

# User's Manual of KGS2

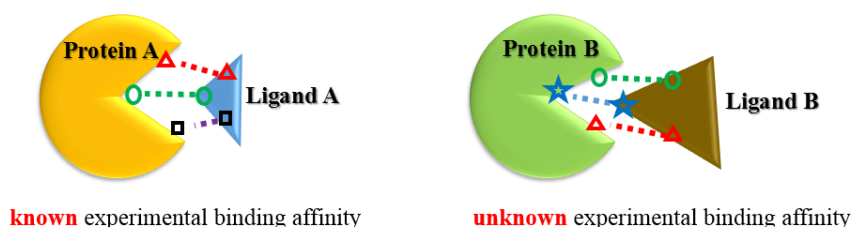
## Contents

Introduction .....	2
How to use KGS2 .....	4
Uncompress the package .....	4
General synopsis for running KGS2 .....	4
Workflow for running KGS2 .....	5
References .....	12

## Introduction

The KGS2 program is developed by Dr. Jie Liu in Dr. Renxiao Wang's group at the Shanghai Institute of Organic Chemistry, Chinese Academy of Sciences.

KGS2 is a software patch of scoring functions, which has its objective to improve the prediction accuracy of scoring functions. Our basic assumption is that molecular systems with similar structures have similar properties, a strategy that has been applied successfully to the computation of some physicochemical properties such as partition coefficient and water solubility. Accordingly, the unknown binding affinity of a given complex can be estimated more reliably from the known binding affinity of a reference complex, which shares a similar pattern of protein-ligand **interactions** with the query complex.



**Figure 1.** The query complex (B) and reference complex (A) share a similar pattern of protein-ligand interactions (different shapes of black marks represent different types of protein-ligand interactions)

The binding scores provided by a reasonable scoring function should correlate well with experimentally determined binding data as follows:

$$\hat{R}_{bind} = b + k \times R_{score,SF} \quad (1)$$

Here,  $\hat{R}_{bind}$  denotes for the expected binding affinity of a reference protein-ligand complex ( $R$ );  $R_{score,SF}$  denotes for the binding score of this complex calculated by a scoring function  $SF$ ; while  $b$  and  $k$ , respectively, are the intercept and the slope of the regression line between the binding scores and experimentally measured binding data of a set of protein-ligand complexes. Similarly, the expected binding affinity of a query protein-ligand complex ( $Q$ ) calculated by the same scoring function is:

$$\hat{Q}_{bind} = b + k \times Q_{score,SF} \quad (2)$$

By subtracting Equation 1 from Equation 2, one has:

$$\hat{Q}_{bind} = \hat{R}_{bind} + k \times (Q_{score,SF} - R_{score,SF}) \quad (3)$$

Replacing the expected binding affinity of  $R$  with the known experimental value ( $R_{exp}$ ), one has:

$$\hat{Q}_{bind} = R_{exp} + k \times (Q_{score,SF} - R_{score,SF}) \quad (4)$$

Equation 4 indicates how the binding affinity of a given protein-ligand complex is computed using the known binding affinity of a proper reference complex as a starting point.

For the convenience of narration, this scoring strategy will be referred to as the KGS2 throughout this article. In principle, any scoring method may be employed to compute the required binding scores of both the reference complex and the query complex in Equation 4. Nevertheless, it is certainly more reasonable in reality to choose a capable scoring method for this purpose. The reference complex can be selected among a database of protein-ligand complexes with reliable structures and binding data. The constant  $k$  in Equation 4 can be derived through a regression analysis between the experimental binding data and the computed binding scores by the employed scoring method on the same database. It is introduced to scale the outcomes of scoring functions, which could be in arbitrary units, to a realistic range comparable to the experimental binding data of the reference complex.

KGS2 is distributed freely to the public. It is currently available at <http://www.sioc-ccbg.ac.cn/software/KGS2/>. Basically, you need to register and sign a license agreement. We will then send you further instructions of how to download this program.

You may direct questions related to this program to the author at: Renxiao Wang, Ph.D.

Copyright of the KGS2 program belongs to the Shanghai Institute of Organic Chemistry, Chinese Academy of Sciences.

## How to use KGS2

The KGS2 program is written in ANSI C++ language and has been tested on LINUX platform. After downloading the program package, please move it to the directory where you would like the program to be ran. Then, use the program through the following three-step procedure.

### Uncompress the package

You can do this in a Linux shell as:

```
tar -xvf KGS2_linux.tar.gz
```

You will get a directory named as "KGS2-process" under your working directory. Under that directory, there are several subdirectories:

"step\_1\_extract\_units/": scripts and files for extracting the interaction units of protein-ligand complexes

"step\_2\_eliminate\_redundancy/": scripts and files for eliminating redundant information in output of step 1

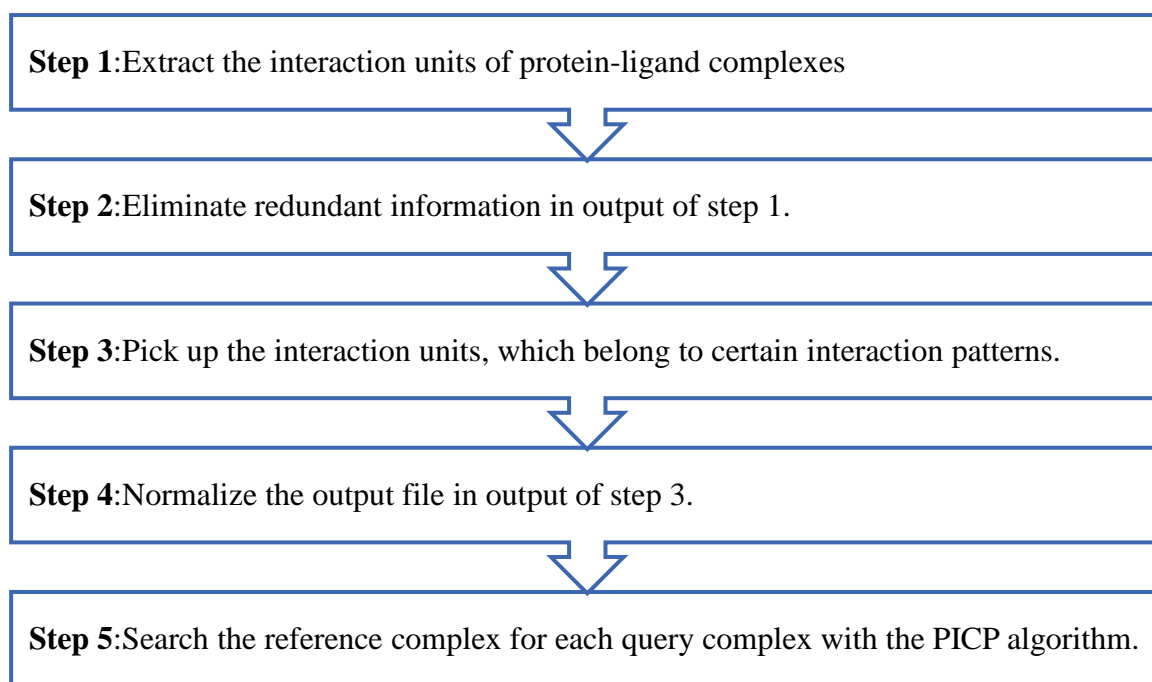
"step\_3\_interaction\_patterns/": scripts and files for picking up the interaction units

"step\_4\_normalize": scripts and files for normalizing the output file of step 3

"step\_5\_search": scripts and files for searching the reference complex for each query complex

### General synopsis for running KGS2

The basic function of KGS2 is to search the reference complexes of a query complex from the specified data set. The standard step by step implementation procedure of KGS2 strategy is assembled in the files and the general procedure is shown in the following figure. The step 1-4 aim to produce the interaction patterns of complexes in a library and step 5 is purpose to search the reference complex for each query complex with the PICP algorithm. Thus, you should run the step1-4 for the first application. You are supposed to edit the scripts and related files to meet your own purpose. In addition, R program is required and please install the R program (<https://www.r-project.org/>) before running KGS2.



**Figure 2.** The general procedure of KGS2

```

rm(list=ls());
setwd("../step_2_eliminate_redundancy/step_2_output/");
data_std<-read.table("../inter_pats.txt");
data_example<-read.table("2osm.ed.txt");
names(data_example)<-c("residue_id", "chain", "inter_name", "patom_1_x", "patom_1_y", "patom_1_z", "patom_2_x", "patom_2_y", "patom_2_z", "patom_3_x", "patom_3_y", "patom_3_z", "latom_x", "latom_y", "latom_z", "C2_X", "C2_Y", "C2_Z", "C1_X", "C1_Y", "C1_Z");
data_example$vector_1_x=(data_example$patom_1_x-data_example$patom_2_x);
data_example$vector_1_y=(data_example$patom_1_y-data_example$patom_2_y);
data_example$vector_1_z=(data_example$patom_1_z-data_example$patom_2_z);
  
```

**Figure 3.** The example input file of KGS2

Note:

Scripts are run in working directory of each step, e.g. step\_1\_extract\_units. Remember to change the absolute directory path of structure files in each script. And all of the lines started with a "#" sign in the input file will be considered as a comment line and is neglected by the program

## Workflow for running KGS2

### ➤ Step 1

Function	Extract all the interaction units of protein-ligand complexes in PDBbind general set v2014 (which contains 10656 complexes) or your own reference library
Working directory	./step_1_extract_units/
The directory of script program	./script_1/

The directory of input files	./step_1_input (Please download the PDBbind general set v2014 or move your own reference library into the working directory. The protein file (XXXX_protein.pdb) and ligand file (XXXX_ligand.mol2) from the same complex should be put in a same folder named with the PDB code (XXXX)).
The directory of output files	./step_1_output
Command	\$ ./run_01.sh (The batch processing script 'run_01.sh' can generate interaction units for each protein-ligand complex in the library. Remember to copy the 'run_01.sh' and the executable file 'xscore' into the directory of input files.)
Result	For the default reference library, 51 of 10656 protein-ligand complexes in the PDBbind general set v2014 do not have structure files. Failed to detect binding pocket for complex (3zyb).
Additional note	The executable file 'xscore' was compiled from source code in ./step_1/script_1/extract/ The usage of the executable file 'xscore' : \$ ./xscore -score protein.pdb ligand.mol2 > output_file

## ➤ Step 2

Function	Eliminate redundant information in output of step 1.
Working directory	./step_2_eliminate_redundancy/
The directory of script program	./script_2/
The directory of input files	./step_2_input (The input files in this folder are obtained and copy from ../step_1_extract_units/step_1_output )
The directory of output files	./step_2_output
Command	\$ ./run_02.sh (The perl script 'process_02.pl' and batch processing script 'run_02.sh' were used to keep the standard protein-ligand interaction units information generated in step 1. Remember to copy the 'run_02.sh' and 'process_02.pl' into the directory of input directory.)

7	A	TYRC5C4C3C.3		14.036	3.095	29.363	14.713	3.025	30.580	15.202
1.810		31.064	10.726	5.484	28.206	13.846	1.938	28.621	15.226	2.331
7	A	TYRC5C4C3S.3		14.036	3.095	29.363	14.713	3.025	30.580	15.202
1.810		31.064	11.291	4.524	26.810	13.846	1.938	28.621	15.226	2.331
7	A	TYRC5C4C3C.ar		14.036	3.095	29.363	14.713	3.025	30.580	15.202
1.810		31.064	9.464	2.135	28.133	13.846	1.938	28.621	15.226	2.331
7	A	TYRC5C6C5C.3		14.036	3.095	29.363	13.846	1.938	28.621	14.323
0.723		29.083	10.726	5.484	28.206	13.846	1.938	28.621	15.226	2.331
7	A	TYRC5C6C5S.3		14.036	3.095	29.363	13.846	1.938	28.621	14.323
0.723		29.083	11.291	4.524	26.810	13.846	1.938	28.621	15.226	2.331
7	A	TYRC5C6C5C.ar		14.036	3.095	29.363	13.846	1.938	28.621	14.323
0.723		29.083	9.464	2.135	28.133	13.846	1.938	28.621	15.226	2.331
7	A	TYRC5C607C.3		14.036	3.095	29.363	13.846	1.938	28.621	13.180
1.997		27.419	10.726	5.484	28.206	13.846	1.938	28.621	15.226	2.331

**Figure 4.** The output file of step\_2

The following is the definition of each column in output file of step\_2:

=====						
residue_ID	chain	interaction_unit_name(residue patom-1 patom-2 patom-3 latom)				
7	A	TYRC5C4C3C.3				
patom_1_x	patom_1_y	patom_1_z	patom_2_x	patom_2_y	patom_2_z	
14.036	3.095	29.363	14.713	3.025	30.580	
patom_3_x	patom_3_y	patom_3_z	latom_x	latom_y	latom_z	
15.202	1.810	31.064	10.726	5.484	28.206	
beta_C_x	beta_C_y	beta_C_z	alpha_C_x	alpha_C_y	alpha_C_z	
13.846	1.938	28.621	15.226	2.331	33.585	
=====						

Notes: 'patom' stand for protein atom, 'latom' stand for ligand atom. 'beta C' stand for residue's beta atom, 'alpha' stand for residue's alpha carbon atom.

### ➤ Step 3

Function	Pick up the interaction units, which belong to certain interaction patterns.
Working directory	./step_3_interaction_patterns/
The directory of script program	./script_3/
The directory of input files	./step_3_input (The input files in this folder are obtained and copy from ../step_2_eliminate_redundancy/step_2_output)
The directory of output files	./step_3_output
Command	\$ mkdir R_source_files ( make a new folder to save source files of R statistics language and remember to change working directory of R source file 'run_demo.R' to './step_3_output/')  \$ ./process_04.pl list_2 run_demo.R (generate the R source files and the list_2 contains the code name of complexes in library)

	<pre>\$ ls  ./R_source_files  &gt;  list_3  \$ ./process_05.pl  list_3  &gt; run_03.R  \$ /home/Software/R-3.1.1/bin/R (Users are supposed to change the absolute path of R program to launch software)  &gt; rm(list=ls()); &gt; setwd("./"); &gt; source("run_03.R");</pre>
Additional note	<pre>'process_04.pl' --- generate the R source files 'process_05.pl' --- generate the batch file 'run_03.R' 'run_demo.R' --- the template R source files 'run_developed.R' --- obsolete template R source files 'inter_pats.txt' --- the interaction patterns developed on the PDBbind general set v2014</pre>

```
ALAC1CNC.2 1.596719 0.127953 0.490533 0.127953 0.347022 0.474544 0.490533 0.474544 2.514014 0.246624 4.580712 -0.450043
ALAC1CNC.2 1.292571 1.196523 0.048769 1.196523 2.132712 -0.705378 0.048769 -0.705378 1.595534 -3.15694 3.08341 1.001765
ALAC1CNC.2 1.215681 -1.1878 -0.21339 -1.1878 2.384505 -0.547196 -0.21339 -0.547196 1.45074 3.943087 2.880994 0.274594
ALAC1CNC.2 0.778389 -0.327733 -0.660382 -0.327733 4.011773 -0.659938 -0.660382 -0.659938 1.847703 -4.614066 -0.602345 -0.887016
ALAC1CNC.2 0.146917 -0.129909 -0.062149 -0.129909 0.550113 -0.217803 -0.062149 -0.217803 0.466519 4.449914 -2.871075 -0.164762
ALAC1CNC.2 2.287065 0.493031 0.210837 0.493031 1.401362 0.404897 0.210837 0.404897 0.364995 0.228814 1.183229 -3.712547
ALAC1CNC.2 0.666535 0.169677 0.377769 0.169677 0.326547 -0.21672 0.377769 -0.21672 1.061891 4.782112 0.058735 -0.72985
ALAC1CNC.2 3.426925 0.814715 0.217082 0.814715 1.732978 -0.656294 0.217082 -0.656294 0.771823 1.39671 -2.582218 -3.053647
ALAC1CNC.2 1.688475 -0.446342 -0.157217 -0.446342 3.097314 -0.561978 -0.157217 -0.561978 0.44854 1.592032 1.575628 3.496967
ALAC1CNC.2 1.879063 -0.689674 0.143801 -0.689674 2.326911 0.015268 0.143801 0.015268 0.263744 -1.417987 -1.240273 3.641986
```

**Figure 5.** The parameter file (inter\_pats.txt) of step\_3

The following is the definition of each column in file './step\_3\_interaction\_patterns/inter\_pats.txt'

```
=====
Interaction pattern name (residue patom-1 patom-2 patom-3 latom)
ALAC1CNC.2
Covariance matrix of gaussian component :
1.596719 0.127953 0.490533
0.127953 0.347022 0.474544
0.490533 0.474544 2.514014
Mean value of gaussian component :
0.246624 4.580712 -0.450043
=====
```



```

"TYRC5C4C3C.3" 7 "A" 15.226 2.331 33.585 10.726 5.484 28.206
"PHEC2C1NC.ar" 8 "A" 11.87 0.551 33.384 9.464 2.135 28.133
"VALC2C1NC.ar" 10 "A" 11.969 -3.489 29.541 9.464 2.135 28.133
"ARGC3C2C10.co2" 13 "A" 16.152 2.121 24.667 16.578 8.524 23.744
"TRPC3C2C1C.ar" 38 "A" 7.305 6.459 38.203 6.683 5.941 31.441
"LYSC3C4C50.co2" 44 "A" 10.247 12.071 36.901 7.116 9.957 31.433
"GLY0CC1C.2" 50 "A" 11.288 14.033 32.598 7.452 9.433 30.354
"GLNC1C0C.3" 51 "A" 12.215 10.686 31.059 10.472 6.967 27.934

```

**Figure 6.** The output file of step\_3

The following is the definition of each column in output file of step\_3:

```

=====
interaction_unit_name  residue_ID  chain
"TYRC5C4C3C.3"      7          "A"
alpha_C_x    alpha_C_y    alpha_C_z    latom_x    latom_y    latom_z
15.226      2.331      33.585      10.726    5.484      28.206
=====

```

#### ➤ Step 4

Function	Normalize the output file in folder './step_3_interaction_patterns/step_3_output/'.
Working directory	./step_4_normalize/
The directory of script program	./script_4/
The directory of input files	./step_4_input (The input files in this folder are obtained and copy from ../step_3_interaction_patterns/step_3_output/)
The directory of output files	./step_4_output
Command	./run_04.sh (The perl script 'process_06.pl' and batch processing script 'run_05.sh' were used to normalize the data in step 3. Remember to copy the 'run_04.sh' and 'process_06.pl' into the directory of input directory.)

```

C.3      7      A      C1      10.726      5.484      28.206
TYR      7      A      C1      15.226      2.331      33.585
C.ar     8      A      C1      9.464      2.135      28.133
PHE      8      A      C1      11.870      0.551      33.384
C.ar    10      A      C1      9.464      2.135      28.133
VAL     10      A      C1      11.969      -3.489      29.541
0.co2   13      A      C1      16.578      8.524      23.744
ARG     13      A      C1      16.152      2.121      24.667
C.ar    38      A      C1      6.683      5.941      31.441
TRP     38      A      C1      7.305      6.459      38.203
0.co2   44      A      C1      7.116      9.957      31.433
LYS     44      A      C1      10.247      12.071      36.901

```

**Figure 7.** The output file of step\_4

The following is the definition of each column in output file of step\_3:

C.3	7	A	C1	10.726	5.484	28.206
TYR	7	A	C1	15.226	2.331	33.585
C.ar	8	A	C1	9.464	2.135	28.133
PHE	8	A	C1	11.870	0.551	33.384

1<sup>st</sup> column: the atom type of ligand atom (odd rows) or the residue name (even rows)

2<sup>nd</sup> column: the residue number

3<sup>rd</sup> column: the protein chain

4<sup>th</sup> column: the label of using alpha carbon to represent the residue

5<sup>th</sup> column: x coordinates

6<sup>th</sup> column: y coordinates

7<sup>th</sup> column: z coordinates

Each couple of lines represents an interaction pair. While the odd row includes the information of ligand atom, the even row consists of the information of alpha carbon from its interactive protein residue. Notably, the 2nd~4th columns which includes the protein information are also written into the odd row for convenience.

## ➤ Step 5

Function	Search the reference complex for each query complex with the PICP algorithm.
Working directory	./step_5_search/
The directory of script program	./
The directory of input files	./all (The input files in this folder are obtained and copy from './step_4_normalize/step_4_output/')
The output file	./run_out_log
Command	<pre>\$ ./picp.exe ./all/10gs_ok.txt &gt; run_out_log (calculate single complex; in folder './step_5_search/') or \$ ./batch_search.sh (calculate all complexes in PDBbind_general_set; in folder './step_5_search/')  \$ awk '{printf "%s\t%s\t%.2f\n",\$1, \$2, \$4}' ./run_out_log   grep -v "none" &gt; ./result_PICP_general_2014.txt (get the final result, run in folder './step_5_search/')</pre>

11gs	3gss	0.52
13gs	2gss	0.22
184l	186l	0.65
185l	1l83	0.57
186l	184l	0.65
187l	1li3	0.64
188l	1l83	0.68

**Figure 8.** The output file of step\_5

The following is the definition of each column in output file of step\_3:

11gs	3gss	0.52
13gs	2gss	0.22
1 <sup>st</sup> column: query complex (PDB entry)		
2 <sup>nd</sup> column: reference complex (PDB entry)		
3 <sup>rd</sup> column: the similarity between query complex and reference complex computed with Tanimoto method		

## References

- [1] Kota, Kasahara.; Matsuyuki, Shirota.; Kengo, Kinoshita. Comprehensive Classification and Diversity Assessment of Atomic Contacts in Protein–Small Ligand Interactions. *J. Chem. Inf. Model.*, 2013, 53, 241–248
- [2] Tiejun, Cheng.; Zhihai Liu.; Renxiao Wang. A knowledge-guided strategy for improving the accuracy of scoring functions in binding affinity prediction. *BMC Bioinformatics* 2010, 11, 193
- [3] Renxiao, Wang.; Luhua, Lai.; Shaomeng, Wang. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *Journal of Computer-Aided Molecular Design*, 2002, 16, 11–26
- [4] Stefano, Forli.; Arthur J. Olson. A Force Field with Discrete Displaceable Waters and Desolvation Entropy for Hydrated Ligand Docking. *J. Med. Chem.* 2012, 55, 623–638
- [5] Howard, J.F.; Paul, L. Pocket Similarity: Are Carbons Enough? *J.Chem.Inf.Model.* 2010, 50, 1466-1475.