Next Steps in Gene Prioritisation Exercises

Before you begin, copy all the data files from "penelopeCloud" (or from rnaseq2026.buschlab.org) to your home directory. Then go to g:Profiler (biit.cs.ut.ee/gprofiler/) and start the exercises.

1. Use the `Nic.id.sig.tsv` file (which contains a list of Ensembl IDs of zebrafish genes that are significantly differentially expressed after exposure to nicotine) as input to g:Profiler. Make sure you change the organism to "Danio rerio (Zebrafish)" but otherwise just use the default settings. Have a look at the results and the detailed results.

2. Try doing the same with `Amp.id.sig.tsv` and `Oxy.id.sig.tsv` as well. Are you able to spot any common functional enrichments between all three treatments or between a particular pair of treatments?

3. All of the files are ordered by p-value (with the lowest p-value at the top). Try analysing all three lists with the "Ordered query" option turned on. What effect does this have on the results?

4. Try all three lists with both "Bonferroni" and "Benjamini-Hochberg" multiple testing correction, rather than the default "g:SCS". What effect does this have on the results?

5. Try changing the background to "All known genes" rather than the default of "Only annotated genes". How do the results change? Then try using `Nic.id.all.tsv` as a custom background. What effect does this have? Why might it not be an appropriate background? What would be a better background?

   Try making a custom background that includes all the genes that passed independent filtering (i.e. those that have an adjusted p-value that isn't "`NA`"). You will have to use, for example, the `Nic.annotation.all.tsv` file to do this. What effect does this have on the results? If you manage that, then try making a custom background that includes all the genes with non-zero total counts. You will have to use, for example, the `Nic.counts.all.tsv` file to do

this. What effect does this have on the results?

Note: The second half of this question (and some of the subsequent questions) will involve manipulating the TSV files in some way. It's up to you how you do this. You could import the files into Google Sheets. You could use Excel or Numbers on your own computer. You could use the LibreOffice Calc software on the course VMs. You could use R or Python if you know them. You could even use command line tools like `awk`! Just use whatever you are most comfortable with.

6. Split the three gene lists into up-regulated and down-regulated genes. Analyse each of these lists separately. What effect does it have on the results? Make sure you use an appropriate background.

7. Analyse all three lists together using the "Run as multiquery" option. It should help you find commonalties between the three comparisons. Do you find more in common than you did in Q2 above? Which pair of comparisons has the most in common? Can you use g:Profiler to output a figure for publication that illustrates the commonalties?

8. For each of the three lists, use g:Orth to convert the list from zebrafish Ensembl stable IDs (e.g. ENSDARG00000041294) to human Ensembl stable IDs (e.g. ENSG00000196408). Take each list of human IDs and run g:Profiler/g:GOSt again. How do the results change? Combine all three human lists and run them together using the "Run as multiquery" option. Does this have any effect on the commonalties?

9. Use the g:Convert tool to convert one of the lists from Ensembl stable IDs to Entrez Gene accessions (ENTREZGENE_ACC). How many genes are lost in the conversion. Try using the Entrez Gene accessions as input to g:Profiler/g:GOST. Does it have any effect on the results?

Optional Tasks:

1. Do you have your own gene list from a current or previous experiment? Try analysing it with g:Profiler.

2. If you don't have your own gene list then try finding a list in a paper relevant to your work. Ideally one that did functional enrichment analysis of some kind. Can you reproduce their results using g:Profiler?

3. Try using Enrichr on one of the gene lists. In what ways does it differ from g:Profiler?

4. If you know R, then try installing the `gprofiler2` package and analysing one of the gene lists in R. See biit.cs.ut.ee/gprofiler/page/r for more information.

5. If you know Python, then try installing the `gprofiler-official` package and analysing one of the gene lists in Python. See pypi.org/project/gprofiler-official/ for more information.