Extraction d'information des bases de séquences biologiques avec R

21 novembre 2006

Résumé

Le module *seqinr* fournit des fonctions pour extraire et manipuler des séquences d'intérêt (nucléotidiques et protéiques) présentes dans les banques en lignes telles que *EMBL*, *GenBank*, *SwissProt* ... Cette fiche présente une méthode d'extraction des séquences codantes de *Gallus gallus* à partir de *GenBank*. Les données sont ensuite utilisées pour tracer le diagramme de distribution des pI dans cette espèce.

Table des matières

1 Connexion à une banque de séquences

Question 1 : Charger le module *seqinr* puis utiliser la fonction `choosebank()` pour obtenir la liste des banques en ligne accessibles à R.

```
library(seqinr)
choosebank()
```

```
 [1] "genbank"      "embl"         "emblwgs"      "swissprot"    "ensembl"
 [6] "refseq"       "nrsub"        "nbrf"         "hobacnucl"    "hobacprot"
[11] "hovernucl"    "hoverprot"    "hogennucl"    "hogenprot"    "hoverclnu"
[16] "hoverclpr"    "homolensprot" "homolensnucl" "greview"      "emglib"
[21] "HAMAPnucl"    "HAMAPprot"    "hoppsigen"    "nurebnucl"    "nurebprot"
[26] "taxobacgen"   "hogendnucl"   "hogendprot"
```

FIG. 1 – Gallus gallus.

Vingt-huit banques sont disponibles à ce jour.

Question 2 : Choisir *GenBank* et charger les paramètres de connexion à cette banque dans l'objet `BD` (*B*anque de *D*onnées). L'objet `BD` ainsi créé est une *liste* dont les éléments sont accessibles individuellement par leur nom (`BD$nom`)[1] ou par leur indice (`BD[i]`)[2].

```
BD <- choosebank("genbank")
BD
```

```
$socket
                description                       class                        mode
"->pbil.univ-lyon1.fr:5558"                    "socket"                       "a+"
                        text                      opened                   can read
                      "text"                    "opened"                      "yes"
                  can write
                       "yes"

$bankname
[1] "genbank"

$totseqs
[1] "67483962"

$totspecs
[1] "409589"

$totkeys
[1] "5900596"

$release
[1] "GenBank Rel. 156 (15 October 2006) Last Updated: Nov 14, 2006"

$status
[1] "on"

$details
[1] "GenBank Rel. 156 (15 October 2006) Last Updated: Nov 14, 2006"
[2] "67,974,843,782 bases; 64,033,602 sequences; 3,450,359 subseqs; 442,815 refers."
[3] "Software by M. Gouy, Lab. Biometrie et Biologie Evolutive, Universite Lyon I"
```

1. `socket` désigne le nom de l'objet contenant les paramètres de connexion à la banque. Dans cet exemple le nom du serveur utilisé pour indexer *GenBank* est *pbil.univ-lyon1.fr*. Il est accessible *via* le port TCP numéro *5558*. Cette banque est accessible non seulement en lecture mais aussi en écriture (possibilité de déposer des séquences).
2. `bankname` désigne le nom de l'objet contenant le nom de la banque interrogée : *genbank*.
3. `totseqs` désigne le nom de l'objet contenant le nombre total de séquences proposées par la banque : 67622895 séquences à ce jour.
4. `totspecs` désigne le nom de l'objet contenant le nombre d'espèces présentes dans la banque : 410732 espèces différentes à ce jour.
5. `release` désigne le nom de l'objet contenant les informations sur la dernière mise à jour de la banque.

2 Interrogation de la banque

Question 3 : Utiliser la fonction `query()` et son aide pour créer une liste nommée `Chicken` contenant les noms des séquences de l'espèce *Gallus gallus* qui sont des séquences codantes et qui ne sont pas partielles. Combien de séquences répondent à ces critères ?

```
query(listname = "Chicken", query = "sp=Gallus gallus et t=cds et no k=partial")
names(Chicken)
```

```
[1] "call"     "name"     "nelem"    "typelist" "req"      "socket"
```

```
Chicken$nelem
```

[1] Voir les annexes A.1 et A.2.
[2] Voir l'annexe A.3.

```
[1] 6165
```

```
 Chicken$req[[1]]
```

```
[1] "A44511.PE1"
attr(,"class")
[1] "SeqAcnucWeb"
attr(,"socket")
                description                    class                     mode
"->pbil.univ-lyon1.fr:5558"                 "socket"                    "a+"
                       text                   opened                 can read
                     "text"                 "opened"                    "yes"
                  can write
                      "yes"
attr(,"length")
[1] "2592"
attr(,"frame")
[1] "0"
attr(,"ncbigc")
[1] "1"
```

Au total, 6165 séquences répondent aux critères. La première séquence de la liste a pour nom `A44511.PE1`.

3 Chargement des séquences et calcul des pI

Question 4 : Récupérer la liste des noms des séquences (dans un vecteur nommé `myProtName`), la liste des séquences elles-mêmes (dans un vecteur nommé `myProtSeq`) et la liste des pI théoriques calculés à partir de ces séquences (dans un vecteur nommé `myProtpI`). Pour cela, utiliser la fonction `sapply()` qui permet d'appliquer une même instruction à tous les éléments d'une liste[3].

```
myProtName <- sapply(Chicken$req, getName)
myProtSeq <- sapply(Chicken$req, getTrans)
myProtpI <- sapply(myProtSeq, computePI)
```

Question 5 : Tracer un histogramme montrant la répartition des pI dans les 6165 séquences ainsi récupérées (fonction `hist()`) :

```
hist(myProtpI, ylim = c(0, 0.3), col = "lightgrey",
        main = "Distribution des pI\n Gallus gallus (6165 sequences)",
        freq = FALSE, las = 1, border = "lightgrey", xlab = "pI", ylab = "Fréquences")
lines(density(myProtpI), col = "red")
```

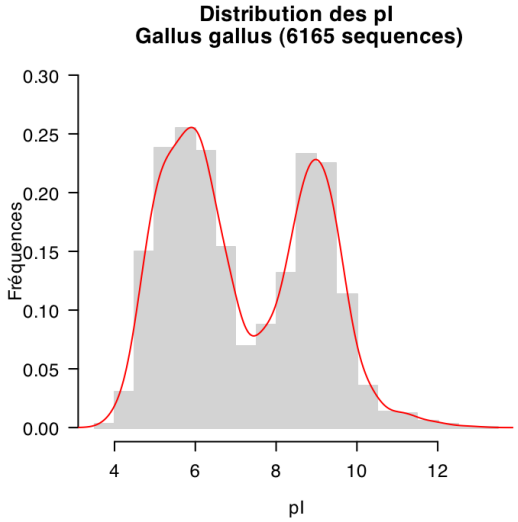

FIG. 2 – Histogramme de distribution des pI théoriques pour 6165 séquences de *Gallus gallus*. Cette distribution présente *a priori* un premier groupe de pI autour de 6 et un second autour de 9. La distribution multi-modale des pI, retrouvée dans les 115 des organismes étudiés, est liée aux propriétés chimiques des acides aminés et non à une pression évolutive qui aurait sélectionné des protéines de pI sensiblement différents du pH intra-cellulaire [5].

[3] Attention, les opérations de chargement des séquences et de calcul des pI peuvent prendre plusieurs dizaines de minutes.

4 Fermeture de la connexion

Utiliser la fonction `closebank()` pour fermer l'accès à la banque en fin de session.

```
closebank()
```

A Données de connexion à la base

A.1 Accès direct par nom

Les objets peuvent être appelés par leur nom, directement dans la liste sur le modèle `nom_liste$nom_objet`. La fonction `names()` donne les noms des objets présents dans la liste :

```
names(BD)
```

```
[1] "socket"   "bankname" "totseqs"  "totspecs" "totkeys"  "release"  "status"
[8] "details"
```

```
BD$socket
```

```
               description                      class                       mode
"->pbil.univ-lyon1.fr:5558"                 "socket"                       "a+"
                       text                   opened                  can read
                     "text"                 "opened"                      "yes"
                  can write
                      "yes"
```

```
BD$bankname
```

```
[1] "genbank"
```

```
BD$totseqs
```

```
[1] "67622895"
```

```
BD$totspecs
```

```
[1] "410732"
```

```
BD$release
```

```
[1] "GenBank Rel. 156 (15 October 2006) Last Updated: Nov 19, 2006"
```

A.2 Fonctions `attach()` et `detach()`

Une variante consiste à utiliser la fonction `attach()` qui permet d'adresser les objets d'une liste uniquement par leur nom. C'est un peu plus simple, en particulier pour programmer dans R ... mais il ne faut pas oublier de lui associer la fonction `detach()`.

```
attach(BD)
```

```
        The following object(s) are masked from BD ( position 3 ) :

         bankname details release socket status totkeys totseqs totspecs
```

```
socket
```

```
               description                      class                       mode
"->pbil.univ-lyon1.fr:5558"                 "socket"                       "a+"
                       text                   opened                  can read
                     "text"                 "opened"                      "yes"
                  can write
                      "yes"
```

```
bankname
```

```
[1] "genbank"
```

```
totseqs
```

```
[1] "67622895"
```

```
 totspecs
```

```
[1] "410732"
```

```
 release
```

```
[1] "GenBank Rel. 156 (15 October 2006) Last Updated: Nov 19, 2006"
```

```
detach(BD)
```

A.3 Accès par indice

Enfin, les objets sont aussi accessibles par leur numéro de rang dans la liste :

```
 BD[1]
```

```
$socket
                description                    class                     mode
"->pbil.univ-lyon1.fr:5558"                 "socket"                     "a+"
                     text                   opened                  can read
                   "text"                 "opened"                     "yes"
                can write
                    "yes"
```

```
 BD[2]
```

```
$bankname
[1] "genbank"
```

```
 BD[3]
```

```
$totseqs
[1] "67622895"
```

```
 BD[4]
```

```
$totspecs
[1] "410732"
```

```
 BD[6]
```

```
$release
[1] "GenBank Rel. 156 (15 October 2006) Last Updated: Nov 19, 2006"
```

Références

[1] D. Charif and J.R. Lobry. Seqinr 1.0-2 : a contributed package to the r project for statistical computing devoted to biological sequences retrieval and analysis. In H.E. Roman U. Bastolla, M. Porto and M. Vendruscolo, editors, *Structural approaches to sequence evolution : Molecules, networks, populations*, Biological and Medical Physics, Biomedical Engineering. Springer Verlag, New York, 2006.

[2] Jambeck P. Gibas C. *Introduction à la bioinformatique*. O'Reilly, 2002.

[3] Lobry J. R. Mélanges et points isoélectriques de protéines. Fiche TD avec le logiciel R : tdr223, 09 2005.

[4] R Development Core Team. *R : A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, isbn 3-900051-07-0 edition, 2006.

[5] Sylvester N. Weiller G. F., Caraux G. The modal distribution of protein isoelectric points reflect amino acid properties rather than sequence evolution. *Proteomics*, 4 :943–949, 2004.