

# ADER17

Functional Enrichment Analysis

# Extracting meaning from a list of genes

Git  
CG18519  
del  
CG13319  
RNaseX25  
CG15611  
CG11309  
CG1657  
CG6404  
CG9987  
sf1  
CG4300  
CG4065  
phr6-4  
Myo31DF  
...



- ‘Omics lists of “interesting” genes (e.g., RNAseq, microarrays, proteomics) are essentially meaningless on their own
- We’re typically interested in understanding phenomena at the cellular or organismal level, rather than the gene level

# Extracting meaning from a list of genes

- In order to extract meaning from our list we have to abstract from the gene level to a higher level, in concert with the biological question we are addressing
- We're focusing on the most common case, which is the functional level, but enrichment analysis can also be applied to other biologically meaningful aspects (e.g., the chromosome level, the transcription regulation level, etc)
- To go from the gene level to the functional level, we need a functional classification of our genes

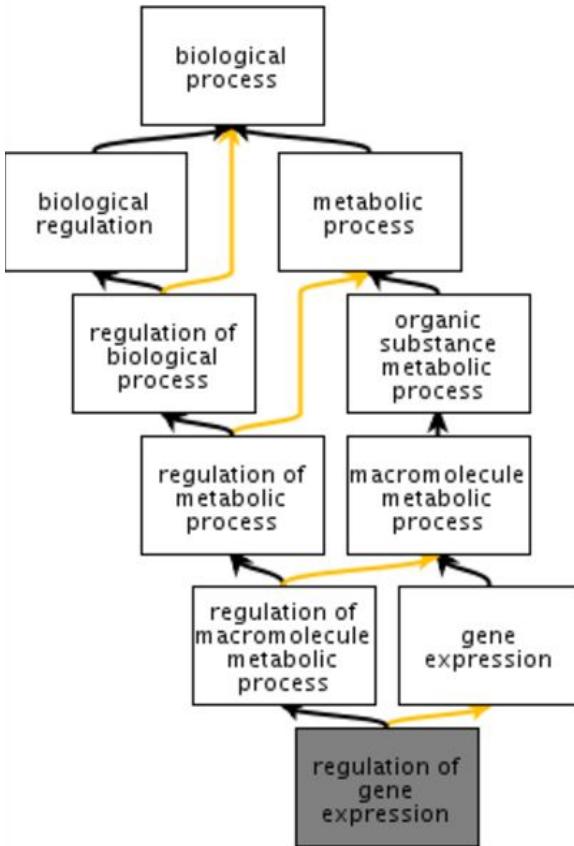
# Functional annotations

- Functional annotations are assignments of functional classifications to genes
- There are multiple functional classifications in use for different functional aspects:
  - Enzymes: EC classification
  - Metabolic pathways: KEGG
- The Gene Ontology (GO) is the most comprehensive functional classification scheme available, and thus is the most commonly used for enrichment analysis

# The Gene Ontology

- GO is a functional classification scheme that covers three levels of gene function, called GO types or aspects:
  - Molecular Function: the individual functional level (e.g., GTPase, transcription factor)
  - Biological Process: the cellular and/or organismal functional level (e.g., signalling, muscle development)
  - Cellular Component: the locational level (e.g., cellular membrane, nucleus)
- Each GO type is in essence a (near-)independent classification scheme

# The Gene Ontology

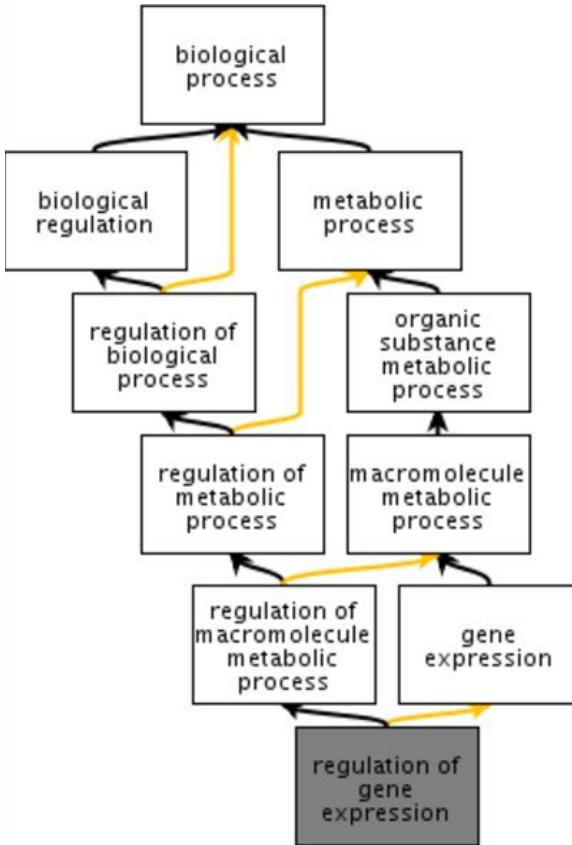


- Each GO type is structured as a directed acyclic graph (i.e., a relaxed hierarchy with multi-parenting)
- In addition to subclass ('is a') relations, there are 'part of', 'regulates', and 'occurs in' relations
- Although GO types are 'is a' orthogonal, *molecular functions* can be 'part of' *biological processes*, and both can 'occur in' *cellular components*

# GO slims

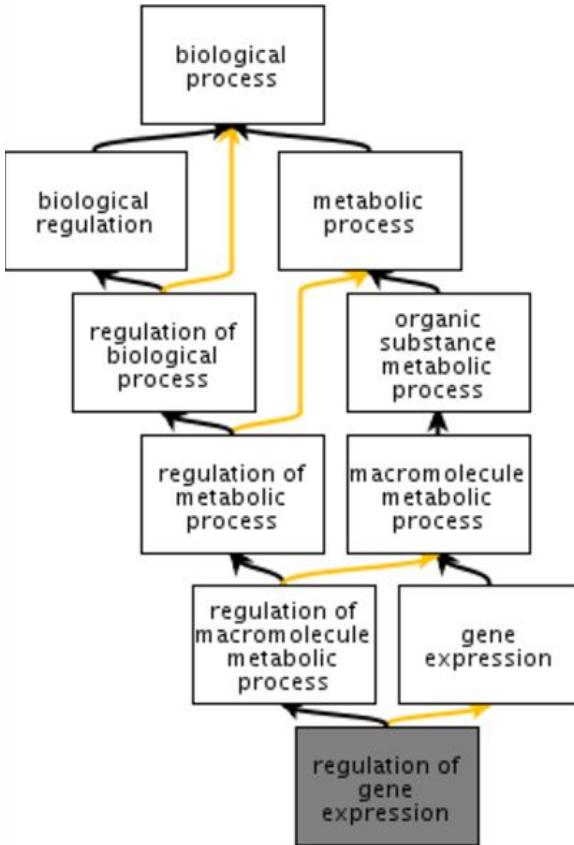
- GO slims are ‘trimmed’ versions of GO where the specific fine grained terms have been removed and only broader terms are present
- They usually cover the whole breadth of GO, albeit slims for particular species may exclude sections that are not applicable to that species
- GO slims are useful for giving an overview of the GO annotations of a genome or a large collection of genes, when a broad classification is sufficient

# GO annotations



- A GO annotation is the assignment of a GO term to a gene (product)
- A gene may have multiple annotations, even of the same GO type
- According to the **true path rule**, a gene annotated to a term is implicitly annotated to each ('is a') ancestor of that term; other relations apply to it accordingly

# GO annotations



- Are assigned an evidence code that encodes the type of evidence supporting the annotation
- Electronic annotations (IEA) are considered less reliable than manually curated annotations
- Genes can be explicitly annotated to both a term and its ancestor if the annotations have different evidence codes (otherwise it would be redundant)

# Getting GO annotations

- GO annotations of genes and proteins are available in most major genome databases, in UniProt, and in dedicated search engines such as AmiGO (<http://amigo.geneontology.org>) and QuickGO (<https://www.ebi.ac.uk/QuickGO/>)
- You can download complete genome/database annotation sets from:
  - <http://www.geneontology.org/page/download-annotations>
  - <http://www.ensembl.org/biomart>

# Task

- Go to BioMart through Galaxy (Galaxy > Get Data > BioMart) and get the GO annotations for the mouse genome in tsv (Gene Stable ID; GO term accession).
- Download the latest version of GO and upload it into Galaxy.

# Extracting meaning from a list of genes

Git → somatic muscle development, ...  
CG18519  
del → oogenesis, oocyte development, ...  
CG13319  
RNaseX25  
CG15611  
CG11309  
CG1657 → regulation of protein transport, ...  
CG6404  
CG9987  
sfl → segment polarity determination, ...  
CG4300  
CG4065  
phr6-4  
Myo31DF → determination of left/right symmetry, ...  
...

- By going from genes to GO annotations, we start to see some meaning...
- But there are too many genes to examine manually...
- And we cannot discern patterns easily, much less assess their significance!!!

# Enrichment analysis

Git → developmental process  
CG18519  
del → developmental process  
CG13319  
RNaseX25  
CG15611  
CG11309  
CG1657 → ...  
CG6404  
CG9987  
sf1 → developmental process  
CG4300  
CG4065  
phr6-4  
Myo31DF → developmental process  
...

- How can we assess if an observed GO term frequency is statistically significant?
- We need to compute the probability of said frequency arising from chance
- This is an application of [Fisher's exact test](#), which in the genomic context is typically called enrichment analysis

# The statistics behind enrichment analysis

- The [hypergeometric distribution](#) describes the probability of  $k$  successes in  $n$  random draws, without replacement, from a finite population of size  $N$  that contains exactly  $K$  objects with that feature:
  - $k$  is the sample frequency
  - $n$  the sample size
  - $K$  is the population frequency
  - $N$  is the population size
- It applies to card games like Poker, to lottery games such as Euromillions, and generically to most “sampling” situations

# The statistics behind enrichment analysis

- A list of genes selected under a given criteria (such as differential expression) is, for all intents and purposes a “sample” of the genome (or of the genes under analysis).
- If we’re interested in a specific feature, such as GO term, then:
  - Sample frequency: number of genes in the set annotated with the term
  - Sample size: total number of genes (with any annotation) in the set
  - Population frequency: number of genes in the genome (or under analysis) annotated with the term
  - Population size: total number of genes (with any annotation) in the genome (or under analysis)

# The statistics behind enrichment analysis

- In sampling events, the null hypothesis is usually that the sample (or the frequency of the desired feature therein) is the product of chance alone
- We reject the null hypothesis when the likelihood of observing **at least** the recorded frequency by chance alone is lower than our significance threshold (e.g., 0,01)
- This is the one-tailed version of [Fisher's exact test](#) or the Hypergeometric test for over-representation

# The statistics behind enrichment analysis

- The p-value of the one-tailed Fisher's exact test is given by:

$$P_{\text{Hypg}}(x \geq k | n, K, N) = \sum_{i=k, \dots, \min(n, K)} P_{\text{Hypg}}(x = i | n, K, N)$$

- Note that you cannot draw more successes ( $k$ ) than the number of successes in the population ( $K$ ) or the number of draws ( $n$ )
- You can compute individual probabilities using spreadsheet software or the p-value (cumulative probability) using  $R$
- As usual, this p-value corresponds to the probability of making a type I error, i.e., erroneously rejecting the null hypothesis

# Correcting for multiple tests

- In enrichment analysis, we typically want to perform multiple tests – as many as the number of functional aspects of the genes in our set
- If we made 1000 related tests, we would expect to obtain a p-value of 0.001 in one of them by chance alone
- Thus, when performing multiple tests, it is necessary to correct the p-value to account for this

# Correcting for multiple tests

- **Family-wise error rate (FWER):** control the probability of making at least one false discovery – more conservative but safer
  - Bonferroni correction: multiply the p-values by the number of tests to obtain corrected p-values
- **False discovery rate (FDR):** control the ratio of false discoveries – more powerful
  - Benjamini-Hochberg correction: step-wise correction that produces q-values, which indicate the ratio of false discoveries you are accepting if you reject the null hypothesis

# GO enrichment analysis

- GO is hierarchical, and we must consider both direct and inherent annotations when doing enrichment analysis
- This enables integration: with specific terms like “somatic muscle development” and “segment polarity determination”, we would be unable to find a pattern, but we can find it with the more generic “developmental process”
- However, it also makes the analysis of the results more difficult: we often get enriched terms at several levels of specificity, many of which are interrelated, and this is not always readily apparent

# GO enrichment analysis

- Multiple test corrections should be apply to multiple tests in a family:
  - The three GO types correspond to different families, and thus should be treated separately for the purpose of multiple test corrections
- Some hierarchically related tests are redundant:
  - If the study frequency of “DNA binding” is the same as that of “binding”, then testing the latter is unnecessary – it can only be significant if the former is also significant, and the more specific the term the more meaning we can derive

# What to test for enrichment

- Biological process is typically the most interesting GO type to test for enrichment, but molecular function and cellular component may also be relevant for validation in particular studies (e.g., in a proteomics study where you sampled only membrane proteins, you should check for enrichment of “cellular membrane”)
- In gene expression studies, we can analyze all differentially expressed genes together, or separate them into overexpressed and underexpressed genes, analyze them separately, and compare the results – both approaches may make sense, depending on the study and the goal of the analysis

# Tools for enrichment analysis

- Web tools:
  - GOrilla: <http://cbl-gorilla.cs.technion.ac.il/>
  - GO: <http://www.geneontology.org/page/go-enrichment-analysis>
- Galaxy tools:
  - GOEnrichment [IGC]: <https://github.com/DanFaria/GOEnrichment>
  - Ontologizer: <http://ontologizer.de/>
- R tools:
  - gsea
  - GOstats
  - topGO

# Task

- Run an enrichment analysis test on GOrilla. Use the [FEA\\_dataset1](#), which contains the overexpressed genes from the Drosophila melanogaster dataset with 300 random genes differentially expressed.
  - Choose “Two unranked lists of genes” as the running mode
  - Paste or upload the study set into “target set” and the population set into “background set”
  - Use the biological process ontology, then repeat the analysis for the molecular function ontology
- Are there significantly enriched terms at 0.01 and 0.001 significance without multiple test corrections? And with the correction?

# Interpreting enrichment analysis results

- **Statistically significant ≠ biologically meaningful**
- More specific terms are generally more meaningful, and are easier to interpret
- Generic terms are often challenging to interpret, but this does not mean they are meaningless
- Broader experiments will generally lead to results that are more difficult to interpret

# Interpreting enrichment analysis results

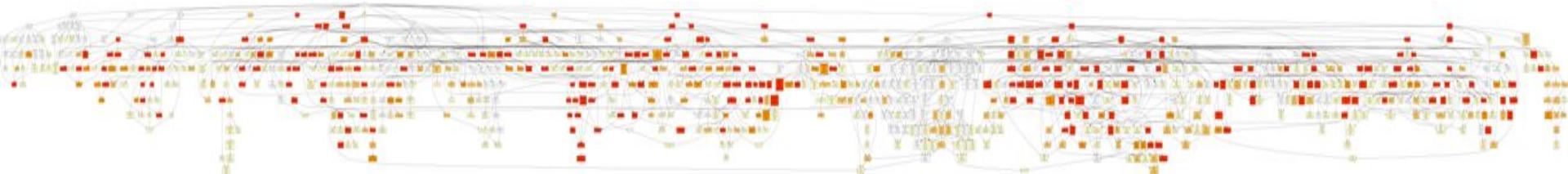
- Some subsets (e.g., only underexpressed or only overexpressed genes) may be more meaningful or easier to interpret than others

Top enriched GO terms, mouse iPSCs vs ESCs up
single-organism process
oxidation-reduction process
single-organism metabolic process

Top enriched GO terms, mouse iPSCs vs ESCs down
modulation of synaptic transmission
regulation of nervous system development
behavior

# Interpreting enrichment analysis results

- Related terms should be considered together as a cluster
- Graph views can help interpret the results and identify such clusters
- But GO is large, and broad experiments tend to produce thousands of differentially expressed genes, resulting in hundreds of enriched terms and **very large graphs**



# Interpreting enrichment analysis results

- There are solutions available for conflating these clusters and thus simplifying the output of functional enrichment results
  - Integrated into enrichment analysis tools ([GOEnrichment](#))
  - Stand-alone tools ([REVIGO](#))
- These can facilitate the interpretation of the data, but evidently, some information is lost

# Task

- Run an enrichment analysis test on the GOEnrichment tool in Galaxy, using the [FEA\\_dataset2](#).
  - Use the go.obo and Mouse annotation file you got from BioMart earlier, as well as the dataset files.
  - Run the program with the "Summarize Output" parameter set to off (and otherwise default parameters), then analyze the results tables and graph files.
- Repeat the previous run, but this time with "Summarize Output" set to on.
  - Compare the results tables and graph files.

# Task

- Run another enrichment analysis test on the GOEnrichment tool, using [FEA\\_dataset3](#), but this time do it in the command line.
  - You can download the mouse annotation file you were using directly from Galaxy, and the GOEnrichment tool from the [latest release on GitHub](#).
  - You will have to process the dataset to generate the lists of overexpressed and underexpressed genes. Hint: paste the dataset into a spreadsheet, and sort it and manipulate it there, then copy your up and down study sets, as well as your population set into text files.
  - Run the tool with the same parameters as your previous run in Galaxy: read the [Readme file](#) or type "java -jar GOEnrichment.jar --help" to learn how to specify the command line parameters.
  - Analyze the results tables and graph files, comparing the overexpressed and underexpressed results.