MCMap - A tool for Mapping Transient Complexes

Version 1.0

Johannes M. Foerster, Ina Poehner, and G. Matthias Ullmann*

Computational Biochemistry, University of Bayreuth, 95447 Bayreuth, Germany

E-mail: Matthias.Ullmann@uni-bayreuth.de

Phone: +49 (0)921 55 3545. Fax: +49 (0)921 55 3071

MANUAL

If you use this software for a published work, please cite:

Johannes M. Foerster, Ina Poehner, and G. Matthias Ullmann:

MCMap – A Computational Tool for Mapping Energy Landscapes of Transient Protein-Protein Interactions *submitted*

Contents

1 Introduction

The program MCMap uses a rigid-body protein-protein docking approach, which can be used to map energy landscapes of transient protein complexes resulting in a global distribution of ligand-receptor complexes. Since the ligand is sampling the 3-dimensional space around the receptor, MCMap can even be applied if little is known about the binding site. In order to evaluate the attraction or repulsion, long-range electrostatic interactions are probed and calculated, followed by a random translation and rotation move of the ligand. The detailed algorithm is described in Foerster et al.[1]

In MCMap, the ligand moves randomly in the electrostatic field of the receptor. After every move, the electrostatic interaction energy of the receptor and the ligand is calculated. The acceptance of each Monte Carlo (MC) move is done according to the Metropolis criterion.[2] If the the energy difference of the new MC move is negative, the move is accepted. If ΔE_s^{int} is positive, it is Boltzmann weighted and compared to a random number. Is the random number smaller than the weighted energy difference, then the MC move is accepted, else the move is rejected and a new MC move is performed:

$$\Delta E_s^{int} = \begin{cases} < 0 & \text{accept} \\ > 0 & \begin{cases} \text{Ran}() < e^{\frac{-\Delta E}{RT}} & \text{accept} \\ \text{Ran}() > e^{\frac{-\Delta E}{RT}} & \text{reject} \end{cases} \end{cases} \tag{1}$$

Accepted orientations are saved as coordinate transformation matrices, which are required for generating the ensemble orientations from the initial structure. This way of saving the various orientations saves storage space compared to a storage of full structures. Moreover, it is easily possible to use only a subset of the original structure, for instance only the prosthetic groups. With this, orientations on a "substructural" level can be generated. In the following sections, the installation, the usage and the structure of input files for MCMap are described. A simulation of cytochrome c and cytochrome c peroxidase from *Saccharomyces cerevisiae* is provided as an example.

2 Installation

MCMap was written and tested on Ubuntu 16.04. It will most likely run on any UNIX-like system. However, we do not have any experience on other operating systems. Make sure to have the following libraries installed for the compiling:

```
zlib1g-dev
libc6-dev-i386
```

To unpack and compile the program, please use the following commands in UNIX:

```
tar xvfz mcmap.V1.tgz
cd  mcmap.V1
sh install.sh
```

3 Usage

The program can be executed with the following command:

```
mcmap [input file]
```

In the input file, all simulation settings of the program are defined. Lines starting with '!' or '#' are comment lines. The following variables have to be defined:

receptor Structure input file of the receptor. The structure file has to be in PQRM format consisting of a PQR file format with one additional column holding the mass of the atoms.

ligand Structure input file of the ligand. The structure file has to be in PQRM format consisting of a PQR file format with one additional column holding the mass of the atoms.

number_of_runs Defines how often the docking simulation is restarted. With each run, the ligand is placed at the inner radius sphere R_{in} in order to initiate a new docking procedure.

mcsteps Number of Monte Carlo steps of a single run.

rad_in Defines the radius R_{in} (in Å) around the center of mass (CM) of the receptor from which the ligand starts the docking procedure. At this distance, ideally, nearly no electrostatic potential of the receptor should be present or the electrostatic potential should be centrosymmetric.

rad_out Defines the radius R_{out} (in Å) around the CM of the receptor where the ligand and the receptor are considered as too far away from each other for an interaction.

temperature Temperature of the simulation in Kelvin. Used in the Metropolis criterion.

pot_in Fine electrostatic potential file of the receptor. The grid spacing of the fine grid has to be smaller than the one of the coarse potential and has to fulfill certain size criteria (see separation parameter). The potential has to be in OpenDX format and in units of kcal/mol. The potential can be generated with APBS.

pot_out Coarse electrostatic potential file of the receptor. The grid has to be in OpenDX format and in units of kcal/mol. The potential can be generated with APBS.

nprint If a positive value is defined, a trajectory file is generated. If the value is < 0, no trajectory file is generated. A trajectory file is an orientation file, where every accepted MC step is included.

nprintdens Number of steps after which a density point is generated in the docking density file (OpenDX file). This value is introduced to adjust the sampling density.

nprintdist Number of steps after which a structure is printed to the orientation file. This value is introduced to adjust the sampling density.

istuck Number of steps after which a run is restarted if the position of the ligand does not change anymore (always rejected). The ligand is assumed to be caught in a local minimum. In order to save simulation time, a new docking run is initiated.

dev_trans Represents the maximal change of ligand translation (shift in x / y / z direction) in Å in a MC step.

dev_rot Represents the maximal change of ligand rotation (rotation in x / y / z direction) in degrees in a MC step.

excl Name of the exclusion grid file. The exclusion grid prohibits collisions of the ligand with the receptor and consists of a regular grid around the surface of the receptor. As soon as a collision is monitored, a new MC run is initiated. The exclusion grid can be generated with the tool `make-excl`.

traj Name for the output file of the docking trajectory.

structdist Name for the orientation file. The orientation file includes the energies and coordinate transformations of the accepted docking steps. The coordinate transformations can be used to generate the orientations of the complex from the initial structures for an individual or a number of MC steps (see utility program `print-coor` in section 5.1).

separation Distance from which the step size is reduced to 10% of `dev_trans` and `dev_rot` and at which the energy calculation switches from the coarse potential grid to the fine potential grid. The exclusion grid is also tested starting from this distance. The minimum separation can be calculated as follows:

$$max(|\text{dist}(\text{CM}_{ligand})|) + max(|\text{dist}(\text{CM}_{receptor})|) + 4.0 \text{ Å} \qquad (2)$$

If the separation is not set, it will automatically be calculated with equation 2.

lowenergy Defines a low energy value. If the elecrostatic interaction energy is lower than this value, the energy is written to the file defined under "structene".

structene Name for the low energy file. If the energy of an orientations is lower than the energy defined in "lowenergy", the orientation data is written to this file. This output is not dependent on the statistical "nprint" parameters. This file can be seen as an orientation file including only low energy docking orientations.

cmdist Defines the center-of-mass distance of the receptor and the ligand in Å at which the orientation is written to the dockdensity dx-file and

the orientation file.

dockdensity Name of the docking density output file for the whole simulation. The file is saved as OpenDX format and can be inspected with VMD[3] or PyMOL.[4]

4 Files and File formats

4.1 Structure files

The ligand and the receptor structure have to be in a PQRM format, which consists of a PQR format (reference: http://apbs-pdb2pqr.readthedocs.io/en/latest/formats/pqr.html; last visited Dec 2017) and the mass of the atom in the last column. The PQRM files have the following format (in columns):

```
FieldName AtomNo AtomName ResidueName ChainID ResidueNo X Y Z Charge Radius Mass
```

for example

```
ATOM   4630  N   THR   995      34.639  19.002  42.195  -0.300   1.550  14.007
ATOM   4631  H1  THR   995      35.362  19.741  42.302   0.330   1.000   1.008
ATOM   4632  H2  THR   995      34.592  18.300  42.949   0.330   1.000   1.008
ATOM   4633  H3  THR   995      33.703  19.469  42.047   0.330   1.000   1.008
ATOM   4634  CA  THR   995      34.843  18.378  40.856   0.210   1.700  12.011
ATOM   4635  HA  THR   995      35.754  17.804  40.880   0.100   1.000   1.008
ATOM   4636  CB  THR   995      33.655  17.403  40.497   0.140   1.700  12.011
ATOM   4637  HB  THR   995      34.088  16.535  39.946   0.090   1.000   1.008
ATOM   4638  OG1 THR   995      32.741  18.072  39.575  -0.660   1.500  15.999
```

The fields are described in the following:

FieldName String specifying the type of PQR entry. It should either be ATOM or HETATM.

AtomNo Integer providing the atom index.

AtomName String providing the atom name.

ResidueName String providing the residue name.

ChainID An optional string which provides the chain ID of the atom.

ResidueNumber Integer providing the residue index.

X Y Z 3 floats representing the atomic coordinates.

Charge Float representing the atomic charge (in electrons).

Radius Float representing the atomic radius (in Å).

Mass Float representing the atomic mass (in atomic mass unit).

The PQRM files can either be generated from a coordinate (*.crd) and a psf file from CHARMM with the program `psfcrd2pqr`, which is part of the MCMap suite, or it can be generated by adding the mass of each individual atom to the last column of a PQR file. In the example run script, a conversion of the CHARMM output files to a PQRM file can be seen.

4.2 Potential files

MCMap needs 2 electrostatic potential files of the receptor, a coarse and a fine one. The potential has to be in OpenDX format and can be generated with APBS.[5] For further information about the OpenDX format, visit
http://apbs-pdb2pqr.readthedocs.io/en/latest/formats/opendx.html (last checked Dec 2017).

4.3 Exclusion/Inclusion grid

The exclusion or inclusion grid is in binary format and is created by the utility `make-excl`. The exclusion grid prohibits collisions of the ligand with the receptor and consists of a regular grid around the receptor surface. The inclusion grid is bigger than the exclusion grid and is also located within a certain distance to the surface of the receptor. Ensemble orientations touching the inclusion grid are considered for further analysis.

4.4 Orientation file

In the orientation file, the output of a MCMap run is saved. The file contains the energy and the coordinate transformation matrix of every saved MC step. This allows to generate each docked complex from the initial structures. Every entry consists of 17 columns:

```
Descriptor   Energy   X-Tr   Y-Tr   Z-Tr   X-Cent   Y-Cent   Z-Cent   3x3-TransMatrix
```

The fields are described in the following:

Descriptor A description string.

Energy	The electrostatic interaction energy of the ligand and the receptor for the individual MC step. The energy of the docking step is saved in kT.
X-Tr Y-Tr Z-Tr	3 translation coordinates necessary for shifting the complex.
X-Cent Y-Cent Z-Cent	3 coordinates representing the center for coordinate transformation.
3x3-trans-matrix	Coordinate transformation matrix consisting of 9 numbers.

4.5 Docking density file

The docking density file represents the distribution of accepted ligand orientations in OpenDX format. If a docking event occurs at a grid point, the number on the grid point is incremented, which results in a distribution of densities for all docking results. The file can be inspected with any molecular viewer.

5 Utilities

MCMap includes separate utility programs, which help preparing and processing the input and output files.

5.1 print-coor

The program `print-coor` is used for analyzing the results of a MCMap run. `Print-coor` can generate contact maps and different representations to give an overview of the ensemble. It can generate the structure for a single orientation or multiple ones, a density representation of the whole ensemble or a distribution representation. `Print-coor` can also modify the ensemble in a way to only consider the docking events within a certain distance to the receptor.

```
Usage: print-coor -m [mode] -o [orientation-file.gz] -r [receptor] -l [ligand]
                  -n [nprint] -d [dist] -s [spacing] -inc [inclusion grid]
```

The arguments **mode**, **orientation-file**, **receptor** and **ligand** are mandatory. The orientation-file is obtained by a MCMap run and needs to be gzipped. The receptor and ligand files represent the structure files (PQRM) used to get the orientation file in a MCMap simulation.

By the keyword mode the analysis method can be chosen. The different possibilities are explained in the following:

lig Every orientation of the ligand relative to the provided receptor structure is printed to the directory Struct in PQRM-format. If nprint is defined, every nth orientation is generated. Before usage the directory "Struct" needs to be created.

rec Every orientation of the receptor relative to the provided ligand structure is printed to the directory Struct in PQRM-format. If nprint is defined, every nth orientation is generated. Before usage the directory "Struct" needs to be created.

cm-lig The coordinates of ligand CMs for all orientations are printed to STDOUT in PDB-format.

cm-rec The coordinates of receptor CMs for all orientations are printed to STDOUT in PDB-format.

dx-lig The docking density of the ligand is given to STDOUT in OpenDX-format. The distance and grid spacing parameters (in Å) are used to define the size of the OpenDX cube and are mandatory.

dx-rec The docking density of the receptor is given to STDOUT in OpenDX-format. The distance and grid spacing parameters (in Å) are used to define the size of the OpenDX cube and are mandatory.

min-dist All orientations within a defined distance **d** to the CM of the receptor are considered for output. An OpenDX-file with the minimal approach density is written to STDOUT. The distance and grid spacing parameters are mandatory (in Å).

contact The mode writes a contact map histogram within a defined distance **d**. The distance variable **d** is mandatory.

aa-contact The mode writes a contact map histogram of amino acid residues for contacts within a distance **d**. The distance is given by the variable **d**.

ens-surf The mode creates a subensemble with the help of an inclusion grid. All ligand orientations touching the inclusion grid are considered for the new

ensemble. The distribution of structures is printed to STDOUT as a density in OpenDX format. The subensemble orientation file is saved under "mod_structures.tar.gz". An example of the inclusion method can be seen in Figure 1.

5.2 psfcrd2pqr

The program `psfcrd2pqr` uses the output files of CHARMM (type *.CRD and *.PSF) to create structure files. The structure files can be chosen to be either in PQR, PQRC, PQRM or PQRX format.

```
Usage: psfcrd2pqr  [psf] [crd] [output format]
Possible formats:
     [PQR]  -- plain PQR
     [PQRC] -- PQR with chainid
     [PQRM] -- PQR with atom mass
     [PQRX]
```

5.3 pqrm-info

`Pqrm-info` gives you information about the structure you use. The structure file has to be in PQRM format. The program provides the total charge, the coordinates of the center of mass and minimal and maximal coordinates of the structure.

```
Usage: pqrm-info [structure-file]
```

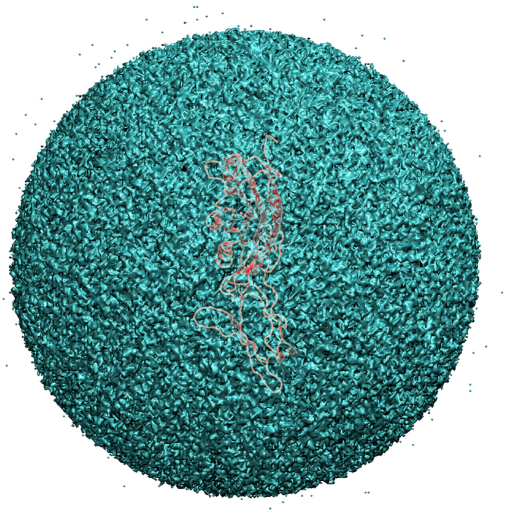
5.4 shift-center

The program shifts the center-of-mass of the structure to the origin of the coordinate system. The structure file has to be in PQRM format.

```
Usage: shift-center [structure-file]
```

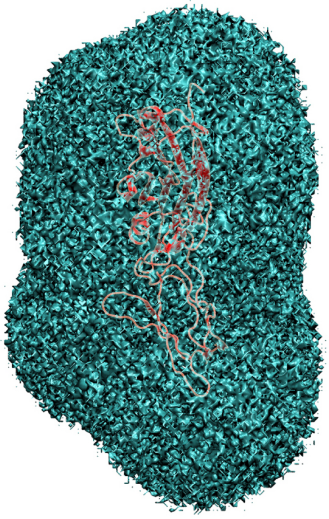
5.5 make-excl

The program creates an exclusion grid with a distance r around the structure. A probe with a radius r (in Å) is used to obtain the surface of the structure. The grid spacing can be



(a) All orientations included

(b) Inclusion grid



(c) Inclusion selected orientations

Figure 1: Visualization of the surface inclusion function in `print-coor`. Part 1.a: The whole encounter ensemble is shown in cyan while the receptor backbone is shown in red. Part 1.b: The inclusion grid surrounding the receptor is shown in green. The distance to the surface of the receptor is 3 Å. Part 1.c: The changed encounter ensemble is colored cyan. Compared to the whole encounter ensemble, only encounters penetrating the inclusion grid (1.b) are selected.

defined by the parameter 'spacing' (in Å). The structure file has to be in PQRM format.

```
Usage: make-excl [structure-file] [probe radius] [spacing]
```

5.6 grid2pdb

The program can convert the binary exclusion/inclusion grids to PDB files. A dummy atom is placed on every grid point. The PDB file is printed to STDOUT. This helps to determine the dimension of the inclusion or the exclusion grid. Warning: With a very fine grid spacing, the PDB files can get very big.

```
Usage: grid2pdb [grid-file]
```

5.7 get-potential

The program `get-potential` can calculate the potential of the receptor at the position of the ligand. The program needs a potential file in OpenDX format and a structure with point charges which is within the borders of the potential.

```
Usage: get-potential [potential-file] [structure-file]
```

5.8 calc-energy

The program `calc-energy` is similar to get-potential, but calculates the electrostatic interaction energy of a ligand within the potential of a receptor. The program needs a potential file in OpenDX format and a structure with point charges which is within the borders of the potential. Optionally, an exclusion grid can be provided to the program, which is then used to test whether the ligand structure collides with the receptor before calculating the energy.

```
Usage: calc-energy [potential-file] [structure-file] [exclusion-grid]
```

5.9 histogram

The program `histogram` counts the number of data points within a certain bin. You can choose the size of the bin with the parameter 'step'. As an output `histogram` gives a range

of bins with the data points summed up. An example is provided after the usage.

```
usage: histogram [file] [step]
```

If you have an energy distribution with 26524 orientation:

```
-1.383991E+01
-1.560415E+01
-1.579327E+01
-1.500038E+01
-1.549712E+01
-4.089399E+00
-2.580790E+00
-1.550029E+00
-9.840530E-01
-9.177106E-01
      .
      .
      .
```

`histogram` clusters and counts the energies depending on the bin size. With a bin size of 4 kT the energy distribution results in the following:

```
# 26524 lines
# Min: -1.884058e+01  Max: 2.152845e+00
# Histogram Vector: -5 ... 1
-2.000000e+01 54
-1.600000e+01 7440
-1.200000e+01 7260
-8.000000e+00 3264
-4.000000e+00 4827
0.000000e+00 3678
4.000000e+00 1
```

5.10 histogram2d

The program `histogram2d` is similar to `histogram` with the difference of creating input for a two dimensional plot. As an input `histogram2d` needs a file with two correlating datasets.

For each dimension you have to define the step size (step1, step2). You can choose between three different normalization methods for the histogram plot. As an output histogram2d plots x- and y- coordinates with the counts summed up for each of the individual coordinates.

```
usage: histogram2d [file] [step1] [step2] [norm-flag]
    norm-flag = 0 -- no normalization
    norm-flag = 1 -- on the highest value
    norm-flag = 2 -- on total number (probability)
```

6 Analysis of contact count histograms with PyCoALA

6.1 Introduction

PyCoALA, the **Py**thon **Co**ntact **A**rea **L**ocalization and **A**nalysis tool, is a Python program to identify and visualize contact areas between a receptor and a ligand from an ensemble of encounter complexes processed either with `print-coor contact` or `print-coor aa-contact` mode (cp. also Section 5.1). The contact count data will be visualized in color-coded histogram plots. In addition, for an overview of the location and composition of contact hot spots, residues of receptor and ligand which are most frequently involved in contacts with the partner can be mapped back onto a protein surface for visualization by a PyMOL[4] script. The program further supports the subtraction of two related simulation data sets to visualize the main areas and residues where the biggest differences arise both by a contact histogram plot and, with the help of PyMOL,[4] on the protein surfaces. This tool provides a quick way of connecting contact data heatmaps with the visualization of the main contributors on a protein surface.

6.2 Required Python modules and external programs

The script was mainly tested using Python 3.5 under Kubuntu 16.04. Usage of Python 3 is recommended for substantial speed-up of the analysis.

Required additional modules not being part of the Python Standard Library are NumPy and matplotlib.[6,7]

As an external program for visualizing the contact hot spots on the protein structures, PyMOL Molecular Viewer[4] is required. Running the scripts was tested with PyMOL versions 1.7-1.9.

6.3 Usage: Config file and options

The program can be run with the following command:

```
python[3.5] path-to/PyCoALA.py [path-to/config file]
```

The config file defines all settings for running PyCoALA. Comments in the config file can be made by adding a # symbol in front of the comment. In the config file the following variables can be defined (detailed descriptions of all variables follows below):

Input = Input file(s) of the contact count histogram data obtained with `print-coor contact` (atomic level) or `print-coor aa-contact` mode (amino acid level). **Mandatory!**

Orientations = Number of total encounter complexes of a simulation for normalizing the contact count data. **Mandatory** when performing a **subtraction** of results from two runs.

Plot = Name of the plotted histogram output file. Plot will be written as PNG.

Title = Title for the plot.

xlabel = Label for the x-axis (receptor atoms/amino acids).

ylabel = Label for the y-axis (ligand atoms/amino acids).

Logscale = Optional switch for plotting count data on a logarithmic scale; only if using single, non-normalized input.

Receptor = PQRM file of the receptor.

Ligand = PQRM file of the ligand.

Display = Percentage of residues with respect to the total number of possible contacts which yielded top counts (or top and lowest counts for subtraction) and will be visualized with PyMOL scripts.

MaxContacts = Output file name for a text report of the top contacts to be visualized on the protein surface and their corresponding contact counts.

ReceptorPml = Output file name for the receptor PyMOL script.

LigandPml = Output file name for the ligand PyMOL script.

PyMOL scripts can be run on UNIX systems by issuing

```
pymol script.pml
```

A more detailed description of the available parameters and default values can be found below.

6.3.1 Input

Input files for PyCoALA are files obtained from `print-coor contact` with counts reported at atomic level or `print-coor aa-contact` mode with counts data reported at amino acid detail (see also Section 5.1 for further details). Specifying input is **mandatory**, the program always requires at least one input file (but no more than two). If a single input file is specified, the program runs in standard mode. If two input files are specified, separated by comma, these will be used for subtraction (file 1 - file 2 results). If hydrogen atoms are reported in data at atomic detail, these will be ignored for the analysis.

The input file is expected to be in the following format:

```
Receptor                                  Ligand
Number   Chain:AANo:AminoAcid:Atom    Number   Chain:AANo:AminoAcid:Atom    ContactCount
```

for example

```
1  -:1:THR:N            1 -:995:THR:N          1163
1  -:1:THR:N            2 -:996:GLU:N          1817
1  -:1:THR:N            3 -:997:PHE:N          688
```

The fields are described in the following:

Number Consecutive integer number to label each atom of receptor/ligand; starting from 1.

Chain An optional string which provides the chain ID. If not specified, '-' is used instead.

AANo Integer for amino acid index.

AminoAcid String for Amino acid or hetero group as three letter code.

Atom String for atom. Only N is displayed if data is at amino acid resolution.

ContactCount Integer for counting how often the given pair of atoms/amino acids from receptor and ligand was within the specified contact distance in all collected encounter complexes.

6.3.2 Orientations

The number of total encounter complexes obtained in the simulation can be used for normalizing the contact count data. Providing this option is **mandatory** when using subtraction mode. In this case, the total number of orientations needs to be specified for each of the two input files (two orientations, given in the same order as the corresponding input).

If a single input file will be visualized, providing orientations is optional. If the number of orientations is provided, normalization to the total number of orientations will be performed.

Orientations can be provided in two ways: Either, the output of MCMap, a gzipped orientation file (structdist.gz; cf. also Section 4.4), can be provided and the total number of orientations will be determined by PyCoALA, or the number of orientations can also be given as an integer value.

6.3.3 Parameters related to plotting contact count histograms

User-specified options for the histogram plots include the file name for the output (**Plot**), title and axis-labels, as well as using a logarithmic scaling for the color gradient for non-normalized input (no orientations specified). If normalization is performed, the histograms are scaled by the inverse of the highest absolute value of the histogram to yield a maximum value of 1. The scaling factor is appended to the title. Examples for plots produced with PyCoALA can be found in Figure 2. The color gradient is chosen automatically dependent on the input type (standard input: rainbow color spectrum; subtraction: red-white-blue).

The following parameters can be set:

Plot	Name for the output file of the histogram plotting. The plot will be written as a PNG file. Specifying a name for the output is optional; if the variable is not provided, the default file name '*countHistogramPlot.png*' will be used. Warning: If the specified file name already exists, previous plots will be overwritten without further notification.
Title, xlabel and ylabel	Title and axis labels for the contact histogram plot. Any given title/label string needs to be surrounded by `""`. By default, the receptor amino acids/atoms will be displayed on the x-axis and the ligand amino acids/atoms on the y-axis. If no title/axis labels are given in the config file, the following defaults are used: *This is my plot title*, *This is my x-axis label* and *This is my y-axis label*.

Logscale Optional switch to use a logarithmic scale for the histogram color gradient. For any normalized input (normalized single runs as well as subtraction runs) this setting is suppressed. It can be used as an option when plotting non-normalized total counts for single input. Possible options are **True** or **False**. By default it is set to *False*. A comparison of example data with total counts plotted on normal vs. logarithmic scale can be found from Figure 2, panel a and b.

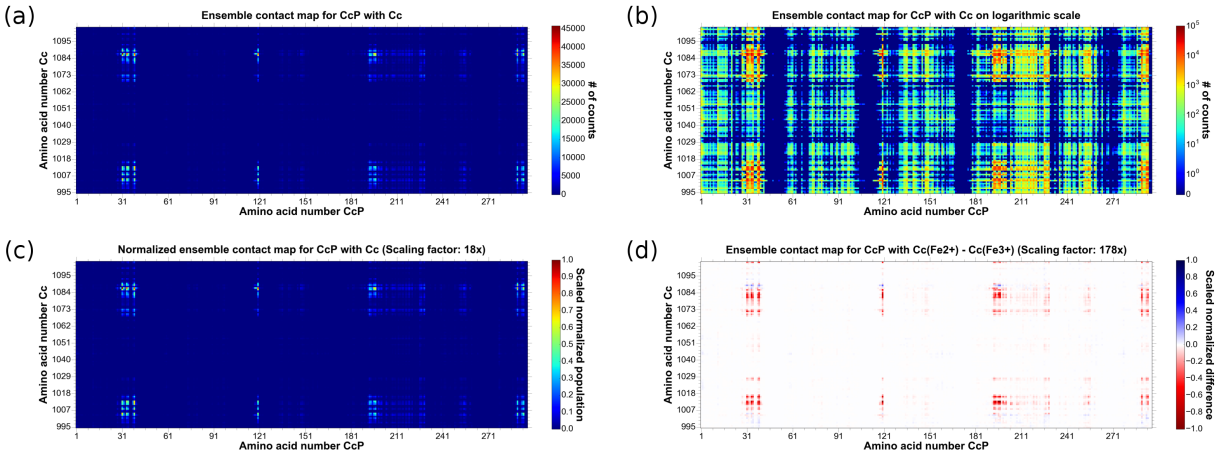


Figure 2: Example contact histogram plots for simulations of yeast cytochrome *c* peroxidase (CcP) with cytochrome *c* (Cc). In all plots, amino acids of CcP are shown on the x- and amino acids of Cc on the y-axis. Counts are represented by color gradients. (a) Total number of counts for CcP and Cc. (b) Identical to (a), but with color gradient on logarithmic scale. (c) Like (a), but values normalized to the total number of orientations. Note the scaling factor given with the plot title. (d) Example subtraction plot. Normalized data from two related, independent simulations with different oxidation states of Cc heme iron is subtracted and scaled to highlight the areas of greatest differences.

6.3.4 Parameters for mapping contacts back onto protein surfaces

The location and nature of contact hot spots or areas of main differences between two related, independent simulations can be visualized with the help of PyMOL[4] scripts written by PyCoALA. User-specified options include the structure files of receptor and ligand (**Receptor, Ligand**), specifying the percentage of residues to be visualized on the protein surface (**Display**) and file names for a text report with contact counts (**MaxContacts**) and the PyMOL scripts for receptor and ligand (**ReceptorPml**, **LigandPml**).
PyMOL[4] scripts can be run on UNIX systems by issuing

```
pymol script.pml
```

The surface representations are prepared using two different color schemes: For input from a single simulation, contact hot spots are visualized with an amino acid type based color scheme as described below:

Acidic: colored in red (D, E).

Basic: colored in blue (H, K, R).

Polar: colored in green (C, G, N, Q, S, T).

Nonpolar: colored in gray (A, I, L, M, P, V).

Aromatic: colored in purple (F, W, Y).

Any residue of unknown type, which is not part of the scheme will be colored in cyan. An example of a corresponding surface representation is shown in Figure 3.

For subtraction results, areas of major differences between two related, independent simulations, are visualized in blue for positive values (higher counts in first simulation) and red for negative values (higher counts in second simulation). These positive and negative difference hot spots are further organized in four bins and bins are colored from dark blue/ dark red for most positive/negative values to lighter shades of blue/red for less positive/negative ones. Residues, which occur both among positive and negative difference hot spots are highlighted in purple (this is possible, since contacts are recorded as pairs of amino acids/atoms of receptor and ligand and the same residue of one partner may be in contact with different residues of the other protein in the two simulations). An example of a representation of difference hot spots on a protein surface together with a graphical representation of the color scheme can be found in Figure 4.

The following parameters can be set in the PyCoALA config file:

Receptor, Ligand Receptor and ligand PQRM files (for more information on PQRM files, see also Section 4.1). If no files were specified, the default placeholders *dummyreceptor* and *dummyligand* will be used. In this case, a warning is displayed and the written PyMOL scripts likely need to be edited by hand.

Display Floating point number to define the percentage of residues to be displayed in the PyMOL script with reference to the total number of all possible contact counts (m receptor atoms/amino acids × n ligand atoms/amino acids; considered 100%). Note that very small numbers are typically sufficient. Default percentage is *0.05*. In standard mode, residues with highest counts are selected. For subtraction mode, the selected percentage will be shown both for residues with positive as well as negative counts after subtraction of the data sets.

MaxContacts Output file for a text report of the top contacts to be visualized by the PyMOL script to provide a reference with the corresponding count values. By default, an ASCII text file *topContactPairs.txt* will be written. Warning: If the specified filename already exists, the text report will be overwritten without further notification.

ReceptorPml, LigandPml Output files for receptor and ligand PyMOL scripts. Default file extension is .pml; By default *dummyreceptor.pml* and *dummyligand.pml* will be written. Warning: If the specified filenames already exist, the scripts will be overwritten without further notification.

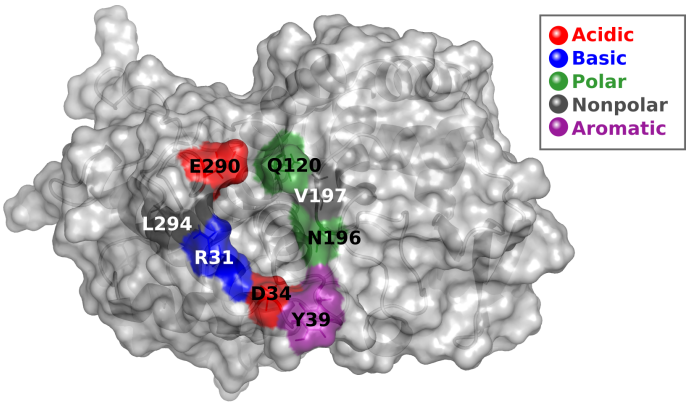


Figure 3: Hot spot residues of yeast cytochrome *c* peroxidase for contacts with cytochrome *c* visualized with a PyMOL script written by PyCoALA. The top 0.1% of all available contacts are displayed and colored according to amino acid types as indicated in the color scheme inset on the right.

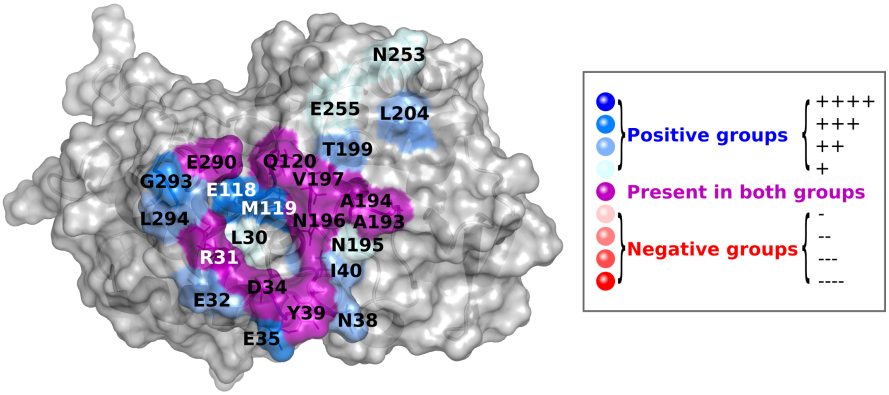


Figure 4: Difference hot spots of yeast cytochrome *c* peroxidase for contacts with cytochrome *c* (Cc) with oxidized heme iron subtracted from Cc with reduced heme iron. The PyMOL script written by PyCoALA was used to display both the top and the lowest 0.1% of all available contacts. The color scheme for subtraction data can be seen in the inset on the right.

7 Example of a MCMap run with yeast Cytochrome *c* and Cytochrome *c* Peroxidase

The example for MCMap is split in two parts. The first part includes the structure preparation and the docking simulation for the model system cytochrome *c* and cytochrome *c* peroxidase of yeast. In the second part, the simulation results are processed and prepared for visualization.

In part one, the CHARMM output files (*.crd, *.psf) from PDB 2PCC[8] are converted to the PQRM format. Then, the electrostatic potential of the receptor is calculated and finally, the MCMap simulation is done. Before running the example, define the paths to the necessary programs (APBS,[5] MCMap).

The first part of the example can be started with the command:

```
csh master.csh
```

For a detailed description, have a look at the file "master.csh". Every step is documented. The docking simulations takes about 20 minutes on a single core Intel® Core™ i7-4790 with a speed of 3.6 GHz.

In order to start the analysis with the subprogram `print-coor` to create different ensemble representation run the following command:

```
csh master.csh 5
```

Here, all visualization methods of `print-coor` are tested and the contact map of the ligand and the receptor for all orientations is calculated. The contact map is then processed and plotted with the tool `PyCoALA`, which also creates a PyMOL script for mapping the contacts to the structure surfaces.

References

(1) Foerster, J. M.; Poehner, I.; Ullmann, G. M. MCMap – A Computational Tool for Mapping Energy Landscapes of Transient Protein-Protein Interactions. **2018**, submitted.

(2) Metropolis, N.; Ulam, S. The Monte Carlo method. *Journal of the American Statistical Association* **1949**, *44*, 335–341.

(3) Humphrey, W.; Dalke, A.; Schulten, K. VMD: visual molecular dynamics. *J. Mol. Graphics* **1996**, *14*, 33–8, 27–8.

(4) Schrödinger, LLC,

(5) Baker, N. A.; Sept, D.; Joseph, S.; Holst, M. J.; McCammon, J. A. Electrostatics of nanosystems: Application to microtubules and the ribosome. *P. Natl. Acad. Sci. USA* **2001**, *98*, 10037–10041.

(6) van der Walt, S.; Colbert, S. C.; Varoquaux, G. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing In Science & Engineering* **2011**, *13*, 22–30.

(7) Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering* **2007**, *9*, 90–95.

(8) Pelletier, H.; Kraut, J. Crystal structure of a complex between electron transfer partners, cytochrome c peroxidase and cytochrome c. *Science* **1992**, *258*, 1748–1755.