

RASPD+: Fast protein-ligand binding free energy prediction using simplified physicochemical features

Stefan Holderbach,¹ Lukas Adam,¹ B. Jayaram,³ Rebecca C. Wade,^{1,2,4} and Goutam Mukherjee^{1,2,4}

¹Molecular and Cellular Modelling Group, Heidelberg Institute of Theoretical Studies (HITS),
Schloss-Wolfsbrunnenweg 35, 69118 Heidelberg, Germany

²Center for Molecular Biology (ZMBH), DKFZ-ZMBH Alliance, Heidelberg University, Im
Neuenheimer Feld 282, 69120 Heidelberg, Germany

³Supercomputing Facility for Bioinformatics & Computational Biology, Department of
Chemistry, Kusuma School of Biological Sciences, Indian Institute of Technology Delhi,
Hauz Khas, New Delhi, 110016, India

⁴Interdisciplinary Center for Scientific Computing (IWR), Heidelberg University, Im
Neuenheimer Feld 205, 69120 Heidelberg, Germany

Email: mcmsoft@h-its.org

Version 1.0, Release date: 10 July 2020

What is RASPD+?

RASPD+ (RApid Screening of hit molecules for target proteins via Physicochemical Descriptors+) is a computationally fast protocol for identifying lead-like molecules based on predicted binding free energy against a target protein with a 3D structure and a defined ligand binding pocket. RASPD was originally developed at the Supercomputing Facility for Bioinformatics and Computational Biology, (<http://www.scfbio-iitd.res.in/>), Indian Institutes of Technology Delhi (IITD) by Goutam Mukherjee and B. Jayaram¹, and development continued at Heidelberg Institute for Theoretical Studies (HITS) in the [Molecular and Cellular Modeling group](#). In version 1.0 of the RASPD+ software, new feature like scaffold search was added and several machine learning algorithms were introduced. The model was trained on around 4000 non-metallo protein-ligand complexes retrieved from the PDBBIND refined data set. For details see Ref 5.

Performance of RASPD+

A Pearson correlation coefficient of 0.74 and an RMSE ± 1.86 kcal mol⁻¹ (**Figure 1**) were obtained when predicting binding energies for test sets consisting of 493 completely unseen protein–ligand complexes.^{2,3} . The performance of RASPD+ is comparable with that of other scoring functions like KDeep and other methods⁴ but does not require docking of ligands into protein binding sites. Using this method, it is possible to screen a million molecule library against a target protein of known binding

pocket within a couple of minutes. The RASPD+ code (folder name: RASPDplus) is freely available for download.

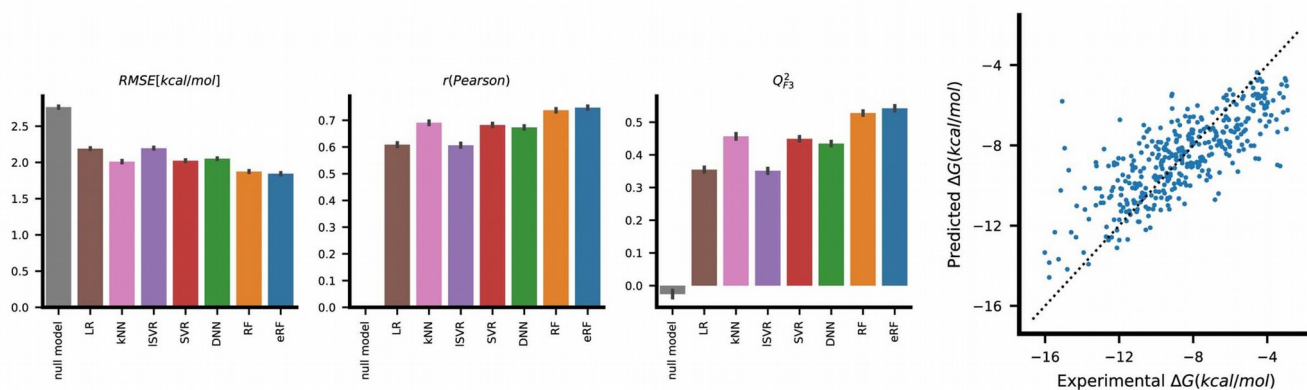


Figure 1: Performance of RASPD+ on 493 non-metallo protein-ligand complexes.

Computation time of RASPD+

Purpose	Run time/ligand	Script
Screening a million-molecule library (after molecules have been parameterized)	1.3 ms	job_run_million.sh
Screening a customized molecule library (after molecules have been parameterized)	1.3 ms	job_run_customized.sh
Generation of physicochemical parameters of small molecules	145 ms	lig_parameters_gen.sh
SMILES translator	255 ms	lig_parameters_gen.sh
SMILES translation + parameter generation + screening	400 ms	single_molecule_scanning.sh
Similarity searches using RDKit	396 min*	scaffolds_search.sh

*Computation time for similarity search of a ligand (query SMILES string) was checked against SMILES codes of million molecules.

The run time of RASPD+ was tested on Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz×12 , 64 bits 32 GB RAM machine.

- (1) Mukherjee, G.; Jayaram, B. A Rapid Identification of Hit Molecules for Target Proteins via Physico-Chemical Descriptors. *Phys. Chem. Chem. Phys.* **2013**, *15* (23), 9107–9116.
- (2) Li, Y.; Su, M.; Liu, Z.; Li, J.; Liu, J.; Han, L.; Wang, R. Assessing Protein--Ligand Interaction Scoring Functions with the CASF-2013 Benchmark. *Nat. Protoc.* **2018**, *13* (4), 666–680.
- (3) Wang, R.; Fang, X.; Lu, Y.; Wang, S. The PDBbind Database: Collection of Binding Affinities for Protein- Ligand Complexes with Known Three-Dimensional Structures. *J. Med. Chem.* **2004**, *47* (12), 2977–2980.
- (4) Jiménez, J.; Skalic, M.; Martinez-Rosell, G.; De Fabritiis, G. K Deep: Protein--Ligand Absolute Binding Affinity Prediction via 3d-Convolutional Neural Networks. *J. Chem. Inf. Model.* **2018**,

58 (2), 287–296.

- (5) Holderbach, S.; Lukas, A.; Jayaram, B.; Wade, R. C.; Mukherjee, G. RASPD+: Fast protein-ligand binding freeenergy prediction using simplifiedphysicochemical features. Manuscript submitted (2020).

Installation instructions

Environment required to run RASPD+: [Linux 64-bit](#), (RASPD+ has been tested under Ubuntu 18.04 LTS)

Machine requirements: [64-bit CPU machine with at least 2 GB RAM and 16 GB available disk space](#).

Step 1: Clone the git repository containing RASPD+ (folder name: RASPDplus)

The size of the repository is ~ **GB**.

You can download/clone the RASPDplus repository from <https://github.com/HITS-MCM/RASPDplus>

The model weights need to be downloaded separately from zenodo

<https://doi.org/10.5281/zenodo.3937426> and placed into the weights directory

Step 2: Download TRAPP from <https://www.h-its.org/downloads/trapp/> and install it according to its instructions

RASPD+ needs the package manager, conda, to manage dependencies for the python components.

Therefore,

Step 3: is to install miniconda in your local machine (skip this step if you have miniconda or anaconda already installed):

- Download the current version of the miniconda installer from:
<https://docs.conda.io/en/latest/miniconda.html#linux-installers>
- Install miniconda on your machine with the following command:
`bash Miniconda3-latest-Linux-x86_64.sh`

Step 4: You need to inform the system where RASPDplus, conda and the TRAPP code are. This can be done by editing init.sh file

Location of init.sh: <path_to_RASPDplus_repository>/config/init.sh

In either case set the three variables as follows:

raspd_root: path of the cloned git repository containing RASPD+

conda_root: path to your conda installation (e.g. /home/your_user_name/miniconda)

TRAPP: path of the downloaded repository containing TRAPP

Step 5: source <path_to_RASPDplus_repository>/config/init.sh

Finally, go to the directory of the git repository:

Step 6: cd <path_to_RASPDplus_repository>

and run

bash install.sh

This will create the necessary conda environments, download the python dependencies, and compile the included C/C++ code.

How to run the scripts

Script name	How to run
job_run_million.sh	bash job_run_million.sh 1NHZ 486 erf
lig_parameters_gen.sh and job_run_customized.sh	bash lig_parameters_gen.sh molecules.txt and then, bash job_run_customized.sh 1NHZ 486 erf
single_molecule_scanning.sh	bash single_molecule_scanning.sh molecules.txt 1NHZ.pdb 486 all or, bash single_molecule_scanning.sh lig.mol2 1NHZ.pdb 486 all or, bash single_molecule_scanning.sh lig.sdf 1NHZ.pdb 486 all or, bash single_molecule_scanning.sh lig.pdb 1NHZ.pdb 486 all

Purpose	Script name	Protein Structure	Active site Identifier	Ligand structure	Ligand File format
Screening of an existing library	job_run_million.sh	Required e.g.; 1NHZ.pdb	Required e.g.; 486	Not required	--
Screening of a customized library	lig_parameters_gen.sh and job_run_customized.sh	Required e.g.; 1NHZ.pdb	Required e.g.; 486	Required e.g.; molecules.txt	SMILES
Screening of single/small-molecule dataset (text file)	single_molecule_scanning.sh	Required e.g.; 1NHZ.pdb	Required e.g.; 486	Required e.g.; molecules.txt or, lig.pdb or, lig.mol2 or, lig.sdf	SMILES, pdb, mol2 and sdf

All these four scripts including scaffolds_search.sh are available at the following location
<path_to_RASPDplus_repository>/scripts/copy/ folder.

*** Screening the existing library ***

Existing million molecule library was prepared by downloading the molecules from the ZINC v12 DATABASE (<http://zinc12.docking.org/>; Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. ZINC: A Free Tool to Discover Chemistry for Biology. *J. Chem. Inf. Model.* **2012**, *52*, 1757-1768) from ChEMBL vendor.

- Script name: "**job_run_million.sh**"
- Location of the script:

[<path_to_RASPDplus_repository>/scripts/copy/job_run_million.sh](#)

Please note that the [<path_to_RASPDplus_repository>](#) location and the script execution directory should be different.

➤ How to run:

First copy the "job_run_million.sh" file to a current working directory (anywhere other than the [<path_to_RASPDplus_repository>](#) location).

Please note that in this directory the protein-ligand complex which was downloaded from the RCSB protein data bank, must be present. Scanning of a million molecules against this protein will be carried out

➤ Command:

```
bash job_run_million.sh <protein-4-letter-code (without ".pdb" extension)> <ligand-3-letter-code> <method>
```

Please note that "protein pdb file" file must be present **in the same directory** where the script, "job_run_million.sh" is executed.

For example, if the protein-4-letter-code is 1NHZ.pdb and the three-letter code of ligand (residue name) that is present in this protein pocket (active site) is 486, then,

```
bash job_run_million.sh 1NHZ 486 erf
```

Here,

"erf" is the machine learning method.

If you want to change the default range of physicochemical parameters or the cut-off binding free energy, you can edit the "select_parameter.txt" file before running the script, [job_run_million.sh](#). The default cut-off binding free energy value is +1000 kcal/mol.

➤ Location of the file: [<path_to_RASPDplus_repository>/data/select_parameter.txt](#)

➤ Output of the script, job_run_million.sh:

FinalResult.txt (Contains predicted binding free energies of the million molecules)

target.smi (Contains SMILES Code of the million molecules)

"select_parameter.txt" file contains the following range of parameters.

Number_of_Cores: 32 (up to 32 cores)

Wiener_Index_Minimum_Range: 0

Wiener_Index_Maximum_Range: 5000000

H-Bond-Donor_Minimum_Range: 0

H-Bond-Donor_Maximum_Range: 10000

H-Bond_Acceptor_Minimum_Range: 0

H-Bond_Acceptor_Maximum_Range: 10000

LogP_Minimum_Range: -1000

LogP_Maximum_Range: 1000

Molar_Refractivity_Minimum_Range: 0
Molar_Refractivity_Maximum_Range: 10000
Molecular_Weight-Minimum_Range: 0
Molecular_Weight-Maximum_Range: 10000
Predicted_Binding_Energy: 1000

* **Screening of a customized library (>100 molecules)** *

There are two steps.

- Script names: "**lig_parameters_gen.sh**" and "**job_run_customized.sh**"
- **Step-1:** Generate the parameters for the customized small molecules. The parameters will be saved automatically in the **<path_to_RASPDplus_repository>/customized_data/** directory.
- The script to generate the parameters for customized small molecules is "**lig_parameters_gen.sh**" and is located in **<path_to_RASPDplus_repository>/scripts/copy/lig_parameters_gen.sh**

Please note that the <path_to_RASPDplus_repository> location and the script execution directory should be different.

N.B.: The file format for the customized small molecules is **SMILES** (https://en.wikipedia.org/wiki/Simplified_molecular-input_line-entry_system).

All the SMILES codes MUST BE STORED in a single file with the file extension "***.txt**".

For example, "molecules.txt" is a file that contains the following SMILES codes:

CC[NH2+]CC[C@@H]1CCC(=O)N1Cc1cccc1C

CC(=O)Nc1nc2c(s1)cccc2C

c1ccc(cc1)SC1=CS(=O)(=O)CC1

C=CCn1nc(nn1)NC(=O)c1cccc(c1)F

This *.txt file MUST BE present in the current working directory where the job will be executed by a script named "lig_parameters_gen.sh".

This SMILES code is converted to *.pdb format by an Open Babel command. Please note that Open Babel generates different conformations of a molecule for the same SMILES string if it is run more than once. Thus, the value of D_{\max} (the maximum distance of an atom in a ligand from its center of mass) of the ligand will change and this may affect the final scoring.

- Command to run:

```
bash lig_parameters_gen.sh molecules.txt
```

The parameters will be saved automatically in the following directory

```
<path_to_RASPDplus_repository>/customized_data/
```

- **Step-2:** Run the “**job_run_customized.sh**” script to estimate the binding affinities of the customized small molecules against the target protein.
- Location of the script:

```
<path_to_RASPDplus_repository>/scripts/copy/job_run_customized.sh
```

Please note that the `<path_to_RASPDplus_repository>` location and the script execution directory should be different.

- Command to run:

```
bash job_run_customized.sh <protein-4-letter-code (without “.pdb” extension)> <ligand-3-letter-code>  
<method>
```

Please note that the “*.pdb” file must be present where the script “**job_run_customized.sh**” is executed.

For example, if the protein-4-letter code is 1NHZ.pdb and the three-letter code of the ligand (residue name) which is present in this protein is 486, then,

```
bash job_run_customized.sh 1NHZ 486 erf
```

Here,

“erf” is the machine learning method.

If you want to change the default range of physicochemical parameters or cut-off binding free energy, you can edit the “select_parameter.txt” file before running the script, `job_run_customized.sh`. The default cut-off binding free energy value is zero. This means that all the predicted binding free energies that have values of 0 or less will be selected and the rest will be discarded.

- Location of the file: `<path_to_RASPDplus_repository>/data/select_parameter.txt`
- Output of the script, `job_run_customized.sh`:
 - FinalResult.txt** (Contains predicted binding free energies of the query molecules)
 - target.smi** (Contains SMILES codes of the query small molecules)

“select_parameter.txt” file contains the following range of parameters .

```
Number_of_Cores: 32 (up to 32 cores)
```

```
Wiener_Index_Minimum_Range: 0
```

```
Wiener_Index_Maximum_Range: 5000000
```


H-Bond-Donor_Minimum_Range: 0
H-Bond-Donor_Maximum_Range: 10000
H-Bond_Acceptor_Minimum_Range: 0
H-Bond_Acceptor_Maximum_Range: 10000
LogP_Minimum_Range: -1000
LogP_Maximum_Range: 1000
Molar_Refractivity_Minimum_Range: 0
Molar_Refractivity_Maximum_Range: 10000
Molecular_Weight-Minimum_Range: 0
Molecular_Weight-Maximum_Range: 10000
Predicted_Binding_Energy: 1000

* **Screening a single/small-dataset (<~100) of molecules**

- The name of the script: "**single_molecule_scanning.sh**"
- The input file format for single molecule screening is ***.txt**

Here, the ***.txt** file contains one or several SMILES codes of the small molecules.

Additionally, *.pdb, *.sdf or *.mol2 format can be used instead for single molecule affinity prediction against a protein target of interest. Please note that the "***.txt, *.pdb, *.sdf or *.mol2**" file must be present where the script "**single_molecule_scanning.sh**" is executed.

This SMILES code/*.sdf/*.mol2 file is converted to *.pdb format by using an Open Babel command. Please note that Open Babel generates different conformations of a molecule for the same SMILES string if run more than once. Thus, the value of D_{\max} (the maximum distance of an atom in a ligand from its center of mass) of the ligand will change and this may affect the final scoring.

- Script location:

<path_to_RASPDplus_repository>/scripts/copy/single_molecule_scanning.sh

Please note that the <path_to_RASPDplus_repository> location and the script execution directory should be different.

- Command to run:

bash single_molecule_scanning.sh <ligand.pdb> <Protein.pdb> <Identifier ID> <Method name>

For example, if ligand.pdb is lig.pdb

Protein.pdb is 1NHZ.pdb

Identifier ID is 486

Methods is erf (say), then the command is:

bash single_molecule_scanning.sh lig.pdb 1NHZ.pdb 486 erf

Please note that all the input files MUST BE in the current working directory.

All the jobs will be executed in this folder.

- Output of the script, single_molecule_scanning.sh:

FinalResult.txt (Contains the predicted binding free energies of the query molecules)

```
*****  
* Searching scaffolds in a database of a million/customized small molecules *  
*****
```

- Script name: "scaffolds_search.sh"
- Script location: <path_to_RASPDplus_repository>/scripts/copy/scaffolds_search.sh
Please note that the location of the script and the script execution folders must be different.
- **Input information needed:**

Full path of JobID, user specified query scaffolds in a file named "scaffolds.txt". **Please note that** "scaffolds.txt" file must be present where the script, "scaffolds_search.sh" is executed.

What are query scaffolds and JobID?

The scaffold search script will run after RASPD+. RASPD+ screens a million or a customized small molecule database against a target protein and the final output are (i) a file that contains the predicted binding free energies (**FinalResult.txt**) and (ii) SMILES codes (**target.smi**) for the small molecules.

The SMILES codes of small molecules may contain several scaffolds/functional groups. If one needs to select an active scaffold from it, the SMILES codes of this query active_scaffold need to be supplied as a file name (**scaffolds.txt**). Please DO NOT give a file name other than scaffolds.txt.

- Command to run:

```
bash scaffolds_search.sh <full path of the JobID>
```

For example,

```
bash scaffolds_search.sh 75171776_1NHZ_486
```

if the location of scaffolds_search.sh and the 75171776_1NHZ_486/ folder is the same. However, if the locations are not same, then provide the full path of the folder,

```
bash scaffolds_search.sh /home/<user_name>/Desktop/75171776_1NHZ_486
```

The Linux command to locate the complete path of a folder/file is:

```
readlink -f <foldername>
```

- Output of the script, scaffolds_search.sh:

target_scaffold.smi (Contains user specified scaffolds that are present in the target.smi file, the million/customized small molecule database)

target_scaffold_be.txt (Contains predicted binding free energies of the query scaffolds)

The machine learning methods available for RASPD+ screening are:

Extremely Random Forest (erf)

Random Forest (rf)

Deep Neural Network (dnn)

k-Nearest neighbors (knn)

linear Support Vector Regression (svr)

non-linear Epsilon Support Vector Regression (esvr)

Linear Regression (lr)

The combination of all seven methods (all)