HAL
open science

Incorporating stereo information within the graph kernel framework

Pierre-Anthony Grenier, Luc Brun, Didier Villemin

To cite this version:

Pierre-Anthony Grenier, Luc Brun, Didier Villemin. Incorporating stereo information within the graph kernel framework. [Research Report] GREYC CNRS UMR 6072, Universite de Caen. 2013. hal-00809066v2

HAL Id: hal-00809066

https://hal.science/hal-00809066v2

Submitted on 7 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Incorporating stereo information within the graph kernel framework

Pierre-Anthony Grenier†, Luc Brun†, and Didier Villemin‡

†GREYC UMR CNRS 6072, ‡LCMT UMR CNRS 6507,
Caen, France
{pierre-anthony.grenier,didier.villemin}@ensicaen.fr,
luc.brun@greyc.ensicaen.fr

Abstract. Molecules being often described using a graph representation, graph kernels provide an interesting framework which allows to combine machine learning and graph theory in order to predict molecule's properties. However, some of these properties are induced both by the covalent bound relationships between atoms and by constraints on the relative positioning of these atoms. Graph kernels based solely on the graph representation of a molecule do not encode the relative positioning of atoms and are consequently unable to predict accurately molecule's properties connected with this relative positioning. In this report, ordered structured object are introduced in order to incorporate spatial constraints within the graph kernel framework. The incorporation of this new features within the graph kernel framework allows to predict accurately stereo information hence overcoming the previous limitation.

Keywords: Graph kernel, Chemoinformatics, Chirality.

1 Introduction

The purpose of Chemoinformatic is to predict properties of molecules, in order to facilitate drug design. Chemoinformatics is based on the similarity principle: two structurally similar molecules should have similar properties.

One common method to predict chemical properties consist to design a vector of descriptors from a molecule and use statistical machine learning algorithms to predict molecule's properties. Such methods [4, 3], can use structural information, physical properties or biological activities in order to compute vectors of descriptors. However, such an approach requires to either select a random set of pre defined descriptors (before a variable selection step) or to use an heuristic definition of appropriate descriptors by a chemical expert. In both cases, the transformation of the graph into a finite vector of features induces a loss of information.

Another approach consist to encode a molecule by a graph, and use it to predict properties.

Definition 1. Molecular graph

A molecular graph is a labeled graph $G = (V, E, \mu, \nu)$ representing a molecule. The unlabeled graph (V, E) encodes the structure of the molecule, each node $v \in V$ encoding an atom and each edge $e = (v, w) \in E$ a bond between two atoms v and w. μ associates to each vertex $v \in V$ a label $\mu(v)$ encoding the nature of the atom and ν associates to each edge e a type of bond $\nu(e)$ (single, double, triple or aromatic).

Several methods based on graph theory use this representation to predict properties. One approach consists to search subgraphs with a large difference of frequencies between a set of positive and a set of negative examples [5]. Another approach consists to encode each class of molecules by a graph prototype and to measure the structural similarity between each prototype and an input molecule [6]. However, these methods can not be easily combined with machine learning algorithms. This is not the case of graph kernel methods, which can be coupled to machine learning algorithm provided that the kernel is definite positive. Let $\mathcal{G}$ be the set of graph. A definite positive kernel is a symmetric function $k : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ such that:

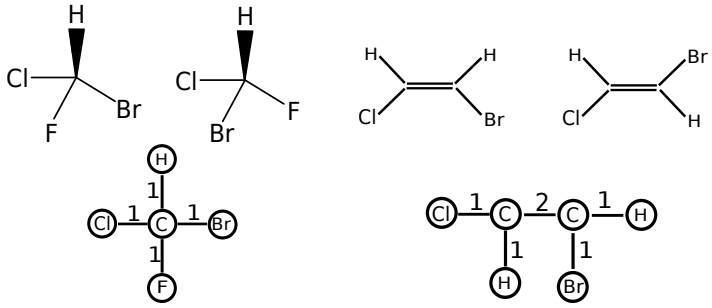
$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(G_i, G_j) \geq 0 \text{ where } n > 0,\ G_1, \ldots, G_n \in \mathcal{G},\ c_1, \ldots, c_n \in \mathbb{R}$$

Such a definite positive kernel corresponds to a scalar product between two vectors $\psi(G)$ and $\psi(G')$ in an Hilbert space.

A large family of graph kernel methods, associates a bag of patterns to each graph, and define the kernel value from a measure of similarity between those bags [7–9, 1]. In [7] a graph kernel is defined as a measure of similarity between set of walks extracted from each graph. But those walks are linear features and thus have limited expressiveness. An infinite set of tree patterns is used in [8] to define kernels. However, the similarity between two graphs is based on an implicit enumeration of their common tree patterns which does not allow to readily analyze the influence of a pattern on the prediction. Finally [9] and [1] are based on an explicit enumeration of patterns. In [9], a predefined set of unlabeled subgraphs, called graphlets, is enumerated for each graph and in [1] all subtrees of a labeled graphs up to size 6, called treelets are enumerated. One advantage of [1] is that, unlike [9], the label of the graph are taken into account by the graph similarity measure.

However, some molecules may have a same molecular formula, a same molecular graph but a different relative positioning of their atoms. Such molecules are said to be stereoisomers. Different stereoisomers may be associated to different properties. However, usual graph kernels based on the molecular graph representation are not able to capture any dissimilarity between these molecules. From a more local point of view, an atom or two connected atoms are called stereocenters if a permutation of the positions of two atoms belonging to the union of their neighborhoods produces a different stereoisomer.

In order to get an intuition of stereoisomerism, let us consider an acyclic molecular graph rooted on an atom of carbon with four neighbors, each neighbor



(a) Asymmetric carbon

(b) Carbons connected by a double bond

Fig. 1. Two types of stereocenters.

being associated to a different subtree. Such an atom, called an asymmetric carbon, is a stereocenter and has two different spatial configurations of its neighbors encoded by a same molecular graph (Figure 1(a)). Using molecule represented in Figure 1(a), one configuration corresponds to the case where the three atoms (Cl,F,Br) considered from the atom H are encountered in this order when turning counter-clockwise around the central carbon atom. The alternative stereoisomer corresponds to the case where this sequence of atoms is encountered clockwise when considered from the same position. This example corresponds to a particular form of stereoisomerism, called chirality, where the molecule has no center nor plane of symmetry. In this case, molecules are said to be chiral.

Two carbons, connected by a double bond, can also define stereoisomers (Figure 1(b)). Indeed, on the left side of Figure 1(b) both hydrogen atoms are located on the same side of the double bond while they are located on opposite sides on the stereoisomer represented on the right. In this case both carbon atoms of the double bond correspond to a stereocenter. This example correspond to another stereoisomerism form, called geometric isomerism, where stereoisomers have at least one center or one plane of symmetry.

To distinguish those configurations, we introduce the two following subsets of the set of vertices V of a molecular graph:

Definition 2. Potential Asymmetric Carbons

Let us denote V_{PAC} the subset of V containing all vertices encoding atoms of carbon with four neighbors:

$$V_{PAC} = \{v \in V \mid \mu(v) = \text{'}C\text{'} \text{ and } |V(v)| = 4\}$$

Since being an atom with four neighbors is a necessary condition to define an asymmetric carbon, the set V_{PAC} contains all vertices which may encode such atoms.

Definition 3. Set of double-bonds connecting carbon atoms

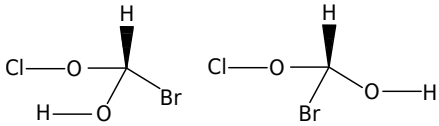


Fig. 2. Asymmetric carbons with identical neighborhood.

The subset of V containing all atoms of carbon which share a double bond with another carbon is noted V_{DB}:

$$V_{DB} = \left\{ v \in V \mid \exists e(v,w) \in E, \nu(e) = 2, \begin{pmatrix} |V(v)| = |V(w)| = 3 \\ \text{and} \\ \mu(v) \quad = \mu(w) \quad = \text{'}C\text{'} \end{pmatrix} \right\}$$

An atom of carbon with two double bounds must have a degree equal to two. Hence, each vertex v belonging to V_{DB} is incident to a single double bound and we denote $n_{=}(v)$ the other carbon connected by this double bond. Note that $n_{=}(v) \in V_{DB}$.

Brown et al. described in [2] a method which includes information related to the spatial configuration of atoms within the tree-pattern kernel [8]. However, this method only considers the direct neighbors of a stereocenter while, as shown by Figure 2, the difference between two subtrees of a stereocenter may not be located on the root of the subtree. In this last case [2] considers as identical two different stereocenters and is thus unable to recover their different properties.

In this paper we propose a method to incorporate the spatial configuration of atoms within a graph kernel based on a subtree enumeration [1]. This method remains valid when the spatial configuration is not encoded in the direct neighborhood of a stereocenter. In Section 2, we define a graph encoding of stereoisomers and we introduce stereo vertices as vertices encoding stereocenters. Next, in Section 3, we restrict our attention to acyclic molecules. Such a restriction allows us to efficiently characterize a stereo vertex by a rooted tree. In Section 4, we define the smallest tree characterizing a stereo vertex and use this information to design a graph kernel between molecules. Finally, we demonstrate the validity of our kernel through experiments in Section 5.

2 Encoding of stereoisomers

2.1 Ordered Structured Object

Definition 4. Structured Objects
A structured object S is an object to which we can associate an unique labeled graph $G(S) = (V, E, \mu, \nu)$.

A structured object can be for example a graph (associated to itself) or a rooted tree (associated to an acyclic graph).

An usual method in chemistry to encode stereoisometry consists to encode a relative order on the neighborhood of each vertex. In order to encode such an information, we introduce the notion of order on structured object.

Definition 5. Ordered Structured Objects

An ordered structured object $S = (\hat{S}, ord)$ is a structured object $\hat{S}$, associated to a graph $G(S) = (V, E, \mu, \nu)$, together with a function ord which maps each vertex v belonging to a subset V_{ord} of V onto an ordered list of a subset of its neighborhood $V(v)$:

$$ord \begin{cases} V_{ord} \rightarrow & V^* \\ v \quad \rightarrow & v_1 \dots v_n \end{cases}$$

where $\{v_1, \dots, v_n\} \subset V(v)$ denotes a subset of the neighborhood of v.

We denote $|ord(v)| = n$ the length of the ordered list for any $v \in V_{ord}$. We have thus $0 < |ord(v)| \leq |V(v)|$.

Note that, the notation V_{ord} which denotes the subset of V for which function ord is defined will be used in the remaining part of this document.

Definition 6. Set of Ordered Structured Objects

A set of ordered structured objects $\mathscr{S}$ is a set $\mathscr{S} = \{S = (\hat{S}, ord)\}$ from which we can define a set of isomorphism $\text{Isom}(\hat{S}, \hat{S}') \subset \text{Isom}(G(S), G(S'))$ between any two structured objects $\hat{S}$ and $\hat{S}'$. This set of isomorphism must respects the following properties:

1. $\forall S_1, S_2 \in \mathscr{S}^2$, $f \in \text{Isom}(\hat{S}_1, \hat{S}_2) \Leftrightarrow f^{-1} \in \text{Isom}(\hat{S}_2, \hat{S}_1)$
2. $\forall S_1, S_2, S_3 \in \mathscr{S}^3$, $f \in \text{Isom}(\hat{S}_1, \hat{S}_2), g \in \text{Isom}(\hat{S}_2, \hat{S}_3) \Rightarrow g \circ f \in \text{Isom}(\hat{S}_1, \hat{S}_3)$
3. $\forall S \in \mathscr{S}$, $\text{Isom}(\hat{S}, \hat{S})$ is a group.
4. $\forall (S, S') \in \mathscr{S}^2, \forall f \in \text{Isom}(\hat{S}, \hat{S}')$

$$\begin{cases} f(V_{ord}) = V'_{ord} \\ \forall v \in V_{ord}, |ord(v)| = |ord'(f(v))| \end{cases}$$

The first three conditions impose that our restricted set of isomorphim relationships satisfies the usual properties of isomorphisms: the inverse and the composition of two isomorphisms is still an isomorphism (conditions 1 and 2) and considering a set of automorphisms, the identity belongs to our valid set of isomorphisms (condition 3). The last condition imposes that the set of vertices on which the function ord is defined remains stable by an isomorphims. It further imposes that an isomorphism does not modify the number of vertices on which the order relationship is defined for each vertex.

Definition 7. Isomorphism between ordered structured objects

Let us consider a set of ordered structured objects $\mathscr{S}$. Two ordered structured objects $S = (\hat{S}, ord)$ and $S' = (\hat{S}', ord')$ are said to be isomorphic $S \underset{o}{\simeq} S'$ iff there is an isomorphism between the structured objects $\hat{S}$ and $\hat{S}'$ which is coherent with the order on the subsets of the neighborhoods:

$$S \underset{o}{\simeq} S' \Leftrightarrow \exists f \in \mathrm{Isom}(\hat{S}, \hat{S}') \ s.t.$$
$$\forall v \in V_{ord} \text{ with } ord(v) = v_1 \dots v_n,\ ord'(f(v)) = f(v_1) \dots f(v_n)$$

In this case, f is called an ordered isomorphism between S and S', and we denote $\mathrm{IsomOrd}(S, S') \subset \mathrm{Isom}(\hat{S}, \hat{S}')$ the set of ordered isomorphism between S and S'.

Proposition 1. Ordered structured object isomorphism induces an equivalence relationship.

Proof. Let us consider a set of ordered structured objects $\mathscr{S}$. We have thus to show that the structured object isomorphism relationship is reflexive, symmetric and transitive:

1. The isomorphism between structured object is reflexive.
   Let $S = (\hat{S}, ord) \in \mathscr{S}$ an ordered structured object associated to a graph $G(S) = (V, E, \mu, \nu)$, and let f denotes the identity function on V ($f(v) = v, \forall v \in V$).
   Then $f \in \mathrm{Isom}(\hat{S}, \hat{S})$ (Definition 6, condition 3) and:
   $$\forall v \in V_{ord} \subset V,\ f(v_1) \dots f(v_n) = ord(f(v)) = ord(v) = v_1 \dots v_n$$
   where $\{v_1, \dots, v_n\} \subset V(v)$ denotes a subset of the neighborhood of v.
   Therefore, $S \underset{o}{\simeq} S$.
2. The isomorphism between ordered structured object is symmetric.
   Let $S_a = (\hat{S}_a, ord_a) \in \mathscr{S}$ and $S_b = (\hat{S}_b, ord_b) \in \mathscr{S}$ two ordered structured object, respectively associated to $G(S_a) = (V_a, E_a, \mu_a, \nu_a)$ and $G(S_b) = (V_b, E_b, \mu_b, \nu_b)$, such that $S_a \underset{o}{\simeq} S_b$ and let us further denote by f the ordered isomorphism between S_a and S_b.
   By definition f is also an isomorphism between $\hat{S}_a$ and $\hat{S}_b$, therefore it exists (Definition 6, condition 1) an isomorphism f^{-1} between $\hat{S}_b$ and $\hat{S}_a$. Let us consider a vertex v_b in $V_{ord_b} \subset V_b$ and $v_a = f^{-1}(v_b) \in V_{ord_a}$ (Definition 6, condition 4). Since $S_a \underset{o}{\simeq} S_b$, we have:
   $$\begin{cases} ord_a(v_a) = v_{a1} \dots v_{an} \text{ and} \\ ord_b(v_b) = ord_b(f(v_a)) = v_{b1} \dots v_{bn} \text{ with } v_{bi} = f(v_{ai}), \forall i \in \{1, \dots, n\} \end{cases}$$
   Hence:
   $$\begin{cases} ord_b(v_b) = v_{b1} \dots v_{bn} \text{ and} \\ ord_a(v_a) = ord_a(f^{-1}(v_b)) = v_{a1} \dots v_{an} = f^{-1}(v_{b1}) \dots f^{-1}(v_{bn}) \end{cases}$$
   Thus $S_b \underset{o}{\simeq} S_a$ and f^{-1} is an ordered isomorphism between S_b and S_a.
3. The isomorphism between ordered structured object is transitive.
   Let $S_a = (\hat{S}_a, ord_a) \in \mathscr{S}$, $S_b = (\hat{S}_b, ord_b) \in \mathscr{S}$ and $S_c = (\hat{S}_c, ord_c) \in \mathscr{S}$ three ordered structured object, respectively associated to $G(S_a) = (V_a, E_a, \mu_a, \nu_a)$,

$G(S_b) = (V_b, E_b, \mu_b, \nu_b)$ and $G(S_c) = (V_c, E_c, \mu_c, \nu_c)$, such that $S_a \underset{o}{\simeq} S_b$ and $S_b \underset{o}{\simeq} S_c$. We denote by f the ordered isomorphism between S_a and S_b, and by g the ordered isomorphism between S_b and S_c.
As the isomorphism between structured objects is transitive we have $g \circ f \in \text{Isom}(\hat{S}_a, \hat{S}_c)$ (Definition 6, condition 2).
Let us consider a vertex v_a in $V_{ord_a} \subset V_a$ with $v_b = f(v_a) \in V_{ord_b} \subset V_b$ and $v_c = g(v_b) = g \circ f(v_a) \in V_{ord_c} \subset V_c$. We have since $S_a \underset{o}{\simeq} S_b$ and $S_b \underset{o}{\simeq} S_c$:

$$\begin{cases} ord_a(v_a) & = v_{a1} \dots v_{an} \text{ and} \\ ord_b(f(v_a)) & = ord_b(v_b) = f(v_{a1}) \dots f(v_{an}) \underset{not.}{=} v_{b1} \dots v_{bn}, \\ ord_c(g(v_b)) & = g(v_{b1}) \dots g(v_{bn}) \end{cases}$$

Therefore:

$$ord_c(g(v_b)) = ord_c(g \circ f(v_a)) = g \circ f(v_{a1}) \dots g \circ f(v_{an})$$

Thus $S_a \underset{o}{\simeq} S_c$ and $g \circ f$ is an ordered isomorphism between S_a and S_c.

In conclusion, the isomorphism between ordered structured objects is reflexive, symmetric and transitive. It is therefore, an equivalence relationship. □

2.2 Re-ordering function

A spatial configuration of atoms may be encoded by several equivalent orders. We thus introduce the notion of re-ordering function, which associates to each vertex of an ordered structured object a permutation on a subset of its neighborhood.

Definition 8. Re-ordering functions

Let us consider a set of ordered structured objects $\mathscr{S}$. A re-ordering function σ_S on an ordered structured object $S = (\hat{S}, ord)$, associated to a graph $G(S) = (V, E, \mu, \nu)$, associates to each vertex $v \in V_{ord}$ a permutation φ_v on $\{1, \dots, |ord(v)|\}$.

$$\sigma_S \begin{cases} V_{ord} \rightarrow & \mathcal{P} \\ \quad v \quad \rightarrow \varphi_v \in \Pi_{|ord(v)|} \end{cases}$$

where Π_n is the group of permutations of n elements and $\mathcal{P}$ is the union of Π_n for all $n \in \mathbb{N}$.

Application of a re-ordering function on an ordered structured object provides a new ordered structured object defined as follows:

Definition 9. Re-ordered structured objects

Let us consider a set of ordered structured objects $\mathscr{S}$. Let $S = (\hat{S}, ord)$ denotes an ordered structured object, $\sigma_S(S) = (\hat{S}, ord_{\sigma_S})$ is defined as the ordered structured object obtained after applying the re-ordering function σ_S on the order of the object:

$$\forall v \in V_{ord} \ s.t. \ \begin{pmatrix} ord(v) = v_1, \dots, v_n \\ \text{and} \\ \sigma_S(v) = \varphi_v, \end{pmatrix} \ ord_{\sigma_S}(v) = v_{\varphi_v(1)}, \dots, v_{\varphi_v(n)}$$

Note that $\hat{S} = \widehat{\sigma_S(S)}$. In other words, a re-ordering of an ordered structured object does not change the associated structured object. Re-ordering operations being defined as functions, these functions may be combined using composition operations:

Definition 10. Composition of re-ordering functions

Let us consider a set of ordered structured objects $\mathscr{S}$. Let σ_S and σ'_S denote two re-ordering functions on an ordered structured object $S = (\hat{S}, ord)$. The composition of σ_S and σ'_S is a re-ordering function denoted by $\sigma_S \circ \sigma'_S$ and defined as follows:

$$\sigma_S \circ \sigma'_S \begin{pmatrix} V_{ord} & \to \mathcal{P} \\ v & \to \sigma_S(v) \circ \sigma'_S(v) \in \Pi_{|ord(v)|} \end{pmatrix}$$

where Π_n is the group of permutations of n elements and $\mathcal{P}$ is the union of Π_n for all $n \in \mathbb{N}$.

The identity for the composition is the re-ordering function Id_S such that $\forall v \in V_{ord}$, $Id_S(v) = Id_{|ord(v)|}$ where Id_n is the identity permutation on Π_n.

2.3 Structured object having equivalent order

Re-ordering functions previously defined may apply any re-ordering on a structured object hence removing the notion of order on these objects. In order to obtain a useful notion of re-ordering, we have to define more precisely which properties should satisfies a valid family of re-ordering functions.

Definition 11. Valid re-orderings

Let us consider a set of ordered structured objects $\mathscr{S}$. For each $S = (\hat{S}, ord) \in \mathscr{S}$, let us denote by Σ_S a set of re-ordering functions on S. A valid family of re-ordering functions is a set $\Sigma = \{\Sigma_S,\ S \in \mathscr{S}\}$ which satisfies the two following properties :

- For any $S \in \mathscr{S}, \Sigma_S$ is a group for the composition.
- For any two ordered structured objects $S = (\hat{S}, ord)$ and $S' = (\hat{S}', ord')$ whose associated un-order structured objects are isomorphic by a function f, any re-ordering function $\sigma \in \Sigma_S$ is equal, up to the isomorphism f^{-1}, to a re-ordering function of $\Sigma_{S'}$.

$$\left.\begin{matrix} \forall f \in \mathrm{Isom}(\hat{S}, \hat{S}'), \\ \forall \sigma \in \Sigma_S \end{matrix}\right) \sigma \circ f^{-1} \in \Sigma_{S'}.$$

The first constraint of Definition 11 states that the set of re-ordering functions of an ordered structured objects may be combined freely using composition operations. The second constraint, involves that two ordered structured objects with isomorphic un-ordered structured objects should have, up to the isomorphism function, equivalent set of re-ordering functions. Note that this last constraint is equivalent to the following equation:

$$\left.\begin{matrix} \forall f \in \mathrm{Isom}(\hat{S}, \hat{S}'), \\ \forall \sigma \in \Sigma_S \end{matrix}\right) \exists \sigma' \in \Sigma_{S'} \,|\, \sigma' \circ f = \sigma$$

Proposition 2. Let us consider a set of ordered structured objects $\mathscr{S}$, and a valid family of re-ordering functions Σ. The group of re-ordering functions $\Sigma_{\sigma(S)}$ of any re-ordered structured object $\sigma(S)$ is equal to the group Σ_S of re-ordering functions of S:

$$\left.\begin{array}{l}\forall S \in \mathscr{S}, \\ \forall \sigma \in \Sigma_S,\end{array}\right) \Sigma_{\sigma(S)} = \Sigma_S$$

Proof. Let us consider $\sigma \in \Sigma_S$. Since S and $\sigma(S)$ only differ by the order defined on each vertices, the identity function Id is a valid isomorphism between the unorder structured objects associated to $\sigma(S)$ and S. Then, using Definition 11, for any $\sigma' \in \Sigma_{\sigma(S)}$, it exists a re-ordering function $\sigma'' \in \Sigma_S$ such that:

$$\forall v \in V_{ord},\ \sigma'(v) = \sigma''(Id(v)) = \sigma''(v)$$

We have thus $\sigma' = \sigma''$ and thus $\Sigma_{\sigma(S)} \subset \Sigma_S$. The reverse inclusion is shown in the same way. □

Definition 12. Equivalent orders

Let us consider a set of ordered structured objects $\mathscr{S}$ and two of its ordered structured objects $S_a = (\hat{S}_a, ord_a) \in \mathscr{S}$ and $S_b = (\hat{S}_b, ord_b) \in \mathscr{S}$. These structured objects are said to be equivalent $S_a \underset{\Sigma}{\simeq} S_b$ according to a valid family of re-ordering functions Σ if:

$$\exists \sigma \in \Sigma_{S_a} \in \Sigma,\ \sigma(S_a) \underset{o}{\simeq} S_b \tag{1}$$

In other word, we consider that two ordered structured objects are equivalent if, up to a valid re-ordering σ we can establish an ordered structured object isomorphism f between them. In that case the ordered isomorphism f is called an equivalent ordered isomorphism through σ between S_a and S_b and we denote $\mathrm{IsomEqOrd}_\sigma(S_a, S_b)$ the set of equivalent ordered isomorphism through σ between S_a and S_b. We further denote by $\mathrm{IsomEqOrd}(S_a, S_b)$ the union of all $\mathrm{IsomEqOrd}_\sigma(S_a, S_b)$ for all $\sigma \in \Sigma_{S_a}$.

$$\mathrm{IsomEqOrd}(S_a, S_b) = \bigcup_{\sigma \in \Sigma_{S_a}} \mathrm{IsomEqOrd}_\sigma(S_a, S_b)$$

We will now prove that the equivalence order relationship is, as suggested by its name, an equivalence relationship.

Proposition 3. Let $\mathscr{S}$ be a set of ordered structured objects and Σ denotes a valid family of re-ordering functions. The equivalent order relationship based on this family is reflexive.

$$\forall S \in \mathscr{S},\ S \underset{\Sigma}{\simeq} S$$

Proof. Let Σ denotes a valid family of re-ordering functions and $S = (\hat{S}, ord)$ an ordered structured object.

By Definition 11, Σ_S is a group and therefore $Id_S \in \Sigma_S$. We have by definition of Id_S (Definition 10):

$$\forall v \in V_{ord}, ord_{Id_S}(v) = ord(v).$$

We have by Definition 7, $Id_S(S) \underset{o}{\simeq} S$ with $Id_S \in \Sigma_S$.

Thus $S \underset{\Sigma}{\simeq} S$. □

Lemma 1. *Let $\mathscr{S}$ be a set of ordered structured objects and Σ a valid family of re-ordering functions. Let us consider two ordered structured objects $S_a = (\hat{S}_a, ord_a) \in \mathscr{S}$ and $S_b = (\hat{S}_b, ord_b) \in \mathscr{S}$ such that $S_a \underset{o}{\simeq} S_b$. Let $\sigma_a \in \Sigma_{S_a}$ and $\sigma_b \in \Sigma_{S_b}$ two re-ordering functions such that:*

$$\forall v \in V_{ord_a},\ \sigma_a(v) = \sigma_b(f(v)),$$

where f is an ordered isomorphism between S_a and S_b.

Then we have $\sigma_a(S_a) \underset{o}{\simeq} \sigma_b(S_b)$.

Proof. Let us consider $f \in \mathrm{IsomOrd}(S_a, S_b)$, and a vertex $v \in V_{ord_a}$ with $u = f(v)$. We have by Definition 7:

$$\forall v \in V_{ord_a} \begin{cases} ord_a(v) = v_1 \dots v_n \text{ and} \\ ord_b(u) = u_1 \dots u_n, \text{ with } u_i = f(v_i), \forall i \in \{1, \dots, n\} \end{cases}$$

Let us further denote by φ_v the permutation defined on vertex v both by σ_a and σ_b: $\varphi_v = \sigma_a(v) = \sigma_b(f(v))$.

Given the re-ordered structured objects $\sigma_a(S_a) = (\hat{S}_a, ord_{\sigma_a})$ and $\sigma_b(S_b) = (\hat{S}_b, ord_{\sigma_b})$ we have by Definition 9:

$$\begin{cases} ord_{\sigma_a}(v) = v_{\varphi_v(1)} \dots v_{\varphi_v(n)} \text{ and} \\ ord_{\sigma_b}(u) = u_{\varphi_v(1)} \dots u_{\varphi_v(n)}. \end{cases}$$

Since sequences $v_1 \dots v_n$ and $u_1 \dots u_n$ satisfy $u_i = f(v_i)$ and since a same permutation φ_v is applied on both sequences we have:

$$\forall i \in \{1, \dots, n\}\ u_{\varphi_v(i)} = f(v_{\varphi_v(i)})$$

The isomorphism f maps thus the order encoded by $\sigma_a(S_a)$ around each vertex of S_a onto the order defined by $\sigma_b(S_b)$ on the corresponding vertex of S_b.

Moreover, since f is an isomorphism between ordered structured objects, it also corresponds to an isomorphism between un-ordered structured objects and we have by Definition 7, $\sigma_a(S_a) \underset{o}{\simeq} \sigma_b(S_b)$.

□

Proposition 4. Let $\mathscr{S}$ be a set of ordered structured objects and Σ a valid family of re-ordering functions. The equivalent order relationships based on this family is symmetric:

$$\forall (S_a, S_b) \in \mathscr{S}^2,\ S_a \underset{\Sigma}{\simeq} S_b \Leftrightarrow S_b \underset{\Sigma}{\simeq} S_a.$$

Proof. Let us consider a set of ordered structured objects $\mathscr{S}$, Σ a valid family of re-ordering functions and two ordered structured objects $S_a = (\hat{S}_a, ord_a) \in \mathscr{S}$ and $S_b = (\hat{S}_b, ord_b) \in \mathscr{S}$ such that $S_a \underset{\Sigma}{\simeq} S_b$ We have by Definition 12:

$$\exists \sigma \in \Sigma_{S_a} s.\ t.\ \sigma(S_a) \underset{o}{\simeq} S_b.$$

As Σ_{S_a} is a group (Definition 11), it exists a re-ordering function $\sigma^{-1} \in \Sigma_{S_a}$ such that $\sigma^{-1}(\sigma(S_a)) = S_a$.

Let us denote by f the ordered isomorphism between $\sigma(S_a)$ and S_b. By Definition 11, since $\sigma(S_a) \underset{o}{\simeq} S_b$, the re-ordering function $\sigma^{-1} \in \Sigma_{S_a}$ should be equivalent to some re-ordering function $(\sigma^{-1})'$ in Σ_{S_b}. In other words:

$$\exists (\sigma^{-1})' \in \Sigma_{S_b} \text{ such that } \forall v \in V_{ord_a}, \sigma^{-1}(v) = (\sigma^{-1})'(f(v)).$$

We have thus by Lemma 1: $\sigma^{-1}(\sigma(S_a)) \underset{o}{\simeq} (\sigma^{-1})'(S_b)$. Therefore $S_a \underset{o}{\simeq} (\sigma^{-1})'(S_b)$, and by symmetry of the ordered isomorphism $(\sigma^{-1})'(S_b) \underset{o}{\simeq} S_a$. So by Definition 12, $S_b \underset{\Sigma}{\simeq} S_a$. Both cases being symmetric, the reverse implication is proved in the same way.

□

Proposition 5. Let $\mathscr{S}$ a set of ordered structured objects and Σ a valid family of re-ordering functions. The equivalent order relationship based on this family is transitive:

$$\forall (S_a, S_b, S_c) \in \mathscr{S}^3, \begin{pmatrix} S_a \underset{\Sigma}{\simeq} S_b \text{ and} \\ S_b \underset{\Sigma}{\simeq} S_c \end{pmatrix} \Rightarrow S_a \underset{\Sigma}{\simeq} S_c$$

Proof. Let us consider a set of ordered structured objects $\mathscr{S}$, Σ a valid family of re-ordering functions and three ordered structured objects $S_a = (\hat{S}_a, ord_a) \in \mathscr{S}$, $S_b = (\hat{S}_b, ord_b) \in \mathscr{S}$ and $S_c = (\hat{S}_c, ord_c) \in \mathscr{S}$ such that $S_a \underset{\Sigma}{\simeq} S_b$ and $S_b \underset{\Sigma}{\simeq} S_c$.

Using Definition 12, it exists two re-ordering functions $\sigma_a \in \Sigma_{S_a}$ and $\sigma'_b \in \Sigma_{S_b}$ such that $\sigma_a(S_a) \underset{o}{\simeq} S_b$ and $\sigma'_b(S_b) \underset{o}{\simeq} S_c$.

Let us denote the isomorphism between un-ordered structured objects $\hat{S}_b$ and $\widehat{\sigma_a(S_a)}$ by f_{ba}. Since S_b and $\sigma(S_a)$ are isomorph and since $\sigma'_b \in \Sigma_{S_b}$, it must exists (by Definition 11) a re-ordering function $\sigma'_a \in \Sigma_{\sigma_a(S_a)}$ such that:

$$\forall v \in V_{ord_b},\ \sigma'_b(v) = \sigma'_a(f_{ba}(v)).$$

Then by Lemma 1, we have $\sigma'_b(S_b) \underset{o}{\simeq} \sigma'_a(\sigma_a(S_a))$. Since the ordered isomorphism is an equivalence relationship (Proposition 1) and $\sigma'_b(S_b) \underset{o}{\simeq} S_c$, we have $\sigma'_a(\sigma_a(S_a)) \underset{o}{\simeq} S_c$.

Since Σ_{S_a} is a group (Definition 11) and since $\sigma_a \in \Sigma_{S_a}$ and $\sigma'_a \in \Sigma_{\sigma_a(S_a)} = \Sigma_{S_a}$ (Proposition 2) we have: $\sigma'_a \circ \sigma_a \in \Sigma_{S_a}$.

So by Definition 12, $S_a \underset{\Sigma}{\simeq} S_c$.

□

Theorem 1. *Let $\mathscr{S}$ be a set of ordered structured objects and Σ a valid family of re-ordering functions. The equivalent order relationship based on this family is an equivalence relationship.*

Proof. The equivalent order relationship is reflexive (Proposition 3), symmetric (Proposition 4) and transitive (Proposition 5). □

2.4 Ordered graphs

We now apply the definition of ordered structured objects to labeled graphs, in order to define ordered graphs.

Definition 13. Set of Ordered Graphs
An ordered graph $S = (G = (V, E, \mu, \nu), ord)$ is an ordered structured object with $G(S) = G$ and a function ord which maps each vertex of $V_{ord} \subset V$ to an ordered list of its neighbors:

$$ord \begin{cases} V_{ord} \rightarrow & V^* \\ v & \rightarrow v_1 \dots v_n \end{cases}$$

where $V(v) = \{v_1, \dots, v_n\}$ denotes the neighborhood of v.

The set of ordered graph is denoted $\mathcal{OG}$. The set of isomorphism defined between two un-ordered graphs (Definition 6) is the usual set of isomorphism between labeled graphs.

As the set of ordered graphs is a set of structured objects, we can define, for ordered graphs, re-ordering functions (Definition 8). The definition of orders and re-ordering functions depends on the application at end. Nevertheless, if re-orderings fulfill the definition of a valid family of re-ordering functions (Definition 11) we can define the equivalence relationship between ordered graphs defined by Definition 12 (Theorem 1).

Let us now define a stereo vertex, which encodes a stereocenter when ordered graphs represents molecules.

Definition 14. Stereo vertices
Let Σ be a valid family of re-ordering functions on $\mathcal{OG}$. Let $G = (V, E, \mu, \nu, ord)$ be an ordered graph. A vertex $v \in V_{ord} \subset V$ of degree n is called a stereo vertex iff:

$$\forall (i, j) \in \{1, \dots, n\}^2 \text{ with } i \neq j, \nexists f \in \text{IsomEqOrd}(G, \tau_{i,j}(G)) \text{ with } f(v) = v.$$

where $\tau_{i,j}$ is a re-ordering function equals to the identity on all vertices except v for which it permutes the vertices of index i and j in $ord(v)$.

In other words, a vertex is a stereo vertex if any permutation of its neighbors produces an ordered graph with a non-equivalent order.

2.5 Ordered graphs and re-ordering functions encoding of a molecule

We now restrict our attention on molecular graphs (Definition 1) and let us define from them molecular ordered graphs.

The molecular ordered graph of a molecule is defined by first defining its molecular graph (Definition 1) $G = (V, E, \mu, \nu)$ which encodes relationships between atoms together with the type of atom and bond respectively associated to each vertex and each edge.

Definition 15. Molecular ordered graph

A molecular ordered graph is a couple $S = (G, ord)$ where G corresponds to a molecular graph. The function ord is defined on a set V_{ord} defined as:

$$V_{ord} = V_{PAC} \cup V_{DB}$$

where V_{PAC} and V_{DB} denote respectively the set of potential asymmetric carbons (Definition 2) and the set of carbons of degree 3 connected by a double bound (Definition 3).

The function ord is defined as follows for each vertex $v \in V_{ord}$:

- If $v \in V_{PAC}$:
  We set randomly one of its neighbor v_1 at the first position. The three other neighbors of v are ordered such that if we look at v from v_1, the three remaining neighbors are ordered clockwise (Section 1). One of the three orders (defined up to circular permutations) fulfilling this condition is chosen randomly (Figure 3(a)).
- If $v \in V_{DB}$:
  Let us consider $w = n_=(v)$ and the two neighborhoods $V(v) = \{w, a, b\}$ and $V(w) = \{v, c, d\}$. The order on the neighborhood of v is set as $ord(v) = w, a, b$ and the order on w's neighborhood is set as $ord(w) = v, c, d$, whereby a, b, c, d are traversed clockwise when turning around the double bond for a given plane embedding (Figure 3(b)).

We denotes $\mathcal{OM}$ the set of ordered molecular graphs.

Definition 16. Set of molecular re-ordering functions

We define for each molecular ordered graph S, a set of re-ordering function Σ_S^M. Σ_S^M contains all the re-ordering functions σ such that:

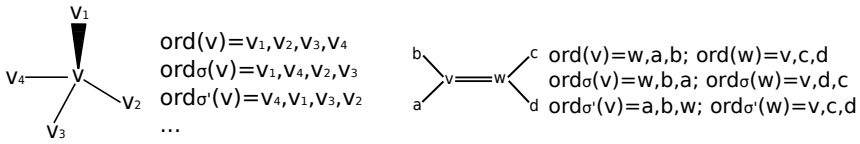
- For each v in V_{PAC} $\sigma(v)$ is an even permutation:

$$\forall v \in V_{PAC},\ \epsilon(\sigma(v)) = 1.$$

- For each v in V_{DB}, $\sigma(v)$ and $\sigma(n_=(v))$ have the same parity:

$$\forall v \in V_{DB},\ \epsilon(\sigma(v)) = \epsilon(\sigma(w)) \text{ with } w = n_=(v).$$

where ϵ denotes the signature of a permutation.



(a) Element of V_{PAC} (b) Two elements of V_{DB}

Fig. 3. Example of elements of V_{PAC} and of V_{DB} with their ordered list (top) and the ordered lists obtained using two permutations $\sigma \in \Sigma_S^M$ and $\sigma' \in \Sigma_S^M$

Proposition 6. For any molecular graph S, Σ_S^M is a group for the composition.

Proof. We have to show that Σ_S^M admits an identity element, is closed under composition and admits for each re-ordering function an inverse element.

1. Σ_S^M admits an identity element Id_S.
   Let us consider the re-ordering function Id_S such that:
   $$\forall v \in V_{ord}, Id_S(v) = Id_{|V(v)|}$$
   where Id_n is the identity permutation on Π_n.
   Since the identity is an even permutation we have:
   $$\begin{cases} \forall v \in V_{PAC} \ \epsilon(Id_S(v)) = 1 \\ \forall v \in V_{DB} \ \ \epsilon(Id_S(v)) = \epsilon(Id_S(n_=(v)) = 1 \end{cases}$$
   Thus by definition of Σ_S^M, $Id_S \in \Sigma_S^M$.
2. $\forall(\sigma, \sigma') \in \left(\Sigma_S^M\right)^2$, $\sigma \circ \sigma' \in \Sigma_S^M$.
   Let σ and σ' denote two re-ordering functions of Σ_S^M.
   - If $v \in V_{PAC}$:
     Since ϵ is a morphism between $\Pi_{[ord(v)|}$ and $(\{-1, 1\}, \times)$ we have:
     $$\forall v \in V_{PAC},\ \epsilon(\sigma(v) \circ \sigma'(v)) = \epsilon(\sigma(v))\epsilon(\sigma'(v)) = 1.1 = 1$$
   - If $v \in V_{DB}$:
     Let us consider $w = n_=(v)$. Since $\sigma(v)$ and $\sigma(w)$ on one hand and $\sigma'(v)$ and $\sigma'(w)$ on the other hand have a same signature, we have:
     $$\begin{aligned} \forall v \in V_{DB}, \epsilon(\sigma(v) \circ \sigma'(v)) &= \epsilon(\sigma(v))\epsilon(\sigma'(v)) \\ &= \epsilon(\sigma(w))\epsilon(\sigma'(w)) \\ &= \epsilon(\sigma(w) \circ \sigma'(w)) \end{aligned}$$
     Permutations $\sigma(v) \circ \sigma'(v)$ and $\sigma(w) \circ \sigma'(w)$ have thus a same parity.

   Thus by definition of Σ_S^M, $\sigma \circ \sigma' \in \Sigma_S^M$.
3. $\forall \sigma \in \Sigma_S^M$, $\exists \sigma^{-1} \in \Sigma_S^M$ such that $\sigma \circ \sigma^{-1} = Id_S$, where Id_S is the identity element of Σ_S^M.
   Let us consider σ^{-1} such that:
   $$\forall v \in V_{ord},\ \sigma^{-1}(v) = (\sigma(v))^{-1}.$$
   We have by Definition 10, $\sigma \circ \sigma^{-1} = Id_S$.
   We have to prove that, for each $\sigma \in \Sigma_S^M$, $\sigma^{-1} \in \Sigma_S^M$.

– If $v \in V_{PAC}$:
$$\epsilon(\sigma^{-1}(v)) = \epsilon(\sigma(v)^{-1}) = \epsilon(\sigma(v)) = 1$$
Thus $\sigma^{-1}(v)$ is even.
– If $v \in V_{DB}$:
Let us consider $w = n_{=}(v)$. Since $\sigma(v)$ and $\sigma(w)$ have a same parity and:
$$\begin{cases} \epsilon(\sigma^{-1}(v)) &= \epsilon(\sigma(v)) \\ \epsilon(\sigma^{-1}(w)) &= \epsilon(\sigma(w)), \end{cases}$$
$\sigma^{-1}(v)$ and $\sigma^{-1}(w)$ have also a same parity.

Thus by definition of Σ_S^M, $\sigma^{-1} \in \Sigma_S^M$.

As the composition of functions is associative and Σ_S^M admits an identity element for the composition, is closed under composition, and admits for each re-ordering function an inverse element we can conclude that Σ_S^M is a group for the composition.

□

Proposition 7. For any two molecular ordered graphs S and S' whose associated un-order graphs are isomorphic by a function f, and for any re-ordering function $\sigma \in \Sigma_S^M$, we have $\sigma \circ f^{-1} \in \Sigma_{S'}^M$:
$$\left.\begin{array}{l} \forall f \in \mathrm{Isom}(\hat{S}, \hat{S}'), \\ \forall \sigma \in \Sigma_S^M \end{array}\right) \sigma' = \sigma \circ f^{-1} \in \Sigma_{S'}^M.$$

Proof. Let us consider two molecular ordered graphs S and S' whose associated un-order graphs G and G' are isomorphic by a function f. We denote by f^{-1} the isomorphism between G' and G and define the re-ordering function σ' as $\sigma \circ f^{-1}$.

Let us prove that $\sigma' \in \Sigma_{S'}^M$.

– If $v' \in V'_{PAC}$:
We have by definition of σ', $\sigma'(v') = \sigma(v)$ with $v = f^{-1}(v')$.
As $v' \in V'_{PAC}$ we have , $\mu(v') =$ 'C' and $|V(v')| = 4$. Since f^{-1} is an isormorphism between G' and G we have:
$$\begin{cases} \mu(v) &= \mu(f^{-1}(v')) &= \mu(v') &= \text{'C'} \\ |V(v)| &= |V(f^{-1}(v'))| &= |V(v')| &= 4 \end{cases}$$
Thus by definition of V_{PAC}, $v \in V_{PAC}$. Moreover, since $\sigma'(v') = \sigma(v)$, $\sigma(v)$ has the same even parity than $\sigma'(v')$.
– If $v' \in V'_{DB}$:
Given $w' = n_{=}(v')$, we have by definition of σ':
$$\begin{cases} \sigma'(v') &= \sigma(v) \quad \text{with} \quad v = f^{-1}(v') \\ \sigma'(w') &= \sigma(w) \quad \text{with} \quad w = f^{-1}(w') \end{cases}$$

As v' and $w' \in V'_{DB}$ we have $\mu(v') = \mu(w') =$ 'C' and $|V(v')| = |V(w')| = 3$. Since f^{-1} is an isormorphism between G' and G we have:

$$\begin{cases} \mu(v) & = \mu(f^{-1}(v')) & = \mu(v') & = \text{'}C\text{' and} \\ \mu(w) & = \mu(f^{-1}(w')) & = \mu(w') & = \text{'}C\text{'} \\ \\ |V(v)| & = |V(f^{-1}(v'))| & = |V(v')| & = 3 \text{ and} \\ |V(w)| & = |V(f^{-1}(w'))| & = |V(w')| & = 3 \end{cases}$$

As $v' \in V'_{DB}$, it exists an edge $e'(v', w')$ connecting v' and w' with a label $\nu(e') = 2$. Such an edge is preserved by the isomorphism f^{-1} between G' and G and it thus exists an edge $e(v, w)$ in G between $v = f^{-1}(v')$ and $w = f^{-1}(w')$ with $\nu(e) = 2$.
We have thus by definition of V_{DB}, $\{v, w\} \subset V_{DB}$ and since $\sigma \in \Sigma_S^M$ we have by definition of Σ_S^M:

$$\epsilon(\sigma(v)) = \epsilon(\sigma(w))$$

Therefore, by definition of σ':

$$\epsilon(\sigma'(v')) = \epsilon(\sigma(v)) = \epsilon(\sigma(w)) = \epsilon(\sigma'(w'))$$

Permutations $\sigma'(v')$ and $\sigma'(w')$ have thus the same parity.

Thus $\sigma' = \sigma \circ f^{-1} \in \Sigma_{G'}^M$.

□

Theorem 2. *The set of molecular re-ordering functions $\Sigma^M = \{\Sigma_S^M,\ S \in \mathcal{OM}\}$ is a valid family of re-ordering functions. Therefore the equivalent order relationship based on this family is an equivalence relationship.*

Proof. The set of molecular re-ordering functions Σ^M is a valid family of re-ordering functions by Proposition 6 and 7. Thus by Theorem 1, the equivalent order relationship based on this family is an equivalence relationship.

□

Remark 1. Let $S = (G, ord)$ with $G = (V, E, \mu, \nu)$ denote a molecular ordered graph. We have, by construction of re ordering functions, the following property:

If we select for each vertex $v \in V_{ord}$ a neighbor n_v, we can always find a re-ordering function σ of Σ_S^M such that the ordered list of each vertex v of V_{ord} in $\sigma(S)$ starts by its selected neighbor n_v.

Given our encoding of the relative positioning of atoms by orders defined in this section, we encode the spatial configuration of atoms within the neighborhood of each of its vertex. Our equivalence relationship between molecular ordered graphs allows to check if two molecules have a same spatial configuration.

Stereocenters are defined as stereo vertices (Definition 14). Indeed, a vertex is a stereo vertex if any permutation of its neighbors produces an ordered graph with a non-equivalent order, called a different stereoisomer within the chemistry framework.

3 Equivalence order relationship between ordered tree

Let us now restrict our attention to acyclic graphs in order to obtain an efficient way to determine if two molecular graphs have equivalent orders.

Given a rooted tree, the father of each node v is denoted by p_v. The tree itself is denoted by $\hat{T} = (r, G)$ where r denotes the root of the tree and $G = (V, E, \mu, \nu)$ the acyclic graph associated to $\hat{T}$.

Definition 17. Ordered rooted tree

An ordered rooted tree $T = (\hat{T}, ord)$ with $\hat{T} = (r, G)$ is an ordered structured object. Its associated acyclic labeled graph is $G = (V, E, \mu, \nu)$. The function ord maps each internal vertex to an ordered list of its children:

$$ord \begin{cases} V_{ord} \rightarrow & V^* \\ v \quad \mapsto & v_1 \dots v_n \end{cases}$$

where $\{v_1, \dots, v_n\}$ denotes the children of v.

We denote by $\mathcal{OT}$ the set of ordered trees.

Note that the function ord is defined on all internal vertices of the tree we have thus:

$$\forall T = (\hat{T}, ord) \in \mathcal{OT}, \text{ with } \begin{pmatrix} \hat{T} = (r, G), \\ G = (V, E, \mu, \nu) \end{pmatrix} V_{ord} = V - Leaf(T)$$

where $Leaf(T)$ denotes the set of leaves of T.

Moreover, the order relationship of each vertex of an ordered rooted tree is defined on all its children, i.e. all its neighbors but its parent. We have thus:

$$\forall T = (\hat{T}, ord) \in \mathcal{OT}, \text{ with } \hat{T} = (r, G) \begin{cases} & |ord(r)| = |V(v)| \\ \forall v \in V_{ord}, v \neq r & |ord(v)| = |V(v)| - 1 \end{cases}$$

An isomorphism between rooted trees may be considered as an isomorphism between graphs which maps the roots of both trees one on the other. Hence the set of ismorphisms $\text{Isom}(\hat{T}, \hat{T}')$ between two rooted tree may be considered as a subset of the isomorphisms between the associated acyclic graphs: $\text{Isom}(\hat{G}, \hat{G}')$ (Definition 6).

Given these isomorphisms we define for $\mathcal{OT}$ ordered isomorphisms between ordered rooted trees (Definition 7). Such isomorphisms preserve both the structure of both trees and their orderings. The order defined on each vertex of the trees belonging to $\mathcal{OT}$ depends of the considered application. The valid family of re-ordering functions Σ (Definition 11), which may be defined on $\mathcal{OT}$ also depends of this application. Given both orders and a valid family of re-ordering functions we can build an equivalence relationship (Definition 12 and Theorem 1) between ordered trees encoding the fact that up to re-orderings two rooted trees are structurally similar and have a same order.

Following [10], we associate to each ordered rooted tree T, an unique depth-first string encoding $\text{DFSE}(T)$. This string is based on the sequence of node and edge labels obtained by traversing the tree in a depth-first order and uses respectively \$ and # to represent backtracks and the end of the string encoding.

Definition 18. Depth-First String Encoding

The depth first string encoding of an ordered rooted tree $T = (\hat{T}, ord)$ denoted by $\mathrm{DFSE}(T)$ is defined as the sequence of node and edge labels encountered when traversing T using a depth-first method based on the order defined by function ord. Each backtrack during this traversal is encoded by the symbol $'\$'$ while the end of the string is encoded by the symbol $'\#'$.

Remark 2. As shown by [10](Lemma 2.2), two isomorphic ordered trees have the same depth-first string encoding and conversely.

$$T_1 \underset{o}{\simeq} T_2 \Leftrightarrow \mathrm{DFSE}(T_1) = \mathrm{DFSE}(T_2)$$

Definition 19. Depth-first canonical string and Depth-first canonical form of a tree

Let us consider an ordered tree T and a valid family of re-ordering functions Σ:

- The depth-first canonical string $\mathrm{DFCS}_\Sigma(T)$ of T is the minimal depth-first string encoding among all possible ordered trees $\sigma(T)$ obtained by applying $\sigma \in \Sigma_T \in \Sigma$ on T:

$$\mathrm{DFCS}_\Sigma(T) = \min_{\sigma \in \Sigma_T} \mathrm{DFSE}(\sigma(T))$$

- The depth-first canonical form $\mathrm{DFCF}_\Sigma(T)$ of T according to Σ is the ordered tree $\sigma(T)$ whose depth-first string encoding is minimal (and thus equals to $\mathrm{DFCS}_\Sigma(T)$):

$$\mathrm{DFSE}(\mathrm{DFCF}_\Sigma(T)) = \mathrm{DFCS}_\Sigma(T)$$

  The depth-first canonical form is unique up to ordered isomorphism (Remark 2).

Proposition 8. Two ordered rooted trees $T_a = (\hat{T_a}, ord_a)$ and $T_b = (\hat{T_b}, ord_b)$ have equivalent orders according to a valid family of re-ordering functions Σ iff their depth-first canonical string according to Σ are equals:

$$T_a \underset{\Sigma}{\simeq} T_b \Leftrightarrow \mathrm{DFCS}_\Sigma(T_a) = \mathrm{DFCS}_\Sigma(T_b)$$

Proof. Let us first prove that $\mathrm{DFCS}_\Sigma(T_a) = \mathrm{DFCS}_\Sigma(T_b) \Rightarrow T_a \underset{\Sigma}{\simeq} T_b$ and then the reverse implication.

1. We suppose that $\mathrm{DFCS}_\Sigma(T_a) = \mathrm{DFCS}_\Sigma(T_b)$.
   Let us denote $\sigma_a \in \Sigma_{T_a}$ the re-ordering function such that $\sigma_a(T_a) = \mathrm{DFCF}_\Sigma(T_a)$ and $\sigma'_b \in \Sigma_{T_b}$ the one such that $\sigma'_b(T_b) = \mathrm{DFCF}_\Sigma(T_b)$.
   By definition $\mathrm{DFCS}_\Sigma(T_a) = \mathrm{DFSE}(\sigma_a(T_a))$ and $\mathrm{DFCS}_\Sigma(T_b) = \mathrm{DFSE}(\sigma'_b(T_b))$.
   Since $\mathrm{DFCS}_\Sigma(T_a) = \mathrm{DFCS}_\Sigma(T_b)$ we have $\mathrm{DFSE}(\sigma_a(T_a)) = \mathrm{DFSE}(\sigma'_b(T_b))$, and thus $\sigma_a(T_a) \underset{o}{\simeq} \sigma'_b(T_b)$ (Remark 2).

As Σ_{T_a} is a group (Definition 11), it exists a re-ordering function $\sigma_a^{-1} \in \Sigma_{T_a}$ such that $\sigma_a^{-1}(\sigma_a(T_a)) = T_a$.
Let us denote by f the ordered isomorphism between $\sigma_a(T_a)$ and $\sigma'_b(T_b)$. By Definition 11, since $\sigma_a(T_a) \underset{o}{\simeq} \sigma'_b(T_b)$, the re-ordering function $\sigma_a^{-1} \in \Sigma_{T_a}$ should be equivalent to some re-ordering function σ_b^{-1} in Σ_{T_b}. In other words:

$$\exists \sigma_b^{-1} \in \Sigma_{T_b} \text{ such that } \forall v \in V_a, \sigma_a^{-1}(v) = \sigma_b^{-1}(f(v)).$$

We have thus by Lemma 1: $\sigma_a^{-1}(\sigma_a(T_a)) \underset{o}{\simeq} \sigma_b^{-1}(\sigma'_b(T_b))$. Therefore $T_a \underset{o}{\simeq} \sigma_b^{-1}(\sigma'_b(T_b))$. As Σ_{T_b} is a group, $\sigma_b^{-1} \circ \sigma'_b \in \Sigma_{T_b}$, so $T_a \underset{\Sigma}{\simeq} T_b$.

2. We suppose that $T_a \underset{\Sigma}{\simeq} T_b$.
We denote $\sigma'_b \in \Sigma_{T_b}$, the re-ordering function such that $\sigma'_b(T_b) = \mathrm{DFCF}_\Sigma(T_b)$. As $T_a \underset{\Sigma}{\simeq} T_b$, $\exists \sigma \in \Sigma_{T_a}$ such that $\sigma(T_a) \underset{o}{\simeq} T_b$. Let us denote by f the ordered isomorphism between T_b and $\sigma(T_a)$.
By Definition 11, since $T_b \underset{o}{\simeq} \sigma(T_a)$, the re-ordering function $\sigma'_b \in \Sigma_{T_b}$ should be equivalent to some re-ordering function σ'_a in Σ_{T_a}. In other words:

$$\exists \sigma'_a \in \Sigma_{T_a} \text{ such that } \forall v \in V_b, \sigma'_b(v) = \sigma'_a(f(v)).$$

We have thus by Lemma 1: $\sigma'_a(\sigma(T_a)) \underset{o}{\simeq} \sigma'_b(T_b)$. Therefore

$$\mathrm{DFSE}(\sigma'_a(\sigma(T_a))) = \mathrm{DFSE}(\sigma'_b(T_b)) = \mathrm{DFCS}_\Sigma(T_b)$$

Since Σ_{T_a} is a group, we have $\sigma'_a \circ \sigma \in \Sigma_{T_a}$. Therefore:

$$\mathrm{DFCS}_\Sigma(T_b) = \mathrm{DFSE}(\sigma'_a(\sigma(T_a))) \geq \mathrm{DFCS}_\Sigma(T_a).$$

Both cases being symmetric the reverse inequality is shown by considering $\sigma'_a \in \Sigma_{T_a}$ such that $\sigma'_a(T_a) = DFCF_\Sigma(T_a)$. Therefore:

$$\mathrm{DFCS}_\Sigma(T_a) = \mathrm{DFCS}_\Sigma(T_b).$$

By 1 and 2 we have proven that

$$T_a \underset{\Sigma}{\simeq} T_b \Leftrightarrow \mathrm{DFCS}_\Sigma(T_a) = \mathrm{DFCS}_\Sigma(T_b).$$

□

Let Σ be a valid family of re-ordering functions. An ordered tree T can have two vertices connected to a same parent and whose associated subtrees are equivalent according to Σ. Any path from a leaf of T passing through one of these two vertices is equivalent to another path passing through the other vertex. From a more global point of view, a permutation exchanging these two subtrees on the depth-first canonical form of T would lead to an isomorphic ordered tree. We thus consider that these two vertices are equivalent.

Definition 20. Equivalent ordered sub-tree

Let us consider an ordered rooted tree T and a valid family of re-ordering functions Σ.

- Two child of a same parent whose associated rooted sub-trees are equivalent according to Σ are said to be equivalent:

$$v_i \sim v_j \Leftrightarrow \exists (v,\sigma) \in V \times \Sigma_T \; s.t. \begin{cases} p_{v_i} = p_{v_j} = v \text{ and} \\ (\sigma(v))(i) = j \text{ and } \mathrm{DFCF}_\Sigma(\sigma(T)) \underset{o}{\simeq} \mathrm{DFCF}_\Sigma(T) \end{cases} \tag{2}$$

- The representative of each class is defined as the vertex with the minimal index within the ordered list of children of its parent:

$$\forall i \in \{1,\ldots,n\} \; rep(v_i) = min\{j \mid v_j \sim v_i\}. \tag{3}$$

  The representative of a class is properly defined since two equivalent nodes must have a same parent.

3.1 Ordered trees and re-ordering functions encoding of an acyclic molecule

To define a molecular ordered tree $T = (\hat{T}, ord_T)$, from an acyclic molecular ordered graph $G = (\hat{G}, ord_G)$, we have to define a root and for each vertex an order on its children. By definition, the function ord_T of T is defined on (Definition 17):

$$V^T_{ord} = V - Leaf(T)$$

where $Leaf(T)$ denotes the set of leaves of T.

On the other end, the set of vertices of an ordered molecular graph $G = (\hat{G}, ord_G)$ on which the function ord_G is defined (Definition 15) is equal to:

$$V^M_{ord} = V_{PAC} \cup V_{DB}$$

Since a molecular ordered graph does not provide an order for all vertices, while all vertices of an ordered tree but its leaves are ordered, the definition of an order on a molecular tree imposes to fix a priori the order on the child of some vertices if this order is not provided by the molecular ordered graph.

Definition 21. Molecular ordered tree

A molecular ordered tree $T = (\hat{T}, ord_T)$ with $\hat{T} = (r, G_T)$ is defined from a molecular ordered graph $G = (\hat{G}, ord_G)$ with $\hat{G} = (V, E, \mu, \nu)$ by setting $G_T = \hat{G}$, $r \in V$ and by defining ord_T from ord_G as follows:

Given the root r, let us consider a re-ordering function of Σ^M_G such that (Remark 1):

$$\forall v \in V^M_{ord} - \{r\}, \; ord_{\sigma(G)}(v) = p_v.v_1 \ldots .v_n$$

If $V \neq \{r\}$, the function ord_T on r is set equals to:

$$ord_T(r) = \begin{cases} ord_{\sigma(G)}(r) & \text{if } r \in V_{ord}^M \\ random(child(r)) & \text{otherwise} \end{cases}$$

where $random(child(r))$ denotes a random ordering of the child of r ($child(r)$).

For all other vertices:

- If $v \in V_{ord}^T - V_{ord}^M - \{r\}$

$$ord_T(v) = random(child(v))$$

- If $v \in V_{ord}^T \cap V_{ord}^M - \{r\}$
  We have $ord_{\sigma(G)}(v) = p_v.v_1.\ldots.v_n$ where $v_1 \ldots v_n$ corresponds to an ordering of child(v). We set thus $ord_T(v)$ as:

$$ord_T(v) = v_1 \ldots v_n$$

Definition 22. Set of molecular re-ordering functions for tree

The set of re-ordering functions Σ_T^M is defined by :

- If $v \in V_{ord}^T - V_{ord}^M$,
  Permutation $\sigma(v)$ can be any permutation.
  Therefore the order on those vertex have no influence, it only allows us to determine the depth-first string encoding and the depth-first canonical string of a molecular ordered tree.
- If $v \in V_{PAC}$,
  $\sigma(v)$ is an even permutation:

$$\epsilon(\sigma(v)) = 1$$

- If $v \in V_{DB}$,
  Permutations $\sigma(v)$ and $\sigma(n_=(v))$ have a same parity :

$$\epsilon(\sigma(v)) = \epsilon(\sigma(w)) \text{ with } w = n_=(v)$$

Given a unique code associated to an ordered rooted tree, the chirality of a vertex may be efficiently tested if one can transpose Definition 14 to ordered rooted trees:

Proposition 9. Let $T = (\hat{T}, ord_T)$ with $\hat{T} = (r, G)$ be an ordered rooted tree encoding an acyclic molecule, and Σ_T^M the set of molecular re-ordering functions for T. r is a stereo vertex if:

$$\forall (i,j) \in \{1, \ldots, |V(r)|\}^2 \text{ with } i \neq j,\ T \underset{\Sigma}{\not\simeq} \tau_{i,j}(T)$$

where $\tau_{i,j}$ is a re-ordering function equals to the identity on any vertex but r where it permutes children of index i and j in the ordered list of r.

Proof. Using acyclic molecular graphs, an equivalent ordered isomorphism between ordered rooted trees corresponds to an equivalent ordered isomorphism between ordered graphs with an additional constraint on the mapping of both roots. If we can find an isomorphism between T and $\tau_{i,j}(T)$ such an isomorphism f satisfies $f(r) = r$ and also corresponds to an isomorphism between ordered graphs. Conditions of Definition 14 are thus violated and r is not a stereo vertex. The reverse implication may be demonstrated using the same type of reasoning. □

4 From a global to a local characterization of stereo information

Proposition 9 allows us to determine if a vertex induces a stereo property for a molecule. Such a proposition concerning the whole molecule induces a global characterization of stereo information. However, such a proposition does not allow to characterize the minimal subgraph of a molecule which induces the stereo property of a vertex. Using acyclic graphs, such a minimal subgraph corresponds to the smallest ordered sub-tree, rooted on a stereo vertex v which allows to characterize v using Proposition 9.

4.1 Minimal stereo subtree of an asymmetric carbon

Let v be a stereo vertex representing an asymmetric carbon ($v \in V_{PAC}$). We denote its neighbors $v_1, \ldots, v_4$. We consider the ordered tree T rooted on v and described in Section 3 and the family of re-ordering functions for molecular tree Σ_T^M. We note $T_1, \ldots, T_4$ the subtrees of T rooted on the children of v. For any $i \in \{1, 2, 3, 4\}$ we denote T_i^j the subtree of T_i composed of all nodes with a depth lower than j. According to Proposition 9, the stereo information of v may be characterized from its subtrees T_i^j iff all pairs of subtrees are not equivalent. Indeed, in such a case no transposition of two subtrees T_i^j and $T_k^{j'}$ can induce a rooted tree with equivalent order. Therefore for each $i \in \{1, 2, 3, 4\}$, we define the minimal subtree associated to v_i as $T_i^{j^*(i)}$ with:

$$j^*(i) = \min\{j \mid \forall k \in \{1, \ldots, 4\} - \{i\},\ T_i^j \underset{\Sigma}{\not\simeq} T_k^j\}.$$

For example in Figure 4, the root of T_1 is a Chloro atom while the root of each other T_i is an oxygen atom, thus the subtree T_1^1 reduced to the Chloro atom has a sufficient depth to be distinguished from all other subtrees T_i^k, $i \neq 1$ and we have $j^*(1) = 1$. The minimal stereo subtree of v is the subtree of T rooted on v, where v has for children $T_1^{j^*(1)}, \ldots, T_4^{j^*(4)}$. The asymmetric carbon is then represented by the depth-first canonical string of this tree according to Σ_T^M.

To find $j^*(i)$, we increase j for each T_i^j until $T_i^j \underset{\Sigma}{\not\simeq} T_k^j$ for each $k \in \{1, \ldots, 4\}$, $k \neq i$. At each iteration we compute $\mathrm{DFCS}_{\Sigma^M}(T_i^j)$ for each $i \in \{1, \ldots, 4\}$. Therefore the calculus of the minimal stereo subtree of v is performed in $\mathcal{O}((\max_i |T_i^{j^*(i)}|)^2)$ which is bounded by $\mathcal{O}(|V|^2)$.

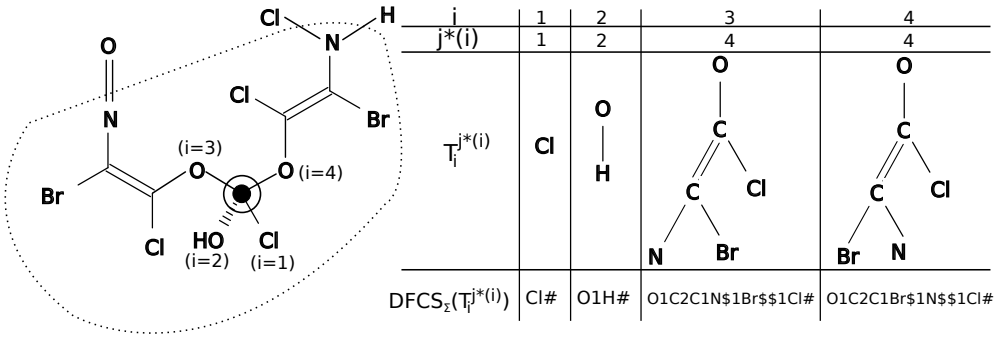


Fig. 4. Left: An asymmetric carbon ⦿ with its minimal stereo subtree (*surrounded by a dotted line*). Right: minimal subtrees rooted on its children.

4.2 Minimal stereo subtree of double bond

Let v_a be one carbon of a double bond, $v_a \in V_{DB}$. We denote $n_=(v_a) = v_b$ and $e = (v_a, v_b)$ the double bond between them. Let us denote by v_a^1 and v_a^2 the two remaining neighbors of v_a. Considering the ordered tree T rooted on v_a, v_a is a stereo vertex only if the subtrees rooted on the children of v_a do not have equivalent orders (Proposition 9). This implies that the two subtrees rooted on v_a^1 and v_a^2 do not have equivalent orders. This last necessary condition is however not sufficient. Indeed if the subtrees rooted on the remaining neighbors v_b^1 and v_b^2 of v_b have equivalent orders, then one can apply a re-ordering function $\sigma \in \Sigma_T^M$ on T which simultaneously permutes the subtrees rooted on v_a^1 and v_a^2 and the subtrees rooted on v_b^1 and v_b^2 (by definition of V_{DB} and Σ^M). The resulting rooted tree $\sigma(T)$ has an equivalent order to T (Definition 12) but also to $\tau(T)$, where τ permutes only vertices v_a^1 and v_a^2 in the ordered list of children of v_a. In such a case, v_a is not a stereo vertex(Proposition 9). Therefore, if v_b is not a stereo vertex, v_a is also not a stereo vertex and conversely.

Hence v_a and v_b are stereo vertices, only if the two following conditions are satisfied:

- subtrees rooted on v_a^1 and v_a^2 do not have equivalent orders and
- subtrees rooted on v_b^1 and v_b^2 also do not have equivalent orders.

In order to encode this constraint, we define as in Section 4.1 the minimal subtrees rooted on v_a^1 (T_a^1) and v_a^2 (T_a^2) with non-equivalent orders together with the minimal subtrees rooted on v_b^1 (T_b^1) and v_b^2(T_b^2) with non-equivalent orders. We denote by T_a and T_b the two ordered rooted trees rooted on v_a and v_b. The subtrees of these two roots being respectively (T_a^1, T_a^2) and (T_b^1, T_b^2).

The tree encoding the chirality of the double bond is then defined as an ordered rooted tree, whose root corresponds to a virtual vertex (not corresponding to any atom) connected to the two subtrees T_a and T_b. As in Section 4.1, the computation of the minimal stereo subtree is bounded by $\mathcal{O}(|V|^2)$. Figure 5a represents a double bond between two carbon atoms with its minimal stereo subtree (Figure 5b).

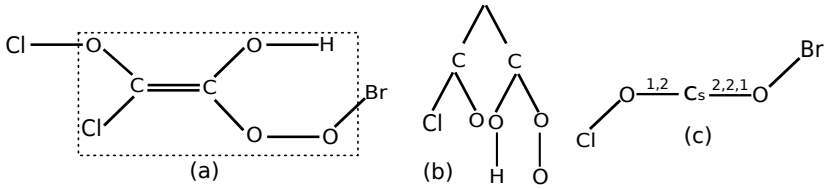


Fig. 5. A double bond (a), its minimal stereo subtree (b) and its contraction (c).

4.3 Graph Contraction

Using results in Section 4.1 and 4.2, each stereo vertex may be associated to a minimal stereo subtree and a depth-first canonical string according to Σ^M representing it (Section 3). However, properties of a molecule are both determined by its set of minimal stereo subtrees and by relationships between these trees and the remaining part of the molecule. In order to obtain a local characterization of such relationships, we propose to contract the minimal stereo subtree of each stereo vertices.

Let us consider a stereo vertex s and its minimal stereo subtree $T = (\hat{T}, ord_T)$, with $\hat{T} = (r, G_T)$, $G_T = (V_T, E_T, \mu, \nu)$ associated to a depth-first canonical string according to Σ^M, we denote this string $c_s = \text{DFCS}_{\Sigma^M}(T)$. We define for this tree a set of connection vertices:

$$V_{\text{con}} = \{v \in \text{Leaf}(T) \mid d(v) > 1\}$$

and a set of edges to contract:

$$K_T = E_T - E_{\text{con}} \text{ with } E_{\text{con}} = \{(v, p_v) \in V_{\text{con}} \times V_T\}.$$

The contraction of K_T creates a new graph $G_s = (V_s, E_s)$, with a contracted node n_s labeled by c_s and $V_s = V - (V_T - V_{\text{con}}) \cup \{n_s\}; E_s = E - K_T$ (Figure 5c).

Each edge of E_{con} connects an element l of V_{con} to n_s in G_s. The label of $e = (n_s, l)$ has to encode the position of l in the minimal stereo subtree. We thus consider the path connecting r to l in the minimal stereo subtree:

$$CP(l) = v_1, .., v_n \text{ with } v_1 = r \text{ and } v_n = l.$$

Let us denote i_j the index of v_j in the ordered list of children of p_{v_j}. The sequence $i_2 \dots i_n$ defines a unique path in the stereo subtree associated to n_s. Such a sequence may thus be considered as a proper label of edge e. However as mentioned in Section 3, some paths may pass through equivalent subtrees and should thus be considered as equivalent. In order to encode such an equivalence relationship we define the label of e as:

$$\nu(e) = \bigodot_{i=2}^{n} rep(v_i)$$

where rep is defined by Equation 3, Definition 20 and $\bigodot$ denotes the concatenation operator.

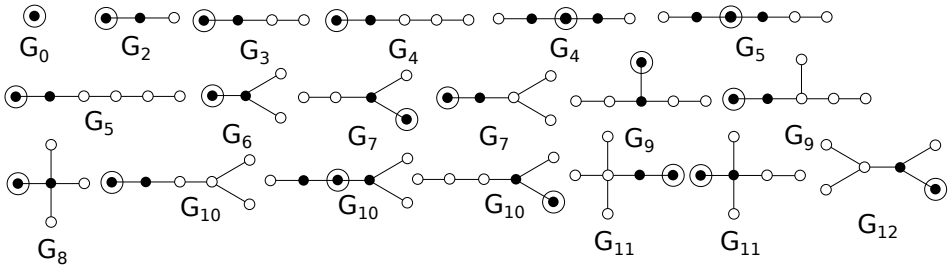


Fig. 6. The set of stereotreelet with n_s(⊙), elements of V_{con}(•), elements of $V - V_{con}$(○)

4.4 StereoTreelet

For each stereo vertex s we have a graph G_s. The stereotreelets of G_s are defined as all subtrees of G_s whose size is lower than 6 and which include n_s. Since each neighbors v of n_s corresponds to a leaf of the minimal stereo tree of s, the edge (v, n_s) is already encoded within the code c_s of n_s. Consequently, we impose that each neighbor v of n_s in a stereotreelet must have at least another neighbor (different of n_s). This constraint induces the set of stereotreelets represented in Fig. 6. The set of stereotreelet $\mathcal{T}(G)$ of G is defined as the union of stereotreelets of each G_s.

When all stereotreelets of G have been enumerated, we compute its spectrum $s(G)$ which corresponds to a vector representing the treelet distribution. Each component of this vector is equal to the frequency of a given stereotreelet t: $s(G) = (f_t(G))_{t \in \mathcal{T}(G)}$ with $f_t(G) = |(t \subseteq G)|$. The kernel between two graphs G and G' is defined as a sum of kernels between the different number of treelets common to both graphs:

$$k(G, G') = \sum_{t \in \mathcal{T}(G) \cap \mathcal{T}(G')} K(f_t(G), f_t(G')).$$

5 Experiments

We have tested our method on a dataset of acyclic chiral molecules [11] related to a regression problem. This dataset is composed of 90 molecules together with their optical rotations. In practice, we only select 35 molecules, since almost all molecules have only one stereocenter, and for 55 molecules this stereocenter is unique in the dataset. Such molecules correspond to a property represented only once in the dataset which can thus not be accurately predicted. The property to predict, the optical rotation, is connected with chirality and has a standard deviation of 38.25 for the 35 selected molecules.

For our experiment we use a leave-one-out cross-validation on the dataset to predict the optical rotation of each molecule. The predicted rotations are computed by using both kernel ridge regression and the weighted mean of known

Table 1. Optical rotation prediction for the acyclic chiral dataset.

Method	Kernel Ridge		Weighted Average		Gram's matrix
	Average Error	RMSE	Average Error	RMSE	computations (s)
Random Kernel	31.7	39.5	32.0	39.3	0.03
KMean [12]	31.0	38.7	32.3	39.6	153.84
Treelet Kernel [1]	26.0	33.9	28.9	37.4	0.49
Stereotreelet Kernel	21.0	25.6	11.6	**16.3**	0.13

values using the similarity measure provided by the kernel

$$\hat{y} = \frac{\sum_i k(G_i, G) \times y_i}{\sum_i k(G_i, G)}$$

We present in Table 1 the average errors, Root Mean Squared Errors (RMSE) and computation times of the Gram matrix for our method and the ones of [12, 1] which do not take into account stereo information. Results obtained by using a random Gram matrix are also shown.

Weighted mean provides much better results for our kernel since on this dataset each molecule has a non null similarity with a reduced number of molecules (less than 10). Such a reduced number of data do not allow kernel ridge regression to perform reliable prediction. Other methods provide similar results than those obtained using a random Gram matrix. These results are also comparable with the variance of the dataset. Such a result means that the similarity measures provided by alternative kernels are not correlated with the property to predict. This last point may be explained by the fact that optical rotation is connected to stero information which is not encoded by these kernels.

6 Conclusion

In this report we proposed a new model which allows to encode the relative positioning of vertices within a graph. Such a model is quite flexible since it does not require any coordinate information and may be defined on only some vertices of a graph. We also introduced the notion of minimal stereo subtree which, in the acyclic case, corresponds to the minimal subgraphs which allows to explain the stereo property of a vertex.

We applied this model to encode the stereo information of molecules and vertices. Based on the minimal stereo subtree of each vertex we defined a graph kernel between stereoisomers. Our experiments show promising results and our future work will consist to create larger datasets and to extend our method to graphs including cycles.

References

1. Gaüzère, B., Brun, L., and Villemin, D. *Two new graphs kernels in chemoinformatics.* In Pattern Recognition Letters, April 2012.

2. Brown, J., Urata, T., Tamura, T., Arai, M., Kawabata, T., and Akutsu, T. *Compound analysis via graph kernels incorporating chirality* J. Bioinform. Comp. Bio. S1: 6381, 2010.
3. Golbraikh, A., and Tropsha, A. *QSAR Modeling Using Chirality Descriptors Derived from Molecular Topology.* J. Chem. Inf. Comput. Sci. 2003, 43, 144-154.
4. Cherqaoui, D., and Villemin, D. *Use of a neural network to determine the boiling point of alkanes.* J. Chem. Soc. Faraday Trans. 90, 97102, 1994.
5. Poezevara, G., Cuissart, B., and Crémilleux, B. *Discovering emerging graph patterns from chemicals.* In Proc. of the 18th ISMIS 2009, pages 4555, Prague, 2009. LNCS.
6. Brun, L., Conte, D., Foggia, P., Vento, M., Villemin, D. *Symbolic learning vs. graph kernels: An experimental comparison in a chemical application.* In: Proceedings of the 14th Conference on Advances in Databases and Information Systems (ADBIS 2010), pp. 3140, 2010.
7. Kashima, H., Tsuda, K., and Inokuchi, A. *Kernels for graphs*, chapter 7, pages 155170. MIT Press, 2004.
8. Mahé, P., and Vert, J.-P. *Graph kernels based on tree patterns for molecules.* Machine Learning, 75(1):335, October 2008.
9. Shervashidze, N., Vishwanathan, S. V., Petri, T. H., Mehlhorn, K., and Borgwardt, K. M. *Efficient graphlet kernels for large graph comparison.* In Proceedings of AIStats, pages 488495, 2009.
10. Chi, Y., Yang, Y., and Muntz, R. R. *Canonical forms for labeled trees and their applications in frequent subtree mining.* Knowledge and Information Systems, 8(2):203234, 2005.
11. Zhu, H. J., Ren, J., and Pittman C. U., Jr. *Matrix model to predict specific optical rotations of acyclic chiral molecules.* Tetrahedron 2007, 63, 22922314.
12. Suard, F., Rakotomamonjy, A., and Bensrhair., A. *Kernel on bag of paths for measuring similarity of shapes.* In European Symposium on Artificial Neural Networks,2002.