CB2-101: Final Problem Set*

Contents

How to submit you answers

Submit all your answers in `ipynb` or `Rmd` format notebooks. You use `github_document` or `html_notebook` when using Rmd file.

First for the repository on github and then commit you answers in separate folders corresponding to each problem. Then create a pull request against the source repository from your forked copy.

Problem 1

The PFAM domain distribution for human proteome can be found at ftp://ftp.ebi.ac.uk/pub/databases/Pfam/current_release/proteomes/9606.tsv.gz. The first column of this file is the protein accession number. The location of the domain hit for each gene is given by the columns 2-5. Columns 2-3 are alignment start and end. Columns 4-5 are envelope start and end. Envelopes are generally considered the location of a domain on a gene. Write a R scrpt that takes 9606.tsv.gz file as a first argument, a protein accession number as a second argument, and a location (integer) as a third argument. The program should print the domain name (hmm_name), if the location falls within a domain for a given protein accession. The program should return nothing if the position is outside the boundaries of domains. We should be able to run the program like this

```
> problem1.R ../data/9606.tsv.gz O95931 20
> Chromo
```

Hint: You should create a `list` using the protein accession as key and location start and end as values. You might want to create a nested list or two separate lists.

*If you can solve all of the problems in this problem set, you're ready for research in bioinformatics. Contact us. There is a job waiting for you.

Problem 2

Swissvar is a database of human gene, their variations, and disease associations. The file can be downloaded from here: ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/variants/humsavar.txt. The 2nd column of the file is the protein accession numbers. These are the same accession numbers used in the domain file in Problem 1. The 6th column is `dbSNP` and reports the variation at a particular location. Using these two files, create a sorted list of domains according to the total number of their variations. The domains with higher variations should be on top. The program should not take any argument and output the domain list on STDOUT. The output should have two columns, separated by tab: domain name (hmm_name) and a number indicating variation, like this:

```
Domain  Variation
BRAC1   150
Chromo  100
...
```

Remember, your output will differ from the above shown output. The first line is the header. **Note**: You may skip writing a `run.sh` file for this problem.

Hint:

1. If needed, the location of the variation can be extracted from the `"AA change"` column of SwissVar data. For e.g., p.His52Arg means the variation is at the location 52.
2. Parsing ‘humsavar.txt" file is tricky. You may use the following R code to get a clean data.

```r
r <- read.table(
"ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/variants/humsavar.txt",
header = F,
skip = 49, sep = "", fill = T,
stringsAsFactors = F, flush = T,
nrows=78710)
r<- r[, -ncol(r)]
```

Problem 3

The first column of `humsavar.txt` file contains the gene name and the rest of the columns contains the other information. Using this file (A) list out the top five genes that are mutated in various human disease. (B) plot the frequency distribution of disease variants in human genome across all the genes in the file (C) calculate the average number disease causing mutations across all genes in human genome and mark this number on the previous plot as veritcal red line.

Hint: Remember to skip the information lines in the file and also note that `type of variant` column contains both disease causing and non-disease causing variants.

Problem 4

From the Swissvar file in Problem 2, we found the number of variations present in each domain. But this may be due to an artifact of domain abundance in human genome. Highly abundant domains will have higher chance of accumulating variations. We will test this hypothesis using a correlation between the abundance of domain and the accumulated variation. We calculated the abundance of domain in problem 3.

First run the scripts in the problems 2 and 3 and save their outputs in files. The output should remain in their original locations. **Caution**: The rows in the files are different. You many need to write a separate R script to `merge` the columns of the file. [Hint: Have a look at the `?merge()`]

Use a Rscript to read the files created in problem 3 and 4 (or, a merged file). Draw a linear regression plot between the abundance in X-axis and number of variation in Y-axis. The script should also report the correlation between these two variables.

Hint:

1. Read the tables from within your script. Read the files from their original location, like this:

   ```
   abundance <- read.table("../problem3/abundance.txt",header=T)
   ```

2. If you have used an intermediate script to merge the two files, then you should modify your RScript accordingly.

3. Don't forget to create a `run.sh` for this problem.

4. You can output the plot in a PDF file if you want. The correlation should be done using the R function `cor.test`. The test should be `two.sided` and method can be anything.

Problem 5

Use Fermi estimation (Lecture 1) to estimate a quantity starting from very little knowledge. The more creative you are in creating the problem, the more kudos you will get. Describe the question you are trying to answer and how did you derive the answer.