Master de Bioinformatique — **Perl**
Universités Bordeaux 1 & 2 — **Mai 2006**

Perl - Partie 8
Introduction au(x) module(s) Bioperl

Résumé. Le projet **Bioperl** est un ensemble de codes Perl pour la bioinformatique qui est développé depuis 1995. Bioperl utilise les aspects programmation orientée objet de Perl.

Le principal intérêt des modules Bioperl est de manipuler des séquences, de permettre l'accès à diverses bases de données de biologie (localement ou sur internet) et de filtrer les sorties de divers programmes.

Documentation : cf les liens indiqués sur la page web du cours Perl.

Installation. Si ce n'est déjà fait, installez sur votre machine le module principal de Bioperl (Core package). Consultez pour cela le site de référence `http://www.bioperl.org`, en particulier la rubrique `Downloads` :
`http://www.bioperl.org/wiki/Getting_BioPerl`
à partir de laquelle vous pouvez télécharger le fichier `current_core_stable.tar.gz` puis accéder aux instructions d'installation sous Unix :
`http://www.bioperl.org/wiki/Installing_Bioperl_for_Unix`.

Exercice1. `Bio::Perl`
Accès simplifié à Bioperl à travers quelques fonctions, sans utiliser la notion d'objet.

1. Consultez et exécutez les scripts `1_roa1.pl` et `1_AI129902.pl`.
2. Consultez la documentation du module `Bio::Perl`, et testez d'autres fonctions.

Exercice2. `Bio::Seq`
Objet séquence, avec ses "features".

1. Consultez et exécutez le script `2_seqex.pl`.
2. Consultez la documentation du module `Bio::Seq`, et testez d'autres fonctions.

Exercice3. `Bio::SeqIO`
Manipulation de fichiers de séquences sous différents formats (fasta, swissprot, embl, genbank, etc...).

1. Ecriture de séquences dans un fichier : consultez et exécutez le script `3_seqout.pl`.
2. Relecture de séquences à partir d'un fichier : consultez et exécutez le script `3_seqin.pl`.

3. Convertir d'un format dans un autre : consultez et exécutez le script d'exemple `3_fasta2genbank.pl`. Puis, écrivez un convertisseur général : écrivez le script `x2y.pl` qui permet l'utilisation :
   `x2y.pl infile infileformat outfile outfileformat`
   Remarque : pour pouvoir créer des "pipeline", on peut utiliser le paramètre `-fh` à la place de `-file` : consultez et exécutez les scripts `all2y.pl` et `all2all.pl`.
4. Consultez la documentation du module `Bio::SeqIO`, et testez d'autres fonctions.

Exercice4. `Bio::DB::...`
Permet de récupérer des séquences depuis les bases de données classiques (SwissProt, GenBank, EMBL, etc...).

1. Consultez et exécutez le script `4_accesSwissProt.pl`.
2. Modifiez le script précédent pour qu'il affiche la séquence récupérée sous le format fasta (`4_accesSwissProt_fasta.pl`).
3. Testez d'autres bases de données et d'autres fonctionnalités (d'autres get, etc).

Exercice5. `Bio::SearchIO`
Permet de "parser" les résultats de recherches dans les bases de données (BLAST, FASTA, etc...)

1. Consultez et exécutez le script `5_first_blast.pl` avec le fichier `report.txt`.
2. Consultez la documentation du module `Bio::SearchIO`, et testez d'autres fonctions.

Exercice6. Autres à voir...

1. `Bio::Tools::Run::RemoteBlast` (plus complet que la fonction `blast_sequence` de `Bio::Perl`).
2. `Bio::Align` et `Bio::AlignIO` manipulations des alignements.