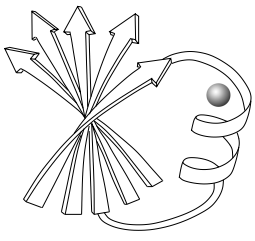
SLIDE — Screening for Ligands by Induced-fit Docking

User Guide

Version 1.1 – September 2000



Protein Structural Analysis and Design Laboratory
Department of Biochemistry and Molecular Biology
Michigan State University
slide@sol.bch.msu.edu

Contents

Chapter 1

Introduction

SLIDE stands for "Screening for Ligands by Induced-fit Docking", and it is a tool that is designed for searching large data sets of small molecule structures for potential ligands that are likely to bind to a given binding site of a target protein. The binding site is represented by surface residues, water molecules, and a set of favorable interaction points above the protein surface. Based on the choice of these template points, SLIDE is able to either find potential ligands that mimic known ligand binding modes, or to return a large variety of potential novel ligands.

SLIDE docks potential ligands into the binding site in order to evaluate shape and chemical complementarity of protein and ligand. During this docking process, induced flexibility of protein side chains is modeled by applying directed minimal rotations to rotatable side-chain bonds, which are basically all single bonds. SLIDE uses an approach based on mean-field theory to decide on the optimal, i.e., lowest-cost, conformational changes in both molecules that generate a shape-complementary interface, and both ligand and protein rotations are utilized.

Each protein-ligand complex is ranked based on the number of intermolecular hydrogen bonds and the hydrophobic complementarity of the contact interface. Ideally, SLIDE, reduces a database of some hundred thousands of compounds down to a ranked list of potential ligands, out of which the, say, top one-hundred compounds should be reasonable candidates for further inspection by either computational docking or binding assays.

This is the user manual for SLIDE, which provides an overview of the methodology and the use of SLIDE. It is not a software documentation, which describes the software, i.e., the data structures and modules used in the implementation of SLIDE. It should rather explain to the user the interface of SLIDE, including all auxiliary scripts and the directory structure of the file system that holds SLIDE's input and output files.

Chapter 2

Methods

This chapter describes the methods that are used in SLIDE. A more detailed description can be found in the following references:

- V. Schnecke, C. A. Swanson, E. D. Getzoff, J. A. Tainer, and L. A. Kuhn (1998) "Screening a Peptidyl Database for Potential Ligands to Proteins Including Side-Chain Flexibility", *Proteins: Structure, Function, and Genetics* 33, 74–87, and
- V. Schnecke and L. A. Kuhn (2000) "Virtual Screening with Solvation and Ligand-Induced Complementarity", *Perspectives in Drug Discovery and Design*, 20, in press.

The purpose of this chapter is to review the basic methods and to point out how the variables that can be modified by the user in a parameter file (see Section 3.4.1) fit in.

2.1 Template Design

Based on the classification of Kuntz *et al.*[1], SLIDE is a descriptor-matching approach, i.e., it uses a binding site template that defines points for favorable interactions with protein surface atoms. During the search, interaction centers (H-bond donors/acceptors or hydrophobic rings) of the screened molecules are mapped onto these template points to provide a basis for docking the potential ligands. Such a template can either be rather small and based on known ligand binding modes (Figure 2.1), or larger and automatically generated without any bias towards known ligands.

A template can include four different types of interaction points located in the binding site above the protein surface:

- **Hydrogen-bond donor.** During screening, SLIDE will place a hydrogen-bond donor of the

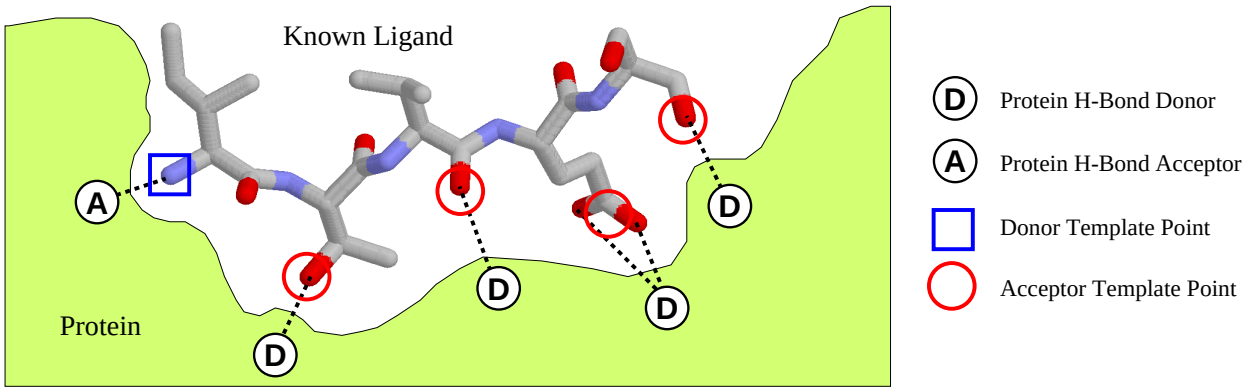


Figure 2.1: An example for a template that is based on the binding mode of a known ligand for the target protein. Either the position of a single hydrogen-bond donor and acceptor in this ligand or the average positions of two donors or acceptors, if they form a hydrogen bond to the same protein atom, are defining a template point.

ligand onto this point, which is within favorable hydrogen-bonding distance of a protein hydrogen-bond acceptor.

- **Hydrogen-bond acceptor.** Point within favorable hydrogen-bonding distance to a protein hydrogen-bond donor.
- **Hydrogen-bond donor/acceptor (doneptor).** This point is within hydrogen-bonding distance to both a hydrogen-bond acceptor and a donor of the protein, so either a ligand hydrogen-bond donor or acceptor can be placed here, or a hydroxyl group that can accept and donate at the same time.
- **Hydrophobic interaction center.** This point is placed above a hydrophobic surface patch of the protein, and SLIDE can match the center of a ligand carbon ring onto this point.

2.1.1 Template based on known binding modes

The program `average_template` (page 38) can be used to generate a template based on known ligand binding modes. It reads a set of ligand structures provided in the same reference frame like the binding site of the protein and identifies H-bond donors and acceptors and positions of centers of all hydrophobic carbon rings. Each of these points is a potential template point, close points of

the same type can be clustered using complete linkage clustering to reduce the number of template points.

2.1.2 Unbiased template

The template can be automatically generated using the program `unbiased_template`. (page 39). In this case the template is only based on the ligand-free structure of the protein, which reduces bias towards known ligands.

For this, the binding site is filled with random points that are 2.5 to 5.0Å from a protein atom. To determine favorable hydrogen-bonding positions, each of these points is checked for donors or acceptors in the protein within a distance of 2.5 and 3.5Å; for protein donors the angle between the donor, the donated hydrogen, and the probe point is also taken into account. Hydrophobic points are points which have a hydrophobic score of at least 3. This score is calculated as the number of hydrophobic atoms, defined as carbon atoms bonded only to other carbon atoms or hydrogen atoms, located between 2.5 and 5.0Å from the points minus the number of hydrophilic atoms, defined as other, non-carbon atoms. The values for the number of neighboring atoms of each type were chosen after an examination of the neighbors of surface points in a series of structures. To ensure even sampling, a large number of points, generally 10,000 to 40,000, are placed. After eliminating points either too close or too far from the protein and points not classified as either hydrogen bond forming or hydrophobic, there are generally 5,000 to 10,000 points remaining. All points of the same class are then clustered using complete-linkage clustering to yield a feasible number of template points, 100 to 150. Included in this program is the ability to specify different clustering thresholds for hydrophobic and hydrogen bond forming template points. The clustering threshold defines a maximal distance of points in a given cluster. Using a lower clustering threshold for hydrophobic points would yield a finer sampling of hydrophobic points, but more of them. The same would be true if using a lower clustering threshold for hydrogen bond forming points.

2.1.3 Unbiased template using a grid for initial point placement

A second method of automatic template generation is implemented using the `grid_template` program (page 40). This method is similar to the unbiased method described above, except the initial placement of points is at the vertices of a three-dimensional grid placed in the binding site. The grid spacing is generally 0.25–0.50 Å. This method provides two advantages over the random method: (1) templates can be repeatedly generated whereas the random method will produce slightly different templates each time it is run due to the initial random point placement and (2) the binding site is uniformly filled with initial points whereas the random placement can cause some

areas to be underrepresented. In both methods, initial points are clustered to give a tractable final set of points. In general, both methods produce a similar final set of template points due to this clustering.

2.1.4 Template with key points

There is a third way to design a template for the binding site, which is a mixture of the two approaches above. In this approach the user can assign specific key template points in the set of all template points. In this case only triangles of template points will be indexed in the hash tables (Section 2.3), if they include at least one of these key template points. Key template points are template points that are marked by an '*' following the type specifier in each line of the template file (for a description of the template file format see Figure 3.5, page 28). In the two other template design approaches described above all of the points in the template file are marked as key template points by default. The purpose of the key template points is to ensure that a least one interaction center of each docked ligand is located in a specific area of the binding site. In addition to this the use of key template points significantly speeds up the search time.

2.2 Binding-Site Waters

SLIDE can consider binding-site waters during docking. Typically, these water molecules are taken from the crystal structure of the ligand-free protein and *Consolv* is used to predict, if the waters are likely to be conserved upon ligand docking and mediate interactions between the two molecules. *Consolv* is a k-nearest-neighbor classifier system that predicts whether a water is conserved or displaced upon ligand binding based on its local environment[2]. Only those waters that are predicted as being conserved are kept, and based on *Consolv*'s votes, there is a penalty considered for displacement of waters that are predicted as conserved. This penalty is coupled to *Consolv*'s prediction confidence. For the format of the water file see Figure 3.6 on page 28, which also describes how to include the prediction confidence. Information on obtaining and using *Consolv* can be found under "Software" on the Protein Structural Analysis and Design Laboratory web site. (`http://www.bch.msu.edu/labs/kuhn`). Water molecules can also be included as a part of the protein PDB file itself, in which case they are never displaced.

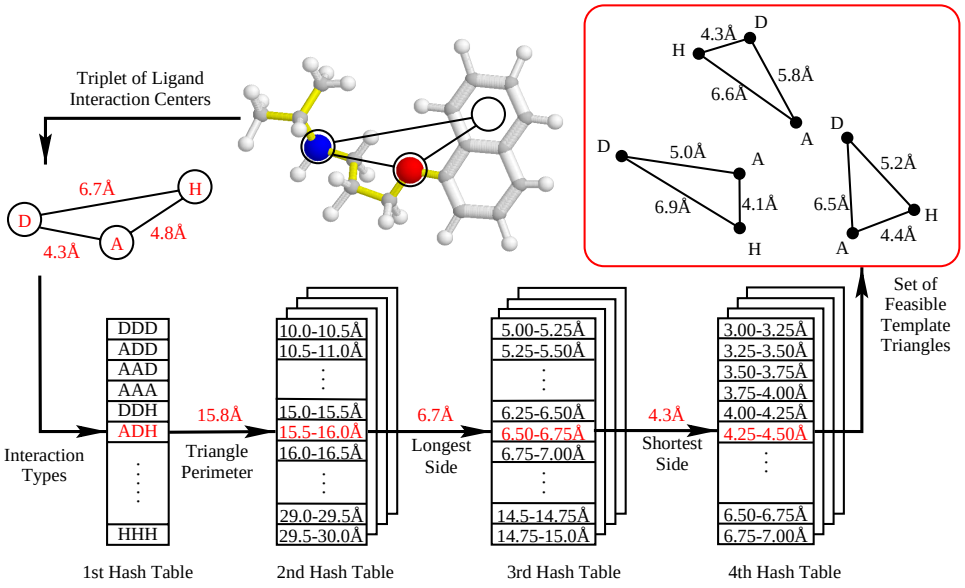


Figure 2.2: For each set of three ligand interaction centers all feasible template triangles are directly accessed via a four-level hashing scheme, which indexes the precomputed template triangles based on their chemical and geometrical properties.

2.3 Multi-Level Hashing

A multi-level hashing approach is used to directly access all feasible template triangles. Before the search, all possible triangles are generated out of the set of template points and are indexed via four levels of hash tables (see Figure 2.2). The parameter `MAX_TEMPLATE_TRIANGLES` defines the maximal number of triangles of template points that can be accessed via the multi-level hashing. This variable can be set in the parameter files (for a description of the use of parameter files see Section 3.4, page 22). The first hash table is based on the chemistry of the interaction points of the triangles. The four different types of template points provide 20 indices for the first hash table. The indices in the second hash table are based on the perimeter of the triangles, the third on the length of their longest side, and the fourth on the length of the shortest side. By using these four properties for a given set of three ligand interaction centers, the direct access to all template triangles with compatible properties is very efficient. However, all these triangles have to be checked for a feasible one-to-one mapping before the ligand is transformed into the binding site based on a least-squares fit superposition of the three ligand interaction centers onto the three template points. There are two variables that control thresholds for the triangle mapping,

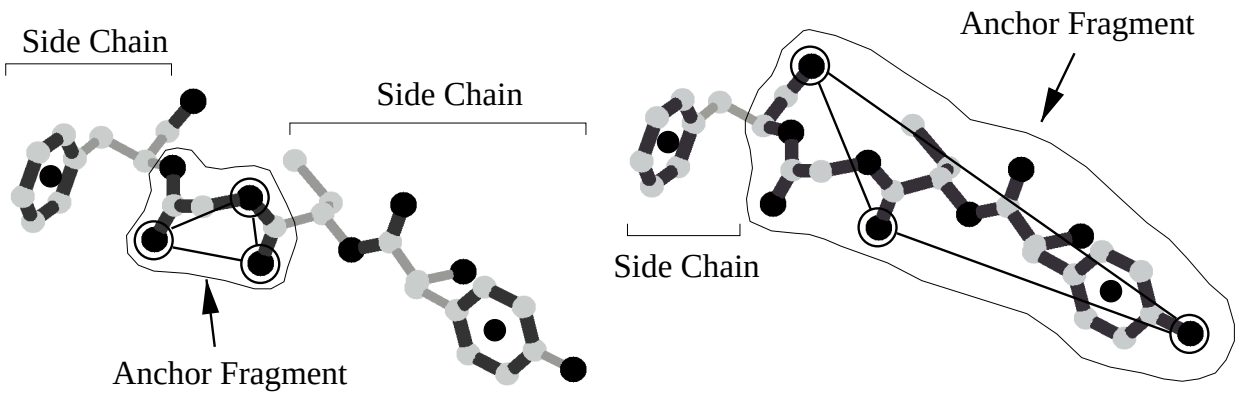


Figure 2.3: Two different triangle mappings are shown for one ligand. The choice of the three ligand interaction centers determines the rigid part of the ligand, its anchor fragment, all rotatable bonds in the remaining parts of the ligand (the side chains) are fully rotatable to resolve collisions after docking.

`DME_THRESHOLD` and `RMS_THRESHOLD`, both are set in the parameter files. `DME_THRESHOLD` defines the upper bound for the distance matrix error (DME) for the distances between the ligand triangle in comparison to the distances in the template triangle for a given one-to-one mapping. The mapping with lowest DME, if below the threshold, is used to superimpose the triangles onto each other. `RMS_THRESHOLD` defines the maximal accepted value for the root-mean square deviation of this superposition.

For all computations associated with the triangle mapping, those matchings that include hydrophobic interaction points or centers, respectively, have to be less exact. Weighted DME, RMSD, and least-squares-fit superpositions are computed to take this into account by weighting distances involving hydrophobic points with a factor of 0.3 relative to hydrogen-bond points.

The matched ligand interaction centers define the anchor fragment, and all chemically and geometrically feasible anchor fragments are tested for each ligand candidate. All flexible bonds within this part of the ligand are rigidified. The remaining parts of the ligand are kept flexible (Figure 2.3), such that all single bonds in these parts can be rotated later, if necessary to resolve collisions with protein atoms. Collisions of the anchor fragment with protein main-chain atoms are resolved by iterative translations of the fragment as a rigid body. The allowed overlap between anchor-fragment atoms and main-chain atoms in the target protein (including C_β's) is specified by the parameter `ANCHOR_OVERLAP`, the maximal translation distance is controlled by the parameter

ANCHOR_TRANSLATION. If all main-chain collisions can be resolved, the side-chains are added to the anchor fragment, in the conformation found for the ligand in the database.

2.4 Modeling of Induced Complementarity

Before resolving collisions and computing the chemical complementarity for a protein-ligand complex, all ligand dockings with low shape complementarity are ruled out. In 89 complexes [3], which were used to tune SLIDE's scoring function, on average 88% of the ligand carbon atoms were placed within a distance of 4.0Å of any protein atom. Based on this observation, in SLIDE all ligand dockings with more than 50% of carbon atoms without any protein atom within 4.0Å are discarded before doing the computationally expensive modeling of induced complementarity.

Induced fit for the interface between the two molecules is modeled by resolving any collisions of their flexible parts by directed rotations of single bonds either in the ligand or in side chains of the protein. The variable SIDE_CHAIN_OVERLAP in the parameter files defines the tolerated intermolecular overlap, INTRA_OVERLAP is the corresponding variable for the threshold of intramolecular overlaps. The radii used for protein and ligand atoms were taken from reference [4], for the protein the united atom model is used, i.e., hydrogen positions are not considered, and for the ligand atoms it is assumed that hydrogens are included in the structure of the ligands in the screening database. There are typically several applicable rotations to resolve an intermolecular collision, and an approach based on mean-field theory [5] is used to decide which rotations will improve the shape complementarity in the current conformation.

For each pairwise intermolecular collision, those bonds are identified that can be rotated to resolve it. The extent to which intramolecular collisions within the rotated side chain are tolerated during the optimization is controlled by the variable INTERMEDIATELY_TOLERATED_OVERLAP (see Section 3.4, page 22). All rotatable bonds that can be used for resolving a collision are stored in a system together with the corresponding minimum rotation angle and the number of non-hydrogen atoms that will be displaced by the rotation. These values provide the basis for the cost of a rotation. A probability is assigned to each rotation, and all rotations that can be used to resolve one particular collision are initialized with equal probabilities. During the mean-field based optimization, these probabilities converge to assign higher values to those rotations that represent a near-optimal choice to resolve a maximal number of collisions with minimal conformational changes in both molecules, without bias to one or the other.

In each cycle of the mean-field optimization process, a mean cost is computed for each rotation in the system, which is based on the cost associated with this rotation and on correlations with

other rotations in the system. Two rotations correlate, when they are not independent of each other, hence, only one of them should be applied at the end of an iteration of the mean-field optimization process. There are both positive and negative correlations, which decrease or increase the mean cost, respectively, and their contributions are weighted by the probabilities of the corresponding rotations. It is especially beneficial when a rotation of a single bond in the system can resolve more than one collision. An example for a negative correlation is a rotation that would displace another bond that is included in the system, so that all corresponding computations regarding that bond would no longer be valid.

The probabilities for all rotations in the system are updated at the end of each cycle, taking into account the mean costs of alternative rotations for the same collision. After up to ten cycles, the probabilities have converged to an approximate optimal set of values, which assigns the highest probabilities to those rotations that solve most of the collisions with the lowest overall cost. These rotations are applied if they do not cause any unacceptable intramolecular collisions. If there are still overlaps, up to ten iterations of the mean-field optimization are done to generate shape-complementary conformations of the two molecules. This comprehensive approach to simultaneously modeling protein and ligand flexibility provides a more realistic representation of induced complementarity than has achieved for other screening approaches, which focus only on ligand flexibility.

2.5 Scoring

Shape-complementary complexes are then ranked by a scoring function $\mathrm{SCORE}(\mathrm{P}, \mathrm{L})$ consisting of a term $\mathrm{HPHOB}(\mathrm{P}, \mathrm{L})$ for the hydrophobic complementarity of the interface between protein P and ligand L and a term for the number of intermolecular hydrogen bonds $\mathrm{HBONDS}(\mathrm{P}, \mathrm{L})$. For computation of the number of intermolecular hydrogen bonds, SLIDE identifies the favorable position of the shared hydrogen if the distance between donor and acceptor is less than 3.5 Å. For the protein and for ligands without hydrogen positions provided in the corresponding database structure, the positions of the hydrogens are computed; for all other cases the hydrogen position is taken from the crystal structure, and terminal hydrogens are rotated to optimize hydrogen bonding when applicable[6]. Donation to multiple acceptors is considered if the angular constraints are fulfilled. A distance of 1.0 Å is used between the donor and the hydrogen atom, and a range of 120° to 180° is accepted for the D–H$\cdots$A angle[7]. All hydrogen bonds are considered as giving equivalent contributions to the overall complementarity.

The hydrophobicity measure is based on residue and atom type for the protein and on atom type the ligand. The corresponding values were taken from a statistical survey of hydration of

atomic types in 56 protein structures[8]. The contribution of a single ligand atom is based on the comparison of its hydrophobicity value with the average hydrophobicity of the surrounding protein surface atoms. Given the hydrophobicity $h(a)$ of an atom a, with $h(a) \in [0...635]$ calculated as the average number of hydrations per 1000 occurrences of that atom type (Table II in ref. [8]), a value of 0 represents a maximally hydrophobic atom, 635 is maximally hydrophilic, and 317 is intermediate. The hydrophobic complementarity of the contact surface between protein P and ligand L is computed as:

$$\mathrm{HPHOB(P,L)} = \sum_{\substack{l_i \in L \\ \#P_i > 0}} \frac{\mathrm{avg}\{h'(l_i), \overline{h}(P_i)\}}{\max\{\mathrm{abs}\,(\,h'(l_i) - \overline{h}(P_i)\,), 32\}}$$

where

$$h'(l_i) = \max\{317 - h(l_i), 0\}$$

considers only the hydrophobic contribution of ligand atoms l_i. The hydrophobicity $\overline{h}(P_i)$ of the protein neighborhood P_i for a single ligand atom l_i is defined as the average hydrophobic contribution of all protein atoms p_j within a distance of 4.0 Å of l_i:

$$\overline{h}(P_i) = \max\{\,(\,317 - \frac{1}{\#P_i} \cdot \sum_{p_j \in P_i} h(p_j)\,)\,, 0\,\}$$

The denominator in each term of the sum describing the hydrophobic score ($\mathrm{HPHOB(P,L)}$) is always greater than or equal to 32, which is 10% of the maximum score for a single ligand atom. This ensures that the overall $\mathrm{HPHOB(P,L)}$ score is not dominated by a few contacts with very small differences between protein and ligand hydrophobicity.

The scoring function $\mathrm{SCORE(P,L)}$ for a collision-free complex is a linear combination of the two terms:

$$\mathrm{SCORE(P,L)} = A * \mathrm{HPHOB(P,L)} + B * \mathrm{HBONDS(P,L)}$$

The weights A and B have been reevaluated to reflect the affinities of 89 high-resolution complexes listed in reference[3]. These complexes had an average $\mathrm{HPHOB(P,L)}$ value of 28.7 and 7.8 hydrogen bonds. Values of 0.59 and 2.76 for A and B, respectively, give a reasonable approximation to the series of measured affinity values for these complexes (linear correlation coefficient of 0.615), which yields a relative contribution of 1.29:1.0 of hydrogen bonds over the hydrophobicity term. Although the weights were tuned based on experimentally observed affinities, it has to be emphasized that the goal is not to predict a binding affinity for a ligand identified and docked by SLIDE, but to relatively rank the potential ligands based on their complementarities in the given (non-optimized) conformation.

Chapter 3

Installation

The installation of SLIDE distinguishes between the global software directory, which is addressed with the environment variable `$SLIDE_DIR`, and the root of the directory structure that holds the input and output files for the screening runs, which is addressed by the environment variable `$SLIDE_DATA_DIR`. While the first variable is a pointer to the global installation of SLIDE, where the binaries, parameter files, and auxiliary scripts are located, the second variable is usually pointing to a directory that is local for the particular user, who does the screening runs. In this chapter are mainly the directory structure and the file format for SLIDE's input and output data files defined.

3.1 Hardware and Software Requirements

SLIDE has been tested on various Unix Systems, including Sun SPARC running Solaris 2.5–2.7, Intel running Solaris 2.7 and 2.8, and SGI Octane running Irix64. The particular hardware requirements depend on the application cases. The multi-level hashing is the most memory-demanding part. The minimal process size of any screening run is between 20 and 30 MB, depending on the system configuration (e.g., for Sun Solaris Systems the process size can be about 1.5 times larger on an Intel computer in comparison to a Sun SPARC machine for the same runs). The total process size depends on the number of indexed template triangles. With large, automatically generated templates with up to 200 points the process size can get close to 100 MB. However, the number of indexed triangles depends on the distribution of the template points, since there are upper and lower bounds for the perimeter and side lengths of those triangles, which are actually indexed in the set of all possible triangles.

In terms of disk space, the installation of SLIDE takes about 4 MB for the source code, the binary, and utility programs and scripts. All ligands in the screening databases have to be available

in Sybyl mol2 format and have to be generated and stored in addition to the vendor database that is screened. This can be quite space consuming, but the mol2 format provides all information that is needed to asses flexibility and hydrogen-bond capability of the compounds in the screening database, and this format is used by several other tools, which allows the direct use of SLIDE's output for further inspection (e.g., docking).

The required disk space for SLIDE's output depends on the number of potential ligands that are found and docked. In addition to the conformation of the docked ligand in mol2 format, the final positions of interfacial water molecules and the conformations for the rotated side-chains of the target protein are stored for each hit in the corresponding output directories (see Section 3.5). For a database of small ligands (e.g., CSD) the approximate disk space required for each final ligand conformation is in the order of less than 10 kB.

In terms of software requirements, an ANSI-C compiler is necessary (SLIDE has been tested with the SUN cc and GNU gcc compilers), and for the auxiliary scripts, Perl 5.x is needed. For the web interface that is used to browse through the set of potential ligands, a HTTP server that can handle CGI-scripts and a browser (e.g., Netscape) are also required. For visualizing the binding modes for the potential ligands, the molecular graphics tool Rasmol can be used. For use with the CSD and the web interface, the netpbm package, which is distributed with the X-Windows system, is necessary to transform the postscript files showing the compound structures, into a GIF image that is included in the html page for each potential ligand.

3.2 Compiling SLIDE and Customizing Perl Scripts

Before installing and compiling SLIDE, the tar file has to be placed in the directory that will be the root directory of the global SLIDE installation. The environment variable `$SLIDE_DIR` has to be set to the path of this directory before installing and using slide. The tar file SLIDE.TAR.GZ has to be unzipped and untarred by the commands

```
% gunzip slide.tar.gz
% tar xfv slide.tar
```

There is a script called `install_slide.pl` located in the directory `$SLIDE_DIR`, which will do the customization of the installation, which is mainly setting some global variables in the Perl scripts and compiling the source code for SLIDE and some auxiliary programs. If everything works correctly, the (almost) only customization during the installation is to set the call of the Perl binary in the first line of the script `install_slide.pl` to point to the Perl binary on the local system. The command `which perl` executed in a shell should provide this information. After this customization, the call `./install_slide` in the directory `$SLIDE_DIR` starts the

```
/psa/db/csd-select/A   A.pts
/psa/db/csd-select/B   B.pts
/psa/db/csd-select/C   C.pts
/psa/db/csd-select/D   D.pts
/psa/db/csd-select/E   E.pts
/psa/db/csd-select/F   F.pts
```

Figure 3.1: An example for a database definition file. Each line lists a directory, which includes a set of 3D molecule structures in mol2 format, and the name of the index file, which lists all interaction centers in the molecules.

compilation and installation of the SLIDE package.

After installation, it is useful to extend the path with that directory in the global SLIDE directory, which contains the binaries and the script files (directory `$SLIDE_DIR/bin/`).

3.3 Databases

The 3D structures of the molecules that are screened by SLIDE must be provided in the Sybyl mol2 format (Tripos, Inc.). A description of this file format is available on the WWW `http://www.tripos.com/TechBriefs/mol2_format/mol2.html`. In addition to the atom coordinates and the bond network for the molecule structure, this data format also provides information about the bond types and atomic orbitals, which is necessary to asses flexibility and H-bond donor/acceptor properties within the molecules. Partial charges do not have to be assigned to the atoms in these files, since this information is not needed by SLIDE. However, if SLIDE should be able to consider salt bridges, then all corresponding acceptors and donors should have a charge of -1.0 and +1.0 assigned, as it is by default in mol2 files generated from the CSD. Several different databases can be used within one SLIDE installation, each database is defined by a database definition file in the directory `$SLIDE_DIR/databases/`. The files have the extension `.db` and should have a short base name, since this base name will be used to name the corresponding output directories (see Section 3.5).

A database can be spread over several directories, and the database definition file lists all directories that include files belonging to this database (Figure 3.1). For the CSD there are several scripts available to prepare a subset of this database for screening with SLIDE, as described in Section 4.1. The first entry in each line in a database definition file provides the full pathname of a

directory that includes the mol2 files, which hold structures of the molecules in the database. The second entry in each line defines the name of the interaction file (extension `.pts`), which lists all interaction centers for the structures included in that directory. The interaction files are usually generated automatically when building a database.

3.3.1 Building a database

As already described in the previous section, a screening database in SLIDE is a collection of mol2 files, typically distributed over several directories. In each of these directories an interaction file with extension `.pts` contains all interaction centers of the molecules in the corresponding directories. After setting up the mol2 files, the program `compute_interaction_centers` has to be run for each molecule in the database and its output has to be redirected in to the interaction file. For the CSD there is a script `create_pts_files_csd.pl`, which does this after extracting molecules from the CSD (see Examples Section 4.1, page 29). This script also outputs the database definition file. For other databases there is a generic Perl script, `create_pts_files.pl`, which basically performs the same operations.

3.4 Parameter Files

3.4.1 The parameter files

There is a global parameter file for SLIDE, the default location for this file is in the directory `$SLIDE_DIR/params/`, the file is called `slide.parameters`. Figure 3.2 shows a sample parameter file.

In the global parameter file several variables are defined, which can be overwritten in either a local parameter file or by setting corresponding environment variables, e.g., in a batch file that starts several slide runs with different parameter settings. The variables are:

`DME_THRESHOLD:` This variable sets the maximum for the DME of a mapping of a triangle of ligand interaction centers onto a template triangle. All triangles with a lower DME will actually be superimposed onto each other by computing their least-squares fit (see Section 2.3 on page 14). A reasonable value for this variable is 0.2.

`RMS_THRESHOLD:` The maximal value for the root-mean-squares deviation of superimposed triangles, this value is typically identical to the value of the variable `DME_THRESHOLD`.

```
# default parameter file for SLIDE
DME_THRESHOLD                       0.2
RMS_THRESHOLD                       0.2
ANCHOR_TRANSLATION                  0.2
ANCHOR_OVERLAP                      0.1
SIDE_CHAIN_OVERLAP                  0.1
INTRA_OVERLAP                       0.1
INTERMEDIATELY_TOLERATED_OVERLAP    0.5
FINALLY_TOLERATED_OVERLAP           0.5
SCORE_CUTOFF                         30
MAX_TEMPLATE_TRIANGLES          1500000
```

Figure 3.2: A sample global parameter file for SLIDE.

`ANCHOR_TRANSLATION`: When resolving collisions between the rigid anchor fragment and any main-chain atom of the target protein, this variable limits the maximal contribution (in Å) for each dimension in the global translation vector that is defining the direction of the anchor-fragment translation. It is useful to restrict the amount of this translation in order to preserve the interaction described by the triangle mapping. A reasonable value for this variable is 0.2.

`ANCHOR_OVERLAP`: The value of this variable defines the tolerated overlap (in Å) between anchor-fragment atoms and any main-chain atoms of the target protein. A reasonable value for this variable is 0.1.

`SIDE_CHAIN_OVERLAP`: The tolerated van der Waals overlap (in Å) for a pair of protein-ligand atoms, a reasonable value for this variable is 0.1.

`INTRA_OVERLAP`: The tolerated overlap (in Å) for contacts between two atoms within protein or ligand.

`INTERMEDIATELY_TOLERATED_OVERLAP`: The sum of all intramolecular overlaps (in Å, involving atoms of a rotated side chain) that is tolerated during the mean-field based optimization when resolving collisions. This value is the overlap in addition to the amount controlled by the parameter `INTRA_OVERLAP`. The higher this value, the more likely is it that accepted intramolecular collisions cannot be resolved in later interactions of the mean-field optimization. A reasonable value for this parameter is 0.5.

```
#  definition of hydrogen-bond donor and acceptors
donor         N. ( H )
acceptor      O.3 ( 1 * ) [ N. ]
```

Figure 3.3: An example for entries in the hydrogen-bond parameter file. The first line describes a nitrogen that is bonded to at least one hydrogen, which can be donated to form a H-bond, the second line describes an acceptor, a sp^3 oxygen that is bonded to at least one other atom, but not bonded to any nitrogen. `[..]` describes atoms that are not allowed to be connected to the atom specified in the beginning of the line in order to act as a donor or acceptor, `(..)` describes atoms that have to bonded to this atom in order to accept or donate a hydrogen bond.

`FINALLY_TOLERATED_OVERLAP`**:** The sum of all overlaps (intra and inter) in both molecules that is tolerated in a final docking of a ligand unless no collision-free conformation was found for this ligand. A default value of 0.5 is reasonable to ensure that after the last iteration of the mean-field optimization no ligand orientation is ruled out because of only minor remaining collisions. Maximal one conformation with remaining collisions is stored for each compound, and this will be replaced by a collision-free docking later during the search, if that has a higher score.

`SCORE_CUTOFF`**:** This variable defines the lower score for all ligands that are stored in the output directories. Feasible values depend on the size of the potential ligands that are most likely to be found and on the selectivity of the template. For larger compounds a value of 30 is reasonable, for smaller compounds or very restricted binding sites a value of 20 or even 15 might be more suitable. If too many potential ligands are found during the search, the score cutoff can be controlled afterwards by using the Perl-script `filter_ligands.pl` to delete all ligands with a score below a given cutoff value (see Section 5.8.14).

`MAX_TEMPLATE_TRIANGLES`**:** This variable defines the maximal number of template triangles that are indexed in the multi-level hashing. The number of indexed triangles mainly determines the memory usage of SLIDE, since each single template triangle needs usually 36 bytes of memory.

3.4.2 The hydrogen-bond parameter file

The hydrogen-bond parameter file is stored in SLIDE's global parameter-file directory `$SLIDE_DIR/params/` and is called `hbond.defn`. Figure 3.3 shows two entries in the hydrogen-bond parameter file. This file is read in by the program `get_interaction_centers`, which identifies H-bond donors and acceptors and hydrophobic centers in molecules provided in mol2 file format. The syntax is the same as the one used in DOCK[9–11] parameter files. The first entry in each line defines the interaction type (`donor` or `acceptor`), the second entry the atom type together with an optional orbital specifier, and then a list of atoms that have to be bonded with this atom (in `(..)`) and other atom types (in `[..]`) that are not allowed to be bonded with this atom in order to donate or accept a hydrogen bond.

3.4.3 The flexible-bond parameter file

The flexible-bond parameter file is also stored in SLIDE's global parameter-file directory `$SLIDE_DIR/params` and is called `flex.defn`. It describes all types of flexible bonds by listing all pairs of atoms that can be connected by a flexible bond. Each entry consists of three lines, a name entry with the keyword `name` as the first entry in the line followed by two lines with the keyword `definition`, each line describing an atom entry in the same manner like in the hydrogen-bond definition file described in the previous section.

3.5 Results Output

The output files are organized in a fixed directory structure, the root directory of which is specified by the environment variable `$SLIDE_DATA_DIR`. The Perl-script `slide_setup.pl` can be used to generate the directory structure for a new screening experiment. The first level under this main data directory is based on the specifier of the target protein, e.g., on its PDB-code. The second level is based on the template. Target protein and template define a screening run, and in the corresponding subdirectory all input and output information is stored.

The input for a screening run is specified in the subdirectory `in/`: There is a PDB file with the binding-site residues (file `<target>.rad`), which is typically generated using the Perl script `binding_site_residues.pl` (see page 42 in the Reference Section). The file `template` lists the positions of the template points in the format defined in Figure 3.5. The file `waters.pdb` lists the conserved binding-site waters including the value for the confidence of the prediction, as described in Figure 28. Two more files, `binding_site.pdb` and `ligand.pdb`, are PDB files

that are used for visualizing the results via a web interface (see Section 4.3, page 33). They are only necessary, if the web interface-feature is used in the installation.

The output of each screening run is stored in subdirectories that are associated with the screening database. The directories `<db>_ligands`, `<db>_targets`, and `<db>_waters` store the potential ligands, the rotated target side chains, and the positions of the conserved water molecules, respectively. All these filenames are based on the reference code for the compound in the database, the ligand is in mol2 format, the side-chain and the water files are in PDB format.

Log files for each screening run are stored in the directory `log`, again associated with the corresponding database specifier, i.e., the file names are `<db>.log`. The log file lists all parameter settings that were used for the screening run, it provides statistics for the multi-level hashing, and it outlines the efficiency of the various filter steps used in SLIDE to efficiently rule out infeasible ligands. For the latter, the number of all molecules that were ruled out in each of the particular steps are listed. At the end of the log file the host, the date, and the overall screening time are listed.

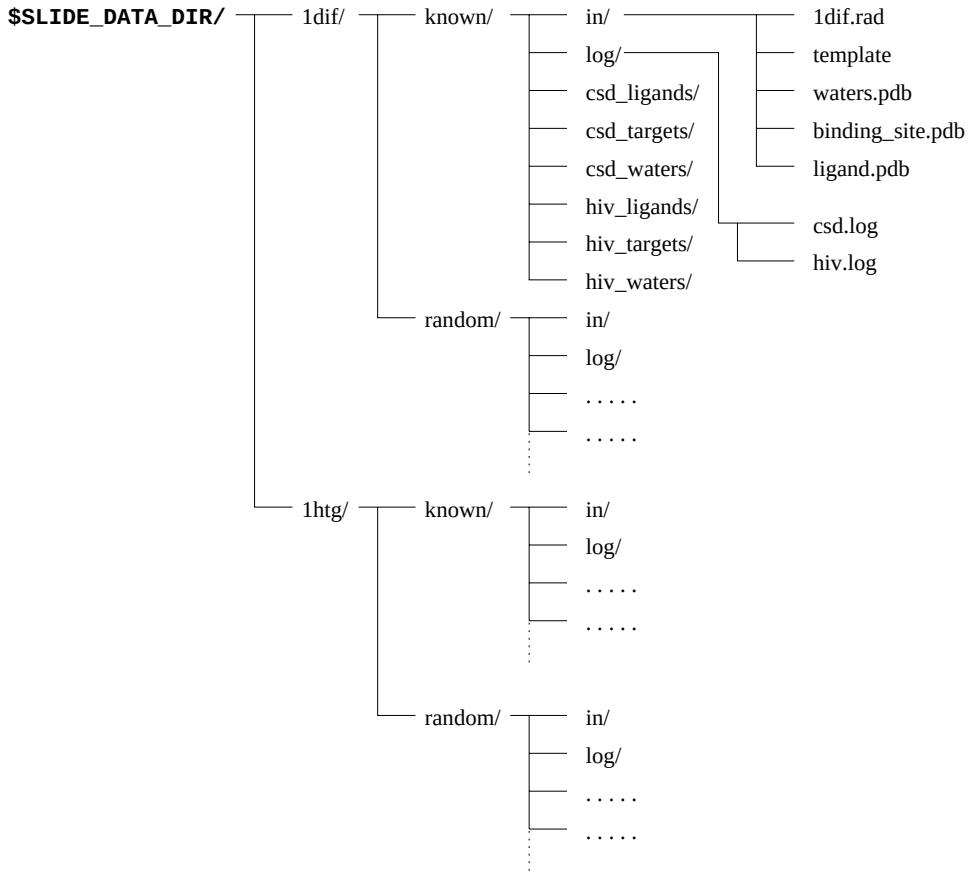


Figure 3.4: A sample directory structure containing SLIDE's input and output files for two target proteins, PDB 1dif and 1htg. For each of these targets there are two different templates ('known' and 'average'), and two databases are screened ('csd' and 'hiv').

```
# a template file with four key points
H*   5.475  70.192  67.458
H*   4.337  68.626  62.846
A*   5.604  64.649  62.291
A    4.337  61.339  68.837
A    0.887  58.801  71.912
A   -0.880  61.099  72.599
D*   6.169  69.585  64.899
D    2.191  69.166  65.940
N    4.451  63.645  60.536
```

Figure 3.5: A sample template file for a template of nine points, two hydrophobic points (`H`), four H-bond acceptor points (`A`), two donor (`D`), and one doneptor (donor/acceptor) point (`N`). The points marked with `*` are key template points, i.e., each template triangle that is indexed in the multi-level hash tables must include at least one of these template points.

```
HETATM 4157  O   HOH   603       0.233  67.584  63.976  0.55 27.16
HETATM 4162  O   HOH   608       0.361  69.609  59.530  0.71 12.40
HETATM 4169  O   HOH   615       3.884  69.138  57.717  0.69 12.05
HETATM 4172  O   HOH   618      -1.808  68.459  66.870  0.62 23.65
```

Figure 3.6: A sample water file which lists four binding-site water molecules in PDB format. The field after the coordinates, which usually lists the occupancy for an atom, lists the prediction for the conservation of this water, based on *Consolv*'s prediction (0.5 weakly conserved to 1.00 strongly conserved). Penalty terms for displacement of a water by a non-polar ligand atom are coupled to this value.

Chapter 4

Examples

The screening database for SLIDE is just a set of mol2 files, stored on a regular Unix file system. The current distributed version of the software (version 1.0) includes several Perl scripts to set up a database from a set of structures extracted from the CSD. Further there are scripts to set up a web interface, which requires a browser like Netscape and an installation of Rasmol for visualizing the results of a screening run, i.e., the set of potential ligands with the information about their score, side-chain rotations during docking, information about displaced and conserved water molecules, and their binding mode within the protein and relativ to a known ligand. While in this chapter only the installation of a CSD subset for screening is outlined, all described scripts can be used as templates for scripts to prepare selected compounds from other databases for screening and visualization of the results.

4.1 Setting Up a CSD Subset for Screening

The standard interface for requesting structures from the Cambridge Structural Database System (CSD) is the program `quest`. This has a graphical user interface to define a query by drawing substructures, entering keywords, or use predefined bit screens.

When preparing a database for screening with SLIDE, `quest` has to output a single mol2 file containing the structures of all compounds that match the search criteria. The final input for SLIDE consists of mol2 files for each of the compounds and `.pts` files that list the interaction centers (positions of H-bond donors/acceptors and carbon ring centers) for all compounds.

There are several steps involved after retrieving the structures from the database before a screening run can be started:

1. The one mol2 file output by `quest` containing thousands of molecules has to be cut into

separate mol2 files for each CSD entry. A sample-file `csd.mol2` is included in the distribution of SLIDE and can be found in the `examples/` subdirectory. To split the screening database into several subdirectories, the Perl script `cut_mol2_csd.pl` has to be used. The corresponding call is:

```
% cut_mol2_csd.pl csd.mol2
```

It generates subdirectories `A/` to `Z/` in the current directory and moves all CSD entries from the large mol2 file into a corresponding subdirectory based on the first letter of their CSD reference code.

2. The Perl script `process_csd_set.pl` checks for all mol2 files in subdirectories `A/` to `Z/`, if they include different residues. Many CSD entries contain more than one molecule due to co-crystallization, or more than one copy of a molecule, depending on the symmetry within the crystal. If a CSD entry contains more than one residue, the original mol2 file is split into several files called `<CSD_refcode>_<resnumber>.mol2`. This script further checks the number of atoms for each residue, mol2 files of residues with less than three and more than 100 atoms are deleted.

3. The Perl script `create_pts_files_csd.pl` computes all interaction centers from the molecules in the mol2 files in subdirectories `A/` to `Z/` and stores them in files `A/A.pts` to `Z/Z.pts`. If a molecule has less then three interaction centers, then the corresponding mol2 file will be deleted, since such a molecule cannot be docked into the binding site due to the required triangle mapping in SLIDE.

To make a database available for screening, the directories including the mol2 files and the corresponding pts files have to be listed in a global database description file in the directory `$SLIDE_DIR/databases`, as described in Section 3.3, page 21. This is automatically done when running the script `create_pts_files_csd.pl`.

When `quest` was run to select compounds from the CSD, a jnl file was generated that lists the reference code for each retrieved compound together with its name, its formula, and the associated publication for this entry. The Perl script `create_csd_set_data.pl` takes this file and generates files `A.data` to `Z.data`, which hold the information from the jnl files, so that they can be extracted, for example, when creating data sheets for the web interface to visualize the results.

4.2 Screening a Database

Each screening run done by SLIDE is identified by a target specifier for the protein and a template identifier for the binding-site template. The directory structure for the input and

output files has to be set up as described in Section 3.5, page 25, which can easily be done using the Perl script `slide_setup.pl`. The input directory for a screening run is `$SLIDE_DATA_DIR/<target>/<template>/in/`. There are several scripts that prepare the input files, some of them need the full PDB file for the target protein in the root directory for this target: `$SLIDE_DATA_DIR/<target>/<target>.pdb`. This file should include only those atoms of the target protein that are considered for screening as being part of the protein, i.e., heteroatoms that should be ignored during screening (e.g., ligands that are located in the binding site) have to be deleted from the original PDB file. All water molecules, which should be considered during screening, have to be included in this file, and must have the residue-name `HOH`. All non-water heteroatoms cannot be diplaced upon ligand docking. In contrast to protein side chains, all positions for these heteroatoms are fixed during screening in the current version of SLIDE (Version 1.0). All heteroatoms with names beginning with `N` or `O`, for nitrogen and oxygen, respectively, can be considered as H-bond donors, acceptors, or doneptors (donor/acceptor), by changing the second and third letter of the atom name to `DD`, `AA`, and `NN`, respectively. If salt bridges should be considered during screening, , the charges of the corresponding donors and acceptors have to be specified in columns 79 and 80 of the atom definition line by listing the strings `1+` and `1-`, respectively, as defined in the PDB file format description available at the RCSB `http://nist.rcsb.org/pdb/`.

In the following the screening of a CSD subset for potential ligands to the human estrogen receptor will be described as an example. There are currently only structures of complexes available in the Protein Data Bank for this receptor. PDB entry 3erd is a complex with the ligand diethylstilbestrol (DES), a pharmaceutical estrogen. This PDB file is included in the SLIDE distribution and can be found in the directory `$SLIDE_DIR/examples`. Since an unbiased template will be generated for this target, we use the template specifier `unbiased`, and for setting up the directory structure for screening the CSD, the call of the Perl script `slide_setup.pl` is:

```
% slide_setup.pl 3erd unbiased csd
```

The original PDB file includes four copies of the ligand-binding domain of the receptor, chains A, B, C, and D, each with bound DES. For screening purposes, only one copy is needed, so only chain A is extracted and stored in file `$SLIDE_DATA_DIR/3erd/3erd.pdb`. The file `$SLIDE_DIR/examples/3erd_A.pdb` has this format.

The next step is the design of the binding-site template. For an unbiased template, this is done with the program `unbiased_template`. Before running that program, the corners of a box outlining the binding site have to be defined. An easy way to do this is to orient the structure in a molecular graphics program in a way that the axes of the base coordinate system are aligned with the screen and then pick the coordinates of atoms that mark the border

of the binding pocket for one or more dimensions. The coordinates for these atoms have to be stored in the file `$SLIDE_DATA_DIR/3erd/unbiased/in/borders.xyz`. The file `$SLIDE_DIR/examples/borders.xyz` is an example for such a file. To generate the unbiased template, the command

```
% unbiased_template 3erd unbiased 60000 2.0
```

is used. A log file called `template.log` will be created in the input directory after completion of the program. This log file lists statistics for the current run. The program first fills the box outlined by the coordinates given in file `borders.xyz` with 60,000 random points. The first line in the log file gives the average number of random points per Å^3, 20 points per Å^3 is a reasonable density. After their generation, these points are filtered to keep only those that are in feasible interaction distance to protein surface atoms, which can be anything between 2,000 and 15,000 points, depending on shape and size of the binding site. Each of these points is assigned one of the four types of template points and then complete-linkage clustering is used to reduce the total number of template points to a tractable number. SLIDE can typically handle up to 200 template points. However, depending on shape of the binding site, the screening time can increase drastically with the template size due to a larger number of potential triangle mappings. For a completely enclosed binding pocket, as in the case of the estrogen receptor, the screening time for 100,000 compounds should be in the range of a few hours, even for very large templates. In the example command line above a threshold of 2.0Å is used for the complete-linkage clustering, which yields about 160 template points. Larger thresholds reduce the template size, but increase the clustering time. A threshold of 2.0Å means that the clustering stops as soon as the longest distance between any pair of points that are joined in a cluster reaches 2.0Å. However, this does not mean that the minimal distance between final cluster points is 2.0Å! Typically, several runs of the program `unbiased_template` are necessary to figure out an optimal threshold for a desired template size.

In addition to template and the log files, the program `unbiased_template` as well as its counterpart, the program `average_template`, generate a file `template.pdb`, which lists the template points as oxygens of water molecules in a PDB file. In these files the temperature factor for the atoms is set to 100, 50, 25, and 0, for hydrophobic, donor, acceptor, and deneptor points, respectively, which gives the user the possibility to automatically color template points differently for the four types using a temperature color spectrum in a molecular graphics program.

By default, all template points are marked as key template points (see Figure 3.5, page 28), i.e., all possible triangles of these points are indexed via the multi-level hashing scheme. The Perl script `unmark_key_template_points.pl` can be used to delete the key-point specifier '*' from the template file, so that particular points can be interactively marked as key points by the

user, even for an automatically generated template.

After template generation, the shell of binding-site residues for the receptor has to be extracted from the PDB file. This is done by the command

```
% binding_site_residues.pl 3erd unbiased 9.0
```

which extracts all residues from the PDB file `$SLIDE_DATA_DIR/3erd/3erd.pdb` and stores them in the file `$SLIDE_DATA_DIR/3erd/unbiased/in/3erd.rad`. All heteroatoms from the base file are automatically included in that file, since it is assumed that those atoms are located in the binding site, anyway.

When binding-site waters are included, these have to be stored in a file called `waters.pdb` in the input directory. The Perl script `binding_site_waters.pl` is used for this:

```
% binding_site_waters.pl 3erd unbiased 5.0
```

This call selects all water molecules from the PDB file `$SLIDE_DATA_DIR/3erd/3erd.pdb` that are within 5.0 Å from any of the template points and stores them in a file `waters.pdb` in the input directory. The user can then interactively change the occupancy entries for the atoms in this PDB file to define a penalty term for displacement of any of these waters by a non-polar ligand atom (see sample water file in Figure 3.6, page 28).

For screening the CSD, the user then has to make four subdirectories `csd_ligands`, `csd_targets`, `csd_waters`, and `log` in the directory `$SLIDE_DATA_DIR/3erd/unbiased/`. The final call to start SLIDE for screening this database for target 3erd with the unbiased template is:

```
% slide 3erd unbiased csd > csd.out &
```

It is reasonable to redirect the output of SLIDE into a file and start the program as a background process. The output file is especially useful to track the state of the screening, or to subsequently figure out, which ligand caused the program to crash. Usually 99.9% of crashes of SLIDE are based on problems with the ligand file format, like unknown atom or bond types. In these cases it is helpful to know, which the last mol2 file was that SLIDE read, so that this can be inspected to track down the problem.

4.3 A Web Interface to the CSD

The Perl script `create_html_files_csd.pl` is creating a global table for the top ligands SLIDE has found for a target protein and sets up a set of HTML files showing data sheets for all the ligands and retrieves a graphical representation of the structure of the potential ligand, which is transformed into a GIF file and included in the database. This script calls the Perl script `create_global_csd_results_table.pl` to generate a list of the top ligands that SLIDE has

found. In this script the variable `$CSD_SET_DATA_DIR` has to be customized to point to the directory in which the output-files `A.data` to `Z.data` of the script `create_csd_set_data.pl` are stored in order to list the compound name from the CSD on each HTML data sheet.

There are two CGI scripts for the HTTP server that are used to send PDB files to clients, which then call a molecular visualization tool like Rasmol to show the binding mode for a particular ligand:

- The Perl script `binding_site.pl` shows the ligand in the binding site including the final conformation for the protein side chains as proposed by SLIDE and all conserved binding-site waters.

- The Perl script `known_ligand.pl` shows the current ligand together with a known ligand, which is specified in the input directory for the corresponding screening run.

All these scripts are described in detail in the Reference Section (Chapter 5, page 41). The script `create_html_files_csd.pl` requires the netpbm package for transformation of the postscript representation for the selected potential ligands retrieved from the CSD into GIF files.

To set up the web interface for the example described in the previous section two files have to be created in the input directory, that hold the known ligand and the binding-site residues for the estrogen receptor. Assuming that the ligand DES from the A chain in PDB 3erd is stored in PDB file `des.pdb` in the input directory (a sample file can be found in the `examples/` directory of the SLIDE distribution), then the Perl script `renum_pdb_atoms.pl` can be used to create the actual ligand file that is requested by the CGI-script `known_ligand.pl`:

`% renum_pdb_atoms.pl des.pdb LIGAND > ligand.pdb` The same script is used to generate the counterpart of the ligand file for the binding site PDB file:

`% renum_pdb_atoms.pl 3erd.rad TARGET > binding_site.pdb` Both calls should be done in the input directory `$SLIDE_DATA_DIR/3erd/unbiased/in/`.

To enable a web bowser like netscape to call Rasmol after following a link on the web pages, Rasmol has to be listed as the application for the corresponing mime types of the data sent by the CGI scripts, which is `chemical/x-pdb`. To do this, in the Netscape 4.x navigator window the menu `Edit->Preferences` and then in the left part of the pop-up window `Navigator->Application` have to be selected. This shows all mime types and the applications that will be started by Netscape, if the navigator receives a document of a prticular type. If there is no entry for the mime-type `chemical/x-pdb`, then a new entry for this type has to be added and an example for the application field in this entry is

```
xterm -geometry 50x10 -e rasmol %s
```

```
wireframe 0.2
spacefill 0.2
set mouse insight
```

Figure 4.1: An sample initization file for Rasmol that has to be stored for example in the home directory of a user. This file changes the default settings for Rasmol to a tube representation for molecules and spheres for single atoms. The last command allows the user to navigate with the mouse in the same way as in MSI's insightII molecular graphics software.

This command opens a small X-terminal and executes Rasmol, which is directly reading the transfered document in PDB format and displays the molecules. It is useful to create a file called `.rasmolrc` in the home directory, which is read by Rasmol upon initialization. The sample file shown in Figure 4.1 shows the molecules as tubes and with the `spacefill` command ensures that single waters in the binding site are visible.

4.4 Summary of Steps Required for Setting Up and Starting a Screening Experiment

Here is a summary of all necessary steps for setting up a screenig experiment and starting SLIDE, in all commands and filenames, the expressions `<target>`, `<template>`, and `<database>`, represent the actual target code, the template specifier, and the database specifier, respectively:

1. To set up the directory structure for the input/output files associated with our screening experiment, we first call the Perl script `slide_setup.pl`:
   ```
   % slide_setup.pl <target> <template> <database>
   ```

2. Then we store the PDB file that defines the structure of our target protein in the root directory for this target and name it accordingly, i.e., this PDB file is `$SLIDE_DATA_DIR/<target>/<target>.pdb`. In addition to the atoms in the residues of the protein, this PDB file should only include heteroatoms (specified by keyword `HETATM`), if they should be considered as a (rigid) part of the protein during screening. Water molecules (residue name `HOH`) will be ignored.

3. Now we have to generate a template for the binding site. This can either be done by running the program `unbiased_template` (page 39), the program

`average_template` (page 39), or by any other approach that provides the template file `$SLIDE_DATA_DIR/<target>/<template>/in/template`, which lists interaction centers that are located in the binding site above the protein surface.

4. The next step is the extraction of all binding-site residues and (non-water) heteroatoms from the file `$SLIDE_DATA_DIR/<target>/<target>.pdb`, which are within a certain distance from the template points. This will be the only residues that are considered for collision and complementarity checks during screening. The corresponding call is extracting all residues with any atoms within 8.0 Å distance from a template point:
   ```
   % binding_site_residues.pl <target> <template> 8.0
   ```

5. If solvent should be considered during screening, the following call extracts all water molecules with 5.0 Å from any template point from the protein PDB file in the root directory for this target and stores them in the waters-file `$SLIDE_DATA_DIR/<target>/<template>/in/waters.pdb`:
   ```
   % binding_site_waters <target> <template> 5.0
   ```
   To correctly consider penalties for replacement of these waters when docking potential ligands, the B-values in the waters file have to be modified as outlined in Figure 3.6 (page 28).

6. Finally, the actual screening run has to be started by the command:
   ```
   % slide <target> <template> <database>
   ```
   If for this experiment any parameter settings other than the global parameters specified in file `$SLIDE_DIR/params/slide.parameters` are used, settings in a parameter file called `$SLIDE_DATA_DIR/<target>/<template>/in/slide.parameters` can override the global parameter values. The command to start SLIDE screens a full database, which is defined in a database definition file within the global directory `$SLIDE_DIR`. For screening local databases, or only a list of a few compounds, SLIDE can be called with different parameters, which is outlined in the Reference Section (page 37).

Chapter 5

Reference Manual

5.1 slide

There are basically four ways to start SLIDE after setting up the databases and the directory structure for the input and output files. SLIDE can be used to screen a globally defined database, a local database, or a set of single compounds specified either as command-line arguments, or listed in a file. SLIDE automatically recognizes, which of the four possible options the user selects:

Usage: `slide target template database`

If only target, template, and database are specified as command-line arguments, the corresponding screening run is started, i.e., database `database` is screened for potential ligands for the protein `target` with binding-site representation `template`. The database `database` is a globally defined database, i.e., there is a corresponding database definition file in the global installation directory `$SLIDE_DIR/databases`.

Usage: `slide target template database mol2-file1 [mol2-file2...]`

This call screens a set of single mol2 files, which are provided as command-line arguments. A selection of one or more mol2 files is screened, using template `template`, and stores the potential ligands in the directory `<database>_ligands`. This mode of running SLIDE is useful when checking the binding mode of, for example, single known ligands, before screening a full database to provide a feeling for the docking and score SLIDE assigns to these compounds in order to optimize the template or the parameter settings. Since this is not a full database screen, no log files are written.

Usage: `slide target template database compound-file`

This call also screens a set of compounds, but other than in the previous case, the names of the

corresponding mol2 files are not passed as command-line arguments, but are rather listed in file, which lists a single mol2 file in each line. Since this is not a full database screen, no log files are written.

Usage: `slide target template database local-db-file`

In this case the fourth command-line argument in the call of SLIDE is the name of a local database definition file, the corresponding file format is outlined in Section 3.3 (page 21). This option is useful, is a user has set up a database within her/his local environment. In this case log files are written in the log file directory.

5.2 slide_score

Usage: `slide_score protein-file ligand-file [waters-file]`

The program `slide_score` runs the scoring function of SLIDE for a given protein-ligand complex. The parameter `protein-file` is a file in PDB format that describes the structure of the target protein, and `ligand-file` is a mol2 file including the ligand in its binding conformation in the same reference frame as the protein. The file `waters-file` lists all water molecules that mediate interactions between the two molecules, this file is optional, all water molecules that are included in the PDB file for the protein are ignored.

5.3 average_template

Usage: `average_template threshold ligand1.mol2 [ligand2.mol2 ...]`

This program generates a template that is based on the chemistry of known ligands, typically based on their binding modes from crystal structures with the target protein. The complexes with these ligands are superimposed onto the structure of the target protein that is used for screening, and the corresponding ligand orientations have to be stored in mol2 format. The program `average_template` reads in one or more ligand mol2 files and computes the centers of their carbon rings and the positions of hydrogen-bond donors and acceptors in these ligands. Then complete-linkage clustering is used with the threshold that is defined as the first command-line parameter to cluster points of the same type. The output files are the template-file `template`, a log-file `template.log`, and a PDB-file `template.pdb` that lists the template points as oxygens of water molecules. In these files the temperature factor for the atoms is set to 100, 50, 25, and 0, for hydrophobic, donor, acceptor, and doneptor points, respectively, which gives the user the

possibility to automatically color template points differently for the four types using a temperature color spectrum in a molecular graphics program.

5.4 unbiased_template

Usage: `unbiased_template target template number_of_points hydrogen-bond_threshold hydrophobic_threshold [ligand.mol2 distance]`

This program generates an unbiased template for the binding site of the protein that is described by the PDB file `$SLIDE_DATA_DIR/<target>/<target>.pdb`. All heteroatoms but waters (residue name `HOH`) from this file are considered. The program requires a file `borders.xyz` in the directory `$SLIDE_DATA_DIR/<target>/<template>/in/` that contains a list of points (as x y z coordinates in each line), which define the borders of the binding site. If a set of 8 points is given as the border points, the box, not necessarily a cube, is created using these 8 points as the corners. When giving 8 points, the points must be in a specific order relative to each other as shown in figure 5.1. If other than 8 points are given, the box is defined as the minimal box (edges parallel to base coordinate system) including all listed points. The box is filled with `number_of_points` random points, out of these all points are deleted that are either located underneath or are too close to the protein surface, or too far away from any protein atom to serve as template points. If a reference molecule is given, points not within `distance` of the molecule are also removed. For each of the remaining points it is checked, whether there is a H-bond donor/acceptor of the protein in feasible distance and angle, or if the point is located above a hydrophobic surface patch. Finally all points of the same type are clustered using complete-linkage clustering with threshold `hydrogen-bond_threshold` for hydrogen bond forming points (acceptor, donor, or doneptor) and with threshold `hydrophobic_threshold` for hydrophobic points.

The program creates several output files, all are stored in the input directory `$SLIDE_DATA_DIR/<target>/<template>/in/`. The file `template` is the template definition file that is read by SLIDE during screening, the file `template.log` is a log file, and the file `template.pdb` lists the template points as oxygens of water molecules. In these files the temperature factor for the atoms is set to 100, 50, 25, and 0, for hydrophobic, donor, acceptor, and doneptor points, respectively, which gives the user the possibility to automatically color template points differently for the four types using a temperature color spectrum in a molecular graphics program. There are five files that list points in MSI's usr format and can be read by MSI's insightII to visualize the distribution of random points onto which the template is based. These files are called `points.usr`, `hphob.usr`, `donor.usr`, `acceptor`, and `doneptor`, and include

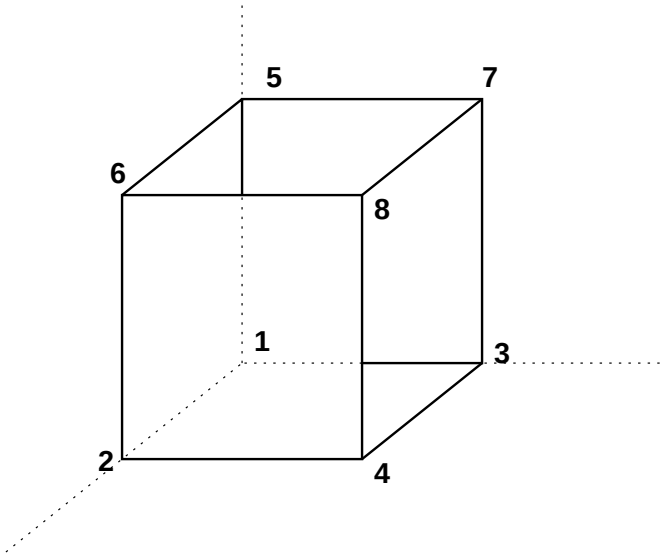


Figure 5.1: Point relationship used for 8 cornered box definition in `unbiased_template` and `grid_template`. If using an 8 cornered box in the random template creators, the points must be order in such a way as the program is able to correctly identify the connectivity. Rotations of the box, which move the points in space but retain their connective relationships, are allowed.

all random points, the hydrophobic points, and all H-bond donor, acceptor, and doneptor points, respectively.

5.5 grid_template

Usage: `grid_template target template grid_spacing hydrogen-bond_threshold hydrophobic_threshold [ligand.mol2 distance]`

This program is similar to the `unbiased_template` described above. Instead of placing the initial set of potential template points at random in the binding site box, defined by the `borders.xyz` file, it places the initial points at the vertices of a three dimensional grid with grid spacing `grid_spacing`. All other aspects are similar to the `unbiased_template`.

5.6 check_connectivity

Usage: `check_connectivity mol2_file`

The program `check_connectivity mol2_file` is a reduced version of the program `compute_interaction_centers`, which is described in the previous section. It reads in a mol2 file and checks, if all atoms in the molecule described in this file are connected by bonds. If the mol2 file contains more than one unconnected residue, it outputs an error message and terminates with an error return value.

5.7 generate_rasmol_script

Usage: `generate_rasmol_script mol2_file`

This program reads a mol2 file and outputs a Rasmol script called `<mol2_file>.ras` (with `.ras` having replaced the original `.mol2` file extension). This script can be invoked in Rasmol with the command `script <mol2_file>.ras`. It then reads the mol2 file for the molecule, colors rotatable bonds in yellow, and shows H-bond donors/acceptors as large spheres (donors in blue, acceptors in red, and doneptors in cyan). This script is helpful to visualize SLIDE's interpretation of flexibility and chemistry for a molecule.

5.8 Perl Scripts

5.8.1 binding_site.pl

Usage: (called from web browser)
`http://web.server/cgi-bin/binding_site.pl?SLIDE_DATA_DIR!target!template!database!ligand_name`

This script sends a file in PDB format that shows the ligand docked into the binding site. In the header of the output it specifies as the MIME-type `chemical/x-pdb`, so that the receiving client can start an application like Rasmol to show this PDB file. To facilitate coloring, chains for each molecule have been designated as follows: chain P, the target protein; chain R, side-chains of the target protein that were rotated in their non-rotated conformations; chain T, side-chains of the target protein that were rotated in their rotated conformations; chain K, the known ligand molecule docked in the known conformation; and chain L, the docked ligand. The script will look for the protein binding site first as

SLIDE_DATA_DIR/target/template/in/binding_site.pdb, and if this is not found, it will load SLIDE_DATA_DIR/target/template/in/target.rad. The ligand molecule must exist in PDB format in the target/template/in directory as ligand.pdb.

5.8.2 binding_site_residues.pl

Usage: binding_site_residues.pl target template distance

This script reads the PDB-file target.pdb, which should be located in the base directory for this target protein, which is $SLIDE_DATA_DIR/target, and the template file, and outputs a PDB file target.rad in the input directory for this template, which includes only those residues of the target protein that have at least one atom within the distance distance from any template point. These residues are the only ones that are considered during collision checks when docking the ligand. A value of 9.0 or 10.0 for the distance is reasonable and yields a file with usually around 700 atoms. Smaller values reduce the number of atoms and thus the computation time for all intermolecular and intra-protein bump checks. However, it is actually more than just one shell of binding-side residues necessary to ensure that no rotations of protein side chains are applied, that generate collisions with residues further away from the binding site.

5.8.3 binding_site_waters.pl

Usage: binding_site_waters.pl pdb_file target template distance

This script extracts all water molecules from the PDB file pdb_file which are within distance Å from any of the template points. These water molecules are then stored in the file $SLIDE_DATA_DIR/<target>/<template>/in/waters.pdb and read by SLIDE during screening.

5.8.4 check_number_of_atoms_mol2.pl

Usage: check_mol2_files.pl directory min_atoms max_atoms

This Perl script checks all mol2 files in directory <directory> and deletes those that have less than <min_atom> or more than <max_atom> atoms.

5.8.5 create_csd_set_data.pl

Usage: create_csd_set_data.pl jnl_file

This script takes the jnl file output by `quest` during a CSD request and extracts the information about the compound names, publication titles, and authors into files `A.data` to `Z.data` based on the first letter of the reference codes. These files are read by the script `create_global_csd_results_table.pl` when composing the global results table for a CSD screening run, and by the script `create_html_files_csd.pl`, which composes the HTML data sheets for the top ligands. The variable `$CSD_SET_DATA_DIR` in the script `create_global_csd_results_table.pl` has to be customized to point to a directory that holds the output-files `A.data` to `Z.data` generated by `create_csd_set_data.pl`.

5.8.6 create_global_csd_results_table.pl

Usage: `create_global_csd_results_table.pl target template number`

This Perl script outputs a table of the top-`<number>` CSD ligands that SLIDE found for target `target` using binding-site template `template`. It is called by the Perl-script `create_html_files_csd.pl`. If the compound number for all CSD compounds has to be listed in this table, the script `create_csd_set_data.pl` (page 42) has to be run to extract the corresponding information from the jnl file that was generated during the extraction of the compound subset from the CSD when setting up the screening database. The variable `$CSD_SET_DATA_DIR` in the script `create_global_csd_results_table.pl` has to be customized to point to the directory that includes the data files.

5.8.7 create_global_genericDB_results_table.pl

Usage: `create_global_genericDB_results_table.pl target template database number`

This Perl script is similar to the `create_global_csd_results_table.pl`, except it works for any database given. The script outputs a table of the top-`<number>` ligands that SLIDE found for target `target` using binding-site template `template` when screening the database `database`. It is called by the Perl-script `create_html_files_genericDB.pl`.

5.8.8 create_html_files_csd.pl

Usage: `create_html_files_csd.pl target template number html_dir`

This script sets up a set of HTML files that enable browsing through the potential ligands SLIDE has found when screening for protein `target` using template `template`. `number` is the number of top ligands to be shown. `html_dir` is the directory that holds the subdirectories

with the HTML files for the different screening experiments. It is usually located in the directory structure of the HTTP server, which serves these pages over the web.

The script `create_html_files_csd.pl` first extracts all information for the potential ligands that SLIDE found from the mol2 files stored in the output ligand directory (see Section 3.5, page 25). It combines this information in a table that lists the top ligands up to the rank specified in the command-line parameter `number` when calling this script. It then starts `quest` to request the graphical representation for each molecule from the CSD. For each of the top ligands a data sheet, like the one shown in Figure 5.2, is generated and these files together with GIF files showing the graphical representation for the molecule are stored in the directory specified by the variable `$HTML_DIR` in the header of the script.

5.8.9 create_html_files_genericDB.pl

Usage: `create_html_files_genericDB.pl target template number database html_dir`

This Perl script is similar to `create_html_files_csd.pl`, except it can be run on a generic database. It sets up a set of HTML files that enable browsing through the potential ligands SLIDE has found when screening for protein `target` using template `template` on database `database`. `number` is the number of top ligands to be shown. `html_dir` is the directory that holds the subdirectories with the HTML files for the different screening experiments. It is usually located in the directory structure of the HTTP server, which serves these pages over the web. A significant difference between `create_html_files_csd.pl` and this script is that this script does not serve images of the potential ligands, as there is no 2-dimensional information stored in a generic ligand database.

5.8.10 create_pts_files.pl

Usage: `create_pts_files.pl`

This script is the generic version of the Perl script `create_pts_files_csd.pl`. It computes the interaction centers in all mol2 files in all subdirectories in the current directory and lists them in pts-files together with the filenames of the corresponding mol2 files. The program `compute_interaction_centers` is run for each mol2 file. If a molecule has less than three interaction centers, then it will be ignored.

SLIDE's ligands for 2apr (template random_key_waters)

Table – Rank 0 – Rank 2 – Binding site – Known ligand

Rank 1: CAKHAX

Score: 49.2
H-Bonds: 6 (3 water-mediated), lost 3 intra-target/water h-bonds
Hphob: 69.9 (100 % of C's buried)
Rotations: 1 ligand and 14 target rotations (10 MFT iterations)
Remaining collisions: 3 (0.434 A total overlap)
Waters: 8 conserved, 1 displaced, 1 displaced by polar atom, 0 translations
Number of binding modes: 1
Name: Manicol

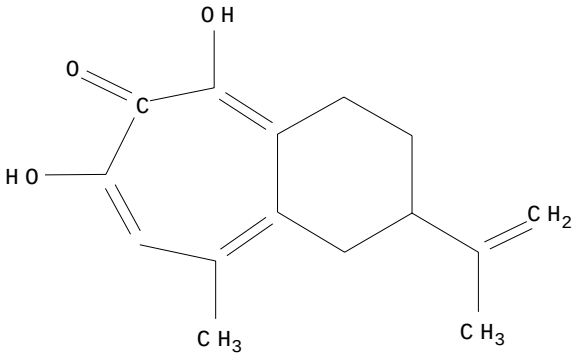


Figure 5.2: This is an example for a data sheet of the top-ranked CSD ligand found by SLIDE for an aspartic protease (PDB 2apr). The second line consists of links to other pages (a table listing all ligands, the previous and the next ligand in the ranking) and two links that request PDB files via CGI scripts, which show the ligand docked into the binding site and in its orientation relatively to a known ligand. SLIDE's rank is listed together with the CSD reference code for this molecule. The value of the scoring function is listed, which is computed based on the number of hydrogen bonds, including water-mediated interactions and a penalty term based on lost hydrogen bonds due to protein side-chain rotations and displacement of water molecules upon ligand docking. The hydrophobic complementarity is listed together with the percentage of ligand carbons that is in contact with any protein atom. The number of rotations in both molecules is given together with the number of iterations for the mean-field theory based computations to resolve collisions; the number of remaining collisions and their amount is also listed, if applicable. For the binding-site waters the number of conserved and displaced waters is given together with the total number of translations of any water molecule when resolving collisions with ligand or protein atoms. The last information is the number of different binding modes suggested for this ligand and its name, as listed in the CSD.

5.8.11 create_pts_files_csd.pl

Usage: `create_pts_files_csd.pl`

This script computes the interaction centers in all mol2 files in subdirectories `A/` to `Z/` in the current directory and lists them in files `A/A.pts` to `Z/Z.pts` together with the filenames of the corresponding mol2 files. The program `compute_interaction_centers` is run for each mol2 file. If a molecule has less than three interaction centers, then it will be ignored. At the end, this script creates the database definition file `$SLIDE_DIR/databases/csd.db` that defines the CSD screening database in the global SLIDE installation.

5.8.12 cut_mol2_csd.pl

Usage: `cut_mol2_csd.pl mol2_file`

This Perl script takes the output mol2-file `<mol2_file>` generated by `quest` from the CSD and extracts single mol2-files called `<refcode>.mol2` named by the reference-code `<refcode>` for the corresponding CSD entry.

5.8.13 extract_mol2_residues.pl

Usage: `extract_mol2_residues.pl directory`

This Perl script checks for all mol2 files in directory `<directory>`, if they contain more than one residue. If a mol2-file `<file>.mol2` contains more than one residue, then the residues are extracted into separate mol2 files called `<file>_<residue>.mol2` and stored in the same directory. The original mol2 file is deleted.

5.8.14 filter_ligands.pl

Usage: `filter_ligands.pl target template db cutoff`

This script checks SLIDE's score for all ligands found in database `db` when screening for target `target` with template `template`. The command-line parameter `cutoff` defines the lower bound for the score, all ligands with the corresponding target and water files in the directories `<db>_targets` and `<db>_waters`, respectively, that have a lower score than `cutoff` are deleted.

5.8.15 known_ligand.pl

Usage: (called from web browser)
`http://web.server/cgi-bin/known_ligand.pl?SLIDE_DATA_DIR!target!template!database!ligand_name`

This script sends a file in PDB format that shows the ligand docked in SLIDE's binding mode relative to the known ligand that is stored in the input directory for the screening run in the file `ligand.pdb`. In the header of the output it specifies as the MIME-type `chemical/x-pdb`, so that the receiving client can start an application like Rasmol to show this PDB file. The molecules are assigned chain identifiers to facilitate coloring and rendering as follows: chain K, the known ligand, and chain L, the docked ligand. The ligand molecule must exist in PDB format in the `SLIDE_DATA_DIR/target/template/in` directory as `ligand.pdb`.

5.8.16 pdb_to_template.pl

This script reads the file `template.pdb` in the current directory and outputs a template-file `template`, which is used by SLIDE to identify the binding-site template. The input file contains a set of water molecules, which are centered at the position of template points. Based on the temperature factor for each of these atoms, the type for each template point is selected. When generating a binding-site template by either the program `average_template` or `unbiased_template`, both files are created automatically. The purpose of the script `pdb_to_template.pl` is to generate a regular template file out of a PDB file `template.pdb`, in which atoms were moved when interactively changing the template, e.g, in a program for crystallographic fitting of structures.

5.8.17 process_csd_set.pl

This Perl script processes a set of compounds selected from the CSD. Before this script, the script `cut_mol2_csd.pl` has to be run on the CSD output file generated by `quest`, to extract the CSD entries in mol2 files and store them in subdirectories according to their reference codes (see Section 4.1, page 29). This script calls the Perl-scripts `extract_mol2_residues.pl` and `check_number_of_atoms_mol2.pl` to extract residues from all mol2 files in subdirectories `A/` to `Z/` and to delete all files describing molecules with less than three or more than 200 atoms using the Perl-script `check_number_of_atoms_mol2.pl`.

5.8.18 renum_pdb_atoms.pl

Usage: `renum_pdb_atoms.pl pdb_file type`

This script renumbers the atoms in the PDB file `pdb_file` and is especially used for preparing the binding site and ligand file for the web interface. The command-line parameter `type` has to be one of the keywords `TARGET` or `LIGAND`, when renumbering the atoms for the binding site or the ligand, respectively. The main purpose is to assign a chain ID to these files so that they can be easily distinguished in Rasmol when using the web interface.

5.8.19 show_ligands.pl

Usage: `show_ligands.pl target template database`

This script generates a Rasmol script that can be used to browse through the ligands that SLIDE found in the database `database` for protein `target` using binding-site template `template`. The script reads in all corresponding mol2 files based on their ranking, shows the ligands in rasmol, and outputs the score and other relevant information in the terminal window in which Rasmol was started.

The script `show_ligands.pl` is an alternative to the web interface. The web interface is certainly a more convenient way to browse through the results, however, in cases where no simple graphical representation for the 2D structure of the ligands is provided with the database, the script `show_ligands.pl` is the only way to visualize the structures for a larger set of potential ligands.

5.8.20 slide_setup.pl

Usage: `slide_setup.pl target template [database]`

This script creates all subdirectories in the output directory specified by the environment variable `$SLIDE_DATA_DIR`, which are will hold the input/output data associated with a screening experiment. The database specifier is optional, and this script can also be used to add subdirectories to the directory structure of an existing screening experiment, for example, when screening a new database.

5.8.21 template_to_pdb.pl

This script reads the file `template` in the current directory, which lists the interaction centers of the binding-site template for a screening experiment. It outputs a file `template.pdb`, which is

a PDB file that includes water molecules at the positions of the template points listed in the input file.

Appendix A

Version History

- **Version 1.1** – Release Date: September 2000.
  - Bug fixes and changes to source code to facilitate future changes, but which have no functional effect in this release
  - Change in hydrophobic point definition from one in an area of average hydrophobic character to one having a required minimum number of hydrophobic atoms and no more than a maximum number of hydrophilic atoms
  - Addition of ability to define 8 corners of a box to define a binding-site to place points for template generation (Previous versions only used a maximum and minimum for each of coordinate.)
  - Addition of the ability to handle differing clustering thresholds for hydrophobic and hydrogen bond forming template points
  - Addition of key point inclusion in template2pdb script and addition of pdb2template script.
- **Version 1.0** – Release Date: September 1999

  Initial SLIDE release.

Appendix B

Known Problems

- Some of the output `mol2` files may have formatting problems that cause incorrect additional bonds to be generated in some molecular graphics programs. The correct bonds are being formed and the SLIDE is using the correct bond information, but the visualization may not be correct.
- SLIDE expects the number of atoms listed in a `mol2` file header to match the number of actual atoms in the file. If these numbers do not match, an error will occur.

Bibliography

[1] I. D. Kuntz, E. C. Meng, and B. K. Shoichet. Structure–based molecular design. *Accounts of Chemical Research*, 27(5):117–123, 1994.

[2] M. L. Raymer, P. C. Sanschagrin, W. F. Punch, S. Venkataraman, E. D. Goodman, and L. A. Kuhn. Predicting conserved water-mediated and polar ligand interactions in proteins using a k-nearest-neighbors genetic algorithm. *Journal of Molecular Biology*, 265:445–464, 1997.

[3] M. D. Eldridge, C. W. Murray, T. R. Auton, G. V. Paolini, and R. P. Mee. Empirical scoring functions: I. The development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. *Journal of Computer-Aided Molecular Design*, 11:425–445, 1997.

[4] A.-J. Li and R. Nussinov. A set of van der Waals and coulombic radii of protein atoms for molecular and solvent-accessible surface calculation, packing evaluation, and docking. *Proteins: Structure, Function, and Genetics*, 32:111–127, 1998.

[5] P. Koehl and M. Delarue. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. *Journal of Molecular Biology*, 239:249–275, 1994.

[6] R. W. W. Hooft, C. Sander, and G. Vriend. Positioning hydrogen atoms by optimizing hydrogen-bond networks in protein structures. *Proteins: Structure, Function, and Genetics*, 26:363–376, 1996.

[7] S. M. Habermann and K. P. Murphy. Energetics of hydrogen bonding in proteins: A model compound study. *Protein Science*, 5:1229–1239, 1996.

[8] L. A. Kuhn, C. A. Swanson, M. E. Pique, J. A. Tainer, and E. D. Getzoff. Atomic and residue hydrophilicity in the context of folded protein structures. *Proteins: Structure, Function, and Genetics*, 23:536–547, 1995.

[9] I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, and T. E. Ferrin. A geometric approach to macromolecule-ligand interactions. *Journal of Molecular Biology*, 161:269–288, 1982.

[10] B. K. Shoichet and I. D. Kuntz. Matching chemistry and shape in molecular docking. *Protein Engineering*, 6(7):723–732, 1993.

[11] A. R. Leach and I. D. Kuntz. Conformational analysis of flexible ligands in macromolecular receptor sites. *Journal of Computational Chemistry*, 13(6):730–748, 1992.

Index