mer. 13 janvier 2010
Guillaume F.

Traitement des données issues du séquençage

Ce que ce document explique :
Ce texte explique comment transformer les données de séquences nucléiques et protéiques trouvées dans les bases de données ou les banques de données, pour les réutiliser en entrée dans les programmes d'analyse des génomes.

Ce qu'il faut maîtriser au préalable : les expressions régulières

Introduction

Les programmes de génomiques permettent l'analyse de séquences et de génomes entier. Ils permettent des comparaisons entre deux séquences ou bien une recherche par similarité d'une séquence connue dans un génome. Ils permettent de construire des phylogénies. Ils permettent également de rechercher des motifs particuliers dans des séquences nucléiques ou protéiques.

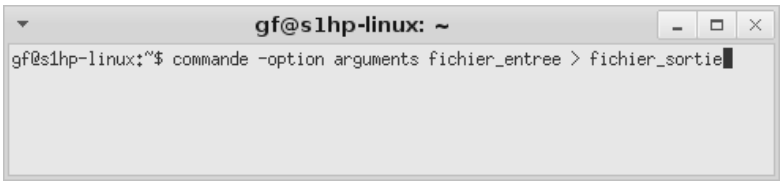
Donner un entrée un fichier de séquence sous-entend une préparation de ce fichier. La majeure partie du temps de travail du génomicien n'est pas occupée par l'analyse en elle-même mais par cette préparation. La modification du format des séquences, l'extraction des données de séquences d'une fiche de séquence disponible sur Internet sont les deux étapes clés de cette préparation.

Quel système d'exploitation adopter ? Travailler en bioinformatique dans le domaine de la génomique demande à manipuler d'énormes fichiers (banques, génomes annotés) ou des milliers de petits fichiers (résultats blast sur un protéome). L'environnement unix ou linux est réellement adapté pour un tel travail. Des lignes de commandes permettent de réaliser facilement une grande partie du travail de modification et de recherche dans les fichiers. Les lignes de commandes Unix permettent également de lancer en chaîne, quasi industriellement, des programmes tels que blast. Il devient possible de travailler efficacement et rapidement sans changer d'interface de travail (la console) depuis le stade de l'échantillonnage des données issues du séquençage jusqu'à celui de l'analyse, en passant par celui de préparation des données. Dans les laboratoires de génomique, unix sera selon toute probabilité déjà installé. Sur un ordinateur personnel, il est possible d'installer ou d'exécuter dans la mémoire vive linux gratuitement.

Table des matières

Rappel de la syntaxe d'une commande unix

Dans le terminal, la syntaxe d'une commande est
commande -option arguments



Exemple :
La commande ls liste le contenu des fichiers, l'option -l affiche en plus les attributs du contenu listé

ls /media -l >/media/usb/liste.txt

permet de lister le contenu (avec attributs) du répertoire intitulé « media » et écrit le résultat suivant dans un fichier texte intitulé « liste.txt » sur un répertoire « usb » :

```
dr-xr-x---  1 root plugdev partition_ntfs
lrwxrwxrwx  1 root root cdrom -> cdrom0
drwxrwxrwx 11 root plugdev documents
drwx------ 13 gf   gf  usb
```

Introduction à la commande sed

sed -n '/signal_de_début_de_l'_extraction_du_texte/,/motif_de_fin/p' fichier_analysé

Sed extraira la ligne qui contient le signal de début de l'extraction du texte, puis toutes les lignes jusqu'à celle (inclue) qui contient le motif d'arrêt. Sed permet donc d'extraire un paragraphe.

sed -n '/jour/,/cinqui/p' fichier_d'entrée > fichier_de_sortie

Sed retourne :

```
Bonjour
Ceci est une ligne du fichier que j'analyse.
Ceci est une ligne supplémentaire.
Chacun son tour ! Encore une autre ligne à la suite.
Cette ligne est la cinquième ligne.
```

sed -n '/jour/,/autre/p' fichier_d'entrée > fichier_de_sortie

```
Bonjour
Ceci est une ligne du fichier que j'analyse.
Ceci est une ligne supplémentaire.
Chacun son tour ! Encore une autre ligne à la suite.
```

sed -n '/^Bo/,/our$/p' fichier_d'entrée > fichier_de_sortie

```
Bonjour
Ceci est une ligne du fichier que j'analyse.
```

Sed renvoit toujours au moins deux lignes. Cela explique l'apparition de la seconde ligne bien que j'ai spécifié que le texte recherché se finissait par *our* en fin de ligne (*$*).

Problème : ma séquence fait partie d'un fichier de banque dans lesquelles se trouvent plusieurs milliers de séquences. Comment n'extraire que la fiche m'intéressant ?

Il faut extraire le paragraphe avec « sed ».

```
sed -n '/début_du_paragraphe_à_extraire/,/fin_du_paragraphe/p' fichier_d'entrée > fichier_de_sortie
```

Cas concret du fichier plat d'une banque banq.embl dans laquelle seule m'intéresse la séquence P15638 :

```
(...)
AC   P15637;
DT   01-APR-1990, integrated into UniProtKB/Swiss-Prot.
DT   01-MAR-1997, sequence version 2.
DT   24-DEC-2008, entry version 45.
SQ   SEQUENCE   477 AA;  53719 MW;  17486555C0E5077C CRC64;
     LLSCQGDSGG PLVCMNDNHM TLLGIISWGV GCGEKDIPGV YTKVTNYLGW IRDNWLQ
//
AC   P15638;
DT   01-APR-1990, integrated into UniProtKB/Swiss-Prot.
DT   01-FEB-1996, sequence version 2.
DT   24-NOV-2009, entry version 82.
SQ   SEQUENCE   477 AA;  53719 MW;  17486555C0E5077C CRC64;
     MVNTMKTKLL CVLLLCGAVF SLPRQETYRQ LARGSRAYGV ACRDEKTQMI YQQQESWLRP
     HDACQGDSGG PLVCMNDNHM TLLGIISWGV GCGEKDIPGV YTKVTNYLGW IRDNMRP
//
AC   P15639;
DT   03-DEC-1992, integrated into UniProtKB/Swiss-Prot.
DT   05-NOV-1998, sequence version 2.
SQ   SEQUENCE   477 AA;  53719 MW;  17486555C0E5077C CRC64;
     MVNTMKTKK CVLLLCGAVF SLPRQETYRQ LARGSRAYGV ACRDEKTQMI YQQQESWLRP
//
(...)
```

j'extrais la fiche avec :

```
sed -n '/^AC   P15638/,/^\/\//p' banq.embl > fiche_P15638.embl
```

^x signifie que le caractère x est à rechercher en début de ligne

\/ protège le caractère / afin qu'il soit interprété comme un texte

\/\/ est compris « // » : c'est le double slash que vous trouverez à la fin de chaque fiche.

p indique qu'il faut conserver/afficher la ligne/texte sélectionnée

Je peux vérifier le contenu de fiche_P15638.embl avec la commande « more » :

```
more fiche_P15638.embl
```

La console retourne :

```
AC   P15638;
DT   01-APR-1990, integrated into UniProtKB/Swiss-Prot.
DT   01-FEB-1996, sequence version 2.
DT   24-NOV-2009, entry version 82.
SQ   SEQUENCE   477 AA;  53719 MW;  17486555C0E5077C CRC64;
     MVNTMKTKLL CVLLLCGAVF SLPRQETYRQ LARGSRAYGV ACRDEKTQMI YQQQESWLRP
     HDACQGDSGG PLVCMNDNHM TLLGIISWGV GCGEKDIPGV YTKVTNYLGW IRDNMRP
//
```

Problème : ma séquence est inclue dans une fiche EMBL et contient des espaces. Comment préparer ma séquence ?

Exemple concret pris sur une fiche **EMBL** :

```
AC   P15638;
DT   01-APR-1990, integrated into UniProtKB/Swiss-Prot.
DT   01-FEB-1996, sequence version 2.
DT   24-NOV-2009, entry version 82.
SQ   SEQUENCE   477 AA;  53719 MW;  17486555C0E5077C CRC64;
     MVNTMKTKLL CVLLLCGAVF SLPRQETYRQ LARGSRAYGV ACRDEKTQMI YQQQESWLRP
     EVRSKRVEHC RCDRGLAQCH TVPVKSCSEL RCFNGGTCWQ AASFSDFVCQ CPKGYTGKQC
     EVDTHATCYK DQGVTYRGTW STSESGAQCI NWNSNLLTRR TYNGRRSDAI TLGLGNHNYC
     RNPDNNSKPW CYVIKASKFI LEFCSVPVCS KATCGLRKYK EPQLHSTGGL FTDITSHPWQ
     AAIFAQNRRS SGERFLCGGI LISSCWVLTA AHCFQERYPP QHLRVVLGRT YRVKPGKEEQ
     TFEVEKCIVH EEFDDDTYNN DIALLQLKSG SPQCAQESDS VRAICLPEAN LQLPDWTECE
     LSGYGKHKSS SPFYSEQLKE GHVRLYPSSR CTSKFLFNKT VTNNMLCAGD TRSGEIYPNV
     HDACQGDSGG PLVCMNDNHM TLLGIISWGV GCGEKDIPGV YTKVTNYLGW IRDNMRP
//
```

Je souhaite soumettre à un logiciel la séquence dépourvue d'espace et nettoyée de ses entêtes.
La commande « sed » permet de chercher-remplacer du texte, d'extraire un paragraphe et d'effacer une ligne.

1/ Je ne conserve que la séquence, même si je ne connais pas la séquence :

```
sed -n '/^SQ/,/^\/\//p' fiche_P15638.embl > P15638.txt
```

Le retour de

```
more sequence_P15638.txt
```

est :

```
SQ   SEQUENCE   477 AA;  53719 MW;  17486555C0E5077C CRC64;
     MVNTMKTKLL CVLLLCGAVF SLPRQETYRQ LARGSRAYGV ACRDEKTQMI YQQQESWLRP
     EVRSKRVEHC RCDRGLAQCH TVPVKSCSEL RCFNGGTCWQ AASFSDFVCQ CPKGYTGKQC
     EVDTHATCYK DQGVTYRGTW STSESGAQCI NWNSNLLTRR TYNGRRSDAI TLGLGNHNYC
     RNPDNNSKPW CYVIKASKFI LEFCSVPVCS KATCGLRKYK EPQLHSTGGL FTDITSHPWQ
     AAIFAQNRRS SGERFLCGGI LISSCWVLTA AHCFQERYPP QHLRVVLGRT YRVKPGKEEQ
     TFEVEKCIVH EEFDDDTYNN DIALLQLKSG SPQCAQESDS VRAICLPEAN LQLPDWTECE
     LSGYGKHKSS SPFYSEQLKE GHVRLYPSSR CTSKFLFNKT VTNNMLCAGD TRSGEIYPNV
     HDACQGDSGG PLVCMNDNHM TLLGIISWGV GCGEKDIPGV YTKVTNYLGW IRDNMRP
//
```

2/ et **3/** Je retire la ligne d'en tête « SQ » et la ligne finale « // » en tapant :

```
sed '/^SQ/d'    P15638.txt > P15638_sans_SQ.txt
sed '/^\/\/ /d'    P15638_sans_SQ.txt > P15638__sans_SQ_sans_slash.txt
```

d ignore/ne conserve pas la ligne sélectionnée.

Visualiser le fichier P15638__sans_SQ_sans_slash.txt :

```
more P15638__sans_SQ_sans_slash.txt
```

Cela retourne :

```
     MVNTMKTKLL CVLLLCGAVF SLPRQETYRQ LARGSRAYGV ACRDEKTQMI YQQQESWLRP
     EVRSKRVEHC RCDRGLAQCH TVPVKSCSEL RCFNGGTCWQ AASFSDFVCQ CPKGYTGKQC
     EVDTHATCYK DQGVTYRGTW STSESGAQCI NWNSNLLTRR TYNGRRSDAI TLGLGNHNYC
     RNPDNNSKPW CYVIKASKFI LEFCSVPVCS KATCGLRKYK EPQLHSTGGL FTDITSHPWQ
     AAIFAQNRRS SGERFLCGGI LISSCWVLTA AHCFQERYPP QHLRVVLGRT YRVKPGKEEQ
     TFEVEKCIVH EEFDDDTYNN DIALLQLKSG SPQCAQESDS VRAICLPEAN LQLPDWTECE
     LSGYGKHKSS SPFYSEQLKE GHVRLYPSSR CTSKFLFNKT VTNNMLCAGD TRSGEIYPNV
     HDACQGDSGG PLVCMNDNHM TLLGIISWGV GCGEKDIPGV YTKVTNYLGW IRDNMRP
```

4/ Il faut maintenant se débarrasser des espaces :

```
sed 's/ //g'    P15638__sans_SQ_sans_slash.txt > sequence_P15638.txt
```

s cherche et remplace. Ici sed remplace un espace par rien.
g indique que cette opération doit s'effectuer à toutes les occurrences trouvées. (Remplacer « g » par un nombre indique à quelle occurrence l'opération doit être effectuée).
On peut accoler deux « sed s » avec -e : sed 's/ //g' -e 's/x/y/g'

En réalité, à la place de réaliser ces quatre commandes, il est préférable de les additionner en une seule avec un tube (|, AltGr 6)ou « pipe » en anglais :

```
sed -n '/^SQ/,/^\/\//p'   fiche_P15638.embl   | sed '/^SQ/d' | sed '/^\/\//d' | sed 's/ //g' > sequence_P15638.txt
```

S'il existe des lignes blanches dans votre fichier, vous pouvez les éliminer en recherchant les lignes comportant dès le début de la ligne (^) le caractère espace présent un nombre indéterminé de fois (*) jusqu'à la fin de la ligne ($) :

```
sed '/^ *$/d'
```

Pour enlever également les retours à la ligne dans une séquence, vous pouvez ajouter ce filtre[1] :

```
sed -r ':a;N;$!ba;s/\n//g'
```

Cette utilisation de sed sort bien évidemment du cadre de ce tutoriel.

1 D'après Pierre Poulain et Patricks Fuchs, Université Paris 7