Abnahme zur Studienleistung in Data-Mining/Microarrayanalyse mit R

Von Victor Steffens – 199757

Skript-Beschreibung

Gemäß der Aufgabenstellung für die Studienleistung in Data-Mining wurde ein R-Paket fertiggestellt, um Transkriptom-Daten über das R-Paket „Shiny“ zu visualisieren. Hierbei habe ich mich für eine Visualisierung einer Heatmap entschieden.

Das Skript akzeptiert als Eingabe eine Count-Tabelle, wie sie zum Beispiel von den Programmen Feature-Counts aus dem Subreadpackage (http://subread.sourceforge.net/), Salmon(https://combine-lab.github.io/salmon/) oder Kallisto(https://pachterlab.github.io/kallisto/about) erstellt werden. Auf den Spalten werden die Sample-Namen aufgetragen und auf den Zeilen die annotierten Gene. Die Einträge der Matrix entsprechen den Zählungen der reads für ein Gen pro Sample.

Da solche Tabellen für eine komplette Darstellung über eine Heatmap meistens zu groß sind, werden die ersten 50 der variantesten Gene selektiert und aufgetragen.

Beim Ausführen des Skriptes erscheint das Interaktive Shiny-UI und bietet einem die Möglichkeit zwischen Distanzmaß und Clustering-Methode auszuwählen.

An Distanzmaßen stehen zur Auswahl:
‚Euclidian‘, ‚Maximum‘, ‚Manhattan‘, ‚Canberra‘, ‚Binary‘, ‚Minkowski‘.

An Clustering-Methoden stehen zur Auswahl:
‚ward.D‘, ‚ward.D2‘, ‚single‘, ‚complete‘, ‚average‘.

Je nachdem welches Distanzmaß und welche Clusteringmethode eingesetzt werden, verändert sich das Dendrogramm links neben der Heatmap entsprechend.

Die passende Heatmap wird im Main-Panel des Shiny-UI angezeigt.

Installationsanleitung

Das von mir erstellte R-Skript bedient sich an den Funktionen von drei verschiedenen Zusatzpaketen: edgeR, pheatmap und shiny

Sollten diese Pakete noch nicht installiert sein, können diese über die R-Konsolen-Eingabe wie folgt installiert werden:

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("edgeR")
```

Sowie

```
install.packages("shiny")
```

und

```
install.packages("pheatmap")
```

Das Paket wurde unter Windows 10 mit der R-Version 3.6.2 und R-Studio erfolgreich getestet.

Verwendete R-Funktionen/Skriptaufbau

Die Tabelle wird über die Kernfunktion `read.table()` in R eingelesen.

Um die Count-Tabelle zu normalisieren wurde mit der in EdgeR integrierten Funktion `cpm()` der CPM-Wert (Counts per million) ermittelt. Anschließend wurde mit der Funktion `apply()` die Varianz jedes Gens über die Versuchsreihe berechnet, um anschließend eine kleine Auswahl der am meisten in ihrer Expression variierenden Gene zu treffen.

Die in Shiny integrierte Funktion `pageWithSidebar()` erstellt das interaktive Userinterface. Die darin eingebettete Funktion `selectInput()` zeichnet das Auswahl-Dropdown-Menü in denen man das Distanzmaß und die Clusteringmethode bestimmen kann.

Im `mainPanel()` wird die Heatmap angezeigt.

Auf Serverseite bietet die Funktion `reactive()` die Möglichkeit, die von UI-Seite ausgewählten Werte aufzufangen und in den Plot zu übertragen.

Das Package pheatmap mit seiner Hauptfunktion `pheatmap()` ermöglicht einem das zeichnen einer Heatmap aus einer Matrix. Die darin verankerten Attribute `clustering_distance_cols`, `clustering_distance_rows` und `clustering_method` dienen hier zum Auswählen des angewandten Distanzmaßes und der Clusteringmethode.

Verwendeter Datensatz

Als Beispieldatensatz hab ich mich für die Counts-Tabelle aus der wissenschaftlichen Publikation „MOV10 and FMRP Regulate AGO2 Association with MicroRNA Recognition Elements“ (https://doi.org/10.1016/j.celrep.2014.10.054) entschieden. Mov10 ist mutmaßlich eine RNA-Helikase, die zusammen mit FMRP (fragile X mental retardation protein) an den miRNA (micro-RNA)-„Pathway“ verknüpft ist. In diesem Versuch werden acht Datensätze mit Replikaten zu drei verschiedenen Konditionen untersucht (Mov10 Overexpression, Mov10 knockdown, Mov10 irrelevant knockdown).

Eine potenziell interessante Fragestellung wäre, welche Expressionsmuster bei Verlust oder Anstieg des Mov10-Proteins entstehen.

Interpretation der Ergebnisse

Auf den erstellten Heatmaps lässt sich vor allem sehr gut der Unterschied des MOV10-Gens über die „Samples“ hinweg erkennen. Während sich in den knockdown Kandidaten die Expression gering hält, erkennt man sehr gut die unterschiedliche Einfärbung in den irrelevant knockdown und overexpression Replikaten. Auf diese Weise lässt sich beispielsweise das Experiment-Design verifizieren.

Es sind auch andere Gene je nach Kondition verschieden exprimiert. Durch die Einfärbung lassen sich solche Gene besonders gut identifizieren. Dadurch kann man über weitere gezielte Forschungsansätze Gen-Regulations-Netzwerke entschlüsseln und sie steuern.