RankIt – Short Manual

U. Fechner, G. Schneider
July 2004

Johann Wolfgang Goethe-Universität, Institut für Organische Chemie und Chemische Biologie
Marie-Curie-Str. 11, D-60439 Frankfurt, Germany. Email: u.fechner@chemie.uni-frankfurt.de

This work was supported by the Beilstein-Institut
zur Förderung der Chemischen Wissenschaften, Frankfurt.

If you wish to publish results obtained using *RankIt*, please cite:

Fechner U, Franke L, Renner S, Schneider P, Schneider G (2003) Comparison of correlation vector methods for ligand-based similarity searching. *J. Comput. Aided Mol. Des*. *17*, 687-698.

1. Background: Chemical Similarity Searching

Chemical similarity searching is a popular approach to identify novel molecules revealing similar biological activity to a query structure by pair-wise compound comparison. The result of a similarity search in a compound database is a ranked list. High-ranking compounds in such a list are assumed to be more similar to the query structure than low-ranking compounds. The similarity index should consider molecules as different which do not share important attributes. The definition of "important" attributes heavily depends on the query structure and therefore on its associated binding partner. Early-phase virtual screening and compound library design often employs similarity searching routines for diversity analysis and the selection of activity-enriched subsets [1]. Ligand-based similarity indices are frequently and successfully used for this purpose [2,3]. Many different approaches have been described, and it is not trivial to select the most appropriate concept for a given task. Basically, these techniques rely on i) representative reference structures (also termed "query" or "seed" structures), ii) molecular descriptors that are correlated with biological activity, and iii) an appropriate similarity index.

The aim of a similarity search can be characterized in one of the following two ways. First, it can be applied with a set of *n* known active molecules. Then, one can evaluate the employed parameters (query structures, descriptor, distance metric) by means of the enrichment factor. This application of a similarity search is called "retrospective screening". In contrast, "prospective screening" can be performed to find molecules that potentially exhibit activity for the same target as the query structure. The decision which specific parameters should be employed for a prospective screen has to be made on prior gathered experience, and retrospective screening provides a useful means for this purpose.

2. Evaluating a Ranked List

The enrichment factor *ef* provides a possibility to rate a similarity search [4]. Given a database containing D_{all} compounds, of which D_{act} have known biological activity against a desired target. A certain fraction *F*, e.g. the top 10 %, is taken from a similarity ranked list. The fraction contains F_{all} compounds of which F_{act} are experimentally validated active. Provided that D_{act} is randomly distributed among D_{all} the expected number of active molecules among F_{all} is

$$F_{act} = F_{all} \frac{D_{act}}{D_{all}} . \qquad \text{(Eq. 1)}$$

Thus, a similarity search can be qualified by calculating the enrichment of active molecules within F_{all} over a random (equal) distribution of the active molecules:

$$ef = \left(\frac{F_{act}}{F_{all}} \right) \Big/ \left(\frac{D_{act}}{D_{all}} \right),$$ where (Eq. 2)

ef is the enrichment factor. An enrichment factor above 1 is returned by a method that is superior to a random selection of compounds within F_{all}. The enrichment factor can be visualized by plotting F_{all} / D_{all} on the x-axis and F_{act} / D_{act} on the y-axis (“enrichment curve”). A well-performing similarity search should result in a curve above the diagonal line.

If the query of a similarity search does not consist of a single molecule but of a set of *n* molecules (known actives), *n* similarity searches are performed. Each active molecule represents the query structure in turn. Pair-wise comparison is performed against the *m* compounds forming the database and the *n-1* actives. Consequently, one obtains *n* ranked lists. The enrichment factor has to be calculated for each of the *n* list. The final enrichment factor of such a similarity search is the average of the enrichment factors of the individual lists. To obtain a final ranked list the compounds are sorted according to their average positions within the individual ranked lists that resulted from the *n* similarity searches.

References

(1) Barnard, J.M., Downs, G.M., and Willett, P. Descriptor-Based Similarity Measures for Screening Chemical Databases, in Böhm, H.J., and Schneider, G. (Eds.) *Virtual Screening of bioactive Molecules*, Wiley-VCH 2000, Weinheim, New York, 59-80.

(2) Schneider, G., and Nettekoven, M. Ligand-based combinatorial design of selective purinergic receptor (A(2A)) antagonists using self-organizing maps. *J. Comb. Chem*., 5 (2003), 233.

(3) Schuffenhauer A., Floersheim P., Acklin P., and Jacoby E. Similarity metrics for ligands reflecting the similarity of the target proteins. *J. Chem. Inf. Comput. Sci*., 43 (2003), 391.

(4) Xu, H., and Agrafiotis, D.K. Retrospect and prospect of virtual screening in drug discovery, *Curr. Top. Med. Chem.*, 2 (2002**)**, 1305.

3. *RankIt*: Distance Indices

In the current version of *RankIt* seven similarity indices are implemented. Equations for continuous variables are given in Table 1.

Table 1. Distance indices for continuous variables that are implemented in *RankIt*. *A* and *B* are molecules, *i* and *j* are molecular descriptors, *n* is the total number of descriptors, x_{jA} the value of the *j*th descriptor of moelcule A, $S_{A,B}$ denotes the similarity between A and B, and $D_{A,B}$ the distance between A and B (adopted from [1]).

Metric / Index	**Equation**	**Range**
Manhattan Distance	$D_{A,B} = \sum_{j=1}^{j=n} \left| x_{jA} - x_{jB} \right|$	8 to 0
Euclidian Distance	$D_{A,B} = \sqrt{\sum_{j=1}^{j=n} \left(x_{jA} - x_{jB} \right)^{2}}$	8 to 0
Tanimoto Coefficient	$S_{A,B} = \dfrac{\sum_{j=1}^{j=n} x_{jA} x_{jB}}{\sum_{j=1}^{j=n} \left(x_{jA} \right)^{2} + \sum_{j=1}^{j=n} \left(x_{jB} \right)^{2} - \sum_{j=1}^{j=n} x_{jA} x_{jB}}$	-0.333 to +1
Soergel Distance	$D_{A,B} = \dfrac{\sum_{j=1}^{j=n} \left| x_{jA} - x_{jB} \right|}{\sum_{j=1}^{j=n} \max(x_{jA}, x_{jB})}$	0 to 1
Cosine Coefficient	$S_{A,B} = \dfrac{\sum_{j=1}^{j=n} x_{jA} x_{jB}}{\sqrt{\sum_{j=1}^{j=n} \left(x_{jA} \right)^{2} \sum_{j=1}^{j=n} \left(x_{jB} \right)^{2}}}$	-1 to +1
Dice Coefficient	$S_{A,B} = \dfrac{2\sum_{j=1}^{j=n} x_{jA} x_{jB}}{\sum_{j=1}^{j=n} \left(x_{jA} \right)^{2} + \sum_{j=1}^{j=n} \left(x_{jB} \right)^{2}}$	-1 to +1
Spherical Distance	$D_{A,B} = a\cos(A_N * B_N)$, where $A_N = \dfrac{A}{\left| A \right|} = \sum_{j=1}^{j=n} \left[\dfrac{x_{jA}}{\sqrt{\sum_{i=1}^{i=n} x_{iA}^{2}}} \right]$ and $B_N = \dfrac{B}{\left| B \right|} = \sum_{j=1}^{j=n} \left[\dfrac{x_{jB}}{\sqrt{\sum_{i=1}^{i=n} x_{iB}^{2}}} \right]$	0 to p

4. RankIt: Usage of the Program

If the program is run without any command line parameters, a detailed help text will be provided.

The following description differentiates between *command line options* and *arguments*. The former are simple switches (e.g., *-h*), whereas the latter need an additional value (e.g., *-m 1*).

The basic usage of the program is as follows:

```
rankIt [Options] < INFILE
```

where *INFILE* has to be a tab separated file with an identifier in the first column and the data (descriptor) values in the following columns. If the input file has no header row one has to use the *-n* option. The end of line identifier of *INFILE* has to be in unix style (*\n*). The name of the output file is specified with the *-o* argument.

-h
Display a detailed help text and exit.

-v
Display detailed version information and exit.

-n
This switch indicates that the input file does not contain a header row. Without this switch it is assumed that the input file contains a header row, i.e. the first row is not considered as a data row.

-a STRING
If *STRING* is a substring of the identifier of a molecule this molecule is considered as “active”.

-o OUTFILE
The name of the output file is specified.

-m INT | [1,7] | DEFAULT = 1
This argument determines the similarity index that is used in the pair-wise comparison of molecules. Values 1 to 7 correspond to the following distances (Table 1):
1 Manhattan Distance
2 Euclidian Distance
3 Tanimoto Coefficient
4 Soergel Distance
5 Dice Coefficient
6 Cosine Coefficient
7 Spherical Distance

-s
Without this option the output file contains extensive information. If this option is given only the summary of the virtual screening will be provided in the output file.

-l INT | [1,100] | DEFAULT = 100
The value of this argument indicates which percentage of each similarity-ranked list is given in the output file. By default, the lists are provided in their entirety.

A typical scenario for the use of *RankIt* is the **calculation of an enrichment factor** based on chemical similarity. Numerical descriptors (e.g., physicochemical properties) of the compounds belonging to both the active (query) and the inactive (database) class have to be calculated and saved as a tab-separated text file. An example of an input file looks like this:

```
Identifier              desc1 desc2       ...   descN
Compound1_GPCR          0.37  0.32        ...   9.12
Compound2_GPCR          4.28  0.00        ...   0.06
...
Compound121_GPCR        2.91  0.81        ...   3.29
Compound122_COX2        2.99  6.34        ...   8.54
Compound123_COX2        0.00  0.81        ...   0.89
...
CompoundM_TargetClass   1.18  9.10        ...   8.43
```

The first line of the input file is the header row. If the input does not contain such a header row the *RankIt* command line option *-n* has to be used. The first column of the input file is the identifier of each compound, the following columns contain the descriptor values. The identifier is employed to differentiate between active (query) and inactive (database) compounds. Thus, the names have to allow for such discrimination. In the above example this is achieved by the incorporation of a target class acronym in the name strings. For example, if the enrichment of compounds that belong to the “GPCR” target class should be calculated the command line argument *-a GPCR* has to be given. Please note that the value of the *-a* option is case-sensitive and does not accept regular expressions. Assuming that the Tanimoto Coefficient is used as a similarity index the complete command to start *RankIt* is:

```
rankIt -o OUTFILE -m 3 -a GPCR <INFILE
```

where OUTFILE and INFILE are the names of the outfile and infile, respectively.

Another scenario is a ligand-based **virtual screening of a database** to identify compounds that are similar to a known ligand against a certain biological target. The beforehand calculated descriptor file of the known ligand and the database have to be merged:

```
Identifier        desc1 desc2 desc3 ...   descN
KnownLigand       0.37  0.32  4.37  ...   9.12
Compound1         4.28  0.00  2.18  ...   0.06
...
CompoundM         1.18  9.10  0.00  ...   8.43
```

where the compound with the identifier *knownLigand* is the query compound and all compounds with the identifier *compound1* to *compoundM* constitute the database. To obtain a list sorted in ascending order with regard to the similarity to the known ligand the following command is used to execute *RankIt*

```
rankIt -o OUTFILE -a knownLignd <INFILE
```

where OUTFILE and INFILE are the names of the outfile and infile, respectively.