively accepts binary trajectory files from AMBER, Desmond, GROMACS, LAMMPS, and NAMD. The dihedral angles calculation in MD DaVis is done with `mdtraj`, and the electrostatics are calculated on a set of PDB files that all MD engines can export. Therefore, MD DaVis does not exclusively rely on GROMACS for these calculations. However, the H-bond/Contact visualization and secondary structure visualization rely on the output from the GROMACS tools `hbond` and `do_dssp`, respectively. That is why we recommended converting the trajectory to GROMACS format with `mdtraj`.

### 1.1.3 Molecular Dynamics Analysis Tools

The following is a non-exhaustive list of tools for analysis of MD simulations:

- AMBER (Case et al., 2021)
- Bio3d (Grant et al., 2006)
- CPPTRAJ (Roe and Cheatham, 2013)
- MDAnalysis (Michaud-Agrawal et al., 2011)
- GROMACS (Berendsen et al., 1995)
- GROMOS++ (Eichenberger, et al., 2011)
- MD-TASK (Brown et al., 2017)
- MDplot (Margreitter and Oostenbrink, 2017)
- MDTraj (McGibbon et al., 2015)
- MDTRA (Popov et al., 2013)
- TRAVIS (Brehm and Kirch-ner, 2011)

## References

1. Berendsen, H. J. C. et al. (1995) GROMACS: A message-passing parallel molecular dynamics implementation. *Computer Physics Communications*, 91, 43-56.
2. Brehm, M. and Kirchner, B. (2011) TRAVIS - A Free Analyzer and Visualizer for Monte Carlo and Molecular Dynamics Trajectories. *J. Chem. Inf. Model.*, 51, 2007-2023.
3. Brown, D. K. et al. (2017) MD-TASK: a software suite for analyzing molecular dynamics trajectories. *Bioinformatics*, 33, 2768-2771.
4. Case, D. A. et al. (2021) Amber 2021 University of California, San Francisco.
5. Eichenberger, A. P. et al. (2011) GROMOS++ Software for the Analysis of Biomolecular Simulation Trajectories. *J. Chem. Theory Comput.*, 7, 3379-3390.
6. Grant, B. J. et al. (2006) Bio3d: an R package for the comparative analysis of protein structures. *Bioinformatics*, 22, 2695-2696.
7. Margreitter, C. and Oostenbrink, C. (2017) MDplot: Visualise Molecular Dynamics. *The R Journal*, 9, 164-186.
8. McGibbon, R. T. et al. (2015) MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophys J*, 109, 1528-1532.
9. Michaud-Agrawal, N. et al. (2011) MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. *J. Comput. Chem.*, 32, 2319-2327.
10. Popov, A. V. et al. (2013) MDTRA: A molecular dynamics trajectory analyzer with a graphical user interface. *Journal of Computational Chemistry*, 34, 319-325.
11. Roe, D. R., and Cheatham, T. E. (2013) PTRAJ and CPPTRAJ: Software for Processing and Analysis of Molecular Dynamics Trajectory Data. *J. Chem. Theory Comput.*, 9, 3084-3095.

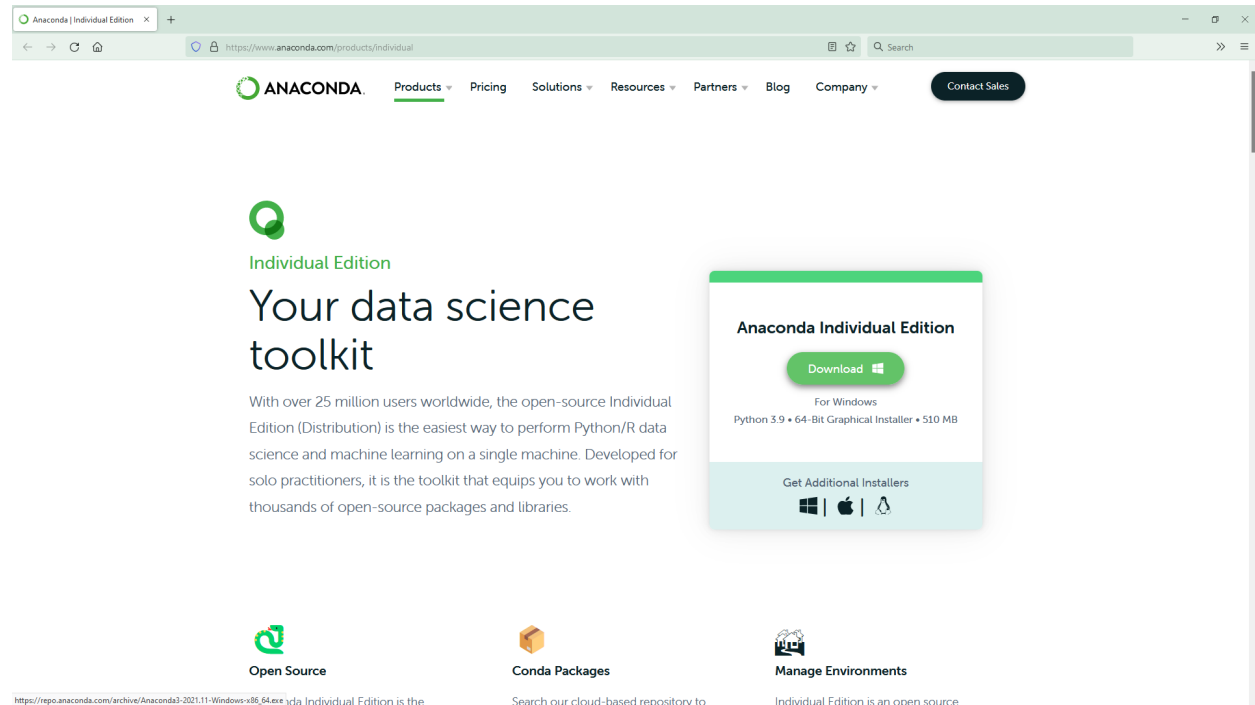
## 1.2 Install

### 1.2.1 System Requirements

- A 64-bit operating system (We have tested the installation on Ubuntu 20.04, Windows 10, and Windows 11, but any modern operating system should work.)
- An [Anaconda](#) or [Miniconda](#) Python distribution (3.9 or greater)
- An internet connection

### 1.2.2 Installation Instructions

**Step 1:** Download and install Anaconda (<https://www.anaconda.com/products/individual>) or Miniconda (<https://docs.conda.io/en/latest/miniconda.html>) Python distribution.



**Step 2:** Create the `conda` environment with MD DaVis dependencies. Open a Terminal or Anaconda prompt (on Windows) and issue the following command:

```
conda env create djmaity/md-davis
```

This creates a `conda` environment called `md-davis` with all required dependencies.

**Step 3:** Activate the `md-davis` environment with:

```
conda activate md-davis
```

**Step 4:** Install MD DaVis in this environment with `pip`:

```
pip install md-davis
```

**Step 5:** Obtain the *external dependencies* as stated below.

### 1.2.3 Installation for Development

**Step 1:** Download and install Anaconda (<https://www.anaconda.com/products/individual>) or Miniconda (<https://docs.conda.io/en/latest/miniconda.html>) Python distribution.

**Step 2:** Create the `conda` environment with MD DaVis dependencies. In this case, you should use:

```
conda env create djmaity/md-davis-dev
```

This would create a `conda` environment called `md-davis-env` with the core dependencies and install additional libraries for packaging, documentation, and linting. You can also change the name of the `conda` environment by modifying the command above:

```
conda env create djmaity/md-davis-dev --name my_env
```

This will name the conda environment to `my_env`. Make sure that there are no other conda environments with the same name. The `djmaity/md-davis-dev` in the above commands uses the environment files uploaded to <https://anaconda.org/djmaity/environments>. These are the same files `environment.yml` and `dev_environment.yml` in the source code, which may also be used to create the conda environment and install the dependencies:

```
conda env create --file dev_environment.yml
```

See <https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html> to learn more about conda environments.

**Note:** We recommend using the environment files over installing the dependencies individually because installing all the dependencies together allows conda to resolve the appropriate version of packages to install and avoid package conflicts.

**Step 3:** Activate the conda environment with:

```
conda activate md-davis-dev
```

Remember to change `md-davis-dev` to the appropriate environment name if you modify it.

**Step 4:** Obtain a copy of the source code by cloning the public repository:

```
git clone git@github.com:djmaity/md-davis.git
```

Or, download the [tarball](#):

```
curl -OL https://github.com/djmaity/md-davis/tarball/master
```

**Step 5:** Once you have a copy of the source, it can be installed with:

```
pip install path/to/source/code
```

The path to the directory containing the `setup.py` file has to be provided in the command above. You may want to install it as an editable package with:

```
pip install -e path/to/source/code
```

This will allow changes to the source code to be immediately reflected in the installed package.

Or, go to the directory containing the `setup.py` file and use:

```
python setup.py install
```

You can also install the development version directly with:

```
pip install https://github.com/djmaity/md-davis/archive/master.zip
```

**Step 6:** Obtain the *external dependencies* as stated below.

## 1.2.4 External Dependencies

### GROMACS

Currently, most analyses have to be performed with GROMACS, and the output is provided to MD DaVis for visualization. We have successfully used MD DaVis on simulations and analysis performed with various GROMACS versions from 5.1.4 to 2021.3. Other analysis tools may be used as long as the input to MD DaVis can be appropriately formatted. See *Interfacing MD DaVis to Other Analysis Tools*.

### DSSP

The secondary structure of a trajectory is calculated by the GROMACS tool `do_dssp`, which requires [DSSP](#). The latest and only available version of [DSSP](#) is 2.3.0. The executable is called `mkdssp` now. Help GROMACS find it by any of the following means:

- Rename `mkdssp` to `dssp`
- Make a symlink called `dssp` to `mkdssp`
- Set the DSSP environment

### PyMOL

You do not need to obtain PyMOL separately if you use the installation methods outlined above using `conda`. Unfortunately, PyMOL is not available in the [python package index](#). Therefore, it cannot be automatically installed with `pip`. However, Open-Source PyMOL and Commercial/Educational PyMOL are available in `conda` channels [conda-forge](#) and [schrodinger](#), respectively.

### Open-Source PyMOL

The command in **Step 2** automatically installs [Open-Source PyMOL](#) available from [conda-forge](#). It can also be installed with:

```
conda install -c conda-forge pymol-open-source=2.5.0
```

**Warning:** Open-Source PyMOL 2.4.0 has a bug where it cannot open Gaussian cube files. DelPhi to output phimap volumetric data is in this format, which is used in the [electrostatics](#) calculations.

Alternatively, on Linux, PyMOL can be installed with the system package manager, e.g., `apt` in Ubuntu or `dnf` in Fedora. However, it is not possible to install PyMOL into a virtual environment using this method. Therefore, MD DaVis must be installed in the system python, which may create conflicts with existing python packages on which many system programs may depend.

On Windows, if you are not using `conda`, then pre-built Open-Source PyMOL can be downloaded from [Christoph Gohlke's page](#) distributing unofficial windows binaries for python extension packages. However, we have faced issues with using this package with other dependencies of MD DaVis.

## Commercial/Educational PyMOL

The [Commercial/Educational PyMOL](#) can be used instead of Open-Source PyMOL by installing the [md-davis-pymol](#) or [md-davis-pymol-dev](#) environment in **Step 2** of the above-mentioned installation methods:

```
conda env create djmaity/md-davis-pymol
```

Alternatively, it can be installed with:

```
conda install -c schrodinger pymol-bundle
```

## DelPhi and MSMS

Python dependencies are automatically installed. However, the electrostatics calculation requires the following two programs, which must be obtained separately.

- [Delphi C++](#) version greater than or equal to 8.1
- [MSMS](#)

## 1.3 Uninstall

MD DaVis can be easily uninstalled like any other python package, with:

```
pip uninstall md-davis
```

As with any python package, this does not remove the dependencies installed by MD DaVis. That is why installing MD DaVis in a conda environment is recommended. Then, the whole conda environment may be entirely removed without affecting other python packages on the system.

```
conda env remove --name md-davis
```

where `md-davis` is the name of the conda environment. Modify the command to provide the appropriate name for the conda environment if you change it.

---

**Note:** On Linux, if MD DaVis was installed as root or with `sudo` (highly discouraged), the uninstall command should also be run with `sudo`.

---

## 1.4 User Guide

### 1.4.1 Quickstart

The following steps walks you through the process of using the MD DaVis *graphical user interface (GUI)* and *command line interface (CLI)* to assimilate the analysis data and creating interactive plots.

### Step 1: Calculate the required quantities

MD DaVis has been designed to natively work with the output from GROMACS analysis tools. However, the same analysis performed with other tools may be visualized by MD DaVis as long as the input files can be appropriately formatted. Root mean squared deviation (RMSD) and radius of gyration ( $R_G$ ) will be used to create free energy landscapes and the rest will be used for residue property plot. Ensure that the trajectories and structures are processed and corrected for periodic boundary conditions.

#### Root mean squared deviation (RMSD)

```
gmx rms -f trajectory.trr -s structure.pdb -o rmsd.xvg
```

The structure provided will be used as the reference for calculating RMSD of the trajectory.

#### Radius of gyration ( $R_G$ )

```
gmx gyrate -f trajectory.trr -s structure.pdb -o rg.xvg
```

#### Root mean squared fluctuation (RMSF)

```
gmx rmsf -f trajectory.trr -s structure.pdb -res -o rmsf.xvg
```

This would use the whole molecule for RMSF calculation. You may calculate the RMSF of each chain separately in a protein with multiple chains. This would reflect the fluctuations only due to backbone motion and eliminate the fluctuation from relative motion of chains.

#### Solvent accessible surface area (SASA)

To calculate the average SASA per residue pass `-or` option to `gmx sasa`.

```
gmx sasa -f trajectory.trr -s structure.pdb -o sasa.xvg -or resarea.xvg
```

#### Secondary structure using DSSP

```
gmx do_dssp -f trajectory.trr -s structure.pdb -o dssp -ssdump sec_str -sc sec_str_count
```

### Step 2: Collate the data into an HDF file

Use the collate tab in the *MD DaVis GUI* to collect all the analysis data calculated above into a HDF file (data.h5) for organized storage and input to subsequent commands.



The image shows the MD DaVis software interface with the 'Collate' tab selected. The interface includes several input fields and buttons for configuring data collation.

**MD DaVis**

Landscapes | **Electrodynamics** | **Collate** | Sequence

Name: Protein

Label: *<i>HTML</i>* Protein Label

Text Label: Protein Label

Structure: /data/structure.pdb Browse

Trajectory: /data/trajectory.xtc Browse

Time series

RMSD: /data/rmsd.xvg Browse

Rg: /data/rg.xvg Browse

☒ Calculate dihedral angles

Chunk Size: 1000

Residue properties

DSSP: /data/sec\_str.dat Browse

Root-mean-square fluctuation (RMSF)

RMSF: /data/rmsf.xvg Browse

SASA: /data/resarea.xvg Browse

Electrostatics: electrostatics ▼

HDF File: /data/data.h5 Browse

Create HDF File

According to the image the files calculated in Step 1 are in /data directory.

See the [collate user guide page](#) for more details.

### Step 3: Plot the free energy landscape

You can directly plot the free energy landscape from the RMSD and Rg files using the `md-davis landscape_xvg` command or the GUI as shown below:

The image shows the MD DaVis GUI with the 'Landscape' tab selected. The 'Free Energy' radio button is chosen, and the temperature is set to 300.00 K. The 'Perspective' view is selected. The title is 'Protein Landscape'. The shape of each landscape is defined by 100 X-bins and 100 Y-bins, with the 'Use Common Ranges' checkbox checked. The axis labels are X: RMSD (in nm), Y: Radius of Gyration (in nm), and Z: Free Energy (in kJ mol<sup>-1</sup>). The size and layout of the plot are set to 1080 width and 1080 height. The 'Save landscapes to' field is empty, and the 'Output HTML file' is '/data/FEL.html'. A 'Create Free Energy Landscape' button is present. The input section shows X: /data/rmsd.xvg and Y: /data/rg.xvg, both with 'Browse' buttons. The Name is 'protein\_name' and the Label is 'Protein Label'.

MD DaVis

Landscape | Electrodynamics | Collate | Sequence

☐ Histogram ☒ Free Energy Temperature (in K) 300.00

☒ Perspective ☐ Orthographic

Title Protein Landscape

Shape of each landscape

X-bins 100 Y-bins 100 ☒ Use Common Ranges

Axis Labels

X RMSD (in nm) Y Radius of Gyration (in nm) Z Free Energy (in kJ mol<sup>-1</sup>)

Size and layout of the plot

Width 1080 Height 1080

Save landscapes to  Browse

Output HTML file /data/FEL.html Browse

Create Free Energy Landscape

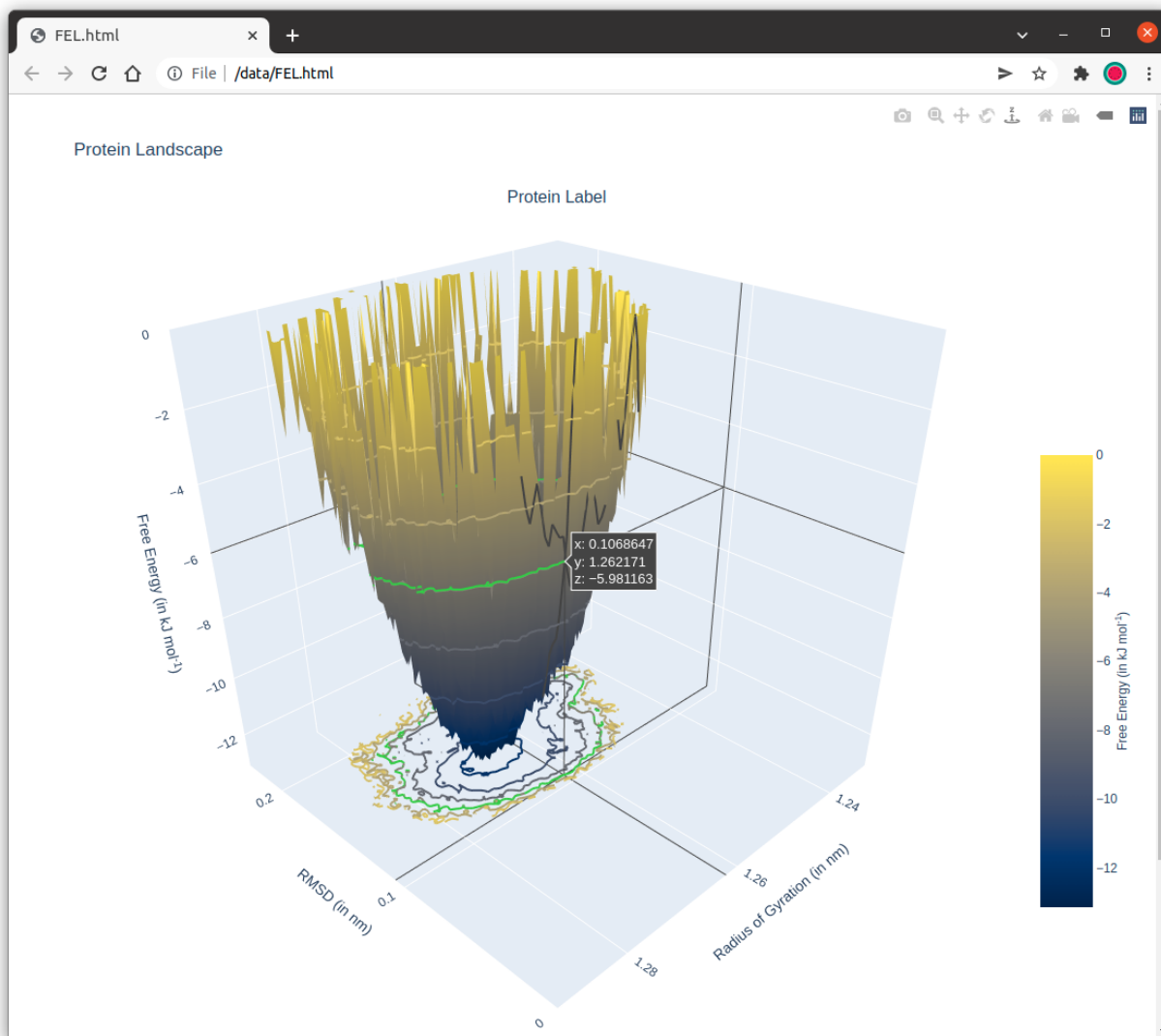
Input

X /data/rmsd.xvg Browse

Y /data/rg.xvg Browse

Name protein\_name Label Protein Label

This will create a free energy landscape like the image shown below; click the image to open the interactive HTML plot.



Or, you can plot the free energy landscape from the HDF file created in Step 2.

```
md-davis landscape -T 300 --common data.h5 -o FEL.html
```

This command will create the FEL.html file with the interactive landscape. It will not open the file like other plotting commands, so check the working directory for the output html file.

#### Step 4: Plot the residue property plot

**Step 4a:** Create a pickle file (a serialized binary file used to store python objects) with the residue dataframe using:

```
md-davis residue data.h5 -o residue_dataframe.p
```

**Step 4b:** Plot the pickle file from the previous command using:

```
md-davis plot_residue residue_dataframe.p -o plot.html
```

## 1.4.2 Plotting Xmgrace (.xvg) Files

The GROMACS molecular dynamics software bundles numerous analysis tools. The output from these are in (.xvg) format to be viewed with xmgrace. Although xmgrace has some powerful tools for quick calculation of mean standard deviation, etc., the plots are cumbersome to deal with and difficult to compare multiple files.

MD DaVis provides the command `xvg` to plot such files using `plotly` or `matplotlib`.

The .xvg files are space or tab delimited text files with time and data in the first and second columns, respectively.

```
# gmx rms is part of G R O M A C S:
#
# Gromacs Runs On Most of All Computer Systems
#
@ title "RMSD"
@ xaxis label "Time (ps)"
@ yaxis label "RMSD (nm)"
@TYPE xy
@ subtitle "System after lsq fit to System"
0.0000000 0.0000031
10.0000000 0.0677038
20.0000000 0.0837483
30.0000000 0.0894995
```

MD DaVis can also plot multiple Grace (.xvg) files, which is the format for the output files from many GROMACS analysis tools. For example, an interactive plot with RMSD and RG from multiple trajectories can be created for quick comparison.

```
md-davis xvg <path/to/file.xvg>
```

Replace `<path/to/file.xvg>` with the location of your .xvg file.

Create interactive plot for time series data: root mean squared deviation (RMSD) and radius of gyration (R:subscript: G)

```
<iframe src="/_static/acylphosphatase_rmssg_rg.html" frameborder="0" width="100%" height="500px"></iframe>
```

---

**Note:** The titles and axis labels are parsed from the lines in the xvg file starting with @. It is expected that the when plotting multiple xvg files, they contain the same kind of data. Therefore, the titles and axis labels in the last supplied file are used.

---

### md-davis xvg

Plot xmgrace (.xvg) files generated by GROMACS.

Arguments:

`xvg_files` (Files): Input xmgrace (.xvg) files'

```
md-davis xvg [OPTIONS] XVG_FILES...
```

## Options

**-o, --output** <output>  
Output filename

**--matplotlib, --plotly**  
Use plotly for plotting

## Arguments

**XVG\_FILES**  
Required argument(s)

### 1.4.3 Sequence

### 1.4.4 Collate Analysis Data

The collate function of MD Davis uses [HDF5](#) file format to store the heterogeneous data obtained from consolidating and organizing the data from multiple calculations using the h5py Python module. [HDF5](#) being an open binary format, allows users to open these files directly in C, C++, or FORTRAN programs. It is much smaller to store and faster to load than text files. Moreover, the data can be inspected with the HDF View GUI.

**Step 1:** Create a TOML file (input.toml) as shown below specifying the output files from GROMACS.

```
name = 'protein_name_no_spaces'
output = 'protein_data.h5'
label = 'Protein from <i>some organism</i>'
text_label = 'Protein Name'

trajectory = 'trajectory.xtc'
structure = 'structure.pdb'

[timeseries]
rmsd = 'rmsd.xvg'
rg = 'rg.xvg'

[dihedral]
chunk = 101          # Number of frames to read at a time

[residue_property]
secondary_structure = 'sec_str.dat'
rmsf = 'rmsf.xvg'
sasa = 'resarea.xvg'
```

If the `chunk` under `dihedral` is provided, MD DaVis will calculate backbone dihedral angles for all frames and the torsional flexibility (circular standard deviation) of each dihedral angle.

In case, the RMSF for each chain was calculated separately, the files may be provided with `rmsf` key as follows:

```
[residue_property]
rmsf = ['rmsf_chain_A.xvg', 'rmsf_chain_B.xvg']
```

**Step 2:** Collect all the data from the output files into a single HDF file using the following command:

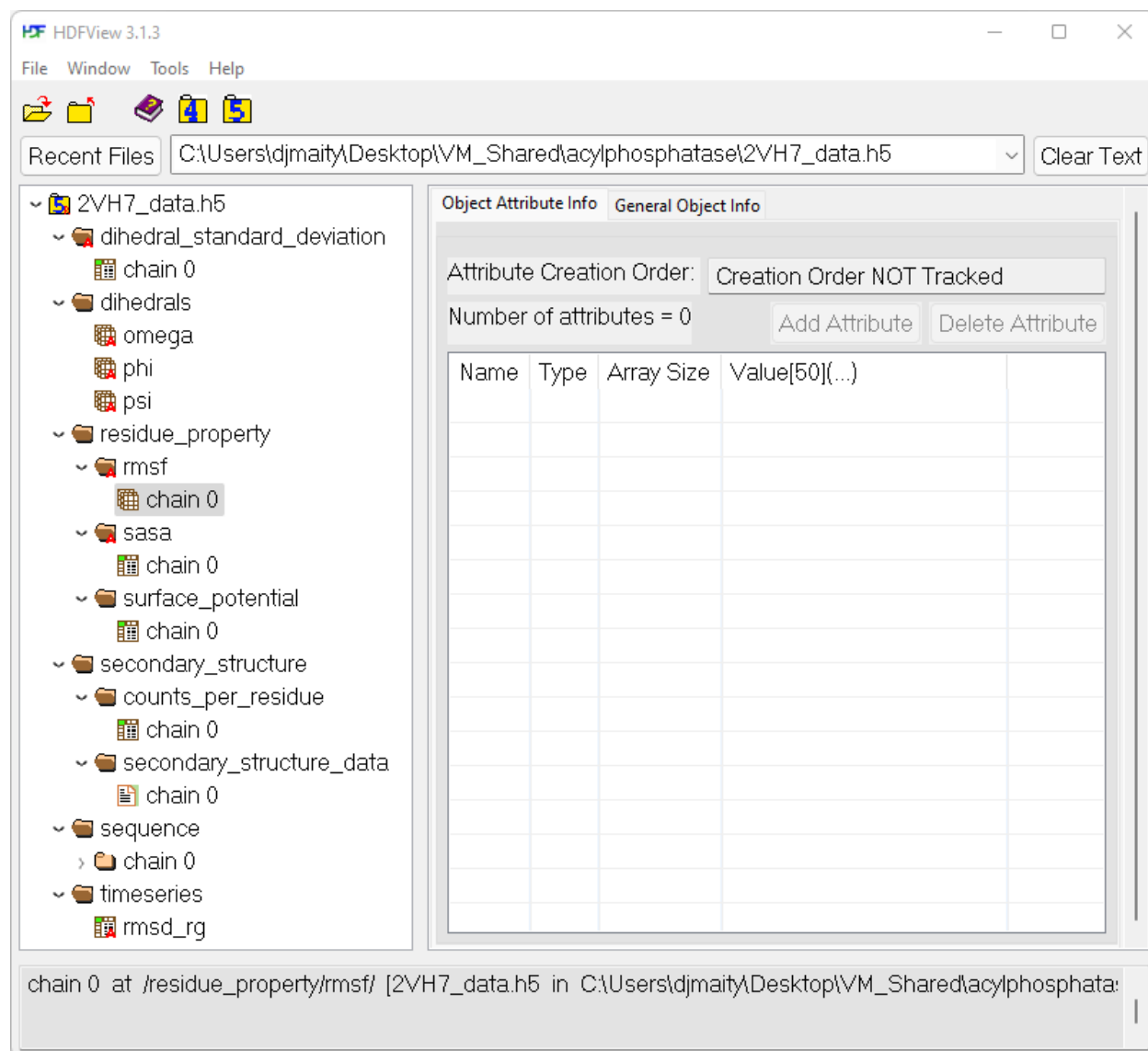
```
md-davis collate input.toml
```

Multiple input files can also be provided.

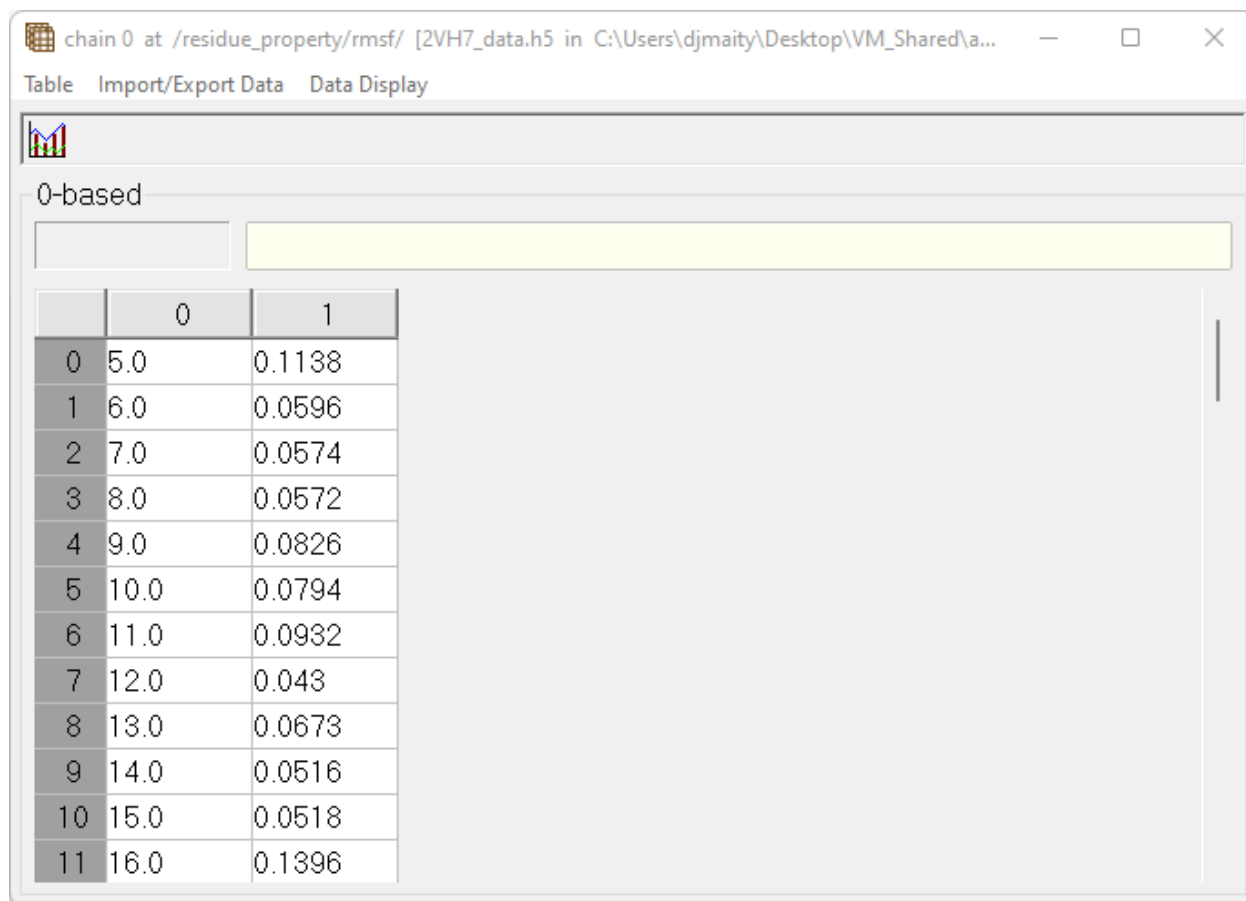
```
md-davis collate input1.toml input2.toml
```

**Note:** The command can be run multiple times to incrementally add data into a HDF5 file by modifying the TOML file. For example, first adding the RMSD and Rg then adding dihedral and so no. However, sometimes if a variable or group already exists in the HDF5 file then h5py will fail to create the file. Please the delete the exisintg file and rerun the command by providing all data at once.

## 1.4.5 HDF Files



Double clicking the 'Chain 0' under 'rmsf' opens up the data:



### 1.4.6 Free Energy Landscapes

The free energy landscape of a protein is a continuous function between the protein conformations and the free energy. MD DaVis creates free energy landscapes using the method described in [Tavernelli et al., 2003](#). First, the user must provide two quantities of interest to classify the protein conformations, such as root mean square deviation (RMSD) and radius of gyration ( $R_G$ ), or the projection on the first two eigenvectors from principal component analysis. Then, the two variables are binned to make a 2D histogram, and the free energy landscape is obtained by Boltzmann inversion of the histogram using:

$$\Delta\varepsilon_i = -k_B T \ln \left( \frac{n_i}{n_{max}} \right)$$

where,  $\Delta\varepsilon_i$  is the change in free energy between the conformations in the  $i$ th bin and the maximally occupied bin,  $n_i$  and  $n_{max}$  are the number of conformations in these, respectively. This formula results in 0 energy for the most probable state and all other states having positive energy. Since the absolute free energy is impossible to calculate and only the change in free energy is meaningful, the maximum value in the free energy landscape is subtracted from all the bins so that the most probable state has the most negative free energy.

The interactive 3D plot of the free energy landscape can be rotated and viewed from various directions, zoomed in and out, and labels on the surface inspected to determine the valleys and wells.

A unique feature of MD DaVis is that rotating one landscape in a plot with multiple landscapes rotates all to the same view.

The trajectory points can also be animated or plotted on top of the free energy landscape; this allows inspecting the traversal of wells in the free energy landscapes during the MD simulation.

The landscapes can be plotted using common ranges and same number of bins, enabling their comparison. The wonderful feature of the plot created using MD DaVis is that it is an interactive html, and rotating one subplot rotates all others to the same view. These features facilitate the quick and efficient comparison of the free energy landscapes.

---

**Note:** Plotting free energy landscapes requires a large number of frames, therefore, always save enough number of frames during the simulation. The number of frames required to obtain a reasonable representation of the free energy landscape depends on flexibility of the protein. If the protein accesses a large number of conformational states a greater number of frames would be required. As a ball park, at least 10000 frames are required.

---

---

**Note:** These commands do not open the output html file in a web browser like other plotting commands, so please check the working directory for the output html file. If a file with the same name exists, it will be overwritten.

---

### Free Energy Landscapes from xvg files

The quickest way to plot the free energy landscape is from text files containing the two properties of interest.

```
md-davis landscape_xvg -c -T 300 -x rmsd_1.xvg -y rg_1.xvg -n "name_1" -l "FEL for 1"
```

Here, `.xvg` the file containing the RMSD and  $R_G$  precalculated using GROMACS is provided as the `-x` and `-y` input files.

While providing other properties or files not created by GROMACS ensure that they are two column files

If the output filename is not provided the default name is `landscape.html`.

`-c` specifies that the ranges of the all the landscapes must be the same

`-T 300` specifies to use 300 K as the temperature for the boltzmann inversion. If this option is not provided, only the 2D histogram of the data is plotted.

An excellent feature of MD DaVis is that while creating the 2D histogram, it creates a data structure in which the index of each frame in a particular bin is stored. This feature allows one to find the frames in a particular bin of the landscape, and the frames in the minima are generally of particular interest.

To plot multiple free energy landscapes as subplots just provide the inputs one after the other. The `-x`, `-y`, `-n`, and `-l` for one trajectory must be provided together before the next set of inputs from the other trajectory.

```
md-davis landscape_xvg -c -T 300 -x rmsd_file_1.xvg -y rg_file_1.xvg -n "name_1" -l  
↪ "Free Energy Landscape for 1" -x rmsd_file_1.xvg -y rg_file_1.xvg -n "name_2" -l "Free_  
↪ Energy Landscape for 2" -x rmsd_file_1.xvg -y rg_file_1.xvg -n "name_3" -l "Free_  
↪ Energy Landscape for 3"
```

### Plot Free Energy Landscapes from HDF Files

If the RMSD and  $R_G$  have been collated into HDF files for each trajectory. These may be plotted using:

```
md-davis landscape rmsd_rg -c -T 300 file1.h5 file2.h5 file3.h5
```



## Plot Free Energy Landscapes Overlaid with Trajectory Points

One must save the landscape created by `landscape_xvg` or `landscape` with the `-s` option before this one can be used. Since the output generated for single landscape is big, visualization of multiple landscapes becomes impractical. So, it only plots one landscape at a time. Select the desired landscape in `landscapes.h5` by providing its index with `-i`. By default only the first landscape is plotted

```
md-davis landscape animation landscapes.h5 -i 0 --static -o trajectory.html
```

## Step 4: Free energy Landscapes

### Create and plot free energy landscapes using common bins and ranges

```
md-davis landscape rmsd_rg -T 300 --common --select backbone output1.h5 output2.h5 -s ↵
↵ landscapes.h5
```

This command will create an html file with the interactive landscapes. It will not open the file like other plotting commands, so check the working directory for the output html file. ### Plot free energy landscape overlaid with trajectory points One must save the landscape created by the previous command with `-s` before this one can be used. Since the output generated for single landscape is big, visualization of multiple landscapes becomes impractical. So, it only plots one landscape at a time. Select the desired landscape in `landscapes.h5` by providing its index with `-i`. By default only the first landscape is plotted

```
md-davis landscape animation landscapes.h5 -i 0 --static -o trajectory.html
```

### Plot free energy landscape overlaid with trajectory points One must save the landscape created by the previous command with `-s` before this one can be used. Since the output generated for single landscape is big, visualization of multiple landscapes becomes impractical. So, it only plots one landscape at a time. Select the desired landscape in `landscapes.h5` by providing its index with `-i`. By default only the first landscape is plotted

```
md-davis landscape animation landscapes.h5 -i 0 --static -o trajectory.html
```

**Note:** The first rotation may not sync across all subplots. Please rotate again to sync the view of all the subplots. Also, zooming a subplot is also not synchronized immediately. After zooming a subplot rotate the same subplot to sync the zoom on all subplots.

## Plot Free Energy Landscapes using the MD DaVis GUI

The free energy landscapes from `xvg` files can also be created using the Landscape tab in the MD DaVis GUI. Plotting from HDF files and animated/overlaid trajectories is not supported using the GUI but shall be incorporated soon.

Linux

Windows

MD DaVis

**Landscape** | Electrodynamics | Collate | Sequence

☐ Histogram ☒ Free Energy Temperature (in K)

☒ Perspective ☐ Orthographic

Title

Shape of each landscape

X-bins  Y-bins  ☒ Use Common Ranges

Axis Labels

X  Y  Z 

Size and layout of the plot

Width  Height

Save landscapes to

Output HTML file

Input

X

Y

Name  Label

MD DaVis

Landscape Electroynamics Collate Sequence

☐ Histogram ☒ Free Energy Temperature (in K) 300.00

☒ Perspective ☐ Orthographic

Title

Shape of each landscape

X-bins 100 Y-bins 100 ☒ Use Common Ranges

Axis Labels

X RMSD (in nm) Y Radius of Gyration (in nm) Z Free Envergy (in kJ mol<sup>-1</sup>)

Size and layout of the plot

Width 1920 Height 1080

Save landscapes to Browse

Output HTML file Browse

Create Free Energy Landscape

Input

X Browse

Y Browse

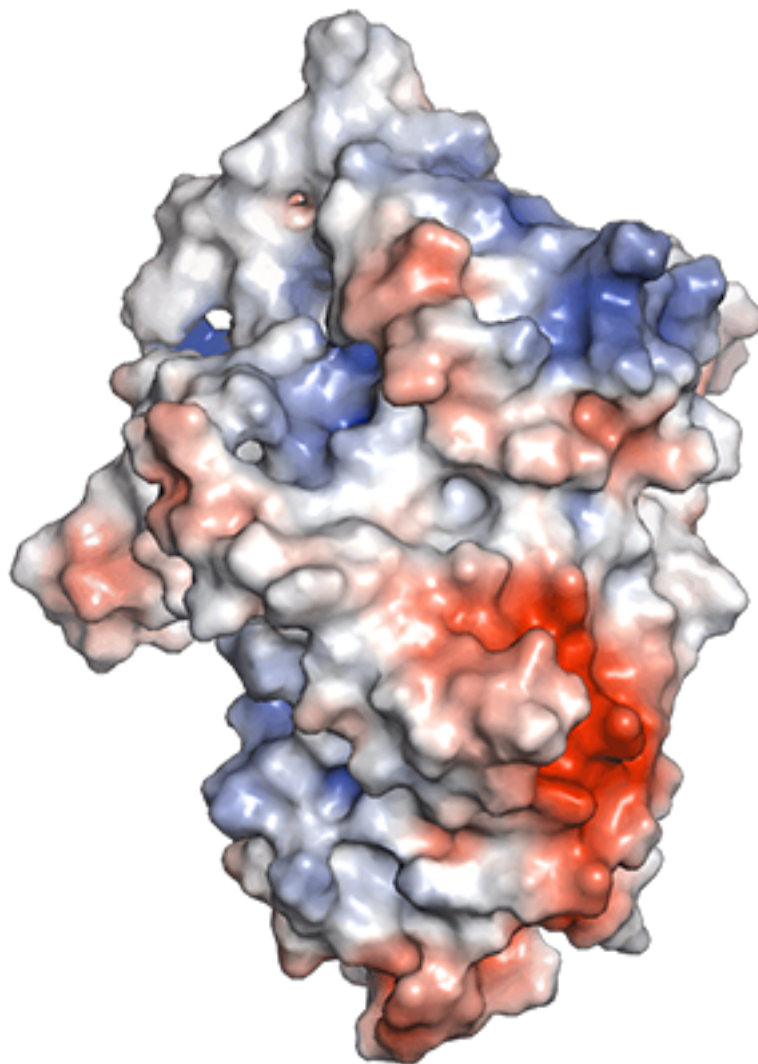
Name Label

-

+

### 1.4.7 Surface Electrostatic Potential Per Residue

Usually, the distribution of electrostatic potential around a protein molecule is visualized by solving the Poisson-Boltzmann equation using a grid base approach (APBS/Delphi) and coloring the molecular surface with the obtained electrostatic potentials, as shown below.



This only allows for qualitative inspection of one structure or conformation at a time. In contrast, MD DaVis can calculate the surface electrostatic potential per residue from electrostatics calculations on a sample of conformation from the MD trajectory. Thereby, incorporating dynamical information into the analysis. The steps involved are as follows:

1. Obtain a sample of conformations from the trajectory as PDB files. Ideally, all the conformations should be centered at the origin and aligned. Optionally, the reference conformation used for alignment may be oriented with its first, second and third [principal axis](#) along the x, y, and z axis, respectively. When comparing samples from two or more trajectories ensure that the reference structures for each are aligned to reduce discrepancy caused by grid selection during electrostatics calculation.
2. Use `md-davis electrostatics` to run MSMS and DelPhi on each PDB file.

```
md-davis electrostatics --surface -m <MSMS> -d <DELPHI> -o <OUTPUT_DIR> [PDB_FILES]
```

This calculates the triangulated molecular surface using the [MSMS](#) program (Sanner et al., 1996) and evaluates the electrostatic potentials at the vertices of the triangulated surface using [Delphi](#) (Li et al., 2019). In addition, the volumetric data for electrostatic potential at grid points around the molecule are also saved by DelPhi in [Gaussian CUBE format](#). This can be used for 3D vislization of the surface electrostatic potentials (Figure above) or [electric field dynamics](#). It is advisable to align the PDB files and specify the grid size for Delphi calculation, so that the same grid points are used for each calculation.

3. Specify the path to the directory containing the surface potential files in the input TOML file used by :ref: `collate`.

```
[residue_property]
surface_potential = '<OUTPUT_DIR>'
```

4. The total and mean electrostatic potential per residue is calculated when the data is *collated* into an HDF file.

```
md-davis collate input.toml
```

This will search for all .pot files in the specified <OUTPUT\_DIRECTORY>. The vertices corresponding to each residue are identified along with the corresponding electrostatic potential. The total and mean of the electrostatic potentials on the vertices for each residue are calculated. The total gives information about the charge on the surface as well as the exposed surface area, while the mean nullifies the contribution from the exposed surface area. The mean and standard deviation of both are calculated and saved in the output HDF file.

5. Follow the steps for creating a *residue property plot* to visualize the data.

```
md-davis plot residue output_residue_wise_data.p
```

**Note:** A discrepancy between the molecular surface calculated by MSMS and the molecular surface used by DelPhi to detect the protein's interior was observed for buried residues in multimeric proteins. DelPhi uses a different dielectric constant for the molecule's interior, so the potential on these residues was very high.

```
md-davis electrostatics --surface -m <MSMS_EXECUTABLE> -d <DELPHI_EXECUTABLE> -o <OUTPUT_
  ↳ DIRECTORY> [PDB_FILES]
```

md-davis electrostatics is a wrapper for running Delphi and reporting the electrostatic potential at the vertices of a triangulated surface obtained using MSMS. Therefore, these must be installed for this command to work, which may be downloaded from <http://compbio.clemson.edu/delphi> and <http://mgltools.scripps.edu/downloads#msms>, respectively.

The default parameters used for running Delphi are:

- 2000 linear iterations
- maximum change in potential of  $10^{-10}$  kT/e.
- The salt concentration is set to 0.15 M
- solvent dielectric value of 80 (dielectric of water)

If you want to use different set of parameters, you may run Delphi without the md

The atomic charges and radii from the CHARMM force field are provided by default. The parameter files containing the atomic charges and radii for DelPhi are available at: <http://compbio.clemson.edu/downloadDir/delphi/parameters.tar.gz>

The output potential map is saved in the cube file format (used by Gaussian software); the other formats do not load in molecular visualization software.

The following charge and radius files are provided with the source code for MD DaVis.

```
md_davis/md_davis/electrostatics/charmm.crg    md_davis/md_davis/electrostatics/
charmm.siz
```

If you receive a warning during Delphi run regarding missing charge or radius. Then the missing properties must be added to these files or whichever files you provide to md-davis electrostatics.

### 1.4.8 Electric Field Dynamics

The electrostatic potentials calculated in *Surface Electrostatic Potential Per Residue* can be visualized as a 3D animation of electric field lines using:

```
md-davis electrodynamics --ss_color --surface --name Human_AcP 2VH7/2VH7_electrostatics
```

This creates a [PyMOL](#) session with the conformations as frames in the animation as shown below:

1. The coordinates of the reference structure are translated to place the center of mass of the molecule at the origin and rotated so that the first, second, and third principal axes are along the x, y, and z-axes, respectively.
2. The frames sampled from the trajectory are aligned to the reference.
3. The electrostatic potentials are obtained for each sampled structure using Delphi. The box for each calculation is centered at the origin, and the number of grid points is manually set to the same value for each structure to ensure the same box size during each calculation.
4. The surface electrostatic potentials calculated per residue or atom are written into the output PDB file's B-factor or occupancy column.
5. The output PDB file and the corresponding electric field from the sample are visualized as frames in PyMOL (Schrödinger, LLC, 2015), which can animate the dynamics of the electric field lines.

### 1.4.9 Residue Property Plot

Understanding the function–dynamics relationship of protein often re-quires comparing two or more properties, e.g., the effect of dihedral fluctuation on the RMSF of the protein. The two properties, dihedral fluctuation and RMSF, can be easily plotted with any plotting program. Later, one might be interested in knowing the trend between the RMSF and solvent exposure of the residues. Traditionally, each time new prop-erties are compared, new plots need to be prepared. This repetitive pro-cess is alleviated by making overlaid plots of all the residue properties at once (Figure 1A). MD DaVis can plot the following quantities: RMSF, torsional flexibility, secondary structure, solvent accessible surface area, and surface electrostatic potential on each residue. Showing all the prop-erties as an overlaid plot can be overwhelming and cluttering. However, the option to interactively turn on or off the visualization for each data series clears the clutter and highlights the similarities and differences. Moreover, the labels and annotations that appear on hovering the cursor over certain regions improve the interpretability of these plots, thereby granting substantial time savings. The data obtained from multiple trajectories can be overlaid in a single interactive plot for ease of comparison, which removes the need for making multiple plots and immediately brings out the distinguishing features between the trajectories. The novel feature of MD DaVis is that it can align the data for similar proteins by inserting required gaps along the x-axis using an alignment file. Aligning the residues on the x-axis aligns the peaks, highlighting the similarities between the datasets, which is incredibly powerful when comparing the dynamical information from similar proteins.

**4. Create interactive plot of containing the following residue level properties:**

- root mean squared fluctuation (RMSF)
- torsional flexibility (circular standard deviation of backbone dihedral angles)
- Secondary structure
- Solvent accessible surface area
- Mean and standard deviation for the total surface electrostatic potential per residue
- Mean and standard deviation for the mean surface electrostatic potential per residue

It can also use an alignment to align the residue level data from different proteins along the x-axis. This ensures that the peaks line up properly for better interpretation.

Traditionally, each time the analysis of MD trajectories are compared new statis plots have to be created.

Conventional analysis requires plotting the data each time new properties are compared. This repetitive process is alleviated by making overlaid plots of all the informative residue propertie

---

**Note:** the paths in the input toml file is relative to the location where the md-davis command will be called from. To avoid any confusion try using absolute paths.

---

Note that each data is optional in the residue property plot. Therefore, a plot can be created even if the secondary structure is not available, albeit devoid of rich structural information.

## Secondary Structure

-ssdump option will output a file `sec_str.dat` containing the secondary structure for the full trajectory as single letter codes.

Code	Secondary structure
H	-helix
G	3 <sub>10</sub> -helix
I	-helix
B	-bridge
E	strand
T	Turn
S	Bend
~	Loop

This would be processed by MD DaVis to calculate the percentage of occurrence of each secondary structure at each residue location.

---

**Note:** For the GROMACS command `do_dssp` to work the DSSP binary must be available on your system. Download DSSP from <ftp://ftp.cmbi.ru.nl/pub/software/dssp/>

---

## How to interact with the plot

## Step 3: Plotting overlaid residue data **Step 3a:** Create a pickle file with the residue dataframe using:

```
md-davis residue dataframe --prefix name1 output1.h5 data1.p
```

The optional argument `-a annotations.json` can be provided to place a mark at certain residue locations. The contents of `annotations.json` should be of the following form:

```
{
  "chain 0": {"Active Site": [23, 41], "Substrate Binding Site": [56]},
  "chain 1": {"Nucleotide Binding Regions": [15, 18]}
}
```

Each type of annotation is rendered with a different mark. Following annotations are available at present: \* Active Site \* Nucleotide Binding Regions \* NADP Binding Site \* Substrate Binding Site \* Metal Binding Site \* Cofactor Binding Site \* Mutation

**Step 3b:** If your proteins are of different lengths and you need the peaks to be aligned, create a JSON file as shown below.

```
{
  "alignment": "path/to/alignment_file.clustal_num",
  "locations": {
    "name1": "name1_residue_wise_data.p",
    "name2": "name2_residue_wise_data.p",
    "name3": "name3_residue_wise_data.p"
  },
  "output": "acylphosphatase_residue_wise_data_aligned.p"
}
```

The contents of the alignment file, `alignment_file.clustal_num` must be in CLUSTAL format.

**Step 3b:** Plot the residue data pickle file from the previous command using:

```
md-davis plot residue data1.p data2.p
```

Note the numbers at the end of the `--rmsf` options are the start and end time for the RMSF calculation in nanosecond. These will be inserted as attributes in the HDF file and must be provided. In case, the RMSF for each chain was calculated separately, the files may be provided to `--rmsf` option in the correct order followed by the start and end times.

The optional argument `-a annotations.json` can be provided to place a mark at certain residue locations. The contents of `annotations.json` should be of the following form:

```
{
  "chain 0": {"Active Site": [23, 41], "Substrate Binding Site": [56]},
  "chain 1": {"Nucleotide Binding Regions": [15, 18]}
}
```

Each type of annotation is rendered with a different mark. Following annotations are available at present: \* Active Site \* Nucleotide Binding Regions \* NADP Binding Site \* Substrate Binding Site \* Metal Binding Site \* Cofactor Binding Site \* Mutation

**Step 3b:** If your proteins are of different lengths and you need the peaks to be aligned, create a JSON file as shown below.

```
{
  "alignment": "path/to/alignment_file.clustal_num",
  "locations": {
    "name1": "name1_residue_wise_data.p",
    "name2": "name2_residue_wise_data.p",
    "name3": "name3_residue_wise_data.p"
  },
  "output": "acylphosphatase_residue_wise_data_aligned.p"
}
```

The contents of the alignment file, `alignment_file.clustal_num` must be in CLUSTAL format; for example:

```
CLUSTAL O(1.2.4) multiple sequence alignment

name1      --STARPLKSVDYEVFGRVQGVCFRMYAEDEARKIGVVGWVKNTSKGTVTGQVQGPEEKV      58
name2      -----PRLVALVKGRVQGVGYRAFAQKALELGLSGYAENLPDGRVEVVAEGPKEAL      52
name3      ---VAKQIFALDFEIFGRVQGVFFRKHTSHEAKRLGVRGWCNTRDGTVKGQLEAPMMNL      57
```

(continues on next page)



(continued from previous page)

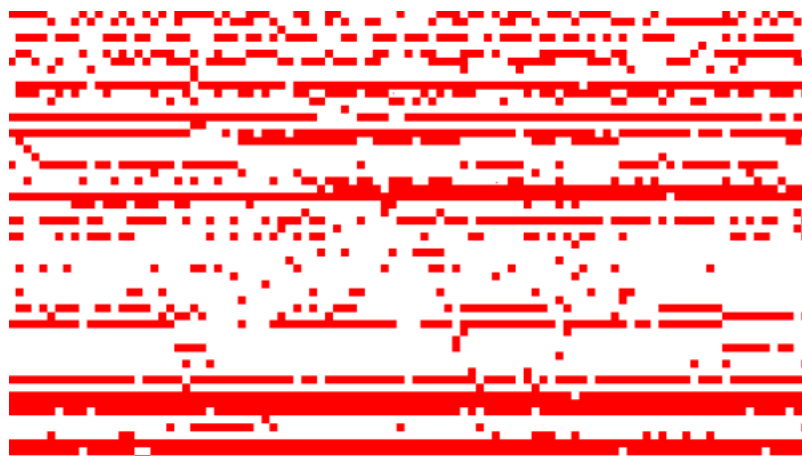
```

: *.***** :* . :. . : *: * * * . :
name1      NSMKSWLSKVGSPSSRIDRTNFSNEKTISKLEYSNFSVRY 98
name2      ELFLHHLKQ--GPRLARVEAVEVQWGEE--AGLKGFHVV- 87
name3      MEMKHWLENNRIPNAKVSKAEFSQIQEIEDYTFTSFDIKH 97
: : * : * :
```

## 1.4.10 Contact/H-bond matrix

### Motivation

GROMACS has an excellent tool to calculate hydrogen bonds (H-bonds) or contacts from trajectories called `gmx hbond`. It is written in C++, and the loops are parallelized. Therefore, its performance cannot be matched by any Python code. The default output from `gmx hbond` is a `.xvg` file containing the total number of H-bonds/contacts between the selected groups over all frames, which is not very informative. The option `-hbm` outputs the existence matrix for all H-bonds/contacts over all frames in an `X PixMap (.xpm)` image:



Only a few image viewers support `.xpm` file format, and the image cannot be easily interpreted because it does not contain any labels for the participating atoms and residues in the H-bonds/contacts. To overcome these limitations, use MD DaVis for analyzing the output files created by `gmx hbond`.

### Steps

1. Calculate the H-bonds/contacts using `gmx hbond` with the options `-hbm` and `-hbn`:

```
gmx hbond -f trajectory.trr -s structure.tpr -num hbnum.xvg -hbm hb_matrix -hbn hb_index
```

To calculate the contacts provide `--contact` option. `gmx hbond` requires the `.tpr` file to be supplied to the `-s` option, and other structure formats like `.pdb` or `.gro` are not accepted. However, the `md-davis hbond` command in the next step requires a `.pdb` file. The first frame of the trajectory can be saved in a `.pdb` file as follows:

```
gmx trjconv -f trajectory.trr -s structure.tpr -o structure.pdb -dump 0
```

The order of rows in the `hb_matrix.xpm` image is the same as the order of atoms in the `hb_index.ndx` file. MD DaVis utilizes this to determine the atoms in each H-bond/contact (rows in the `.xpm` file).

2. Create and save the H-bonds/contacts data:

```
md-davis hbond -f hb_matrix.xpm --index hb_index.ndx -s structure.pdb --pickle hb_data.p_
↪-g hbonds_Protein
```

3. Plot the H-bonds/contacts matrix:

```
md-davis plot_hbond --percent --total_frames 101 --cutoff 33 -o 2VH7_hbond_matrix.html_
↪2VH7_hbonds.p
```

The above command plots the percentage of the H-bonds, which is calculated for each H-bond as follows:

number of frames the H-bond is observed / total number of frames \* 100

The cutoff of 33 % is set to plot only those H-bonds whose occurrence is greater than 33 %.

### Bonus Tip

Scroll down the hb\_index.ndx file to find the name of the group containing the atoms participating in H-bonds/contacts. It contains three or two columns of data for H-bonds or contacts, respectively.

```
1235 1243 1244 1248 1249 1260 1261 1263 1264 1280 1285 1286 1301 1302 1320
1321 1339 1340 1356 1361 1362 1380 1381 1389 1390 1392 1393 1410 1413 1414
1421 1424 1425 1433 1434 1436 1437 1456 1457 1468 1469 1473 1474 1492 1493
1508 1509 1525 1530 1531
[ hbonds_Protein ]
    9    10    35
   36    37   773
   55    56  1285
   62    63   725
   62    63   726
```

Hydrogen bonds (H-bonds) and contacts in a protein or between protein and ligand are calculated to understand the interactions essential for its function. GROMACS has the hbond utility to calculate the H-bonds and contacts purely from geometric considerations. By default, the command only returns the number of H-bonds/contacts detected in each frame, which is not informative enough. However, by providing suitable options to the hbond utility, an index file with the list of the atoms involved in each H-bond/contact detected and a matrix with the H-bonds/contacts can be obtained. The matrix has H-bonds/contacts listed in the index file along one dimension and the trajectory frames along the other dimension. The elements of the matrix denote the presence of a hydrogen bond at a particular frame. Unfortunately, the matrix does not have any labels, and its size becomes huge for long simulations making it uninterpretable. MD DaVis processes these files to calculate the number of frames with the H-bond/contact, which can be converted to a percentage. This can then be represented as a matrix where the participating atoms are along the two dimensions of the matrix (Figure 1D).

### 1.4.11 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

## Types of Contributions

### Report Bugs

When [reporting a bug](#) please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

### Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

### Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

### Write Documentation

MD DaVis could always use more documentation, whether as part of the official MD DaVis docs, in docstrings, or even on the web in blog posts, articles, and such.

### Feature Requests and Feedback

The best way to send feedback is to file an [issue](#).

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

### Development

Create the development environment and install the dependencies using the [dev\\_environment.yml](#) file. This installs packages for linting, packaging and building documentation in addition to the core dependencies.

```
conda env create -f dev_environment.yml -n md_davis_dev
conda activate md_davis_dev
```

Install md-davis in editable mode:

```
pip install -e md-davis
```

## 1.4.12 History

### 0.1.0 (2019-09-06)

- First release on PyPI.

### 0.2.0 (2020-04-10)

- Second alpha release on GitHub.

### 0.3.0 (2021-10-01)

- First stable feature-rich release

### 0.4.0 (2022-02-23)

- GUI for MD DaVis added; invoke with md-davis-gui
- Fixed plotting xvg files without title
- Changed data type of resi from int to int32 in collate.py

### 0.4.1 (2022-03-02)

- Fixed installation on Linux due to open-source PyMOL 2.4.0
- Fixing readthedocs documentation compilation
- Bug Fix: landscape\_animation was not working due change of landscapes from array to dict
- Removed docopt dependency
- Renamed md\_davis to md-davis in documentations

## 1.5 Tutorials

### 1.5.1 Acylphosphatase Homologs

In this tutorial, we shall analyze the molecular dynamics (MD) trajectories of four acylphosphatase (AcP) homologs:

PDB ID	Organism
2VH7	Human
2GV1	E. coli
2ACY	Bovine
1URR	Fruit Fly

Download the data from [acylphosphatase.tar.gz](http://www.mdtutorials.com/acylphosphatase.tar.gz)

If you are new to running MD simulations using GROMACS, please refer to the legendary GROMACS tutorial *Lysozyme in Water* by Dr. Justin A. Lemkul at <http://www.mdtutorials.com/>.

## RMSD and $R_G$

Often the first step after a successful MD simulation is to calculate the root-mean-square deviation (RMSD) and radius of gyration ( $R_G$ ). In GROMACS, these can be calculated using `gmx rms` and `gmx gyrate`, respectively.

```
gmx rms -f 2VH7/2VH7_trajectory.xtc -s 2VH7/2VH7_structure.pdb -o 2VH7/2VH7_rmsd
gmx gyrate -f 2VH7/2VH7_trajectory.xtc -s 2VH7/2VH7_structure.pdb -o 2VH7/2VH7_rg
```

2VH7\_rmsd.xvg and 2VH7\_rg.xvg are text files containing the RMSD and  $R_G$ , respectively. Repeat the process for all the other trajectories.

**Note:** If you have installed MD DaVis in a virtual or conda environment as suggested in the installation instructions, make sure to activate it before running the `md-davis` commands.

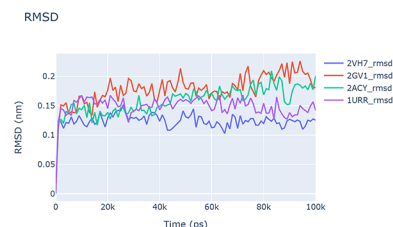
Plot 2VH7\_rmsd.xvg using:

```
md-davis xvg 2VH7/2VH7_rmsd.xvg -o 2VH7/2VH7_rmsd.html
```

You should obtain a plot like this:

To plot the RMSD from the four trajectories together, use:

```
md-davis xvg 2VH7/2VH7_rmsd.xvg 2GV1/2GV1_rmsd.xvg 2ACY/2ACY_rmsd.xvg 1URR/1URR_rmsd.xvg -o AcP_rmsd.html
```



Similarly, the  $R_G$  can also be plotted using the `md-davis xvg` command.

## Create Free Energy Landscapes

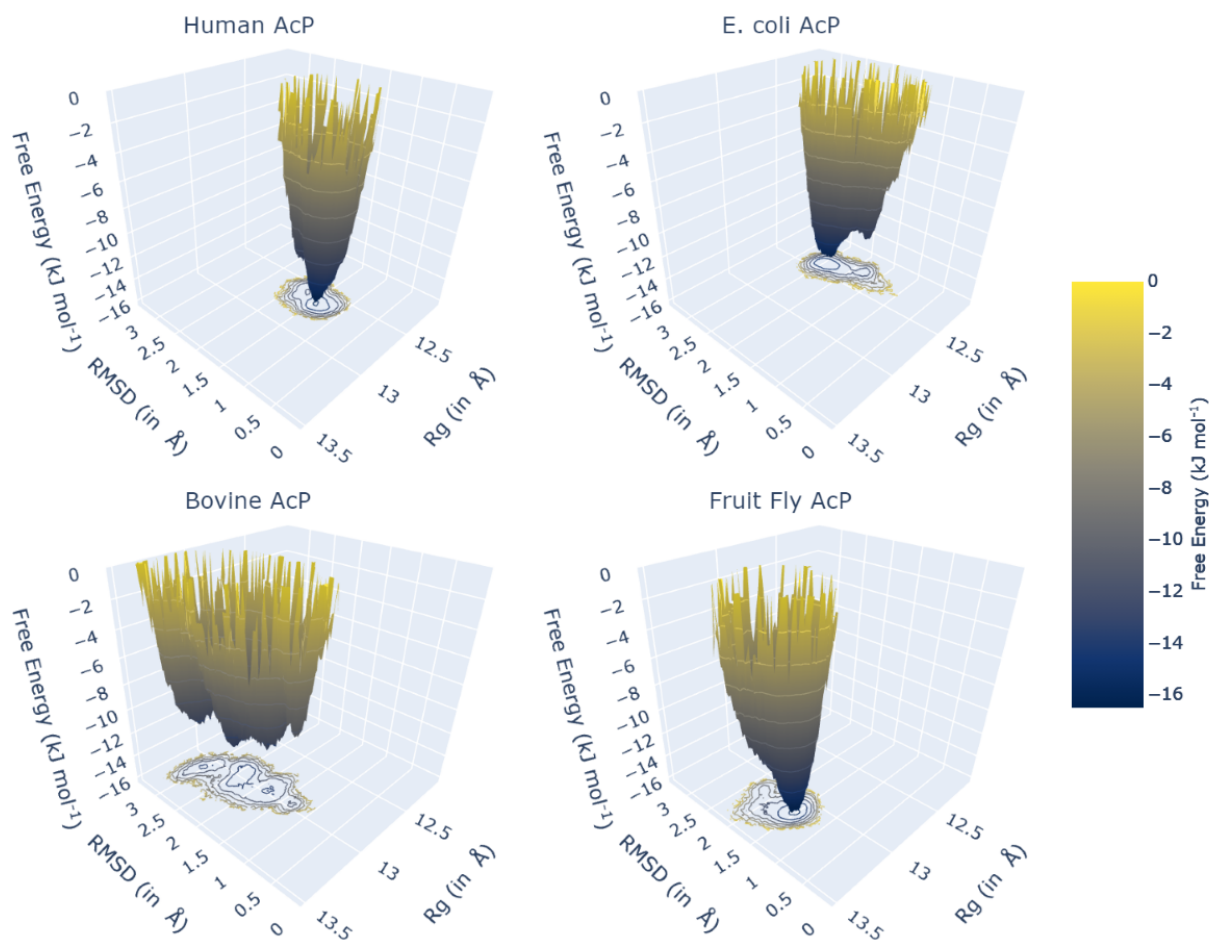
Next, we will create the free energy landscape using RMSD and  $R_G$  as the  $x$  and  $y$  variables. The sample trajectories provided with this tutorial only contain 100 frames to keep their sizes small. Thus, the RMSD and  $R_G$  files created at the beginning of this tutorial only contain 100 timesteps each. Using 100 points to create a free energy landscape would not be accurate. Therefore, the RMSD and  $R_G$  files calculated from the full 1 s trajectory containing 100,000 frames are provided with the tutorial. Use these files to create the free energy landscape. For human acylphosphatase, the free energy landscape can be created with:

```
md-davis landscape_xvg -T 300 -x 2VH7/2VH7_rmsd_full.xvg -y 2VH7/2VH7_rg_full.xvg -n
  ↪ "2VH7" -l "Human AcP" -o 2VH7_landscape.html
```

Here, `-T 300` specifies 300 K as the temperature of the system. Now to plot all four landscapes together:

```
md-davis landscape_xvg -T 300 --common -x 2VH7/2VH7_rmsd_full.xvg -y 2VH7/2VH7_rg_full.
  ↪ xvg --name "2VH7" --label "Human AcP" -x 2GV1/2GV1_rmsd_full.xvg -y 2GV1/2GV1_rg_full.
  ↪ xvg --name "2GV1" --label "E. coli AcP" -x 2ACY/2ACY_rmsd_full.xvg -y 2ACY/2ACY_rg_
  ↪ full.xvg --name "2ACY" --label "Bovine AcP" -x 1URR/1URR_rmsd_full.xvg -y 1URR/1URR_rg_
  ↪ full.xvg --name "1URR" --label "Fruit Fly AcP" -o AcP_FEL.html
```

In the command above, remember to provide `-x`, `-y`, `--name`, and `--label` together before those for the subsequent trajectory. The option `--common` instructs MD DaVis to create the four landscapes using identical ranges and binning, which allows us to compare the landscapes reliably. The output from the above command is shown below; click the image to view the interactive HTML file.



## Electrostatic Potential and Electric Field Dynamics

1. Create a sample of frames for calculating the electrostatic potential with DelPhi

```
mkdir 2VH7/2VH7_electrostatics/
gmx trjconv -f 2VH7/2VH7_trajectory.xtc -s 2VH7/2VH7_structure.pdb -o 2VH7/2VH7_
electrostatics/2VH7_frame.pdb -dt 10000 -sep
```

2. MD DaVis has the `electrostatics` command, which is a wrapper for running DelPhi and reporting the electrostatic potential at the vertices of a triangulated surface obtained using MSMS

```
md-davis electrostatics --surface -m ~/msms_i86_64Linux2.2.6.1/msms.x86_64Linux2.2.6.1 -
d ~/delphicpp_v8.4.5_serial -o 2VH7/2VH7_electrostatics/ 2VH7/2VH7_electrostatics/2VH7_
frame*.pdb
```

In the command above, the MSMS directory and the DelPhi executable are placed in the home folder. Adjust the path according to your system.

3. The electrostatic potential on the surface and the dynamics of the electric field around the molecule can be visualized with the following command:

```
md-davis electrodynamics --ss_color --surface --name Human_AcP 2VH7/2VH7_electrostatics
```

## Residue Properties Plot

1. Calculate the root-mean-square fluctuation, solvent accessible surface area, and secondary structure using GROMACS:

```
gmx rmsf -f 2VH7/2VH7_trajectory.xtc -s 2VH7/2VH7_structure.pdb -o 2VH7/2VH7_rmsf
gmx sasa -f 2VH7/2VH7_trajectory.xtc -s 2VH7/2VH7_structure.pdb -o 2VH7/2VH7_sasa.xvg -
↳ or 2VH7/2VH7_resarea.xvg
gmx do_dssp -f 2VH7/2VH7_trajectory.xtc -s 2VH7/2VH7_structure.pdb -o 2VH7/2VH7_dssp -
↳ ssdump 2VH7/2VH7_dssp -sc 2VH7/2VH7_dssp_count
```

Repeat for the remaining trajectories. We will also plot the torsional flexibility, but that will be calculated by MD DaVis later.

**Note:** For the `gmx do_dssp` command to work, the `dssp` or `mkdssp` binary must be available on your system. Download it from <ftp://ftp.cmbi.ru.nl/pub/software/dssp/> and ensure GROMACS can find it by setting the DSSP environment variable to point to its location on your system.

2. Collect and store all the calculated properties into an HDF file. To do that, first, create a TOML file as shown below, telling MD DaVis the location of each file.

```
name = '2VH7'
output = '2VH7_data.h5'
label = 'Human AcP'
text_label = 'Human AcP'

trajectory = '2VH7_trajectory.xtc'
structure = '2VH7_structure.pdb'

[timeseries]
  rmsd = '2VH7_rmsd_full.xvg'
  rg = '2VH7_rg_full.xvg'

[dihedral]
  chunk = 101

[residue_property]
  secondary_structure = '2VH7_dssp.dat'
  rmsf = '2VH7_rmsf.xvg'
  sasa = '2VH7_resarea.xvg'
  surface_potential = '2VH7_electrostatics' # directory containing electrostatic_
↳ calculations
```

Input TOML file for each trajectory is provided with the tutorial files. Next, collate all the data using MD DaVis, which can process multiple TOML files and create the respective HDF file.

```
md-davis collate 2VH7/2VH7_input.toml 2GV1/2GV1_input.toml 2ACY/2ACY_input.toml 1URR/
↳ 1URR_input.toml
```

3. Combine the data from the HDF file into a pandas dataframe with:

```
md-davis residue 2VH7_data.h5 2GV1_data.h5 2ACY_data.h5 1URR_data.h5 -o AcP_residue_data.
↳ p
```

4. Plot the residue properties:

```
md-davis plot_residue AcP_residue_data.p -o AcP_residue_data.html
```

Now, we can also align the residues of the different trajectories to align the peaks in the data.

1. obtain the sequence of residues in FASTA format from each PDB file using the `sequence` command in MD DaVis:

```
md-davis sequence 2VH7/2VH7_structure.pdb -r fasta
```

2. Use a sequence alignment program or webserver like [Clustal Omega](#) or [T-coffee](#) to obtain the alignment of these sequences in ClustalW format.

```
CLUSTAL O(1.2.4) multiple sequence alignment

2GV1_structure      ---MSKVCIIAWVYGRVQGVGFRYTTQYEAKRLGLTGYAKNLDDGSVEVVACGEEGQVEK      57
1URR_structure      -VAKQIFALDFEIFGRVQGVFFRKHTSHEAKRLGVRGWCNTRDGTGKQLEAPMMNLME      59
2VH7_structure      ----TLISVDYEIFGKVQGVFFRKHTQAEKGKLGVLGVQNTDRGTVQGQLQGPISKVRH      56
2ACY_structure      AEGDTLISVDYEIFGKVQGVFFRKYTQAEKGKLGVLGVQNTDQGTGTVQGQLQGPASKVRH      60
                   ..:  ::*:***** **  *.  *:***: *:  *  *:*:  .  ::.

2GV1_structure      LMQWLKSGGPR SARVERVLSEPH--HPSGELTDFRIR-      92
1URR_structure      MKHWLENNRIPNAKVSKAEFSQIQEIEDYTFTSFDIKH      97
2VH7_structure      MQEWLETRGSPKSHIDKANFNNEKVILKLDYSDFQIVK      94
2ACY_structure      MQEWLETKGSPKSHIDRASFHNEKVIVKLDYTDQIVK      98
                   :  .**:.  .:~::~.  .  .  :.* *
```

3. Create a TOML file to specify which alignment file corresponds to which chain and which sequence label corresponds to which data, as shown below:

```
[names]
2GV1 = '2GV1_structure'
1URR = '1URR_structure'
2VH7 = '2VH7_structure'
2ACY = '2ACY_structure'

[alignment]
'chain 0' = 'AcP_alingment.clustal_num'
```

4. Run the `md-davis residue` command passing the TOML file with the `--alignment` option to generate the pandas dataframes.

```
md-davis residue 2VH7_data.h5 2GV1_data.h5 2ACY_data.h5 1URR_data.h5 --alignment AcP_
↳ alignment_input.toml -o AcP_residue_data_aligned.p
```



- Plot the aligned data frames.

```
md-davis plot_residue AcP_residue_data_aligned.p -o AcP_residue_data_aligned.html
```

## Hydrogen Bond Matrix

- Calculate the hydrogen bonds using the hbond utility in GROMACS.

```
gmx hbond -f 2VH7/2VH7_trajectory.xtc -s 2VH7/2VH7_md.tpr -num 2VH7/2VH7_hbnum.svg -hbm_
↳ 2VH7/2VH7_hb_matrix -hbn 2VH7/2VH7_hb_index
```

- Open the output index file 2VH7\_hb\_index.ndx and scroll down to find the title of the last section containing the list of hydrogen bonds, which is hbonds\_Protein in this case, as shown below:

```
1235 1243 1244 1248 1249 1260 1261 1263 1264 1280 1285 1286 1301 1302 1320
1321 1339 1340 1356 1361 1362 1380 1381 1389 1390 1392 1393 1410 1413 1414
1421 1424 1425 1433 1434 1436 1437 1456 1457 1468 1469 1473 1474 1492 1493
1508 1509 1525 1530 1531
[ hbonds_Protein ]
      9      10      35
     36      37     773
     55      56    1285
     62      63     725
```

- Calculate the occurrence of each hydrogen bond:

```
md-davis hbond -x 2VH7/2VH7_hb_matrix.xpm -i 2VH7/2VH7_hb_index.ndx -s 2VH7/2VH7_
↳ structure.pdb -g hbonds_Protein --save_pickle 2VH7/2VH7_hbonds.p
```

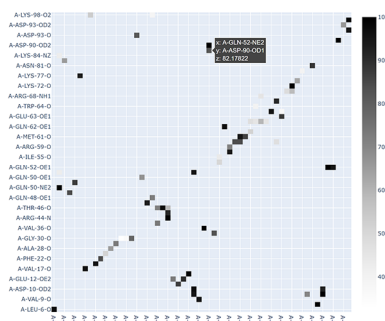
- Plot the hydrogen bonds matrix

```
md-davis plot_hbond --percent --total_frames 101 --cutoff 33 -o 2VH7_hbond_matrix.html_
↳ 2VH7/2VH7_hbonds.p
```

The above command plots the percentage of the H-bonds, which is calculated for each H-bond as follows:

number of frames the H-bond is observed / total number of frames \* 100

The cutoff of 33 % is set to plot only those H-bonds whose occurrence is greater than 33 %.



## 1.5.2 Sickle and Normal Oxyhemoglobin

Coming soon

## 1.6 Graphical User Interface

The GUI (graphical user interface) can be invoked by running the following command from a terminal:

```
md-davis-gui
```

Each MD DaVis functionality is in a separate tab of the GUI. Select the appropriate tab and click the image to see detailed usage.

Landscape

Electrodynamics

Collate

Sequence

Linux

Windows

MD DaVis

**Landscape** | Electrodynamics | Collate | Sequence

☐ Histogram ☒ Free Energy Temperature (in K)

☒ Perspective ☐ Orthographic

Title

Shape of each landscape

X-bins  Y-bins  ☒ Use Common Ranges

Axis Labels

X  Y  Z

Size and layout of the plot

Width  Height

Save landscapes to

Output HTML file

Input

X

Y

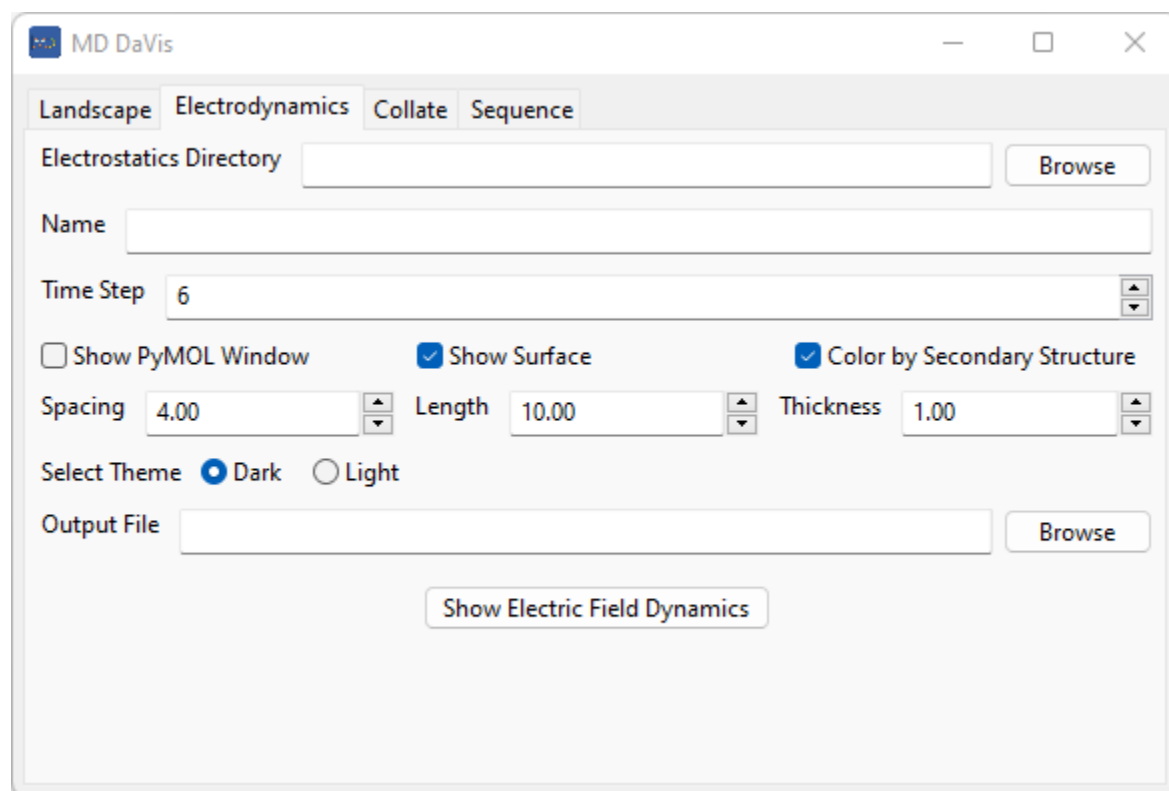
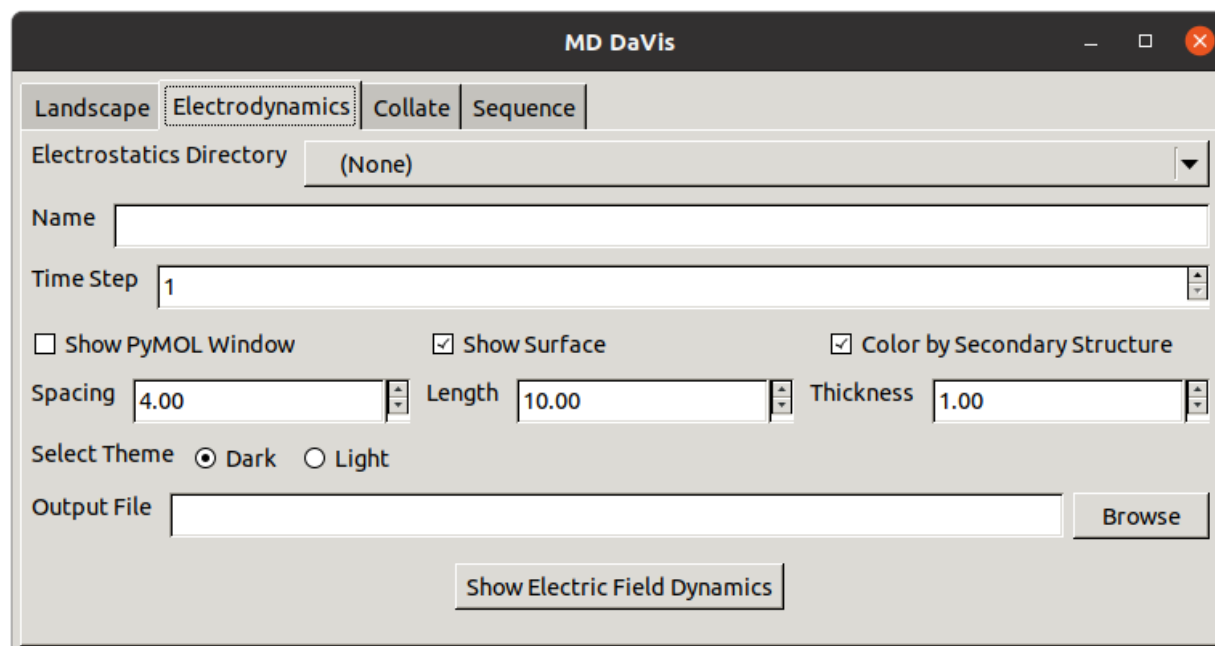
Name  Label

The image shows the MD DaVis software interface with the 'Landscape' tab selected. The interface includes several configuration sections:

- Navigation:** Landscape (selected), Electrostatics, Collate, Sequence.
- Plot Type:** ☐ Histogram, ☒ Free Energy. Temperature (in K) is set to 300.00.
- View:** ☒ Perspective, ☐ Orthographic.
- Title:** A text input field.
- Shape of each landscape:** X-bins: 100, Y-bins: 100, ☒ Use Common Ranges.
- Axis Labels:** X: RMSD (in nm), Y: Radius of Gyration (in nm), Z: Free Energy (in kJ mol<sup>-1</sup>).
- Size and layout of the plot:** Width: 1920, Height: 1080.
- Save landscapes to:** A text field with a 'Browse' button.
- Output HTML file:** A text field with a 'Browse' button.
- Create Free Energy Landscape:** A button.
- Input:** X: A text field with a 'Browse' button. Y: A text field with a 'Browse' button. Name: A text field. Label: A text field. Minus (-) and Plus (+) buttons are also present.

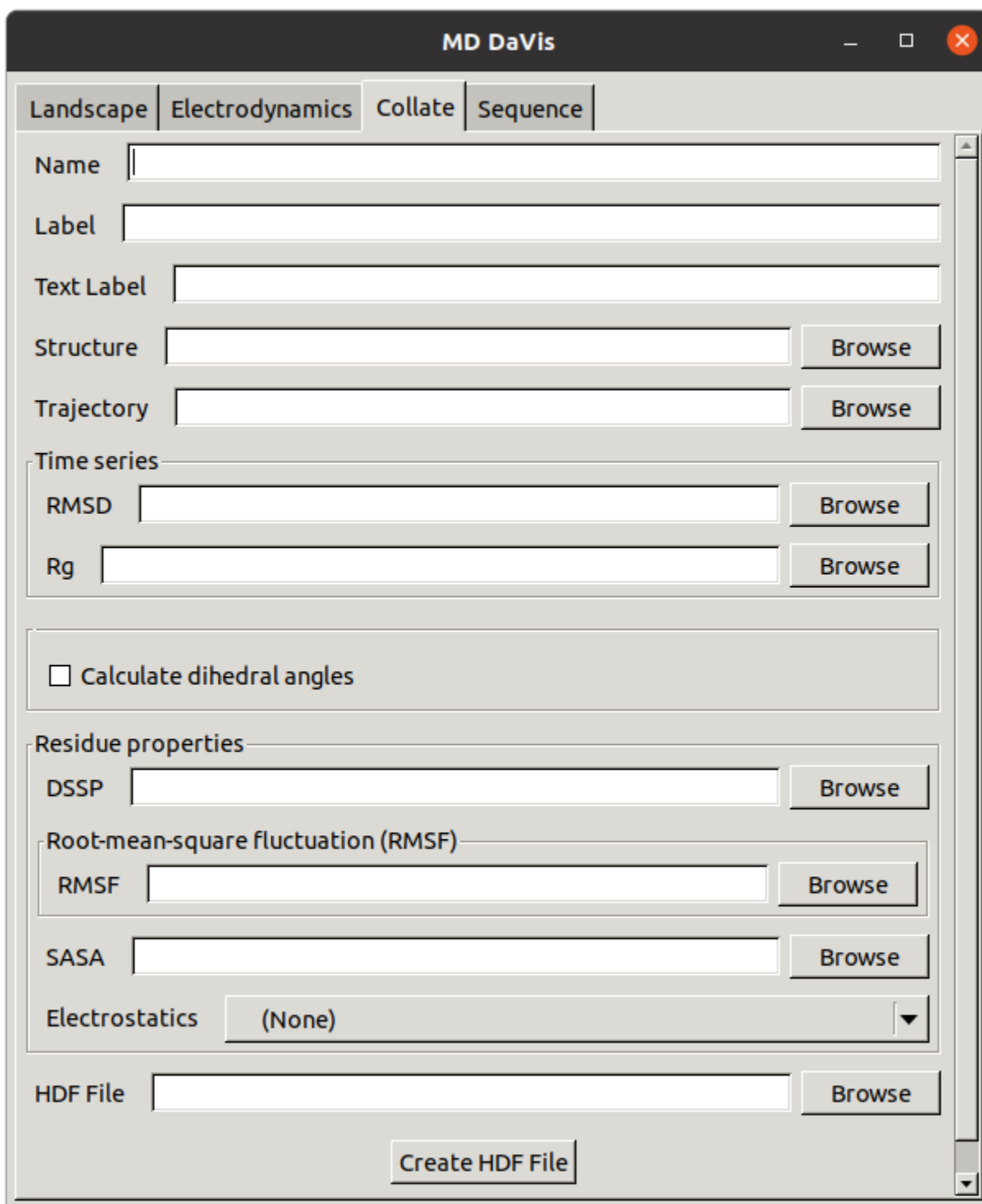
Linux

Windows



Linux

Windows



The image shows a screenshot of the MD DaVis software window. The window has a title bar with the text "MD DaVis" and standard window controls (minimize, maximize, close). Below the title bar is a tabbed interface with four tabs: "Landscape", "Electrostatics", "Collate", and "Sequence". The "Electrostatics" tab is currently selected. The main content area of the window contains several input fields and buttons. At the top, there are three text input fields labeled "Name", "Label", and "Text Label". Below these are two rows of input fields with "Browse" buttons: "Structure" and "Trajectory". A section titled "Time series" contains two rows of input fields with "Browse" buttons: "RMSD" and "Rg". Below this is a checkbox labeled "Calculate dihedral angles". A section titled "Residue properties" contains three rows of input fields with "Browse" buttons: "DSSP", "Root-mean-square fluctuation (RMSF)", and "SASA". Below these is a dropdown menu labeled "Electrostatics" with the value "(None)" selected. At the bottom, there is an input field for "HDF File" with a "Browse" button. A "Create HDF File" button is located at the bottom center of the window.

MD DaVis

Landscape | **Electrostatics** | Collate | Sequence

Name

Label

Text Label

Structure  Browse

Trajectory  Browse

Time series

RMSD  Browse

Rg  Browse

☐ Calculate dihedral angles

Residue properties

DSSP  Browse

Root-mean-square fluctuation (RMSF)

RMSF  Browse

SASA  Browse

Electrostatics (None) ▼

HDF File  Browse

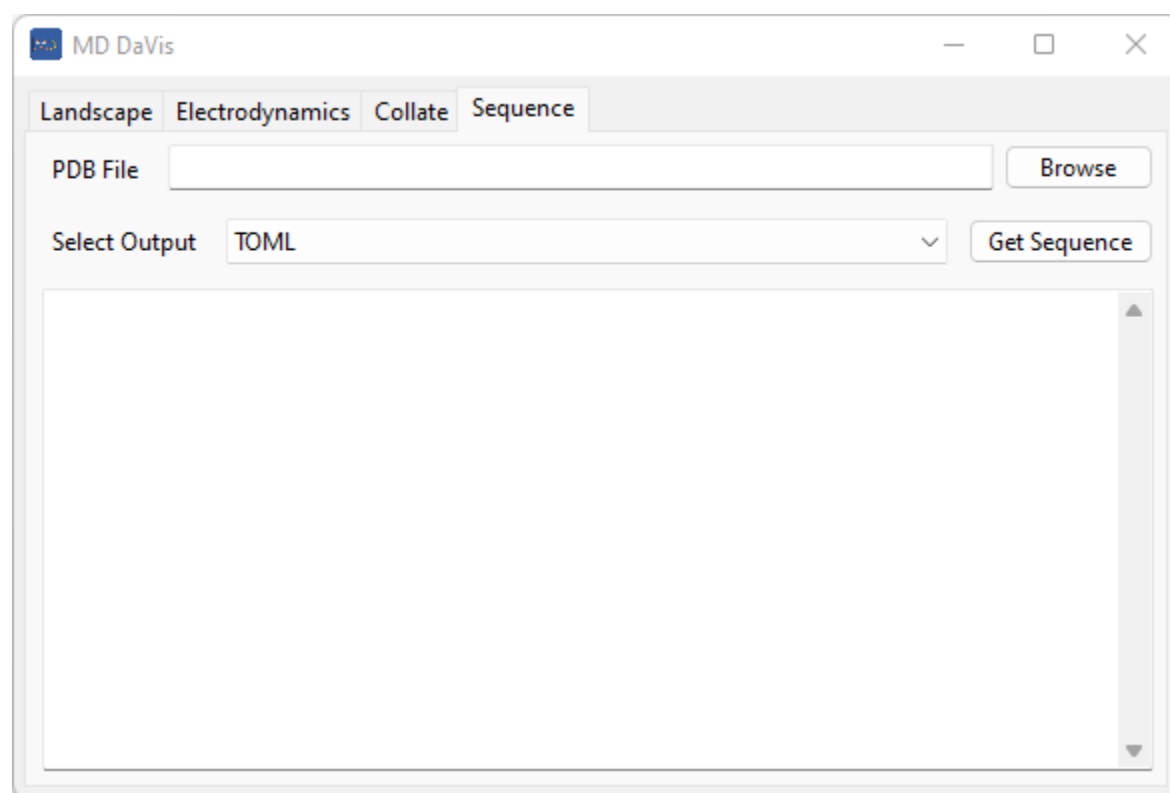
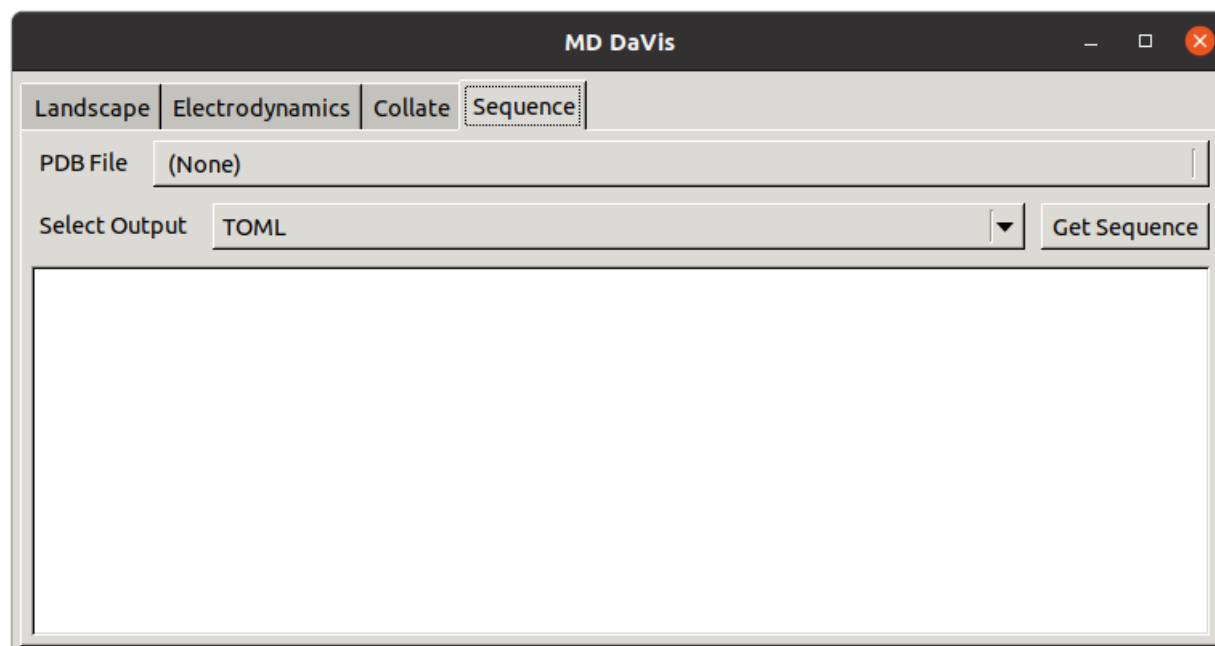
Create HDF File

The image shows the MD DaVis software window with the 'Collate' tab selected. The window has a title bar with the MD DaVis logo and standard window controls. The 'Collate' tab is active, showing a form with the following fields and buttons:

- Name:** A text input field.
- Label:** A text input field.
- Text Label:** A text input field.
- Structure:** A text input field with a 'Browse' button to its right.
- Trajectory:** A text input field with a 'Browse' button to its right.
- Time series:** A section containing:
  - RMSD:** A text input field with a 'Browse' button to its right.
  - Rg:** A text input field with a 'Browse' button to its right.
- Calculate dihedral angles:** A checkbox that is checked.
- Chunk Size:** A text input field containing the value '1000'.
- Residue properties:** A section containing:
  - DSSP:** A text input field with a 'Browse' button to its right.
  - Root-mean-square fluctuation (RMSF):** A section containing a text input field for 'RMSF' with a 'Browse' button to its right.
  - SASA:** A text input field with a 'Browse' button to its right.
  - Electrostatics:** A text input field with a 'Browse' button to its right.
- HDF File:** A text input field with a 'Browse' button to its right.
- Create HDF File:** A button at the bottom center of the window.

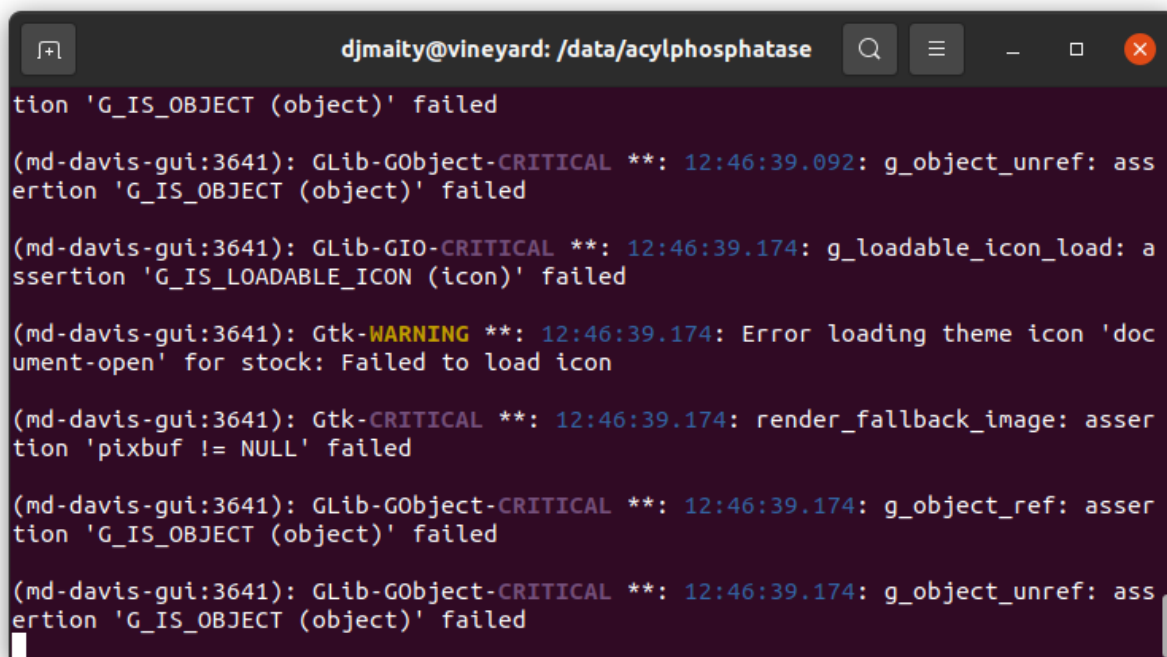
Linux

Windows



**Note:** The GTK+ windowing system in Linux may issue a warning as shown below. This does not affect the functionality of MD DaVis and can be safely ignored.





A screenshot of a terminal window titled 'djmaity@vineyard: /data/acylphosphatase'. The terminal displays several error messages from the MD DaVis GUI. The messages are as follows:

```
tion 'G_IS_OBJECT (object)' failed

(md-davis-gui:3641): GLib-GObject-CRITICAL **: 12:46:39.092: g_object_unref: assertion 'G_IS_OBJECT (object)' failed

(md-davis-gui:3641): GLib-GIO-CRITICAL **: 12:46:39.174: g_loadable_icon_load: assertion 'G_IS_LOADABLE_ICON (icon)' failed

(md-davis-gui:3641): Gtk-WARNING **: 12:46:39.174: Error loading theme icon 'document-open' for stock: Failed to load icon

(md-davis-gui:3641): Gtk-CRITICAL **: 12:46:39.174: render_fallback_image: assertion 'pixmap != NULL' failed

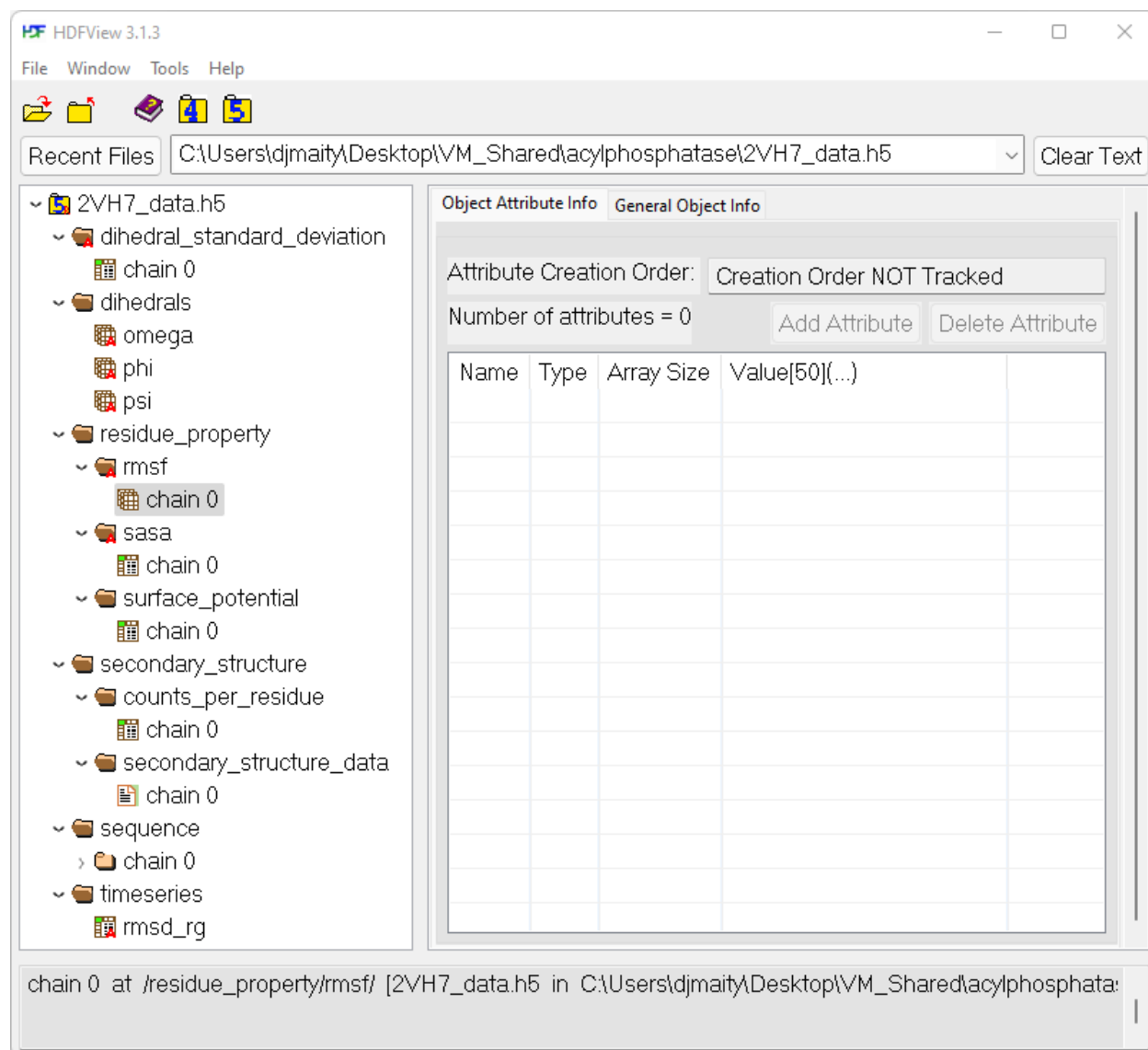
(md-davis-gui:3641): GLib-GObject-CRITICAL **: 12:46:39.174: g_object_ref: assertion 'G_IS_OBJECT (object)' failed

(md-davis-gui:3641): GLib-GObject-CRITICAL **: 12:46:39.174: g_object_unref: assertion 'G_IS_OBJECT (object)' failed
```

This may be annoying and bury some the important messages and output from MD DaVis. Please pay attention to the terminal used to launch the MD DaVis GUI. We will redirect the output to a better interface in a future release.

### 1.6.1 HDFView

The HDFView (<https://www.hdfgroup.org/downloads/hdfview/>) GUI program from [HDF Group](#) can be used to inspect and modify the HDF files created by MD DaVis. The modified file can be then provided to MD DaVis plotting commands. See [HDF Files](#) for details.



## 1.7 Command Line Interface

MD DaVis provides the command line tool `md-davis` for analysis and visualization of data from MD simulations.

### 1.7.1 Getting Help

The list of available commands can be seen with `--help` or `-h`:

```
md-davis --help
```

which provides the following output:

```
Usage: md-davis [OPTIONS] COMMAND [ARGS]...
```

(continues on next page)

(continued from previous page)

MD DaVis: A python package **for** comparative analysis of molecular dynamics trajectories

## Options:

--version Show the version **and** exit.  
 -h, --help Show this message **and** exit.

## Commands:

collate	Input TOML files.
electrodynamics	Click wrapper <b>for</b> get_electrodynamics.
electrostatics	Get the electrostatic potential on the surface...
hbond	Parse contacts evaluated by gmx hbonds
landscape	Plot free energy landscapes <b>from</b> .xvg files...
landscape_animate	
landscape_xvg	Wrapper function <b>for</b> Click
plot_hbond	H-bond <b>or</b> contact dataframe obtained <b>from</b> md_davis...
plot_residue	main function
residue	
sequence	Get the sequence of amino acid residues <b>from</b> a PDB file
xvg	Plot xmgrace (.xvg) files generated by GROMACS.

Similarly, the help for each command can also be accessed with --help or -h:

```
md-davis <command> -h
```

For example, md-davis xvg -h shows the help for [xvg](#) command:

Usage: md-davis xvg [OPTIONS] XVG\_FILES...

Plot xmgrace (.xvg) files generated by GROMACS.

## Arguments:

xvg\_files (Files): Input xmgrace (.xvg) files'

## Options:

-o, --output TEXT Output filename  
 --matplotlib / --plotly Use plotly **for** plotting  
 -h, --help Show this message **and** exit.

## 1.8 Reference

### 1.8.1 Python Module

To use MD DaVis as Python module:

```
import md_davis
```

## 1.8.2 CLI Reference

### md-davis

MD DaVis: A python package for comparative analysis of molecular dynamics trajectories

```
md-davis [OPTIONS] COMMAND [ARGS]...
```

### Options

#### --version

Show the version and exit.

### collate

Input TOML files.

```
md-davis collate [OPTIONS] INPUT_FILES...
```

### Arguments

#### INPUT\_FILES

Required argument(s)

### electrodynamics

Click wrapper for get\_electrodynamics. Otherwise GUI was giving an error

```
md-davis electrodynamics [OPTIONS] DIRECTORY
```

### Options

**-n, --name** <name>

**Required** Name for the PyMOL object to be created for the molecule. (No spaces!)

**--surface**

Show molecular surface

**--ss\_color**

Color the structure based on secondary structure

**--spacing** <spacing>

Spacing of field lines

**-t, --time\_step** <time\_step>

time step for frames to show

**-l, --length** <length>

length of field lines

**-w, --width** <width>  
length of field lines

**--light, --dark**  
Enable dark or light mode

**-o, --output** <output>  
Name for saving the PyMOL session

**--hide**  
Hide PyMOL window

## Arguments

### DIRECTORY

Required argument

## electrostatics

Get the electrostatic potential on the surface points generated by MSMS

```
md-davis electrostatics [OPTIONS] [PDB_FILES]...
```

## Options

**-m, --msms** <PATH>  
**Required** full path to MSMS executable

**-d, --delphi** <PATH>  
**Required** full path to Delphi executable

**-r, --radius** <PATH>  
path to radius file

**-c, --charge** <PATH>  
path to charge file

**-g, --grid\_size** <odd\_int>  
Grid size to use for Delphi calculation

**--surface**  
Calculate electrostatic potential on the surface

**--center**  
Center the grid for Delphi at the origin

**-o, --output** <PATH>  
Directory to place the output files.

## Arguments

### PDB\_FILES

Optional argument(s)

### hbond

Parse contacts evaluated by gmx hbonds

`md-davis hbond [OPTIONS]`

## Options

- x, --xpm\_file <.xpm>**  
Required Contact/H-bond file obtained from GROMACS
- i, --index\_file <.ndx>**  
Required Index file
- s, --structure <.pdb/.gro>**  
Required Structure file
- g, --group <string>**  
Required Group to match from index file to get the list of Contacts/H-bonds
- o, --output <.p>**  
Output Pickle file
- contacts, --hbonds**  
Select the type of input: contacts or H-bonds
- silent, --verbose**  
Choose silent to suppress the displaying to result to standard output
- save\_pickle <save\_pickle>**  
Save the output to a pickle file
- save\_hdf <save\_hdf>**  
Save the output to a HDF file
- save\_csv <save\_csv>**  
Save the output to a CSV file
- save\_matrix <save\_matrix>**  
Save a file containing the contact matrix
- save\_pdb <save\_pdb>**  
Save a pdb file with percentage of frames in the B-factor column

## landscape

Plot free energy landscapes from .xvg files generated by GROMACS or HDF files created by md\_davis collate

```
md-davis landscape [OPTIONS] [HDF_FILES]...
```

### Options

- x** <x>  
Data to plot on x-axis
- y** <y>  
Data to plot on y-axis
- n, --name** <name>  
Names of each landscape object
- l, --label** <label>  
Label to show in plots
- c, --common**  
Use common ranges for all the landscapes
- T, --temperature** <temperature>  
Temperature of the system. If this option is provided the energy landscape is calculated using Boltzmann inversion, else only the histogram is evaluated
- shape** <'X-bins', 'Y-bins'>  
Number of bins in the X and Y direction
- o, --output** <output>  
Name for the output HTML file containing the plots
- s, --save** <FILENAME>  
Name for HDF5 file to save the landscapes
- b, --begin** <int>  
Starting index for the data to include
- e, --end** <int>  
Last index for the data to include
- limits** <limits>  
A dictionary containing the limits for x, y, and x axes
- width** <int>  
Width of the plot
- height** <int>  
Height of the plot
- layout** <layout>  
Layout of subplots
- title** <str>  
Title for the figure

**--axis\_labels** <axis\_labels>

A dictionary of strings specifying the labels for the x, y and z-axis. For example: dict(x='RMSD (in nm)', y='Rg (in nm)', z='Free Energy (kJ mol<sup>-1</sup>)<br>')

**--orthographic, --perspective**

Orthographic projection for 3D plots

**--font** <str>

Font style

**--font\_size** <int>

Size of text and labels in the plot

**--dtick** <dtick>

Tick interval on each axes

**-p, --plot**

Plot the precomputed energy landscape

**-d, --dict** <dict>

Dictionary containing input data

**--order** <order>

Array specifying the order in which to plot the landscapes

**--columns** <columns>

Columns to use (start from second column = 1)

## Arguments

### HDF\_FILES

Optional argument(s)

### landscape\_animate

md-davis landscape_animate [OPTIONS] HDF_FILE
---

## Options

**--static, --animate**

Chose if the trajectory should be plotted or animated on the surface

**-o, --output** <output>

Name for the output HTML file containing the plots

**--select** <str>

Select which landscape to plot in a file containing multiple landscapes

**--begin** <int>

Starting index for the data to include

**--end** <int>

Last index for the data to include



**--step** <int>  
step size to iterate over the data

**--title** <str>  
Title for the figure

**--orthographic, --perspective**  
Orthographic projection for 3D plots

**--axis\_labels** <axis\_labels>  
A dictionary of strings specifying the labels for the x, y and z-axis

**--width** <int>  
Width of the plot

**--height** <int>  
Height of the plot

**--font** <str>  
Font style

**--font\_size** <int>  
Size of text and labels in the plot

**--marker\_size** <int>  
Size of markers in the plot

**--dtick** <dtick>  
Tick interval on each axes

**--hide\_surface** <hide\_surface>  
Hide the energy landscape surface

**--camera** <dict(eye=dict(>  
Dictionary to specify the camera for plotly

## Arguments

### HDF\_FILE

Required argument

### landscape\_xvg

Wrapper function for Click

```
md-davis landscape_xvg [OPTIONS]
```

## Options

- x <x>**  
Required Data to plot on x-axis
- y <y>**  
Required Data to plot on y-axis
- n, --name <name>**  
Required Names of each landscape object
- l, --label <label>**  
Required Label to show in plots
- c, --common**  
Use common ranges for all the landscapes
- T, --temperature <temperature>**  
Temperature of the system. If this option is provided the energy landscape is calculated using Boltzmann inversion, else only the histogram is evaluated
- o, --output <output>**  
Name for the output HTML file containing the plots
- shape <'X-bins', 'Y-bins'>**  
Number of bins in the X and Y direction
- b, --begin <int>**  
Starting index for the data to include
- e, --end <int>**  
Last index for the data to include
- limits <limits>**  
A dictionary containing the limits for x, y, and x axes
- s, --save <FILENAME>**  
Name for HDF5 file to save the landscapes
- title <str>**  
Title for the figure
- orthographic, --perspective**  
Orthographic projection for 3D plots
- width <int>**  
Width of the plot
- height <int>**  
Height of the plot
- font <str>**  
Font style
- font\_size <int>**  
Size of text and labels in the plot

**--dtick** <dtick>

Tick interval on each axes

**--axis\_labels** <axis\_labels>

A dictionary of strings specifying the labels for the x, y and z-axis. For example: dict(x='RMSD (in nm)', y='Rg (in nm)', z='Free Energy (kJ mol<sup>-1</sup>)<br>')

**--layout** <layout>

Layout of subplots

**--columns** <columns>

Columns to use (start from second column = 1)

## plot\_hbond

H-bond or contact dataframe obtained from md\_davis hbond

```
md-davis plot_hbond [OPTIONS] HBOND_FILE
```

## Options

**-t, --total\_frames** <total\_frames>

**Required** Total number of frames in the trajectory used for H-bond/Contact evaluation.

**-c, --cutoff** <cutoff>

Cutoff for percentage or number of Hbonds above which to include in the plot.

**-o, --output** <output>

Output filename

**--percent**

Plot the percentages

**--title** <title>

Title for the plot

## Arguments

**HBOND\_FILE**

Required argument

## plot\_residue

main function

```
md-davis plot_residue [OPTIONS] PICKLE_FILE
```

## Options

**--start** <start>  
Starting index of the plot

**--end** <end>  
Last index of the plot

**-o, --output** <output>  
output HTML file

**--width** <width>  
Width of the plot in pixels

**--height** <height>  
Height of the plot in pixels

**-t, --title** <title>  
title for the plot

**-l, --line\_color** <line\_color>  
Array of colors for lines

**-f, --fill\_color** <fill\_color>  
Array of colors for shaded error bars

**--show\_markers, --hide\_markers**  
Show the markers on the line plots

**--show\_error\_bars, --hide\_error\_bars**  
show error bars for electrostatic potential and SASA

## Arguments

**PICKLE\_FILE**  
Required argument

## potential\_into\_pdb

`md-davis potential_into_pdb [OPTIONS] POTENTIAL_FILE PDB_FILE OUTPUT_FILENAME`

## Options

**--atomic\_potentials, --residue\_potential**  
Write electrostatic potential per atom into the mean

**--save\_mean\_potentials, --save\_total\_potentials**  
Save total or mean electrostatic potential per residue or atom

**--rescale**  
Rescale and limit the electrostatic potential to -9.99 to 9.99 so that the b-factor columns strictly conforms to the PDB format.

**--fillna**

Fill nan values in the b-factor column with 0.00

**Arguments****POTENTIAL\_FILE**

Required argument

**PDB\_FILE**

Required argument

**OUTPUT\_FILENAME**

Required argument

**residue**

```
md-davis residue [OPTIONS] INPUT_FILES...
```

**Options**

**-o, --output** <output>

**-a, --annotations** <annotations>

Annotations

**--alignment** <alignment>

Alignment file to align the residue data

**Arguments****INPUT\_FILES**

Required argument(s)

**sequence**

Get the sequence of amino acid residues from a PDB file

Determines the sequence from the order of amino acid residues in the Protein Data Bank (PDB) file. The SEQRES records in the PDB file are not used, because it is sometimes different from the actual list of amino acids in the PDB file.

It is assumed that a PDB file used for dynamics will not contain any missing residues. The sequence gets truncated at the first missing residue

```
md-davis sequence [OPTIONS] STRUCTURE
```

## Options

**-l, --label** <label>

**-r, --return-type** <return\_type>

*dict* returns a dictionary containing chain labels as keys and each chain's sequence as values. *fasta* returns each chain's sequence separately in FASTA format.

**Options** dict | fasta | modeller | toml

## Arguments

### STRUCTURE

Required argument

## xvg

Plot xmgrace (.xvg) files generated by GROMACS.

Arguments:

xvg\_files (Files): Input xmgrace (.xvg) files'

`md-davis xvg [OPTIONS] XVG_FILES...`

## Options

**-o, --output** <output>

Output filename

**--matplotlib, --plotly**

Use plotly for plotting

## Arguments

### XVG\_FILES

Required argument(s)

## INDICES AND TABLES

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





## Symbols

- T
  - md-davis-landscape command line option, [51](#)
  - md-davis-landscape\_xvg command line option, [54](#)
- alignment
  - md-davis-residue command line option, [57](#)
- animate
  - md-davis-landscape\_animate command line option, [52](#)
- annotations
  - md-davis-residue command line option, [57](#)
- atomic\_potentials
  - md-davis-potential\_into\_pdb command line option, [56](#)
- axis\_labels
  - md-davis-landscape command line option, [51](#)
  - md-davis-landscape\_animate command line option, [53](#)
  - md-davis-landscape\_xvg command line option, [55](#)
- begin
  - md-davis-landscape command line option, [51](#)
  - md-davis-landscape\_animate command line option, [52](#)
  - md-davis-landscape\_xvg command line option, [54](#)
- camera
  - md-davis-landscape\_animate command line option, [53](#)
- center
  - md-davis-electrostatics command line option, [49](#)
- charge
  - md-davis-electrostatics command line option, [49](#)
- columns
  - md-davis-landscape command line option, [52](#)
- md-davis-landscape\_xvg command line option, [55](#)
- common
  - md-davis-landscape command line option, [51](#)
  - md-davis-landscape\_xvg command line option, [54](#)
- contacts
  - md-davis-hbond command line option, [50](#)
- cutoff
  - md-davis-plot\_hbond command line option, [55](#)
- dark
  - md-davis-electrodynamics command line option, [49](#)
- delphi
  - md-davis-electrostatics command line option, [49](#)
- dict
  - md-davis-landscape command line option, [52](#)
- dtick
  - md-davis-landscape command line option, [52](#)
  - md-davis-landscape\_animate command line option, [53](#)
  - md-davis-landscape\_xvg command line option, [54](#)
- end
  - md-davis-landscape command line option, [51](#)
  - md-davis-landscape\_animate command line option, [52](#)
  - md-davis-landscape\_xvg command line option, [54](#)
  - md-davis-plot\_residue command line option, [56](#)
- fill\_color
  - md-davis-plot\_residue command line option, [56](#)
- fillna
  - md-davis-potential\_into\_pdb command

line option, [56](#)

--font  
md-davis-landscape command line option, [52](#)  
md-davis-landscape\_animate command line option, [53](#)  
md-davis-landscape\_xvg command line option, [54](#)

--font\_size  
md-davis-landscape command line option, [52](#)  
md-davis-landscape\_animate command line option, [53](#)  
md-davis-landscape\_xvg command line option, [54](#)

--grid\_size  
md-davis-electrostatics command line option, [49](#)

--group  
md-davis-hbond command line option, [50](#)

--hbonds  
md-davis-hbond command line option, [50](#)

--height  
md-davis-landscape command line option, [51](#)  
md-davis-landscape\_animate command line option, [53](#)  
md-davis-landscape\_xvg command line option, [54](#)  
md-davis-plot\_residue command line option, [56](#)

--hide  
md-davis-electrodynamics command line option, [49](#)

--hide\_errorBars  
md-davis-plot\_residue command line option, [56](#)

--hide\_markers  
md-davis-plot\_residue command line option, [56](#)

--hide\_surface  
md-davis-landscape\_animate command line option, [53](#)

--index\_file  
md-davis-hbond command line option, [50](#)

--label  
md-davis-landscape command line option, [51](#)  
md-davis-landscape\_xvg command line option, [54](#)  
md-davis-sequence command line option, [58](#)

--layout  
md-davis-landscape command line option, [51](#)  
md-davis-landscape\_xvg command line option, [55](#)

--length  
md-davis-electrodynamics command line option, [48](#)

--light  
md-davis-electrodynamics command line option, [49](#)

--limits  
md-davis-landscape command line option, [51](#)  
md-davis-landscape\_xvg command line option, [54](#)

--line\_color  
md-davis-plot\_residue command line option, [56](#)

--marker\_size  
md-davis-landscape\_animate command line option, [53](#)

--matplotlib  
md-davis-xvg command line option, [17](#), [58](#)

--msms  
md-davis-electrostatics command line option, [49](#)

--name  
md-davis-electrodynamics command line option, [48](#)  
md-davis-landscape command line option, [51](#)  
md-davis-landscape\_xvg command line option, [54](#)

--order  
md-davis-landscape command line option, [52](#)

--orthographic  
md-davis-landscape command line option, [52](#)  
md-davis-landscape\_animate command line option, [53](#)  
md-davis-landscape\_xvg command line option, [54](#)

--output  
md-davis-electrodynamics command line option, [49](#)  
md-davis-electrostatics command line option, [49](#)  
md-davis-hbond command line option, [50](#)  
md-davis-landscape command line option, [51](#)  
md-davis-landscape\_animate command line option, [52](#)  
md-davis-landscape\_xvg command line option, [54](#)  
md-davis-plot\_hbond command line option,

---

```

    55
md-davis-plot_residue command line
    option, 56
md-davis-residue command line option, 57
md-davis-xvg command line option, 17, 58
--percent
    md-davis-plot_hbond command line option,
    55
--perspective
    md-davis-landscape command line option,
    52
    md-davis-landscape_animate command line
    option, 53
    md-davis-landscape_xvg command line
    option, 54
--plot
    md-davis-landscape command line option,
    52
--plotly
    md-davis-xvg command line option, 17, 58
--radius
    md-davis-electrostatics command line
    option, 49
--rescale
    md-davis-potential_into_pdb command
    line option, 56
--residue_potential
    md-davis-potential_into_pdb command
    line option, 56
--return-type
    md-davis-sequence command line option, 58
--save
    md-davis-landscape command line option,
    51
    md-davis-landscape_xvg command line
    option, 54
--save_csv
    md-davis-hbond command line option, 50
--save_hdf
    md-davis-hbond command line option, 50
--save_matrix
    md-davis-hbond command line option, 50
--save_mean_potentials
    md-davis-potential_into_pdb command
    line option, 56
--save_pdb
    md-davis-hbond command line option, 50
--save_pickle
    md-davis-hbond command line option, 50
--save_total_potentials
    md-davis-potential_into_pdb command
    line option, 56
--select
    md-davis-landscape_animate command line
    option, 52
--shape
    md-davis-landscape command line option,
    51
    md-davis-landscape_xvg command line
    option, 54
--show_error_bars
    md-davis-plot_residue command line
    option, 56
--show_markers
    md-davis-plot_residue command line
    option, 56
--silent
    md-davis-hbond command line option, 50
--spacing
    md-davis-electrodynamics command line
    option, 48
--ss_color
    md-davis-electrodynamics command line
    option, 48
--start
    md-davis-plot_residue command line
    option, 56
--static
    md-davis-landscape_animate command line
    option, 52
--step
    md-davis-landscape_animate command line
    option, 52
--structure
    md-davis-hbond command line option, 50
--surface
    md-davis-electrodynamics command line
    option, 48
    md-davis-electrostatics command line
    option, 49
--temperature
    md-davis-landscape command line option,
    51
    md-davis-landscape_xvg command line
    option, 54
--time_step
    md-davis-electrodynamics command line
    option, 48
--title
    md-davis-landscape command line option,
    51
    md-davis-landscape_animate command line
    option, 53
    md-davis-landscape_xvg command line
    option, 54
    md-davis-plot_hbond command line option,
    55

```

md-davis-plot_residue command line option, 56	md-davis-hbond command line option, 50
--total_frames	-i md-davis-hbond command line option, 50
md-davis-plot_hbond command line option, 55	-l md-davis-electrodynamics command line option, 48
--verbose	md-davis-landscape command line option, 51
md-davis-hbond command line option, 50	md-davis-landscape_xvg command line option, 54
--version	md-davis-plot_residue command line option, 56
md-davis command line option, 48	md-davis-sequence command line option, 58
--width	-m md-davis-electrostatics command line option, 49
md-davis-electrodynamics command line option, 48	-n md-davis-electrodynamics command line option, 48
md-davis-landscape command line option, 51	md-davis-landscape command line option, 51
md-davis-landscape_animate command line option, 53	md-davis-landscape_xvg command line option, 54
md-davis-landscape_xvg command line option, 54	-o md-davis-electrodynamics command line option, 49
md-davis-plot_residue command line option, 56	md-davis-electrostatics command line option, 49
--xpm_file	md-davis-hbond command line option, 50
md-davis-hbond command line option, 50	md-davis-landscape command line option, 51
-a md-davis-residue command line option, 57	md-davis-landscape_xvg command line option, 54
-b md-davis-landscape command line option, 51	md-davis-electrodynamics command line option, 49
md-davis-landscape_xvg command line option, 54	md-davis-electrostatics command line option, 49
-c md-davis-electrostatics command line option, 49	md-davis-hbond command line option, 50
md-davis-landscape command line option, 51	md-davis-landscape command line option, 51
md-davis-landscape_xvg command line option, 54	md-davis-landscape_animate command line option, 52
md-davis-plot_hbond command line option, 55	md-davis-landscape_xvg command line option, 54
-d md-davis-electrostatics command line option, 49	md-davis-plot_hbond command line option, 55
md-davis-landscape command line option, 52	md-davis-plot_residue command line option, 56
-e md-davis-landscape command line option, 51	md-davis-residue command line option, 57
md-davis-landscape_xvg command line option, 54	md-davis-xvg command line option, 17, 58
-f md-davis-plot_residue command line option, 56	-p md-davis-landscape command line option, 52
-g md-davis-electrostatics command line option, 49	-r md-davis-electrostatics command line option, 49
	md-davis-sequence command line option, 58
	-s md-davis-hbond command line option, 50
	md-davis-landscape command line option, 51
	md-davis-landscape_xvg command line option, 54

-t  
 md-davis-electrodynamics command line option, 48  
 md-davis-plot\_hbond command line option, 55  
 md-davis-plot\_residue command line option, 56

-w  
 md-davis-electrodynamics command line option, 48

-x  
 md-davis-hbond command line option, 50  
 md-davis-landscape command line option, 51  
 md-davis-landscape\_xvg command line option, 54

-y  
 md-davis-landscape command line option, 51  
 md-davis-landscape\_xvg command line option, 54

'Y-bins'>  
 md-davis-landscape command line option, 51  
 md-davis-landscape\_xvg command line option, 54

## D

DIRECTORY  
 md-davis-electrodynamics command line option, 49

## H

HBOND\_FILE  
 md-davis-plot\_hbond command line option, 55

HDF\_FILE  
 md-davis-landscape\_animate command line option, 53

HDF\_FILES  
 md-davis-landscape command line option, 52

## I

INPUT\_FILES  
 md-davis-collate command line option, 48  
 md-davis-residue command line option, 57

## M

md-davis command line option  
 --version, 48

md-davis-collate command line option  
 INPUT\_FILES, 48

md-davis-electrodynamics command line option  
 --dark, 49  
 --hide, 49  
 --length, 48  
 --light, 49  
 --name, 48  
 --output, 49  
 --spacing, 48  
 --ss\_color, 48  
 --surface, 48  
 --time\_step, 48  
 --width, 48  
 -l, 48  
 -n, 48  
 -o, 49  
 -t, 48  
 -w, 48  
 DIRECTORY, 49

md-davis-electrostatics command line option  
 --center, 49  
 --charge, 49  
 --delphi, 49  
 --grid\_size, 49  
 --msms, 49  
 --output, 49  
 --radius, 49  
 --surface, 49  
 -c, 49  
 -d, 49  
 -g, 49  
 -m, 49  
 -o, 49  
 -r, 49  
 PDB\_FILES, 50

md-davis-hbond command line option  
 --contacts, 50  
 --group, 50  
 --hbonds, 50  
 --index\_file, 50  
 --output, 50  
 --save\_csv, 50  
 --save\_hdf, 50  
 --save\_matrix, 50  
 --save\_pdb, 50  
 --save\_pickle, 50  
 --silent, 50  
 --structure, 50  
 --verbose, 50  
 --xpm\_file, 50  
 -g, 50  
 -i, 50  
 -o, 50  
 -s, 50

```
-x, 50
md-davis-landscape command line option
-T, 51
--axis_labels, 51
--begin, 51
--columns, 52
--common, 51
--dict, 52
--dtick, 52
--end, 51
--font, 52
--font_size, 52
--height, 51
--label, 51
--layout, 51
--limits, 51
--name, 51
--order, 52
--orthographic, 52
--output, 51
--perspective, 52
--plot, 52
--save, 51
--shape, 51
--temperature, 51
--title, 51
--width, 51
-b, 51
-c, 51
-d, 52
-e, 51
-l, 51
-n, 51
-o, 51
-p, 52
-s, 51
-x, 51
-y, 51
'Y-bins'>, 51
HDF_FILES, 52
md-davis-landscape_animate command line
  option
--animate, 52
--axis_labels, 53
--begin, 52
--camera, 53
--dtick, 53
--end, 52
--font, 53
--font_size, 53
--height, 53
--hide_surface, 53
--marker_size, 53
--orthographic, 53
--output, 52
--perspective, 53
--select, 52
--static, 52
--step, 52
--title, 53
--width, 53
-o, 52
HDF_FILE, 53
md-davis-landscape_xvg command line option
-T, 54
--axis_labels, 55
--begin, 54
--columns, 55
--common, 54
--dtick, 54
--end, 54
--font, 54
--font_size, 54
--height, 54
--label, 54
--layout, 55
--limits, 54
--name, 54
--orthographic, 54
--output, 54
--perspective, 54
--save, 54
--shape, 54
--temperature, 54
--title, 54
--width, 54
-b, 54
-c, 54
-e, 54
-l, 54
-n, 54
-o, 54
-s, 54
-x, 54
-y, 54
'Y-bins'>, 54
md-davis-plot_hbond command line option
--cutoff, 55
--output, 55
--percent, 55
--title, 55
--total_frames, 55
-c, 55
-o, 55
-t, 55
HBOND_FILE, 55
md-davis-plot_residue command line option
--end, 56
```

```

--fill_color, 56
--height, 56
--hide_errorBars, 56
--hide_markers, 56
--line_color, 56
--output, 56
--show_errorBars, 56
--show_markers, 56
--start, 56
--title, 56
--width, 56
-f, 56
-l, 56
-o, 56
-t, 56
PICKLE_FILE, 56
md-davis-potential_into_pdb command line
    option
--atomic_potentials, 56
--fillna, 56
--rescale, 56
--residue_potential, 56
--save_mean_potentials, 56
--save_total_potentials, 56
OUTPUT_FILENAME, 57
PDB_FILE, 57
POTENTIAL_FILE, 57
md-davis-residue command line option
--alignment, 57
--annotations, 57
--output, 57
-a, 57
-o, 57
INPUT_FILES, 57
md-davis-sequence command line option
--label, 58
--return-type, 58
-l, 58
-r, 58
STRUCTURE, 58
md-davis-xvg command line option
--matplotlib, 17, 58
--output, 17, 58
--plotly, 17, 58
-o, 17, 58
XVG_FILES, 17, 58

```

## O

```

OUTPUT_FILENAME
    md-davis-potential_into_pdb command
        line option, 57

```

## P

```

PDB_FILE

```

```

md-davis-potential_into_pdb command
    line option, 57
PDB_FILES
    md-davis-electrostatics command line
        option, 50
PICKLE_FILE
    md-davis-plot_residue command line
        option, 56
POTENTIAL_FILE
    md-davis-potential_into_pdb command
        line option, 57

```

## S

```

STRUCTURE
    md-davis-sequence command line option, 58

```

## X

```

XVG_FILES
    md-davis-xvg command line option, 17, 58

```