rDock
Reference Guide

rDock Development Team

August 27, 2015

Contents

1 Preface

It is intended to develop this document into a full reference guide for the rDock platform, describing the software tools, parameter files, scoring functions, and search engines. The reader is encouraged to cross-reference the descriptions with the corresponding source code files to discover the finer implementation details.

2 Acknowledgements

Third-party source code. Two third-party C++ libraries are included within the rDock source code, to provide support for specific numerical calculations. The source code for each library can be distributed freely without licensing restrictions and we are grateful to the respective authors for their contributions.

- Nelder-Meads Simplex search, from Prof. Virginia Torczon's group, College of William and Mary, Department of Computer Science, VA.
  (http://www.cs.wm.edu/va/software/)
- Template Numerical Toolkit, from Roldan Pozo, Mathematical and Computational Sciences Division, National Institute of Standards and Technology
  (http://math.nist.gov/tnt/)

3 Introduction

The rDock platform is a suite of command-line tools for high-throughput docking and virtual screening. The programs and methods were developed and validated from 1998 to 2002 at RiboTargets (more recently, Vernalis) for proprietary use. The original program (RiboDock) was designed for high-throughput virtual screening of large ligand libraries against RNA targets, in particular the bacterial ribosome. Since 2002 the programs have been substantially rewritten and validated for docking of drug-like ligands to protein and RNA targets. A variety of experimental restraints can be incorporated into the docking calculation, in support of an integrated Structure-Based Drug Design process. In 2006, the software was licensed to the University of York for maintenance and distribution and, in 2012, Vernalis and the University of York agreed to release the program as Open Source software.

rDock is licensed under GNU-LGPL version 3.0 with support from the University of Barcelona - rdock.sourceforge.net.

4 Configuration

Before launching rDock, make sure the following environment variables are defined. Precise details are likely to be site-specific.

- **RBT_ROOT environment variable**: should be defined to point to the rDock installation directory.
- **RBT_HOME environment variable**: is optional, but can be defined to to point to a user project directory containing rDock input files and customised data files.
- **PATH environment variable**: $RBT_ROOT/bin should be added to the $PATH environment variable.
- **LD_LIBRARY_PATH**. $RBT_ROOT/lib should be added to the $LD_LIBRARY_PATH environment variable.

Input file locations. The search path for the majority of input files for rDock is:

- Current working directory
- $RBT_HOME, if defined
- The appropriate subdirectory of $RBT_ROOT/data/. For example, the default location for scoring function files is $RBT_ROOT/data/sf/.

The exception is that input ligand SD files are always specified as an absolute path. If you wish to customise a scoring function or docking protocol, it is sufficient to copy the relevant file to the current working directory or to $RBT_HOME, and to modify the copied file.

Launching rDock. For small scale experimentation, the rDock executables can be launched directly from the command line. However, serious virtual screening campaigns will likely need access to a compute farm. In common with other docking tools, rDock uses the embarrassingly parallel approach to distributed computing. Large ligand libraries are split into smaller chunks, each of which is docked independently on a single machine. Docking jobs are controlled by a distributed resource manager (DRM) such as Condor or SGE.

5 Cavity mapping

Virtual screening is very rarely conducted against entire macromolecules. The usual practice is to dock small molecules in a much more confined region of interest. rDock makes a clear distinction between the region the ligand is allowed to explore (known here as the docking site), and the receptor atoms that need to be included in order to calculate the score correctly. The former is controlled by the cavity mapping algorithm, whilst the latter is scoring function dependent as it depends on the distance range of each component term (for example, vdW range >> polar range). For this reason, it is usual practice with rDock to prepare intact receptor files (rather than truncated spheres around the region of interest), and to allow each scoring function term to isolate the relevant receptor atoms within range of the docking site.
rDock provides two methods for defining the docking site:

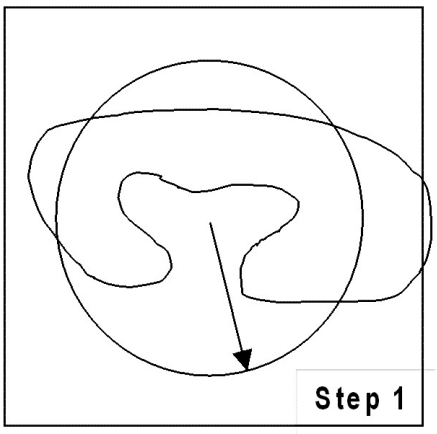
- Two sphere method
- Reference ligand method

Note All the keywords found in capital letters in following cavity mapping methods explanation (e.g. RADIUS), make reference to the parameters defined in *prm* rDock configuration file. For more information, go to section 8.4 - Cavity mapping on page 25.

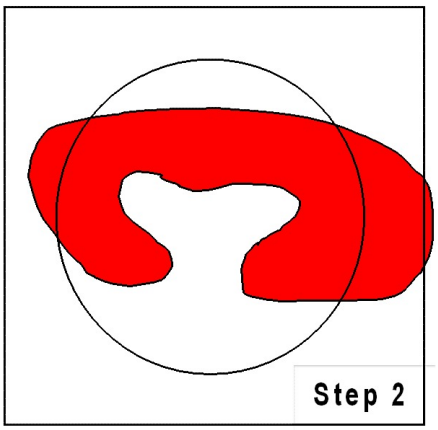
5.1 Two sphere method

The two sphere method aims to find cavities that are accessible to a small sphere (of typical atomic or solvent radius) but are inaccessible to a larger sphere. The larger sphere probe will eliminate flat and convex regions of the receptor surface, and also shallow cavities. The regions that remain and are accessible to the small sphere are likely to be the nice well defined cavities of interest for drug design purposes.

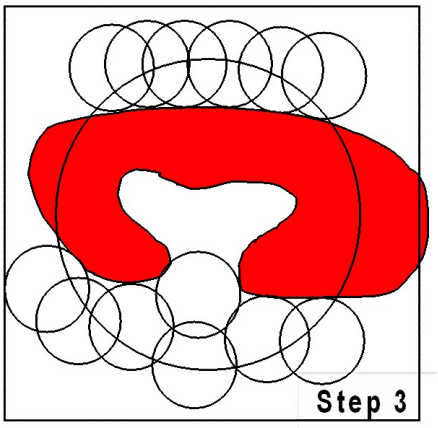
1. A grid is placed over the cavity mapping region, encompassing a sphere of radius=RADIUS, center=CENTER. Cavity mapping is restricted to this sphere. All cavities located will be wholly within this sphere. Any cavity that would otherwise protrude beyond the cavity mapping sphere will be truncated at the periphery of the sphere.



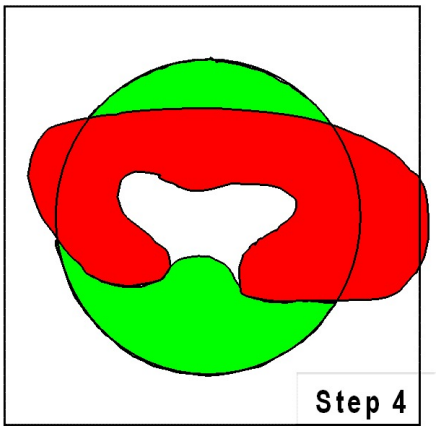
2. Grid points within the volume occupied by the receptor are excluded (coloured red). The radii of the receptor atoms are increased temporarily by VOL_INCR in this step.



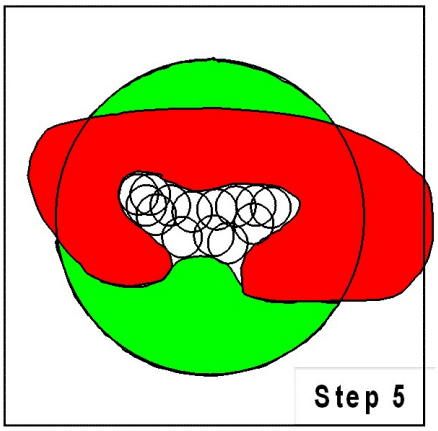
3. Probes of radii LARGE_SPHERE are placed on each remaining unallocated grid point and checked for clashes with receptor excluded volume. To eliminate edge effects, the grid is extended beyond the cavity mapping region by the diameter of the large sphere (for this step only). This allows the large probe to be placed on grid points outside of the cavity mapping region, yet partially protrude into the cavity mapping region.



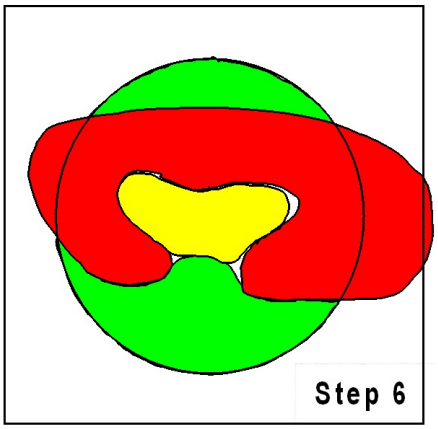
4. All grid points within the cavity mapping region that are accessible to the large probe are excluded (coloured green).



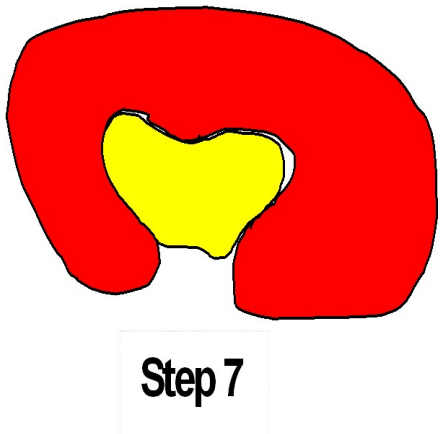
5. Probes of radii SMALL_SPHERE are placed on each remaining grid point and checked for clashes with receptor excluded volume (red) or large probe excluded volume (green)



6. All grid points that are accessible to the small probe are selected (yellow).



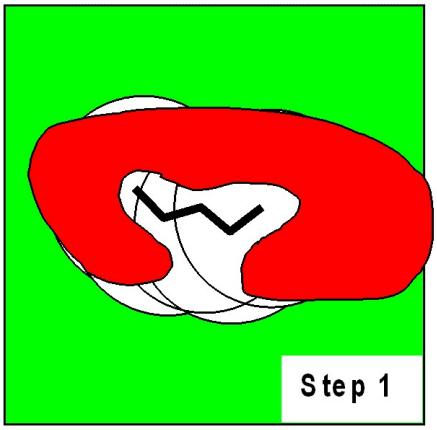
7. The final selection of cavity grid points is divided into distinct cavities (contiguous regions). In this example only one distinct cavity is found. User-defined filters of MIN_VOLUME and MAX_CAVITIES are applied at this stage to select a subset of cavities if required. Note that the filters will accept or reject intact cavities only.



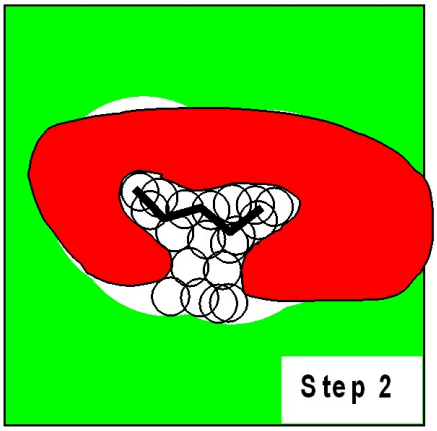
5.2 Reference ligand method

The reference ligand method provides a much easier option to define a docking volume of a given size around the binding mode of a known ligand, and is particularly appropriate for large scale automated validation experiments.

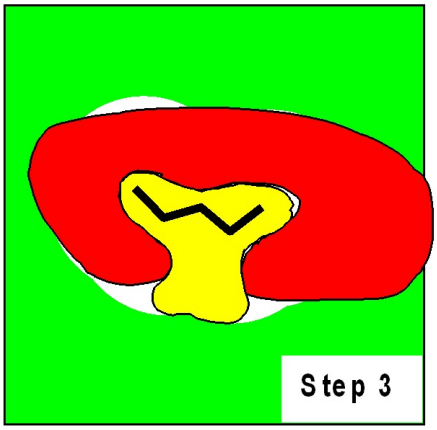
1. Reference coordinates are read from REF_MOL. A grid is placed over the cavity mapping region, encompassing overlapping spheres of radius=RADIUS, centered on each atom in REF_MOL. Grid points outside of the overlapping spheres are excluded (coloured green). Grid points within the volume occupied by the receptor are excluded (coloured red). The vdW radii of the receptor atoms are increased by VOL_INCR in this step.



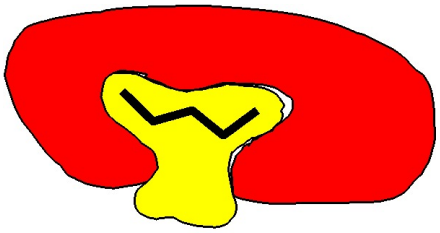
2. Probes of radii SMALL_SPHERE are placed on each remaining grid point and checked for clashes with red or green regions.



3. All grid points that are accessible to the small probe are selected (yellow).



4. The final selection of cavity grid points is divided into distinct cavities (contiguous regions). In this example only one distinct cavity is found. User-defined filters of MIN_VOLUME and MAX_CAVITIES are applied at this stage to select a subset of cavities if required. Note that the filters will accept or reject intact cavities only.



Step 4

6 Scoring function reference

6.1 Component Scoring Functions

The rDock master scoring function (S_{total}) is a weighted sum of intermolecular (S_{inter}), ligand intramolecular (S_{intra}), site intramolecular (S_{site}), and external restraint terms ($S_{restraint}$) (Equation 1). S_{inter} is the main term of interest as it represents the protein-ligand (or RNA-ligand) interaction score (Equation 2). S_{intra} represents the relative energy of the ligand conformation (Equation 3). Similarly, S_{site} represents the relative energy of the flexible regions of the active site (Equation 4). In the current implementation, the only flexible bonds in the active site are terminal OH and NH3+ bonds. $S_{restraint}$ is a collection of non-physical restraint functions that can be used to bias the docking calculation in several useful ways (Equation 5).

$$S^{total} = S^{inter} + S^{intra} + S^{site} + S^{restraint} \tag{1}$$

$$\begin{aligned} S^{inter} &= W^{inter}_{vdw} \cdot S^{inter}_{vdw} + W^{inter}_{polar} \cdot S^{inter}_{polar} + W^{inter}_{repul} \cdot S^{inter}_{repul} + W^{inter}_{arom} \cdot S^{inter}_{arom} + W_{solv} \cdot S_{solv} + \\ &+ W_{rot} \cdot N_{rot} + W_{const} \qquad (2) \\ S^{intra} &= W^{intra}_{vdw} \cdot S^{intra}_{vdw} + W^{intra}_{polar} \cdot S^{intra}_{polar} + W^{intra}_{repul} \cdot S^{intra}_{repul} + W^{intra}_{dihedral} \cdot S^{intra}_{dihedral} \qquad (3) \\ S^{site} &= W^{site}_{vdw} \cdot S^{site}_{vdw} + W^{site}_{polar} \cdot S^{site}_{polar} + W^{site}_{repul} \cdot S^{site}_{repul} + W^{site}_{dihedral} \cdot S^{site}_{dihedral} \qquad (4) \\ S^{restraint} &= W_{cavity} \cdot S_{cavity} + W_{tether} \cdot S_{tether} + W_{nmr} \cdot S_{nmr} + W_{ph4} \cdot S_{ph4} \qquad (5) \end{aligned}$$

S_{inter}, S_{intra}, and S_{site} are built from a common set of constituent potentials, which are described below. The main changes to the original RiboDock scoring function [ref] are:

i the replacement of the crude steric potentials (S_{lipo} and S_{rep}) with a true van der Waals potential, S_{vdW}

ii the introduction of two generalised terms for all short range attractive (S_{polar}) and repulsive (S_{repul}) polar interactions

iii the implementation of a fast weighed solvent accessible surface area (WSAS) solvation term

iv the addition of explicit dihedral potentials

6.1.1 van der Waals potential

We have replaced the S_{lipo} and S_{rep} empirical potentials used in RiboDock with a true vdW potential similar to that used by GOLD [ref]. Atom types and vdW radii were taken from the Tripos 5.2 force field and are listed in the Appendix Section (page 45, table 13.1). Energy well depths are switchable between the original Tripos 5.2 values and those used by GOLD, which are calculated from the atomic polarisability and ionisation potentials of the atom types involved. Additional atom types were created for carbons with implicit hydrogens, as the Tripos force field uses an all-atom representation. vdW radii for implicit hydrogen types are increased by 0.1 Å for each implicit hydrogen, with well depths unchanged. The functional form is switchable between a softer 4-8 and a harder 6-12 potential. A quadratic potential is used at close range to prevent excessive energy penalties for atomic clashes. The potential is truncated at longer range ($1.5 \cdot r_{min}$, the sum of the vdW radii).

The quadratic potential is used at repulsive energies between e_{cutoff} and e_0, where e_{cutoff} is defined as a multiple of the energy well depth (e_{cutoff} = ECUT * e_{min}), and e_0 is the energy at zero separation, defined as a multiple of e_{cutoff} (e_0 = E0 * e_{cutoff}). ECUT can vary between 1 and 120 during the docking search (see below)[ref to ga section], whereas E0 is fixed at 1.5.

6.1.2 Empirical attractive and repulsive polar potentials

We continue to use an empirical Bohm-like potential to score hydrogen-bonding and other short-range polar interactions. The original RiboDock polar terms (S_{H-bond}, $S_{posC-acc}$, $S_{acc-acc}$, $S_{don-don}$) are generalised and condensed into two scoring functions, S_{polar} and S_{repul} (Equations 6 and 7, also taking into account equations 8 to 13), which deal with attractive and repulsive interactions respectively. Six types of polar interaction centres are considered: hydrogen bond donors (DON), metal ions (M+), positively charged carbons (C+, as found at the centre of guanidinium, amidinium and imidazole groups), hydrogen

bond acceptors with pronounced lone pair directionality (ACC_LP), acceptors with in-plane preference but limited lone-pair directionality (ACC_PLANE), and all remaining acceptors (ACC). The ACC_LP type is used for carboxylate oxygens and O_{sp2} atoms in RNA bases, with ACC_PLANE used for other Osp2 acceptors. This distinction between acceptor types was not made in RiboDock, in which all acceptors were implicitly of type ACC.

$$S_{polar} = \sum_{IC1-IC2} f_1(|\Delta R_{12}|) \cdot ANG_{IC1} \cdot ANG_{IC2} \cdot f_2(IC1) \cdot f_2(IC2) \cdot f_3(IC1) \cdot f_3(IC2) \quad (6)$$

$$S_{repul} = \sum_{IC1-IC2} f_1(\Delta R_{12}) \cdot ANG_{IC1} \cdot ANG_{IC2} \cdot f_2(IC1) \cdot f_2(IC2) \cdot f_3(IC1) \cdot f_3(IC2) \quad (7)$$

$$f_1(\Delta X) = \begin{cases} 1 & \Delta X \leq \Delta X_{Min} \\ 1 - \frac{\Delta X - \Delta X_{Min}}{\Delta X_{Max} - \Delta X_{Min}} & \Delta X_{Min} < \Delta X \leq \Delta X_{Max} \\ 0 & \Delta X > \Delta X_{Max} \end{cases} \quad (8)$$

$$f_2(i) = sgn(i)(1 + 0.5|c_i|) \quad (9)$$

$$sgn(i) = \begin{cases} -1 & ACC, ACC\_LP, ACC\_PLANE \\ +0.5 & C+ \\ +1.0 & DON, M+ \end{cases} \quad (10)$$

$$c_i = \text{formal charge on primary atom of interaction centre i} \quad (11)$$

$$f_3(\Delta X) = \begin{cases} \sqrt{\frac{N_i}{25}} & \text{Macromolecular interaction centres} \\ 1 & \text{Ligand interaction centres} \end{cases} \quad (12)$$

$$N_i = \text{number of non-hydrogen macromolecule atoms within 5Å radius of primary atom of interaction centre i} \quad (13)$$

Individual interaction scores are the product of simple scaling functions for geometric variables, formal charges and local neighbour density. The scaling functions themselves, and the formal charge assignment method, are retained from RiboDock. Metals are assigned a uniform formal charge of +1. C+ is considered to be a weak donor in this context and scores are scaled by 50 % relative to conventional donors by the assignment of sgn(i)=0.5 in Equation 8. Pseudo-formal charges are no longer assigned to selected RNA base atoms. The geometric functions minimally include an interaction distance term, with the majority also including angular terms dependent on the type of the interaction centres. Geometric parameters and the angular functions are summarised in Appendix Section (page 46, tables 13.2 and 13.3 respectively).

The most notable improvements to RiboDock are that attractive (hydrogen bond and metal) interactions with ACC_LP and ACC_PLANE acceptors include terms for φ and θ (as defined in ref 3) to enforce the relevant lone pair directionality. These replace the α_{ACC} dependence, which is retained for the ACC acceptor type. No distinction between acceptor types is made for attractive interactions with C+ carbons, or for repulsive interactions between acceptors. In these circumstances all acceptors are treated as type ACC. Such C+-ACC interactions, which in RiboDock were described by only a distance function, ($S_{posC-acc}$) now include angular functions around the carbon and acceptor groups. Repulsive interactions between donors, and between acceptors, also have an angular dependence. This allows a stronger weight, and a longer distance range, to be used to penalise disallowed head-to-head interactions without forbidding allowable contacts. One of the issues in RiboDock was that it was not possible to include neutral acceptors in the acceptor-acceptor repulsion term with a simple distance function.

6.1.3 Solvation potential

The desolvation potential in rDock combines a weighted solvent accessible surface area approach [WSAS[ref4]] with a rapid probabilistic approximation to the calculation of solvent accessible surface areas [ref5] based on pairwise interatomic distances and radii (Equation 14, taking into account equations 15 to 20).

$$S_{solv} = (\Delta G_{WSAS}^{site,bound} - \Delta G_{WSAS}^{site_0,unbound}) + (\Delta G_{WSAS}^{ligand,bound} - \Delta G_{WSAS}^{ligand_0,unbound}) \quad (14)$$

$$
\begin{aligned}
r_s &= 0.6\text{Å} && (15)\\
p_{ij} &= \begin{cases} 0.8875 & \text{1-2 intramolecular connections} \\ 0.3516 & \text{1-3 intramolecular connections} \\ 0.3156 & \text{1-4 intramolecular connections and above} \\ 0.3156 & \text{intermolecular interactions} \end{cases} && (16)\\
S_i &= 4\pi(r_i + r_s)^2 && (17)\\
b_{ij} &= \pi(r_i + r_s)(r_j + r_i + 2r_s - d)\left(1 - \tfrac{r_j - r_i}{d}\right) && (18)\\
A_i &= S_i \textstyle\prod_j 1 - \frac{p_i p_{ij} b_{ij}}{S_i} && (19)\\
\Delta G_{WSAS} &= \textstyle\sum_{i=1}^{n_i} w_i A_i && (20)
\end{aligned}
$$

The calculation is fast enough therefore to be used in docking. We have redefined the solvation atom types compared to the original work[4] and recalibrated the weights against the same training set of experimental solvation free energies in water (395 molecules). The total number of atom types (50, including 6 specifically for ionic groups and metals) is slightly lower than in original work (54). Our atom types reflect the fact that rDock uses implicit non-polar hydrogens. The majority of types are a combination of hybridisation state and the number of implicit or explicit hydrogens. All solvation parameters are listed in Appendix Section (page 46, table 13.4).

S_{solv} is calculated as the change in solvation energy of the ligand and the docking site upon binding of the ligand. The reference energies are taken from the initial conformations of the ligand and site (as read from file) and not from the current pose under evaluation. This is done to take into account any changes to intramolecular solvation energy. Strictly speaking the intramolecular components should be reported separately under S_{intra} and S_{site}but this is not done for reasons of computational efficiency.

6.1.4 Dihedral potential

Dihedral energies are calculated using Tripos 5.2 dihedral parameters for all ligand and site rotatable bonds. Corrections are made to account for the missing contributions from the implicit non-polar hydrogens.

6.2 Intermolecular scoring functions under evaluation

6.2.1 Training sets

We constructed a combined set of protein-ligand and RNA-ligand complexes for training of rDock. Molecular data files for the protein-ligand complexes were extracted from the downloaded CCDC/Astex clean-list[ref6] and used without substantive modification. The only change was to convert ligand MOL2 files to MDL SD format using Corina [ref], leaving the coordinates and protonation states intact.

Protein MOL2 files were read directly. The ten RNA-ligand NMR structures from the RiboDock validation set were supplemented with five RNA-ligand crystal structures (1f1t, 1f27, 1j7t, 1lc4, 1mwl) prepared in a similar way. All 15 RNA-ligand structures have measured binding affinities.

58 complexes (43 protein-ligand and 15 RNA-ligand) were selected for the initial fitting of component scoring function weights. Protein-ligand structures were chosen (of any X-ray resolution) that had readily available experimental binding affinities [ref 7].102 complexes were used for the main validation of native docking accuracy for different scoring function designs, consisting of 87 of the 92 entries in the high-resolution (R<2Å) clean-list (covalently bound ligands removed - 1aec, 1b59, 1tpp, 1vgc, 4est), and the 15 RNA-ligand complexes.

6.2.2 Scoring Functions Design

Component weights (W) for each term in the intermolecular scoring function (S_{inter}) were obtained by least squares regression of the component scores to ΔG_{bind} values for the binding affinity training set described above (58 entries). Each ligand was subjected first to simplex minimisation in the docking site, starting from the crystallographic pose, to relieve any minor non-bonded clashes with the site. The scoring function used for minimisation was initialised with reasonable manually assigned weights. If the fitted weights deviated significantly from the initial weights the procedure was repeated until convergence. Certain weights (W_{repul}, W_{rot}, W_{const}) were constrained to have positive values to avoid non-physical, artefactual models. Note that the presence of W_{rot} and W_{const} in S_{inter} improves the quality of the fit to the binding affinities but does not impact on native ligand docking accuracy.

Ten intermolecular scoring functions were derived with various combinations of terms (Table 6.1). SF0 is a baseline scoring function that has the van der Waals potential only. SF1 adds a simplified polar potential, without f2 (formal charge) and f3 (neighbour density) scaling functions, and with a single acceptor type (ACC) that lacks lone-pair directionality. SF2 has the full polar potential (f2 and f3 scaling functions, ACC, ACC_LP and ACC_PLANE acceptor types) and adds the repulsive polar potential. SF3 has the same functional form as SF2 but with empirical weights in regular use at RiboTargets. SF4 replaces the repulsive polar potential with the WSAS desolvation potential described above. SF5 has the same functional form as SF4 but with empirical weights in regular use at RiboTargets. SF6 combines the repulsive polar and desolvation potentials. SF7 has the same functional form as SF2 and SF3 but with weights for W_{vdW} and W_{polar} taken from SF5. SF8 and SF9 add the crude aromatic term from RiboDock [ref] to SF3 and SF5 respectively. The S_{intra} functional form and weights were held constant, and equivalent to SF3, to avoid any differences in ligand conformational energies affecting the docking results. As the S_{site} scores are calculated simultaneously with S_{site} (for computational reasons) the S_{site} functional form and weights vary in line with S_{inter}. There is surprisingly little variation in correlation coefficient (R) and root mean square error (RMSE) in predicted binding energy over the ten scoring functions (Table 6.1). The best results are obtained with SF4 (R=0.67, RMSE=9.6 kJ/mol).

Table 6.1: Intermolecular scoring function weights under evaluation

SF	$\mathbf{W}_{vdW}$	$\mathbf{W}_{polar}$	$\mathbf{W}_{solv}$	$\mathbf{W}^{a}_{repul}$	$\mathbf{W}_{arom}$	$\mathbf{W}^{a}_{rot}$	$\mathbf{W}^{a}_{const}$	$\mathbf{R}^{c}$	$\mathbf{RMSE}^{c}$
0	1.4	-	-	-	-	0	0	0.62	10.9
1	1.126	2.36	-	-	-	0.217	0	0.64	10.2
2	1.192	2.087	-	2.984	-	0	0	0.63	10.4
3	1.000^{b}	3.400^{b}	-	5.000^{b}	-	0	0	0.59	10.9
4	1.317	3.56	0.449	-	-	0	0	0.67	9.6
5	1.500^{b}	5.000^{b}	0.500^{b}	-	-	0.568	4.782	0.62	10.7
6	1.314	4.447	0.500^{b}	5.000^{b}	-	0	0	0.62	10.4
7	1.500^{b}	5.000^{b}	-	5.000^{b}	-	0.986	12.046	0.55	12.9
8	1.000^{b}	3.400^{b}	-	5.000^{b}	-1.6^{b}	0	0	0.53	11.8
9	1.500^{b}	5.000^{b}	0.500^{b}	-	-1.6^{b}	0.647	5.056	0.58	11.5

a = constrained to be > zero; b = fixed values; c = correlation coefficient (R), and root mean squared error (RMSE) between S_{inter} and ΔG_{bind}, for minimised experimental ligand poses, over binding affinity validation set (58 entries).

6.2.3 Scoring Functions Validation

The ability of the ten intermolecular scoring functions (SF0 to SF9) to reproduce known ligand binding modes was determined on the combined test set of 102 protein-ligand and RNA-ligand complexes. The intra-ligand scoring function (S_{intra}) was kept constant, with component weights equivalent to SF3, and a dihedral weight of 0.5. Terminal OH and NH3 groups on the receptor in the vicinity of the docking site were fully flexible during docking. Ligand pose populations of size N_{pop}=300 were collected for each complex and intermolecular scoring function combination. The population size was increased to N_{pop}=1000 for two of the most promising scoring functions (SF3 and SF5).

Protein-ligand docking accuracy is remarkably insensitive to scoring function changes. Almost half of the ligand binding modes can be reproduced with a vdW potential only (SF0). The addition of a simplified polar potential (SF1) increases the accuracy to over 70% of protein-ligand test cases predicted to within 2Å RMSD. The success rate increases further to 78% with SF3, which has the full attractive and repulsive polar potentials, and empirically adjusted weights relative to SF2. Subsequent changes to the component terms and weights, including the addition of the desolvation potential, have little or no impact on the protein-ligand RMSD metric.

The nucleic acid set shows a much greater sensitivity to scoring function changes. This can in part be explained by the smaller sample size that amplifies the percentage changes in the RMSD metric, but nevertheless the trends are clear. There is a gradual increase in docking accuracy from SF0 (37%) to SF3 (52%), but absolute performance is much lower than for the protein-ligand test set. This level of docking accuracy for nucleic acid-ligand complexes is broadly consistent with the original RiboDock

scoring function, despite the fact that the original steric term (LIPO) has been replaced by a true vdW potential. The introduction of the desolvation potential in place of the empirical repulsive polar potential (in SF4 and SF5) results in a substantial improvement in accuracy, to around 70% of test cases within 2Å RMSD. Subsequent changes (SF6 to SF9) degrade the accuracy. The lower performance of SF7, which has the higher weights for the VDW and POLAR terms taken from SF5 but lacks the desolvation potential, demonstrates that it is the desolvation term itself that is having the beneficial effect, and not merely the reweighting of the other terms. The inclusion of the geometric aromatic term in SF8 and SF9 has a detrimental impact on the performance of SF3 and SF5 respectively.

Overall, SF5 achieves optimum performance across proteins and nucleic acids (76.7% within 2Å RMSD). SF3 (no desolvation potential) and SF5 (with desolvation potential) were selected as the best intermolecular scoring functions. Finally, these two scoring functions, SF3 and SF5, were the ones implemented in rDock with the names of "`dock.prm`" and "`dock_solv.prm`", respectively.

Note In Virtual Screening campaigns, or in experiments where score of different ligands is compared, the best scoring poses for each molecule (as defined by the lowest S_{total} within the sample) are sorted and ranked by S_{inter}. In other words, the contributions to S_{total} from S_{intra}, S_{site} and $S_{restraint}$ are ignored when comparing poses between different ligands against the same target. The rationale for this is that, in particular, the ligand intramolecular scores are not on an absolute scale and can differ markedly between different ligands.

6.3 Code Implementation

Scoring functions for docking are constructed at run-time (by class RbtSFFactory) from scoring function definition files (rDock .prm format). The default location for scoring function definition files is $RBT_ROOT/data/sf/.

The total score is an aggregate of intermolecular ligand-receptor and ligand-solvent interactions (branch SCORE.INTER), intra-ligand interactions (branch SCORE.INTRA), intra-receptor, intra-solvent and receptor-solvent interactions (branch SCORE.SYSTEM), and external restraint penalties (branch SCORE.RESTR).

The SCORE.INTER, SCORE.INTRA and SCORE.SYSTEM branches consist of weighted sums of interaction terms as shown below. Note that not all terms appear in all branches. See the rDock draft paper for more details on the implementation of these terms.

Table 6.2: Scoring function terms and C++ implementation classes

Term	**Description**	**INTER**	**INTRA**	**SYSTEM**
VDW	van der Waals	RbtVdWIdxSF	RbtVdwIntraSF	RbtVdwIdxSF
VDW	van der Waals (grid based)	RbtVdwGridSF	N/A	N/A
POLAR	Attractive polar	RbtPolarIdxSF	RbtPolarIntraSF	RbtPolarIdxSF
REPUL	Repulsive polar	RbtPolarIdxSF	RbtPolarIntraSF	RbtPolarIdxSF
SOLV	Desolvation	RbtSAIdxSF	RbtSAIdxSF	RbtSAIdxSF
DIHEDRAL	Dihedral potential	N/A	RbtDihedralIntraSF	RbtDihedralTargetSF
CONST	Translation/rotational binding entropy penalty	RbtConstSF	N/A	RbtConstSF
ROT	Torsional binding entropy penalty	RbtRotSF	N/A	N/A

Two intermolecular scoring functions (SCORE.INTER branch) have been validated. These are known informally as the standard scoring function and the desolvation scoring function (referred to as SF3 and SF5 respectively in the rDock draft paper). The standard intermolecular scoring function consists of VDW, POLAR and REPUL terms. In the desolvation scoring function, the REPUL term is replaced by a more finely parameterised desolvation potential (SOLV term) based on a Weighted Solvent-Accessible Surface Area (WSAS) model. The ligand intramolecular scoring function (SCORE.INTRA branch) remains constant in both cases, and has the same terms and weights as the standard intermolecular scoring function.

Table 6.3: Scoring function data files

File	Description
RbtInterIdxSF.prm	Intermolecular scoring function definition (standard scoring function, SF3)
RbtInterGridSF.prm	As above, but vdW term uses a precalculated grid
RbtSolvIdxSF.prm	Intermolecular scoring function definition (desolvation scoring function, SF5)
calcgrid_vdw1.prm	vdW term only (ECUT=1), for calculating vdW grid (used by rbcalcgrid)
calcgrid_vdw5.prm	vdW term only (ECUT=5), for calculating vdW grid (used by rbcalcgrid)
Tripos52_vdw.prm	vdW term parameter file Tripos52_dihedrals.prm Dihedral term parameter file
solvation_asp.prm	Desolvation term parameter file

Note External restraint penalty terms are defined by the user in the system definition .prm file. Originally, rDock did not support flexible receptor dihedrals or explicit structural waters, and the overall scoring function consisted of just the SCORE.INTER and SCORE.INTRA branches. At that time, the intermolecular scoring function definition file (e.g. RbtInterIdxSF.prm) represented precisely the SCORE.INTER terms, and the intramolecular definition file (RbtIntraSF.prm) represented precisely the SCORE.INTRA terms. With the introduction of receptor flexibility and explicit structural waters (and hence the need for the SCORE.SYSTEM branch), the situation is slightly more complex. For implementation reasons, many of the terms reported under SCORE.SYSTEM (with the exception of the dihedral term) are calculated simultaneously with the terms reported under SCORE.INTER, and hence their parameterisation is defined implicitly in the intermolecular scoring function definition file. In contrast, the ligand intramolecular scoring function terms can be controlled independently.

7 Docking protocol

7.1 Protocol Summary

7.1.1 Pose Generation

rDock uses a combination of stochastic and deterministic search techniques to generate low energy ligand poses. The standard docking protocol to generate a single ligand pose uses 3 stages of Genetic Algorithm search (GA1, GA2, GA3), followed by low temperature Monte Carlo (MC) and Simplex minimization (MIN) stages.

Several scoring function parameters are varied between the stages to promote efficient sampling. The ECUT parameter of the S_{inter} vdW potential (defining the hardness of the intermolecular close range potential) is increased from 1 in the first GA stage (GA1) to a maximum of 120 in the MC and MIN stages, with intermediate values of 5 in GA2 and 25 in GA3. The functional form of the S_{inter} vdW potential is switched from a 4-8 potential in GA1 and GA2 to a 6-12 potential in GA3, MC and MIN.

In a similar fashion, the overall weight of the S_{intra} dihedral potential is ramped up from an initial value of 0.1 in GA1 to a final value of 0.5 in the MC and MIN stages, with intermediate values of 0.2 in GA2 and 0.3 in GA3. In contrast, the S_{intra} vdW parameters (as used for the ligand intramolecular potential) remain fixed at the final, hard values throughout the calculation (ECUT=120, 6-12 potential).

Overall, we found this combination of parameter changes allows for efficient sampling of the very poor starting poses, whilst minimising the likelihood that poor ligand internal conformations are artificially favoured and trapped early in the search, and ensures that physically realistic potentials are used for final optimisation and analysis.

7.1.2 Genetic Algorithm

The GA chromosome consists of the ligand centre of mass (com), the ligand orientation, as represented by the quaternion (q) required to rotate the ligand principal axes from the Cartesian reference axes, the ligand rotatable dihedral angles, and the receptor rotatable dihedral angles. The ligand centre of mass and orientation descriptors, although represented by multiple floating point values (com.x, com.y, com.z, and q.s, q.x, q.y, q.z respectively), are operated on as intact entities by the GA mutation and crossover operators.

For so-called free docking, in which no external restraints other than the cavity penalty are imposed, the initial population is generated such that the ligand centre of mass is constrained to lie on a randomly selected grid point within the defined docking volume, and the ligand orientation and all dihedral angles are randomised completely. Mutations to the ligand centre of mass are by a random distance along a randomly oriented unit vector. Mutations to the ligand orientation are performed by rotating the ligand principal axes by a random angle around a randomly oriented unit vector. Mutations to the ligand and receptor dihedral angles are by a random angle. All mutation distances and angles are randomly selected from rectangular distributions of defined width.

A generation is considered to have passed when the number of new individuals created is equal to the population size. Instead of having a fixed number of generations, the GA is allowed to continue until the population converges. The population is considered converged when the score of the best scoring pose fails to improve by more than 0.1 over the last three generations. This allows early termination of poorly performing runs for which the initial population is not able to generate a good solution.

During initial testing the impact of a wide variety of GA parameters (Table 7.1) were explored on a small, representative set of protein-ligand complexes (3ptb, 1rbp, 1stp, 3dfr). We measured the frequency that the algorithm was able to find the experimental conformation, and the average run time. Optimum results were obtained with a steady state GA, roulette wheel selection, a single population of size 100 * (number of rotatable bonds), a crossover:mutation ratio of 40:60, and mutation distribution widths of ligand translation 2Å, ligand rotation 30° and dihedral angle 30°. These parameters have been found to be generally robust across a wide variety of systems.

7.1.3 Monte Carlo

The method and parameters for low temperature Monte Carlo are similar to those described for phase 4 of the RiboDock simulated annealing search protocol. The overall number of trials is scaled according to the number of rotatable bonds in the ligand, from a minimum of 500 ($N_{rot} = 0$) to a maximum of 2000

Table 7.1: Summary of GA parameter space explored, and final values.

Parameter	Values Explored	Final Value
Number of populations	1, 2, 3, 4, 5	1
Selection operator	Roulette wheel, Rank	Roulette wheel
Mutation	Rectangular, Cauchy	Rectangular
GA	Generational, Steady state	Steady state
Elitism	Yes, No	No
No of individuals modified in each generation	All values from 1 to population size	0.5 * population size
Population size	50, 75, 100, 125, 150, 200, 400, 800 * number of rotatable bonds	100 * number of rotatable bonds
Probability of choosing Crossover vs. Mutation	0.0, 0.05, 0.1 ... 0.9, 0.95, 1.0	0.4
Torsion Step	3, 12, 21, 30°	30°
Rotational Step	3, 12, 21, 30°	30°
Translation Step	0.1, 0.8, 1.4, 2.0 Å	2.0 Å

($N_{rot} = 15$). Maximum step sizes are: translation 0.1Å, ligand rotation 10° and dihedral angle 10°. Step sizes are halved if the Metropolis acceptance rate falls below 0.25.

7.1.4 Simplex

The Nelder-Mead's Simplex minimisation routine operates on the same chromosome representation as the GA, with the exception that the composite descriptors (centre of mass and orientation) are decomposed into their constituent floating point values.

7.2 Code Implementation

Docking protocols are constructed at run-time (by class RbtTransformFactory) from docking protocol definition files (rDock .prm format). The default location for docking protocol files is $RBT_ROOT/data/scripts/. The docking protocol definition file defines the sequence of search algorithms that constitute a single docking run for a single ligand record. Each search algorithm component operates either on a single chromosome representing the system degrees of freedom, or on a population of such chromosomes.
The chromosome is constructed (by RbtChromFactory) as an aggregate of individual chromosome elements for the receptor, ligand and explicit solvent degrees of freedom, as defined by the flexibility parameters in the system definition file.

Table 7.2: Chromosome elements

Element	Defined by	Class	Length
Position	Centre of mass	RbtChromPositionElement	3
Orientation	Euler angles for principal axes	RbtChromPositionElement	3
Dihedral	Dihedral angle for rotatable bond	RbtChromDihedralElement	1 per bond
Occupancy	Explicit water occupancy state	RbtChromOccupancylElement	1 per water

7.3 Standard rDock docking protocol (dock.prm)

As stated above in this section, the standard rDock docking protocol consists of three phases of a Genetic Algorithm search, followed by low-temperature Monte Carlo and Simplex minimisation.

By way of example, the `dock.prm` script is documented in detail. The other scripts are very similar.

Scoring Function The scoring function definition is referenced within the docking protocol definition file itself, in the SCORE section. This section contains entries for the INTER, INTRA and SYSTEM scoring function definition files.

Table 7.3: Search algorithm components and C++ implementation classes

Component	Class	Operates on
Randomise population	RbtRandPopTransform	Chromosome population
Genetic algorithm search	RbtGATransform	Chromosome population
Monte Carlo simulated annealing	RbtSimAnnTransform	Single chromosome
Simplex minimisation	RbtSimplexTransform	Single chromosome
Null operation	RbtNullTransform	N/A

Table 7.4: Docking protocol data files

File	Description
score.prm	Calculates score only for initial conformation (standard scoring function)
score_solv.prm	As above, but uses desolvation scoring function
minimise.prm	Simplex minimisation of initial conformation (standard scoring function)
minimise_solv.prm	As above, but uses desolvation scoring function
dock.prm	Full docking search (standard scoring function)
dock_solv.prm	As above, but uses desolvation scoring function
dock_grid.prm	Full docking search (standard scoring function, grid-based vdW term)
dock_solv_grid.prm	Full docking search (desolvation scoring function, grid-based vdW term)

```
SECTION SCORE
        INTER RbtInterIdxSF.prm
        INTRA RbtIntraSF.prm
        SYSTEM RbtTargetSF.prm
END_SECTION

SECTION SETSLOPE_1
        TRANSFORM RbtNullTransform
        # Dock with a high penalty for leaving the cavity
        WEIGHT@SCORE.RESTR.CAVITY 5.0
        # Gradually ramp up dihedral weight from 0.1->0.5
        WEIGHT@SCORE.INTRA.DIHEDRAL 0.1
        # Gradually ramp up energy cutoff for switching to quadratic
        ECUT@SCORE.INTER.VDW 1.0
        # Start docking with a 4-8 vdW potential
        USE_4_8@SCORE.INTER.VDW TRUE
        # Broader angular dependence
        DA1MAX@SCORE.INTER.POLAR 180.0
        # Broader angular dependence
        DA2MAX@SCORE.INTER.POLAR 180.0
        # Broader distance range
        DR12MAX@SCORE.INTER.POLAR 1.5
END_SECTION
```

Genetic Algorithm All sections that contain the TRANSFORM parameter are interpreted as a search algorithm component. The value of the TRANSFORM parameter is the C++ implementation class name for that transform. An RbtNullTransform can be used to send messages to the scoring function to modify key scoring function parameters in order to increase search efficiency. All parameter names that contain the @ symbol are interpreted as scoring function messages, where the string before the @ is the scoring function parameter name, the string after the @ is the scoring function term, and the parameter value is the new value for the scoring function parameter. Messages are sent blind, with no success feedback, as the docking protocol has no knowledge of the composition of the scoring function terms.
Here, we start the docking with a soft 4-8 vdW potential, a reduced dihedral potential, and extended polar ranges (distances and angles) for the intermolecular polar potential. These changes are all designed

to aid sampling efficiency by not overpenalising bad contacts in the initial, randomised population, and by encouraging the formation of intermolecular hydrogen bonds.

```
SECTION RANDOM_POP
        TRANSFORM RbtRandPopTransform
        POP_SIZE 50
        SCALE_CHROM_LENGTH TRUE
END_SECTION
```

Creates an initial, randomised chromosome population. If SCALE_CHROM_LENGTH is false, the population is of fixed size, defined by POP_SIZE. If SCALE_CHROM_LENGTH is true, the population is proportional to the overall chromosome length, defined by POP_SIZE multiplied by the chromosome length.

```
SECTION GA_SLOPE1
        TRANSFORM RbtGATransform
        PCROSSOVER 0.4 # Prob. of crossover
        XOVERMUT TRUE # Cauchy mutation after each crossover
        CMUTATE FALSE # True = Cauchy; False = Rectang. for regular mutations
        STEP_SIZE 1.0 # Max relative mutation
END_SECTION
```

First round of GA.

```
SECTION SETSLOPE_3
        TRANSFORM RbtNullTransform
        WEIGHT@SCORE.INTRA.DIHEDRAL 0.2
        ECUT@SCORE.INTER.VDW 5.0
        DA1MAX@SCORE.INTER.POLAR 140.0
        DA2MAX@SCORE.INTER.POLAR 140.0
        DR12MAX@SCORE.INTER.POLAR 1.2
END_SECTION
```

Increases the ligand dihedral weight, increases the short-range intermolecular vdW hardness (ECUT), and decreases the range of the intermolecular polar distances and angles.

```
SECTION GA_SLOPE3
        TRANSFORM RbtGATransform
        PCROSSOVER 0.4 # Prob. of crossover
        XOVERMUT TRUE # Cauchy mutation after each crossover
        CMUTATE FALSE # True = Cauchy; False = Rectang. for regular mutations
        STEP_SIZE 1.0 # Max relative mutation
END_SECTION
```

Second round of GA with revised scoring function parameters.

```
SECTION SETSLOPE_5
        TRANSFORM RbtNullTransform
        WEIGHT@SCORE.INTRA.DIHEDRAL 0.3
        ECUT@SCORE.INTER.VDW 25.0
        # Now switch to a convential 6-12 for final GA, MC, minimisation
        USE_4_8@SCORE.INTER.VDW FALSE
        DA1MAX@SCORE.INTER.POLAR 120.0
        DA2MAX@SCORE.INTER.POLAR 120.0
        DR12MAX@SCORE.INTER.POLAR 0.9
END_SECTION
```

Further increases the ligand dihedral weight, further increases the short-range intermolecular vdW hardness (ECUT), and further decreases the range of the intermolecular polar distances and angles. Also switches from softer 4-8 vdW potential to a harder 6-12 potential for final round of GA, MC and minimisation.

```
SECTION GA_SLOPE5
        TRANSFORM RbtGATransform
```

```
        PCROSSOVER 0.4 # Prob. of crossover
        XOVERMUT TRUE # Cauchy mutation after each crossover
        CMUTATE FALSE # True = Cauchy; False = Rectang. for regular mutations
        STEP_SIZE 1.0 # Max relative mutation
END_SECTION
```

Final round of GA with revised scoring function parameters.

```
SECTION SETSLOPE_10
        TRANSFORM RbtNullTransform
        WEIGHT@SCORE.INTRA.DIHEDRAL 0.5 # Final dihedral weight matches SF file
        ECUT@SCORE.INTER.VDW 120.0 # Final ECUT matches SF file
        DA1MAX@SCORE.INTER.POLAR 80.0
        DA2MAX@SCORE.INTER.POLAR 100.0
        DR12MAX@SCORE.INTER.POLAR 0.6
END_SECTION
```

Resets all the modified scoring function parameters to their final values, corresponding to the values in the scoring function definition files. It is important that the final scoring function optimised by the docking search can be compared directly with the score-only and minimisation-only protocols, in which the scoring function parameters are not modified.

```
SECTION MC_10K
        TRANSFORM RbtSimAnnTransform
        START_T 10.0
        FINAL_T 10.0
        NUM_BLOCKS 5
        STEP_SIZE 0.1
        MIN_ACC_RATE 0.25
        PARTITION_DIST 8.0
        PARTITION_FREQ 50
        HISTORY_FREQ 0
END_SECTION
```

Monte Carlo Low temperature Monte Carlo sampling, starting from fittest chromosome from final round of GA.

```
SECTION SIMPLEX
        TRANSFORM RbtSimplexTransform
        MAX_CALLS 200
        NCYCLES 20
        STOPPING_STEP_LENGTH 10e-4
        PARTITION_DIST 8.0
        STEP_SIZE 1.0
        CONVERGENCE 0.001
END_SECTION
```

Minimisation Simplex minimisation, starting from fittest chromosome from low temperature Monte Carlo sampling.

```
SECTION FINAL
        TRANSFORM RbtNullTransform
        WEIGHT@SCORE.RESTR.CAVITY 1.0 # revert to standard cavity penalty
END_SECTION
```

Finally, we reset the cavity restraint penalty to 1. The penalty has been held at a value of 5 throughout the search, to strongly discourage the ligand from leaving the docking site.

8 System definition file reference

Although known previously as the receptor .prm file, the system definition file has evolved to contain much more than the receptor information. The system definition file is used to define:

- Receptor input files and flexibility parameters (the section called Receptor definition)
- Explicit solvent input file and flexibility parameters (the section called Solvent definition)
- Ligand flexibility parameters (the section called Ligand definition).
- External restraint terms to be added to the total scoring function (e.g. cavity restraint, pharmacophoric restraint)

8.1 Receptor definition

The receptor can be loaded from a single MOL2 file, or from a combination of Charmm PSF and CRD files. In the former case the MOL2 file provides the topology and reference coordinates simultaneously, whereas in the latter case the topology is loaded from the PSF file and the reference coordinates from the CRD file. For historical compatibility reasons, receptor definition parameters are all defined in the top-level namespace and should not be placed between SECTION.END_SECTION pairs.

Caution If MOL2 and PSF/CRD parameters are defined together, the MOL2 parameters take precedence and are used to load the receptor model.

Table 8.1: Receptor definition parameters

Parameter	Description	Type	Default	Range of values
Parameters specific to loading receptor in MOL2 file format				
RECEPTOR_FILE	Name of receptor MOL2 file	Filename string	No default value	Valid MOL2 filename
Parameters specific to loading receptor in Charmm PSF/CRD file format				
RECEPTOR_TOPOL_FILE	Name of receptor Charmm PSF file	Filename string	No default value	Valid Charmm PSF filename
RECEPTOR_COORD_FILE	Name of receptor Charmm CRD file	Filename string	No default value	Valid Charmm CRD filename
RECEPTOR_MASSES_FILE	Name of rDock-annotated Charmm masses file	Filename string	No default value	masses.rtf top_all2_prot_na.inp
General receptor parameters, applicable to either file format				
RECEPTOR_SEGMENT_NAME	List of molecular segment names to read from either MOL2 or PSF/CRD file. If this parameter is defined, then any segment/chains not listed are not loaded. This provides a convenient way to remove cofactors, counterions and solvent without modifying the original file.	Comma separated list of segment name strings (without any spaces)	Empty (i.e. all segments read from file)	Comma-separated list of segment name strings

RECEPTOR_FLEX	Defines terminal OH and NH3+ groups within this distance of docking volume as flexible.	float (Angstroms)	Undefined (rigid receptor)	> 0.0 (3.0 is a reasonable value)
Advanced parameters (should not need to be changed by the user)				
RECEPTOR_ALL_H	Disables the removal of explicit non-polar hydrogens from the receptor model. *Not recommended*	boolean	FALSE	TRUE or FALSE
DIHEDRAL_STEP	Maximum mutation step size for receptor dihedral degrees of freedom	float (degrees)	30.0	>0.0

8.2 Ligand definition

Ligand definition parameters need only be defined if you wish to introduce tethering of some or all of the ligand degrees of freedom. If you are running conventional free docking then this section is not required. All ligand definition parameters should be defined in SECTION LIGAND. Note that the ligand input SD file continues to be specified directly on the rbdock command-line and not in the system definition file.

Table 8.2: Ligand definition parameters

Parameter	Description	Type	Default	Range of values
Main user parameters				
TRANS_MODE	Sampling mode for ligand translational degrees of freedom	enumerated string literal	FREE	FIXED TETHERED FREE
ROT_MODE	Sampling mode for ligand whole-body rotational degrees of freedom	enumerated string literal	FREE	FIXED TETHERED FREE
DIHEDRAL_MODE	Sampling mode for ligand internal dihedral degrees of freedom	enumerated string literal	FREE	FIXED TETHERED FREE
MAX_TRANS	(for TRANS_MODE = TETHERED only) Maximum deviation allowed from reference centre of mass	float (Angstroms)	1.0	>0.0
MAX_ROT	(for ROT_MODE = TETHERED only) Maximum deviation allowed from orientation of reference principle axes	float (degrees)	30.0	>0.0 - 180.0
MAX_DIHEDRAL	(for DIHEDRAL_MODE = TETHERED only) Maximum deviation allowed from reference dihedral angles for any rotatable bond	float (degrees)	30.0	>0.0 - 180.0

Advanced parameters (should not need to be changed by the user)				
TRANS_STEP	Maximum mutation step size for ligand translational degrees of freedom	float (Angstroms)	2.0	>0.0
ROT_STEP	Maximum mutation step size for ligand whole-body rotational degrees of freedom	float (degrees)	30.0	>0.0
DIHEDRAL_STEP	Maximum mutation step size for ligand internal dihedral degrees of freedom	float (degrees)	30.0	>0.0

8.3 Solvent definition

Solvent definition parameters need only be defined if you wish to introduce explicit structural waters into the docking calculation, otherwise this section is not required. All solvent definition parameters should be defined in SECTION SOLVENT.

Table 8.3: Solvent definition parameters

Parameter	Description	Type	Default	Range of values
Main user parameters				
FILE	Name of explicit solvent PDB file	File name string	No default value (mandatory parameter)	Valid PDB filename
TRANS_MODE	Sampling mode for solvent translational degrees of freedom. If defined here, the value overrides the per-solvent translational sampling modes defined in the solvent PDB file	enumerated string literal	FREE	FIXED TETHERED FREE
ROT_MODE	Sampling mode for solvent whole-body rotational degrees of freedom. If defined here, the value overrides the per-solvent rotational sampling modes defined in the solvent PDB file	enumerated string literal	FREE	FIXED TETHERED FREE
MAX_TRANS	(for TRANS_MODE = TETHERED waters only) Maximum deviation allowed from reference water oxygen positions. The same value is applied to all waters with TRANS_MODE = TETHERED; it is not possible currently to define per-solvent MAX_TRANS values	float (Angstroms)	1.0	>0.0

MAX_ROT	(for ROT_MODE = TETHERED waters only) Maximum deviation allowed from orientation of reference principal axes. The same value is applied to all waters with ROT_MODE = TETHERED; it is not possible currently to define per-solvent MAX_ROT values	float (degrees)	30.0	>0.0 - 180.0
OCCUPANCY	Controls occupany state sampling for all explicit solvent. If defined here, the value overrides the per-solvent occupancy states defined in the solvent PDB file	float	1.0	0.0 - 1.0
Advanced parameters (should not need to be changed by the user)				
TRANS_STEP	Maximum mutation step size for solvent translational degrees of freedom	float (Angstroms)	2.0	>0.0
ROT_STEP	Maximum mutation step size for solvent wholebody rotational degrees of freedom	float (degrees)	30.0	>0.0
OCCUPANCY_STEP	Maximum mutation step size for solvent occupancy state degrees of freedom	float (degrees)	1.0	0.0 - 1.0

Solvent occupancy state sampling. OCCUPANCY = 0 permanently disables all solvent; OCCUPANCY = 1.0 permanently enables all solvent; OCCUPANCY between 0 and 1 activates variable occupancy state sampling, where the value represents the initial probability that the solvent molecule is enabled. For example, OCCUPANCY = 0.5 means that the solvent is enabled in 50% of the initial GA population. However, the probability that the solvent is actually enabled in the final docking solution will depend on the particular ligand, the scoring function terms, and on the penalty for solvent binding. The occupancy state chromosome value is managed as a continuous variable between 0.0 and 1.0, with a nominal mutation step size of 1.0. Chromosome values lower than the occupancy threshold (defined as 1.0 - OCCUPANCY) result in the solvent being disabled; values higher than the threshold result in the solvent being enabled.

8.4 Cavity mapping

The cavity mapping section is mandatory. You should choose one of the mapping algorithms shown below. All mapping parameters should be defined in SECTION MAPPER.

Table 8.4: Two sphere site mapping parameters

Parameter	Description	Type	Default	Range of values
Main user parameters				
SITE_MAPPER	Mapping algorithm specifier	string literal	RbtSphere-SiteMapper	fixed

CENTER	(x,y,z) center of cavity mapping region	Bracketed cartesian coordinate string (x,y,z)	None	None
RADIUS	Radius of cavity mapping region	float (Angstroms)	10.0	> 0.0 (10.0-20.0 suggested range)
SMALL_SPHERE	Radius of small probe	float (Angstroms)	1.5	> 0.0 (1.0-2.0 suggested range)
LARGE_SPHERE	Radius of large probe	float (Angstroms)	4.0	> SMALL_SPHERE (3.5 - 6.0 suggested range)
MAX_CAVITIES	Maximum number of cavities to accept (in descending order of size)	integer	99	>0
	Advanced parameters (less frequently changed by the user)			
VOL_INCR	Receptor atom radius increment for excluded volume	float (Angstroms)	0.0	>= 0.0
GRID_STEP	Grid resolution for mapping	float (Angstroms)	0.5	>0.0 (0.3 - 0.8 suggested range)
MIN_VOLUME	Minimum cavity volume to accept (in $\mathring{A}^3$, not grid points)	float (Angstroms^3)	100	>0 (100-300 suggested range)

Table 8.5: Reference ligand site mapping parameters

Parameter	**Description**	**Type**	**Default**	**Range of values**
	Main user parameters			
SITE_MAPPER	Mapping algorithm specifier	string literal	RbtLigandSiteMapper	fixed
REF_MOL	Reference ligand SD file name	string	ref.sd	None
RADIUS	Radius of cavity mapping region	float (Angstroms)	10.0	> 0.0 (10.0-20.0 suggested range)
SMALL_SPHERE	Radius of small probe	float (Angstroms)	1.5	> 0.0 (1.0-2.0 suggested range)
LARGE_SPHERE	Radius of large probe	float (Angstroms)	4.0	> SMALL_SPHERE (3.5 - 6.0 suggested range)
MAX_CAVITIES	Maximum number of cavities to accept (in descending order of size)	integer	99	>0
	Advanced parameters (less frequently changed by the user)			
VOL_INCR	Receptor atom radius increment for excluded volume	float (Angstroms)	0.0	>= 0.0

GRID_STEP	Grid resolution for mapping	float (Angstroms)	0.5	>0.0 (0.3 - 0.8 suggested range)
MIN_VOLUME	Minimum cavity volume to accept (in $\AA^3$, not grid points)	float ($\AA^3$)	100	>0 (100-300 suggested range)

8.5 Cavity restraint

The cavity restraint penalty function is mandatory and is designed to prevent the ligand from exiting the docking site. The function is calculated over all non-hydrogen atoms in the ligand (and over all explicit water oxygens that can translate). The distance from each atom to the nearest cavity grid point is calculated. If the distance exceeds the value of RMAX, a penalty is imposed based on the value of (distance - RMAX). The penalty can be either linear or quadratic depending on the value of the QUADRATIC parameter. It should not be necessary to change any the parameters in this section. Note that the docking protocol itself will manipulate the WEIGHT parameter, so any changes made to WEIGHT will have no effect.

```
SECTION CAVITY
        SCORING_FUNCTION RbtCavityGridSF
        WEIGHT 1.0
        RMAX 0.1
        QUADRATIC FALSE
END_SECTION
```

8.6 Pharmacophore restraints

This section need only be defined if you wish to dock with pharmacophore restraints. If you are running conventional free docking then this section is not required. All pharmacophore definition parameters should be defined in SECTION PHARMA.

Table 8.6: Pharmacophore restraint parameters

Parameter	Description	Type	Default	Range of values
CONSTRAINTS_FILE	Mandatory pharmacophore restraints file	File name string	None (mandatory parameter)	Valid file name
OPTIONAL_FILE	Optional pharmacophore restraints file	File name string	None (optional parameter)	Valid file name, or empty
NOPT	Number of optional restraints that should be met	Integer	0	Between 0 and number of restraints in OPTIONAL_FILE

WRITE_ERRORS	Ligands with insufficient pharmacophore features to match the mandatory restraints are always removed prior to docking. If this parameter is true, the pre-filtered ligands are written to an error SD file with the same root name as the docked pose output SD file, but with an _errors.sd suffix. If false, the pre-filtered ligands are not written	Boolean	false	true or false
WEIGHT	Overall weight for the pharmacophore penalty function	Float	1.0	>= 0.0

Calculation of mandatory restraint penalty. The list of ligand atoms that matches each restraint type in the mandatory restraints file is precalculated for each ligand as it is loaded. If the ligand contains insufficient features to satisfy all of the mandatory restraints the ligand is rejected and is not docked. Note that the rejection is based purely on feature counts and does not take into account the possible geometric arrangements of the features. Rejected ligands are optionally written to an error SD file. The penalty for each restraint is based on the distance from the nearest matching ligand atom to the pharmacophore restraint centre. If the distance is less than the defined tolerance (restraint sphere radius), the penalty is zero. If the distance is greater than the defined tolerance a quadratic penalty is applied, equal to (nearest distance - tolerance)^2.

Calculation of optional restraint penalty. The individual restraint penalties for each restraint in the optional restraints file are calculated in the same way as for the mandatory penalties. However, only the NOPT lowest scoring (least penalised) restraints are summed for any given docking pose. Any remaining higher scoring optional restraints are ignored and do not contribute to the total pharmacophore restraint penalty.

Calculation of overall restraint penalty. The overall pharmacophore restraint penalty is the sum of the mandatory restraint penalties and the NOPT lowest scoring optional restraint penalties, multiplied by the WEIGHT parameter value.

8.7 NMR restraints

To be completed. However, this feature has rarely been used.

8.8 Example system definition files

Full system definition file with all sections and common parameters enumerated explicitly

```
RBT_PARAMETER_FILE_V1.00
TITLE HSP90-PU3-lig-cavity, solvent flex=5
RECEPTOR_FILE PROT_W3_flex.mol2
RECEPTOR_SEGMENT_NAME PROT
RECEPTOR_FLEX 3.0
SECTION SOLVENT
        FILE PROT_W3_flex_5.pdb
        TRANS_MODE TETHERED
        ROT_MODE TETHERED
        MAX_TRANS 1.0
```

```
        MAX_ROT 30.0
        OCCUPANCY 0.5
END_SECTION
SECTION LIGAND
        TRANS_MODE FREE
        ROT_MODE FREE
        DIHEDRAL_MODE FREE
        MAX_TRANS 1.0
        MAX_ROT 30.0
        MAX_DIHEDRAL 30.0
END_SECTION
SECTION MAPPER
        SITE_MAPPER RbtLigandSiteMapper
        REF_MOL ref.sd
        RADIUS 5.0
        SMALL_SPHERE 1.0
        MIN_VOLUME 100
        MAX_CAVITIES 1
        VOL_INCR 0.0
        GRIDSTEP 0.5
END_SECTION
SECTION CAVITY
        SCORING_FUNCTION RbtCavityGridSF
        WEIGHT 1.0
END_SECTION
SECTION PHARMA
        SCORING_FUNCTION RbtPharmaSF
        WEIGHT 1.0
        CONSTRAINTS_FILE mandatory.const
        OPTIONAL_FILE optional.const
        NOPT 3
        WRITE_ERRORS TRUE
END_SECTION
```

9 Molecular files and atoms typing

Macromolecular targets (protein or RNA) are input from Tripos MOL2 files (Rbt-MOL2FileSource) or from pairs of Charmm PSF (RbtPsfFileSource) and CRD (RbtCrd-FileSource) files. Ligands are input from MDL SD files (RbtMdlFileSource). Explicit structural waters are input optionally from PDB files (RbtPdbFileSource). Ligand docking poses are output to MDL SD files.
The rDock scoring functions have been defined and validated for implicit non-polar hydrogen (extended carbon) models only. If you provide all-atom models, be aware that the non-polar hydrogens will be removed automatically. Polar hydrogens must be defined explicitly in the molecular files, and are not added by rDock. Positive ionisable and negative ionisable groups can be automatically protonated and deprotonated respectively to create common charged groups such as guanidinium and carboxylic acid groups.
MOL2 is now the preferred file format for rDock as it eliminates many of the atom typing issues inherent in preparing and loading PSF files. The use of PSF/CRD files is strongly discouraged. The recommendation is to prepare an all-atom MOL2 file with correct Tripos atom types assigned, and allow rDock to remove non-polar hydrogens on-the-fly.

9.1 Atomic properties.

rDock requires the following properties to be defined per atom. Depending on the file format, these properties may be loaded directly from the molecular input file, or derived internally once the model is loaded:

- Cartesian (x,y,z) coordinates
- Element (atomic number)
- Formal hybridisation state (sp, sp2, sp3, aromatic, trigonal planar)
- Formal charge
- Distributed formal charge (known informally as group charge)
- Tripos force field type (rDock uses a modified version of the Sybyl 5.2 types, extended to include carbon types with implicit non-polar hydrogens)
- Atom name
- Substructure (residue) name
- Atomic radius (assigned per element from $RBT_ROOT/data/RbtElements.dat)

Note The rDock scoring functions do not use partial charges and therefore partial charges do not have to be defined. The atomic radii are simplified radii defined per element, and are used for cavity mapping and in the polar scoring function term, but are not used in the vdW scoring function term. The latter has its own indepedent parameterisation based on the Tripos force field types.

9.2 Difference between formal charge and distributed formal charge

The formal charge on an atom is always an integer. For example, a charged carboxylic acid group (COO-) can be defined formally as a formal double bond to a neutral oxygen sp2, and a formal single bond to a formally charged oxygen sp3. In reality of course, both oxygens are equivalent. rDock distributes the integer formal charge across all equivalent atoms in the charged group that are topologically equivalent. In negatively charged acid groups, the formal charge is distributed equally between the acid oxygens. In positively charged amines, the formal charge is distributed equally between the hydrogens. In charged guanidinium, amidinium, and imidazole groups, the central carbon also receives an equal portion of the formal charge (in addition to the hydrogens). The distributed formal charge is also known as the group charge. The polar scoring functions in rDock use the distributed formal charge to scale the polar interaction strength of the polar interactions.

9.3 Parsing a MOL2 file

MOLECULE, ATOM, BOND and SUBSTRUCTURE records are parsed. The atom name, substructure name, Cartesian coordinates and Tripos atom type are read directly for each atom. The element type (atomic number) and formal hybridisation state are derived from the Tripos type using an internal lookup table. Formal charges are not read from the MOL2 file and do not have to be assigned correctly in the file. Distributed formal charges are assigned directly by rDock based on standard substructure and atom names as described below.

9.4 Parsing an SD file

Cartesian coordinates, element and formal charge are read directly for each atom. Formal bond orders are read for each bond. Atom names are derived from element name and atom ID (e.g. C1, N2, C3 etc). The substructure name is MOL. Formal hybridisation states are de- rived internally for each atom based on connectivity patterns and formal bond orders. The Tripos types are asssigned using internal rules based on atomic number, formal hybridisation state and formal charges. The integer formal charges are distributed automatically across all topologically equivalent atoms in the charged group.

9.5 Assigning distributed formal charges to the receptor

rDock provides a file format independent method for assigning distributed formal charges directly to the receptor atoms, which is used by the MOL2 and PSF/CRD file readers. The method uses a lookup table based on standard substructure and atom names, and does not require the integer formal charges to be assigned to operate correctly.
The lookup table file is $RBT_ROOT/data/sf/RbtIonicAtoms.prm. Each section name represents a substructure name that contains formally charged atoms. The entries within the section represent the atom names and distributed formal charges for that substructure name. The file provided with rDock contains entries for all standard amino acids and nucleic acids, common metals, and specific entries required for processing the GOLD CCDC/Astex validation sets.

Important You may have to extend RbtIonicAtoms.prm if you are working with non-standard receptor substructure names and/or atom names, in order for the distributed formal charges to be assigned correctly.

10 rDock file formats

10.1 .prm file format

The .prm file format is an rDock-specific text format and is used for:

- system definition files (known previously as receptor .prm files)
- scoring function definition files
- search protocol definition files

The format is simple and allows for an arbitrary number of named parameter/value pairs to be defined, optionally divided into named sections. Sections provide a namespace for parameter names, to allow parameter names to be duplicated within different sections. The key features of the format are:

- The first line of the file must be RBT_PARAMETER_FILE_V1.00 with no preceeding whitespace.
- Subsequent lines may contain either:
  1. comment lines
  2. reserved keywords TITLE, SECTION, or END_SECTION
  3. parameter name/value pairs
- Comment lines should start with a # character in the first column with no preceeding whitespace, and are ignored.
- The reserved words must start in the first column with no preceeding whitespace.
- The TITLE record should occur only once in the file and is used to provide a title string for display by various scripts such as run_rbscreen.pl. The keyword should be followed by a single space character and then the title string, which may contain spaces. If the TITLE line occurs more than once, the last occurence is used.
- SECTION records can occur more than once, and should always be paired with a closing END_SECTION record. The keyword should be followed by a single space character and then the section name, which may NOT itself contain spaces. All section names must be unique with- in a .prm file. All parameter name/value pairs within the SECTION / END_SECTION block belong to that section.
- Parameter name/value pairs are read as free-format tokenised text and can have preceeding, trailing, and be separated by arbitrary whitespace. This implies that the parameter name and value strings themselves are not allowed to contain any spaces. The value strings are interpreted as numeric, string, or boolean values as appropriate for that parameter. Boolean values should be entered as TRUE or FALSE uppercase strings.

Caution The current implementation of the .prm file reader does not tolerate a TAB character immediately following the TITLE and SECTION keywords. It is very important that the first character after the SECTION keyword in particular is a true space character, otherwise the reserved word will not be detected and the parameters for that section will be ignored.

Example .prm file In the following example, RECEPTOR_FILE is defined in the top level namespace. The remaining parameters are defined in the MAPPER and CAVITY namespaces. The indentation is for readability, and has no significance in the format.

```
RBT_PARAMETER_FILE_V1.00
TITLE 4dfr oxido-reductase

RECEPTOR_FILE 4dfr.mol2

SECTION MAPPER
        SITE_MAPPER RbtLigandSiteMapper
        REF_MOL 4dfr_c.sd
        RADIUS 6.0
```

```
        SMALL_SPHERE 1.0
        MIN_VOLUME 100
        MAX_CAVITIES 1
        VOL_INCR 0.0
        GRIDSTEP 0.5
END_SECTION

SECTION CAVITY
        SCORING_FUNCTION RbtCavityGridSF
        WEIGHT 1.0
END_SECTION
```

10.2 Water PDB file format

rDock requires explicit water PDB files to be in the style as output by the Dowser program. In particular:

- Records can be HETATM or ATOM
- The atom names must be OW, H1 and H2
- The atom records for each water molecule must belong to the same subunit ID
- The subunit IDs for different waters must be distinct, but do not have to be consecutive
- The atom IDs are not used and do not have to be consecutive (they can even be duplicated)
- The order of the atom records within a subunit is unimportant
- The temperature factor field of the water oxygens can be used to define the per-solvent flexibility modes. The temperature factors of the water hydrogens are not used.

Table 10.1: Conversion of temperature factor values to solvent flexibility modes

PDB temperature factor	Solvent translational flexibility	Solvent rotational flexibility
0	FIXED	FIXED
1	FIXED	TETHERED
2	FIXED	FREE
3	TETHERED	FIXED
4	TETHERED	TETHERED
5	TETHERED	FREE
6	FREE	FIXED
7	FREE	TETHERED
8	FREE	FREE

Example Valid rDock PDB file for explicit, flexible waters

```
REMARK tmp_1YET.pdb xtal_hoh.pdb
HETATM 3540 OW HOH W 106  28.929  12.684  20.864  1.00  1.0
HETATM 3540  H1 HOH W 106  28.034  12.390  21.200  1.00
HETATM 3540  H2 HOH W 106  29.139  12.204  20.012  1.00
HETATM 3542 OW HOH W 108  27.127  14.068  22.571  1.00  2.0
HETATM 3542  H1 HOH W 108  26.632  13.344  23.052  1.00
HETATM 3542  H2 HOH W 108  27.636  13.673  21.806  1.00
HETATM 3679 OW HOH W 245  27.208  10.345  27.250  1.00  3.0
HETATM 3679  H1 HOH W 245  27.657  10.045  26.409  1.00
HETATM 3679  H2 HOH W 245  26.296  10.693  27.036  1.00
HETATM 3680 OW HOH W 246  31.737  12.425  21.110  1.00  4.0
HETATM 3680  H1 HOH W 246  31.831  12.448  22.106  1.00
HETATM 3680  H2 HOH W 246  30.775  12.535  20.863  1.00
```

10.3 Pharmacophore restraints file format

Pharmacophore restraints are defined in a simple text file, with one restraint per line. Each line should contain the following values, separated by commas or whitespace:

x y z coords of restraint centre, tolerance (in Angstroms), restraint type string.

The supported restraint types are:

Table 10.2: Pharmacophore restraint types

String	**Description**	**Matches**
Any	Any atom	Any non-hydrogen atom
Don	H-bond donor	Any neutral donor hydrogen
Acc	H-bond acceptor	Any neutral acceptor
Aro	Aromatic	Any aromatic ring centre (pseudo atom)
Hyd	Hydrophobic	Any non-polar hydrogens (ifpresent), any C sp3 or S sp3, any C or S not bonded to O sp2, any Cl, Br, I
Hal	Hydrophobic, aliphatic	Subset of Hyd, sp3 atoms only
Har	Hydrophobic, aromatic	Subset of Hyd, aromatic atoms only
Ani	Anionic	Any atom with negative distributed formal charge
Cat	Cationic	Any atom with positive distributed formal charge

11 rDock programs

Programs summary tables:

Table 11.1: Core rDock C++ executables

Executable	Used for	Description
rbcavity	Preparation	Cavity mapping and preparation of docking site (.as) file.
rbcalcgrid	Preparation	Calculation of vdW grid files (usually called by make_grid.csh wrapper script).
rbdock	Docking	The main rDock docking engine itself.

Table 11.2: Auxiliary rDock programs

Executable	Used for	Description
sdtether	Preparation	Prepares a ligand SD file for tethered scaffold docking. Annotates ligand SD file with tethered substructure atom indices. Requires OpenBabel python bindings.
rbhtfinder	Preparation	Used to optimise a high-throughput docking protocol from an initial exhaustive docking of a small representative ligand library. Parametrize a multi-step protocol for your system.
make_grid.csh	Preparation	Creates the vdW grid files required for grid-based docking protocols (dock_grid.prm and dock_solv_grid.prm). Simple front-end to rbcalcgrid.
rbmoegrid	Analysis	Converts rDock vdW grids to MOE grid format for visualisation.
rblist	Analysis	Outputs miscellaneous information for ligand SD file records.
sdrmsd	Analysis	Calculation of ligand Root Mean Squared Displacement (RMSD) between reference and docked poses, taking into account ligand topological symmetry. Requires OpenBabel python bindings.
sdfilter	Analysis	Utility for filtering SD files by arbitrary data field expressions. Useful for simple post-docking filtering by score components.
sdsort	Analysis	Utility for sorting SD files by arbitrary data field. Useful for simple post-docking filtering by score components.
sdreport	Analysis	Utility for reporting SD file data field values. Output in tab-delimited or csv format.
sdsplit	Utility	Splits an SD file into multiple smaller SD files of fixed number of records.
sdmodify	Utility	Sets the molecule title line of each SD record equal to a given SD data field.

11.1 Programs reference

11.1.1 rbdock

rbdock – the rDock docking engine itself.

```
$RBT_ROOT/bin/rbdock
{-i input ligand MDL SD file}
{-o output MDL SD file}
{-r system definition .prm file}
{-p docking protocol .prm file}
[-n number of docking runs/ligand]
[-s random seed]
[-T debug trace level]
[[-t SCORE.INTER threshold] | [-t filter definition file]]
[ -ap -an -allH -cont ]
```

Simple exhaustive docking. The minimum requirement for rbdock is to specify the input (-i) and output (-o) ligand SD file names, the system definition .prm file (-r) and the docking protocol .prm file (-p). This will perform one docking run per ligand record in the input SD file and output all docked ligand poses to the output SD file. Use -n to increase the number of docking runs per ligand record.

High-throughput docking 1. The -t and -cont options can be used to construct high-throughput protocols. If the argument following -t is numeric it is interpreted as a threshold value for SCORE.INTER, the total intermolecular score between ligand and receptor/solvent. In the absence of -cont, the threshold acts as an early termination filter, and the docking runs for each ligand will be terminated early once the threshold value has been exceeded. Note that the threshold is applied only at the end of each individual docking run, not during the runs themselves. If the -cont (continue) option is specified as well, the threshold acts as an output pose filter instead of a termination filter. The docking runs for each ligand run to completion as in the exhaustive case, but only the docking poses that exceed the threshold value of SCORE.INTER are written to the output SD file.

High throughput docking 2. Alternatively, if the argument following -t is non-numeric it is interpreted as a filter definition file. The filter definition file can be used to define multiple termination filters and multiple output pose filters in a generic way. Any docking score component can be used in the filter definitions. run_rbscreen.pl generates a filter definition file for multi-stage, highthroughput docking, with progressive score thresholds for early termination of poorly performing ligands. The use of filter definition files is preferred over the more limited SCORE.INTER filtering described above, whose use is now deprecated.

Automated ligand protonation/deprotonation. The -ap option activates the automated protonation of ligand positive ionisable centres, notably amines, guanidines, imidazoles, and amidines. The -an option activates the automated deprotonation of ligand negative ionisable centres, notably carboxylic acids, phosphates, phosphonates, sulphates, and sulphonates. The precise rules used by rDock for protonation and deprotonation are quite crude, and are not user-customisable. Therefore these flags are not recommended for detailed validation experiments, in which care should be taken that the ligand protonation states are set correctly in the input SD file. Note that rDock is not capable of converting ionised centres back to the neutral form; these are unidirectional transformations.

Control of ligand non-polar hydrogens. By default, rDock uses an implicit non-polar hydrogen model for receptor and ligand, and all of the scoring function validation has been performed on this basis. If the -allH option is not defined (recommended), all explicit non-polar hydrogens encountered in the ligand input SD file are removed, and only the polar hydrogens (bonded to O, N, or S) are retained. If the -allH option is defined (not recommended), no hydrogens are removed from the ligand. Note that rDock is not capable of adding explicit non-polar hydrogens, if none exist. In other words, the -allH option disables hydrogen removal, it does not activate hydrogen addition. You should always make sure that polar hydrogens are defined explicitly. If the ligand input SD file contains no explicit non-polar hydrogens, the -allH option has no effect. Receptor protonation is controlled by the system definition prm file.

11.1.2 rbcavity

rbcavity – Cavity mapping and preparation of docking site (.as) file file.

```
$RBT_ROOT/bin/rbcavity
{-r system definition .prm file}
[ -ras -was -d -v -s ]
[-l distance from cavity]
[-b border]
```

Exploration of cavity mapping parameters. $rbcavity - r.prmfile$ You can run rbcavity with just the -r argument when first preparing a new receptor for docking. This allows you to explore rapidly the impact of the cavity mapping parameters on the generated cavities, whilst avoiding the overhead of actually writing the docking site (.as) file to disk. The number of cavities and volume of each cavity are written to standard output.

Visualisation of cavities. $rbcavity - r.prmfile - d$ If you have access to InsightII you can use the -d option to dump the cavity volumes in InsightII grid file format. There is no need to write the docking site (.as) file first. The InsightII grid files should be loaded into the reference coordinate space of the receptor and contoured at a contour level of 0.99.

Writing the docking site (.as) file. $rbcavity - r.prmfile - was$ When you are happy the mapping parameters, use the -was option to write the docking site (.as) file to disk. The docking site file is a binary file that contains the cavity volumes in a compact format, and a pre-calculated cuboid grid extending over the cavities. The grid represents the distance from each point in space to the nearest cavity grid point, and is used by the cavity penalty scoring function. Calculating the distance grid can take a long time (whereas the cavity mapping itself is usually very fast), hence the -was option should be used sparingly.

Analysis of cavity atoms. $rbcavity - r.prmfile - ras - ldistance$ Use the -l options to list the receptor atoms within a given distance of any of the cavity volumes, for example to determine which receptor OH/NH3+ groups should be flexible. This option requires access to the pre-calculated distance grid embedded within the docking site (.as) file, and is best used in combination with the -ras option, which loads a previously generated docking site file. This avoids the time consuming step of generating the cavity distance grid again. If -l is used without -ras, the cavity distance grid will be calculated on-the-fly each time.

Miscellaneous options. The -s option writes out various statistics on the cavity and on the receptor atoms in the vicinity of the cavity. These values have been used in genetic programming model building for docking pose false positive removal. The -v option writes out the receptor coordinates in PSF/CRD format for use by the rDock Viewer (not documented here). Note that the PSF/CRD files are not suitable for simulation purposes, only for visualisation, as the atom types are not set correctly. The -b option controls the size of the cavity distance grid, and represents the border beyond the actual cavity volumes. It should not be necessary to vary this parameter (default = 8A) unless longer-range scoring functions are implemented.

11.1.3 rbcalcgrid

rbcalcgrid – Calculation of vdW grid files (usually called by make_grid.csh wrapper script).

```
$RBT_ROOT/bin/rbcalcgrid
{-rsystem definition file}
{-ooutput suffix for generated grids}
{-pvdW scoring function prm file}
[-ggrid step]
[-bborder]
```

Note that, unlike rbdock and rbcavity, spaces are not tolerated between the command-line options and their corresponding arguments. See $RBT_ROOT/bin/make_grid.csh for common usage.

11.1.4 make_grid.csh

Creates vdW grids for all receptor prm files listed on command line. Front-end to rbcalcgrid.

11.1.5 rbmoegrid

rbmoegrid - calculates grids for a given atom type

```
Usage:  rbmoegrid -o <OutputRoot> -r <ReceptorPrmFile> -p <SFPrmFile>
                  [-g <GridStep> -b <border> -t <tripos_type>]

Options:          -o <OutFileName> (.grd is suffixed)
                  -r <ReceptorPrmFile> - receptor param file (contains active
                  site params)
                  -p <SFPrmFile> - scoring function param file
                  (default calcgrid_vdw.prm)
                  -g <GridStep> - grid step (default=0.5A)
                  -b <Border> - grid border around docking site (default=1.0A)
                  -t <AtomType> - Tripos atom type (default is C.3)
```

11.1.6 sdrmsd

sdrmsd – Calculation of ligand Root Mean Squared Displacement (RMSD) between reference and docked poses. It takes into account molecule topological symmetry. Requires OpenBabel python bindings.

```
$RBT_ROOT/bin/sdrmsd [options] {reference SD file} {input SD file}
```

With two arguments. sdrmsd calculates the RMSD between each record in the input SD file and the first record of the reference SD file. If there is a mismatch in the number of atoms, the record is skipped and the RMSD is not calculated. The RMSD is calculated over the heavy (non-hydrogen) atoms only. Results are output to standard output. If some record was skipped, a warning message will be printed to standard error.

With fitting. A molecular superposition will be done before calculation of the RMSD. The output will specify an RMSD_FIT calculation was done.

```
sdrmsd -f reference.sdf input.sdf
sdrmsd --fit reference.sdf input.sdf
```

Output a SD file. This option will write an output SD file with the input molecules adding an extra RMSD field to the file. If fitting was done, the molecule coordinates will also be fitted to the reference.

```
sdrmsd -o output.sdf reference.sdf input.sdf
sdrmsd --out=output.sdf reference.sdf input.sdf
```

11.1.7 sdtether

sdtether – Prepares a ligand SD file for tethered scaffold docking. Requires OpenBabel python bindings. Annotates ligand SD file with tethered substructure atom indices.

```
$RBT_ROOT/bin/sdtether {ref. SDfile} {in SDfile} {out SDfile} "{SMARTS query}"
```

sdtether performs the following actions:

- Runs the SMARTS query against the reference SD file to determine the tethered substructure atom indices and coordinates.
- If more than one substructure match is retrieved (e.g. due to topological symmetry, or if the query is too simple) all substructure matchs are retained as the reference and all ligands will be tethered according to all possible matchs.
- Runs the SMARTS query against each record of the input ligand SD file in turn.
- For each substructure match, the ligand coordinates are transformed such that the principal axes of the matching substructure coordinates are aligned with the reference substructure coordinates.
- In addition, an SD data field is added to the ligand record which lists the atom indices of the substructure match, for later retrieval by rDock.
- Each transformed ligand is written to the output SD file.
- Note that if the SMARTS query returns more than one substructure match for a ligand, that ligand is written multiple times to the output file, once for each match, each of which will be docked independently with different tethering information

11.1.8 sdfilter

sdfilter – Post-process an SD file by filtering the records according to data fields or attributes.

```
Usage:  sdfilter -f'$<DataField> <Operator> <Value>' [-s<DataField>] [sdFiles]
or              sdfilter -f<filename> [-s<DataField>] [sdFiles]

Note:   Multiple filters are allowed and are OR'd together.
        Filters can be provided in a file, one per line.

        Standard Perl operators should be used. e.g.
        eq ne lt gt le ge for strings
        == != < > <= >= for numeric

        _REC (record #) is provided as a pseudo-data field
        if -s option is used, _COUNT (#occurrences of DataField)
        is provided as a pseudo-data field

        If SD file list not given, reads from standard input
        Output is to standard output
```

For examples, read section EXAMPLES SECTION

11.1.9 sdreport

sdreport – Produces text summaries of SD records

```
Usage:  sdreport [-l] [-t[<FieldName,FieldName...>]] [-c<FieldName,FieldName...>]
        [-id<IDField>] [-nh] [-o] [-s] [-sup] [sdFiles]

 -l (list format) output all data fields for each record as processed
 -t (tab format) tabulate selected fields for each record as processed
 -c (csv format) comma delimited output of selected fields for each record as
        processed
 -s (summary format) output summary statistics for each unique value of ligand ID
 -sup (supplier format) tabulate supplier details (from Catalyst)
 -id<IDField> data field to use as ligand ID
 -nh don't output column headings in -t and -c formats
 -o use old (v3.00) score field names as default columns in -t and -c formats,
        else use v4.00 field names
 -norm  use normalised score field names as default columns in -t and -c formats
                (normalised = score / #ligand heavy atoms)

Note:   If -l, -t or -c are combined with -s, the listing/table is output within
        each ligand summary
        -sup should not be combined with other options
        Default field names for -t and -c are RiboDock score field names
        Default ID field name is Name

        If sdFiles not given, reads from standard input
        Output is to standard output
```

11.1.10 sdsplit

sdsplit – Splits SD records into multiple files of equal size

```
Usage:  sdsplit [-<RecSize>] [-o<OutputRoot>] [sdFiles]

        -<RecSize>      record size to split into (default = 1000 records)
        -o<OutputRoot>  Root name for output files (default = tmp)

        If SD file list not given, reads from standard input
```

11.1.11 sdsort

Sorts SD records by given data field

```
Usage:  sdsort [-n] [-r] [-f<DataField>] [sdFiles]

        -n                numeric sort (default is text sort)
        -r                descending sort (default is ascending sort)
        -f<DataField>     specifies sort field
        -s                fast mode. Sorts the records for each named compound
                          independently (must be consecutive)
        -id<NameField>    specifies compound name field (default = 1st title line)

Note:   _REC (record #) is provided as a pseudo-data field

        If SD file list not given, reads from standard input
        Output is to standard output
        Fast mode can be safely used for partial sorting of huge SD files of
        raw docking hits without running into memory problems.
```

11.1.12 sdmodify

Script to set the first title line equal to a given data field

```
Usage:  sdmodify -f<DataField> [sdFiles]

         If sdFiles not given, reads from standard input
         Output is to standard output
```

11.1.13 rbhtfinder

Script that simulates the result of a high throughput protocol.

```
1st) exhaustive docking of a small representative part of the
     whole library.
2nd) Store the result of sdreport -t over that exhaustive dock.
     in file <sdreport_file> that will be the input of this
     script.
3rd) rbhtfinder <sdreport_file> <output_file> <thr1max> <thr1min> <ns1> <ns2>
     <ns1> and <ns2> are the number of steps in stage 1 and in
     stage 2. If not present, the default values are 5 and 15
     <thrmax> and <thrmin> setup the range of thresholds that will
     be simulated in stage 1. The threshold of stage 2 depends
     on the value of the threshold of stage 1.
     An input of -22 -24 will try protocols:
            5   -22     15      -27
            5   -22     15      -28
            5   -22     15      -29
            5   -23     15      -28
            5   -23     15      -29
            5   -23     15      -30
            5   -24     15      -29
            5   -24     15      -30
            5   -24     15      -31
     Output of the program is a 7 column values. First column
     represents the time. This is a percentage of the time it
     would take to do the docking in exhaustive mode, i.e.
     docking each ligand 100 times. Anything
     above 12 is too long.
     Second column is the first percentage. Percentage of
```

```
ligands that pass the first stage.
Third column is the second percentage. Percentage of
ligands that pass the second stage.
The four last columns represent the protocol.
All the protocols tried are written at the end.
The ones for which time is less than 12%, perc1 is
less than 30% and perc2 is less than 5% but bigger than 1%
will have a series of *** after, to indicate they are good choices
WARNING! This is a simulation based in a small set.
The numbers are an indication, not factual values.
```

An example file would look like as follows:

```
#3 steps as the running filters (set by the "3" in next line)
3
if - -10 SCORE.INTER 1.0 if - SCORE.NRUNS 9 0.0 -1.0,
if - -20 SCORE.INTER 1.0 if - SCORE.NRUNS 14 0.0 -1.0,
if - SCORE.NRUNS 49 0.0 -1.0,
#1 writing filter (defined by the "1" in next line)
1
- SCORE.INTER -10,
```

In other (more understandable) words:
First, rDock runs 3 consecutive steps:

1. Run 10 runs and check if the SCORE.INTER is lower than -10, if it is the case:
2. Then run 5 more runs (until 15 runs) to see if the SCORE.INTER reaches -20. If it is the case:
3. Run up to 50 runs to freely sample the different conformations the molecule displays.

And, second:
For the printing information, only print out all those poses where SCORE.INTER is better than -10 (for avoiding excessive printing).

11.1.14 rblist

rblist - output interaction center info for ligands in SD file (with optional autoionisation)

```
Usage:  rblist -i<InputSDFile> [-o<OutputSDFile>] [-ap] [-an] [-allH]

Options:        -i<InputSDFile> - input ligand SD file
                -o<OutputSDFile> - output SD file with descriptors
                (default=no output)
                -ap - protonate all neutral amines, guanidines, imidazoles
                (default=disabled)
                -an - deprotonate all carboxylic, sulphur and phosphorous acid
                groups (default=disabled)
                -allH - read all hydrogens present (default=polar hydrogens only)
                -tr - rotate all 2ndry amides to trans (default=leave alone)
                -l - verbose listing of ligand atoms and rotable bonds
                (default = compact table format)
```

12 Common Use cases

This section does not pretend to be a comprehensive User Guide. It does, however, highlight the key steps the user must take for different docking strategies, and may serve as a useful checklist in writing such a guide in the future.

12.1 Standard docking

By standard docking, we refer to docking of a flexible, untethered ligand to a receptor in the absence of explicit structural waters or any experimental restraints.

12.1.1 Standard docking workflow

1. Prepare a MOL2 file for the protein or nucleic acid target, taking into account the atom typing issues described above for MOL2 file parsing. The recommendation is to prepare an all-atom MOL2 file and allow rDock to remove the non-polar hydrogens on-the-fly.
   Important Make sure that any non-standard atom names and substructure names are defined in $RBT_ROOT/data/sf/RbtIonicAtoms.prm in order for the assignment of distributed formal charges to work correctly. Make sure that the Tripos atom types are set correctly. rDock uses the Tripos types to derive other critical atomic properties such as atomic number and hybridisation state.
   Note The rDock MOL2 parser was developed to read the CCDC/Astex protein.mol2 files, therefore this validation set is the de facto standard reference. You should compare against the format of the CCDC/Astex MOL2 files if you are in doubt as to whether a particular MOL2 file is suitable for rDock

2. Prepare a system definition file. At a minimum, you need to define the receptor parameters, the cavity mapping parameters (SECTION MAPPER) and the cavity restraint penalty (SECTION CAVITY). Make sure you define the RECEPTOR_FLEX parameter if you wish to activate sampling of terminal OH and NH3+ groups in the vicinity of the docking site.

3. Generate the docking site (.as) file using rbcavity. You will require a reference bound ligand structure in the coordinate space of the receptor if you wish to use the reference ligand cavity mapping method.

4. Prepare the ligand SD files you wish to dock, taking into account the atom typing issues described above for SD file parsing. In particular, make sure that formal charges and formal bond order are defined coherently so that there are no formal valence errors in the file. rDock will report any perceived valence errors but will dock the structures anyway. Note that rDock never samples bond lengths, bond angles, ring conformations, or non-rotatable bonds during docking so initial conformations should be reasonable.

5. Run a small test calculation to check that the system is defined correctly. For example, run rbdock from the command line with a small ligand SD file, with the score-only protocol (-p score.prm) and with the -T 2 option to generate verbose output. The output will include receptor atom properties, ligand atom properties, flexibility parameters, scoring function parameters and docking protocol parameters.

6. When satisfied, launch the full-scale calculations. A description of the various means of launching rDock is beyond the scope of this guide.

12.2 Tethered scaffold docking

In tethered scaffold docking, the ligand poses are restricted and forced to overlay the substructurecoordinates of a reference ligand. The procedure is largely as for standard docking, except that:

- Ligand SD files must be prepared with the rbtether utility to annotate each record with the matching substructure atom indices, and to transform the coordinates of each ligand so that the matching substructure coordinates are overlaid with the reference substructure coordinates. This requires a Daylight SMARTS toolkit license.

- The system definition file should contain a SECTION LIGAND to define which of the the ligand degrees of freedom should be tethering to their reference values. Tethering can be applied to position, orientation and dihedral degrees of freedom independently. Note that the tethers are applied directly within the chromosome representation used by the search engine (where they affect the randomisation and mutation operators), and therefore external restraint penalty functions to enforce the tethers are not required.

Important The reference state values for each tethered degree of freedom are defined directly from the initial conformation of each ligand as read from the input SD file, and not from the reference SD file used by rbtether. This is why the ligand coordinates are transformed by rbtether, such that each ligand record can act as its own reference state. The reference SD file used by rbtether is not referred to by the docking calculation itself.
It follows from the above that tethered ligand docking is inappropriate for input ligand SD files that have not already been transformed to the coordinate space of the docking site, either by rbtether or by some other means.

12.2.1 Example ligand definition for tethered scaffold

This definition will tether the position and orientation of the tethered substructure, but will allow free sampling of ligand dihedrals.

```
SECTION LIGAND
        TRANS_MODE TETHERED
        ROT_MODE TETHERED
        DIHEDRAL_MODE FREE
        MAX_TRANS 1.0
        MAX_ROT 30.0
END_SECTION
```

12.3 Docking with pharmacophore restraints

In pharmacophore restrained docking, ligand poses are biased to fit user-defined pharmacophore points. The bias is introduced through the use of an external penalty restraint, which penalises docking poses that do not match the pharmacophore restraints. Unlike tethered scaffold docking, there is no modification to the chromosome operators themselves, hence the search can be inefficient, particularly for large numbers of restraints and/or for ligands with large numbers of matching features. Pre-screening of ligands is based purely on feature counts, and not on geometric match considerations.
The implementation supports both mandatory and optional pharmacophore restraints. The penalty function is calculated over all mandatory restraints, and over (any NOPT from N) of the optional restraints. For example, you may wish to ensure that any 4 from 7 optional restraints are satisfied in the generated poses.
The procedure is largely as for standard docking, except that:

- You should prepare separate pharmacophore restraint files for the mandatory and optional restraints. Note that optional restraints do not have to be defined, it is sufficient to only define at least one mandatory restraint.
- The system definition file should contain a SECTION PHARMA to add the pharmacophore restraint penalty to the scoring function.

12.4 Docking with explicit waters

Explicit structural waters can be loaded from an external PDB file, independently from the main receptor model, by adding a SECTION SOLVENT to the system definition file. The user has fine control over the flexibility of each water molecule. A total of 9 flexibility modes are possible, in which the translational and rotational degrees of freedom of each water can be set independently to FIXED, TETHERED, or FREE. Thus, for example, it is possible to define a water with a fixed oxygen coordinate (presumably at a crystallographically observed position), but freely rotating such that the orientation of the water hydrogens can be optimised by the search engine (and can be ligand- dependent).

Note In the current implementation, solvent refers strictly to water molecules, and the format of the water PDB file is very strictly defined. In future implementations it is anticipated that other, larger (and possibly flexible) molecules will be loadable as solvent, and that other file formats will be supported.

Explicit waters workflow

1. Prepare a separate PDB file for the explicit waters according to the format prescribed (the section called Water PDB file format)

2. Add a SECTION SOLVENT to the system definition file and define the relevant flexibility parameters (Table 8.3, Solvent definition parameters). The minimal requirement is to define the FILE parameter.

3. Decide whether you wish to have different per-solvent flexibility modes (defined via the occupancy values and temperature factor values in the PDB file (Table 10.1, Conversion of temperature factor values to solvent flexibility modes)), or whether you wish to have a single flexibility mode applied to all waters (defined via the TRANS_MODE and ROT_MODE values in the SECTION SOLVENT of the receptor .prm file)
   Important If you wish to use per-solvent flexibility modes (that is, you wish to set different modes for different waters) make sure that you do not define TRANS_MODE or ROT_MODE entries in the SECTION SOLVENT as these values will override the per-solvent values derived from the temperature factors in the PDB file.

4. If you have defined any waters with TETHERED translational or rotational degrees of freedom, define MAX_TRANS and/or MAX_ROT values as appropriate (or accept the default values. The tethered ranges are applied to all tethered waters and can not be defined on a per-solvent basis at present.

13 Appendix

Table 13.1: Van der Waals parameters in Tripos 5.2 force field.

Atom Type	R	K	IP	POL	Description
H	1.5	0.042	13.6	4	Non-polar hydrogen
H.P	1.2	0.042	13.6	4	Polar hydrogen
C.3	1.7	0.107	14.61	13.8	C sp3 (0 implicit H)
C.3.H1	1.8	0.107	14.61	16.38	C sp3 (1 implicit hydrogen)
C.3.H2	1.9	0.107	14.61	19.27	C sp3 (2 implicit H)
C.3.H3	2	0.107	14.61	22.47	C sp3 (3 implicit H)
C.2	1.7	0.107	15.62	13.8	C sp2 (0 implicit H)
C.cat	1.7	0.107	15.62	13.8	C sp2 (guanidinium centre)
C.2.H1	1.8	0.107	15.62	16.38	C sp2 (1 implicit hydrogen)
C.2.H2	1.9	0.107	15.62	19.27	C sp2 (2 implicit H)
C.ar	1.7	0.107	15.62	13.8	C aromatic (0 implicit H)
C.ar.H1	1.8	0.107	15.62	16.38	C aromatic (1 implicit hydrogen)
C.1	1.7	0.107	17.47	13.8	C sp (0 implicit H)
C.1.H1	1.8	0.107	17.47	16.38	C sp (1 implicit hydrogen)
N.4	1.55	0.095	33.29	8.4	N sp3+ (cationic)
N.3	1.55	0.095	18.93	8.4	N sp3
N.pl3	1.55	0.095	19.72	8.4	N trigonal planar (non-amide)
N.am	1.55	0.095	19.72	8.4	N trigonal planar (amide)
N.2	1.55	0.095	22.1	8.4	N sp2
N.ar	1.55	0.095	22.1	8.4	N aromatic
N.1	1.55	0.095	23.91	8.4	N sp
O.3	1.52	0.116	24.39	5.4	O sp3
O.2	1.52	0.116	26.65	5.4	O sp2
O.co2	1.52	0.116	35.12	5.4	O carboxylate
S.3	1.8	0.314	15.5	29.4	S sp3
S.o	1.7	0.314	15.5	29.4	sulfoxide
S.o2	1.7	0.314	15.5	29.4	sulfone
S.2	1.8	0.314	17.78	29.4	S sp2
P.3	1.8	0.314	16.78	40.6	
F	1.47	0.109	20.86	3.7	
Cl	1.75	0.314	15.03	21.8	
Br	1.85	0.434	13.1	31.2	
I	1.98	0.623	12.67	49	
Na	1.2	0.4			
K	1.2	0.4			
UNDEFINED	1.2	0.042			

R = radius (Å); K = well depth (kcal/mol); IP = Ionization potential (eV); POL = polarisability (10^{25} cm^3).

Table 13.2: Geometrical parameters for empirical terms.

Term	$\mathbf{X}^a$	$\mathbf{X}_0^b$	$\mathbf{X}_{min}^c$	$\mathbf{X}_{max}^d$	Description
S_{polar}	R_{12}	R + 0.05Å	0.25Å	0.6Å	Distance between interaction centres
	α_{DON}	180°	30°	80°	Angle around donor H
	αACC	180°	60°	100°	Angle around acceptor
	α_{C+}	180°	60°	100°	Angle between C+ACC vector and normal to plane of guanidinium group
	ϕ_{ACC_LP}	45°	15°	15°	From [refrdock] Figure 2.
	θ_{ACC_LP}	0°	20°	60°	From [refrdock] Figure 2.
	ϕ_{ACC_PLANE}	0°	60°	75°	From [refrdock] Figure 2.
	θ_{ACC_PLANE}	0°	20°	60°	From [refrdock] Figure 2.
Srepul	R_{12}	R + 1.1Å	0.25Å	0.6Å	Distance between interaction centres
	α_{DON}	180°	30°	60°	Angle around donor H
	α_{ACC}	180°	30°	60°	Angle around acceptor
Sarom	R_{perp}	3.5Å	0.25Å	0.6Å	From ref [] Figure 3
	α_{Slip}	0°	20°	60°	From ref [] Figure 3

a = Geometric variable; b = Ideal value; c = Tolerance on ideal value; d = Deviation at which score is reduced to zero.

Table 13.3: Angular functions used to describe attractive and repulsive polar interactions.

$\mathbf{IC1}^a$	$\mathbf{ANG}_{IC1}^b$	$\mathbf{IC2}^a$	$\mathbf{ANG}_{IC2}^b$
Attractive (S_{polar})			
DON	$f_1(\lvert\Delta\alpha_{DON}\rvert)$	ACC_LP	$f_1(\lvert\Delta\phi_{ACC\_LP}\rvert) \cdot f_1(\lvert\Delta\theta_{ACC\_LP}\rvert)$
DON	$f_1(\lvert\Delta\alpha_{DON}\rvert)$	ACC_PLANE	$f_1(\lvert\Delta\phi_{ACC\_PLANE}) \cdot f_1(\lvert\Delta\theta_{ACC\_PLANE}\rvert)$
DON	$f_1(\lvert\Delta\alpha_{DON}\rvert)$	ACC	$f_1(\lvert\Delta\alpha_{ACC}\rvert)$
M+	1	ACC_LP	$f_1(\lvert\Delta\phi_{ACC\_LP}) \cdot f_1\lvert(\Delta\theta_{ACC\_LP}\rvert)$
M+	1	ACC_PLANE	$f_1(\lvert\Delta\phi_{ACC\_PLANE}\rvert) \cdot f_1(\lvert\Delta\theta_{ACC\_PLANE}\rvert)$
M+	1	ACC	$f_1(\lvert\Delta\alpha_{ACC}\rvert)$
		ACC_LP	
C+	$f_1(\lvert\Delta\alpha_{C+}\rvert)$	ACC_PLANE	$f_1(\lvert\Delta\alpha_{ACC}\rvert)$
		ACC	
Repulsive (S_{repul})			
DON	$f_1(\lvert\Delta\alpha_{DON}\rvert)$	DON	$f_1(\lvert\Delta\alpha_{DON}\rvert)$
DON	$f_1(\lvert\Delta\alpha_{DON}\rvert)$	M+	1
DON	$f_1(\lvert\Delta\alpha_{DON}\rvert)$	C+	1
M+	1	C+	1
C+	1	C+	1
ACC_LP		ACC_LP	
ACC_PLANE	$f_1(\lvert\Delta\alpha_{ACC}\rvert)$	ACC_PLANE	$f_1(\lvert\Delta\alpha_{ACC}\rvert)$
ACC		ACC	

a = Interaction centre types; b = angular functions in Equations 6-13.

Table 13.4: Solvation parameters (a = Frequency of occurrence in training set).

Atom type	Description	$\mathbf{N}^a$	$\mathbf{r}_i$	$\mathbf{p}_i$	$\mathbf{w}_i$
C_sp3	Apolar carbon sp3 with 0 implicit H	48	1.7	2.149	0.8438
CH_sp3	Apolar carbon sp3 with 1 implicit H	59	1.8	1.276	0.0114
CH2_sp3	Apolar carbon sp3 with 2 implicit H	487	1.9	1.045	0.0046
CH3_sp3	Apolar carbon sp3 with 3 implicit H	409	2	0.88	0.0064
C_sp2	Apolar carbon sp2 with 0 implicit H	10	1.72	1.554	0.0789
CH_sp2	Apolar carbon sp2 with 1 implicit H	45	1.8	1.073	-0.0014
CH2_sp2	Apolar carbon sp2 with 2 implicit H	26	1.8	0.961	0.0095

C_sp2p	Positive charged carbon sp2	2	1.72	1.554	-0.7919
C_ar	Apolar aromatic carbon with 0 implicit H	116	1.72	1.554	0.017
CH_ar	Apolar aromatic carbon with 1 implicit H	357	1.8	1.073	-0.0143
C_sp	Carbon sp	24	1.78	0.737	-0.0052
C_sp3_P	Polar carbon sp3 with 0 implicit H	6	1.7	2.149	-0.0473
CH_sp3_P	Polar carbon sp3 with 1 implicit H	22	1.8	1.276	-0.0394
CH2_sp3_P	Polar carbon sp3 with 2 implicit H	130	1.9	1.045	-0.0078
CH3_sp3_P	Polar carbon sp3 with 3 implicit H	69	2	0.88	0.0033
C_sp2_P	Polar carbon sp2 with 0 implicit H	57	1.72	1.554	-0.2609
CH_sp2_P	Polar carbon sp2 with 1 implicit H	30	1.8	1.073	-0.0227
CH2_sp2_P	Polar carbon sp2 with 2 implicit H	1	1.8	0.961	-0.005
C_ar_P	Polar aromatic carbon with 0 implicit H	53	1.72	1.554	0.0759
CH_ar_P	Polar aromatic carbon with 1 implicit H	34	1.8	1.073	-0.0015
H	Explicit apolar hydrogen (not used)	0	1.2	1	0
HO	Polar hydrogen bonded to O	54	1	0.944	0.0499
HN	Polar hydrogen bonded to N	54	1.1	1.128	-0.0242
HNp	Positively charged polar hydrogen bonded to N	23	1.2	1.049	-1.9513
HS	Polar hydrogen bonded to S	4	1.2	0.928	0.0487
O_sp3	Ether oxygen	31	1.52	1.08	-0.138
OH_sp3	Alcohol/phenol oxygen	48	1.52	1.08	-0.272
O_tri	Ester oxygen	59	1.52	1.08	0.0965
OH_tri	Acid oxygen (neutral)	6	1.52	1.08	-0.0985
O_sp2	Oxygen sp2	83	1.5	0.926	-0.1122
ON	Nitro group oxygen	18	1.5	0.926	-0.0055
Om	Negatively charged oxygen (carboxylate etc)	7	1.7	0.922	-0.717
N_sp3	Nitrogen sp3 with 0 attached H	8	1.6	1.215	-0.6249
NH_sp3	Nitrogen sp3 with 1 attached H	11	1.6	1.215	-0.396
NH2_sp3	Nitrogen sp3 with 2 attached H	11	1.6	1.215	-0.215
N_sp3p	Nitrogen sp3+	6	1.6	1.215	-0.1186
N_tri	Amide nitrogen with 0 attached H	15	1.55	1.028	-0.23
NH_tri	Amide nitrogen with 1 attached H	8	1.55	1.028	-0.4149
NH2_tri	Amide nitrogen with 2 attached H	6	1.55	1.028	-0.1943
N_sp2	Nitrogen sp2	3	1.55	1.413	-0.0768
N_sp2p	Nitrogen sp2+	5	1.55	1.413	-0.2744
N_ar	Aromatic nitrogen	26	1.55	1.413	-0.531
N_sp	Nitrogen sp	6	1.55	1	-0.1208
S_sp3	Sulphur sp3	15	1.8	1.121	-0.0685
S_sp2	Sulphur sp2	5	1.8	1.121	-0.0314
P	Phosphorous	10	1.8	1.589	-1.275
F	Fluorine	99	1.47	0.906	0.0043
Cl	Chlorine	132	1.75	0.906	-0.0096
Br	Bromine	37	1.85	0.898	-0.0194
I	Iodine	9	1.98	0.876	-0.0189
Metal	All metals	0	0.7	1	-1.6667
UNDEFINED	Undefined types	0	1.2	1	0