



## 1 Preparing to use SLIDE

(User Guide Chapter 3)

**Setting SLIDE environment variables.** Add the following lines to your `.cshrc` file. (If you typically use a shell environment other than `c` shell and find you are having trouble with system interpretation of the environment variables, we suggest switching to a `c` shell by typing `csh` at the command prompt and working in that window.)

```
setenv SLIDE_DIR <global_installation_dir> (where the program resides, e.g.,  
/usr/bin/slide)
```

```
setenv PATH ${SLIDE_DIR}/bin:${PATH} (to add $SLIDE_DIR/bin to the path)
```

```
mkdir <user_SLIDE_output_dir> (make a directory where all your SLIDE output will be  
written, e.g., /home/username/slide_data)
```

```
setenv SLIDE_DATA_DIR <user_SLIDE_output_dir>
```

Then, in the window(s) in which you'll be doing steps 2-4 below:

```
source .cshrc
```

**SLIDE tutorials.** To familiarize yourself with common ways of running SLIDE, we recommend working through the five tutorials in `$SLIDE_DIR/docu/Slide_Tutorial_v3.3.5.pdf` and comparing your output with that provided in `$SLIDE_DIR/examples/slide_data` (where `$SLIDE_DIR` is the directory in which you installed SLIDE). In case of any discrepancies, contact [slide@sol.bch.msu.edu](mailto:slide@sol.bch.msu.edu).

The steps in this Quick Guide explain how to set up a small-molecule ligand database for screening by SLIDE, generate an effective protein template, and run SLIDE to dock and screen ligand candidates for your target protein. Text in `<angular brackets>` should be replaced by actual names that you provide. In all the commands below:

`<target>` is the nametag you assign for the directory for your protein target

`<template>` is the nametag you assign to that protein target's template directory

`<database>` is the nametag you provide for the database of ligand candidates to be docked

The annotated example of a SLIDE data directory (second to last section of this guide) will help you relate these nametags to the resulting file structure created by SLIDE. For more details on the methodology behind the steps in SLIDE, see the SLIDE User Guide in the `$SLIDE_DIR/docu` directory and the following publications.

**SLIDE publications** are available at <http://www.kuhnlab.bmb.msu.edu/software/slide/>

The primary references are: M. I. Zavodszky, P. C. Sanschagrin, R. S. Korde, and L. A. Kuhn (2002) *J. Comp.-Aided Molecular Design*, 16, 883-902 and V. Schnecke, C. A. Swanson, E. D. Getzoff, J. A. Tainer, and L. A. Kuhn (1998) *Proteins* 33, 74-87.

## 2 Preparing SLIDE input data

(User Guide Sect. 3.3.3, 3.4, 3.6.2)

**Preparing the protein PDB file for screening.** In the protein PDB file, hydrogen atoms will be removed by SLIDE. For terminal H's that can assume different positions relative to the donor and pre-donor atoms (e.g., H of a hydroxyl group), SLIDE uses favored bond lengths and angles to calculate whether an H atom from the protein or ligand can be placed in a position that allows hydrogen bonding.

Non-water cofactors (HETATMs) in the binding site are handled as rigid parts of the protein during docking. Alternatively, they can be removed from the PDB file by text-editing, and this should be done for ligands that occlude the intended binding site for the ligand candidates being screened. If you choose to include cofactors as part of the protein, cofactor hydrogen bond donor, acceptor or donor/acceptor atom names in the PDB file should be relabeled (left-justified) as NDD (nitrogen acting as H-bond donor), NAA (nitrogen acting as H-bond acceptor), NNN (nitrogen acting as H-bond donor and/or acceptor), ODD (oxygen acting as H-bond donor), OAA (oxygen acting as H-bond acceptor), or ONN (oxygen acting as H-bond donor/acceptor). These atoms will be recognized by SLIDE as contributing to the complementarity score between protein and ligand.

Bound water molecules (HOH residues) in the PDB file are ignored. Water molecules can also be treated as a rigid part of the protein structure by preparing them as cofactors, as described above. However, this places strong constraints on docking, since the water molecules are then non-displaceable. Template points will be placed outside the water layer such that ligand atoms at the template point positions can interact with the water molecules. SLIDE scoring does not currently consider water-mediated interactions, since doing so decreased the scoring accuracy.

**Preparing ligand candidates for screening.** Correct preprocessing of the molecules to be screened by SLIDE is important. An excellent source of several million, commercially available compounds in 3D structure format, with partial charge and proton assignments, is ZINC (<http://zinc.docking.org>). Only one conformation is provided for each structure in ZINC, and other low-energy conformations should be sampled using the Omega software from OpenEye Scientific Software (<http://www.eyesopen.com>), as described below.

*To prepare a mol2 file for docking a small molecule, given a PDB file containing the ligand, text-edit the PDB file to contain only the ligand (in particular, its ATOM and/or HETATM lines), and then use the molcharge program (OpenEye; <http://www.eyesopen.com>) to calculate partial charges for the atoms and convert the ligand atomic coordinates to mol2 format:*

```
molcharge -in 1a4w_cryst_ligand.pdb -out 1a4w_cryst_ligand.mol2 -am1bcc
```

Note that the character in the middle of "am1bcc" corresponds to the number one rather than the letter L. This file conversion may result in a missing molecule name on the line following "@<TRIPOS>MOLECULE" in the mol2 file. If so, text-edit the file and provide a short name for the molecule on that line, which will be used by SLIDE to name the docked coordinate files. AM1BCC charges will not be assigned to molecules with unusual atom types, for example B, Co, etc. In that case, AM1 charges can be assigned, using the "-am1" flag instead. (Note: The unusual atom type will be named Du by molcharge, and will need to be renamed to the original atom type in the mol2 file.)

*If only 2D structure files are available for the small molecules of interest, an initial, low-energy 3D conformer can be generated for each molecule using a tool such as Corina (<http://www.molecular-networks.com/products/corina>). Partial charges and hydrogen atom positions for polar groups in ligand candidates can then be assigned, as described above. Recommended Corina usage:*

```
corina -i t=sdf -o t=mol2 -d wh,stergen,preserve,rc,rs,mc=10,names
<input_filename>.sdf <output_filename>.mol2
```

*Sampling 3D conformers of flexible ligands.* Given a 3D conformer, SLIDE will dock the ligand candidate and model limited ligand and protein side-chain flexibility. However, SLIDE does not exhaustively search ligand conformations during docking. Therefore, it is important to provide a panel of low-energy ligand conformations as input for screening if a ligand or ligand candidate is not necessarily near its bioactive conformation. We recommend using Omega, which is freely available to academic users (<http://www.eyesopen.com>), to generate low-energy conformations as input for SLIDE screening. When available, we also recommend including Protein Data Bank (<http://www.rcsb.org/pdb>) and Cambridge Structural Database (<http://www.ccdc.cam.ac.uk/products/csd>) crystal structures of the ligand, as part of the panel of low-energy conformations provided to SLIDE for screening. We use the following options in Omega version 2.1.0:

```
omega2 -in <input_filename>.mol2 -out <output_filename>.mol2 -includeInput
[true/false] -warts true -buildff mmff94s -fraglib <fraglib.oeb.gz location,
provided with omega2>
```

If `-includeInput` is set to “true”, which results in the input conformation for each molecule being included in the multi-mol2 file of conformers, then a conformer ID should be added to that conformation before running SLIDE. In Omega2.1 output, this is the first set of coordinates for each molecule. The SLIDE script `mol2_add_conf_0.pl` can be used to add the identifier “\_0” to the initial conformer of each molecule. In general, each ligand candidate should have a distinct name and conformer ID or errors will result, such as output file overwriting.

*Underscores in molecule names.* SLIDE uses the entry contained under `@TRIPOS<MOLECULE>` in the mol2 file as the molecule name. If the `GROUP_CONFORMERS` option is set to `TRUE` in the `slide.parameters` file (the default), such that only the best docking is kept for a set of conformers of a molecule, then SLIDE expects molecules to be named as follows:

```
<MOLECULENAME>_<CONFORMERID>
```

where the molecule name is everything preceding the final underscore character ‘\_’ and the conformer ID is everything following that underscore. For instance, `ZINC_123456789_1` would be assigned as the first conformer of molecule `ZINC_123456789`. If an underscore already exists in the `MOLECULENAME` and conformers were not generated for it, then either a conformer ID should be added to the molecule name (i.e. “\_0”), all underscores should be removed from the name, or `GROUP_CONFORMERS` should be set to false. Otherwise, SLIDE will assign anything after the final underscore as the conformer ID (i.e. `ZINC_123456789` will assigned as the 123456789<sup>th</sup> conformer of the molecule “ZINC”).

**Setting up the SLIDE database structure for creating a new template and screening database.** The command, `setup_dbase`, creates all the subdirectories needed to run SLIDE, then extracts interaction centers (for hydrophobic centers and hydrogen-bond donors and/or acceptors) from the molecules in `<dbase_loc>` and stores them in the `$SLIDE_DATA_DIR/databases/<database>` directory as “.pts” files. First, have your protein target PDB file (with ligands removed from the binding site) and ligand database ready. The ligand database (which may consist of a single ligand to be docked) should be in its own directory, referred to as `<dbase_loc>` below. The `<template>` nametag below is used for naming a new subdirectory to contain the template files to be generated; thus, `setup_dbase` should also be done before template generation. SLIDE now supports a mixture of single-molecule mol2 files and multi-mol2 files (each containing up to thousands of molecular entries) in the database directory, `<dbase_loc>`. Once the target PDB file and ligand database are ready, type the command:

```
setup_dbase <target> <template> <database> <dbase_loc> <target>.pdb
Ex: setup_dbase 7dfr unbiased known_ligands /db/csd ~/7dfr_prepared.pdb
```

*File structure created by setup\_dbase.* In the command above, <target>.pdb is the filename of the prepared protein PDB file (with path). In the example above, /db/csd is the directory where the files for the known\_ligands database are located and ~/7dfr\_prepared.pdb is the protein PDB file. The nametag <template>, which describes the template and names the corresponding directory, typically corresponds to one of the types described below in section 3, e.g., “biased” or “unbiased”. However, the template directory can be named anything, as long as the same nametag is used for all steps in SLIDE. An example file structure appears near the end of this document, corresponding to the input and output of SLIDE runs from the Tutorial in `soft/linux64/slide/docu`.

*Naming of interaction point files created by setup\_dbase.* Typically the user does not need to work with the `pts` files created by `setup_dbase`. However, if you are interested in how this data is organized, the chemical-type labeled interaction points (`.pts`) files generated from the input multi-mol2 database files (containing multiple molecules and/or conformers) are named as `<multi-mol2_filename>_<n>.pts` in the `$SLIDE_DATA_DIR/databases/<database>` directory, where `<n>` is any number starting from 0. Each `.pts` file can hold the interaction centers for up to 7000 ligands. Subsequent `.pts` will be created if there are more than 7000 ligands in a multi-mol2 file, generating `<multi-mol2_filename>_<n+1>.pts`, etc.. It is recommended that multi-mol2 files not exceed 1 GB in size (smaller is better) due to system file handling limitations on 32-bit machines, as well as increased access times for molecules near the end of a large file. For mol2 files containing only one molecule, the `.pts` files are named `singleton_<n>.pts`. Each contains interaction centers for up to 7000 individual mol2 files. Subsequent `.pts` files will be created if there are more than 7000 single-molecule mol2 files in a single subdirectory in `<dbase_loc>`, e.g. `<singleton>_<n+1>.pts`, and so on.

**Reusing an existing database with a new protein structure or template.** If files of interaction points were already created for your screening database (by previously running `setup_dbase`), and you would like to screen that database for a new protein target or create a new template for an existing protein target, then you can use `slide_setup.pl` instead of `setup_dbase`. This script avoids recalculating the ligand interaction points for the database, and instead just sets up the new target and template subdirectories under your existing `$SLIDE_DATA_DIR`, if one or the other does not already exist. Usage:

```
slide_setup.pl <target> <template> <database>
Ex: slide_setup.pl 7dfr unbiased known_ligands
```

### 3 Generating a protein template

**How is a SLIDE template used?** The template reflects favorable points to place ligand hydrophobic groups or hydrogen-bonding atoms for making good interactions with the protein. Note that screened ligands do not need to fall entirely within the template volume; instead, three interaction points of the ligand candidate must match three points of the template for docking to occur. At that point, side-chain flexibility is modeled if needed to remove van der Waals overlaps between atoms, followed by full scoring of all atomic interactions for ligand candidates that dock well sterically.

The template is not only useful in SLIDE as the basis for docking ligands by matching ligand interaction points with template points, but also can guide the optimization of known ligands by indicating sites where the ligand has good or poor complementarity with the protein. For instance, a hydrophobic

template point overlaying with a polar atom in the ligand represents an unfavorable contact, and the presence of template points in an unfilled pocket represents missed opportunities for interacting with the protein. The PDB-formatted files provided as a result of template generation (`template.pdb` or its component files, `hphob.pdb`, `donor.pdb`, `acceptor.pdb`, and `doneptor.pdb`) can be read into a molecular graphics program and colored according to the chemistry type each file represents, for visualization alongside the protein structure and known ligands or docked ligand candidates. In each `.pdb` file, acceptor template points have the B-value (last column) set to 0; doneptor points, 25; donor points, 50; and hydrophobic points, 100.

A good template should thoroughly sample the binding site of interest, while not including extraneous points. It is highly recommended to visually check the template with molecular graphics alongside your protein target structure and any known ligand(s) (for reference) before proceeding to screen a database. Multiple templates can be generated and kept under the same `$$SLIDE_DATA_DIR/<target>` directory. See the `slide_setup.pl` command at the end of section 2.

**Handling heteroatoms in template generation.** Water molecules in the protein input PDB file are ignored in template generation. If desired, water sites can be added manually as donor/acceptor (doneptor; N) template points, indicating favorable positions to place H-bonding atoms. Do this by text-editing `$$SLIDE_DATA_DIR/<target>/<template>/in/template` (note that this is not the `template.pdb` file) to add the coordinates of water molecules as new template points. In the `template` file, points are labeled according to the atom type at that position that could interact favorably with the protein: A for hydrogen-bond acceptor, D for hydrogen-bond donor, N for hydrogen-bond acceptor and/or donor (due to presence of both hydrogen-bond acceptor and donor atoms in the protein at an appropriate distance and angle) and H for hydrophobic atom. The use of \* to design especially important template points is described below in the “Key points” section. Other heteroatoms (non-protein atoms) in the protein PDB file contribute to the steric aspect of docking and will prevent docking if they block the intended binding site. See the beginning of section 2 on protein PDB file preparation.

**Creating a “biased” ligand-based (pharmacophore) template** (User Guide Sect. 5.3, 5.9.3). This type of template, created using the `-g` option, is based on clustering hydrophobic or hydrogen-bonding interaction centers from a set of known ligands. Essentially, this generates a 3D pharmacophore as the basis for docking and will bias dockings towards known ligand interactions. It will also bias any redocking experiment, and thus should not be used for validating the ability of SLIDE to redock the known ligands. For the clustering threshold, specify 0 if a single known ligand is used as input or 0.5-2.0 Å if multiple ligands are used. This threshold governs the distance between atoms of the same type (e.g., H-bond donor) in the overlaid, known ligands for them to be considered as representing a shared feature and clustered to generate a single template point. Before generating a protein template, set up the template subdirectory structure by using `setup_dbase` or `slide_setup.pl`, as described on the previous page.

```
temp_gen -g <target> <template> <clustering_threshold> <ligand_1>.mol2
[<ligand_N>.mol2]
```

```
Ex: temp_gen -g 7dfr biased 2.0 /ligands/AZUDEC10_*.mol2
```

where `<ligand_1>.mol2` [`<ligand_N>.mol2`] are known ligands given in their protein-bound orientations (such that they overlay with each another in the binding site).

**Creating an unbiased, protein-based template** (User Guide Sect. 5.4, 5.9.3). This version of `temp_gen` creates an unbiased template representing ideal positions for ligand atoms to interact, from the protein’s perspective. In this case, any ligand(s) provided as input only defines the central volume of the binding site, to which a cushion is added. Hydrogen-bonding template points are set down in this volume based

on geometrically defined ideal positions. Hydrophobic probe points are densely sampled in the volume, and then clustered down to a representative set of points non-overlapping with hydrogen-bonding points. Before generating an unbiased template, set up the template subdirectory structure by using `setup_dbase` or `slide_setup.pl`, as described on the previous page.

To generate an unbiased template with the binding site defined based on the minimum and maximum x, y, and z coordinates of atoms in a set of known ligand(s) in their protein-bound orientations:

```
temp_gen -l <target> <template> <Hbonding_point_density> <grid_spacing>
<clustering_threshold> <ligand_1>.mol2 [<ligand_2>.mol2 <ligand_N>.mol2]
```

**Ex:** `temp_gen -l 7dfr unbiased dense 0.5 3.5 AZUDEC10_2.mol2` (note that the flag `-l` includes the letter L, not the number one)

To specify the binding site using a sphere centered at  $[x,y,z]$  with radius  $<r>$  (typically 9-10 Å):

```
temp_gen -c <target> <template> <Hbonding_point_density> <grid_spacing>
<clustering_threshold> <x> <y> <z> <r>
```

To specify the binding site using an irregular box (see User Guide Sect. 5.4 for details on specifying the `borders.xyz` file that is read by this command):

```
temp_gen -b <target> <template> <Hbonding_point_density> <grid_spacing>
<clustering_threshold>
```

*Representing protein-bound metal interactions in unbiased template generation.* (User Guide Sect. 3.6.2, 5.5) Binding-site metal interactions require special processing, due to the short, almost covalent bonds they form with ligands potentially being interpreted as van der Waals collisions during docking. When calculating van der Waals collisions between a metal atom and a ligand candidate, SLIDE automatically reduces the radius of the metal atom to avert this problem. The minimum distance for favorable interactions was determined for each metal type from the metal-ligand distance distribution retrieved from the Metalloprotein Database (<http://metallo.scripps.edu>), `Temp_gen` creates acceptor template points evenly distributed on the surface of a sphere around the metal at optimal interaction distance for that type of metal according to the same database. SLIDE recognizes Ca, Co, Cu, Fe, K, Mg, Mn, Na, and Zn. Metal template points closer than 2.5 Å to any protein or cofactor atom are automatically deleted.

*Choosing unbiased template parameters to generate a template of appropriate size for efficient screening.* A typical good template has fewer than 100 points, with about 1/3 of the points being hydrophobic and 2/3 H-bonding, in the case of a binding site that is equally hydrophobic and polar. Finer template point sampling (dense rather than sparse, or sparse rather than minimal) is suggested for hydrogen-bonding points because the tolerance on matching is small, to ensure good H-bond geometry for docked ligands.

If the template includes more than 150 points, then prune the template to a smaller size. First, remove any peripheral template points by using the script `prune_template.pl` with a threshold less than the 3.0 Å default used by `temp_gen`. The template prior to automated pruning by SLIDE includes "prepruned" in the name, and the pruned template is simply named `template`. Individual template points also can be pruned by text-editing `$SLIDE_DATA_DIR/<target>/<template>/in/template` (not the `template.pdb` file) to remove points that appear peripheral to the site, when the template's PDB file is visualized in molecular graphics. If there are still too many points, you can generate a new template using a slightly larger hydrophobic point clustering threshold in `temp_gen`. A finer value for `<grid_spacing>`, which is used for initial sampling of hydrophobic sites, may require more time and/or

system memory to run. A typical value is between 0.5 and 0.7 Å, and `<clustering_threshold>` (which regulates the final spacing of hydrophobic template points in unbiased template generation), is typically set from 3.0-4.0 Å. Smaller values yield more template points. There is interplay between the clustering threshold and the triangle side length parameters used for docking by SLIDE, so changing these parameters significantly is not recommended. To prune a template further, set the hydrogen-bonding point sampling to `sparse` or `minimal`. The docked ligand can extend beyond the template region, and interactions of those atoms will still be scored.

*If a known ligand does not dock well when you are validating SLIDE with your target protein, it may help to instead create a more finely sampled template. For example, set the `<Hbonding_point_density>` to `dense`, reduce the `<grid_spacing>` value to 0.3 and reduce the `<clustering_threshold>` to 2.5 (measured in Å). Alternatively, you can assess whether OrientScore (used by SLIDE to detect the best orientation of each ligand) scores a suboptimal docking for your ligand more favorably than it scores the crystallographic orientation by running `slide_score` on the crystallographic orientation (see section 7 on Standalone scoring) and comparing the OrientScore values in the docked and crystallographic ligand mol2 files. If this is a problem and you would like to test different scoring methods for their ability to detect better SLIDE dockings of known ligands, you can set the `OUTPUT_ALL_MATCHES` option in the `param.h` file (Section 6), and recompile SLIDE. It will then output all sterically acceptable orientations of each molecule. Then, compare the AffiScore rather than OrientScore values of these docked orientations (written in the docked mol2 files), or run external scoring software on the docked mol2 and target pdb files, to assess whether an alternative scoring approach improves detection of accurate dockings of the known ligand. We found that Szybki rigid-body energy minimization of the docked protein-ligand complex (using the default settings of OpenEye's Szybki software, taking ~1 second per complex) typically improves SLIDE dockings by 0.4Å when the initial docking is within 2Å RMSD of the correct orientation.*

**Creating a merged (biased + unbiased) template.** A biased template represents known ligand interactions, and an unbiased template represents additional opportunities for making good interactions in the binding site. Thus, we often merge the two templates into one for screening databases. First, set up the directory to hold the new merged template:

```
slide_setup.pl <target> <template> <database>
```

where the `<target>` and `<database>` tags are the names of preexisting subdirectories you created with `setup_dbase`. The nametag `<template>` can be set to `merged` or another name of your choosing. Then, concatenate the unbiased and biased templates you have already created into a new `template` file in the `merged/in` directory, by typing the following Unix `cat` command (all on a single line):

```
cat $SLIDE_DATA_DIR/<target>/unbiased/in/template
$SLIDE_DATA_DIR/<target>/biased/in/template >
$SLIDE_DATA_DIR/<target>/merged/in/template
```

**Using key points to filter dockings to match a region of the binding site.** SLIDE dockings are filtered such that only those dockings matching at least one of the key points in the template (marked with `*` in the file `$SLIDE_DATA_DIR/<target>/<template>/in/template`) will be retained. By default, all the hydrogen-bonding points are set as key points, ensuring that at least one good polar interaction is made between the protein and a docked ligand candidate. However, the user can edit the template file to include or remove key point `*`'s, whenever docking into a certain region of the binding site is desired (User Guide Fig.3.5). If the `template.pdb` file is read into a molecular graphics program along with the protein target, the residue numbers for key points (which appear as water molecules) can be identified by clicking or

labeling those points. The x,y,z coordinates of the chosen points can then be cross-referenced between the `template.pdb` and `template` file. At that point, the selection of key points can be changed by altering the \* designations of those lines in the `template` file. At least one key point must be set. If all points are set as key points, then all possible dockings will be considered. Setting fewer key points accelerates SLIDE runs by filtering out more dockings, but also can strongly bias the docking to a certain region of the protein. A parameter in the `params.h` file (see the section, “Recommended values for parameters”, below) allows you to require each docking to match at least two key points in a set of key points, instead of one. This will constrain the dockings more strongly to the region of interest, and result in fewer dockings overall.

**Filtering dockings to match two regions in the binding site.** When the goal is to identify ligands that interact with two separate regions of the binding site, you can use the SLIDE v. 3.4 feature that filters dockings to identify those match two sets of key points. See the details in section 8 of this guide.

## 4 Running SLIDE

(User Guide Sect. 3.4, 5.1, 5.9.1)

```
run_slide <target> <template> <database> <sphere_radius> <overwrite existing
files? y/n/r> [<output_file>]
```

```
Ex: run_slide 7dfr unbiased csd_ligands 9.0 y 7dfr_unbiased_CSD_results.txt
```

Only those residues of the target protein with at least one atom within the distance `<sphere_radius>`, typically 9-10 Å, of any other template point are considered during docking. If a previous run has been performed for this target, template, and database combination, you can specify whether or not you want to overwrite the previous output files, or to restart from a certain molecule. If you choose not to overwrite the files, then SLIDE will produce an error message and terminate. You will then be required to set up a new template and/or database directory for the new run, or to move the old data to a different location (if you wish to preserve it).

The three `<overwrite existing files>` options have the following behavior:

**y:** deletes the contents of the `<database_ligands>` and `<database_targets>` directories, and restarts the run

**n:** starts a SLIDE run unless there is output already existing for that run, in which case the current run will abort

**r:** will not delete molecules in the `<database_ligands>` or `<database_targets>` directories. Will continue writing files starting at the `RESTART_MOLECULE` specified in the `slide.parameters` file. This is useful for restarting a previously terminated run, but can be dangerous if used without a `RESTART_MOLECULE`, or if the `RESTART_MOLECULE` is set incorrectly, as this may mix old and new data. (Note: the `RESTART_MOLECULE` should be entered without the conformer tag at the end in order to work properly, if `GROUP_CONFORMERS` is set to true)

SLIDE parameters can be fine tuned by creating a `slide.parameters` file in `$$SLIDE_DATA_DIR/<target>/<template>/in/` (see Screening parameters in section 6, below). If there is no `slide.parameters` file in `$$SLIDE_DATA_DIR/<target>/<template>/in/`, the default `slide.parameters` file in `$$SLIDE_DIR/params/` is used.

Once all necessary files are located, SLIDE is run. The results are summarized in `<output_file>` in `$$SLIDE_DATA_DIR/<target>/<template>/in/`. It is highly recommended to specify an `<output_file>`, so that all the information needed to reproduce the run is preserved. Otherwise the information is directed to `stdout` and will not be captured. All other SLIDE output is written in the subdirectories of `$$SLIDE_DATA_DIR/<target>/<template>`, as described in the next section.

## 5 Interpreting the scoring output

(User Guide Sect. 3.7)

The scoring of SLIDE has changed between version 2.3 and 3.0 and significantly improves the detection of the best orientation of a ligand candidate on average, and estimation of its relative affinity. Note that more favorable scores are now *more negative (not positive)*, and they estimate the binding affinity in kcal/mol. If not redirected into a file, the output of the SLIDE run appears on the screen showing the following information for each docked ligand orientation:

- 1) [Ligand Name]\_[Conformer #]\_[Orientation #]
- 2) Orientation Score
- 3) (Affinity Score - Affinity Constant) / # Heavy Ligand Atoms
- 4) Affinity Score (weighted sum of terms 5, 6, 7, plus constant of -5.218)
- 5) Hydrophobic Complementarity Term Between Ligand & Protein
- 6) Polar Term (weighted sum of polar terms 9, 10 & 11 below)
- 7) Unsat Polar Term (weighted sum of unsatisfied polar terms 12 & 13 below)
- 8) Number of Protein-Ligand Hydrophobic Contacts
- 9) Number of Protein-Ligand H-bonds
- 10) Number of Protein-Ligand Salt-bridges
- 11) Number of Metal-Ligand Bonds
- 12) Number of Interfacial Unsatisfied Polar Atoms
- 13) Number of Interfacial Unsatisfied Charged Atoms
- 14) Percent of Ligand Carbons Buried Against the Protein (see `params.h`, below)
- 15) Remaining van der Waals Collisions (#)
- 16) Sum of remaining van der Waals Overlaps (Å)

By default, the top scoring ligand orientation is saved. By setting the `GROUP_CONFORMERS` parameter to `FALSE` in the `slide.parameters` file, it is possible to output the best scoring orientation for each conformer of each ligand. Additionally, all dockings, regardless of score, can be saved by defining `OUTPUT_ALL_MATCHES` in the `params.h` file located in `$$SLIDE_DIR/src/slide/inc/`. Results are written into the following files:

```
Docked ligands (.mol2):          $$SLIDE_DATA_DIR/<target>/<template>/<database>_ligands/
Rotated protein side chains (.pdb): $$SLIDE_DATA_DIR/<target>/<template>/<database>_targets/
Error and log files:             $$SLIDE_DATA_DIR/<target>/<template>/log/
Output file (if specified):      $$SLIDE_DATA_DIR/<target>/<template>/in/
```

## 6 Recommended values for parameters in SLIDE

Template parameters [Default values in brackets]

(User Guide Sect. 5.4)

Grid spacing: 0.5-0.7 Å. [0.5]

Radius for defining binding site in unbiased template generation: 9-10 Å. [9.0] This allows protein atoms 9-10 Å from any template point to contribute to assessment of van der Waals overlaps and score evaluation.

Clustering threshold for unbiased, knowledge-based templates: 3.0-4.0 Å. [3.5] (Note that the average distance between the cluster centroids written as hydrophobic template points is somewhat less than the clustering threshold.)

Clustering threshold for a ligand-based template: Specify 0 if a single known ligand is used as input, or 0.5-2.0 Å [2.0] if multiple ligands are used.

H-template type: dense, sparse, or minimal [**dense**]

## Screening parameters

(User Guide Sect. 3.5)

### *The slide.parameters file*

The default screening parameters can be found in `$$SLIDE_DATA_DIR/params/slide.parameters`. Larger values of the parameters (except for `max_template_triangles`) will result in more docked ligand orientations. To override the defaults, make a copy of the parameter file with the new values (still named `slide.parameters`) in your `$$SLIDE_DATA_DIR/<target>/<template>/in/` directory. Starting with the defaults is recommended, followed by adjusting as necessary to dock known ligands (minimize false negatives), then adjusting as needed to minimize the number of false positives when screening. Note that there are some changes in the format of this file in SLIDE version 3.0, which also contains additional parameters (`MATCH_2_KEY_POINTS`, `GROUP_CONFORMERS`, and `RESTART_MOLECULE`). Descriptions of the contents of `slide.parameters`, and general recommendations on setting parameter values, follow.

	<u>Default values</u>	<u>Recommended range</u>
DME_THRESHOLD (Å)	0.3	0.2 - 0.3
RMS_THRESHOLD (Å)	0.3	0.2 - 0.3
ANCHOR_TRANSLATION (Å)	0.3	0.2 - 0.3
ANCHOR_OVERLAP (Å)	0.3	0.1 - 0.3
SIDE_CHAIN_OVERLAP (Å)	0.3	0.2 - 0.5
INTRA_OVERLAP (Å)	0.1	0.1
INTERMEDIATELY_TOLERATED_OVERLAP (Å)	2.0	0.5 - 2.0
FINALLY_TOLERATED_MAX_BUMP (Å)	0.5	0.1 - 0.5
FINALLY_TOLERATED_OVERLAP (Å)	2.0	0.5 - 2.0
SCORE_CUTOFF (kcal/mol)	0.0	(check scores of known ligands)
MAX_TEMPLATE_TRIANGLES (#)	1500000	1500000
MATCH_2_KEY_POINTS (True/False)	False	False
GROUP_CONFORMERS (True/False)	True	True
RESTART_MOLECULE (n/a)	None	Molecule name

Alter the parameters relating to DME, RMS, and ANCHOR when there are problems resolving collisions of the anchor fragment (defined by the current interaction point triangle in the ligand) for known ligands, or decrease the clustering parameters for the template, as described at the end of the “Creating an unbiased, protein-based template” subsection, under section 3 of this document.

DME\_THRESHOLD is the maximum distance matrix error in Ångstroms, a lower bound for the RMS deviation following least-squares fitting of the anchor fragment to the template.

RMS\_THRESHOLD is the maximum allowed RMSD (in Å) for the fitting of the anchor fragment to the protein template.

`ANCHOR_TRANSLATION` governs the maximum amount (in Å) the anchor fragment can move (in any direction) during the collision resolution procedure.

`ANCHOR_OVERLAP` governs the amount (in Å) the anchor fragment atoms can overlap with main-chain atoms in the protein.

Change `SIDE_CHAIN_OVERLAP` when there are problems with resolving side-chain collisions (test on known ligands). `SIDE_CHAIN_OVERLAP` governs the maximum overlap (in Å) between any pair of ligand and protein side chains (where ligand side-chain is defined as any group beyond the anchor fragment).

`INTRA_OVERLAP` governs the maximum allowed van der Waals overlap (in Å) within a molecule.

`INTERMEDIATELY_TOLERATED_OVERLAP` is the sum of all intra and intermolecular overlaps (in Å) at any point during docking.

`FINALLY_TOLERATED_OVERLAP` is the maximum sum of all intra and intermolecular overlaps (in Å) at the end of docking.

`SCORE_CUTOFF`: starting with version 3.0 of SLIDE, scores that are more negative reflect greater complementarity between molecules. It is only important to set `SCORE_CUTOFF` to a value other than the default of 0 if you define `OUTPUT_ALL_MATCHES` in the `params.h` file, below. We suggest that you calibrate the choice of a new score cutoff by using the scores of known ligands docked by SLIDE.

`MAX_TEMPLATE_TRIANGLES` is the maximum number of template triangles allowed and determines the memory usage of SLIDE, at ~36 bytes per triangle.

`FINALLY_TOLERATED_MAX_BUMP`: the maximum (in Å) for any single overlap, intra or intermolecular, at the end of docking.

`MATCH_2_KEY_POINTS?`: Tells SLIDE to force triangle matches to include at least 2 key template points, which ensures that at least two interaction points in the ligand match the key point region of the template in spacing and chemistry (otherwise, only one key point needs to be matched). Set to `TRUE` or `FALSE`.

`GROUP_CONFORMERS?`: To reduce output, this setting will select the best output binding pose from the entire set of conformers. Set this option to `TRUE` or `FALSE`. In order to use this feature the conformers must be in a `multimol2` file with the naming convention `<MOLECULE NAME>_<CONFORMER ID>`. If this option is set to true and molecules do not follow this naming convention, then different molecules may be grouped together for conformer selection, resulting in undesirable behavior.

`RESTART_MOLECULE?`: If a SLIDE run is being restarted after a previously terminated run, SLIDE will resume the new run starting at this molecule. The molecule name should not include the conformer # if conformers are being grouped.

### *Parameters from the `params.h` file*

(User Guide Sect. 3.5.4)

The `params.h` file is new to version 3.0 and allows you to modify some compile-time parameters, while other compile-time parameters that should only be changed under rare circumstances remain in the `defs.h` file. You have the option to set the parameter `OUTPUT_ALL_MATCHES` to either `#define` if you wish to output every sterically acceptable docked orientation regardless of score (which is useful if you wish to rescore and rank the protein-ligand interactions with an alternate scoring function), or to `#undef` (default) if you wish to output only the top-scoring orientation.. It is highly recommended if outputting all matches that a moderate score cutoff be used in the `slide.parameters` file so that most of the very unreasonable dockings get filtered out, maintaining a manageable number of output dockings.

Another option, `FILTER_BURIED_CARBOONS`, is set to `#define` by default, and will only output orientations in which at least 50% of the ligand carbon atoms are buried in the protein interface, as is typical for the vast majority of small-molecule ligands. (This was a fixed criterion in version 2.3 of SLIDE.) `FILTER_BURIED_CARBOONS` can now be set to `#undef` for unusual ligands with a large percentage of solvent exposed surface (e.g., cyclosporin binding to cyclophilin A).

Depending on the size of the binding site and ligand candidates, it might be necessary to adjust the parameters controlling the size of the template triangles matched by the ligand interaction point triplets. However, generally we do not recommend changing these parameters, since they work well across a broad range of protein:ligand complexes and are compatible with the default template spacing. These parameters also avoid dockings that match tiny anchor fragments in the ligand to tiny regions in the protein template, since shape and chemical complementarity on a larger scale is preferable. To handle small ligands (with maximum length of 10 Å), SLIDE will automatically switch the default minimum accepted longest triangle side from 5 Å to 3 Å (set as `SMALL_TRIANGLE_MIN_LONGEST_SIDE`) to match important interaction centers such as hydrogen-bonding groups that are relatively close together.

The default parameter set in the `params.h` file is:

```
#undef OUTPUT_ALL_MATCHES
#define FILTER_BURIED_CARBOONS
#define TRIANGLE_MIN_PERIMETER (Å)           7.0
#define TRIANGLE_MAX_PERIMETER (Å)         30.0
#define TRIANGLE_MAX_LONGEST_SIDE (Å)      12.0
#define TRIANGLE_MIN_SHORTEST_SIDE (Å)     2.0
#define TRIANGLE_MAX_SHORTEST_SIDE (Å)    10.0
#define LARGE_TRIANGLE_MIN_LONGEST_SIDE (Å) 5.0

/* For small ligands, TRIANGLE_MIN_LONGEST_SIDE will be set from the parameter
below, overriding the TRIANGLE_MIN_LONGEST_SIDE value used above for larger
ligands */
#define SMALL_LIGAND_DIAMETER (Å)          10.0
#define SMALL_TRIANGLE_MIN_LONGEST_SIDE (Å) 3.0
```

Note: To change these parameters, edit the `/SLIDE_DIR/src/slide/inc/params.h` file, and recompile the program:

```
cd /SLIDE_DIR/src/slide
make clean
make
```

## 7 Standalone SLIDE scoring and tabulating the results of SLIDE screening

### Scoring a protein:ligand complex using the SLIDE scoring function

(User Guide 2.6, 5.2)

- Have the `.pdb` file of the target and the `.mol2` file of the ligand in your current directory (or include the directory path to the files), then enter:

```
slide_score <target>.pdb <ligand>.mol2
```

- The score will appear on the screen (stdout) showing the same output for an orientation as would be produced by SLIDE in most cases, though without the van der Waals overlap terms (User Guide 3.7).

When this standalone version of the SLIDE scoring function is used, it will not perform side-chain rotations to try to resolve van der Waals overlaps, nor measure van der Waals overlaps. If there are significant van der Waals overlaps in the ligand docking being scored, the resulting score is likely to over-assess hydrophobic and H-bond interactions. Thus we recommend only using the standalone `slide_score` program on SLIDE generated dockings, or other van der Waals overlap-free orientations. To re-score a docking, it is recommended that you generate a pdb file containing the rotated side-chains, found in the directory `SLIDE_DATA_DIR/<target>/<template>/<database>_targets`.

To merge the rotated side-chains into the original `<target>.pdb` file, use the perl script `merge_pdb_target.pl` as follows:

```
merge_pdb_target.pl <rotated side chains>.pdb <target>.pdb > <output>.pdb
```

Then, use this new pdb file as the `<target>.pdb` input for `slide_score`. Alternatively, `slide_score` can automatically merge the files by including the `--side_chains` flag:

```
slide_score -p <target>.pdb -l <ligand>.mol2 --side_chains
<rotated_sidechain>.pdb
```

If a large number of ligands are to be re-scored with the same target PDB file, then first generate a file listing the ligand files, including their directory paths, and generate a second file listing the rotated side chain files, including directory paths. For example, the `<rotated_sidechains_file>` specified in the `slide_score` command below will contain a list of file names like:

```
~/slide_data/lvr1/unbiased/mixed_sample_targets/ltbz_ligand_0000.pdb
~/slide_data/lvr1/unbiased/mixed_sample_targets/ZINC00000185_0000.pdb
~/slide_data/lvr1/unbiased/mixed_sample_targets/ZINC00092235_0000.pdb
```

There should be a one-to-one correspondence between the `<ligand_list_file>` entries and the `<rotated_side_chains_file>` entries, both in order and number of entries. If no side-chain rotations are involved for one of the ligands, an empty rotated side chain file should be included for it to maintain the correct correspondence between ligand coordinates and rotated side-chain coordinates. Then, issue the `slide_score` command:

```
slide_score -p <target>.pdb --lig_list <ligand_list_file> --sc_list
<rotated_side_chains_file>
```

## Generating an interaction matchprint summary for docked molecules

SLIDE v3.4 includes a new feature called interaction matchprints for analyzing protein-ligand interactions. It can be helpful in comparing patterns of interactions of different ligand molecules and relating them to known ligands and their relative affinities. The output is in comma separated value (csv) format, which can be read by any spreadsheet program. Sample output (formatted in Excel) looks like:

Interaction matchprints for 42 known thrombin inhibitors, labeled by PDB code, followed by the five new inhibitors from SLIDE screening, labeled by ZINC code. Interactions are highlighted in grey, showing the number of ligand interactions with the corresponding thrombin atom (for polar interactions) or side chain (for hydrophobic interactions).

LIGAND	PHE 40 sc	LEU 41 sc	CYS 42 sc	HIS 57 NE2	HIS 57 sc	CYS 58 sc	TYR 60A sc	TRP 60D sc	LYS 60F NZ	LYS 60F sc	LEU 99 sc	GLU 146 sc	ILE 174 sc	ASP 189 OD1	ASP 189 OD2	ALA 190 O	ALA 190 sc	CYS 191 O	CYS 191 sc	GLU 192 OE2	GLU 192 sc	GLY 193 N	SER 195 N	VAL 213 sc	SER 214 O	TRP 215 sc	GLY 216 N	GLY 216 O	GLU 217 OE2	GLU 217 sc	GLY 219 N	GLY 219 O	CYS 220 sc	
1a3b	0	0	0	1	1	0	3	3	0	0	4	0	1	1	1	1	1	0	0	0	0	1	1	1	1	5	1	1	0	0	0	1	0	
1a46	0	0	1	1	0	0	2	3	0	0	1	0	1	0	0	1	1	0	0	0	0	1	1	0	0	4	1	1	0	0	1	0	1	1
1a4w	0	0	0	0	0	0	2	5	1	0	4	0	2	1	0	1	1	0	0	0	0	0	0	0	3	1	1	0	0	1	0	0	1	1
1a5g	1	0	0	1	1	0	3	3	0	0	4	1	1	2	2	0	1	0	0	0	2	1	1	1	0	3	1	1	0	0	0	1	0	0

To generate matchprint output, include the flag `--interactions_table` when running `slide_score`:

```
slide_score -p <target>.pdb --lig_list <ligand_list_file> --sc_list
<rotated_side_chains_file> --interactions_table
```

See the notes in the preceding section on the contents of the `<ligand_list_file>` and `<rotated_side_chains_file>`. It can be helpful if interactions in crystallographic complexes scored by `slide_score` are included at the top of the `<ligand_list_file>` and `<rotated_side_chains_file>`, while docked ligand candidates are included below. At present, a glitch requires that at least two sets of ligand and rotated side-chain coordinates be included in the `<ligand_list_file>` and `<rotated_side_chains_file>` when the `--interactions_table` option is used. A workaround is to include the same file names twice.

## Organizing and tabulating the results of SLIDE

(User Guide 3.8, 3.9)

To create a table summarizing the docking results after a run has been completed, use the command:

```
results_table.pl <target> <template> <database> <number> [-conf] >
<output>.results_table
```

The `-conf` option should be used when multiple conformers of each ligand candidate are used as input to SLIDE, but aren't grouped using the `GROUP_CONFORMERS` option. In this case, the docked ligand name is expected to have the format `<ligand_name>_<conformer_no>_<orientation_no>.mol2`. When `-conf` is not specified, the program expects the docked ligand names to have the format `<ligand_name>_<orientation_no>.mol2`. The output table goes to `stdout` unless it is redirected into a file. It will contain the `slide.parameters` and `param.h` entries used for the run and the list of the top `<number>` of ligands sorted by their scores.

A version of the above script can be used to delete all dockings not included into the table (whose scores are lower than the score of the last entry in the table):

```
% results_table_del.pl <target> <template> <database> <number> [-conf]
```

For additional options on restricting the number of docking generated and saved during a screening run, see Section 3.9 of the User Guide, the key points information in the "Generating a protein template" section above, and the ability to set a score threshold in the `slide.parameters` file (section 6).

An alternative version of the `results_table` script uses the (AffiScore)/ (# of heavy ligand atoms), called the Ligand Efficiency score. This helps identify ligands in which most atoms contribute favorably to the affinity score, with few poorly interacting atoms. It is used the same way as `results_table.pl`:

```
results_table_ligefficient.pl <target> <template> <database> <number> [-conf]
```

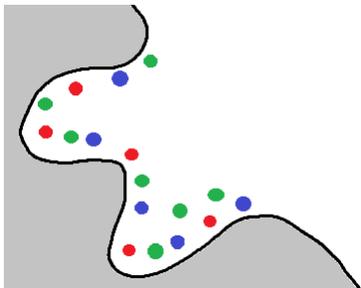
## 8 Working with two regions of key points in SLIDE template generation

SLIDE (v3.4) now supports the definition of two separate sets of key points for interacting with the protein. This is useful when you want to filter dockings to keep those that can interact with two sites in the binding pocket. (The molecular weight range of the compounds you screen will need to be chosen appropriately, so there will be molecules large enough to span the two sites.)

- 0) The two sets of key points option is chosen by modifying the `slide.parameters` file. Copy `SSLIDE_DATA_DIR/params/slide.parameters` to `SSLIDE_DATA_DIR/<target>/<template>/in/` (the directory for your current project), keeping the same `slide.parameters` file name. Then, edit this file to set the value for `MATCH_2_KEY_POINTS` to `TRUE`. A docked molecule must now match at

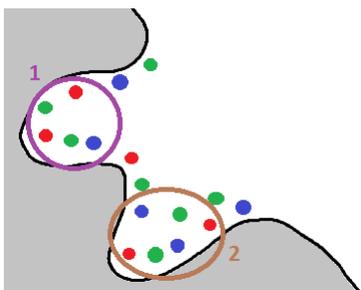
least one point in each set of key points to be kept. The following protocol outlines how to take advantage of this new feature in SLIDE by defining the two key point sets in your `template` file.

- 1) Generate a binding site template using SLIDE's `temp_gen` command. View it in molecular graphics, as described early in section 3 (Generating a protein template).



Grey: protein; Green: hydrophobic template points; Red: hydrogen bond acceptor template points; Blue: hydrogen bond donor template points

- 2) Identify up to two regions (shown by purple and brown circles below) in which screened molecules should bind, and note the residue numbers of template points for each region (this can be accomplished by identifying a residue of interest and identifying all the template points that are within a cutoff radius). At least one template point per desired region should be selected.



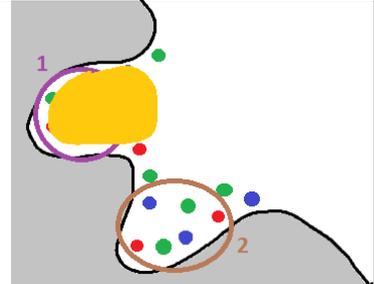
- 3) In the template file (`$SLIDE_DATA_DIR/<target>/<template>/in/template`), by default, all hydrogen bond template points will comprise the first key point region, as indicated by the `*`'s in these lines of the template file. Modify each line so that only those template points you have chosen to be in region 1 are marked with `*`. Then, mark template points in region 2 with a `^` (caret) symbol. Each set of key points can be any combination of hydrogen bond and hydrophobic template points, though in many cases enforcing specific hydrogen bond interactions proves to be more discriminating. Note that a single point can be considered part of both key point regions; however SLIDE will require at least one of each key point set to be matched (if 2 key point matching is set in the preferences). The file will look something like this when complete:

<u>Line in file:</u>	<u>Comment (not included in file):</u>
A -9.754 7.298 4.606	Acceptor point
A^ -5.269 11.302 3.203	Acceptor point in region 2
A* -5.388 5.326 10.570	Acceptor point in region 1
A*^ -6.275 5.818 11.056	Acceptor point in either region 1 or 2
A -6.117 4.267 10.750	Acceptor point

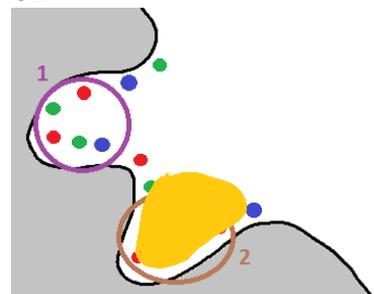
A	-5.629	3.751	5.495	Acceptor point
D	-4.352	1.901	8.606	Donor point
D <sup>^</sup>	-6.760	11.640	2.808	Donor point in region 2
D	-7.428	10.740	2.458	Donor point
D*	-3.565	7.591	10.957	Donor point in region 1
D* <sup>^</sup>	-4.647	11.751	2.153	Donor point in either region 1 or 2
N	-5.793	1.762	5.155	Doneptor point
N*	-4.470	6.438	10.827	Doneptor point in region 1
N	-4.842	2.498	3.432	Doneptor point
N <sup>^</sup>	-4.374	10.032	6.320	Doneptor point in region 2
N*	-2.126	9.726	11.126	Doneptor point in region 1
H	-7.590	7.530	3.700	Hydrophobic point
H*	-6.602	6.100	0.568	Hydrophobic point in region 1
H <sup>^</sup>	-6.848	8.410	2.497	Hydrophobic point in region 2
H <sup>^</sup>	-5.464	8.919	3.468	Hydrophobic point in region 2
H	-5.024	2.132	10.446	Hydrophobic point

### Following are examples of ways to use two sets of key points to filter dockings:

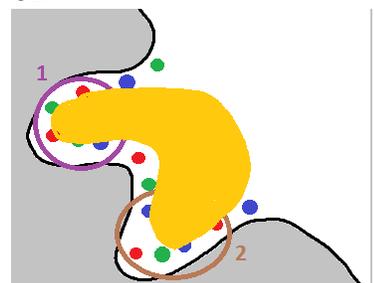
A) To generate all possible ligand dockings:



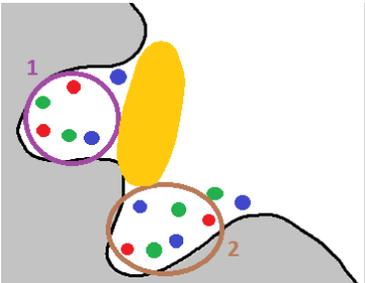
Or



Or



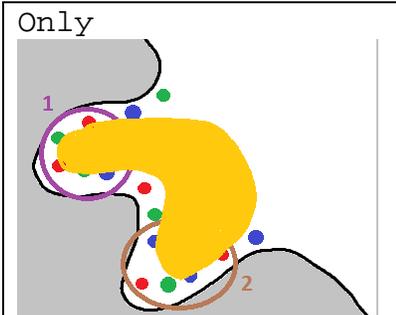
Or



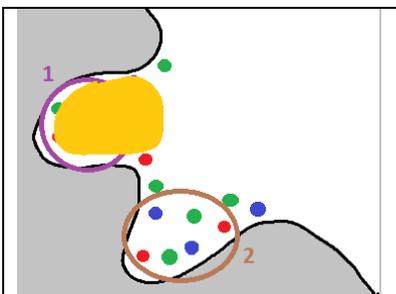
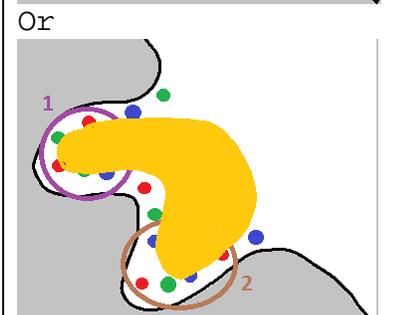
- Set ALL template points as key points (marked with either \*, ^, or \*^)
- Set MATCH\_2\_KEY\_POINTS to False

In this case, a docked molecule (shown as the yellow shape in the schematics at left) matching at least 3 template points and at least one key point in any region of the protein will be retained by SLIDE.

B) To save only those dockings matching a point in BOTH key point regions:

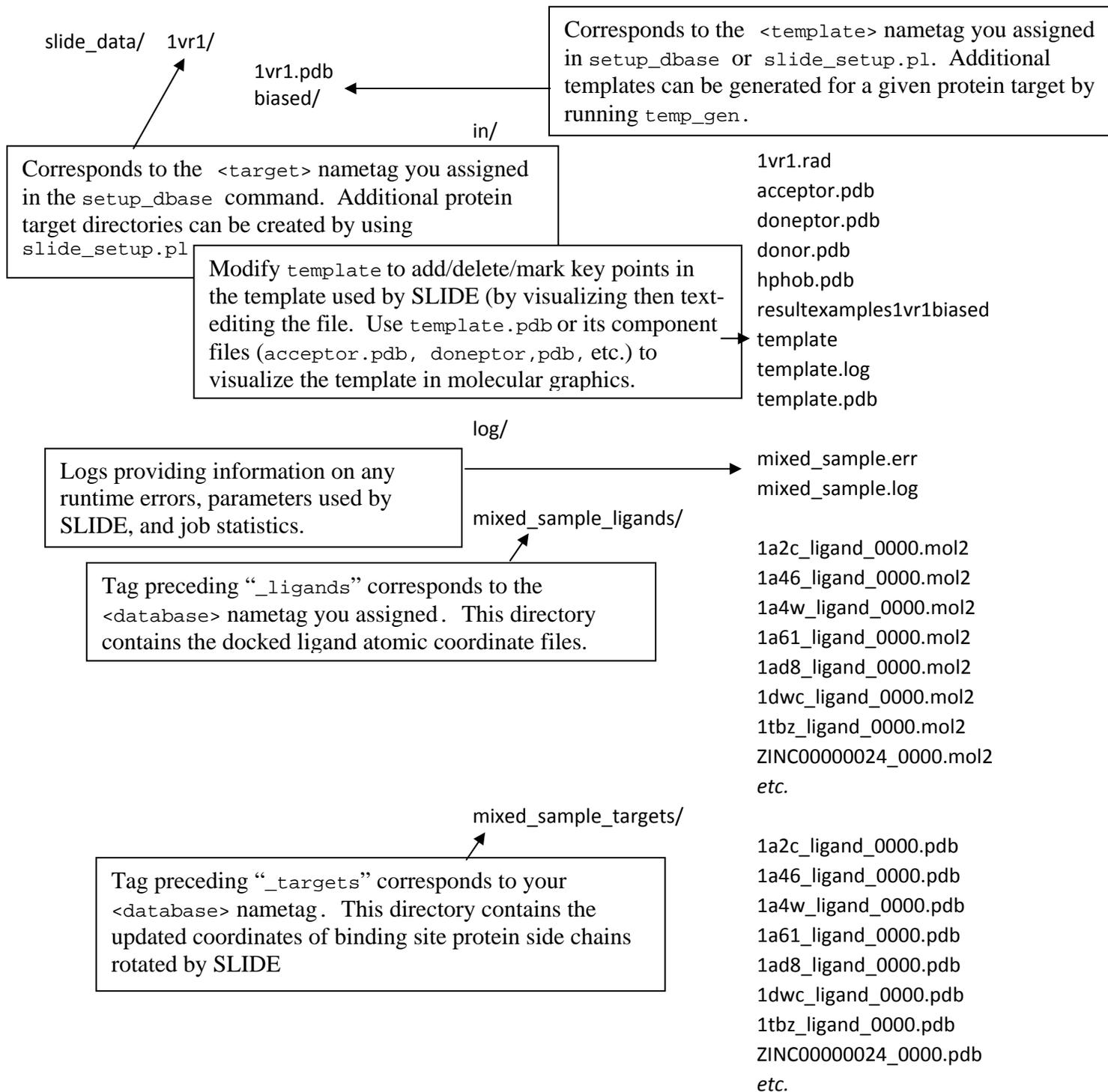
<p>Only</p> 	<ul style="list-style-type: none"> <li>• Mark template points in region 1 with a *</li> <li>• Mark template points in region 2 with a ^</li> <li>• Set MATCH_2_KEY_POINTS to True</li> </ul> <p>Resulting dockings must match a key point in both regions 1 and 2 (as shown for the yellow molecule) to be retained by SLIDE.</p>
---	---

C) To keep molecules only those molecules matching region 1:

	<p>To weakly bias matches to region 1:</p> <ul style="list-style-type: none"> <li>• Mark template points in region 1 with *, ^, or *^</li> <li>• Set MATCH_2_KEY_POINTS to False</li> </ul> <p>To strongly bias matches, requiring at least 2 template points in region 1 to be matched:</p> <ul style="list-style-type: none"> <li>• Mark template points in region 1 with *^</li> <li>• Set MATCH_2_KEY_POINTS to True</li> </ul>
<p>Or</p> 	<p>Note that dockings that also match region 2 may still occur, as shown at left.</p>

## 9 Structure of the examples directory (a typical \$SLIDE\_DATA\_DIR directory)

**Note:** Once a `slide_data` file structure has been set up, do not move or rename files within the directories. If you do, future database, template, and SLIDE runs may not be able to locate these files...



merged/		
	in/	1vr1.rad result_examples_1vr1_merged template template~
	log/	mixed_sample.err mixed_sample.log
	mixed_sample_ligands/	1a46_ligand_0000.mol2 1a4w_ligand_0000.mol2 1a61_ligand_0000.mol2 1dwc_ligand_0000.mol2 1tbz_ligand_0000.mol2 ZINC00000185_0000.mol2 <i>etc.</i>
	mixed_sample_targets/	1a46_ligand_0000.pdb 1a4w_ligand_0000.pdb 1a61_ligand_0000.pdb 1dwc_ligand_0000.pdb 1tbz_ligand_0000.pdb ZINC00000185_0000.pdb <i>etc.</i>
unbiased/		
	in/	1vr1.rad acceptor.pdb acceptor_preprune.pdb acceptor.usr borders.xyz doneptor.pdb doneptor_preprune.pdb doneptor.usr donor.pdb donor_preprune.pdb donor.usr hphob.pdb hphob_preprune.pdb hphob.usr points.usr

```

result_examples_1vr1_unbiased
result_examples_1vr1_unbiased_confs
template
template~
template.log
template.pdb
template_preprune
template_preprune.pdb

log/

mixed_sample_confs.err
mixed_sample_confs.log
mixed_sample.err
mixed_sample.log

mixed_sample_confs_ligands/
1a2c_ligand_0000.mol2
1a46_ligand_0000.mol2
1a4w_ligand_0000.mol2
1a61_ligand_0000.mol2
1dwc_ligand_0000.mol2
1tbz_ligand_0000.mol2
ZINC00000024_0000.mol2
etc.

mixed_sample_confs_targets/
1a2c_ligand_0000.pdb
1a46_ligand_0000.pdb
1a4w_ligand_0000.pdb
1a61_ligand_0000.pdb
1dwc_ligand_0000.pdb
1tbz_ligand_0000.pdb
ZINC00000024_0000.pdb
etc.

mixed_sample_ligands/
1a46_ligand_0000.mol2
1a4w_ligand_0000.mol2
1a61_ligand_0000.mol2
1dwc_ligand_0000.mol2
1tbz_ligand_0000.mol2
ZINC00000185_0000.mol2
etc.

mixed_sample_confs_ligands/
1a2c_ligand_0000.mol2

```

1a46\_ligand\_0000.mol2  
 1a4w\_ligand\_0000.mol2  
 1a61\_ligand\_0000.mol2  
 1dwc\_ligand\_0000.mol2  
 1tbz\_ligand\_0000.mol2  
 ZINC00000024\_0000.mol2  
*etc.*

mixed\_sample\_confs\_targets/

1a2c\_ligand\_0000.pdb  
 1a46\_ligand\_0000.pdb  
 1a4w\_ligand\_0000.pdb  
 1a61\_ligand\_0000.pdb  
 1dwc\_ligand\_0000.pdb  
 1tbz\_ligand\_0000.pdb  
 ZINC00000024\_0000.pdb  
*etc.*

mixed\_sample\_ligands/

1a46\_ligand\_0000.mol2  
 1a4w\_ligand\_0000.mol2  
 1a61\_ligand\_0000.mol2  
 1dwc\_ligand\_0000.mol2  
 1tbz\_ligand\_0000.mol2  
 ZINC00000185\_0000.mol2  
*etc.*

mixed\_sample\_targets/

1a46\_ligand\_0000.pdb  
 1a4w\_ligand\_0000.pdb  
 1a61\_ligand\_0000.pdb  
 1dwc\_ligand\_0000.pdb  
 1tbz\_ligand\_0000.pdb  
 ZINC00000185\_0000.pdb  
*etc.*

1vr1.pdb

databases/

pts files contain chemical interaction points for each of the ligand candidates in the corresponding mol2 file (see mixed\_sample.db)

mixed\_sample\_confs/

Contains location information for ligand files in the database.

mixed\_sample.db  
 number\_of\_mol2\_entries  
 singleton\_1.pts  
 ZINC\_ligands\_0.pts

confs\_1a2c\_ligand\_0.pts  
 confs\_1a46\_ligand\_1.pts  
 confs\_1ad8\_ligand\_4.pts

confs\_1dwc\_ligand\_5.pts  
confs\_1tbz\_ligand\_6.pts  
confs\_ZINC00000024\_7.pts  
*etc.*  
mixed\_sample\_confs.db  
number\_of\_mol2\_entries

## ligands\_to\_be\_docked/

1a2c\_ligand.mol2  
1a46\_ligand.mol2  
1a4w\_ligand.mol2  
1a61\_ligand.mol2  
1ad8\_ligand.mol2  
1dwc\_ligand.mol2  
1tbz\_ligand.mol2  
ZINC00000024.mol2  
*etc.*

## ligands\_to\_be\_docked\_confs/

confs\_1a2c\_ligand.mol2  
confs\_1a46\_ligand.mol2  
confs\_1a4w\_ligand.mol2  
confs\_1a61\_ligand.mol2  
confs\_1ad8\_ligand.mol2  
confs\_1dwc\_ligand.mol2  
confs\_1tbz\_ligand.mol2  
*etc.*  
omega2.log  
omega2.parm  
omega2\_status.txt

## matchprints/

1vr1\_biased\_interactions\_table.csv  
ligand.list  
side\_chain.list

## results/

1vr1\_biased\_example.results\_table  
1vr1\_merged\_example.results\_table  
1vr1\_unbiased\_example\_confs.results\_table  
1vr1\_unbiased\_example.results\_table  
result\_examples\_1vr1\_biased  
result\_examples\_1vr1\_unbiased  
result\_examples\_1vr1\_unbiased\_confs

## 10 Potential problems and solutions

(User Guide Appendix C)

Error	Possible cause	What to do
Segmentation fault (malloc failed) while generating the template	Number of initially generated points too high.	Decrease the volume of the template (preferred) or increase the grid spacing or clustering threshold slightly.
Neighbors of atom <i>xx</i> not found when generating the template.	Incomplete side chain, end of chain or highly unusual geometry.	Nothing to be done. There will be no template points generated for that particular atom.
FATAL ERROR: No protein pdb atoms found inside borders.xyz box (main)  mv: cannot stat '...': no such file or directory	Molecule used to define protein binding site in 'temp_gen -l' is not in the binding site of the protein.	Superimpose the ligand into the binding site, and regenerate the template
Segmentation fault while running SLIDE	Ligand directory is not found.	Check if the setup was done correctly and all the necessary directories are created.
	Files are not found.	Check if a) the SLIDE_DIR and SLIDE_DATA_DIR env. variables are set to the appropriate directories b) all the required files are in the right directories
	Too many template points.	Decrease number of template points preferably by decreasing the volume of the box in which the template is generated
	No key points marked.	Check the template file in the slide/target/template/in/ directory if there are key points marked (default: all key points).
write_ligand_mol2/ write_target_pdb/ write_log_file failed	SLIDE directory structure not correct.	Run <code>slide_setup.pl</code> .
	Out of disk space.	Output top-scoring matches only ( <code>#undef OUTPUT_ALL_MATCHES</code> in <code>params.h</code> and set <code>GROUP_CONFORMERS</code> to "true") or use disk with more space (directories can be linked into the SLIDE directory structure).
SLIDE is running, but no ligand is docked.	The mol2 files of the ligands are not found.	Check if the mol2 files of the ligands are in the directory indicated by <code>\$SLIDE_DATA_DIR/databases/&lt;database&gt;/&lt;database&gt;.db</code>

	The template is not good.	Create a <code>template.pdb</code> file by running “ <code>template_to_pdb template template.pdb</code> ” or “ <code>pdsize template&gt;template.pdb</code> ”. Look at the protein target and the template in Pymol, Insight, or Rasmol and make sure the template is filling out the binding site.
	Parameters are too stringent.	Choose different values for the parameters.
	The ligand has too few interaction points.	Check if the ligand has at least three interaction points. If not, SLIDE will not be able to dock it.
Other error messages		Check User Guide Appendix C.

For further help, please contact [slide@sol.bch.msu.edu](mailto:slide@sol.bch.msu.edu)