Noyau de Treelets appliqué aux graphes étiquetés et aux graphes de cycles

Benoit Gaüzère, Luc Brun, Didier Villemin

Noyau de Treelets Appliqué aux Graphes Étiquetés et aux Graphes de Cycles.

Méthodes à noyaux pour la chémoinformatique

Benoit Gaüzère* — **Luc Brun*** — **Didier Villemin****

* *GREYC UMR CNRS 6072, ENSICAEN, Université de Caen Basse Normandie*

** *Laboratoire de Chimie Moléculaire et Thio-organique UMR 6507, ENSICAEN*

benoit.gauzere@ensicaen.fr, luc.brun@ensicaen.fr, didier.villemin@ensicaen.fr

RÉSUMÉ. *La chémoinformatique utilise des méthodes issues de l'informatique, plus particulièrement la théorie des graphes et l'apprentissage automatique, afin de classifier ou prédire les propriétés de bases de molécules. Dans ce contexte, les noyaux sur graphes fournissent une approche intéressante en combinant les méthodes d'apprentissage automatique et la représentation naturelle des molécules par graphes. Parmi les méthodes basées sur les noyaux sur graphes, la décomposition du graphe en sous structures représente une importante famille de noyau. Dans cet article, nous présentons deux extensions d'un noyau précédemment basé sur les sous structures non étiquetées à l'énumération de sous structures étiquetées et à la prise en compte de l'information cyclique des molécules. Nous proposons également des méthodes de sélection de variables permettant de pondérer un ensemble de sous structures afin d'améliorer la précision de la prédiction.*

ABSTRACT. *Chemoinformatics consists to discover or predict molecule's properties through informational techniques.Computer science's research fields mainly concerned by chemoinformatics are machine learning and graph theory. From this point of view, graph kernels provide a nice framework combining machine learning and graph theory techniques. Among methods based on graph kernels, an major family is based on a decomposition of a graph into substructures. In this paper, we present two extensions of a kernel previously based on unlabeled sub structures to labeled substructures and cyclic information. We also propose selection methods which allow us to weight the set of considered sub structures in order to improve prediction accuracy.*

MOTS-CLÉS : *Chémoinformatique, Noyaux sur Graphes, Apprentissage Automatique.*

KEYWORDS: *Chemoinformatics, Graph Kernels, Machine Learning.*

1. Introduction

L'objet de la chémoinformatique est de prédire ou analyser les propriétés des molécules à l'aide de méthodes informatiques. Une des bases de ce domaine est le *principe de similarité* qui stipule que deux molécules ayant une structure similaire possèdent des activités et/ou des propriétés similaires. Une molécule peut être naturellement représentée par un graphe moléculaire $G = (V, E, \mu, \nu)$, où le graphe non étiqueté (V, E) code la structure de la molécule tandis que μ assigne à chaque noeud son élément chimique correspondant et ν encode le type de liaison entre deux atomes. Une majorité de méthodes est basée sur la corrélation entre un ensemble de descripteurs associés à la molécule et une propriété à prédire. La liste de descripteurs peut être calculée à partir de la structure, des propriétés physiques ou bien encore de l'activité biologique de la molécule (Todeschini *et al.*, 2000). L'ensemble de ces descripteurs est regroupé au sein d'un vecteur de taille fixe. Cette dernière représentation permet d'appliquer à la chémoinformatique un vaste ensemble de méthodes numériques définies dans le cadre de l'analyse de données et des méthodes d'apprentissage. Cependant, la définition d'un vecteur à partir d'un choix de descripteurs implique une sélection a priori de l'information pertinente. De plus, pour certaines applications, la définition du vecteur de caractéristiques reste heuristique. Une seconde famille de méthodes reposant sur la théorie des graphes peut être décomposée en deux sous familles. La première sous famille (Poezevara *et al.*, 2009), issue de l'analyse de données, consiste à trouver des sous-graphes ayant une importante différence de fréquence d'apparition entre deux ensembles d'exemples positifs et négatifs. La seconde sous famille (Brun *et al.*, 2010), reliée à l'apprentissage automatique, construit une description structurelle de chaque classe de molécules, de façon à ce que la classification soit effectuée par un appariement structurel entre la molécule à traiter et chacun des prototypes. Cette famille est toutefois essentiellement restreinte aux problèmes de classification.

Les noyaux sur graphes peuvent être vus comme une mesure de similarité symétrique entre deux graphes. Si $\mathcal{G}$ est un espace de graphe, un noyau k de $\mathcal{G} \times \mathcal{G}$ dans $\mathbb{R}$ associe au jeu de données $\{G_1, \ldots, G_n\}$ une matrice de Gram K définie par $K_{(i,j)} = k(G_i, G_j)$. Le noyau k est dit semi défini positif si il vérifie la propriété suivante :

$$\left.\begin{array}{r} \forall\, G_1, \ldots, G_n \\ \forall\, (c_1, \ldots, c_n) \in \mathbb{R}^n \end{array}\right\} \sum_i^n \sum_j^n c_i\, k(G_i, G_j)\, c_j \geq 0. \qquad [1]$$

Cette propriété revient à s'assurer que la matrice de Gram K ne possède que des valeurs propres positives.

Pour tout noyau semi défini positif k, la valeur de $k(G, G')$, où G et G' désignent deux graphes, correspond à un produit scalaire entre deux vecteurs $\psi(G)$ et $\psi(G')$, la fonction $\psi(.)$ encodant une représentation des graphes dans l'espace de Hilbert $\mathcal{H}$ associé à k. La distance entre les graphes est alors définie par $d^2(G, G') = \|\psi(G) - \psi(G')\|^2 = k(G, G) + k(G', G') - 2k(G, G')$. L'astuce du noyau (*Kernel Trick*) consiste à utiliser la fonction noyau dans n'importe quel algorithme d'apprentissage automatique pouvant s'exprimer uniquement à l'aide de produits scalaires. Ceci

permet d'utiliser des algorithmes numériques utilisant la métrique induite par le produit scalaire sans avoir à calculer explicitement la fonction de plongement $\psi(.)$ des graphes dans $\mathcal{H}$. Les noyaux sur graphes et l'astuce du noyau fournissent donc une connexion naturelle entre les approches structurelles et statistiques de la reconnaissance de formes.

Une première approche pour définir un noyau sur graphes est basée sur l'utilisation de la distance d'édition entre graphes (Neuhaus *et al.*, 2007). Neuhaus propose d'appliquer un noyau Gaussien sur cette mesure de dissimilarité entre graphes afin de définir un noyau. Ces méthodes permettent d'obtenir une mesure encodant la similarité globale de deux graphes. Cependant, la distance d'édition ne permet pas de garantir la semi défini positivité des noyaux sur graphes. Ces noyaux doivent par conséquent être régularisés (Gaüzère *et al.*, 2012; Smola *et al.*, 2003) afin de définir des noyaux valides. Toutefois, cette régularisation modifie la matrice de Gram afin de la rendre définie semi positive, ce qui altère la métrique correspondant à la distance d'édition.

Une autre importante famille de méthodes à noyaux sur graphes est basée sur la construction d'un sac de sous structures pour chaque graphe, la similarité entre deux graphes étant déduite de la similarité entre leurs sacs. Par exemple, (Kashima *et al.*, 2004) et (Vishwanathan *et al.*, 2010) comparent deux graphes en se basant sur la similarité des marches aléatoires dans les deux structures et (Ralaivola *et al.*, 2005) utilise un ensemble de chemins afin de décrire chaque graphe. Bien que l'utilisation de sous structures linéaires permette de limiter la complexité des algorithmes, elle ne permet de prendre en compte que très partiellement les relations topologiques entre les sommets des graphes. D'autres méthodes (Mahé *et al.*, 2008; Ramon *et al.*, 2003) définissent des noyaux basés sur un ensemble infini d'arbres au lieu de structures linéaires. Ces méthodes corrigent ainsi le manque d'expressivité des structures linéaires et, par conséquent, améliorent la pertinence de la mesure de similarité. Cependant, les noyaux basés sur des sacs de sous structures infinis sont calculés à partir d'une énumération implicite des sous structures. Cette énumération implicite ne permet pas d'analyser l'influence de chaque sous structure pour un problème de prédiction donné. Seule une influence a priori basée sur la taille ou sur le nombre de branchement peut être incluse dans le calcul du noyau (Mahé *et al.*, 2008). Au lieu de décomposer les graphes en un ensemble infini de sous structures, le noyau peut être défini à partir de la distribution d'un ensemble prédéfini de sous structures non linéaires (Shervashidze *et al.*, 2009; Gaüzére *et al.*, 2011). L'énumération explicite des sous structures calculée par ces méthodes permet d'appliquer des méthodes de pondération évaluant l'influence de chaque sous structure pour un problème de prédiction donné.

Les méthodes décrites précédemment ne prennent pas en compte l'information cyclique présente dans les molécules et encodées dans les graphes moléculaires. Néanmoins, les cycles d'une molécules ont une forte influence sur les propriétés physiques et chimiques de ces dernières. Par conséquent, cette information doit être prise en compte lors du calcul de la similarité entre molécules. (Horváth *et al.*, 2004) propose de baser le calcul du noyau sur la similarité de l'ensemble des cycles simples extraits de chaque graphe moléculaire à comparer. L'énumération de

l'ensemble des cycles simples étant complexe à calculer, cette méthode ne peut être utilisée efficacement que lorsque les graphes possèdent un faible nombre de cycles simples. Afin de réduire la complexité de l'énumération de l'ensemble des cycles simples, (Horváth, 2005) propose d'utiliser un sous ensemble de cycles simples. Ce sous ensemble est calculé itérativement à partir de l'ensemble des cycles *pertinents*, tels que définis par (Vismara, 1997). Ensuite, les cycles simples additionnels sont itérativement ajoutés en combinant les cycles pertinents avec les cycles précédemment énumérés. Horváth a montré qu'un faible nombre d'itérations était suffisant pour obtenir une précision de prédiction similaire à celle obtenue en utilisant tous les cycles simples.

Dans cet article, nous proposons dans un premier temps d'étendre la méthode définie dans (Gaüzére *et al.*, 2011) aux sous structures étiquetées afin d'étendre le domaine d'application de la méthode. Dans la Section 2, nous présentons un processus permettant d'obtenir une clé canonique identifiant l'étiquetage de chacune des structures énumérées par (Gaüzére *et al.*, 2011). Cette clé canonique permet de distinguer deux structures ayant un même étiquetage. Dans un second temps, la Section 3 présente l'application du noyau sur treelets étiquetés au graphe des cycles pertinents. Cette seconde extension permet de prendre en compte l'information cyclique des molécules afin d'obtenir une mesure de similarité plus précise. Dans la Section 4, nous proposons trois approches permettant de réduire le nombre de sous structures prises en compte dans le noyau afin d'améliorer la précision de la prédiction. Les noyaux définis sont ensuite comparés à diverses méthodes de l'état de l'art dans la Section 5 sur un problème de classification et un problème de régression. Ces deux expériences mettent en relief les avantages apportés par l'énumération de sous structures non linéaires ou de cycles ainsi que par la pondération de l'ensemble des treelets énumérés.

2. Noyaux de treelets étiquetés

2.1. *Énumération des treelets non étiquetés*

La méthode décrite dans (Gaüzére *et al.*, 2011) permet d'énumérer et compter le nombre d'occurrences d'un ensemble fini de sous structures au sein d'un graphe. Les sous structures énumérées, appelées treelets, sont l'ensemble des arbres non étiquetés ayant un nombre de noeuds inférieur ou égal à 6. La limite sur le nombre de noeuds est un compromis entre l'information représentée par les structures et la complexité nécessaire pour énumérer les treelets inclus dans une molécule. En effet, comme démontré par (Otter, 1948), le nombre de treelets non étiquetés augmente exponentiellement avec le nombre de noeuds. De plus, l'influence d'un atome sur une propriété chimique n'excède pas en général trois liaisons atomiques (Isaacs, 1987). Par conséquent, augmenter la taille des treelets amènerait une contrainte sur les méthodes spécifiques d'énumération de treelets sans apporter une augmentation significative de l'information chimique incluse dans le modèle. L'ensemble des treelets est représenté dans la Figure 1. L'énumération de cet ensemble est une extension de l'énumération de graphlets proposée par (Shervashidze *et al.*, 2009) et est principalement effectuée en deux

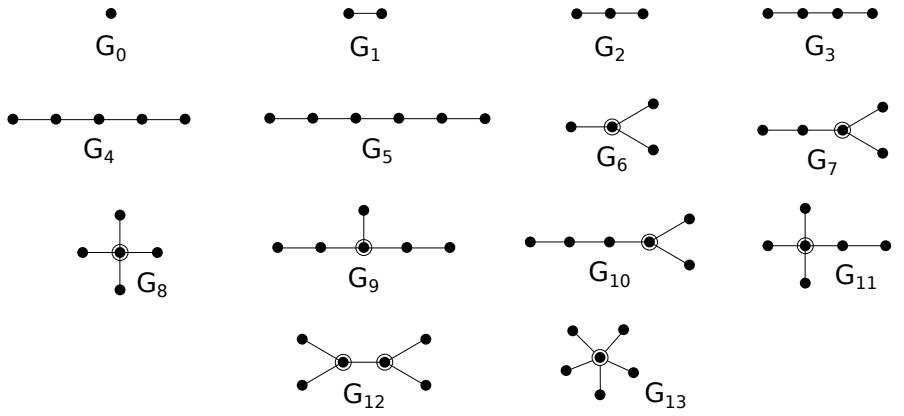


Figure 1. *Ensemble des arbres énumérés par la méthode définie en Section 2.1. Les noeuds entourés représentent les n-étoiles.*

étapes. La première étape consiste à énumérer les treelets linéaires en utilisant une recherche en profondeur à partir de chaque sommet du graphe. Cette première étape permet de calculer le nombre d'occurrences des treelets G_0 à G_5. La seconde étape énumère l'ensemble des motifs non linéaires en analysant le voisinage des treelets spéciaux *3-étoile*, *4-étoile* ou *5-étoile*. Ces treelets correspondent à des arbres composés d'un noeud central de degré 3, 4 ou 5 ainsi que leurs noeuds incidents. Cette première analyse du degré de chaque noeud du graphe permet d'énumérer les treelets G_6, G_8 et G_{13}. Les treelets restants sont énumérés grâce à une analyse du voisinage des noeuds périphériques de chaque *n-étoile*. Une fois cette seconde étape effectuée, un noyau peut être défini dans le cas de graphes non étiquetés en confrontant la fréquence d'apparition de chaque treelet dans les deux graphes à comparer.

2.2. *Énumération des treelets étiquetés*

Toutefois, lors de l'utilisation de cette méthode avec des graphes étiquetés, un treelet correspond à un motif (Section 2.1) associé à un étiquetage. Ainsi, deux sous arbres ayant une structure commune et un étiquetage différent seront associés à deux treelets différents. Afin de différencier ces treelets, il est possible d'utiliser une des méthodes d'étiquetage canonique de molécules (Morgan, 1965; Kuramochi *et al.*, 2004; Faulon *et al.*, 2004). Ces méthodes calculent une représentation canonique d'une molécule encodant à la fois la structure et l'étiquetage de la molécule. Cependant, ces méthodes ne séparent pas clairement l'information structurelle de l'information encodée par l'étiquetage. Le calcul de ces représentations canoniques induit donc un coût de calcul supplémentaire étant donné qu'elles ne permettent pas de réutiliser l'identification structurelle calculée lors de la première étape de l'énumération des treelets (Section 2.1).

Dans cet article, nous proposons d'identifier chaque treelet par un code composé de deux parties : une première partie encodant l'information structurelle et définie par l'index de la structure du treelet (G_0, G_1, etc.) ainsi qu'une seconde partie encodant l'étiquetage du treelet. Cette seconde partie est définie par une clé canonique correspondant à une suite d'étiquettes de noeuds et d'arêtes. Cette séquence est spécifique à chaque structure et est définie de manière à ce que deux treelets aient un même code si et seulement si ils sont isomorphes.

La définition de la clé est triviale pour les structures linéaires, i.e. les chemins. Chaque chemin peut être associé à deux séquences composées alternativement des étiquettes de noeuds et d'arêtes, chacun encodant les deux parcours possibles du chemin. Par convention, la clé associée à une structure linéaire est définie comme la séquence ayant le plus faible ordre lexicographique.

Algorithme 1 Algorithme de calcul de l'étiquetage étendu, tel que défini par Morgan.

Entrées : $G = (V, E)$
Sorties : étiquetage étendu λ
  $\forall v_i \in V, \lambda_i = deg(v_i)$
  $\lambda' = \lambda$
  $k =$ nombre de λ_i différents.
  $k_{+1} = 0$
  while $k > k_{+1}$ **do**
    $k = k_{+1}$
    $\lambda = \lambda'$
    $\forall v_i \in V, \lambda'_i = \sum_{v_j \sim v_i} \lambda_j$
    $k_{+1} =$ nombre de λ'_i différents.
  end while
  return λ

Soit un treelet non linéaire $t = (V, E, \mu_t, \nu_t)$, où μ_t et ν_t correspondent respectivement aux fonctions d'étiquetage des noeuds et des arêtes. La clé canonique définie pour ces structures non linéaires est basée sur le concept de connectivité étendue, tel que défini dans (Morgan, 1965). Ce concept est basé sur une fonction d'étiquetage de noeuds λ de V sur $\mathbb{N}$, appelée étiquetage étendu. Comme décrit dans l'algorithme 1, Cette fonction est définie par un processus itératif qui initialise chaque étiquette $\lambda(v)$ par le degré de v. Chaque noeud est ensuite ré-étiqueté par la somme des étiquettes λ de ses noeuds voisins. Ce processus est itéré tant que le nombre d'étiquettes distinctes augmente. On peut noter que le nombre d'itérations est borné par le nombre de noeuds du graphe puisque le nombre d'étiquettes distinctes possible est majoré par le nombre de noeuds du graphe. L'algorithme n'utilisant que la structure des graphes, l'ensemble des étiquettes obtenues est le même pour deux graphes isomorphes et est unique pour chaque structure d'arbre. La Figure 2 montre l'étiquetage étendu calculé sur les structures non linéaires. Cet étiquetage étendu correspond au degré des noeuds pour tous les treelets sauf pour G_9 et G_{10} où plusieurs itérations sont nécessaires pour différencier un maximum de noeuds.

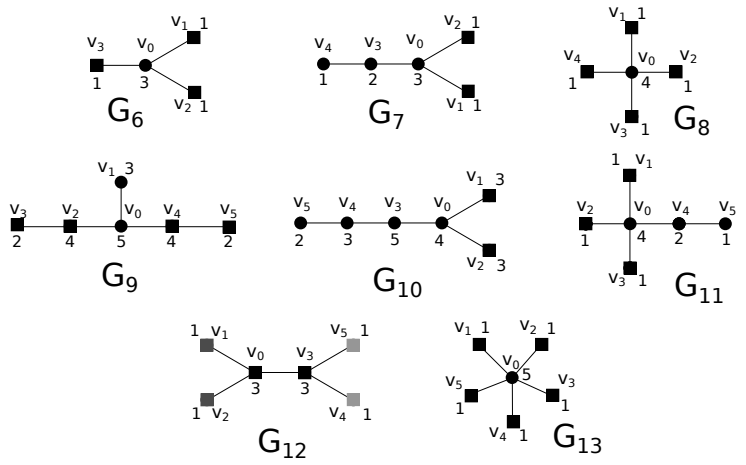


Figure 2. *Structures non linéaires avec l'étiquetage étendu de Morgan. La valeur de l'étiquette étendue est indiquée à proximité de chaque noeud. Les permutations possibles sont représentées par des noeuds en forme de carrés. Les différentes permutations possibles dans une même structure sont représentées par des niveaux de gris différents.*

Puisque deux noeuds adjacents v et v' d'un treelet peuvent être comparés suivant $\lambda(v)$ et $\lambda(v')$, l'étiquetage étendu définit un ordre partiel entre des noeuds adjacents au sein d'un même treelet. Ce tri partiel peut être représenté par un arbre enraciné. Les treelets G_6 à G_{11} ont un étiquetage étendu avec un seul maximum local et les arbres associés sont donc enracinés sur ce noeud (Figure 4(a) et (b)). Le treelet G_{12} possède deux maxima locaux situés sur les noeuds v_0 et v_3. Ce treelet est donc associé à deux arbres enracinés, chacun ayant pour racine v_0 ou v_3 (Figure 4(c) à (e)).

Notre processus de construction de la clé canonique d'un treelet est basé sur un parcours de l'arbre enraciné associé. La conception de la clé nécessite de trier les noeuds enfants de chaque noeud interne de l'arbre afin de définir un parcours unique de l'arbre et donc une clé unique. Cette étape de tri est effectuée par la récursion suivante : La clé de chaque feuille v, dénotée $clé(v)$, est définie par une étiquette vide. Pour chaque noeud v interne de l'arbre, considérons son ensemble de noeuds fils $\{v_1, \ldots, v_n\}$. Cet ensemble est premièrement trié selon $\lambda(v_i)$ et ensuite par la chaîne de caractère définie comme la concaténation de $\nu_t(v, v_i)$, $\mu_t(v_i)$ et $clé(v_i)$. Considérant ce tri sur $\{v_1, \ldots, v_n\}$, la clé associée au noeud v est définie par :

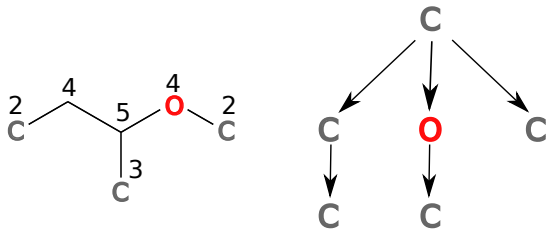
$$clé(v) = \left(\bigodot_{i=1}^{n} \nu_t(v, v_i).\mu_t(v_i) \right) . \bigodot_{i=1}^{n} clé(v_i) \qquad [2]$$

où $\bigodot$ représente l'opérateur de concaténation. En utilisant cette récursion, l'étiquette de chaque noeud est encodée par la clé de son père. Afin de prendre en compte

l'étiquette de la racine, la clé d'un arbre enraciné sur le noeud r est définie comme $\mu_t(r).clé(r)$. Ce processus de calcul de clé pour un treelet est illustré par les Algorithmes 2 et 3.

Les treelets G_6 à G_{11} sont encodés par un seul arbre enraciné, et leurs codes canoniques sont définis comme l'index de la structure, concaténé avec la clé calculée à partir du parcours de l'arbre. Puisque le treelet G_{12} est associé à deux arbres enracinés et donc deux clés, son code canonique est défini comme l'index de la structure (G_{12}) concaténé avec la plus petite clé selon l'ordre lexicographique.

La clé de chaque treelet correspond à la suite d'étiquettes d'arêtes et de sommets rencontrés durant un parcours en profondeur effectué en suivant les index associés à chaque noeud par la méthode de Morgan (Morgan, 1965), comme illustré dans la Figure 3. Toutefois, à la différence de la représentation de Morgan, notre clé n'inclut pas d'information structurelle, qui est encodée par l'index de la structure associée au treelet.



(a) Treelet G_9 et les valeurs de connectivité étendues associées à chaque sommet.

(b) Arbre enraciné associé.

$G_9 - C1C1C1O1C1C$

(c) Code associé.

Figure 3. *Exemple de construction d'une clé à partir d'un treelet G_9 étiqueté.*

2.3. *Clé et isomorphisme de graphe*

Notre clé canonique est basée sur l'étiquetage étendu qui est lui même basé sur la structure du treelet ainsi que sur les fonctions d'étiquetage définies sur les noeuds et les arêtes. Par conséquent, deux treelets isomorphes sont associés à une même clé canonique. À l'inverse, puisque il est possible de construire un treelet linéaire à partir de sa clé canonique, deux treelets linéaires ayant la même clé sont isomorphes. Les treelets correspondant aux structures G_0 à G_5 peuvent donc être uniquement déterminés par leur clé canonique.

Notre processus de construction de la clé triant en priorité les noeuds en fonction de leur clé étendue, l'étiquette d'un noeud ayant une étiquette étendue unique sera située à une position fixe dans la clé associée à ce treelet. L'étiquette d'un tel noeud peut

Algorithme 2 Calcul de clé.

Entrées : Treelet $t = (V, E, \mu, \nu)$, λ
Sorties : Clé(t)

```
r = max_{v∈V} λ(v)
if |r| < 2 then
  clé = μ(r).clé-rec(r)
else
  clé_r1 = μ(r(1)).clé-rec(r(1))
  clé_r2 = μ(r(2)).clé-rec(r(2))
  if clé_r1 > clé_r2 then
    clé = clé_r1.ν(r(1), r(2)).clé_r2
  else
    clé = clé_r2.ν(r(1), r(2)).clé_r1
  end if
end if
return clé
```

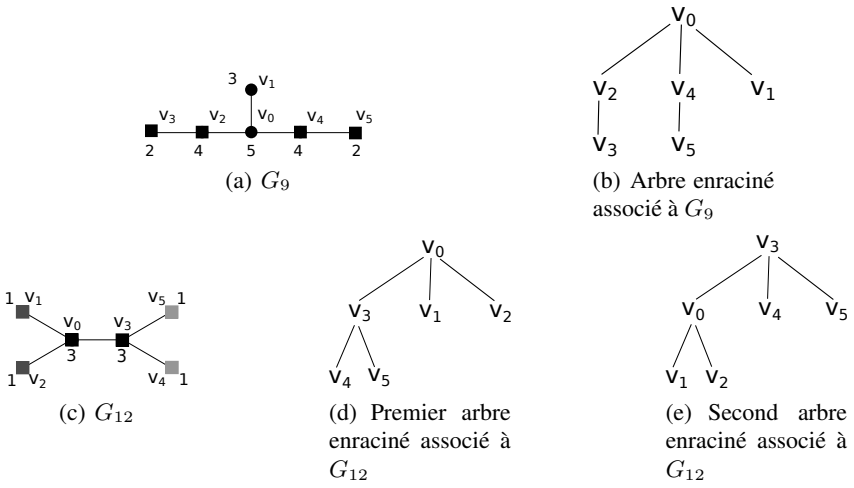
Algorithme 3 clé-rec(v).

Entrées : Treelet $t = (V, E, \mu, \nu)$, λ,$v \in V$
Sorties : Clé(v)

```
for max(λ) → k = 0 do
  for v_i ∈ Γ+(v)|λ(v) == k do
    clé_partielle(v_i) = ν(v, v_i).μ(v_i).clé-rec(v_i)
  end for
  trier(clé_partielle)
  clé = clé.clé_partielle
end for
```

donc être retrouvée sans ambiguïté depuis la clé canonique. Toutefois, un ensemble de noeuds fils $\{v_1, \ldots, v_n\}$ possédant les mêmes étiquettes étendues et ayant le même noeud parent v seront triés selon l'ordre lexicographique de la suite d'étiquettes de noeuds et d'arêtes $\mu_t(v_i)\nu_t(v, v_i)clé(v_i)$. Ce tri permet d'obtenir une clé unique pour deux treelets isomorphes mais ne permet pas de différencier entre les permutations des noeuds $\{v_1, \ldots, v_n\}$. Il est donc nécessaire de vérifier, pour chaque treelet, que les permutations de noeuds autorisées par notre code correspondent à un treelet isomorphe.

Les noeuds ayant une même étiquette étendue dans les structures G_6, G_7, G_8, G_{10}, G_{11} et G_{12} sont représentés dans la Figure 2 par des carrés noirs (■). Pour chaque structure, ces noeuds ayant une même étiquette étendue et de degré un sont les noeuds fils de l'unique noeud auxquels ils sont connectés. Par conséquent, notre clé ne différencie pas les permutations parmi ces noeuds. Puisque ces noeuds ont un degré égal



(a) G_9 (b) Arbre enraciné associé à G_9 (c) G_{12} (d) Premier arbre enraciné associé à G_{12} (e) Second arbre enraciné associé à G_{12}

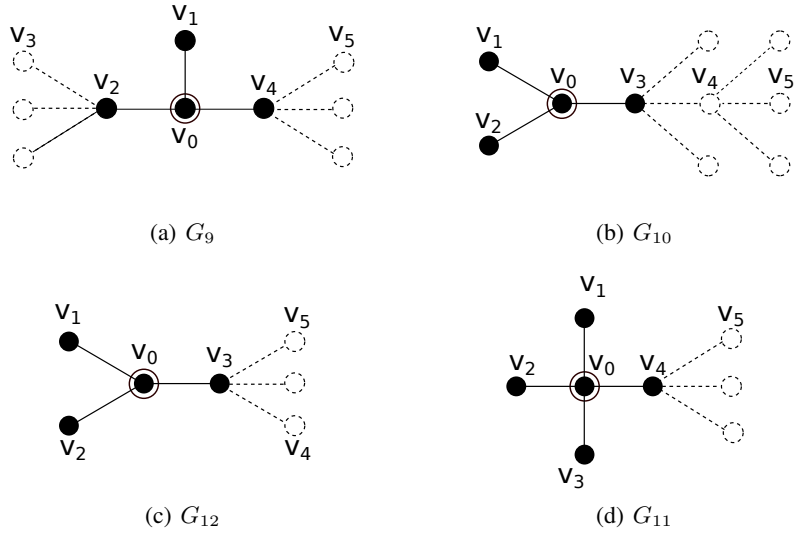
Figure 4. *Exemples de treelets non linéaires avec leur arbre enraciné associé. Les valeurs de λ sont indiquées à coté de chaque sommet sur les figures (a) et (c).*

à un et qu'ils sont connectés à un même noeud, n'importe quelle permutation échangeant deux de ces noeuds conduit à un treelet isomorphe.

L'arbre enraciné associé au treelet G_9 ne permet pas de différencier les deux branches v_2v_3 et v_4v_5 (Figure 4(b)) puisque v_2 et v_4 possèdent les mêmes étiquettes étendues. Toutefois, l'échange simultané de (v_2, v_4) et (v_3, v_5) donne un graphe isomorphe (Figure 4(a)). De la même manière, la clé canonique de G_{12} (Figure 4(c)) ne permet pas de différencier les permutations entre (v_1, v_2), (v_4, v_5) et $(v_0, v_3)(v_1, v_2), (v_4, v_5)$. Toutefois, comme illustré dans la Figure 4(c), ces trois permutations conduisent à des treelets isomorphes. Ainsi, les treelets G_0 à G_{13} peuvent être identifiés par l'association de leur clé canonique et de l'index correspondant à leur structure. Ainsi deux treelets sont isomorphes si et seulement si leurs index identifiant leurs structures ainsi que leurs clés sont égaux.

2.4. *Complexité*

L'énumération des treelets étant restreinte à des arbres, la complexité requise pour énumérer les treelets jusqu'à une taille de 5 (i.e. G_0 à G_8) est bornée par celle requise pour énumérer des graphlets de taille 5 (Shervashidze *et al.*, 2009) : $\mathcal{O}(nd^4)$, où n est égal au nombre de noeuds du graphe et d est égal au degré maximum du graphe. En utilisant une recherche en profondeur, la structure des treelets linéaires peut être énumérée en $\mathcal{O}(nd^5)$. La détection des *p-étoiles* nécessite l'énumération de tous les sous ensembles de p voisins pour chaque noeud du graphe, chaque énumération nécessitant $\mathcal{O}(nd^p)$ opérations, p étant le nombre de noeuds périphériques du treelet étoile.



(a) G_9 (b) G_{10}

(c) G_{12} (d) G_{11}

Figure 5. *Analyse du voisinage nécessaire pour énumérer les treelets ayant 6 sommets et basés sur une* 3-étoile *ou* 4-étoile

L'énumération des treelets G_6,G_8 et G_{13} peut donc être calculée en respectivement $\mathcal{O}(nd^3)$, $\mathcal{O}(nd^4)$ et $\mathcal{O}(nd^5)$ opérations. Les treelets énumérés à partir des *3-étoiles* nécessitent les opérations suivantes :

– G_9 : Pour chaque paire de noeuds périphériques d'une *3-étoile* (e.g. v_2 et v_4 dans la Figure 5(a)), énumérer toutes les paires de sommets (e.g. v_3 et v_5 dans la Figure 5(a)), où chaque noeud appartient au voisinage de l'un des noeuds périphériques sélectionné.

– G_{10} : Pour chaque noeud périphérique, déterminer tous les chemins de longueur 2 (e.g. v_4v_5 dans la Figure 5(b)) commençant sur le noeud périphérique sélectionné.

– G_{12} : Pour chaque noeud périphérique, (e.g. v_3 dans la G_{12}, Figure 5(c)), énumérer toutes les paires de voisins (e.g. v_4 et v_5 dans la Figure 5(c)) du noeud périphérique sélectionné.

L'ensemble de ces opérations nécessite au minimum $\mathcal{O}(d^2)$ opérations. L'ensemble des *3-étoiles* étant énumérés en $\mathcal{O}(nd^3)$, la complexité requise pour énumérer G_9, G_{10} et G_{12} est égale à $\mathcal{O}(nd^5)$ opérations. L'énumération des treelets correspondants au motif G_{11} est effectuée à partir d'une *4-étoile* en sélectionnant un voisin de chaque noeud périphérique (Figure 5(d)). Cette opération étant effectuée en $\mathcal{O}(d)$ et l'énumération des *4-étoiles* en $\mathcal{O}(nd^4)$, l'énumération de l'ensemble des treelets ayant pour motif G_{11} requiert $\mathcal{O}(nd^5)$ opérations. Par conséquent, l'énumération de tous les treelets non étiquetés est effectuée en $\mathcal{O}(nd^5)$. Puisque chaque treelet non étiqueté est fini, la complexité nécessaire pour calculer la clé identifiant l'étiquetage est constante. Par

conséquent, la complexité totale nécessaire pour énumérer l'ensemble des treelets étiquetés reste égale à $\mathcal{O}(nd^5)$. En considérant des graphes ayant un degré borné, la complexité de l'énumération de l'ensemble des treelets est linéaire avec le nombre de noeuds du graphe.

2.5. *Définition du noyau de treelets*

Lorsque tous les treelets d'un graphe G ont été énumérés, un vecteur représentant la distribution des treelets dans G est calculé en se basant sur la clé (Section 2.2). Chaque élément de ce vecteur, dénoté le *spectrum* de G, est égal au nombre d'occurrences d'un treelet dans G :

$$f(G) = (f_t(G))_{t \in \mathcal{T}(G)} \text{ avec } f_t(G) = |(t \trianglelefteq G)| \quad [3]$$

où $\mathcal{T}(G)$ représente l'ensemble des treelets extraits de G et $\trianglelefteq$ l'isomorphisme de sous graphes.

En considérant la fonction f, le noyau entre deux graphes peut être défini comme une somme de noyaux sur l'ensemble des treelets communs aux deux graphes :

$$K_{Treelet}(G, G') = \sum_{t \in \mathcal{T}(G) \cap \mathcal{T}(G')} k\left(f_t(G), f_t(G')\right) \quad [4]$$

où $\mathcal{T}(G) \cap \mathcal{T}(G')$ représente l'ensemble des treelets communs à G et G' et $k\left(f_t(G), f_t(G')\right)$ est un noyau entre les nombres d'occurrences de t dans G et G'.

Afin d'étudier la semi défini positivité du noyau défini dans Équation 4, nous définissons tout d'abord les noyaux de convolution. Les noyaux de convolution d'Haussler (Haussler, 1999) sont définis sur des objets $x \in \mathcal{X}$ décomposables en un ensemble fini $\mathcal{X}_x$. Étant donné un sous noyau $k : \mathcal{X}_x \times \mathcal{X}_x \to \mathbb{R}$, un noyau de convolution d'Haussler $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ est défini comme :

$$K(x, y) = \sum_{(x', y') \in \mathcal{X}_x \times \mathcal{X}_y} k(x', y') \quad [5]$$

Soit une décomposition $\mathcal{X}_G = \{(t, f_t(G)) | t \trianglelefteq G\}$ et un produit tensoriel $(k \otimes k')$ de deux noyaux $k' : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$ et $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, le noyau de treelet (Équation 4) peut être reformulé comme un noyau de convolution :

$$\begin{aligned} K(G, G') &= \sum_{\substack{(t, f_t(G)) \in \mathcal{X}_G \\ (t', f_{t'}(G')) \in \mathcal{X}_{G'}}} (k' \otimes k)(t, f_t(G), t', f_{t'}(G')) \\ K(G, G') &= \sum_{\substack{(t, f_t(G)) \in \mathcal{X}_G \\ (t', f_{t'}(G')) \in \mathcal{X}_{G'}}} k'(t, t') k(f_t(G), f_{t'}(G')) \end{aligned} \quad [6]$$

avec $k(f_t(G), f_{t'}(G'))$ correspondant au noyau entre nombre d'occurrences défini dans l'Équation 4 et $k'(t, t')$ un noyau entre treelets défini tel que $k'(t, t') = 1$ si t et t' sont isomorphes et 0 sinon. Ainsi, le noyau de convolution défini Équation 6 et le noyau de treelets défini Équation 4 sont égaux. Le noyau sur treelets est donc un noyau de convolution d'Haussler. D'après (Haussler, 1999), Si le noyau $k(.,.)$ entre $f_t(G)$ et $f_t(G')$ est un noyau semi défini positif alors le noyau de convolution d'Haussler $K_{Treelet}(G, G')$ est semi défini positif. Lors de l'application du noyau (Section 5), les noyaux Gaussien, linéaire et polynomial, tous semi défini positifs, ont été testé afin de sélectionner le plus adapté à chaque problème.

3. Noyau sur cycles pertinents

Le noyau sur treelets défini dans la Section 2 base la comparaison des molécules sur leurs motifs acycliques communs. Toutefois, les cycles d'une molécule représentent une information structurelle particulièrement importante d'un point de vue chimique. Dans cette section, nous proposons d'appliquer le noyau sur treelets sur un graphe représentant le système cyclique d'une molécule afin de prendre en compte l'information cyclique dans notre noyau sur graphes.

3.1. *Graphe des cycles pertinents*

Un cycle simple est défini comme un sous graphe $C = (V', E', \mu, \nu)$ de G où chaque sommet $v \in V'$ a un degré égal à 2. Chaque cycle $C \subseteq G$ peut être représenté par un vecteur $\vec{C} \in \{0, 1\}^{|E|}$ où $\vec{C}_i$ est égal à 1 si i est une arête de C et 0 sinon. L'ensemble des vecteurs encodant les cycles d'un graphe G définit un espace vectoriel où l'addition de deux cycles C et C' correspond à l'union disjointe de l'ensemble des arêtes composant les deux cycles. Cette opération d'addition correspond alors à un OU exclusif bit à bit (Vismara, 1997) entre les deux représentations vectorielles $\vec{C}$ et $\vec{C'}$ des deux cycles à additionner. L'ensemble des cycles pertinents, $\mathcal{C}_\mathcal{R}$, est défini par l'union des bases de l'espace vectoriel des cycles ayant une taille minimale. La longueur d'une base est définie par la somme des longueurs des cycles la composant. Cette première étape permet de calculer un ensemble canonique de cycles avec une complexité polynomiale par rapport au nombre de noeuds du graphe.

Les relations topologiques entre les cycles pertinents peuvent être encodés par le graphe des cycles pertinents (Vismara, 1995). Ce graphe est défini par $G_\mathcal{C} = (\mathcal{C}_\mathcal{R}, E_{\mathcal{C}_\mathcal{R}}, \mu_{\mathcal{C}_\mathcal{R}}, \nu_{\mathcal{C}_\mathcal{R}})$ où chaque sommet encode un cycle pertinent. Deux sommets sont connectés par une arête si les cycles correspondants à chaque noeud incidents partagent au minimum un sommet dans le graphe initial (Figure 6). Selon (Vismara, 1995), la fonction d'étiquetage $\mu_{\mathcal{C}_\mathcal{R}}(v)$ est définie par le nombre d'arêtes formant le cycle de v tandis que $\nu_{\mathcal{C}_\mathcal{R}}(e)$ est défini comme le couple $(|v(C_1) \cap v(C_2)|, (|e(C_1) \cap e(C_2)|)$ où $v(\mathcal{C}_\mathcal{R})$ et $e(\mathcal{C}_\mathcal{R})$ est respectivement défini par l'ensemble des sommets et l'ensemble des arêtes d'un cycle. Toutefois, ces deux fonctions d'étiquetage encodent seulement la taille des cycles et la taille de leurs connexions. Afin

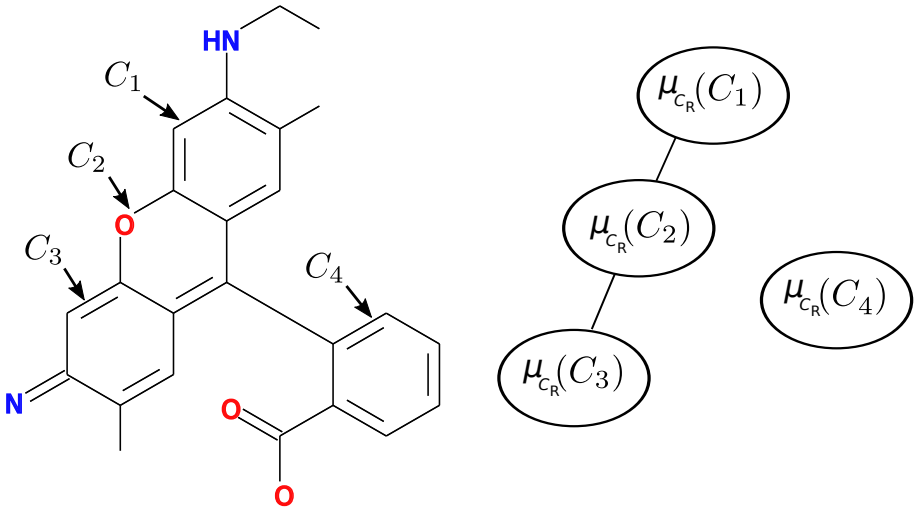


Figure 6. *Un système cyclique et sa représentation sous forme de graphe. La clé canonique $\mu_{C_R}(C_2)$ est égale à C1C1C1O1C1C2 et le code correspondant à l'arête entre C_2 et C_3 est égale à $C1C$.*

d'inclure plus d'informations dans le graphe des cycles pertinents, nous proposons de définir les deux fonctions d'étiquetage comme ceci :

– $\mu_{\mathcal{C}_\mathcal{R}}(C)$: L'étiquette de chaque cycle C est définie par la séquence d'étiquettes d'arêtes et de sommets rencontrées durant le parcours de C. Afin d'obtenir une séquence invariante aux permutations cycliques, on définit $\mu_{\mathcal{C}_\mathcal{R}}(C)$ comme la plus faible séquence selon l'ordre lexicographique.

– $\nu_{\mathcal{C}_\mathcal{R}}(e)$: Une arête e dans C_C encode un chemin entre deux cycles et est décrite par une séquence d'étiquettes d'arêtes et de sommets. Ce chemin pouvant être parcouru depuis ses deux extrémités, on définit $\nu_{\mathcal{C}_\mathcal{R}}(e)$ comme la plus faible séquence d'étiquettes selon l'ordre lexicographique.

Afin d'encoder simultanément la similarité entre les cycles de deux graphes ainsi qu' entre leurs relations topologiques, le noyau sur treelets défini dans la Section 2.5 peut être appliqué sur le graphe des cycles pertinents en utilisant les fonctions d'étiquetage précédemment définies. Ce noyau est alors défini par :

$$K_\mathcal{C}(G, G') = \sum_{t_\mathcal{C} \in \mathcal{T}(G_\mathcal{C}) \cap \mathcal{T}(G'_\mathcal{C})} k(f_{G_\mathcal{C}}(t_\mathcal{C}), f_{G'_\mathcal{C}}(t_\mathcal{C})) \qquad [7]$$

À la différence des noyaux sur cycles existants (Section 1) basés sur la comparaison d'un ensemble de cycles simples, ce noyau encode non seulement la similarité entre les ensembles de cycles pertinents (motif G_0 dans la Figure 1) mais aussi la similarité des relations topologiques entre cycles pertinents (autres treelets dans la Fi-

gure 1). Les étiquettes d'arêtes et de sommets définies permettent d'identifier toutes les combinaisons acycliques de cycles composées d'au maximum 6 cycles. De plus, la méthode d'Horváth nécessite au maximum n^k opérations afin d'effectuer k itérations sur n cycles pertinents. Par comparaison, notre noyau sur treelets nécessite nd^5 opérations dans le pire des cas (Section 2.4), où d est le degré maximal d'un noeud représentant un cycle pertinent. Le noyau sur cycles pertinents possède donc une complexité linéaire avec le nombre de cycles pertinents d'un graphe. Ce noyau peut donc être calculé efficacement lorsque le degré des sommets de G_C est borné.

4. Pondération de treelets

Parmi l'ensemble des treelets trouvés dans un jeu de données, certains d'entre eux n'ont pas d'influence sur l'explication de la propriété recherchée. La prise en compte de ces treelets dans le calcul du noyau entraîne donc des calculs superflus et dégrade le résultat de la prédiction. Le noyau sur treelets peut être reformulé afin d'inclure cette notion d'influence comme ceci :

$$K_{pondere}(G, G') = \sum_{t \in \mathcal{T}(G) \cap \mathcal{T}(G')} w(t) * k\left(f_t(G), f_t(G')\right) \quad [8]$$

où $w : \mathcal{T} \rightarrow \mathbb{R}_+$ est une fonction encodant l'influence d'un treelet dans le calcul du noyau pour un problème de prédiction donné. Ainsi, un $w(t)$ élevé indiquera une forte influence du treelet t sur la propriété à prédire alors qu'un poids égal à 0 indiquera que le treelet n'a pas d'influence et sera donc éliminé du calcul du noyau. La problématique consiste donc à définir la fonction $w(.)$ en fonction de chaque problème de prédiction.

4.1. *Pondération binaire de treelets*

Une première approche consiste à chercher à éliminer les treelets non influents et donc à définir un poids binaire pour chaque treelet. Une méthode triviale serait de tester tous les sous ensembles possibles de treelets mais une telle étude exhaustive implique de tester 2^p ensembles de treelets, avec p représentant le nombre de treelets trouvés dans un jeu de données. Une telle approche est impossible à réaliser, même en considérant un petit nombre de treelets.

Pour définir un ensemble de treelets influents dans un problème de régression, nous proposons d'appliquer deux approches itératives. La première, appelée approche additive (Hocking, 1976), consiste à ajouter à chaque itération un nouveau treelet à l'ensemble des treelets utilisés pour calculer le noyau. Partant d'un ensemble de treelets vide, le treelet ajouté à chaque itération est celui qui donne le meilleur résultat de régression (Alg. 4). La qualité du résultat de la régression est évaluée en calculant la RSS (*Residual Sum of Squares*) définie comme la somme des carrés des erreurs de prédiction commises pour chaque molécule d'un jeu de test. La seconde approche (Hocking, 1976) consiste à partir d'un ensemble composé de tous les treelets

Algorithme 4 Approche additive.

$P = Treelets$
$S = \varnothing$
$nb\_treelets = |P|$
for $i = 0 \rightarrow nb\_treelets$ **do**
  $t = \arg min_t RSS(S \cup \{t\}), t \in P$
  $P = P - \{t\}$
  $S_{i+1} = S_i \cup \{t\}$
end for
return $\arg min_{S_i} RSS(S_i), i \in [0, nb\_treelets]$

Algorithme 5 Approche Soustractive.

$S = Treelets$
$nb\_treelets = |S|$
for $i = 0 \rightarrow nb\_treelets$ **do**
  $t = \arg min_t RSS(S - \{t\}), t \in S$
  $S_{i+1} = S_i - \{t\}$
end for
return $\arg min_{S_i} RSS(S_i), i \in [0, nb\_treelets]$

trouvés et de supprimer un treelet à chaque étape. Cette seconde approche est appelée approche soustractive (Alg. 5). Ces deux méthodes impliquent de tester $\frac{p(p+1)}{2}$ ensembles de treelets.

Une différence importante entre ces deux approches concerne les sous ensembles de treelets associés à une propriété donnée lorsqu'ils sont considérés simultanément. Pour de tels sous ensembles, la suppression d'un treelet par l'approche soustractive entraîne une forte augmentation de la RSS. Ces sous ensembles particuliers sont donc préservés par l'approche soustractive. À l'inverse, ces sous ensembles ont peu de chances d'être sélectionnés par l'approche additive puisque l'ajout de l'un de ces treelets n'améliorera pas de manière significative la RSS.

4.2. *Pondération réelle des Treelets*

La méthode de pondération présentée précédemment ne permet d'obtenir qu'une pondération binaire de chaque treelet, i.e. son inclusion ou non dans le jeu de treelets influents. De plus, cette pondération n'est pas optimale et chaque étape de sélection dépend fortement des étapes précédentes. Afin d'obtenir une pondération réelle et optimale de chaque treelet, nous proposons ici d'adapter une méthode d'apprentissage à noyaux multiples appelée SimpleMKL (Rakotomamonjy *et al.*, 2008) à la sélec-

tion de variables. Le SimpleMKL permet de calculer une pondération optimale d'une combinaison linéaire de noyaux pour un problème de prédiction donné :

$$K_{\text{MKL}}(x, x') = \sum_{i=1}^{N} w_i * k_i(x, x') \quad [9]$$

où $w_i \in \mathbb{R}^+$ et $\sum_i^N w_i = 1$. Afin de déterminer les poids optimaux w_i, la méthode définie par (Rakotomamonjy *et al.*, 2008) consiste à alterner une minimisation d'une fonction objectif $J(w)$ pour un vecteur w fixé avec une descente de gradient dont la direction est définie par la dérivée w par rapport à la fonction objectif $J(w)$. Cette alternance permet de converger vers un vecteur w optimal selon la fonction objectif $J(w)$. Le noyau sur treelets (Équations 4 et 7) est défini comme une somme sur l'intersection de deux ensembles de treelets. Pour un ensemble de treelets $\mathcal{T}$ calculé sur un jeu d'apprentissage, le noyau $k_t(G, G')$ spécifique à un treelet $t \in \mathcal{T}$ est défini comme :

$$k_t(G, G') = \delta(f_t(G))\delta(f_t(G'))k(f_t(G), f_t(G')) \quad [10]$$

avec $\delta(x) = 0$ si $x = 0$, 1 sinon. En reprenant l'Équation 10, les Équations 4 et 7 peuvent se réécrire comme :

$$K_{\mathcal{T}}(G, G') = \sum_{t \in \mathcal{T}} k_t(G, G') \quad [11]$$

où la somme est calculée sur l'ensemble fini $\mathcal{T}$, $\mathcal{T}$ correspondant à l'ensemble des treelets extraits des graphes inclus dans un jeu d'apprentissage. En utilisant cette définition, le noyau sur treelets peut être adapté à la définition du SimpleMKL comme ceci :

$$K_W(G, G') = \sum_{t \in \mathcal{T}} w_t * k_t(G, G') \quad [12]$$

où w_t est le poids optimal calculé pour le treelet t en utilisant le SimpleMKL. La contrainte sur la somme des poids associés à chaque treelet ajoute une contrainte de parcimonie sur la fonction à minimiser afin de sélectionner seulement les treelets les plus influents.

5. Expérimentations

5.1. *Problème de Régression*

Le noyau défini dans cet article a été testé sur un premier jeu de données composé de 185 molécules acycliques (Cherqaoui *et al.*, 1994). Le problème de régression proposé ici consiste à prédire la température d'ébullition des molécules. Les molécules présentes dans ce jeu de données ont la particularité d'être acycliques et composées

Méthode	RMSE ($^\circ C$) leave one out	90/10
(1) Réseaux de Neurones (Cherqaoui *et al.*, 1994)	5.10	NC
(2) KMean (Suard *et al.*, 2002)	12.37	12.24
(3) Random Walks (Vishwanathan *et al.*, 2010)	18.95	18.72
(4) Noyau sur distance d'édition (Neuhaus *et al.*, 2007)	9.04	10.27
(5) Noyau Tree-Pattern (Mahé *et al.*, 2008)	9.50	11.02
(6) Noyau de treelet (TK)	7.80	8.10
(7) TK approche additive	–	7.05
(8) TK approche soustractive	–	6.75
(9) TK avec SimpleMKL	5.14	5.24

Tableau 1. *Prédiction de la température d'ébullition sur des molécules acycliques. RMSE dénote la racine carrée de l'erreur quadratique moyenne.*

	Méthode	Matrice de Gram	Prédiction
(1)	Kmean (Suard *et al.*, 2002)	7.83	0.18
(2)	Random Walks (Vishwanathan *et al.*, 2010)	19.10	0.57
(3)	Noyau sur distance d'édition (Neuhaus *et al.*, 2007)	1.35	0.05
(4)	Noyau Tree-Pattern (Mahé *et al.*, 2008)	4.98	0.03
(5)	Noyau de Treelet (TK)	0.07	0.01
(6)	TK approche additive	0.07	0.01
(7)	TK approche soustractive	0.07	0.01
(8)	TK avec Simple MKL	0.07	0.01

Tableau 2. *Temps d'exécution en secondes mesurés sur le problème de prédiction de température d'ébullition sur des molécules acycliques. Le temps de prétraitement pour les méthodes* (6) *et* (7) *requiert entre* 2 *et* 3 *heures de calcul et* 72 *secondes de calcul pour* (8).

d'hétéroatomes (atomes différents de l'élément chimique Carbone) et sont donc représentées par des graphes acycliques étiquetés. Par conséquent, le noyau sur graphes encodant l'information cyclique des molécules n'a pas été appliqué sur ce jeu de données. Les différentes méthodes à noyaux ont été testées en utilisant deux protocoles différents. Le premier protocole, appelé *leave one out* consiste à effectuer l'apprentissage en utilisant l'ensemble des données sauf une molécule, cette dernière étant prédite par le modèle construit durant la phase d'apprentissage. Le deuxième protocole, noté 90/10 dans le Tableau 1, utilise 90% du jeu de données comme base d'apprentissage afin de prédire les 10% restants. Ce dernier protocole a été répété dix fois afin de prédire la température de chaque molécule du jeu de données et correspond donc à une 10-validation croisée.

La première ligne du Tableau 1 montre le résultat obtenu en utilisant un réseau de neurones basé sur le comptage de 20 sous structures définies a priori par un expert en chimie. Les résultats affichés sont extraits de (Cherqaoui *et al.*, 1994) qui utilise seulement le premier protocole de tests. Les lignes suivantes correspondent à des méthodes utilisant des noyaux sur graphes et la température d'ébullition a été prédite en utilisant une *kernel ridge regression* (Lampert, 2009). Les lignes 2 et 3 du Tableau 1 correspondent à des méthodes basées sur des motifs linéaires. Ces deux méthodes obtiennent les erreurs de prédiction les plus élevées parmi les méthodes testées. En effet, la représentation du graphe moléculaire par des structures linéaires ne permet pas de prendre en compte assez d'information structurelle. La ligne 4 correspond à l'application d'un noyau Gaussien sur une distance d'édition sous optimale entre graphes (Neuhaus *et al.*, 2007). Cette approche obtient de meilleurs résultats que les noyaux précédents grâce à la prise en compte de la similarité globale entre les graphes. Bien que ce noyau ne soit pas défini semi positif pour n'importe quel jeu de graphes, la matrice de Gram calculée sur ce jeu de données est définie semi positive, ce qui permet d'obtenir des résultats satisfaisants. Les méthodes basées sur des motifs non linéaires (Tableau 1, lignes 5 et 6) obtiennent de meilleurs résultats que les noyaux basés sur des motifs linéaires, mais sans atteindre la précision de la méthode basée sur les réseaux de neurones. Ce manque d'efficacité peut être expliqué par le nombre de treelets différents contenus dans le jeu de données. En effet, le noyau de treelets énumère 142 treelets différents dans ce jeu de données et certains d'entre eux apportent peu d'information vis à vis de la propriété à prédire. En conséquence, l'information apportée par ces treelets peut être assimilée à du bruit et dégrade la qualité du résultat. Le résultat de prédiction peut être amélioré en appliquant une des deux méthodes de sélection de treelets décrites dans la Section 4. L'utilisation de ces méthodes (cf. Tableau 1, lignes 6, 7 et 8) permettent de réduire l'erreur de prédiction commise. L'approche additive réduit l'ensemble des treelets utilisés à 26 treelets et l'approche soustractive réduit cet ensemble à 56 treelets. L'approche soustractive obtient de meilleurs résultats grâce au fait qu'elle prend mieux en compte l'information apportée par certaines combinaisons de treelets (Section 4). Les résultats obtenus en *leave one out* n'ont pas été calculés pour les méthodes de sélection binaire. Ceci est du au fait de la complexité requise pour calculer l'ensemble de treelets influents pour chacun des ensembles d'apprentissage induit par un protocole *leave one out* est trop élévée pour être effectuée dans un temps raisonnable. La pondération des treelets via le SimpleMKL (Rakotomamonjy *et al.*, 2008) obtient les meilleurs résultats sur ce jeu de données, Tableau 1 Ligne 9. Cette différence de précision peut s'expliquer par la qualité de la pondération obtenue par le SimpleMKL, pondération plus fine qu'une simple sélection, et par le fait que la pondération calculée par le SimpleMKL est optimale pour le problème de prédiction donné sur la base d'apprentissage. Une différence notable entre notre approche et celle proposée par Pierre Mahé (Mahé *et al.*, 2008) réside dans le fait que l'on calcule explicitement la distribution de chaque treelet dans une molécule. Cette énumération explicite permet de pondérer chaque treelet indépendamment à la différence de la méthode décrite dans (Mahé *et al.*, 2008) qui permet seulement de pondérer chaque sous structure en fonction de sa profondeur ou du nombre de branchements.

Méthode	Précision de la classification
(1) KMean (Suard *et al.*, 2002)	80% (55/68)
(2) KWMean (Dupé *et al.*, 2009)	88% (60/68)
(3) Random Walks (Vishwanathan *et al.*, 2010)	82% (56/68)
(4) Noyau sur distance d'édition (Neuhaus *et al.*, 2007)	90% (61/68)
(5) Noyau Tree-Pattern (Mahé *et al.*, 2008)	96% (65/68)
(6) Noyau de treelets	91% (62/68)
(7) Noyau de treelets avec SimpleMKL	94% (64/68)

Tableau 3. *Comparaison de différents noyaux sur graphes, combinés avec des SVM, sur un problème de classification.*

La première colonne du Tableau 2 présente les temps d'exécution nécessaires pour calculer les matrices de Gram associées aux différents noyaux testés dans cette section (cf. Tableau 1). La seconde colonne du Tableau 2 présente le temps moyen nécessaire pour prédire la température d'ébullition d'une molécule. Grâce au faible degré des noeuds des graphes moléculaires, l'énumération des treelets peut être effectuée efficacement (Tableau 2, Ligne 7). De plus,l'énumération des treelets est effectuée une et une seule fois pour chaque graphe et seulement une somme de noyaux gaussien est effectuée pour chaque paire de graphes. À l'inverse, le fait d'utiliser une énumération implicite oblige d'effectuer deux énumérations pour chaque paire de graphes. Il est important de noter que la sélection de variables influentes n'influent pas sur le temps requis pour calculer la matrice de Gram. Par contre, le temps nécessaire pour pondérer, de manière binaire ou réelle, les treelets est à prendre en compte dans le temps total d'apprentissage. Ainsi, la sélection additive et la sélection soustractive requièrent approximativement entre 2 et 3 heures de calcul alors que la pondération via MKL requiert approximativement 70 secondes de calcul.

5.2. *Problèmes de Classification*

La première expérience de classification évalue différents noyaux sur graphes sur un problème de prédiction défini sur la monoamine oxidase (MAO)[1]. Ce jeu de données est composé de 68 molécules divisées en deux classes : 38 molécules inhibent la monoamine oxidase (médicament antidépresseur) et 30 ne l'inhibent pas. Ces molécules sont composées de différents éléments chimiques et sont donc représentées par des graphes étiquetés. La classification est effectuée par SVM selon un protocole de leave one out. Le Tableau 3 montre les résultats obtenus par les différentes méthodes à noyaux testées. Les trois premières lignes correspondent à des méthodes

1. Tous les jeux de données dans cette section sont disponibles via la page Internet de l'IAPR-TC15 : https ://brunl01.users.greyc.fr/CHEMISTRY/index.html

basées sur des motifs linéaires. La première ligne (Suard *et al.*, 2002) déduit la similarité entre deux graphes de la similarité moyenne entre chaque paire de chemins extraits des graphes. La deuxième ligne (Dupé *et al.*, 2009) est une extension de la méthode précédente où l'importance des chemins non représentatifs est atténuée. La troisième ligne (Vishwanathan *et al.*, 2010) est basée sur le nombre de marches aléatoires en commun dans deux graphes à comparer. La ligne 4 du Tableau 3 montre le résultat obtenu en appliquant un noyau Gaussien sur l'approximation sous optimale d'une distance d'édition entre graphes. Ce noyau, basé sur une mesure de similarité globale des graphes moléculaires, permet de classifier correctement une grande partie des molécules. Le noyau Tree-Pattern (ligne 5) calcule la similarité entre deux graphes à partir du nombre de *tree-patterns* que les deux graphes ont en commun. Enfin, les deux dernières lignes sont le noyau sur treelets défini dans la Section 2 avec et sans pondération de treelets. Les méthodes de sélection de variable binaire n'ont pas été testées sur les problèmes de classification puisque étant définies pour des problèmes de régression et non de classification. Les noyaux sur graphes utilisant des motifs linéaires ne permettent pas de dépasser les 88% de bonne classification (Tableau 3, lignes 1 à 3). À l'inverse, les méthodes basées sur des motifs non linéaires (Tableau 3, Lignes 5 à 7) permettent d'extraire plus d'information structurelle et obtiennent une meilleure précision de classification statistiquement significative que les méthodes basées sur des motifs non linéaires (Tableau 3, Lignes 1 à 3). Cette expérience permet de mettre en relief l'importance de la prise en compte des motifs non linéaires lors de la comparaison de graphes moléculaires. Les meilleurs résultats sont ceux obtenus par le noyau sur motifs d'arbres (Mahé *et al.*, 2008), Ligne 5. Ces résultats ont été obtenus en prenant en compte des sous arbres complexes et de grande taille, i.e. en réglant les paramètres de profondeur maximale et de pénalisation de manière adéquate.

La seconde expérience de classification est un problème issu du Predictive Toxicity Challenge (Toivonen *et al.*, 2003) qui consiste à prédire le pouvoir cancérigène de composés chimiques appliqués à des souris (M) ou rats (R) mâles (M) ou femelles (F). Cette expérience est basée sur dix jeux de données différents pour chaque animal, chacun d'eux étant composé d'un jeu d'apprentissage d'environ 310 molécules chacun et d'un jeu de test distinct composé d'environ 35 molécules. Les jeux de données contiennent des molécules composées d'hétéroatomes et de cycles et sont donc représentées par des graphes étiquetés. Le Tableau 4 contient la somme du nombre de molécules correctement classifiées pour les dix jeux de tests pour chaque méthode et pour chaque type d'animal. Comme le montre le Tableau 4, Lignes 1 et 2, le noyau défini par (Horváth *et al.*, 2004) obtient un meilleur résultat que le noyau sur treelet sur ce jeu de données, montrant l'importance de l'information cyclique pour cette expérience. Le noyau sur treelets appliqué aux graphes des cycles pertinents obtient des résultats légèrement meilleurs que celui défini par Horvárt (Lignes 2 et 3). Ceci peut être expliqué par la prise en compte des relations topologiques entre cycles pertinents qui n'est pas incluse dans le noyau sur cycles de Horváth. Les Lignes 4 et 5 montrent les résultats obtenus par la sélection de treelets en utilisant le SimpleMKL. Cette pondération parcimonieuse permet de réduire le nombre de treelets énumérés du graphe moléculaire d'environ 3500 à 150 structures influentes et d'environ 350 à 50

	Méthode	MM	FM	MR	FR
1	Noyau sur Treelets (TK)	208	205	209	212
2	Horváth	209	207	202	228
3	TK sur graphe des Cycles Pertinents (TC)	211	210	203	232
4	TK avec Simple MKL	217	224	223	250
5	TC avec Simple MKL	216	213	212	237
6	TK + TC avec Simple MKL	219	**226**	**226**	**251**
7	Noyau Gaussien sur Distance d'Édition	**223**	212	194	234

Tableau 4. *Nombre de molécules correctement classifiées sur Predictive Toxicity Challenge.*

structures influentes pour le graphe de cycles pertinents. Cette sélection de sous structures pertinente permet d'améliorer de manière statistiquement significative (valeur p calculée par un test de Friedman (Friedman, 1937) inférieure à 0.01) la précision de la classification du noyau sur treelets appliqué au graphe moléculaire et au graphe des cycles pertinents. Enfin, la combinaison des deux noyaux (Ligne 6) permet d'améliorer légèrement mais de manière non statistiquement significative les résultats de chaque noyau. Cette distinction de l'information cyclique et acyclique peut expliquer en partie la différence statistiquement significative entre la précision de la classification obtenue par le noyau sur treelets et celle obtenue par un noyau Gaussien appliqué sur la distance d'édition sous optimale (Fankhauser *et al.*, 2011) (Ligne 7).

6. Conclusion

Dans cet article, nous avons proposé un noyau sur graphes basé sur la décomposition des graphes en un ensemble de sous structures appelées treelets. À la différence de la majorité des noyaux existants, ce noyau est basé sur des structures non linéaires, les treelets, qui permettent de prendre en compte une grande partie de l'information structurelle présente dans les graphes. De plus, l'énumération explicite des treelets permet de pondérer chaque treelet de manière indépendante. Les différents moyens de pondération permettent de limiter l'impact des structures non associées avec la propriété à prédire. Les différentes expériences ont montré les avantages apportés par la prise en compte de structures non linéaires ainsi que par le choix d'un sous ensemble de treelets influents. D'autre part, nous avons proposé une application du noyau sur treelets sur une extension du graphe des cycles pertinents. Cette extension permet de prendre en compte l'information cyclique présente dans les molécules dans le calcul de leur similarité. La deuxième expérience montre l'intérêt de la combinaison du noyau sur treelets, du noyau sur cycles et de la pondération des sous structures pour obtenir une prédiction précise.

Les futures extensions de ce noyau viseront à inclure la comparaison entre treelets non isomorphes mais ayant une structure similaire. Cette extension permettra de

prendre en compte la similarité du nombre d'occurrence d'une paire de treelets différents, ce qui apportera un degré de liberté supplémentaire dans la définition du noyau. D'autre part, une réflexion sur l'extension du graphe de cycles pertinents permettrait d'encoder plus d'information cyclique et donc d'améliorer la capture de la similarité cyclique entre molécules. Enfin, la prise en compte de l'information 3D des molécules permettrait l'extension des méthodes présentées aux problèmes incluant de la stéréochimie.

7. Bibliographie

Brun L., Conte D., Foggia P., Vento M., Villemin D., « Symbolic learning vs. Graph Kernels : An Experimental Comparison in a Chemical Application », *Proceedings of the 14th Conference on Advances in Databases and Information Systems (ADBIS 2010)*, p. 31-40, 2010.

Cherqaoui D., Villemin D., Mesbah A., Cense J. M., Kvasnicka V., « Use of a Neural Network to Determine the Normal Boiling Points of Acyclic Ethers, Peroxides, Acetals and their Sulfur Analogues », *J. Chem. Soc. Faraday Trans.*, vol. 90, p. 2015-2019, 1994.

Dupé F.-X., Brun L., « Tree covering within a graph kernel framework for shape classification. », *Proceedings of 15 th International Conference on Image Analysis and Processing (ICIAP 2009)*, p. 278-287, 2009.

Fankhauser S., Riesen K., Bunke H., « Speeding Up Graph Edit Distance Computation through Fast Bipartite Matching », *in* X. Jiang, M. Ferrer, A. Torsello (eds), *Graph-Based Representations in Pattern Recognition*, vol. 6658 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, p. 102-111, 2011.

Faulon J. L., Collins M., Carr R. D., « The signature molecular descriptor. 4. Canonizing molecules using extended valence sequences. », *J. Chem. Inf. Comp. Sc.*, vol. 44, n° 2, p. 427-36, 2004.

Friedman M., « The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance », *Journal of the American Statistical Association*, vol. 32, n° 200, p. 675-701, 1937.

Gaüzére B., Brun L., Villemin D., « Two New Graph Kernels and Applications to Chemoinformatics », *8th IAPR - TC-15 Workshop on Graph-based Representations in Pattern Recognition (GBR'11)*, vol. 6658 of *Lecture Notes in Computer Science*, Springer, p. 112-121, 2011.

Gaüzère B., Brun L., Villemin D., « Two New Graph Kernels and Applications to Chemoinformatics », *Pattern Recognition Letters*, vol. 33, n° 15, p. 1183-1193, 2012.

Haussler D., Convolution Kernels on Discrete Structures, Technical report, Dept. of Computer Science, University of California at Santa Cruz, 1999.

Hocking R., « A Biometrics Invited Paper. The Analysis and Selection of Variables in Linear Regression », *Biometrics*, vol. 32, n° 1, p. 1-49, 1976.

Horváth T., « Cyclic Pattern Kernels Revisited », *in* T. Ho, D. Cheung, H. Liu (eds), *Advances in Knowledge Discovery and Data Mining*, vol. 3518 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, p. 139-140, 2005.

Horváth T., Gartner T., Wrobel S., « Cyclic pattern kernels for predictive graph mining », *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, ACM Press, New York, New York, USA, p. 158, 2004.

Isaacs N., *Physical Organic Chemistry*, Longman Sc. Tech, 1987.

Kashima H., Tsuda K., Inokuchi A., *Kernels for graphs*, MIT Press, chapter 7, p. 155-170, 2004.

Kuramochi M., Karypis G., « An efficient algorithm for discovering frequent subgraphs », *Knowledge and Data Engineering, IEEE Transactions on*, vol. 16, n° 9, p. 1038-1051, 2004.

Lampert C. H., *Kernel Methods in Computer Vision*, Now Publishers Inc., Hanover, MA, USA, 2009.

Mahé P., Vert J.-P., « Graph kernels based on tree patterns for molecules », *Machine Learning*, vol. 75, n° 1, p. 3-35, 2008.

Morgan H. L., « The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. », *J. Chem. Doc.*, vol. 5, n° 2, p. 107-113, 1965.

Neuhaus M., Bunke H., *Bridging the gap between graph edit distance and kernel machines*, World Scientific Pub Co Inc, 2007.

Otter R., « The Number of Trees », *The Annals of Mathematics*, vol. 49, n° 3, p. 583-599, July, 1948.

Poezevara G., Cuissart B., Crémilleux B., « Discovering emerging graph patterns from chemicals », *Proceedings of the 18th International Symposium on Methodologies for Intelligent Systems (ISMIS 2009)*, LNCS, Prague, p. 45-55, 2009.

Rakotomamonjy A., Bach F., Canu S., Grandvalet Y., « SimpleMKL », *J. Mach. Learn. Res.*, vol. 9, p. 2491-2521, 2008.

Ralaivola L., Swamidass S. J., Saigo H., Baldi P., « Graph kernels for chemical informatics. », *Neural networks, special issue on Neural Networks and Kernel Methods for Structured Domains*, vol. 18, n° 8, p. 1093-1110, 2005.

Ramon J., Gärtner T., « Expressivity versus efficiency of graph kernels », *1st Int. Workshop on Mining Graphs, Trees and Sequences*, p. 65-74, 2003.

Shervashidze N., Vishwanathan S. V., Petri T. H., Mehlhorn K., Borgwardt K. M., « Efficient Graphlet Kernels for Large Graph Comparison », *12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, p. 488-495, 2009.

Smola A., Kondor R., « Kernels and Regularization on Graphs », *in* B. Schölkopf, M. Warmuth (eds), *Learning Theory and Kernel Machines*, vol. 2777 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, p. 144-158, 2003.

Suard F., Rakotomamonjy A., Bensrhair A., « Kernel on bag of paths for measuring similarity of shapes », *European Symposium on Artificial Neural Networks*, p. 355-360, 2002.

Todeschini R., Consonni V., *Handbook of Molecular Descriptors*, WILEY-VCH, Weinheim, 2000.

Toivonen H., Srinivasan A., King R., Kramer S., Helma C., « Statistical Evaluation of the Predictive Toxicology Challenge 2000-2001 », *Bioinformatics*, vol. 19, n° 10, p. 1183-1193, 2003.

Vishwanathan S., Borgwardt K. M., Kondor I. R., Schraudolph N. N., « Graph Kernels », *J. Mach. Learn. Res.*, vol. 11, p. 1201-1242, 2010.

Vismara P., Reconnaissance et représentation d'éléments structuraux pour la description d'objets complexes. Application à l'élaboration de stratégies de synthèse en chimie organique, PhD thesis, Université Montpellier II, 1995.

Vismara P., « Union of all the minimum cycle bases of a graph », *Electr. J. Comb*, vol. 4, n° 1, p. 73-87, 1997.