gpoezevara@unicaen.fr
gilles.lebrun.caen@gmail.com

1 Contexte et motivations

Une partie des travaux menés au sein de l'équipe COnstraints, Data mining and Graphs (CODAG) du Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen (GREYC) consiste à identifier et extraire des fragments moléculaires influençant la toxicité des molécules. Dans le cadre de ces travaux, le GREYC collabore avec des experts en toxicologie du Centre d'Étude et de Recherche sur le Médicament de Normandie (CERMN). Les collaborateurs du GREYC sont spécialisés dans des problématiques de fouille de graphes : ils représentent une molécule par un graphe (moléculaire) afin d'appliquer leurs méthodes de fouille. Les collaborateurs du CERMN sont eux spécialisés dans des problématiques de toxicologie : l'information extraite des molécules leur permet d'en apprendre plus sur les mécanismes de toxicité. Afin de simplifier la communication des données entre ces deux laboratoires, il serait appréciable de disposer d'une application permettant de visualiser et de manipuler les fichiers échangés. Cette nécessité est l'objet de ce projet : nous vous demandons de créer cette interface graphique en respectant un certain nombre de consignes tout en disposant d'une grande liberté dans son implémentation. Premier conseil : LISEZ LE SUJET EN ENTIER !

2 Cahier des charges

Ce projet a donc pour objectif de vous faire implémenter une application permettant de manipuler des fichiers. Vous devez commentez au maximum votre code source et générer une JAVADOC. De plus, l'application finale devra être exécutée depuis une archive JAR. Pensez également que votre code pourra peut-être être repris par d'autres développeurs : vous devez donc avoir le code le plus générique que possible (par exemple, n'utilisez pas de variables initialisées en dur dans le code mais utilisez plutôt des variables statiques). Nous vous demandons aussi et surtout de respecter l'architecture Modèle, Vue et Contrôleur (MVC) lors de l'implémentation de votre code.

2.1 Le modèle

Le modèle de cette application concerne la lecture, l'édition et l'écriture de fichiers. C'est la partie la plus importante de votre application. Il est possible de distinguer trois types de fichiers, différenciés par leur extension :

sdf : Fichier contenant des descriptions de molécules. Pour chaque molécule, ce fichier donne des informations concernant sa structure atomique et ses propriétés biologiques (par exemple, son poids moléculaire).

gph : Fichier contenant des fragments moléculaires influençant la toxicité des molécules. Ceux-ci sont donnés en format de type graphe. Pour chaque fragment, ce fichier donne des informations diverses sur le fragment (par exemple sa fréquence d'apparition dans les molécules toxiques) ainsi que sa constitution atomique.

bin : Fichier décrivant les molécules contenues dans le fichier sdf avec les fragments moléculaires contenus dans le fichier gph. Pour chaque molécule, ce fichier indique quels fragments moléculaires extraits sont contenus dans la structure de la molécule.

Chaque type de fichier est spécifique et nécessite un traitement approprié. Le fichier de type sdf est fourni par les collaborateurs du CERMN. À partir de ce fichier, les collaborateurs du GREYC fournissent les fichiers de type gph et bin. Ces deux derniers fichiers sont donc le résultat des fouilles réalisées et contiennent les fragments étant responsable de la toxicité des molécules. Le fichier gph recense et décrit les fragments qui influencent la toxicité des molécules stockées dans le fichier sdf, il y a donc un fichier gph différent par fichier sdf. Dans le fichier bin, l'information est un peu

plus complexe. Une molécule est un enchaînement d'atomes et de liaisons atomiques, c'est donc une succession de plusieurs fragments moléculaires. Afin d'indiquer la probabilité qu'une molécule soit toxique, celle-ci est décrite avec les fragments moléculaire extraits. Cela consiste à énumérer pour chaque molécule si celle-ci contient dans sa structure moléculaire un ou plusieurs fragments extraits. Cette information est stockée dans le fichier bin. Un fichier bin est donc en relation directe avec un fichier sdf et un fichier gph précis. Un fichier sdf contient plusieurs molécules, un fichier gph contient plusieurs fragments et un fichier bin contient autant de molécules qu'il y en a dans le fichier sdf et au maximum autant de fragments qu'il y en a dans le fichier bin. Il s'agit en réalité de traiter un triplet de fichiers (sdf,gph,bin) liés par l'information présente dans leur contenu.

2.1.1 Fichier de description des molécules (sdf)

Pour décrire une base de molécules, il est commun d'utiliser un fichier au format sdf. Ce fichier contient de nombreuses informations pour chaque molécule décrite. Voici un exemple commenté d'un tel fichier :

```
  SciTegic06091010432D           // Debut de description d'une molecule

  3  2  0  0  0  0            999 V2000   // La molecule a 3 atomes et 2 liaisons
    1.3000    0.7500    0.0000 C   0  0   // Premier atome contenu dans la molecule
    0.2606    0.1503    0.0000 C   0  0   // Deuxieme atome
    2.3394    0.1503    0.0000 O   0  0   // Troisieme atome
  1  2  2  0                      // Premiere liaison
  1  3  1  0                      // Deuxieme liaison
M  END                           // Fin de la description de la structure
> <HOMO>                         // Balise qui annonce la valeur HOMO
    -9.633412

> <LUMO>                         // Balise qui annonce la valeur LUMO
     1.404117

> <MW>                           // Balise qui annonce le poids moleculaire
44.053

> <CAS>                          // ...
557-75-5

> <SPECIES>                      // ...
rat

> <ROUTE>                        // ...
oral

> <ENDPOINT>                     // ...
LD50

> <DOSE_VALUE>                   // ...
1.4528086878e-003

> <DOSE_UNITS>                   // ...
mol/kg

> <ALogP>                        // ...
9.5e-002

> <num>                          // ...
34

> <logLD50>                      // ...
-2.83779157173422

> <LD50>                         // ...
6.4000581123e-002

> <IndTox>                       // ...
3

> <Toxicity>                     // ...
medium

$$$$                             // Fin de description d'une molecule
```

Dans cet exemple, une seule molécule est décrite mais un fichier sdf contient en réalité plusieurs molécules. Chaque molécule est séparée des autres grâce à l'utilisation de balises de début et de fin de description. On notera que ces balises sont les mêmes pour toutes les molécules. Pour chaque description, il est possible de distinguer deux types d'informations : (i) la structure de la molécule avec le nombre d'atomes et de liaisons chimiques et (ii) des propriétés biologiques de cette molécule comme par exemple son poids moléculaires ou son niveau de toxicité. Nous vous préoccupez pas de la structure de la molécule, seul le nombre d'atomes et le nombre de liaisons chimiques devra être pris en compte. Votre application doit être capable de lire un fichier comme celui-ci et de présenter à l'utilisateur une interface ressemblant par exemple à une feuille Excel :

Numéro	Nombre d'atomes	Nombre de liaisons	HOMO	LUMO	...	IndTox	Toxicity
1	3	2	-9.633412	1.404117	...	3	medium
2	7	5	-3.4567	0.623543	...	1	strong
...	...	...	...	...	...	...	...

Chaque ligne contient une molécule et présente les informations indiquées dans le fichier sdf : chaque information dispose de son propre champ. La première ligne de cet exemple correspond à la description de la molécule dans l'exemple du fichier sdf.

2.1.2 Fichier de fragments moléculaires (gph)

Pour représenter un fragment moléculaire, nous utilisons son graphe moléculaire. Sans rentrer dans la théorie des graphes, nous définissons chaque sommet d'un graphe comme un atome et chaque arrête de ce graphe comme une liaison chimique. Voici un exemple commenté d'un tel fichier :

```
t # 1 & 3 * 9 @ 0 % -1 // Information sur le fragment
v 0 6    // Premier atome
v 1 6    // Deuxieme atome
v 2 8    // Troisieme atome
e 0 1 2  // Premiere liaison
e 0 2 1  // Deuxieme Liaison
```

De la même façon, un seul fragment est décrit dans cet exemple mais un fichier gph contient en réalité plusieurs fragments. Le graphe décrit dans cet exemple correspond à la molécule présentée dans l'exemple de fichier sdf. La description d'un fragment moléculaire (ou graphe) commence par « t ». La première ligne contient de nombreuses informations espacées par des séparateurs différents :
: donne le numéro du graphe.
& : donne la classe du graphe.
* : donne la fréquence du graphe.
@ : donne la valeur de toxicité associée au graphe.
% : donne l'émergence du graphe.
Ensuite, chaque « v » annonce un sommet (ou un atome) et chaque « e » annonce une arrête (ou une liaison chimique). Votre application doit être capable de lire un fichier comme celui-ci et de présenter à l'utilisateur une interface semblable à celle-ci :

Numéro	Nombre d'atomes	Nombre de liaisons	Classe	Fréquence	Toxicité	Emergence
1	3	2	3	9	0	-1
2	12	4	1	34	0.03	22
...	...	...	...	...	...	...

Il suffit d'indiquer pour chaque graphe les différentes informations (fréquence, classe, émergence,) ainsi que le nombre d'atomes et le nombre de liaisons chimiques.

2.1.3 Fichier de molécules décrites par des fragments moléculaires (bin)

Dans ce fichier, nous indiquons pour chaque molécule du fichier sdf, les fragments moléculaires du fichier gph qui compose la structure de cette molécule. Voici un exemple d'un tel fichier :

```
1 6 7 8 9 10 11 12 14 15 40 44 45 57 82 83 84 91 93 94 102 103 108 110 111 112 113
1 57 63 64 84 93 94 95 96 102 103 106 107 108 114 115 126
1 135 136 138 140 154 155 156 157 161 163 165 166 167 168 169
2 40 41 46 47 48 49 145 146 166
2 6 25 27 28 29 30 31 32 33 153 154 155 156 157 158 161 162 163 164 165 166
2 25 27 28  68 69 70 82 83 84 93 94 102 118 119 161 163 166 167 168
2 40 44 45 57 82 83 84 91 93
2 0 16 40 41 46 47 49 53 84 86 92
2 25 33 34 35 36 37 38 39 40 41 42 44 46 47 49 50 52 57 58 60 61 62 84 87 88 89 90
2 0 25 33 35 37 39 40 84 87 88 89 90 91 147 148 149 150 151 152 153 166 167 168 169
2 46 47 48 53 55 84 92
2 25 33 34 35 36 37 38 39 84 87 88 89 90 141 142 143 144 166 167 168 169
2 46 47 48 53 54 55 84 92
2 40 41 43 46 47 49 51 53 84 91 92
2 46 47
2 4 5 16 23 24 40 41 42 44 46 47 49 50 52 57 58 59 82 83 84 85 86 93 117
3 0 4 6 8 10 11 12 13 15 16 23 24 40 41 43 46 47 49 51 53 54 56 84 85 86 91 92 102
3 25 27 28 29 30 31 32 33 34 35 84 87 88 89 135 136 138 145 146 147 148 149 150 151
3 25 33 34 35 37 57 84 87 88 89 102 135 136 138 145 146 147 148 149 150 151 152 154
3 25 26 33 34 35 36 37 38 39 40 41 46 84 87 88 89 102 118 119 121 145 146 147 148
3 25 26 33 34 35 36 37 38 84 87 88 102 135 145 146 147 148 149 150 151 152 154 155
4 6 8 10 15 84 102 107 111
4 40 44 45 46 47 57 84 91 92 93 102 103 104 105 106
4 40 44 45 46 47 53 57 84 91 92 93
4 0 84 102
4 40 41 43 46 57 59 84 91 92 93 102 104 105 106 117
4 40 41 43 46 47 49 51 53 84 91 92
4 16 23
5 6 8 10 11 12 15 40 57 84 91 93 94 102 103 104 106 107 108 109 110 111 112 113
5 0 1 2 3 16 25 26 33 35 37 39 40 44 57 82 145 146 147 148 149 150 151 152 153 166
5 57 84 93 94 102 106 107 110 111 112 113
5 40 41 43 46 47 49 51 53 56 84 91 92 102 104 105
5 40 44 45 57 82 83 84 91 93 102 104 106 107 109 110 111 112 113
5 40 44 45 57 82 83 84 91 93 102 104 106 107 110
5 40 44 57 82 83 84 93 102 106 107 110 111 112 113
6 6 8 12 14 40 44 45 57 82 83 84 91 93 102 103 104 106 107 108 110 111 112 113
6 25 27 28 29 30 31 32 33 40 44 57 82 154 155 156 157 159 160 161 163 166 167 168
6 40 44 45 57 82 83 84 91 93 102 104 106 107 109 110 111 112 113
6 62 84 92 102 105 107 111 113 118 119 120 121 122 123 124 125 166 167 168 169
6 25 33 34 35 36 37 38 40 44 46 57 58 82 83 84 92 93 94 102 105 106 107 110 111 112
6 84 102 107 111 113
6 4 25 26 33 40 41 46 47 49 57 82 83 84 93 94 102 106 107 110 111 112 135 136 145
```

Chaque ligne correspond à une molécule du fichier sdf. La première colonne contient la classe de la molécule (c'est aussi la valeur donnée par le champ « IndTox » dans le fichier sdf). Les colonne suivantes indique le numéro de chaque fragment du fichier gph qui est contenu dans la structure de la molécule. Par exemple, la première molécule est de classe 1 et contient les fragments moléculaires numéro 6,7, ..., 112 et 113. Le numéro d'un fragment moléculaire est donné par la valeur située après « # » dans le fichier gph. Votre application doit être capable de lire un fichier comme celui-ci et de présenter à l'utilisateur une interface ressemblant à celle-ci :

Numéro	Classe	Nombre de fragments
1	1	26
...	...	...
28	4	2
...	...	...

2.1.4 Actions pouvant être menées sur le modèle

Pour chaque type de fichier, votre application devra permettre d'éditer chaque champ de façon à pouvoir modifier la valeur associée. Vous devrez également proposer de changer, ajouter ou enlever un champ de façon à créer un nouveau fichier avec les nouvelles informations : il faudra pour cela repartir de l'ancien fichier et transformer les données pour accomplir la demande de l'utilisateur puisque vous ne stocker pas toutes les informations contenues dans le fichier (notamment la description de la structure moléculaire). Vous devrez également proposer de pouvoir trier les molécules selon un ordre croissant ou décroissant selon la valeur contenue dans un champ

(uniquement numérique). Proposez également de sélectionner les molécules de façon à pouvoir appliquer un changement uniquement à un sous-ensemble de molécules. Pour tester votre application, une archive contenant un exemple de triplet de fichier est disponible à l'url suivante : `http://gpoezevara.free.fr/files/school/triplet_projet.zip` ou sur le moodle de l'IUT.

2.2 La vue

La vue doit proposer une très grande ergonomie dans la gestion de ces fichiers, notamment au niveau de l'édition des différents champs. Pensez que les futurs utilisateurs sont habitués à travailler avec des application commerciales spécialement implémentée pour leur besoins. Par exemple, un panneau à onglet pour passer d'un type de fichier à un autre faciliterait l'utilisation de votre application. A vous de faire preuve d'originalité.

2.3 Le contrôleur

Commencez par implémenter des contrôleurs simples comme des boutons ou des menus. Ensuite, ajoutez des contrôleurs claviers et souris. Un point important à ce niveau est d'offrir une grande modularité dans l'édition des balises utilisées dans les fichiers pour séparer les informations. Prenons l'exemple du fichier sdf, la balise "$$$$" permet de délimiter la fin de description d'une molécule. Cependant il serait très utile de proposer à l'utilisateur de changer cette balise par celle de son choix. Plus votre application offrira de modularité, plus votre code sera générique et plus cela plaira aux correcteurs.

3 Notation

Le cahier des charges présenté ci-dessus constitue la base de votre application. Libre à vous ensuite d'ajouter de nouvelles fonctionnalités. Ce projet représente la moitié de la note qui vous sera attribuée en Algo-Prog, interface graphique SWING. Pour vous notez sur ce projet, nous nous appuierons sur plusieurs indicateurs : (i) l'utilisation de l'application finale, (ii) le code source commenté en JAVADOC, (iii) une soutenance vous permettant de présenter votre projet de façon professionnelle (cahier des charges, problématiques, solutions mise en place, logique d'implémentation, ...) ainsi (iv) qu'un mini rapport sur le processus complet d'implémentation de votre application et (v) un manuel d'utilisation de celle-ci. La meilleure application (si elle est assez bonne) sera régulièrement utilisée au GREYC et au CERMN. Vous devrez réaliser ce projet en binôme ou en trinôme pendant les séances de TD restantes mais aussi sur votre temps libre si nécessaire. Bien évidemment, les correcteurs attendront plus d'un trinôme que d'un binôme. Prenez également conscience que le volume horaire de cours magistraux n'a pas permis de voir toutes les possibilités offertes par la bibliothèque SWING, nous vous conseillons donc fortement d'aller parcourir l'API de JAVA afin de trouver les bons outils à utiliser. Par exemple, certains objets graphiques sont déjà implémentés en MVC et correspondent parfaitement à la vue qui est demandée ici. Afin de respecter l'équilibre dans la notation entre les étudiants, une aide minimum vous sera apportée par les correcteurs. BON COURAGE !