ColorAtom

G. Marcou

January 21, 2014

Abstract

Manuel d'utilisation du logiciel ColorAtom. Ce logiciel utilise un modèle QSAR basé sur les descripteurs ISIDA pour produire une structure chimique où chaque atome porte une contribution atomique de la valeur calculée par le modèle QSAR.

Introduction

Le logiciel ColorAtom produit des structures chimiques où chaque atome est annoté par rapport à sa contribution au résultat d'un modèle QSAR. Ce logiciel fonctionne avec des descripteurs ISIDA. Le principe de fonctionnement est détaillé dans l'article *Mol. Inf.*, **31**, 9, 639-642, 2012.

L'idée est de transposer le concept de calcul différentiel aux modèles QSAR notés $f(\{x\})$. Pour ce faire, chaque descripteur moléculaire x_i parmi l'ensemble $\{x\}$ des descripteurs est décrémenté d'une valeur fixe δ. Le modèle est ré-évalué et la différence dans la prédiction est attribuée au descripteur; ce nombre est interprété comme la dérivée partielle du modèle (équation 1).

$$\frac{\partial f(\{x\})}{\partial x_i} = \frac{f(\{x\}_{\not\ni x_i}, x_i) - f(\{x\}_{\not\ni x_i}, x_i - \delta)}{\delta} \tag{1}$$

Dans le cas de modèles de classification, une perturbation du vecteur de descripteurs d'une molécule n'a d'effet mesurable que si elle induit un changement de classe. Par ailleurs, le sens de la perturbation a une incidence: il est parfois plus facile de provoquer le changement d'une classe selon que la perturbation est positive ou négative. L'effet de la perturbation doit aussi être interprétée différemment: plus il a fallut un perturbation importante de la valeur d'un descripteur pour provoquer la changement de classe, plus ce descripteur doit être considéré comme insensible et moins le poids associé doit être important.

La notion de dérivée partielle doit donc, dans le cas de modèles de classification, notés $\mathfrak{f}$, prendre une valeur arbitraire représentant l'effort demandé pour changer le résultat du modèle. Par exemple, il peut être choisi inverse à la valeur de δ qui a été nécessaire pour changer le résultat du modèle (équation 2). Concrètement, si il avait fallut ajouter ou retrancher la valeur 10, le poids pourrait être choisit égale à 1/10. De cette façon, les poids sont d'autant plus petit (négatifs) que la perturbation aura été importante.

$$\frac{\partial \mathfrak{f}(\{x\})}{\partial x_i} = \frac{1}{\delta_0} \tag{2}$$

$$\delta_0 = \min_{\delta} \left(\{\delta \in \mathbb{N} \mid \mathfrak{f}(\{x\}_{\not\ni x_i}, x_i) \neq \mathfrak{f}(\{x\}_{\not\ni x_i}, x_i \pm \delta)\} \right) \tag{3}$$

A chaque descripteur fragmental est ainsi associé un nombre réel relatif. Plus ce nombre est grand plus l'effet du descripteur sur la valeur prédite par le modèle est important. Le signe indique dans quel sens évolue le valeur prédite.

La molécule est donc décomposée en fragments et si un atome participe à ce fragment, le poids correspondant est ajouté à cet atome. Les atomes les plus souvent impliqués dans des fragments sensibles sont donc particulièrement important. En notant a_n l'atome n d'une molécule, le poids qui lui est associé, pour un modèle quantitatif ou qualitatif respectivement, est:

$$\mathfrak{P}(a_n) = \sum_i \frac{\partial f(\{x\})}{\partial x_i} \delta_{a_n \in x_i} \tag{4}$$

$$\mathfrak{P}(a_n) = \sum_i \frac{\partial \mathfrak{f}(\{x\})}{\partial x_i} \delta_{a_n \in x_i} \tag{5}$$

avec $\delta_{a_n \in x_i} = 1$ si l'atome a_n participe au fragment x_i et $\delta_{a_n \in x_i} = 0$ sinon.

Interface

L'interface du logiciel prend quatre paramètres: le fichier `SDF` de la molécule à colorier, le fichier de description du modèle au format `XML`, un nom de base qui sera utilisé pour nommer les fichiers intermédiaire et le fichier moléculaire final et l'identifiant d'un champs du fichier `SDF` initial. La ligne de commande est la suivante:

```
coloratom -i <input> -o <output> -e <sfield> -m <model>
 -d <delta> -n <levels> --width --height
```

`-i` Fichier moléculaire d'entrée au format SDF

`-o` Nom de base pour les fichiers de sortie. Les fichiers SDF sont des fichiers moléculaires contenant une molécule à laquelle a été adjointe des champs de coloration (champs `ColorAtom` pour les contributions atomiques et champs `ColorAtomLvl` pour la discrétisation de celles-ci, correpondant à un niveau de couleur). Les fichiers SVG sont des images de ces mêmes molécules dans un format vectoriel. Ces fichiers contiennent de fichiers temporaires (.svm, .arff, .pred) qui peuvent être ignorés et qui ont pour but de faciliter le debugging.

`-m` Nom du fichier .XML contenant la définition du modèle.

`-d` Pour les problèmes de classification, l'intensité de la perturbation appliquée au système. Valeur par défaut: 1.

`-n` Le nombre de niveaux de couleurs utilisés. Valeur par défaut: 5

`-e` Nom d'un champs SDF contenant des contributions atomiques pour colorer la structure (par défaut: `AtomColorLvl`)

`--width` Largeur de l'image. Unité arbitraire. Par défaut 500.

`--height` Hauteur de l'image. Unité arbitraire. Par défaut 500.

Le fichier de description du modèle au format XML est une évolution de celui développé pour `ISIDA/Predictor`. Quand plusieurs molécules sont présentes dans un même fichier SDF, chacune générera un couple de fichier SDF et SVG, numérotés par leur ordre d'apparition dans le SDF.

Si l'option `-m` est absente alors la molécule sera colorée en utilisant les contributions atomiques trouvées dans le champs `AtomColorLvl` du fichier SDF. Un autre champs peut-être spécifié à l'aide de l'option `-e`. Le format de ce champs est le suivant:

```
x a1:w1 a2:w2
```

où `x` est le poids de la coloration (il est ignoré), `a1` est un entier désignant un atome et `w1` est un nombre à virgule flotante indiquant la contribution atomique de cet atome. Les contributions pour les différents atomes sont séparés par des espaces.

Format du fichier de description du modèle

L'information est structurés dans un arbre hiérarchique dont les noeuds sont des balises décrivant leur rôle ou leur contenu. Les feuilles de l'arbre peuvent être un texte ou bien une balise sans descendance. Les descendants d'une balise sont contenu entre un élément ouvrant et un élément fermant. Un balise portant le nom *NomBalise* est ouverte avec la commande `<NomBalise>` et fermée avec la commande `</NomBalise>`. Une balise, nommée *BaliseSansDescendance* ne contenant pas de descendance est noté `<BaliseSansDescendance/>`. Si une balise ne contient qu'une balise sans descendance ou un texte composé d'une seule ligne, l'ouverture et la fermeture de la balise doivent être situé sur la même ligne.

Une balise peut être enrichie par des attributs portant une information additionnelle. Ces attributs sont tous indiqués dans la commande ouvrant la balise. Une balise nommée *NomBalise* portant un attribut nommé *NomAttribut* dont la valeur est *Valeur* est indiquée par la commande `<NomBalise NomAttribut=''Valeur''>`.

Le fichier de description du modèle utilise les balises suivantes:

- `Models` balise racine indispensable rassemblant potentiellement tous les modèles à appliquer.
- `Titre` balise contenant un titre pour le modèle décrit par ce fichier. Enfant de `Models`.
- `Author` balise contenant l'information sur l'auteur du modèle. Enfant de `Models`.
- `Comment` balise contenant un texte de commentaires sur le modèle. balise contenant un modèle. Enfant de `Models`.
- `Model` balise décrivant un modèle basé sur un schéma de fragmentation. Plusieurs balises de ce type peuvent se succéder pour gérer un modèle consensus. Enfant de `Models`.
- `ModelType` balise décrivant le type de modèle. Celui-ci est donné par un mot clef étant un balise texte fille. Les mots clefs possibles sont: `MLR`, `ASNN`, `SVM` et `CMD`. Les trois premiers désignent des méthodes supportées en interne par `ISIDA/Predictor` et ne concernent pas ce projet. Le type `CMD` indique que le logiciel attend une ligne de commande pour effectuer des prédicats à partir d'un fichier de descripteurs. Cette balise est accompagnée d'un attribut: `Category`. Les valeurs possibles pour cet attribut sont `CLS`, `CLU` et `REG`, pour des modèles de classification, clustering et régression, respectivement. Ce choix affecte la méthode utiliser pour calculer les dérivées partielle de QSAR. Enfant de `Model`.
- `ModelFiles` balise signalant un nom de fichier. Les logiciels utilisés pour réalisés les prédicats utilisent, en plus du fichier de descripteurs, un ou plusieurs fichiers contenant les paramètres du modèle mathématique employé ou des paramètres, des options du logiciel. Ces fichiers peuvent être énumérés entre les commandes d'ouverture et de fermeture de cette balise. Enfant de `Model`.

- `ModelCmdLines` balise énumérant les lignes de commandes nécessaires à l'exécution du logiciel de prédicat. Actuellement, cette commande exige que le premier paramètre soit le fichier descripteurs, le second le fichier contenant le modèle mathématique et le dernier un fichier de résultat. Le fichier de résultat doit être constitué d'une colonne contenant un prédicat sur chaque ligne. Enfant de `Model`.

- `Fragmentations` balise contenant les descriptions de toutes les fragmentations ISIDA utilisées par le modèle. Un même modèle peut utiliser simultanément plusieurs fragmentations. Enfant de `Model`

- `Fragmentation` balise décrivant un fragmentation ISIDA. Les fragmentations successives sont décrites par une répétition de cette balise. Le type de fragment ISIDA est donné par un mot clef apparaissant comme balise texte fille. Les mots clefs autorisés sont: `IA`, `IB` et `IAB` pour les séquences d'atomes, liaisons et atomes et liaisons; `IIA`, `IIB` et `IIAB` pour les fragments atomes centrés historiques; `IIAe`, `IIBe` et `IIABe` pour les fragments atomes centrés étendus; `IIAx`, `IIBx` et `IIABx` pour les fragments atomes centrés fixes. Des paramètres supplémentaires sont fournis à travers les attributs `Min` et `Max` donnant la longueur minimale et maximale des fragments, `AtomPair` indiquant l'utilisation de paires d'atomes, `StrictFrg` requérant une fragmentation stricte, `UseBenson` indiquant l'usage d'un schéma de Benson, `DynBond` pour des fragments se limitant à ceux contenant des liaisons dynamique et `DynBondA` pour des fragments limités à ceux ne contenant que des liaisons dynamique. Enfant de `Fragmentations`.

En théorie, plusieurs modèles peuvent être combinés. Mais cette possibilité n'est pas encore testée.

Tutoriel

Cette section présente deux exemples d'utilisation du logiciel: dans le cas d'un modèle de régression et dans le cas d'un modèle de classification.

Modèle de régression

Il s'agit ici de coloriser des structures en rapport avec la solubilité acqueuse. Les données et modèles ont été conçus pour les tutoriaux de la troisième école d'été en Chémoinformatique (Chemoinformatics Strasbourg Summer School, CS3) en 2012. Voir: .http://infochim.u-strasbg.fr/spip.php?rubrique147.

Les données sont situées dans le répertoir `LogS`. Ce répertoire contient des fichiers nécessaires à la colorisation d'une structure:

- `testmdlreg.xml` Fichier XML contenant la description du modèle de PLS de solubilité.

- `DoPLS.sh` Script conforme au fonctionnement du logiciel ColorAtom pour appliquer un modèle PLS avec Weka.

- `LogS-PLS` Fichier modèle pour Weka de la solubilité acqueuse avec une PLS.

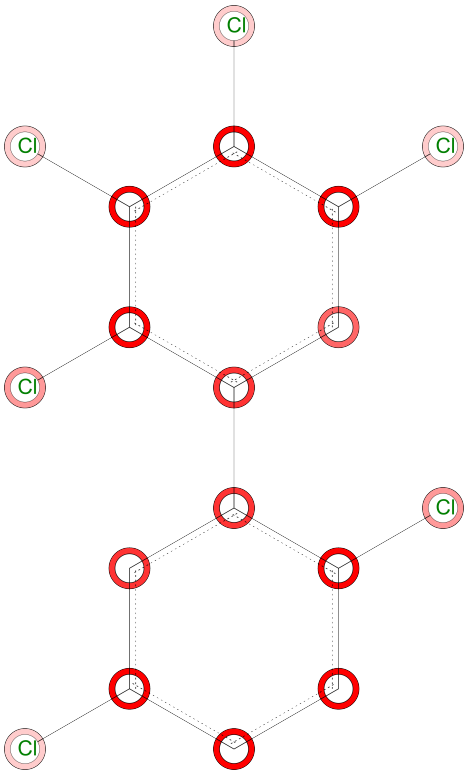

Figure 1: Molécule colorisée par le logiciel ColorAtom selon un modèle PLS de la solubilité sur des descripteurs ISIDA.

- `logs-tIAB12u4.hdr` Fichier liste des descripteurs ISIDA, utilisé pour régénérer les descripteurs selon un schéma prédéfinis.
- `test-logs_109.sdf` Fichier moléculaire d'une molécule au format SDF à colorer.

Des fichiers précisant le contexte du calcul:

- `train-logs.sdf` et `test-logs.sdf` Bases de données utilisées pour générer les descripteurs.
- `train-logs-tIAB12u4.svm` et `test-logs-tIAB12u4.svm` Fichiers de descripteurs utilisés pour entraîner et valider le modèle
- `test-logs_109_1_Colored.sdf` et `test-logs_109_1_Colored.svg` Fichier moléculaire et fichier image de la molécule colorisée.

La ligne de commande suivante:
`coloratom -i test-logs_109.sdf -o test-logs_109 -m testmdlreg.xml`
produira une image telle que l'image 1.

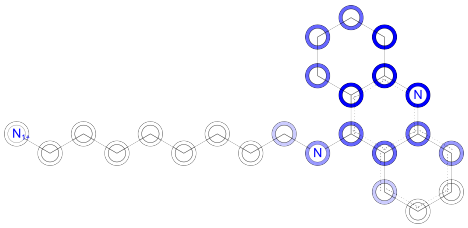

Figure 2: Molécule colorisée par le logiciel ColorAtom selon un modèle Alternate Decision Tree de l'activité Acétyl Choline Estherase avec des descripteurs ISIDA.

0.0.1 Modèle de classification

Il s'agit ici de coloriser des structures en rapport avec l'activité vis-à-vis de l'Acétyl Choline Esthérase (AChE). Les données et modèles ont été conçus pour les tutoriaux de la troisième école d'été en Chémoinformatique (Chemoinformatics Strasbourg Summer School, CS3) en 2012. Voir: .http://infochim.u-strasbg.fr/spip.php?rubrique147.

Les données sont situées dans le répertoire `AChE`. Ce répertoire contient des fichiers nécessaires à la colorisation d'une structure:

- `testclsmdl2.xml` Fichier XML contenant la description du modèle de PLS de solubilité.
- `DoADTree.sh` Script conforme au fonctionnement du logiciel ColorAtom pour appliquer un modèle Alternated Decision Tree avec Weka.
- `ADTree.model` Fichier modèle pour Weka de l'activité sur l'Acétyl Choline Esthérase par un modèle Alternate Decision Tree.
- `train.hdr` Fichier liste des descripteurs ISIDA, utilisé pour régénérer les descripteurs selon un schéma prédéfinis.
- `test-ache_309.sdf` Fichier moléculaire d'une molécule au format SDF à coloriser.

Des fichiers précisant le contexte du calcul:

- `train-ache.sdf` et `test-ache.sdf` Bases de données utilisées pour générer les descripteurs.
- `train-ache_t13l2u3.svm` et `test-ache_t13l2u3.svm` Fichiers de descripteurs utilisés pour entraîner et valider le modèle
- `test-ache_309_1_Colored.sdf` et `test-ache_309_1_Colored.svg` Fichier moléculaire et fichier image de la molécule colorisée.

La ligne de commande suivante:
`coloratom -i test-ache_309.sdf -o test-ache_309`
`-m testclsmdl2.xml -d 10`
produira une image telle que l'image 2.

Conclusion

Le logiciel est dors et déjà capable d'afficher des colorisations réalisées avec des modèles basés sur les descripteurs ISIDA. Il permet également de générer immédiatement des fichiers image au format SVG et des fichiers SDF qui peuvent être traités ultérieurement.

Toutefois, il a été observé qu'il est parfois étonemment difficile de produire une coloration pour des problèmes de classification. Cela est compréhensible pour la classe majoritaire quand est elle est beaucoup peuplée que la classe minoriataire. Mais cela traduit un comportement un peu décevant du logiciel. En effet, il existe des situations où la classification d'un logiciel est instable quand deux descripteurs sont perturbés de concert. Cet effet peut être reproduit par le biais des dérivation de fonctions implicites. Cela se traduit par l'utilisation d'une matrice de corrélation des descripteurs dans le calcul des perturbations à apporter. Une telle matrice pourra être apportée au logiciel à l'aide de la grammaire XML.