EPOS^{BP}

Ensemble of Pockets on Protein Surfaces with BALLPass

User Manual for Version 1.0
http://gepard.bioinformatik.uni-saarland.de/software/epos-bp

Susanne Eyrisch
eyrisch@bioinformatik.uni-saarland.de

1. Input Coordinates

All ligands, solvent molecules, and other hetero atoms have to be removed before running EPOS^{BP}. The protein file must be supplied in PDB or HIN format. If a PDB file contains several models, the PASS algorithm is applied to each of them. We recommend splitting files containing multiple protein chains, because two atoms stemming from the different structures are defined to be identical if their residue name, residue number, and atom name match. (The chain ID is not taken into account to allow for the comparison of pockets detected in two distinct chains of the same protein.)
Ligand coordinate files can be given in MOL2, PDB, or HIN format.

2. Command Line Options

2.1. Reading Protein Structures

-file <PDB/HIN file>	apply the PASS algorithm to a single PDB or HIN file
-list <file>	apply the PASS algorithm to the PDB or HIN files listed in <*file*>
-read <file>	read in the patch files from a previous run listed in <*file*>

2.2. Clustering Similar Pockets

-cluster <cutoff> <use index> <cluster file>	cluster pockets with similarities less than <*cutoff*> percent, write the clustering results and rename the patch and PLA files (required options: 'file', 'list', or 'read'); only set <*use index*> to 1 if the atoms have the same index in all files
-readclust <cluster file>	read in a previously calculated cluster file and apply the clustering to the read-in patches (required option: 'read')

2.3. Calculating Pocket Properties

-analyse <analysis file>	analyse the pocket properties of the different pocket clusters and write the results to an output file
-subpocket <prefix> <sim cutoff>	write the subpocket (PLAs that are present in at least <*sim cutoff*> percent of all PLAs) of each pocket cluster to files with the given prefix (required options: 'analyse')
-overlap <ligand file> <overlap file>	calculate the overlap volume between a given ligand (in pdb, hin, or mol2 format) and the patches (required options: 'file', 'list', or 'read')
-compare <file1> <file2> <similarity table>	write the pairwise similarities of the PLAs or subpocket files listed in *file1* and *file2* to the given output file (format: one file with path per line)

2.4. Miscellaneous

-v	verbose mode: write information about the patches to STDOUT

3. PASS Parameter File ("BALLPass.ini")

If you want to use your own parameters instead of the default values make sure that a parameter file called *BALLPass.ini* is available in your current directory and that the path of the file containing the atom radii is correct.

Entry	**Description**	**Default**
HEAVY_ONLY	ignore hydrogens	1
PARSE_INI_FILE	use parameters defined in local parameter file instead of default values	1
RADIUS_HYDROGEN	radius of a hydrogen atom [Å]	1.2
RADIUS_OXYGEN	radius of an oxygen atom [Å]	1.52
RADIUS_NITROGEN	radius of a nitrogen atom [Å]	1.55
RADIUS_CARBON	radius of a carbon atom [Å]	1.7
RADIUS_SULFUR	radius of a sulfur atom [Å]	1.8
PROBE_SPHERE_RADIUS	radius of a probe in the 1. layer when hydrogens are considered [Å]	1.5
PROBE_SPHERE_RADIUS_ HYDROGEN_FREE	radius of a probe in the 1. layer when hydrogens are ignored [Å]	1.8
PROBE_LAYER_RADIUS	radius of a probe in the accretion layers [Å]	0.7
MINIMUM_PROBE_SEPARATION	minimal distance between two probes [Å]	1.0
BURIAL_COUNT_THRESHOLD	minimal number of surrounding protein atoms for defining a probe as buried probe when hydrogens are considered	75
BURIAL_COUNT_THRESHOLD_	minimal number of surrounding	45

HYDROGEN_FREE	protein atoms for defining a probe as buried probe when hydrogens are ignored	
BURIAL_COUNT_RADIUS	Radius used for computing the burial counts of a probe [Å]	8.0
PW_SQUARE_WELL	Parameter for defining the probe weight envelope function (see Brady and Stouten, 2000)	2.0
PW_GAUSSIAN_WIDTH	Parameter for defining the probe weight envelope function (see Brady and Stouten, 2000)	1.0
ASP_SEPARATION	minimal distance between two ASPs [Å]	8.0
MINIMUM_PROBE_WEIGHT	minimal probe weight for an ASP	1150
CLASH_FACTOR	factor for reducing clashes between probes and protein atoms	0.95
RADII_FILE	file containing the radii of the protein atoms	PARSE.siz

4. File Formats

4.1. Input Files

4.1.1. Reading Multiple Input Files (Option '-list')

One PDB/ HIN file with path per line.

4.1.2. Reading Previously Calculated Pockets (Option '-read')

One patch file with path per line.

4.1.3. Comparing (Sub-) Pockets (Option '-compare')

One PLAs/ subpocket file with path per line.

4.2. Output Files

4.2.1. PLAs-/ Subpocket-Files

Atoms of the input protein in PDB format.

4.2.2. Patch Files

Each probe is represented by a carbon atom (initial layer) or hydrogen atom (accretion layer) in PDB format. The atom name is the atom symbol followed by the number of the layer, the residue name is 'PKT', and the residue number corresponds to the pocket ID.

4.2.3. Cluster Files

```
<file prefix 1>: <old PID> -> <new PID>
<file prefix 1>: <old PID> -> <new PID>
…
<file prefix n>: <old PID> -> <new PID>
```

These output files are generated after the clustering procedure. Note that they may also be generated manually.

4.2.4. Analysis File

```
 PID    freq[%]   mean vol[A^3] min vol[A^3] max vol[A^3] mean pol  min pol   max pol   mean depth[A] min depth[A] max depth[A]
<PID 1> <value 1> <value 2>     <value 3>    <value 4>    <value 5> <value 6> <value 7> <value 8>     <value 9>    <value 10>
<PID 2> <value 1> <value 2>     <value 3>    <value 4>    <value 5> <value 6> <value 7> <value 8>     <value 9>    <value 10>
…
<PID n> <value 1> <value 2>     <value 3>    <value 4>    <value 5> <value 6> <value 7> <value 8>     <value 9>    <value 10>
```

4.2.5. Overlap File

```
<file prefix 1> <value 1> <value 2> PIDs: <PID 1> <PID 2> … <PID n> <value 3>
<file prefix 2> <value 1> <value 2> PIDs: <PID 1> <PID 2> … <PID n> <value 3>
…
<file prefix n> <value 1> <value 2> PIDs: <PID 1> <PID 2> … <PID n> <value 3>
```

Here, <*value 1*> corresponds to the volume of the "reduced" patch (consisting only of those probes that are overlapping with the ligand atoms) and <*value 3*> corresponds to its polarity. <*value 2*> is the percentage of the ligand atoms overlapping with the PASS probes.
Note that the overlaps are calculated per structure and not per patch. All pocket IDs involved in the overlap are listed after the keyword "PIDs:".

4.2.6. Similarity Table

```
<sim(f1:1,f2:1)> <sim(f1:1,f2:2)> … <sim(f1:1,f2:m)>
<sim(f1:2,f2:1)> <sim(f1:2,f2:2)> … <sim(f1:2,f2:m)>
…
<sim(f1:n,f2:1)> <sim(f1:n,f2:2)> … <sim(f1:n,f2:m)>
```

Here, `<sim(f1:i,f2:j)>` is the similarity (percentage of common PLAs) between the *i*th entry in <*file 1*> and the *j*th entry in <*file 2*>.

5. Example Applications

The files needed to run these example applications can be downloaded from
http://gepard.bioinformatik.uni-saarland.de/software/epos-bp

5.1. Calculating and Analyzing the Transient Pockets Opening During a Molecular Dynamics Simulation

```
EPOS -file examples/ex_1/traj.pdb -cluster 75 1 examples/ex_1/traj.clust
-analyse examples/ex_1/traj_analysis.dat -subpocket
examples/ex_1/traj_subpocket 50
```

The input file "`traj.pdb`" contains 100 frames. For each frame representing a snapshot from the MD simulation, the pockets accessible in this conformation are calculated. Subsequently pockets from different frames are assigned to the same cluster as long as their pairwise similarity calculated using the pocket lining atoms (PLAs) is at least 75%. Here, the index of the atoms can be used for identifying identical atoms, as it is consistent in all frames of the trajectory. All pockets assigned to the same cluster are considered as states of the same transient pocket. The results of the clustering is written to "`traj.clust`". Finally, the properties of the transient pockets are calculated and their subpockets are determined, i.e. the PLAs that line this transient pocket in at least 50% of all its states.

5.2. How Open is the Native Binding Pocket in a Conformational Ensemble?

```
EPOS -list examples/ex_2/conf_ensemble.txt -cluster 75 0
examples/ex_2/conf_ensemble.clust -overlap examples/ex_2/ligand.pdb
examples/ex_2/conf_ensemble_overlap.dat
```

"`conf_ensemble.txt`" contains the names of 100 PDB files containing different conformations of the same protein. As the atom index of these files do not necessarily have to be consistent, the residue number, the residue name, and the atom name are used to define the similarity of the PLAs of two pockets. By clustering the individual pockets the set of transient pockets is determined. Afterwards, for each input structure defined in "`conf_ensemble.txt`", the overlap of the pocket probes with the ligand coordinates is used to calculate the overlap volume, the fraction of overlapped ligand atoms, and the ID of the involved transient pockets. These results representing how open the native ligand binding pocket is in this conformational ensemble is written to "`conf_ensemble_overlap.dat`".

5.3. Calculating the Properties of the Native Binding Pocket

```
EPOS -file examples/ex_3/native_bound.pdb -analyse
examples/ex_3/native_bound_analysis.dat -overlap examples/ex_3/ligand.pdb
examples/ex_3/native_bound_overlap.dat
```

This example shows (1) how one can extract the native ligand pocket among a set of pockets without visual inspection and (2) how one can derive reference values for the overlap volume (e.g. if one wants to determine to what degree the native ligand binding pocket is open in a conformational ensemble). Here, the pockets of only one structure are calculated and analyzed. The ID of the pocket accommodating the ligand is given in "`native_bound_overlap.dat`".

5.4. Comparing two Ensembles of Pockets

```
EPOS -compare examples/ex_4/ensemble1.txt examples/ex_4/ensemble2.txt
examples/ex_4/ensemble_comparison.dat
```

Comparing the two sets of transient pockets calculated from different conformational ensembles is often of interest. This can be efficiently done by calculating the pairwise similarities of the subpockets. In this example, "`ensemble1.txt`" contains the file names of the subpockets of the first conformational ensemble and "`ensemble2.txt`" contains the file names of the subpockets of the second one. All pairwise similarities are then written to "`ensemble_comparison.dat`".