Bioinformatics, computer lab XIII
University of Potsdam WS 2019/2020, S. Hartmann

Exploring data!

In this computer exercise, you will statistically describe the types of mutations that were observed in different human populations across the world. The data and the study were discussed in class, they were published in 2017 in the journal eLIFE at `https://elifesciences.org/articles/24284`.

Download the data from Moodle and look at it with less. However, because the rows are much longer than your screen is wide, by default less wraps them into several lines on your screen. To prevent this from happening, you can use the option "-S": `less -S data.txt`.

- each row corresponds to one population. The population codes include the region or city (first three letters before the underscore) and the continental group (last three letters), as discussed in class A full list of population codes is available at `http://www.internationalgenome.org/category/population/`.
- each column corresponds to one of 96 mutational types. They are given as NXN_Y, where the Ns are the bases directly adjacent to the mutated base, and X mutated to Y. "AAA_T" therefore represents the mutation of AAA to ATA.
- table cells list the number of mutations that were observed for each mutational type in each of the populations and that fulfilled certain criteria.

1 Reading the data into R

Start R and read in the table and save it in a variable (e.g., "mutationsTable") using the `read.table` command. Specify that the table contains a header and that the first column represents the row IDs. Then attach the table, so that you're able to access rows and columns by their IDs.

✎ When you type `ls()` to see which objects are currently available in R, the new variable is listed. And when typing its name, you see its contents.

✎ Does the table contain any missing data?

2 Numerically describing the observed mutations

2.1 Rows

You can access the first row, the mutation spectrum of individuals of Africans in Yoruba, using either the row index `mutationsTable[1,]` or the row name `mutationsTable["YRI_AFR",]` or `mutationsTable["YRI",]`. It is also possible to list two rows at once, for example Nigerians and Peruvians: `mutationsTable[c("YRI","PEL"),]`

✎ List the mutation counts for Nigerians and Peruvians (or any other two populations) as described: just by eyeballing it – for which were more mutations observed?

✎ Use a specific R function to determine the minimum and maximum mutation counts for these two populations, i.e., the *range* of observed mutations.

Note: the question above is best answered with the function `range`. If you want to use the function `summary`, you have to first convert the single-row data frame into a vector: summary(as.numeric(mutationsTable["YRI",]))

✎ Now you know that Yorubans have one mutation type for which 479,300 mutations were counted – but which type is it? Find out with the function `which.max`.

✎ Is this the same mutation type for which the Peruvians (or another population) also have the most mutations?

✎ How many mutations were counted in total for Yorubans? for Peruvians? for ...?

2.2 OPTIONAL: Columns

Similar to working with rows, you can access the columns by index or by name, and you can access more than one column by name at once.

The table lists the context in which mutations occur: a mutation from A to G happens at different frequencies, depending on the flanking bases. Take a look at the mutation types CAT_G and GAC_G by listing them together: `mutationsTable[c("CAT_G", "GAC_G")]`

✎ What is the range of of mutation counts for these two types? What are mean and median, and do these values differ by much for these two mutation types? What does this mean? (Note that the function `summary` works directly on the table column; you don't have to convert it to a numerical vector.)

✎ Summarize your findings regarding two populations and two mutation types: What have you learned about mutation spectra? about rows and columns in R?

3 Graphically describing the observed mutations

3.1 One boxplot

✎ To see a boxplot of number of CAT to CGT mutations across all populations, type `boxplot(mutationsTable["CAT_G"]))`.

✎ How about adding a label for the y-axis, using the parameter `ylab` and/or `main`?
✎ Compare the numerical summaries of this data (above) to the boxplot.

✎ To save your boxplot in PDF format, you can use the method that was introduced last week:

```
dev.copy(device=pdf, file= "beautifulBoxplot.pdf")
dev.off()
```

3.2 Two boxplots

As discussed in class, the function `par()` can be used to change various graphics parameters. The parameter `mfcol()`, for example, can be used within this function to specify the number and layout of the graphics that are to be plotted.

✎ Use `par(mfrow=c(1,2))` to specify that the next two graphics you plot will be shown next to each other. Then plot the boxplots for CAT to CGT (CAT_G) mutations and for GAC to GGC mutations (GAC_G) side by side.

✎ Look at the two boxplots to compare the distribution of mutations. Does this agree with your numerical summaries? But is there anything wrong with these boxplots?

Of course there is something wrong! They are not on the same scales, and so the comparison is a little difficult. Adjust the range of the y-axes for all boxplots so they are the same, use the parameter `ylim` for this. For example, `boxplot(data, ylim=c(0,300))` would draw a boxplot with a range of the Y-axis from zero to three hundred.

✎ Redraw the boxplots so that both have the same y-axis scaling. Use as a guideline the largest value that is found within the data set. (You already looked up this value earlier.)

✎ If this plot is the way it should look like, you can now label the axes and add titles to the graphic.

✎ Don't forget to reset the plotting parameters when you are done ((par(mfrow=c(1,1)))).

3.3 All boxplots

3.3.1 By columns (mutation type)

So two of the mutation types occur at very different frequencies in these 96 populations! How about a comparison of all mutation types?

✎ Make the window of the graphics output as wide as your screen is, and then type `boxplot(mutationsTable)` into your R terminal.

Yes, the previous command plots all boxplots for the 96 mutation types side by side, on the same scale! The only problem is that, due to limited space, not all labels of the x-axis have been added. Can we fix that? – You bet!

✎ You'll adjust two optional parameters of the boxplot function: "cex.axis=0.5" will reduce the font size of the axis labels by 50%, and "las=2" will result in horizontal labels instead of vertical labels on the x-axis. Add both parameters to your boxplot command, separated by commas and within the boxplot function.

✎ What happens if you also add the option "col=rainbow(96)" or "col=rainbow(24)" to the

✎ Add labels and a title, if you want. And once you're satisfied with how the graphic looks, you can save it into a PDF file.

✎ Which mutation types occur most frequently? least often? fairly constant across the 96 populations? with greater variation across the populations?

3.3.2 By rows (population)

✎ As you saw, the boxplot function works for a table and plots a boxplot for each column. But what if we wanted to have one boxplot for each population (i.e., for each row)? The easiest way to accomplish this is to transpose the table before applying the boxplot command:

`t(mutationsTable)` transposes the table mutationsTable: this means that rows become columns, and columns become rows.

`boxplot(t(mutationsTable),col=rainbow(26))` plots boxplots for the transposed table: by column, which now correspond to populations! Try it out!

✎ For which populations and continental group(s) were the most mutations observed? the least?

Summary

✎ Summarize what you have learned about human mutation spectra.