

TP 1 : COMPARAISON D'ARN

INTRODUCTION

Vous allez construire un logiciel qui permet de comparer deux **acides ribonucléiques** (ARN). Votre code va lire des entrées au clavier et ensuite afficher une valeur indiquant la similarité entre les deux ARNs. La prochaine section contient une description du logiciel et suivit d'une section contenant les directives du devoir. La dernière section contient les critères d'évaluation.

DESCRIPTION

Nous allons premièrement décrire les entrées du logiciel. Ce sera suivi d'une description de la métrique à calculer. Finalement, nous décrirons la sortie.

ENTRÉES

LECTURE

Votre logiciel va demander trois (3) informations à l'utilisateur, dans l'ordre indiqué :

1. Une première chaîne de caractères représentant le premier ARN (désigné dans le reste de l'énoncé par *a*). Vous devez vérifier que cette chaîne de caractères ne contienne que les lettres 'A', 'C', 'G' et/ou 'U', en majuscules. Les lettres peuvent être dans n'importe quel ordre et peuvent apparaître 0 ou plusieurs fois. La taille de la chaîne résultante doit donner un nombre divisible par 3. Si la chaîne contient une erreur, alors un message d'erreur significatif est affiché et le programme se termine.
2. Une deuxième chaîne de caractères représentant le deuxième ARN (désigné dans le reste de l'énoncé par *b*). Vous devez vérifier que cette chaîne de caractères ne contienne que les lettres 'A', 'C', 'G' et/ou 'U', en majuscules. Les lettres peuvent être dans n'importe quel ordre et peuvent apparaître 0 ou plusieurs fois. La taille de la chaîne résultante doit donner un nombre divisible par 3. Si la chaîne contient une erreur, alors un message d'erreur significatif est affiché et le programme se termine.
3. Finalement, une valeur entière doit être lue (désigné dans le reste de l'énoncé par *d*). Cette valeur doit être plus grande que 0. Elle indique la distance maximum utilisée pour le calcul de la métrique. S'il y a une erreur, alors un message d'erreur est affiché et le programme se termine.

TECHNIQUE

Vous NE POUVEZ PAS utiliser la classe Clavier. Il est conseillé d'utiliser Scanner. Un petit exemple de code lisant une chaîne de caractère et une valeur est disponible sur Moodle.

TRAITEMENT D'ERREUR

Lorsque le programme rencontre une erreur dans les entrées, il doit afficher un message d'erreur significatif. Utilisez la méthode `System.err.println` pour afficher les messages d'erreur. Après l'affichage du message d'erreur, le programme doit terminer. Utilisez la méthode `System.exit(v)` pour terminer l'application. Il est d'usage d'utiliser une valeur (*v*) négative lorsqu'une application termine avec une erreur.

CALCULE DE LA MÉTRIQUE

Le calcul de la métrique va être décrit dans les sous-sections suivantes. Un ARN est décrit à l'aide de **Nucléotide** (les lettres A, C, G, U). Un **Codon** est une suite de trois nucléotides consécutifs. Un codon peut être interprété comme étant un **acide aminé**. Votre logiciel doit réinterpréter tous les codons en acide aminé avant de faire les calculs. Ensuite, votre logiciel devra appliquer un algorithme qui calculera les différences entre les deux ARNs. Finalement, une petite formule est appliquée sur la valeur résultante pour la rendre compréhensible.

EXTRACTION DES ACIDES AMINÉS

Il suffit de lire les nucléotides trois par trois et ensuite consulter la table suivante qui vous indiquera le nom de l'acide aminé équivalent.

Abréviation	Codon(s)	Nom
Ala	GCU, GCC, GCA, GCG	Alanine
Arg	CGU, CGC, CGA, CGG, AGA, AGG	Arginine
Asn	AAU, AAC	Asparagine
Asp	GAU, GAC	Aspartate
Cys	UGU, UGC	Cystéine
Glu	GAA, GAG	Glutamate
Gln	CAA, CAG	Glutamine
Gly	GGU, GGC, GGA, GGG	Glycine
His	CAU, CAC	Histidine
Ile	AUU, AUC, AUA	Isoleucine
Leu	UUA, UUG, CUU, CUC, CUA, CUG	Leucine
Lys	AAA, AAG	Lysine
Met	AUG	Méthionine
Phe	UUU, UUC	Phénylalanine
Pro	CCU, CCC, CCA, CCG	Proline
Pyl	UAG	Pyrrolysine
Sec	UGA	Sélénocystéine
Ser	UCU, UCC, UCA, UCG, AGU, AGC	Sérine
Thr	ACU, ACC, ACA, ACG	Thréonine
Trp	UGG	Tryptophane
Tyr	UAU, UAC	Tyrosine
Val	GUU, GUC, GUA, GUG	Valine
Fin	UAA	Marqueur

Certains acides aminés sont représentés par plusieurs codons. Aussi, UAA n'est pas un acide aminé, mais peut quand même apparaître. Utilisez **Fin** pour représenter ce codon. Votre transformation doit placer les résultats dans une instance de la classe `ArrayList`. Nous allons désigner les deux résultats comme suit.

La séquence d'acides aminés contenus dans l'ARN a sera désignée par $s = [s_1, s_2, \dots, s_n]$

La séquence d'acides aminés contenus dans l'ARN b sera désignée par $t = [t_1, t_2, \dots, t_m]$

Où les s_i et t_j sont des acides aminés.

ALGORITHME POUR CALCULER LA MÉTRIQUE $M(s, t)$

Nous définissons premièrement la **déviations minimale** de l'acide aminé s_i situé à la position i de l'ARN s par rapport à l'ARN t . Il faut trouver un acide aminé t_j dans l'ARN t qui est pareil à l'acide aminé s_i (donc $s_i = t_j$). La déviation entre les deux est la valeur absolue de la différence de leur indice : $abs(i - j)$. Il faut trouver l'acide aminé t_j qui minimise cette déviation (cela nous donne : $\min(\{abs(i - j) | t_j \in t \wedge t_j = s_i\})$). Si cette déviation minimale est plus grande que le coefficient d (troisième entrée du programme) alors elle est remplacée par la valeur de d . Cela nous donne la déviation minimale de s_i par rapport à t .

Ensuite, il suffit de faire la somme des déviations minimales de tous les acides aminés de s par rapport à t et de faire la somme des déviations minimales de tous les acides aminés de t par rapport à s . Additionnez ces deux sommes pour obtenir la déviation minimale totale entre les deux ARNs.

Finalement nous pondérons cette déviation minimale totale en la divisant par $(n + m)d$.

$$M(s, t) = \frac{\sum_{i=1}^n \min(\min(\{abs(i - j) | t_j \in t \wedge t_j = s_i\}), d) + \sum_{j=1}^m \min(\min(\{abs(i - j) | s_i \in s \wedge t_j = s_i\}), d)}{(n + m)d}$$

PETITE FORMULE

Lorsque vous avez trouvé la déviation minimale totale entre les deux ARNs, il reste à placer ce résultat dans la petite formule suivante :

$$K(s, t) = e^{-6M(s,t)^2}$$

Où $\sigma = 23$ (le nombre d'acides aminés différent). Cette formule transforme la déviation en une métrique qui indique la similarité entre les deux ARNs. Remarque : faites ce calcul avec des **doubles**.

SORTIE

Affichez simplement la valeur double $K(s, t)$ obtenu à la console.

DIRECTIVES

1. Le TP est à faire seul ou en équipe de deux.
2. Code :
 - a. Pas de `goto`, continue.
 - b. Les `break` ne peuvent apparaître que dans les `switch`.
 - c. Un seul `return` par méthode.
 - d. Additionnez le nombre de `if`, `for`, `while`, `switch` et `try` pour chaque méthode. Ce nombre ne doit pas dépasser 7.
3. Indentez votre code. Assurez-vous que l'indentation est faite avec des espaces.
4. Commentaires
 - Commentez l'entête de chaque classe et méthode.
 - Une ligne contient soit un commentaire, soit du code, pas les deux.
 - Utilisez des noms d'identificateur significatif.
 - Une ligne de commentaire ne devrait pas dépasser 80 caractères. Continuez sur la ligne suivante au besoin.

- Nous utilisons Javadoc :
 - La première ligne d'un commentaire doit contenir une description courte (1 phrase) de la méthode ou la classe.
 - Courte.
 - Complète.
 - Commencez la description avec un verbe.
 - Assurez-vous de ne pas simplement répéter le nom de la méthode, donnez plus d'information.
 - Ensuite, au besoin, une description détaillée de la méthode ou classe va suivre.
 - Indépendant du code. Les commentaires d'entêtes décrivent ce que la méthode fait, ils ne décrivent pas comment c'est fait.
 - Si vous avez besoin de mentionner l'objet courant, utilisez le mot 'this'.
 - Ensuite, avant de placer les **tags**, placez une ligne vide.
 - Placez les **tag** @param, @return et @throws au besoin.
 - @param : décrivez les valeurs acceptées pour la méthode.
 - Dans les commentaires, placez les noms de variable et autre ligne de code entre les tags `<code> ... </code>`.
 - Écrivez les commentaires à la troisième personne, EN FRANÇAIS.

REMISE

Remettre le TP par l'entremise de Moodle. Placez vos fichiers `*.java` dans un dossier compressé de Windows, vous devez remettre l'archive. Le TP est à remettre avant le 14 février 23 :55.

ÉVALUATION

- Fonctionnalité (7 pts) : des tests partiels vous seront remis. Un test plus complet sera appliqué à votre TP.
- Structure (2 pt) : veillez à utiliser correctement le mécanisme d'héritage et de méthode.
- Lisibilité (2 pts) : commentaire, indentation et noms d'identificateur significatif.