Bioinformatique et Perl

Les éléments de base

Introduction

- **Objectif** : être capable de traiter automatiquement de grands volumes de données et d'en extraire l'information pertinente
- **Exemples** :
  - Recherches de motifs dans des séquences biologiques nucléiques ou protéiques
  - Analyser automatiquement le résultat d'une requête BLAST

Pourquoi Perl ?

- **Perl** : Practical Extraction and Report Language
- Langage de script, syntaxe flexible, rapide à maîtriser
- Particulièrement adapté à l'extraction de données (expressions régulières)
- Très présent dans la communauté bioinformatique (nombreux modules)

Infos pratiques

- http://www.perl.com et http://www.perl.org : sites de références

- Livres : plusieurs chez O'Reilly

- http://dept-info.labri.fr/~dutour : accès au site du cours Perl pour le master Bioinfo

1. Un bref aperçu de Perl

- Hello, World! (fichier hello.pl, exécutable)

```
#!/usr/bin/perl –w

# Premier programme

print « Hello, World!\n »;
```

- Exécution : ./hello.pl

Un premier exemple bioinfo

(variables, tableaux, boucles)

motif.pl

- Recherche d'un motif dans une séquence
- Sortie : tableau de toutes les positions

Tableaux

```
@planetes = ("Mercure", "Venus", "Terre",
   "Mars", "Jupiter", "Saturne", "Uranus",
   "Neptune", "Pluton");

print "troisième planète  :  $planetes[2]\n";

print "les planètes  :  @planetes\n";
```

Exercice

- Utiliser **index** plutôt que **substr**

$pos = index($chaine, $souschaine)

retourne la position du 1er caractère de **$souschaine** dans **$chaine**,

retourne -1 si pas trouvé.

Possibilité de rajouter un 3ème paramètre indiquant l'indice dans **$chaine** à partir duquel on cherche **$souschaine**.

2. Autres notions incontournables

- Tables de hachage
- Expressions régulières
- Fichiers

Tables de hachage

```
%trois_vers_un = (
        Ala =>A, Cys=>C, Asp=>D, Glu=>E,
        Phe=>F, Gly=>G, His=>H, Ile=>I,
        Lys=>K, Leu=>L, Met=>M, Asn=>N,
        Pro=>P, Gln=>Q, Arg=>R, Ser=>S,
        Thr=>T, Val=>V, Trp=>W,Tyr=>Y
);
print "Le code pour l'Arginine (Arg) est
    $trois_vers_un{Arg}\n";
```

Recherches de motifs

- C'est une des fonctionnalités les plus importantes de Perl
- En particulier, à l'aide d'expressions régulières

```
$sequence = "CCATTGCatC*TACAC...";
if ($sequence =~ /[^ATGC]/i ) { print "trouvé : $1\n"; }
if ($sequence =~ /CA.C/i ) { print "trouvé\n"; }
```

Parcours d'un fichier

- Exemple de parcours ligne par ligne d'un fichier passé en paramètre du script Perl :

```
while ( <> ) {
   # traitement
   print $_ ;
}
```

- Par défaut, **$_** contient la ligne à chaque lecture.
- On peut nommer la variable explicitement :
  - **while ($ligne = <>) …**

3. Traiter une sortie de BLAST

blast.pl

- C'est un exemple très classique en bioinfo

- Recherches et occurrences de motifs dans les séquences Query et Sbjct
- Attention : le programme ne fait peut-être pas exactement ce qu'on espère…
  (cf résultats attendus : **blast_res.txt**)

Exercices

Jouez le jeu, ne regardez pas les solutions avant…

- Rendre correct le programme précédent
- Explorer plusieurs solutions