Travail personnel - exploration d'un transcriptome bactérien

DUBii 2019

Jacques van Helden

04/02/2019

Contents

But du travail personnel

Pour vous approprier les commandes présentées ci-dessus, nous vous proposons d'analyser un fichier d'expression complet et de générer différentes représentations graphiques pour acquérir une intuition de la distribution des données.

Nous vous demandons d'écrire toutes les commandes dans un script R bien organisé (divisé en sections pour les différentes étapes), documenté (en utilisant les commentaires R précédés d'un `#`), et qui pourra être exécuté par quelqu'un d'autre (reproductibilité).

Recommandationsg générales pour le code R

- Organisez votre code en sections séparées par des titres, qui correspondent aux différentes étapes de l'analyse.
- Veillez à commenter dans votre étape chaque étape de l'analyse.
- Utilisez des noms de variables explicites, qui permettent instantanément de comprendre ce qu'elles représentent.
- Séparez soigneusement dans des dossiers distincts les fichiers de code et ceux de données et résultats. Votre code est précieux, il représente votre temps de travail. Les résultats peuvent être effacés sans douleur, puisqu'ils peuvent être régénérés par le code.
- Adoptez une convention de nommage cohérente (voir section suivante).

Style de code

Chaque langage de programmation établit des recommandations concernant le style du code, notamment concernant la façon de nommer les variables et fonctions, l'indentation des blocs de code, l'espacement, ...

Pour des raisons historiques, en R il existe plusieurs conventions alternatives pour nommer les variables et fonctions. Pour plus de détail, voici un très bon article de synthèse:

- Rasmus Bååth (2012). *The State of Naming Conventions in R*. The R Journal (2012) 4:2, pages 74-75. [pdf] [DOI:10.32614/RJ-2012-018]

Pour les travaux personnels, nous recommandons les conventions de Google R Style https://google.github.io/styleguide/Rguide.xml#identifiers, avec cependant une flexibilité: pour les variables, nous suggérons d'utiliser la convention *lowerCamelCase* plutôt que *period.separated*.

- variable.name is preferred, variableName is accepted
- GOOD: `avg.clicks`
- OK: `avgClicks`
- BAD: avg_Clicks

Lisez également attentivement les recommandations d'espacement.

Astuces

- Pour délimiter les sections, vous pouvez utiliser une convention : entourez le titre de quadruples croisillons (par exemple : `#### Data download ####`). Cette convention permet à RStudio d'afficher un menu des sections de votre code, pour vous y déplacer plus facilement.

Téléchargement du jeu de données

1. Connectez-vous à la section "Study cases" de ce module d'enseignement.

   https://du-bii.github.io/study-cases/

2. Cliquez sur le lien Bacterial regulons
3. Avec le bouton droit, cliquez sur le lien du tableau Counts per gene, et copiez ce lien.
4. Connectez-vous au serveur RStudio du cluster core de l'IFB: https://rstudio.cluster.france-bioinformatique.fr/.

   Note: pour les séances de travaux pratiques en salle de cours, nous insistons pour que tout le monde utilise le serveur RStudio du cluster IFB. Cependant, pour le travail personnel, rien ne vous empêche d'utiliser votre propre ordinateur. Vous devrez cependant alors installer vous-mêmes les librairies R requises.

5. Créez un nouveau fichier R (File -> New File -> R script), que vous sauvegarderez sous le nom `bacterial_regulon_analysis.R`.
6. rédigez une section de code intitulée `#### Data download ####` (convention RStudio pour les titres de section dans le code R), qui effectuera les opérations suivantes;
   - créer un dossier local "TP_bacterial_regulons" à la racine de votre compte: (`~/TP_bacterial_regulons`)
   - se déplacer dans ce dossier;
   - y télécharger le fichier `cutadapt_bwa_featureCounts_all.tsv` (celui dont vous avez précédemment copié le lien);
   - lister les fichiers contenus dans le dossier.
7. Exécutez le script et vérifiez le résultat.

Exploration des résultats

Entamez une nouvelle section intitulée "Exploration of the transcriptome table".

1. Avec la fonction `read.delim()`, chargez le tableau de comptages RNA-seq (nombre de reads / gène) dans une variable nommée `rawCounts`.
2. Utilisez la commande `summary()` pour calculer des statistiques de base sur chaque colonne du tableau.
3. Convertissez les comptages par la fonction `log2`.
4. Explorez la distribution des valeurs transformées par log2, en utilisant différentes représentations graphiques vues lors de la séance d'introduction: histogramme, boîte à moustache, ...
5. Calculez pour chaque gène la moyenne des log2(counts) par condition, et dérivez-en les valeurs M et A.
6. Dessinez un nuage des points comparant les valeurs moyennes entre conditions.

7. Dessinez un MA plot.

Rendu

Un script R proprement structuré (sections) et documenté (expliquez ce que vous allez faire à chaque étape, documentez les variables), qui pourra être compris et reproduit par un utilsiateur de R.