Clustering Libraries of Compounds into Families: Asymmetry-Based Similarity Measure to Categorize Small Molecules

Samuel Wieczorek, Samia Aci, Gilles Bisson, Mirta B. Gordon, Laurence Lafanechere, Eric Maréchal, Sylvaine Roy

Chapter Number

Clustering Libraries of Compounds into Families: Asymmetry-Based Similarity Measure to Categorize Small Molecules

Wieczorek Samuel[1], Aci Samia[2], Bisson Gilles[3], Gordon Mirta[3], Lafanechère Laurence[4], Maréchal Eric[5], Roy Sylvaine[5]

[1]*Laboratoire de Biologie à Grande Echelle, iRTSV, CEA Grenoble, 17 rue des Martyrs, 38054 Grenoble , France.*

[2]*Centre de Biophysique Moléculaire, UPR CNRS 4301, rue Charles Sadron, 45071 Orléans, France.*

[3]*Laboratoire Techniques de l'Ingénierie Médicale et de la Complexité - Informatique, Mathématiques et Applications de Grenoble, UMR 5525 CNRS - Université Joseph Fourier Grenoble 1, Domaine de la Merci, 38710 La Tronche, France*

[4]*Institut Albert Bonniot, CRI INSERM U823 - Université Joseph Fourier Grenoble 1, Rond-point de la Chantourne, 38706 La Tronche, France*

[5]*Laboratoire de Physiologie Cellulaire et Végétale, UMR 5168 CNRS – CEA – INRA - Université Joseph Fourier Grenoble 1, iRTSV, CEA Grenoble, 17 rue des Martyrs, 38054 Grenoble, France.*

1. Introduction

A small chemical compound, which specifically activates or inhibits a given biological system can become a lead candidate in a drug discovery perspective or a molecular tool for biological research (Stockwell, 2000; Mayer, 2003; Inglese *et al.*, 2007; Maréchal, 2008). The identification and production of such compounds is thus a major issue for biological or therapeutic research. Strategies for small molecule discovery rely largely on high throughput screening (HTS) of chemical libraries, which has traditionally been the purview of industry for the past twenty years, and has become recently available in academic institutions (Stein, 2003; Fox *et al.*, 2006). Such high throughput approaches use robotic handling of miniaturized biological assays and allow the screening of a large number of compounds to select those (called "hits") that produce the wanted and reproducible effect on a given biological target (*e.g.* an enzyme or a whole cell). The size of available

compounds collections to screen is rather large: for instance, the ChemNavigator's database, which proposes commercially available screening compounds from international chemical suppliers currently tracks over 46.7 million chemical samples. Among them over 24.9 millions are claimed to be unique. However, such an amount is still small, relatively to the size of the chemical space: the number of synthesizable compounds is estimated to range from 10^{18} to 10^{200} compounds (Parker and Schreyer, 2004). Yet, the screening of a very large chemical library can be financially expensive, time consuming and the amount of biological material needed might be simply non realistic. Biologists must often lower their ambition and select a limited number of molecules to assay. The design of relevant chemical libraries, often called "core libraries" since they are supposed to accurately reflect the diversity of a very large collection (Dubois *et al.*, 2008), is thus a central issue for screening. The automatic clustering of chemical compounds can generate homogenous subsets based on a similarity measure, and allow a rationale definition for a core library (Willet, 1998).

It has been demonstrated that structurally similar compounds are very likely to have a similar biological activity (Martin *et al.*, 2002). Thus, not only the clustering should allow the identification of compound subsets, from which one representative molecule can be selected to be screened, but this method can also help to increase the diversity of in-house data sets by selecting additional compounds in other identified clusters of molecules. Moreover, it has been shown that it could be useful to select molecules which come from each cluster containing a "hit" compound; those "related" compounds should be tested through further validation screening stages (Engels *et al.*, 2000).
Clustering can also be used in a virtual screening approach, to select relevant virtual libraries (prior to a docking process for instance) or to select the more promising and diverse molecules (after a docking process) to be tested *in vitro*.

Chemical compounds clustering, like any object clustering, implies four steps (Downs and Barnard, 2002):

(1) Identification of relevant descriptors for these objects;
(2) Selection and computation of a similarity (or a distance) measure;
(3) Use of a clustering algorithm to gather objects according to this distance or similarity;
(4) Analysis and qualification of the results.

Molecules are structurally complex objects; it is therefore obvious that the clustering quality relies strongly on the capacity of the distance measure to embrace both the structural likeness and dissimilarities. In this chapter, we focus on the efficiency of an adaptation, for small molecular objects, of a similarity index initially proposed in Inductive Logic Programming (Wieczorek *et al.*, 2006). We compare this novel method with some other structural distances that are customary in chemistry or which have been recently proposed for molecular graph comparisons.

2. Methods for the computation of structural distances between molecules

The computation of structural distances between molecules (represented by graphs) directly or indirectly implies the search for isomorphic partial graphs. Generally, methods use a linearization (SMILE language from Weininger, 1988) or a structure propositionalization of the compound. Thus, a molecule is represented by a vector of descriptors, each one

corresponding to a molecular fragment (Leach and Gillet, 2003). Recently, kernel functions, comparable to distances between graphs (Gartner *et al.*, 2003), have been proposed in the Support Vector Machines (SVM) context. They present good performances in supervised machine learning to predict molecular bio-activity (Mahé *et al.*, 2005) or to solve bioinformatics problems (Menchetti *et al.*, 2005). In these approaches, molecular representation is global: a set of paths (*i.e.* molecular fragments specifically chosen or drawn by chance) is built explicitly or implicitly. It is also possible to value structural distances by dynamically building molecular fragments according to the matching between two molecules. This approach is proposed by Fröhlich *et al.* (2005), in a so-called "global matching" kernel to predict bio-activity. We focus here on a similar strategy with a similarity index I_{pi}, based on the comparison of labeled trees or substructures that allows the classification of molecules in an unsupervised learning machine approach. A short description of the kernel functions used in this comparative study and a deeper explanation of the principle of the I_{pi} similarity index are exposed in the following part.

2.1 Kernel functions

Kernels functions are at the basis of machine learning methods using Support Vector Machines (SVM) approaches. These functions allow working on initial data as if they were in a high-dimensional space without having to transform them explicitly; moreover, kernel methods handle non-linear complex tasks using linear methods in this new space. The main advantage is that the data may be more easily separable in that high-dimensional space (the so-called "feature space") than in the original space. This trick is at the basis of kernel machines. SVM and kernel functions were detailed by Shawe-Taylor and Cristianini (2004) and Schölkopf and Smola (2002).

2.1.1 Tanimoto kernel

This kernel (Ralaivola *et al.*, 2005) is the transformation of the classical Tanimoto distance (Willet, 1998; Flower, 1998) into a kernel function. Molecules are seen as vectors where each dimension is associated with a given molecular fragment and the coordinates indicate if this fragment exists or not in the molecule. To build these vectors, it is necessary to give the maximum length of the considered molecular fragment. This can be defined by allowing the selection of paths from length *1* to a maximum u or allowing the selection of paths of an exact length l.

2.1.2 Weighted Decomposition kernel (2D-WD kernel)

In this kernel (Menchetti *et al.*, 2005), molecules are represented by the set of all possible subgraphs which can be built for a given maximum depth. The kernel function between two molecules x and y weights the exact kernel between each pair of atoms (x_i, y_j) according to the structural information. This one corresponds to the subgraphs that contain all the paths of depth d, built from each atom x_i and y_j.

2.1.3 Optimal Assignment Kernel (OA Kernel)

The Optimal Assignment Kernel (Fröhlich *et al.*, 2005) is based on a dynamical and local graph exploration. Unlike the Tanimoto Kernel, the relational structure of molecules is clearly conserved in this representation. The kernel computation is divided into two steps that are conceptually close to the ones proposed by Bisson (1995). The first step evaluates a

distance between each atom pair (a_i, b_j) from two molecules A and B, thanks to the kernel function named K_{nei} that takes into account the width w of each atom neighboring. The second step matches atoms a_i (from A) with atoms b_j (from B), in order to maximize the K_{nei} (a_i, b_j) sum, which amounts to doing a maximum weight matching in a bipartite graph.

2.2 Structural similarity I_{pi} Index

This similarity index is an adaptation of the index proposed by Bisson (1995) and Wieczorek *et al.* (2006) to chemical structures.

2.2.1 General principles

Each molecule M is described as a non-oriented graph defined by a pair (A, L) where:

- A corresponds to the atoms $\{a_1, \ldots, a_n\}$ of molecule M;
- L corresponds to the covalent bonds between these atoms $\{l_1, \ldots, l_p\}$.

Many similarity coefficients usually used in computational chemistry are based on the size (in number of atoms) of the Maximum Common Substructure (MCS) between two molecules M and M′ (Bunke and Shearer, 1998). As for example, the similarity S_1 which is defined as the relative size of the MCS compared to the size of the biggest molecule :

$$S_1(M, M') = \frac{|MCS(M, M')|}{\max(|M|, |M'|)} \quad (1)$$

The similarity coefficient S proposed here is also based on the MCS but is defined as the mean of two dual asymmetric values $INC(M, M')$ and $INC(M', M)$:

$$S(M, M') = \frac{1}{2}\left(Ipi(M, M') + Ipi(M', M)\right) \quad (2)$$

where the value $INC(M, M')$ is the relative similarity of M towards M', *i.e.* the degree of inclusion of M into M'. It is defined as the relative size of the MCS between M and M' compared to the size of molecule M′:

$$Ipi(M, M') = \frac{|MCS(M, M')|}{|M'|} \quad (3)$$

This trick allows comparing molecules having big differences in size. For instance, if M is smaller than M' and M is nearly included in M', from the point of view of M, the molecule M' is very similar since it contains the same information and we have $INC(M, M') \approx 1$, whereas a classical symmetric similarity would reflect this difference in size. The use of the mean of both inclusion values ($INC(M, M')$ and $INC(M', M)$) leads to a more realistic similarity measure that allows breaking the size bias and focusing deeper on the existence of common substructures as shown on Figure 1.

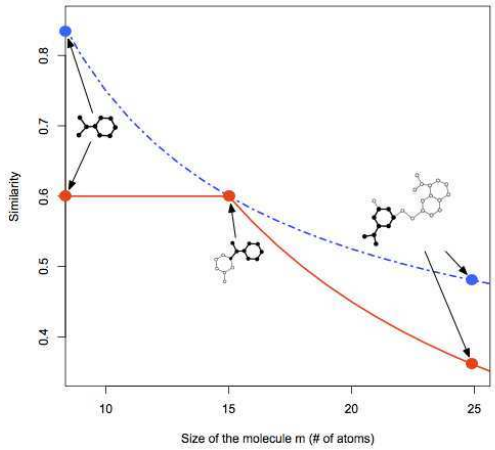


Fig. 1. **Difference between a classical similarity measure and an asymmetric-based one.** In this graph, S_1 (plain line) is a classical similarity measure and S (dashed line) a non-symmetric-based one. This graph shows the behaviour of both S (equation 2) and S_1 (equation 1) between a molecule M of size 15 and another molecule m of increasing size: S= 0.5*(9/15+9/|m|) and $S_1(M, m)$ = 9/max(15, $|m|$). The size of the MCS is 9. One can observe that S_1 remains constant for all molecules which size is lower than $|M|$. In other words, the similarity S_1 confounds those molecules that seem all similar, with respect to their size; which is not the case for S because there is no plateau.

The degree of inclusion *INC (M, M')* is computed in two steps, a local one and a global one. The first local step aims at computing a local similarity between each pair of atoms (a_i, a'_p) belonging respectively to molecules M and M'. The values are stored in a matrix called *SUB*; $SUB[a_i, a'_p]$ is in [0,1]. The same processing is followed for M' towards M and the results are stored in a matrix named *SUB'*. The key idea is to consider that two atoms a_i and a'_p are most similar if they share common physicochemical properties but also if the neighboring atoms to which they are connected by covalent bonds are themselves similar to each other. This recursive definition allows expressing the problem in the form of a non-linear equations system; the resolution of this system consists in the search of a fixed point.

The goal of the second step is to compute a global inclusion between both molecules M and M', *i.e.* the value of *INC (M, M')*. Having local similarities values for each atom pair (a_i, a'_p) and according to the structural connectivity of M, we search the matching that maximizes the global inclusion which can be approximate by the biggest common tree or substructure between the two molecules. Once both *INC (M, M')* and *INC (M', M)* are computed, the total similarity between both molecules is the mean of both values. This mean is then used by the clustering algorithm.

2.2.2 Computation of local similarity (between atoms)

The aim is to compute the value of each element of the matrix *SUB* which corresponds to the local similarity between one atom a_i of $M = (A, L)$ and one atom a'_p of $M' = (A', L')$ (see example in figure 2). It quantifies the inclusion degree of the environment of the atom a_i in the environment of the atom a'_j. The following functions are defined:

S_a: $A \times A' \rightarrow [0,1]$, the similarity between two atoms according to their respective physicochemical properties;

S_l : $L \times L' \rightarrow [0,1]$, the similarity between two covalent bonds according to their respective physicochemical properties;

S: $A \times L \times A' \times L' \rightarrow [0,1]$, the similarity between two pairs (atom, bound);

NbLink (a_i), the number of covalent bonds of a given atom a_i;

Link-of : $A \rightarrow L$, a function returning, for a given atom a_i, the list $\{l_1, \ldots, l_m\}$ of the covalent bonds of a_i;

Ngbr : $A \times L \rightarrow A$, a function which gives for a given atom a_i and a given bound l_m, the neighbouring atom a_j which is connected to a_i by l_m.

The inclusion degree is stored in $SUB[a_i, a'_p]$. Its computation comes down to build a system of non-linear equations, where $SUB[a_i, a'_p]$ is one of the variables to compute. The resolution of this system is obtained by using the Jacobi's iterative method. After each iteration, we have the following equations:

$$S(a_i, l_m, a'_p, l'_t) = 1/2 \times (SUB[Nghbr(a_i, l_m), Nghbr(a'_p, l'_t)] + Sl(l_m, l'_t)) \quad (4)$$

$$SUB[a_i, a'_p] = 1/2\,(S_a(a_i, a'_p) + MaxMatchScore / NbLink(a_i)) \quad (5)$$

MaxMatchScore is computed according to the following processing. Let us define the function *Find_Max_Mapping* (FMM). For two given atoms a_i and a'_p, FMM searches the optimal mapping between the neighbours of a_i and a'_p and so between the corresponding covalent bonds. *FMM* returns the score of this optimal mapping that is *MaxMatchScore*.

This function is a classic problem of matching. Indeed, let L (resp. L') be the list of the covalent bonds in which appears the atom a_i (resp. a'_p). Bonds, which are elements of L and L', can be considered as a bipartite graph elements; thanks to S, the similarity of each quadruplet (a_i, l_m, a'_p, l'_t), is known. Thus, finding the best matching between these elements boils down to maximize the sum of the S values, *i.e.* to solve a maximum weight matching in a bipartite graph. *MaxMatchScore*, the corresponding matching score, is equal to the sum of the S values.

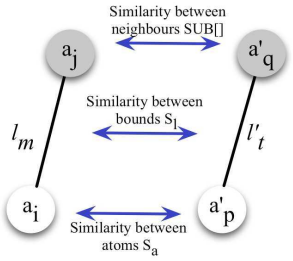
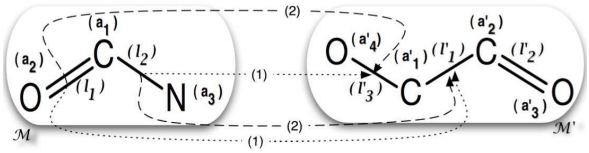


Fig. 2. Summary of the information used to compare two given atoms in a molecule.

To find this optimal matching, we use the Hungarian algorithm also called the Kuhn-Munkres algorithm (Kuhn, 1955; Munkres, 1957) whose complexity is $O(n^3)$. This is not a problem since lists L and L' are rather small: their size corresponds to the considered atom valence (for instance, 4 for the carbon atom). Thus, the local similarity between two atoms is defined recursively and the originality of this approach lies in the fact that S is computed according to $SUB[Nghbr(a_i, l_m), Nghbr(a'_p, l'_t)]$. Lastly, the local similarity between two

atoms corresponds to the average of their physicochemical similarity S_a and the normalized average of the similarity of their neighbouring (see Figure 2).

In equation (2), *MaxMatchScore* is normalized by $NbLink(a_i)$. This gives its asymmetric nature to $SUB[a_i , a'_p]$, whereas a division by $Max\ (NbLink\ (a_i)\ ,\ NbLink\ (a'_p))$ would have kept a symmetric nature for this similarity measure. From a practical point of view, we use the Jacobi's iterative method in a synchronous way, *i.e.* we use two instances of the matrix *SUB*, one for the iteration i and one for the iteration $i+1$: all the values of the matrix SUB_{i+1} are computed using the terms of the matrix SUB_i, so the values for all atoms are simultaneously changed, when SUB_{i+1} is copied in SUB_i. The number of iterations characterizes the depth of the information propagation, *i.e.* the neighbouring size taken into account to compare two atoms. Figure 3 shows an example of this propagation.



For k=0: $\forall a_i, \forall a'_j, SUB_k[a_i, a'_j] = 1$;

Let's compute for instance, for k=1, $SUB_k[a_1, a'_1]$:

$$SUB_1[a_1, a'_1] = 1/2\ S_a(C, C) + MaxMatchScore(a_1) / NbOfLink(a_1)$$
$$= 1/2\ (1 + MaxMatchScore(a_1) / 2)$$

We must find the optimal local matching between the bounds of a_1 and a'_1 and their associated neighbours. Here, there are 2 possibilities represented in dotted line (1) or in dashes (2) .

- If we pair bounds and neighbours according to (1), the score is equal to:

$S(a_1,l_1,a'_1,l'_1) + S(a_1,l_2,a'_1,l'_3) = 1/2(SUB_0[a_2, a'_2] + S_l(l_1, l'_1)) + 1/2(SUB_0[a_3, a'_4] + S_l(l_2, l'_3)) = 1/2(1+0) + 1/2(1+1)=3/2$

- If we pair them according to (2), the score is equal to:

$S(a_1,l_1,a'_1,l'_3) + S(a_1,l_2,a'_1,l'_1) = 3/2$

So, $MaxMatchScore(a_1) = 3/2$ and $SUB_1[a_1, a'_1] = 0.875$

Here are given all the values of the matrix SUB_k for 6 iterations:

	k=0	k=1	k=2	k=3	k=4	k=5
$SUB_k[a_1, a'_1]$	1	0.875	0.78125	0.77344	0.76758	0.76709
$SUB_k[a_1, a'_2]$	1	1	0.9375	0.9375	0.93359	0.93359
$SUB_k[a_1, a'_3]$	1	0.25	0.1875	0.16406	0.16211	0.16138
$SUB_k[a_1, a'_4]$	1	0.25	0.1875	0.1875	0.18555	0.18555
$SUB_k[a_2, a'_1]$	1	0.25	0.25	0.23437	0.23437	0.23340
$SUB_k[a_2, a'_2]$	1	0.5	0.3125	0.29687	0.29102	0.29053
$SUB_k[a_2, a'_3]$	1	1	1	0.98437	0.98437	0.98340
$SUB_k[a_2, a'_4]$	1	0.75	0.71875	0.69531	0.69336	0.69189
$SUB_k[a_3, a'_1]$	1	0.5	0.5	0.48437	0.48437	0.48340
$SUB_k[a_3, a'_2]$	1	0.5	0.46875	0.44531	0.44336	0.44189
$SUB_k[a_3, a'_3]$	1	0.25	0.25	0.23437	0.23437	0.23340
$SUB_k[a_3, a'_4]$	1	0.5	0.46875	0.44531	0.44336	0.44189

After the 5^{th} iteration, the values of the matrix don't change any longer and differences of similarities due to atoms nature but also due to atoms neighbourhood appear : here, for instance, a great similarity appears between a_1 and a'_2, and between a_2 and a'_3.

Fig. 3. Computation and evolution of the matrix *SUB*. Example with the molecule *M* being formamide ($CONH_3$) and the molecule *M'* being glycol-aldehyde ($C_2H_4O_2$).

It can be proved that the system always has a solution and that this is found after few iterations (from *3* to *8* according to the complexity of the molecules). The information propagation decreases as $1/(n+1)^2$ where n is the distance between neighbours.

When using the Hungarian algorithm, the overall complexity for the computation of the *SUB* is in $O(K.V^2.D^3)$ where:

- D is the mean number of neighbours for each atom in the molecules;
- K is the number of iterations of the iterative procedure;
- V is the mean number of atoms in the molecules.

This complexity may be reduced in $O(K.V^2.D^2)$ if D is low, by pre-computing all the possible matching between them.

In this work, to have homogeneity with the other compared methods (see Experimental materials and tests methodology), the molecular representation is kept minimal:
S_a depends only on the type of atoms (C, O, N etc.):

$S_a(a_i, a'_j) = 1$ if atom types are equal, $S_a(a_i, a'_j) = 0$ otherwise.

S_l depends only on the type of bonds (simple, double, triple, aromatic):

$S_l(l_i, l'_j) = 1$ if bound types are equal, $S_l(l_i, l'_j) = 0$ otherwise.

Note that S_a and S_l are generic functions. It would then be very easy to integrate more complex properties for atoms and bonds, such as charge index, pharmacophore points, etc. Modifying S_a and S_l would be sufficient without any change in the global algorithm.

2.2.3 Global similarity (between molecules) computation

In order to compute the global similarity between two molecules *M* and *M'*, *INC(M,M')* , we search for the best matching between atoms from *M* and *M'*. This matching relies on the local similarities stored in matrix *SUB* and it maximizes the global inclusion. This can be achieved by using the Hungarian algorithm, as in *OA Kernel* (Fröelich *et al.*, 2005), considering that we have a bipartite graph, built with atoms of molecules *M* and *M'*. Since local similarities can correspond to different matching, they do not guarantee that the maximum common structure found by the algorithm would be a connected one. However, this is not a real problem in chemistry since applications such as lead discovery or synthesis design might put a premium on unconnected structural solutions.

The matching is therefore searched according to the following heuristic. The best score of local similarities between atoms that we can find in the matrix *SUB*, is taken as a seed. The matching is then propagated according to the structure connectivity of *M* and according to the *SUB* values. In the example shown in Figure 4, *M* and *M'* are two molecules, atoms « *1* » and « *a* » are taken as seeds. Atoms « *2* » and « *5* » (neighbours of « *1* ») are processed to be matched with « *b* » and « *e* » (neighbours of « *a* »). Their matching can be processed according to a greedy algorithm (pairing according to a decreasing ranking of similarity values) or according to a Kuhn algorithm. In the following experiments, we have chosen a greedy algorithm for this matching. When « *2* » and « *5* » are paired, « *3* » (neighbour of « *2* ») and « *f* » and « *g* » (neighbours of « *e* ») are processed and so on. This matching stops when there are no more atoms to match in *M* or as soon as an atom of *M* cannot be matched with an atom of *M'*. Let us note down that :

1. atom « *3* » (neighbour of « *2* ») is processed before « *4* » (neighbour of « *5* ») because « *2* » has been stored before « *5* » in our implementation structure : there is no special criteria for this choice;
2. atom « *4* » (neighbour of « *5* ») has been, for instance, matched with « *c* ». It is not processed again as a neighbour of « *3* », but our algorithm implementation keeps in mind that there is also a connection between « *3* » and « *4* ». So, if there was also a

connection between « *c* » and « *f* », in molecule *M'*, it would be able to find that the best match is a cyclic substructure.

Since *B* is only a possible start point, the procedure is repeated (ten trials[1]), each time taking a different pair of atoms not already matched in a previous trial.

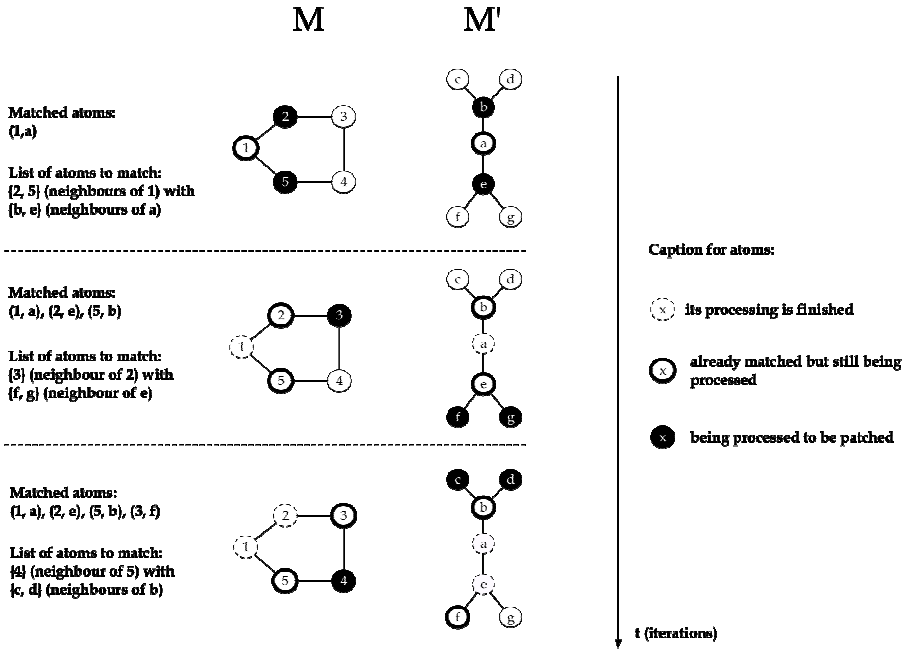


Fig. 4. Example for general principles of the incremental matching.

The complexity for the search of the best match between the two compounds with the use of *SUB* is in $O(V^3)$ where *V* is the mean number of atoms in the molecules. As for the local similarity computation, this complexity may be reduced in $O(1)$ when pre-computing all the possible matching for low values of V, which is frequent in chemistry. The best matching between atoms from two molecules *M* and *M'* is then used to compute the global inclusion of *M* in *M'*, *INC(M,M')*. This is obtained by the sum of all *SUB* matrix elements relating to matched atoms of *M* and *M'*, divided by the number of atoms in *M*. This division by the size of *M* brings additional asymmetry to this measure. Thus, the overall complexity of this approach is in $O(K.V^2.D^2)$.

3. Experimental material and tests methodology

In this paragraph we present the material and methodology for the comparison of the capacity of each similarity measure to return, by classification of simple molecular 2D structures, chemical families defined by experts.

---

[1] This number was established experimentally: indeed, for all data sets, the best matching appeared in the 10th first iterations in more than 99% of cases.

3.1 Chemical libraries

3.1.1 Data sets

Four public chemical datasets published by Sutherland *et al*. (2003) have been used. These authors gathered compounds which had been tested against four different biological targets; these datasets present the advantage of being already divided into well defined and precisely described chemical families. The *Cox2* library contains a set of 467 molecules tested as inhibitors of the cyclooxygenase-2 and divided into 13 families; the *Bzr* library is a set of 405 ligands for the benzodiazepine receptor, divided into 14 families; the *Dhfr* library contains a set of 756 inhibitors of the dihydrofolate reductase, divided into 18 families, with 32 compounds belonging to singleton families. In this study, these singletons were discarded and only 724 compounds divided into 17 families were considered. The *Er* library is a set of 1,009 estrogen receptor ligands: 393 (extracted from the literature) are gathered into 3 structural families and 616 form a miscellaneous group: for our study, we considered only the 3 families extracted from the literature.

Molecular bi-dimensional (2D) structures were provided in SDF files. For standardization purposes, all the molecules were normalized according to a set of normalization rules that was built in accordance with usual chemical usage. This task was achieved using the ChemAxon software Application Programmatic Interface (www.chemaxon.com) and standardization rules were formally defined as chemical reactions in an XML configuration file read by the ChemAxon Standardizer object. The setup file is available upon request.

In order to compare methods within the same description context, independently of the studied distance, we reduced the set of descriptors associated with each molecular graph to the minimum set, which existed in all the compared methods, or which was easily integrated in each distance implementation. In this minimal representation, a molecule is an attributed undirected graph $x=(V, E)$. Each vertex v in V represents an atom and is labeled by the atom type (C, O, N, etc.). Each edge (v,w) in E represents a chemical bound; it is characterized by a type that can be *single*, *double*, *triple* or *aromatic*.

3.3 Kernel functions implementation and clustering algorithm

Mahé *et al,* (2005) provided the implementation of the *Tanimoto Kernel*. For each of the two other kernels (*2D-WD Kernel* and *OA Kernel*), their respective author's implementation was used. Once distance matrices had been computed, molecules were categorized into families using the well-known ascendant hierarchical classification (Johnson, 1967); the chosen implementation was *hcluster* from R software (www.r-project.org/) and the Ward index was the interclass aggregation distance.

3.4 Parameters setting

In the following experimentations, for each of the selected methods, the parameters' values were optimized for best classification capacities.

For the *Tanimoto Kernel,* all values (from 5 to 20) were tested for the parameters u (all paths of length from 1 until u) and l (all paths of exact length l). For each database, the parameter

(either *u* or *l*) was unchanged and the associated values that gave the best results (details can be given upon request). For the *2D-WD*, *OA Kernels* and I_{pi}, the main parameter is the *width* of the neighbourhood, which is taken into account to evaluate the similarity between two atoms. In I_{pi}, this parameter corresponds to the number of iterations used to compute the matrix of similarities between all pairs of atoms. A value of 5 was sufficient in I_{pi} to see the convergence of the *SUB* matrix. Thus, this value was selected for the corresponding parameters in *OA Kernel* and *2D-WD Kernel*.

3.5 Classification evaluation

As emphasized by Candellier *et al.* (2006), classification evaluation is difficult without any validation criteria. It is not the case here since we know, for each dataset, the number and precise content (in terms of molecules) of the families (or classes) that the system must retrieve. To evaluate the difference between the original classification and the learnt one, a confusion matrix was used (Kohavi and Provost, 1998), called *M(O,B)*. Its lines (O_i with i varying from 1 to p) represent the original classes and its columns (B_j with j varying from 1 to q) represent the classes built by the classification system. Each matrix element $n_{i,j}$ represents the number of molecules which are present in both classes O_i and B_j. The built classification is optimal when there is only one value $n_{i,j}$ different from 0 for each line and each column. So a simple way to qualify the classification is to measure the average entropies associated to lines and columns. Two indices based on conditional entropies were considered, the *Confusion Index (CI)* that quantifies the number of merged classes and the S*egmentation Index (SI)* that quantifies the number of split classes.
Given

$$\overline{C_i} = \sum_{j=1}^{q} n_{i,j} \quad \overline{L_j} = \sum_{i=1}^{p} n_{i,j} \quad N = \sum_{i=1}^{p}\sum_{j=1}^{q} n_{i,j} \tag{6}$$

, we define:

$$CI = \sum_{i=1}^{p} \frac{\overline{C_i}}{N} \sum_{j=1}^{q} -\frac{n_{i,j}}{\overline{C_i}} \times \log_2 \frac{n_{i,j}}{\overline{C_i}} \tag{7}$$

$$SI = \sum_{j=1}^{q} \frac{\overline{L_j}}{N} \sum_{i=1}^{p} -\frac{n_{i,j}}{\overline{L_j}} \times \log_2 \frac{n_{i,j}}{\overline{L_j}}$$

The most important index is *CI* since it indicates if the initial classification has been well retrieved by the algorithm. The lower its value is, the better is the matching between both classifications.

4. Comparative analysis of similarity measures

Figure 5 shows the *CI* index evolution for the four chemical libraries. Excepted for *Cox2* library, ranking of the four methods is always the same: I_{pi}, *Tanimoto Kernel*, *OA Kernel* and *2D-WD Kernel*. Considering the results for each base, it can be observed that molecules from the *Cox2* library belong to close scaffolds (*i.e.* core graph structures). However, families are easily recognizable because of few discriminating atoms or chemical functions, whose positions vary in aromatic cycles. I_{pi} index recognizes nearly all the expected families because it is able to detect each atom local environment using its local similarity measurement. I_{pi} categories that do not comply with expertised chemical families

correspond to very small families which have been merged by the classification system. *Tanimoto Kernel*, also, produces a good score whereas *2D-WD* Kernel and *OA Kernel* results were less efficient in returning chemical families.

Molecules from the *Dhfr* library have several similar substructures which are differently connected together from one family to another. Results are the same as for the *Cox2* database but with a greater dispersion between the methods. I_{pi} index shows its capacity to globally recognize molecular structures.

Structures of molecules from the *Bzr* library are very diverse and this variability is sometimes rather great within the original families. In this case, all the methods failed to accurately recover the original classification given by chemists.

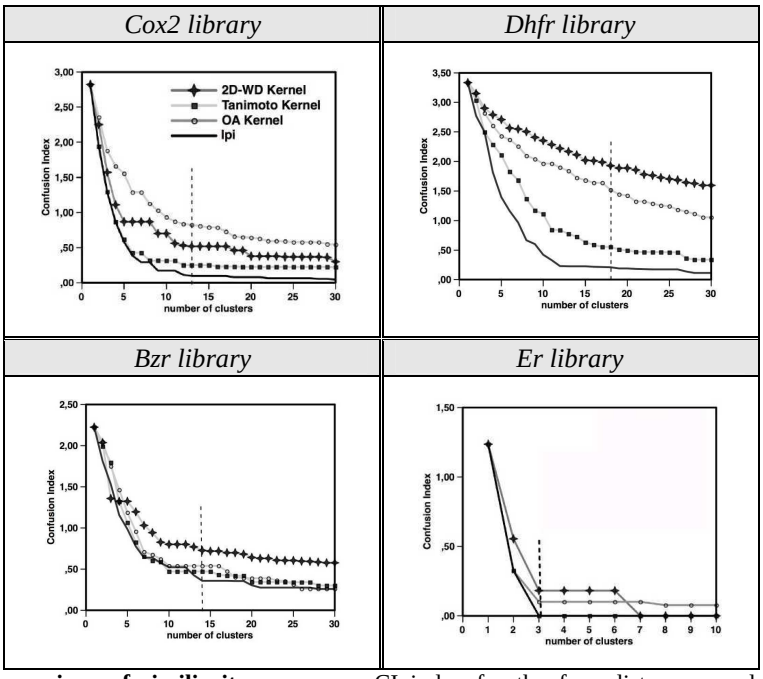


Fig. 5. Comparison of similiarity measures. CI index for the four distances used with HAC clustering method on the four datasets: *Cox2*, *Dhfr*, *Bzr* and *Er*. The vertical line marks the original number of chemical families and the point where a ranking between the four methods can be done.

Finally, the *Er* library contains only three families. Each one is characterized by a specific scaffold which should be easily recognized. Indeed, I_{pi} and *Tanimoto Kernel* found again the three expected families (their CI curves are superimposed). On the other hand, again, *2D-WD Kernel* and *OA Kernel* performance was lower.

The tested approaches correspond to two different strategies. *TanimotoK* and *2D-WD Kernel* represent the molecules by means of paths in the graphs contrary to *OA Kernel* and I_{pi} that take into account the whole structure of the graphs. It is therefore interesting to understand

why for each library, the I_{pi} and *Tanimoto Kernel* methods performed better. In the case of *OA Kernel* and I_{pi}, which use close algorithms, we searched the main modifications that would explain the differences. To this purpose, I_{pi} algorithm was changed to erase its two major differences with *OA Kernel*:

- The asymmetrical calculation of the similarity: in equation (5), *NbLink* (a_i) was replaced by *Max* (*NbLink* (a_i), *NbLink* (a'_p)), and in the global similarity calculation (see 2.2.4), the size of *M* was replaced by the maximum value between the sizes of *M* and *M'*. The measure became thus purely symmetrical;
- The incremental matching taking account of the molecular connectivity: we replaced our matching algorithm (see 2.2.3) by a Kuhn algorithm looking for a maximum weight matching in a bipartite graph built with atoms of *M* and *M'*.

The influence of these modifications and their combination was studied on the classification of the four chemical databases. The results do not depend on the database; taking into account the molecules connectivity brings the main improvement while asymmetry slightly improves the overall classification. Figure 6 details these results on *Dhfr* database.

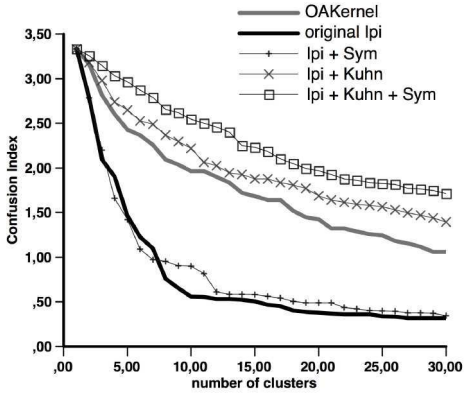


Fig. 6. *CI* indexes for 5 distances used with HAC clustering method on the *Dhfr* dataset, *OA Kernel*, I_{pi} and 3 distances obtained by modification of I_{pi}: Symmetric calculation of similarity index (curve "I_{pi} + Sym"), or Kuhn algorithm replacing incremental matching algorithm ("I_{pi} + Kuhn"), or both modifications ("I_{pi} + Kuhn + Sym"). Replacing Kuhn's algorithm by an incremental one taking account the connectivity gives a strong improvement but whatever the matching algorithm is, the asymmetric nature of the index slightly improves the efficiency of the classification.

In the case of *Tanimoto Kernel* and *2D-WD Kernel*, it is the path selection which is important to get an accurate representation of molecules.

This study has been focused on the comparison of several graph kernels applied to chemical compounds in a supervised classification task; that is to say the families of molecules are known. In this case, one remaining question is the choice of the optimal value for the Confusion Index in order to define the clustering level. In particular, il may be difficult to conclude in areas where CI is quite constant (e.g. for "Original Ipi" and "Ipi + Sym"

methods between 15 and 30 clusters). However, in a real case, the classification is not known and the Confusion Index could not be computed.

5. Conclusion

The evaluation of molecular libraries and, more specifically, molecule categorization into families is important for biologists and chemists before and after *in silico* or *in vitro* molecular screening. In this chapter, we have described some of the important similarity measures currently used, and a new similarity index we recently developed for chemical molecule comparison. It is very difficult to choose a similarity measure for a chemoinformatic purpose, besides empirical considerations like the availability in the software suite used in the laboratory, the familiar utilization of a given similarity measure in ateam or the demonstrated efficiency in an experimental context leading to the selection of the used index for subsequent analyses. Here, we compared methods on four well-known chemical datasets, in order to evaluate the capacity of the algorithms to retrieve the families defined by chemists.

The relatively disappointing results obtained with *2D-WD Kernel* and *OAKernel* seem to indicate that "distances" having good performances in a supervised learning context (activity prediction) are not always adapted to classical clustering algorithms. By comparing I_{pi} and *OA Kernel*, we observed that taking into account the molecular connectivity was important but also that the distance measures based on asymmetrical comparisons could lead to better results than the ones based on a plain symmetric definition.

To complete this study, it should be interesting to integrate SVM clustering (among others, Ben-Hur *et al.*, 2001; Finley and Joachims, 2005) and SVM classification (Rupp *et al.*, 2007) instead of the Hierarchical Ascendant Classification or by testing the MG Kernel extension of Mahé *et al.* (2004). In the case of the asymmetrical measure we introduced here and compared to classical indexes, it is important to further investigate the three steps of I_{pi} to understand clearly which one(s) is (are) the most important for the I_{pi} efficiency in comparison with the *Tanimoto Kernel* or the *2D-WD Kernel*. Indeed, in this overview, one should be surprised by the good results of the *Tanimoto Kernel*, which is clearly less complex to compute than the I_{pi} index. However, this latter presents two advantages (independently of being ranked first for all the tested libraries): on the one hand, it takes into account all the knowledge about the molecules without needing a linearization, so it is not necessary to manually choose the size of the structural keys to use and there is no loss of structural information; on the other hand, by modifying S_a and S_l functions, it is possible to integrate in the measure all the physical and chemical information that the expert would judge useful.

6. Acknowledgments

The authors thank P. Mahé for fruitful discussions and the use of the *Tanimoto Kernel* implementation, part of his CPP software. They would like to acknowledge the ChemAxon Company (http://www.chemaxon.com) for allowing academics to freely use their software, especially here the *Standardizer* to normalize molecules. They are also grateful to J. Bleuse and K. Delbos-Corfield for the reading of this manuscript and their suggestions or their corrections which improved it. *Funding*: this work was partly supported by grants from the

French Ministry of Research "ACI Impbio" program (Accamba project) and from the French "Rhône-Alpes Futur" Foundation.
Conflict of Interest: none declared.

7. References

Ben-Hur, A. *et al*. (2001) Support Vector Clustering. *Journal of Machine Learning Research,* **2**, pp. 125–137

Bisson, G. (1995) Why and how to define a similarity measure for object-based representation systems. *Proceedings of 2nd international conference on building and sharing very large-scale knowledge bases (KBKS)*. IOS press. Enschede (NL), pp. 236–246

Candellier, L. *et al*. (2006) Cascade Evaluation of Clustering Algorithms. *Proceedings of ECML*, Berlin, pp. 574–581.

Downs, G.M. and Barnard,J.M. (2002) Clustering Methods and Their Uses in Computational Chemistry. *Re*V. *Comput. Chem.* **18**, pp. 1–40.

Dubois, J. *et al*. (2008) Collections of Compounds - How to Deal with them ?, *Current Computer-Aided Drug Design*, **4** (3), pp. 156–168

Engels, M.F.M. *et al*. (2000) CerBeruS: A System Supporting the Sequential Screening Process. *J. Chem. Inf. Comput. Sci.* , **40**, pp. 241–245.

Finley, T. and Joachims,T. (2005) Supervised Clustering with Support Vector Machines, in *Proceeding of ICML. Bonn*, Germany. pp. 217 – 224.

Fox, S. *et al*. (2006) High-throughput screening: update on practices and success. *J Biomol Screen*. **11**, pp. 864–869.

Flower, D.R. (1998) On the properties of bit string-based measures of chemical similarity. *J. Chem. Inf. Comput. Sci*, **38**, pp. 378–386.

Fröhlich, H. *et al*. (2005) A Optimal Assignment Kernels for Attributed Molecular Graphs, *In Proc. of Int. Conf. on Machine Learning (ICML)*, pp. 225–232.

Gartner, T. *et al*. (2003) On graph kernels: Hardness results an efficient alternatives. *Proc. of the 16th Annual Conference on Computational Learning Theory and the 7th Annual Workshop on Kernel Machines. Heidelberg* : Springer-Verlag.

Inglese, J. *et al*. (2007) High-throughput screening assays for the identification of chemical probes. *Nat Chem Biol*. **3**, pp. 466–479.

Johnson, S. (1967) Hierarchical Clustering Schemes, *Psychometrika*, **2**, pp. 241–254.

Kohavi, R. and Provost,F. (1998) Glossary of terms. *Machine Learning*, **30**, pp. 271–274.

Kuhn, H.W. (1955) The Hungarian Method for the assignment problem, Naval Research Logistics Quarterly , **2**, pp. 83–97.

Leach, A.R. and Gillet,V.J. (2003) An Introduction to Chemoinformatics, Kluwer Academic Publishers.

Munkres, J. (1957) Algorithms for the Assignment and Transportation Problems, *Journal of the Society of Industrial and Applied Mathematics*, **5** (1), pp. 32–38.

Mahe, P. *et al*. (2004) Extensions of Marginalized Graph Kernels, *In Proc. of the 21st International Conference on Machine Learning (ICML)*. Banff, Alberta.

Mahe, P. *et al*. (2005) Graph kernels for molecular structure-activity relationship with support vector machines. *J. Chem. Inf. Model.* **45**(4), pp. 939–951.

Maréchal E. (2008). Chemogenomics: a discipline at the crossroad of high throughput technologies, biomarker research, combinatorial chemistry, genomics,

cheminformatics, bioinformatics and artificial intelligence. *Comb Chem High Throughput Screen*. **11**, pp. 583–586.

Martin, Y.C. *et al*. (2002) Do Structurally Similar Molecules Have Similar Biological Activity ? *J. Med. Chem.* **45**, pp. 4350–4358.

Menchetti, S. *et al*. (2005) Weighted decomposition kernels, *Proceedings of the International Conference on Machine Learning (ICML)*. Bonn, Germany, pp 585 – 592.

Mayer, T.U. (2003) Chemical genetics: tailoring tools for cell biology. *Trends Cell Biol*. **13**, pp. 270–277.

Parker, C.N. and Schreyer, S.K. (2004) Application of Chemoinformatics to High-Throughput Screening In: *Methods in Molecular Biology*, *vol 275: Chemoinformatics: Concepts, Methods and Tools for Drug Discovery.* Edited by J.Bajorath. Humana Press Inc. Towota, NJ, pp 85–110.

Ralaivola, L. *et al*. (2005) Graph kernels for chemical informatics. *Neural Networks*, **18**, pp. 1093–1110.

Rupp, M. ; Proschak, E.; Schneider, G. (2007) Kernel Approach to Molecular Similarity Based on Iterative Graph Similarity, *J. Chem. Inf. Model.*, 47, pp. 2280-2286

Schölkopf, B. and Smola, A.J. (2002) Learning with Kernels, The MIT press, Cambridge.

Shawe-Taylor, J. and Cristianini, N. (2004) Kernels Methods for Pattern Analysis, Cambridge University Press.

Stein, R.L. (2003) High-throughput screening in academia: the Harvard experience. *J Biomol Screen*. **8**, pp. 615–619.

Stockwell, B.R. (2000) Frontiers in chemical genetics. *Trends Biotechnol*. **18**, pp. 449–455.

Sutherland, J.J. *et al*. (2003) Spline-Fitting with a Genetic Algorithm: A Method for Developing Classification Structure-Activity Relationships, *J. Chem. Inf. Comput. Sci.* **43**, pp. 1906–1915.

Weininger, D. (1988) SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules, *J. Chem. Inf. Comput. Sci.* **28**, pp. 31–36.

Wieczorek, S. *et al*. (2006) Guiding the Search in the NO Region of the Phase Transition Problem with a Partial Subsumption Test. *In proceeding of ECML 2006. LNCS 4212/2006*. 18–22 september, Berlin. pp 817–824.

Willet, P. (1998) Chemical similarity searching. *J. Chem. Inf. Comput. Sci*. **38**, pp. 983–996.