Master de Bioinformatique — Perl
Universités Bordeaux 1 & 2 — Avril 2005

Perl - Partie 2

Expressions régulières
Analyses simples d'un fichier

1 Expressions régulières en Perl

Résumé.

```
.       1 caractère, n'importe lequel
^       début de ligne
$       fin de ligne
*       0 à n  répétitions du groupe précédent
+       1 à n  ...
?       0 ou 1 ...
{m,n}   m à n  ...   : {m,} m ou plus
                       {m}  m exactement
                       remarque : {0,}=*, {1,}=+, {0,1}=?
[]      groupe
        ^ : excepté
        - : intervalle
|       combinaison, ou
()      groupement logique (ex : (eg|le)gs  => eggs ou legs
\       échappement pour accéder aux caractères spéciaux
```

Abréviations des classes de caractères pré-définies :

```
Construction                        Construction de négation
\d [0-9]        1 chiffre           \D [^0-9]        tout sauf 1 chiffre
\w [a-zA-Z0-9_] caractère de mots   \W [^a-zA-Z0-9_] tout sauf 1 car. mots
\s [ \r\t\n\f]  caractère espace    \S [^ \r\t\n\f]  tout sauf 1 car. espace
```

Exercices.

1. Le script Perl `expr_reg.pl` permet de saisir une phrase au clavier et tester la présence ou non dans cette phrase d'un motif reconnu par une expression régulière donnée. Utilisez ce script pour vérifier chacune des expressions régulières que vous proposerez à la question suivante.
2. Donnez les expressions régulières permettant de reconnaître :
   - Au moins un `a` suivi d'un nombre quelconque de `b`.
   - Un `x` au début de la ligne ou un `y` à n'importe quelle position.
   - Un `a` suivi d'un nombre quelconque de `.`, suivi d'un `b`.
   - Une ligne vide.
   - Que donne la comparaison de `\w+\W+\w+` avec "`this is a test`" ?
   - Quels motifs sont reconnus par `a\Wz`, `a\Sz` et `\s[^ f-h]ood\s` ?

2 Quelques exemples...

Nous avons récupéré sur le site de MIPS la table des ORFs des 2 premiers chromosomes de la levure (vous les trouverez dans les fichiers `orf1.txt` et `orf2.txt`).

1. Essayez de déduire une expression régulière décrivant une entrée de cette table d'ORFs.
2. Exécutez le script `mips1.pl` avec comme argument chacun des deux fichiers de table d'ORFs. Ce script calcule le nombre total d'ORFs ainsi que les valeurs moyenne de leurs longueurs (*length*) et de leurs champs *cai*.
   Identifiez les différentes parties de l'algorithme dans le script Perl.
3. On veut distinguer les ORFs qui sont d'un côté ou de l'autre du centromère et faire des statistiques différentes.
   C'est le script `mips2.pl`
4. On veut afficher à l'écran les ORFs par paquets de longueur <= 5kb.
   C'est le script `mips3.pl`
5. On veut créer automatiquement un rapport en format HTML.
   C'est le script `mips4.pl`

3 Exercice 1

Vous trouverez dans le fichier `seq.txt` une séquence typique de gène soumise au NCBI (National Center for Biotechnology Information) ; et dans le fichier `report.txt` le rapport sur les séquences de la base de données du NCBI qui sont les plus proches[1]. Si vous parcourez ce rapport vers le bas, vous tomberez sur une table qui ressemble à ceci :

Sequences producing	High-scoring Segment Pairs :	High Score	Smallest Sum Probability P(N)	N
gb|AA666429|AA666429	SWOvL3CAN01A08SK Onchocerca volvulu...	2069	4.2e-167	1
gb|AA610978|AA610978	SWOvL3CA335SK Onchocerca volvulus i...	869	2.3e-135	2
gb|AA585557|AA585557	SW3D9CA148SK SAW97MLW-BmL3d9 Brugia...	1660	5.7e-133	1
.				
.				
.				
gb|AA381774|AA381774	EST94892 Activated T-cells I Homo s...	145	0.071	1
gb|AA126166|AA126166	zm78c01.r1 Stratagene neuroepitheli...	143	0.12	1
gb|W26243|W26243	26g6 Human retina cDNA randomly pri...	133	0.60	1

La colonne P(N) donne la probabilité que la séquence soumise “match” celle de la banque de données simplement par chance ; pour la première de la table, cette probabilité est 4.2e-167. Ce sont les très petites probabilités (par exemple inférieures à 1e-50) qui sont significatives. Étant donné qu'il existe maintenant des milliers de telles séquences et de tels rapports, il est très utile d'avoir une méthode automatique pour rechercher les “matchs” significatifs. Écrivez un script Perl qui :

1. lit ce rapport,
2. identifie toutes les séquences qui “match” la séquence soumise avec une probabilité inférieure à 1e-50,
3. extrait les noms de ces séquences (i.e. SWOvL3CAN01A08SK) et les affiche avec leurs probabilités.

Notez que les probabilités sont données sous deux formats ; à la fin de la table, elles sont écrites comme de simples nombres décimaux (comme 0.071). Le programme doit savoir gérer ces deux formats.

[1]Résultats d'alignement de séquences par recherche d'homologie par exemple.

4 Exercice 2

Dans le fichier `report.txt` de l'exercice précédent, vous avez déjà exploité une table dans laquelle la description des gènes, terminée par ..., est tronquée. La description complète des gènes se trouve dans la suite du fichier. Par exemple, pour le premier gène, on trouve :

```
>gb|AA666429|AA666429 SWOvL3CAN01A08SK Onchocerca volvulus infective larva cDNA
            (SAW94WL-OvL3) Onchocerca volvulus cDNA clone SWOvL3CAN01A08 5'
```

Version 1 Écrivez un script Perl, le plus simple possible, qui :

1. lit ce rapport,
2. affiche la description complète de tous les gènes,
3. affiche aussi pour chaque gène la ligne contenant son score et sa probabilité.

On obtient ainsi comme début d'affichage :

```
>gb|AA666429|AA666429 SWOvL3CAN01A08SK Onchocerca volvulus infective larva cDNA
            (SAW94WL-OvL3) Onchocerca volvulus cDNA clone SWOvL3CAN01A08 5'
 Score = 2069 (571.7 bits), Expect = 4.2e-167, P = 4.2e-167

>gb|AA610978|AA610978 SWOvL3CA335SK Onchocerca volvulus infective larva cDNA
            (SAW94WL-OvL3) Onchocerca volvulus cDNA clone SW3ICO335 5'
 Score = 869 (240.1 bits), Expect = 2.3e-135, Sum P(2) = 2.3e-135
...
```

Remarque : parfois, il y a plusieurs lignes Score (et alignements correspondants) pour un même gène. Prenez le premier score qui correspond au meilleur alignement, et donc la probabilité la plus faible.
Conseil : pour faciliter votre lecture du résultat, redirigez l'affichage à l'écran dans un fichier `res` que vous pourrez lire par la suite :

```
> ./exo2v1.pl report.txt > res
> more res
```

Version 2 Modifiez le script précédent pour qu'il n'affiche que les gènes dont la probabilité est inférieure ou égale à 0.60.