EASYMIFS & SITEHOUND

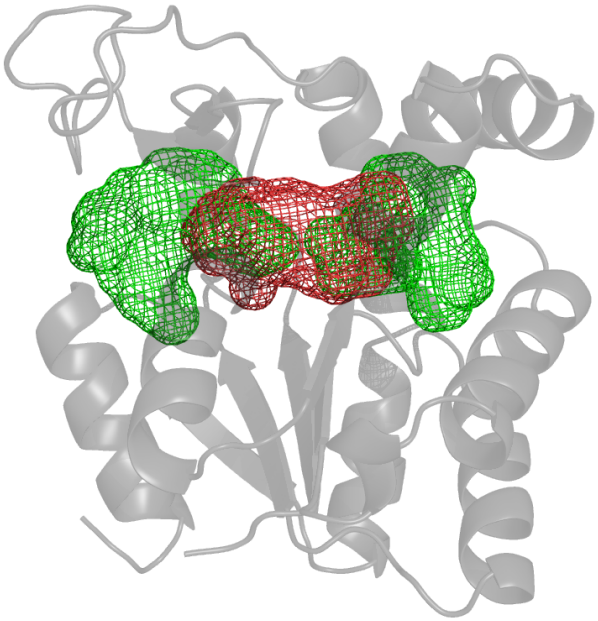
Programs for ligand-binding site identification and characterization in protein structures

User's Guide

http://sitehound.sanchezlab.org

Dario Ghersi [1]

Version 0.1, January 14, 2010



[1](dario [at] sanchezlab.org)

Disclaimer & Acknowledgements

These programs are distributed in the hope that they will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for any purpose. The entire risk as to the quality and performance of the program is with the user.

The EasyMIFs and SiteHound programs have been written by Dario Ghersi. The SiteHound-web server was developed by Marylens Hernandez. Both in the group of Roberto Sanchez in the Department of Structural and Chemical Biology, Mount Sinai School of Medicine.

These programs have been developed in the context of research work supported by grants from the National Science Foundation (NSF) and by the National Institutes of Health (NIH) to Roberto Sanchez. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF or NIH.

Distribution

Distribution of the programs is allowed only with the author's written consent.

Contact: `sitehound@sanchezlab.org`

Contents

1 Introduction **7**

1.1 Binding Site Identification . . . 7

1.2 What are EasyMIFs and SiteHound? . . . 7

1.3 EasyMIFs: a Molecular Interaction Field calculator . . . 7

1.4 SiteHound: a binding site identification tool . . . 8

2 Quick Guide: simple ligand binding site identification **11**

2.1 Using the `auto.py` script . . . 11

2.1.1 `auto.py` example: adenylate kinase (PDB code 1aky) . . . 11

2.2 `auto.py` options . . . 14

3 EasyMIFs **15**

3.1 Running EasyMIFs . . . 15

3.1.1 Example . . . 16

3.2 Output Files . . . 16

3.2.1 .dx . . . 16

3.2.2 .cmp . . . 17

3.3 Visualizing the results . . . 17

4 SiteHound **19**

4.1 Running SiteHound . . . 19

4.1.1 Example . . . 20

4.2 SiteHound Output Files . . . 20

4.2.1 .tmp file . . . 20

4.2.2 _summary.dat file . . . 21

4.2.3 _clusters.dat . . . 21

4.2.4 _predicted.dat . . . 21

4.2.5 _clusters.dx . . . 22

4.2.6 _clusters.pdb . . . 22

4.3 Visualization . . . 22

5 Methods **25**

5.1 Calculation of MIFs in EasyMIFs . . . 25

5.2 Brief overview of clustering in SiteHound . . . 26

A List of EASYMIFs probes 31

B The SITEHOUND-web Server 33

C How to calculate interaction energy maps with Autogrid 35
C.1 Installing `Autogrid` and `Autodock tools` 35
C.2 Preparing the PDBQT file 35
C.3 Preparing the gpf file 35
C.4 Running `Autogrid` 36

D Software Versions 37

Chapter 1

Introduction

1.1 Binding Site Identification

The molecular function of proteins is largely determined by their interaction with other molecules at binding sites on the protein surface. Thus, localization and characterization of a ligand-binding site can contribute to functional annotation of a protein; it can guide mutational experiments, and be useful in predicting or verifying interactions. The identification of ligand binding sites can also be an important part of the drug discovery process. Knowing the location of binding sites facilitates virtual screening for hits, lead optimization, and identification of features that influence the selectivity of binding.

1.2 What are EASYMIFs and SITEHOUND?

EASYMIFs and SITEHOUND are two software tools that in combination enable the identification of binding sites in protein structures using an energy-based approach. EASYMIFs, is a simple Molecular Interaction Field (MIF) calculator; and SITEHOUND, a post processing tool for MIFs that identifies interaction energy clusters corresponding to putative binding sites. While these tools are most commonly used in combination, they can also be used separately. EASYMIFs can be used to calculate MIFs for binding site characterization, Quantitative Structure-Activity Relationship (QSAR) studies, selectivity analysis of protein families, pharmacophoric search, and other applications that require MIFs [1]. SITEHOUND can be used to process the ouput from other MIF or Affinity Map calculation programs, in addition to EASYMIFs, such as `GRID` [2] and the `Autogrid` tool of the `AutoDock` software package [3].

1.3 EASYMIFs: a Molecular Interaction Field calculator

Molecular Interaction Fields (MIFs) describe the spatial variation of the interaction energy between a target molecule and a specific probe, that usually represents a chemical group. Although the interaction energy field is, by definition, a continuous quantity, for computational convenience it is usually discretized on a three-dimensional orthogonal grid that surrounds the molecule of interest. The output of a MIF calculation is therefore represented by an energy map that provides information about the potential energy between the probe and the molecule under analysis. EASYMIFs aims to provide a simple and rapid way to characterize a protein structure from a chemical standpoint at the global or local level (e.g.

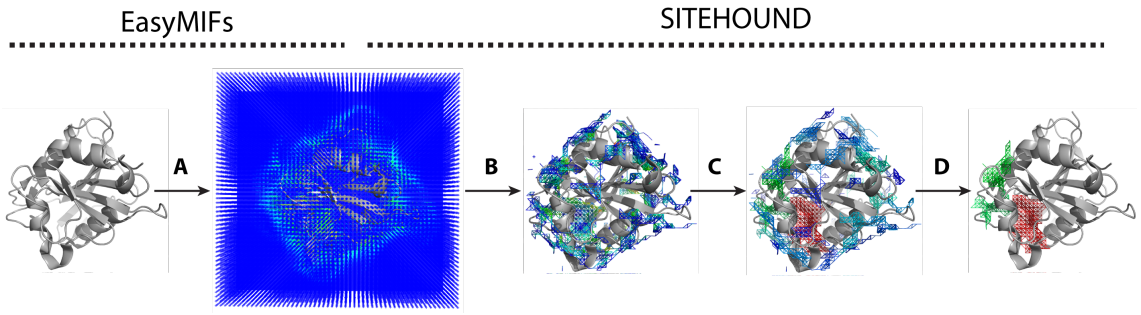


Figure 1.1: **Identification of ligand-binding sites using EASYMIFs and SITEHOUND.** (A) A protein structure is used as input and program EASYMIFs computes the potential interaction energy of a molecular probe with the protein on each point on an orthogonal grid called a Molecular Interaction Field (MIF). (B) Program SITEHOUND processes the MIF by first removing all points that have unfavorable interaction energy, (C) subsequently the remaining points are grouped using a hierarchical clustering algorithm, and the resulting clusters are ranked by their Total Interaction Energy (the sum of the interaction energy of all points in one cluster). (D) Known binding sites are usually found among the top three clusters.

around an active site), returning maps that can be loaded in a Molecular Graphics Software such as PyMol, VMD or Chimera. The calculations are carried out *in vacuo* using the GROMOS [4] force field and a distance dependent dielectric, as described in detail in section 5.1.

1.4 SITEHOUND: a binding site identification tool

The purpose of SITEHOUND is to manipulate the output of the EASYMIFs program (and other programs such as `Autogrid` [3] and `GRID` [2]) in order to predict regions on protein structures that are likely to be involved in binding to small molecules or peptides. The approach is based on the `Q-SiteFinder` algorithm [5], but contains more options and improvements. The program first filters off all the grid points that have an energy value above a user-specified threshold (a negative value) and clusters them according to spatial proximity using **single** or **average** linkage agglomerative clustering (see Section 5.2). Subsequently, the **Total Interaction Energy** (TIE) of each cluster is computed and this value is used to rank the clusters, from the most negative to the least negative. The last step involves printing the results on text files and in the PDB and DX formats, that allow for graphical display of the results on the protein using standard molecular visualization tools (such as `Chimera`, `PyMol` or `VMD`). A convenient Web Interface is available at `http://sitehound.sanchezlab.org` that allows the user to input a PDB file and obtain the results automatically [6] (see Appendix B).

EASYMIFs and SITEHOUND references

[GS09a] D. Ghersi and R. Sanchez. Easymifs and sitehound: a toolkit for the identification of ligand-binding sites in protein structures. *Bioinformatics*, 25(23):3185–6, 2009.

[GS09b] D. Ghersi and R. Sanchez. Improving accuracy and efficiency of blind protein-ligand docking by focusing on predicted binding sites. *Proteins*, 74:417–424, 2009.

[HGS09] M. Hernandez, D. Ghersi, and R. Sanchez. SITEHOUND-web: a server for ligand binding site identification in protein structures. *Nucleic Acids Research*, 37:W413–416, 2009.

Chapter 2

Quick Guide: simple ligand binding site identification

2.1 Using the `auto.py` script

The `auto.py` script is a wrapper around EasyMIFs and SiteHound that automates all the steps required for binding site identification from a PDB file: protein preparation, MIF calculation via EasyMIFs, and binding site identification with SiteHound. The default values are tuned for the methyl probe (CMET; a.k.a. "carbon probe") and the phosphate oxygen probe (OP; a.k.a. "phosphate probe"). The auto.py script can be executed with the following command line:

```
auto.py -i PDB -p PROBE_TYPE
```

where `PDB` corresponds to the input PDB file, and `PROBE_TYPE` to the selected probe (currently CMET and OP have been tested). Usually a `-k` option is added to remove existing hetero atoms (see 2.2). If the calculations are successful, the number of output files will be produced. The main output files are described in the example below, a detailed description of all output files can be found in chapters 3 (EasyMIFs) and 4 (SiteHound).

2.1.1 `auto.py` example: adenylate kinase (PDB code 1aky)

Binding site identification with the "carbon" (CMET) probe on the adenylate kinase structure can be performed with the following command:

```
auto.py -i 1aky.pdb -p CMET -k
```

binding site identification with the "phosphate" (OP) probe only requires a change in the `-p` option:

```
auto.py -i 1aky.pdb -p OP -k
```

`auto.py` output

The output files are tagged with the corresponding probe name (e.g. `1aky_CMET_summary.dat` and `1aky_OP_summary.dat`). Three of the output files are most frequently used: `_summary.dat`, `_predicted.dat`, and `_clusters.pdb`. For descriptions of the remaining output files see sections 3.2 and 4.2.

The `_summary.dat` file contains a list of all identified clusters (predicted binding sites) ranked by Total Interaction Energy (TIE). Real binding sites usually rank among the top three clusters [7] and have TIE values that stand out from the background.

`1aky_CMET_summary.dat`

```
1       -1296.776    110      6.734     18.303     20.679
2       -1123.672     95     12.256     34.283     16.695
3        -944.721     84     13.508     25.486     16.663
4        -597.539     55     23.185     26.306     18.435
5        -498.763     46     14.329     13.118     19.847

[...]

53        -18.213      2     -2.536     39.527     14.917
54         -9.326      1     25.974     44.527     21.974
55         -9.105      1      3.974     11.527     26.974
56         -8.922      1     20.974     10.527     34.974
```

The columns contain the following information: 1: Cluster rank; 2: TIE; 3: Cluster Volume (in $Å^3$); 4-6: X, Y, and Z coordinates of the cluster center. The coordinates can be used to center a docking box around a predicted binding site (see [7] for an example of such an application).

The `_predicted.dat` file lists the residues in the neighborhood of the predicted binding sites. Each line in the file corresponds to a list of residues that are within 4.0 Å of the cluster in the input PDB file. The cluster is indicated in the first column. Only data for the first 10 clusters (ranked by TIE) is included in this file.

`1aky_CMET_predicted.dat`

```
1 GLY_14_  ALA_15_  GLY_16_  THR_19_  GLN_20_  LEU_124_  ARG_128_  ILE_129_  ARG_132
_  SER_141_  TYR_142_  HIS_143_  PHE_146_  ASN_147_  ALA_202_  SER_203_  GLN_204_  P
RO_205_  PRO_206_  VAL_209_  AP5_301_  HOH_509_  HOH_522_  HOH_528_  HOH_530_  HOH_5
38_  HOH_540_  HOH_602_  HOH_624_
2 PRO_13_  THR_35_  GLY_36_  LEU_39_  ARG_40_  MET_57_  GLY_60_  GLY_61_  LEU_62_  V
AL_63_  MET_68_  GLY_90_  ARG_93_  GLN_97_  ARG_165_  ASP_167_  ARG_176_  ALA_179_
TYR_180_  THR_184_  AP5_301_  IMD_302_  HOH_501_  HOH_502_  HOH_505_  HOH_507_  HOH_
511_  HOH_512_  HOH_513_  HOH_527_  HOH_575_

[...]

9 LEU_76_  THR_77_  ASN_78_  ASN_79_  PRO_80_  CYS_82_  LYS_83_  GLN_108_  HOH_597_
10 LYS_73_  LEU_76_  THR_77_  MET_104_  GLU_107_  GLN_108_  HOH_591_
```

The `_clusters.pdb` file contains the grid points that contribute to each of the clusters. The format is that of a PDB file, and can be used to display the clusters in molecular graphics programs such as PyMOL (see Figure 2.1). Each cluster is represented as one residue in the PDB file. The residue names have the format CXX, where XX is the cluster index (e.g cluster 1 has residue name C01). By default all clusters are HETATM entries. Note that for some applications it may be necessary to convert them to ATOM entries. This is the case, for example, to represent the clusters as surfaces in PyMOL (see Figure 2.1).

1aky_CMET_clusters.pdb

```
HETATM 2522  C3  C01 A   1     5.974   17.527  19.974  11.62 100.00
HETATM 2523  C3  C01 A   1     5.974   17.527  21.974  11.29 100.00
HETATM 2524  C3  C01 A   1     6.974   17.527  18.974  10.77 100.00
HETATM 2525  C3  C01 A   1     6.974   17.527  19.974  10.41 100.00
HETATM 2526  C3  C01 A   1     10.974  19.527  21.974  10.92 100.00
HETATM 2527  C3  C01 A   1     10.974  18.527  21.974  14.55 100.00

[...]

HETATM 3534  C3  C53 A  53    -2.026  39.527  17.974  8.93  1.40
HETATM 3535  C3  C53 A  53    -3.026  39.527  11.974  9.28  1.40
HETATM 3536  C3  C54 A  54    25.974  44.527  21.974  9.33  0.72
HETATM 3537  C3  C55 A  55    3.974   11.527  26.974  9.11  0.70
HETATM 3538  C3  C56 A  56    20.974  10.527  34.974  8.92  0.69
```

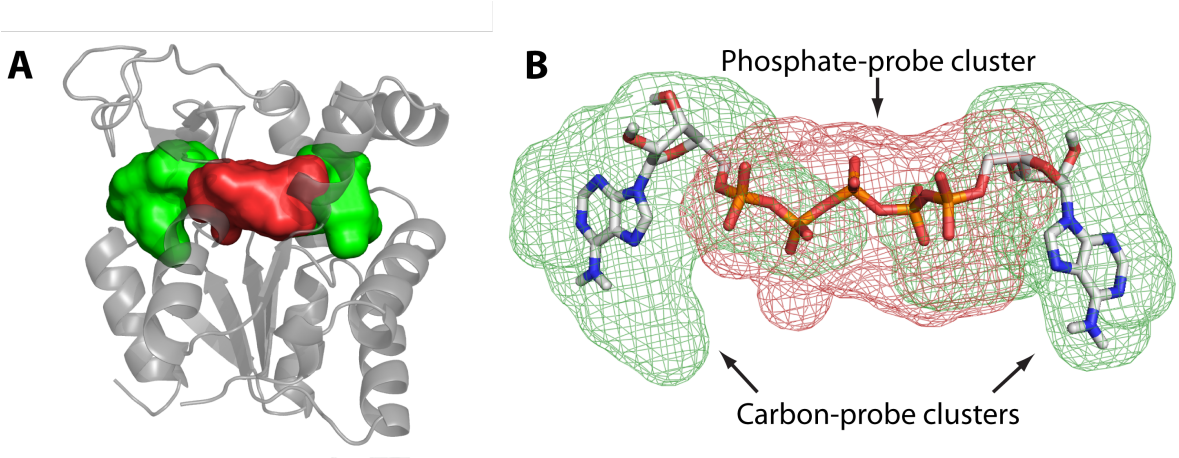


Figure 2.1: **Characterization of the yeast adenylate kinase binding site using EASYMIFs and SITEHOUND** (Figure from reference [6]). **(A)** Ribbon diagram of the yeast adenylate kinase structure showing the top ranking clusters as solid surfaces: phosphate probe cluster (red) and carbon probe clusters (green). **(B)** SITEHOUND clusters superposed on the structure of the Ap5A (bis(adenosine)-5'-pentaphosphate) inhibitor of adenylate kinase [8]. The phosphate probe correctly identifies the pathway of phosphoryl transfer, and the carbon probe correctly identifies the adenosine binding regions. The figure was prepared using the `1aky_CMET_cluster.pdb` and `1aky_OP_cluster.pdb` files from the example, and the PyMOL molecular graphics program.

2.2 `auto.py` options

The `auto.py` script provides all the options necessary to control EasyMIFs and SiteHound. The complete list of options supported by `auto.py` are listed in Table A.1. For more details on the meaning of some of these options see the Chapters 3 and 4.

Table 2.1: `auto.py` command line options.

Options		Description
`-i`	`--input`	The input PDB file
`-p`	`--probe`	The probe type, either CMET (methyl) or OP (phosphate oxygen)
`-k`	`--clean`	Remove HETATM entires from the PDB file
`-r`	`--resolution`	The resolution of the grid in Angstrom (default: 1.0)
`-c`	`--center`	The center of the grid (default: center of the protein)
`-d`	`--dimension`	The dimensions of the box (default: whole protein)
`-e`	`--energy`	(default: -8.9 for CMET, -8.5 for OP)
`-l`	`--linkage`	Either 'single' or 'average' (default: 'average')
`-s`	`--spatial`	The spatial cutoff (default: 1.1 for single linkage, 7.8 for average)
`-x`	`--pdb2gmx`	Print the output of pdb2gmx
`-z`	`--compress`	Use compressed maps
`-o`	`--log`	Send the output messages to a log file
`-h`	`--help`	Show this help

Chapter 3

EASYMIFs

While EASYMIFs and SITEHOUND are commonly used in combination for binding site identification through the `auto.py` script, EASYMIFs can also be run separately to calculate Molecular Interaction Fields (MIFs) for binding site characterization, Quantitative Structure-Activity Relationship (QSAR) studies, selectivity analysis of protein families, pharmacophoric search, and other applications that require MIFs [1]. This chapter describes how to run EASYMIFs directly (i.e. independently of `auto.py`); and contains a more detailed description of the EASYMIFs output files, and how to visualize them. Some details on the EASYMIFs methodology can be found in Section 5.1.

3.1 Running EASYMIFs

EASYMIFs has been tested under Linux, Mac OS X and Windows XP and is currently a command-line only program. After downloading and uncompressing the package, the directory can be moved to any location. It is necessary to copy the files `'atom_types.txt'` and `'ffG43b1nb.params'` (that can be found in the EasyMIFs directory) to the directory from where EASYMIFs is called.

The steps necessary to carry out the calculations are just two:

1. Convert a PDB file into an .easymifs file by calling the python script `prepare_pdb.py` followed by the name of the PDB file.
2. Invoke the EASYMIFs program on the .easymifs file generated in step one.

As with any property calculated from a structure, the results are going to be as good as the quality of the structure. For step 1 to be successful, the protein should not contain missing atoms or residues. Furthermore, it is necessary to strip the protein of all heteroatoms (including water molecules). This can be accomplished by using the '-k' flag in the `'prepare_pdb.py'` python script.

A typical run of EASYMIFs will be as follows:

```
easymifs -f=FILE.easymifs -p=PROBE -c=X,Y,Z -n=NX,NY,NZ -r=SPACING
```

where `FILE` is the name of the structure of interest, `PROBE` is one of the atom types described in the file 'atom_types.txt' and listed in Appendix A, `X,Y,Z` are the coordinates used to center the box, `NX,NY,NZ` are the number of points in the three cartersian axes (must be an odd number) and `SPACING` is the spacing in Angstrom between the points in the grid (recommended values are 1.0 or 0.5 for more detailed calculations).

It is also possible to let the program determine the dimensions of a box large enough to enclose the whole protein, with a clearance of 5Å from the protein in each direction (useful for binding site prediction). In the latter case only the `-f` and `-p` options will be mandatory.

The probes that have been extensively tested are 'CMET' (a methyl-carbon probe) and 'OP' (oxygen of a phosphate group), but many more are available (such as hydroxyl oxygen, peptide nitrogen, metals, etc.). The complete list of probes can be found in the `atom_types.txt` file.

3.1.1 Example

The following example contains a step-by-step description of an interaction energy map calculation performed on the binding site of an D-allose binding protein (PDB code 1rpj). The .easymifs file can be prepared by calling `prepare_pdb.py`, with the -k option to strip the pdb of all the HETATM records:

```
prepare_pdb.py -f 1rpj.pdb -k
```

Afterwards, the actual interaction energy calculation step can be carried out as follows:

```
easyMIFs -f=1rpj.easymifs -p=OW -c=3.91,7.66,11.63 -n=30,30,30 -r=0.5
```

The command above will focus the calculations on the binding site (on a 15 $Å^3$ box) and return an interaction energy map with a resolution of 0.5 Å.

3.2 Output Files

3.2.1 .dx

This file contains the interaction energy map computed by EasyMIFs, in **units of kJ/mol**. The header of the file contains information about the dimensions, the center and the resolution of the box, followed by the actual energy values, arranged in the standard DX format (X slow, Y medium, Z fast):

```
# easymifs output
#
#
#
object 1 class gridpositions counts 31 31 31
origin -3.590 0.160 4.130
delta 0.500 0 0
delta 0 0.500 0
delta 0 0 0.500
object 2 class gridconnections counts 31 31 31
object 3 class array type double rank 0 items 29791 data follows
5.000
5.000
5.000
5.000
...
```

3.2.2 .cmp

If run with the option `-z`, EASYMIFs will produce compressed maps instead of .dx files. The compressed format is particularly useful when doing binding site identification on multiple structures, since the resulting maps will occupy less space (the files will be 4-6 times smaller on average). However, in order to visualize the maps it will be necessary to uncompress them using SITEHOUND, with the `-z` option (see section 4.1).

3.3 Visualizing the results

EASYMIFs produces Interaction Energy Maps in the 'dx' format, that can be conveniently visualized in `PyMOL`, `Chimera`, `VMD` and other molecular graphics packages. The dx file is usually displayed as a contour plot, showing regions of space where the energy value is within a specified range. Figure 3.1 shows the example discussed above. EASYMIFs has been used to calculate an interaction energy map between the protein (in the binding site region) and an hydroxyl probe, shown in gold in the figure. The box around the binding site illustrates the boundaries of the box used in the calculations. To load a .dx file in Chimera, go to `Tools` ⇒ `Volume Data` ⇒ `Volume Viewer`. A window with many options for manipulating .dx file will appear. A particularly convenient tool is a slide control that allows for easy contouring of the interaction energy map. For more information about displaying .dx files in Chimera, please consult the pertaining documentation.

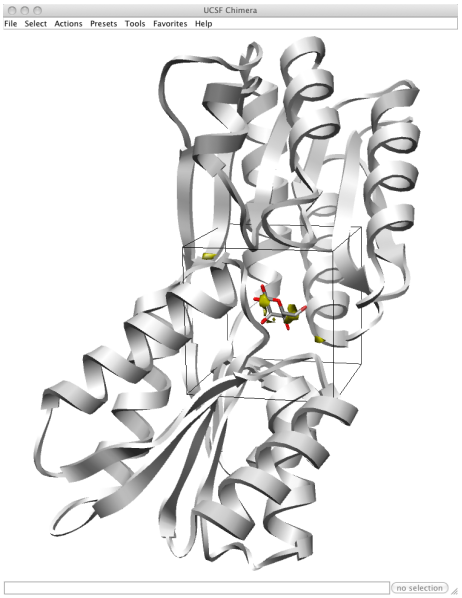


Figure 3.1: **An example of interaction energy calculations on a protein** The protein shown here is a D-allose binding protein (PDB code 1rpj). The box delimitates the area of the protein where the calculations have been carried out. The golden points indicate areas of favorable interaction energy with an hydroxyl probe (energy threshold set to -28 KJ/mol). The ligand is overlaid for comparison, but was removed before computing the interaction energy map.

Chapter 4

SITEHOUND

While EASYMIFS and SITEHOUND are commonly used in combination for binding site identification through the `auto.py` script, SITEHOUND can also be run separately to process the ouput from other MIF or Affinity Map calculation programs, such as `GRID` [2] and the `Autogrid` tool of the `AutoDock` software package [3]. See Appendix C for instructions on how to calculate Affinity Maps with `Autogrid`. This chapter describes how to run SITEHOUND directly (i.e. independently of `auto.py`); and contains a more detailed description of the SITEHOUND output files, and how to visualize them. Some details on the SITEHOUND methodology can be found in Section 5.2.

4.1 Running SITEHOUND

SiteHound is provided as a binary file for a variety of platforms (Windows, Mac-Universal, Linux) and it runs from the command-line. The complete description of the parameters is provided below:

```
sitehound -f=MAP.C.map -t=autogrid -e=-0.3 -l=average -s=7.8
```

Table 4.1: SITEHOUND command line options.

Options	Description
`-f`	The interaction energy map generated by EASYMIFS or `Autogrid`
`-t`	The grid type (either "easymifs" or "autogrid")
`-e`	The energy threshold (all the points whose energy is above that threshold will be removed)
`-l`	The linkage for the clustering algorithm (either "average" or "single")
`-s`	The spatial cutoff, i.e. the level at which the hierarchical tree obtained during the clustering step will be cut
`-p`	The number of clusters for which contributing residues will be reported
`-z`	Decompress a zipped map produced by EASYMIFS and quit

A typical combination (derived from repeated runs on a large set of different protein-ligand complexes) is:

-e=-0.3 -l=average -s=7.8 for **small molecules**
-e=-0.4 -l=single -s=1.1 for **peptides or elongated small molecules**
With maps computed with EASYMIFS (CMET probe) a typical cutoff value for the energy (-e option) is -8.9, whereas for the OP probe (phosphate) is -8.5. It is important to mention that the PDB file used to produce the map should be present in the same directory as the map file, since it will be used to determine which residues are in contact with the clusters.
SITEHOUND can also be used to decompress a zipped map produced by EASYMIFS, with the following command:

```
sitehound -f=map.cmp -z
```

that will produce a "map.dx" file.

4.1.1 Example

The following example contains a step-by-step description of a SITEHOUND run on a dihydrofolate reductase (PDB code 1s3v). The interaction energy map has been computed with EASYMIFS. Please refer to section 3.1.1 for an example. The following command carries out the actual cluster analysis on the interaction energy map:

```
sitehound -f=1s3v_CMET.dx -t=easymifs -l=average -e=-8.9 -s=7.8
```

and yields a set of files whose content is described below.

4.2 SITEHOUND Output Files

`SiteHound` generates different files that can be visually inspected or loaded into statistical packages (such as `R`) for further analysis.

4.2.1 .tmp file

This file is used to store the points that have passed the energy filter and is arranged in the following way:

```
27531 -17.486 36.610 -5.835 -9.366
30277 -16.486 28.610 -4.835 -8.985
30531 -16.486 32.610 5.165 -9.264
```

The first column contains a unique identifier for the point, the following three columns specify the cartesian coordinates of the point and the last column contains the interaction energy value at that particular point.

4.2.2 _summary.dat file

This file contains a summary for all the clusters computed by the program and is organized like this:

```
1       -1476.305       122       -3.536          30.924          2.910
2       -693.843        59         11.032         43.087         15.713
3       -576.476        49        -11.296         32.385          5.965
```

where the first column indicates the cluster index, the second column contains the TIE of that particular cluster, the third point specifies the total number of points that belong to the cluster and the last three columns contain the location of the Center of Energy of the cluster (which is the average of the coordinates of the points that belong to the clusters weighted by interaction energy)

4.2.3 _clusters.dat

This file contains a detailed description of the points contained in all the clusters. An example is reported below:

```
1       -1476.305       -12.709          -7.486          31.610          4.165        59566
1       -1476.305       -17.452           0.514          32.610          0.165        85487
1       -1476.305       -16.186           0.514          31.610          0.165        85426
...
3       -576.476        -12.271         -10.486          32.610          5.165        49929
3       -576.476        -12.048         -10.486          33.610          5.165        49990
3       -576.476        -11.832         -10.486          33.610          6.165        49991
...
```

where the first column refers to the cluster index the point belongs to, the second columns reports the TIE of the cluster, the third column contains the energy of the point, the following three columns contain the cartesian coordinates of the point and the final column reports the unique index associated to the point.

4.2.4 _predicted.dat

This file lists the residues that are in contact with the clusters and that, therefore, have the potential to be involved in binding. A residue is arbitrarily defined to be in contact with a cluster if it has at least one atom within 4.0 Angstrom of a point of the cluster. Below is a typical example:

```
1 TQD_187_  HOH_190_  HOH_193_  HOH_194_  HOH_195_  HOH_196_  HOH_197_  HOH_240_  HOH_241_  HOH_242_
  ILE_7_A VAL_8_A ALA_9_A ILE_16_A LEU_22_A TRP_24_A LEU_27_A GLU_30_A PHE_31_A PHE_34_A THR_56_A SER_59_A
  ILE_60_A PRO_61_A VAL_115_A GLY_116_A GLY_117_A TYR_121_A
2 HOH_239_  HOH_254_  VAL_1_A GLY_2_A SER_3_A LEU_4_A LEU_97_A THR_100_A GLU_101_A LEU_105_A ALA_106_A
  VAL_109_A ASP_110_A VAL_112_A
```

The first column specifies which cluster the residues are in contact with, followed by a list of residues, arranged by residue number and chain.

4.2.5 _clusters.dx

This file contains information about the clusters using the standard DX file (a format also used by the well known program APBS, used to compute electrostatic potential). Most visualization programs are able to handle this format. Figure 4.1 shows a snapshot of the protein 1s3v displayed in `Chimera` together with a .dx file containing the information about the clusters (please refer to section 4.1.1 to learn how this example was generated) and to section 3.2.1 for more information about DX files.

4.2.6 _clusters.pdb

Another option to visualize the results of the calculations carried out by SITEHOUND is to use the '_clusters.pdb' file, that can be loaded in any Molecular Viewers. The clusters have residue name 'C' followed by their ranking number (for example the first cluster has residue name 'C01'), and the chain identifier associated to clusters is the first available letter or number not already utilized by the structure used for the calculation. A few sample lines are shown below:

```
HETATM 1653  C3  C01 B   1    -7.486  31.610  4.165   12.71 100.00
HETATM 1654  C3  C01 B   1    0.514   32.610  0.165   17.45 100.00
HETATM 1655  C3  C01 B   1    0.514   31.610  0.165   16.19 100.00
```

4.3 Visualization

SITEHOUND output can be displayed in most molecular modeling softwares, such as `PyMol`, `Chimera` and `VMD`. Both PDB and DX files can be used. The example shown in figure 4.1 (taken from section 4.1.1) is rendered using a DX file and the visualization tools that `Chimera` offers for handling this file type. Please refer to section 3.3 for more information about visualization.

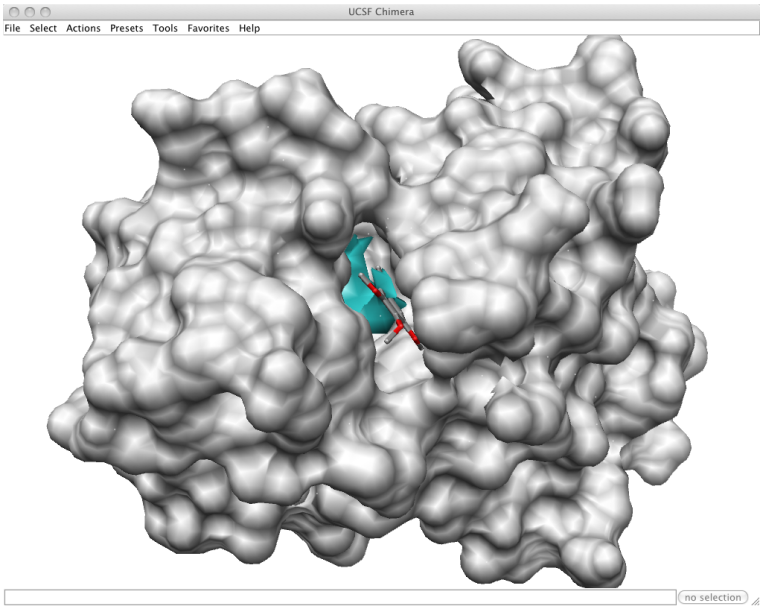


Figure 4.1: **Visualization of binding site detection in `Chimera`** - The Volume Viewer tool available in Chimera allows the user to display the .DX file produced by SITEHOUND. The moving threshold allows the selection of clusters with TIE less than or equal to the threshold itself. In this particular case only the top cluster is shown in cyan, with the ligand overlaid for display

Chapter 5

Methods

5.1 Calculation of MIFs in EASYMIFs

EASYMIFs computes the potential energy between a chemical probe (represented by a particular atom type) and the protein on a regularly spaced grid, using the following equation:

$$V_i = \sum (V_{LJ}(r_{ij}) + V_E(r_{ij})) \tag{5.1}$$

where the potential energy calculated for a probe at a point i in the grid is equal to the sum of a Lennard-Jones and an electrostatics term over all the atoms of the protein. r_{ij} represents the distance between the probe at point i in the grid and an atom j of the protein. The Lennard-Jones and the electrostatics term are expressed by the following two equations:

$$V_{LJ}(r_{ij}) = \frac{C_{ij}^{(12)}}{r_{ij}^{12}} - \frac{C_{ij}^{(6)}}{r_{ij}^{6}} \tag{5.2}$$

$$V_E(r_{ij}) = \frac{1}{4\pi\epsilon_0} \frac{q_i q_j}{\epsilon(r_{ij}) r_{ij}} \tag{5.3}$$

The $C^{(12)}$ and $C^{(6)}$ parameters in the Lennard-Jones term depend on the chosen probe and the particular atom type and are taken from a matrix of LJ-parameters distributed with the GROMACS package[4]. The dielectric constant $\frac{1}{4\pi\epsilon_0}$ has been set to 138.935485. The distance-dependent dielectric sigmoidal function has been taken from Solmajer and Mehler[9] and has the following form:

$$\epsilon(r_{ij}) = A + \frac{B}{1 + \kappa e^{-\lambda B r_{ij}}} \tag{5.4}$$

where $A = 6.02944$; $B = e0A$; $e0 = 78.4$; $\lambda = 0.018733345$; $k = 213.5782$. When the distance between the probe and an atom becomes less than 1.32Å, a dielectric constant of 8 is used. The parameters reported above for the distance-dependent dielectric have been taken from Cui et al.[10]

5.2 Brief overview of clustering in SITEHOUND

The main idea implemented in SITEHOUND is to group the points of the interaction energy map that have passed the energy filter into clusters and to rank them by TIE. It is important to understand the options related to the clustering step in order to effectively use the program. The principles of clustering algorithms and the relevant parameters used by SITEHOUND are discussed here.

The fundamental goal of a clustering algorithm can be considered as finding a partition of a set of points, defined in a multidimensional space, according to some **optimality criterion** (usually, one seeks to minimize intra-clusters distances and maximize inter-clusters distances). It is worth pointing out that the problem is NP-complete, because one should calculate all the possible partitions of the points, a combinatorial problem that scales with the factorial of the number of points. In practice, one can resort to heuristics that make the problem amenable to computation and yield satisfactory results.

More formally, given:

$$\mathbf{x_1} = \{x_{11}, x_{12}, \ldots, x_{1n}\}, \ldots, \mathbf{x_m} = \{x_{m1}, x_{m2}, \ldots, x_{mn}\} \tag{5.5}$$

as a set of m points belonging to an n dimensional space, we can define the following two quantities:

$$D_p(\mathbf{x_1}, \mathbf{x_2}) \tag{5.6}$$

$$D_c(\mathbf{R}, \mathbf{S}) \tag{5.7}$$

that represent the distance between two points $\mathbf{x_1}$ and $\mathbf{x_2}$ and the distance between two clusters $\mathbf{R}$ and $\mathbf{S}$, respectively. A natural choice for D_p in our problem is the simple euclidean distance between the points.

One of the most widely used heuristics to approach the clustering problem is to proceed from to the bottom to the top by iteratively merging clusters until one cluster containing all the points is obtained. This is where the D_c quantity plays a role, by defining the distance between clusters. The name **linkage** is commonly used to indicate this quantity.

SITEHOUND incorporates two types of linkage, *single* and *average*, defined in the following way:

$$D_{c\_single}(\mathbf{R}, \mathbf{S}) = \min_{\mathbf{x_1} \in \mathbf{R}, \mathbf{x_2} \in \mathbf{S}} D_p(\mathbf{x_1}, \mathbf{x_2}) \tag{5.8}$$

$$D_{c\_average}(\mathbf{R}, \mathbf{S}) = \frac{\sum_{\mathbf{x_1} \in \mathbf{R}} \sum_{\mathbf{x_2} \in \mathbf{S}} D_p(\mathbf{x_1}, \mathbf{x_2})}{|\mathbf{R}||\mathbf{S}|} \tag{5.9}$$

where the $|\ |$ notation indicates the cardinality of the set (i.e. the number of points of the cluster).

Two important properties shared by these two linkages are the fact that the distance between clusters increases monotonically at each step. Therefore, it is possible to cut the partition at a particular level

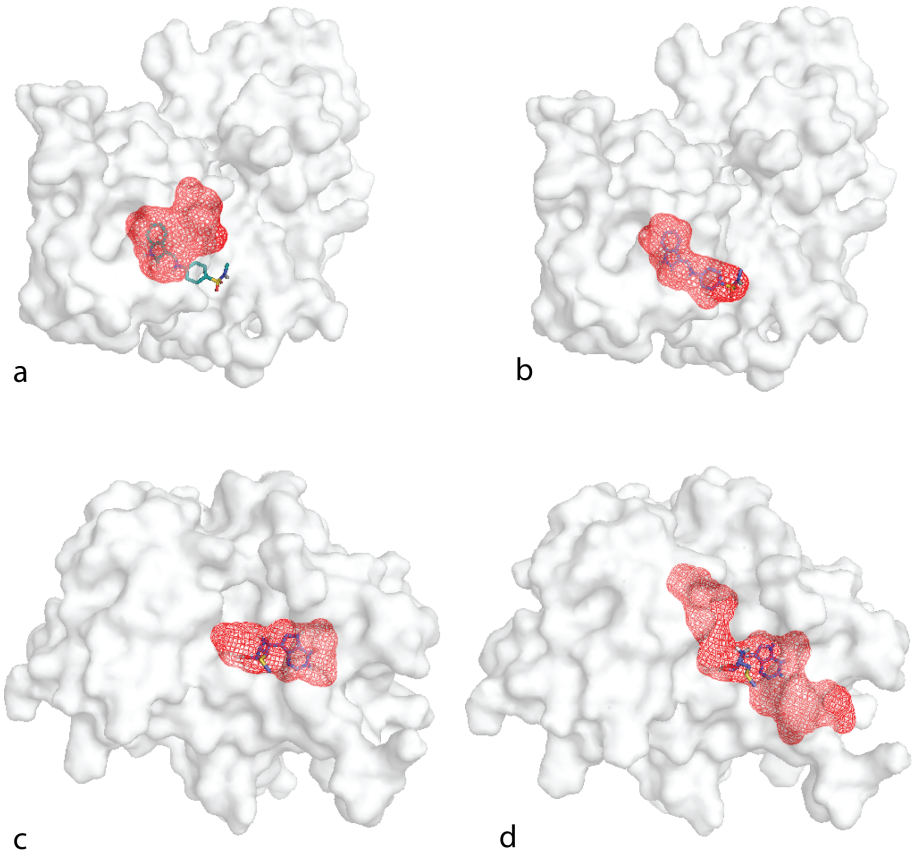


Figure 5.1: **Effects of linkage on clustering results** - a) and b) show the results of average and single linkage on cyclin-dependent kinase 2 (PDB code 1ke5). Single linkage yields a better coverage of the binding pocket, which is quite elongated. On the other hand, for human pregnenolone sulfotransferase (PDB code 1q1q) average linkage is the best choice, since it corresponds more closely to the ligand contour.

obtaining the corresponding clusters. In SITEHOUND this level is called **spatial cutoff**. The type of linkage used affects (to some extent) the shape of the clusters obtained. In general, it can be shown that single linkage tends to yield more elongated clusters, whereas with average linkage the shape of the clusters is closer to a sphere. From a practical point of view, using single linkage can be more meaningful with peptide binding sites or elongated ligands, whereas average linkage performs better with small chemicals. These effects are illustrated in Figure 5.1. In general, it is desirable to run the calculations with both types of linkage, and compare the results. In some instances, with average linkage the binding site is split in two regions, whereas single linkage will tend to show one single site. This information could be valuable in the context of ligand design, since the two regions that show up with average linkage could both be exploited by connecting two fragments with a linker.

Bibliography

[1] Gabriele Cruciani. *Molecular Interaction Fields: Applications in Drug Discovery and ADME prediction*. Wiley-VHC, 2006.

[2] P. J. Goodford. A computational procedure for determining energetically favorable binding sites on biologically important molecules. *J. Med. Chem.*, 28:849–857, 2009.

[3] G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, A. J. Olson, and Nc. Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function. *Journal of Computational Chemistry*, 19(14):1639–1662, 1998.

[4] D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. Berendsen. Gromacs: fast, flexible, and free. *J Comput Chem*, 26(16):1701–18, 2005.

[5] A. T. R. Laurie, R. M. Jackson, and Rs. Q-sitefinder: an energy-based method for the prediction of protein-ligand binding sites. *Bioinformatics*, 21(9):1908–1916, 2005.

[6] M. Hernandez, D. Ghersi, and R. Sanchez. SiteHound-web: a server for ligand binding site identification in protein structures. *Nucleic Acids Research*, 37:W413–416, 2009.

[7] D. Ghersi and R. Sanchez. Improving accuracy and efficiency of blind protein-ligand docking by focusing on predicted binding sites. *Proteins*, 74:417–424, 2009.

[8] U. Abele and G.E. Schulz. High-resolution structures of adenylate kinase from yeast ligated with inhibitor ap5a, showing the pathway of phosphoryl transfer. *Prot. Sci.*, 4:1262–1271, 1995.

[9] T. Solmajer and E.L. Mehler. Electrostatic screening in molecular dynamics simulations. *Protein Eng*, 4(8):911–7, 1991.

[10] M. Cui, M. Mezei, and R. Osman. Prediction of protein loop structures using a local move monte carlo approach and a grid-based force field. *Protein Eng Des Sel*, 21(12):729–35, 2008.

Appendix A

List of EASYMIFs probes

Table A.1: List of EASYMIFs probes

Code	Name
O	carbonyl
OM	carboxyl
OA	hydroxyl
OW	water
N	peptide
NT	terminal
NL	terminal
NR	aromatic
NZ	Arg
NE	Arg
C	bare
CH1	aliphatic
CH2	aliphatic
CH3	aliphatic
HC	hydrogen
H	hydrogen
S	sulfur
CU1+	copper
CU2+	copper
FE	iron
ZN2+	zinc
MG2+	magnesium
CA2+	calcium
P	phosphor
AR	argon
CMET	carbon of CH3-group
OMET	oxygen of CH3-group
NA+	sodium
CL-	chlorine
CCHL	carbon
HCHL	hydrogen
SDMSO	DMSO
OP	phosph-oxygen

Appendix B

The SITEHOUND-web Server

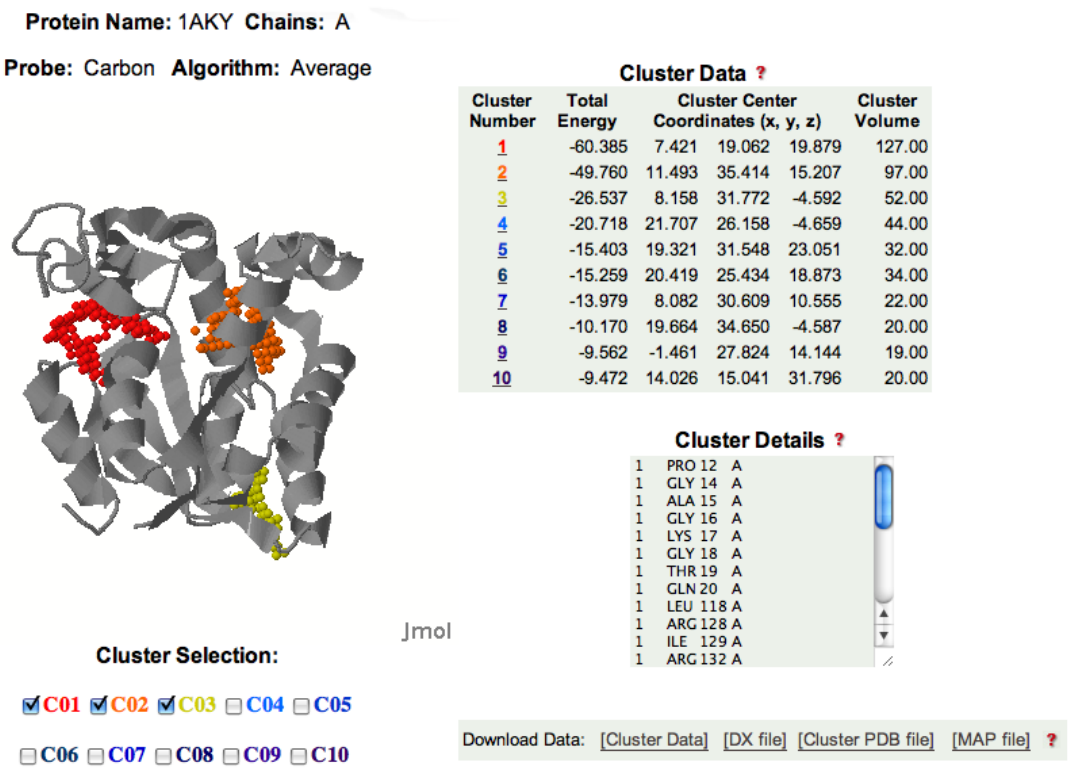


Figure B.1: SITEHOUND-web **results page**. The results of a "carbon" probe calculation with the average linkage clustering are shown for for adenylate kinase (PDB code 1aky). The Web Interface is available at `http://sitehound.sanchezlab.org`.

A streamlined web-based interface to carry out binding site detection using SITEHOUND is available at `http://sitehound.sanchezlab.org`. The interface (Figure B.1) can be used to upload a PDB structure, automatically perform the binding site detection and visualize the results of the calculations on a ribbon representation of the protein. The residues potentially involved in binding are also reported on a per-cluster basis, together with a summary of the main features of the clusters. From the results page (Figure B.1) the user can also download all the files that are produced by SITEHOUND and that

are described in detail in Chapter 4. Furthermore, it is possible to download the '.map' file produced by EASYMIFs or `Autogrid`, which can used by SITEHOUND to carry out the binding detection with combinations of parameters different from the default parameters used by the web server. SITEHOUND-web only allows for the processing of relatively small systems with default parameters. Larger systems, different parameters, and the processing of large numbers of files require the use of the standalone EASYMIFs and SITEHOUND programs described in this manual. For details on SITEHOUND-web see [6].

Appendix C

How to calculate interaction energy maps with Autogrid

C.1 Installing `Autogrid` and `Autodock tools`

SITEHOUND can also use interaction energy maps produced by the `Autogrid` program, which can be downloaded from `http://autodock.scripps.edu/resources/adt/index_html`. An installation of `Autodock Tools` is also recommended, since the package comes with a script for generating PDBQT files (the format used by `Autogrid`) starting from a PDB.

C.2 Preparing the PDBQT file

The first step in binding site detection requires the calculation of an interaction energy with a Carbon probe using the `Autogrid` program. It is recommended to remove from the PDB water molecules or other HETATM records (heteroatoms such as ligands). Again, the Web Server interface to SITEHOUND carries out this step automatically. `Autogrid` uses the PDBQT file format (an enhanced PDB format), that can be easily obtained from a PDB file by using the `prepare_receptor4.py` script that comes with the `Autodock Tools` package. An example of a typical usage is shown below:

```
prepare_receptor4.py -A hydrogens -r PDB
```

where PDB is the name of the PDB file that has to be processed. The '-A hydrogens' forces the addition of polar hydrogens to the protein. If the protein is already protonated, neglect this option.

C.3 Preparing the gpf file

Once the pdbqt file has been successfully produced, it is necessary to create a gpf file that is used by the `Autogrid` program to calculate the interaction energy map.
A convenient script to automate this step is available at ... and can be used in the following way:
`create_gpf.py -r PDBQT -t TEMPLATE -c 5.0 -s 1.0`

where PDBQT stands for the file generated in the previous step, TEMPLATE is a .gpf file that comes with the script and contains a set of standard parameters for `autogrid` and the options '-c' and '-s' specify the clearance of the box and the resolution of the grid in Angstroms, respectively. The script calculates the center of the protein and uses it to center the grid. The size of the box that encloses the protein is determined on the basis of the protein dimensions and the clearance encoded in the '-c' option. The values reported above are typical, but the user can explore other values, bearing in mind that the higher the resolution the larger the computational requirements in terms of time and space.

C.4 Running `Autogrid`

This step yields the .C.map file that is the input to SITEHOUND . In order to obtain this interaction energy map, just type the following:

```
autogrid -p GPF -l GLG
```

where GPF is the .gpf file produced in the previous step and GLG file is the name of the file where `autogrid` stores a log of the calculation (any name will do). At the end, a .C.map file containing the interaction energy map will be generated. Please note that **the units are kcal/mol**, compared to EASYMIFs kJ/mol.

Appendix D

Software Versions

The versions are indicated with a suffix that specifies the date. It is usually best to use the latest version, since it may contain bug fixes or additional features.

- 011209 - EasyMIFs and SiteHound now support compressed maps, useful to save space on large scale analyses