# *Genome Function Phylogenetics*

*Dennis Psaroudakis*

*April 18th 2019*

When we build phylogenetic trees, we often use the sequences of certain genes as the basis for our tree. They are the direct substrate of evolution so that makes sense, but when we think of evolution on a macro level, we don't really think of the sequences, we think of phenotype. Using phenotypes for evolutionary trees is a hard thing to do though because the choice of characteristics can be kind of arbitrary (`http://wiki.c2.com/?BorgesClassificationOfAnimals` Chinesische Tiere (Jorge Luis Borges 1966)), because it is very far away from the material evolution actually happens on, i.e. DNA. In this paper, I tried to go a middle ground: Instead of looking at the genetic sequences or the macromolecular phenotype, I'm looking at the presence of certain biological processes, molecular functions, or cellular components, in other words: along the path of evolution, any meaningful change in an organism goes along with the development, change, or loss of such a function; what good is a change in the DNA sequence if it does not lead to a modified function of that gene?

## *The Gene Ontology*

Historically, the function/role that a gene plays in an organism has always been described in natural language, however the researcher characterizing that gene deemed best. While this is nice to read, it is not very useful if you want to do computation on it, as computers are (still) horrible at understanding natural language and determining the structure in meaning behind the words. Additionally, different people will describe the same thing with different words, which has the potential for misunderstanding.

Ontologies try to alleviate these problems by providing a strictly organized and controlled vocabulary and defined relationships between the terms, so that the same statement always means the same thing, no matter the context or the author. Additionally, ontologies can be understood by computers if all relationships and terms are clearly defined.

The Gene Ontology is such an ontology that describes genes by the properties of their product. In our case these gene products are proteins, and they can be characterized in three different aspects:

- What biological processes is this protein part of? (e.g. photosynthesis or autophagy)
- What molecular functions does the protein carry out? (e.g. ethylene binding or RNA ligation)
- What cellular component is the protein active at? (e.g. outer membrane or nucleus)

"The mission of the GO Consortium is to develop an up-to-date, comprehensive, computational model of biological systems, from the molecular level to larger pathways, cellular and organism-level systems."

— GO Consortium (geneontology.org)

Within each of these aspects, the Gene Ontology defines a huge number of terms (2,675,070 in total), that range from very general to very specific:

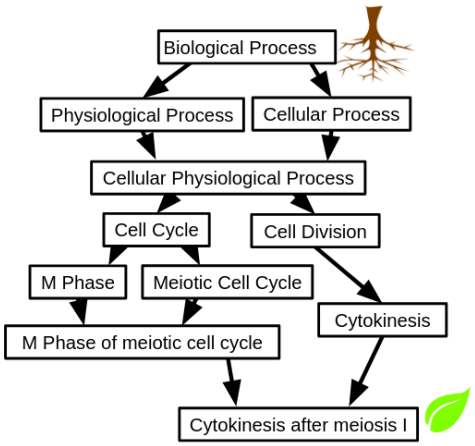



Figure 1: subtree of the *Biological Process* ontology. The terms are organized in such a way that more general terms are always true for any of their more specific child terms. For example, any protein that is part of *Cytokinesis after Meiosis I*, is also obviously part of Cytokinesis, the Cell Cycle etc. That way, a gene that has been annotated with the term *Cytokinesis after Meisosis I* (leaf term), has implicitly been annotated with all of that term's parent terms as well, all the way up to the root term.

When proteins are annotated with these terms instead of just natural language, we can now computationally answer some interesting questions, such as:

- How similar in function is protein A to protein B? (One answer would be: How many steps in the GO graph do I need from term A to B? The fewer steps, the more similar the function)
- If protein A is involved in Biological Process XYZ, what other proteins are involved in that same process?

The Gene Ontology is quite well established in the field, so you will find GO annotations for almost all relevant UniProt entries or use dedicated tools like AmiGO or QuickGO to examine a protein of interest.

### *Data*

Annotating genes with their functions can be done experimentally (e.g. by knocking out a certain gene and seeing what processes in the cell are affected), but that is a time-consuming and expensive process, so methods have been developed that try and predict the function of a given gene. Our lab has developed such a pipeline called GOMAP which combines different prediction approaches and is able to generate high-confidence and very extensive Functional Annotations in a reproducible manner (CITE PAPER). We have been applying this pipeline to whole-genome assemblies of different plant species and generated functional annotations for every gene in each genome. These annotation sets are (or will be shortly) available from `https://dill-picl.org/projects/gomap/gomap-datasets`.

### *Method*

Starting point of our method are the functional annotation sets, one for each genome, which annotate every gene in the genome with one or more GO terms. In more mathematical terms the genome annotation set is a list of tuples $(G, T)$ with $G \in$ Genes in that genome and $T \in$ Terms in the Gene Ontology.

We can use the hierarchical structure of the Gene Ontology to obtain the ancestors $A_i$ of any term $T_i$; in other words the gene $G_i$ is not just annotated with the term $T_i$ itself but also with all GO terms that are a more general statement of that term (e.g. any gene that is part of a metabolic process is thereby also part of a biological process). We do that for all terms $T$ in the dataset and combine all of the terms and their ancestors into one big genome-wide set $S$, irrespectively of the gene they were originally associated with: $S = \bigcup_{i=1}^{x}(T_i \cup A_i)$.

When this superset of annotations is created for each of the datasets, we can use the Jaccard Distance as a measure of how (dis-)similar any two sets are from each other, or in biological terms how different the two genomes are on a functional level:

$$\text{Jaccard Distance}(S_a, S_b) = 1 - \frac{|S_a \cap S_b|}{|S_a \cup S_b|}$$

Applying this formula to all pairwise combinations of the genomes we're looking at yields a $S \times S$ distance matrix that can then serve as the input for a neighbor joining algorithm (provided by PHYLIP).

### *Result*

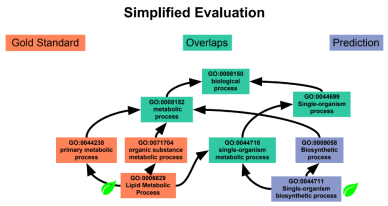
The tree generated by neighbor joining is displayed in Figure 2. 4 of the 5 maize assemblies are grouped together in a clade, while the last one is an outlier to the remaining grasses and sits on a very long branch. In the remaining plants, cotton is an outlier to the legumes, which are all grouped in a single clade.

### *Discussion*

This tree shows surprising similarity to the taxonimic tree one might expect if it had been built on sequences[1]. The maize cultivars are all grouped together and there's even the distinction between the non-stiff stalk (W22 and M017), iodent (Ph207), and stiff stalk (maize_v4) classes. Only maize_v3, which is the same cultivar as maize_v4 (B73) is waaaay out there where it probably doesn't belong. We're currently investigating this but it seems that there is a lot wrong with the genome assembly itself, which would naturally influence the functional annotations derived from it.

| Gene | GO Term |
|---|---|
| Os01g0601625 | GO:0050896 |
| Os01g0601625 | GO:0016021 |
| Os01g0601625 | GO:0016301 |
| Os01g0601651 | GO:0003677 |
| Os01g0601651 | GO:0009699 |
| Os01g0601651 | GO:0050790 |
| Os01g0601651 | GO:0050794 |
| Os01g0601651 | GO:0050896 |
| Os01g0601675 | GO:0007275 |
| Os01g0601675 | GO:0016310 |
| Os01g0601675 | GO:0050789 |
| ... | |

The general idea of using the Jaccard Distance in this context is to measure the overlap of two subtrees in the GO hierarchy. Say, for simplicity, that we're looking at two genomes (here called Gold Standard and Prediction) that each only contain one single GO term (marked by a leaf). First, we add all ancestors of that leaf term to each subtree. Then, we determine the overlap (which corresponds to $S_a \cap S_b$), and divide the number of nodes in this overlap by the number of nodes in either of the two subtrees ($S_a \cup S_b$).




In the case of this example, the Jaccard Distance of Gold Standard and Prediction would be $1 - \frac{4}{9} = \frac{5}{9}$

[1] I should probably display that expected tree here.

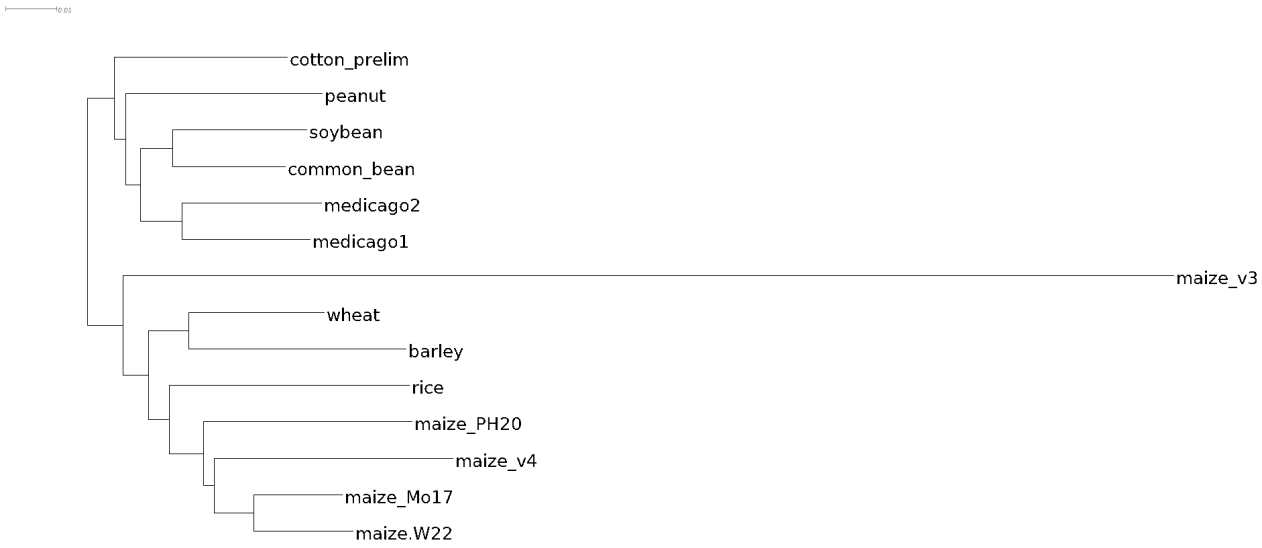



Figure 2: Resulting tree. maize_v3 and maize_v4 are two assemblies of the same maize cultivar, medicago1 and medicago2 are two different genotypes.

*Reproducing the Tree*

To reproduce this tree, follow these steps on a Linux machine that has PHYLIP available:

```
git clone https://github.com/Thyra/EEOB563.git paper_dennis # Clone the repository
cd paper_dennis/final_project # change into the directory
./build_distance_matrix > infile
module load phylip # (if you're on HPC)
neighbor
```

The resulting tree is in `outtree`. Manually root it (e.g. with Dendrogram) outside of the grasses (maize, wheat, barley) and you should end up with the same tree. This will use the pre-summarized `.tree.json` files in annotation_sets, if you want to completely reproduce it from scratch, delete them and only leave the `.gaf.gz` files. But be warned, this will take a while to calculate (probably over two hours).