Orthology based gene functional annotation and genome comparisons

Sequence homology to functionally validated proteins (UniProt database and TAIR10)

The UniProt database (`/data/courses/assembly-annotation-course/CDS_annotation/data/uniprot/uniprot_viridiplantae_reviewed.fa`) contain sequences of functionally validated proteins with known functions. Align your proteins against it and report how many of your annotated proteins have homology to known ones. What can you conclude from this quality check?

```
module load BLAST+/2.15.0-gompi-2021a
makeblastb -in /data/courses/assembly-annotation-
        course/CDS_annotation/data/uniprot/uniprot_viridiplantae_reviewed.fa
        -dbtype prot # this step is already done
blastp -query {maker_proteins.filtered.fasta} -db /data/courses/assembly-
        annotation-
        course/CDS_annotation/data/uniprot/uniprot_viridiplantae_reviewed.fa
        -num_threads 10 -outfmt 6 -evalue 1e-5 -max_target_seqs 10 -out
        {blastp_output}
# Now sort the blast output to keep only the best hit per query sequence
sort -k1,1 -k12,12g {blastp_output} | sort -u -k1,1 --merge >
        {blastp_output}.besthits
```

You can also map the protein putative functions to the MAKER produced GFF3 and FASTA files:

```
cp maker_proteins.filtered.fasta maker_proteins.filtered.fasta.Uniprot
cp filtered.genes.renamed.gff3 filtered.genes.renamed.gff3.Uniprot.gff3
$MAKERBIN/maker_functional_fasta {uniprot_fasta} {blast_output}.besthits
        {maker_proteins.filtered.fasta} >
        {maker_proteins.filtered.fasta.Uniprot}
$MAKERBIN/maker_functional_gff {uniprot_fasta} {blast_output}.besthits
        {filtered.genes.renamed.gff3} >
        {filtered.genes.renamed.gff3.Uniprot.gff3}
# {uniprot_fasta} is /data/courses/assembly-annotation-
        course/CDS_annotation/data/uniprot/uniprot_viridiplantae_reviewed.fa
# {blast_output}.besthits is the blastp besthits file from the previous step
```

Now, get the best blast hit with Arabidopsis thaliana TAIR10 representative gene models:

```
blastp -query {maker_proteins.filtered.fasta} -db /data/courses/assembly-
        annotation-
        course/CDS_annotation/data/TAIR10_pep_20110103_representative_gene_model
        -num_threads 10 -outfmt 6 -evalue 1e-5 -max_target_seqs 10 -out
        {blastp_output}
# Now sort the blast output to keep only the best hit per query sequence
sort -k1,1 -k12,12g {blastp_output} | sort -u -k1,1 --merge >
        {blastp_output}.besthits
```

By the end of all of these steps, you should have the annotated GFF3 and FASTA files with putative functions from UniProt database, pfam domains, and GO terms. You will also have the best blast hit to Arabidopsis thaliana TAIR10 representative gene models.

> **Guiding questions:**
>
> What proportion of proteins have a significant hit to well-annotated proteins (with curated functions) vs. uncharacterized proteins?
> Are there length or completeness biases in proteins without UniProt hits (e.g., short fragments)?
> Do you find the FLC gene (AT5G10140) in your annotation? What is its putative function that you can guess from GO terms and pfam domains?

Comparative Genomics with OrthoFinder and GENESPACE

GENESPACE Overview GENESPACE is an R package designed for synteny- and orthology-constrained comparative genomics. It leverages DIAMOND2 for BLAST-like hits and OrthoFinder for identifying orthogroups and orthologues, followed by extracting syntenic regions using graph- and

cluster-based methods. This process helps define expected gene positions across multiple genomes, facilitating the comparative study of conserved gene order. Learn more in the GENESPACE eLife publication.

- **GENESPACE GitHub Repository**: GENESPACE
- **OrthoFinder GitHub Repository**: OrthoFinder
- **OrthoFinder Results Guide**: Exploring OrthoFinder Results

Step 1: Prepare the GENESPACE files and scripts

GENESPACE requires specific formats for each genome:

- **BED file**: Coordinates for each gene (format: chr, start, end, gene name)

```
chr1 3631 5899 AT1G01010
chr1 5928 8737 AT1G01020
chr1 11649 13714 AT1G01030
```

- **FASTA file**: Peptide sequences, with headers matching the gene names in the BED file.

```
>AT1G01010
MLVMSECKGRDRSPSSSM
>AT1G01020
MGASGRGAGEQQSPSSTG
>AT1G01030
MGASGRGAGRQQSPSSTG
```

Sample Input Preparation

To analyze *Arabidopsis* accessions and compare them to the reference genome (*Arabidopsis thaliana* TAIR10):

1. **Create a BED file**: Extract positions of each gene from the filtered gff3 file. The file should be named `${Accession}.bed` and the content should be in the format `chr\tstart\tend\tgene_name`. To prepare this file, you have to extract gene features from the GFF3 file corresponding to the selected contigs.

   ```
   # Example command to extract gene features from GFF3
   grep -P "\tgene\t" filtered.genes.renamed.gff3 > temp_genes.gff3
   ```

Then use an `awk` command to format it into a BED file. BED is a 0-based format, so ensure to adjust the start position accordingly (subtract 1 from the GFF3 start position). Subset the 9th column to get the gene ID.

```bash
awk 'BEGIN{OFS="\t"} {split($9,a,";"); split(a[1],b,"="); print $1, $4-1, $5, b[2]}' temp_genes.gff3 > ${Accession}.bed
```

2. **Prepare a FASTA file**: Extract the longest protein sequences for the genes identified naming it `${Accession}.fa`.
3. **Organize Files**: Create a working directory structure as below and include *TAIR10* reference files (`TAIR10.fa` and `TAIR10.bed`) found at `/data/courses/assembly-annotation-course/CDS_annotation/`.

Directory Structure

```
/workingDirectory
├─ peptide
│    ├─ Accession1.fa
│    ├─ Accession2.fa
│    └─ TAIR10.fa
├─ bed
│    ├─ Accession1.bed
│    ├─ Accession2.bed
│    └─ TAIR10.bed
```

> **Note:**
>
> You should have at least two accessions plus the reference TAIR10 genome for meaningful comparisons.
> Suggest to use your accession + TAIR10 + three other accessions from the course data folder `/data/courses/assembly-annotation-course/CDS_annotation/data/Lian_et_al`.

Step 2: Prepare GENESPACE in R

Prepare the Rscript to Initialize Directories and run GENESPACE

```
library(GENESPACE)
args <- commandArgs(trailingOnly = TRUE)
# get the folder where the genespace workingDirectory is located
wd <- args[1]
gpar <- init_genespace(wd = wd, path2mcscanx = "/usr/local/bin/MCScanX")
# run genespace
out <- run_genespace(gpar, overwrite = TRUE)
pangenome <- query_pangenes(out,   bed = NULL,   refGenome = "TAIR10",
        transform = TRUE,   showArrayMem = TRUE,   showNSOrtho = TRUE,
        maxMem2Show = Inf)
# save pangenome object as rds
saveRDS(pangenome, file = file.path(wd, "pangenome_matrix.rds"))

# in your next script, you can load the pangenome matrix with:
# pangenome <- readRDS(file.path(wd, "pangenome_matrix.rds"))
# and then use it for downstream analyses, e.g., calculating core, accessory
        and specific genes
```

Save it as `genespace.R` in your scripts folder.

Step 3: Run GENESPACE

To run GENESPACE, you need to prepare the an Rscript as above that you can run using the singulariy container as follows:

```
apptainer exec \
    --bind $COURSEDIR \
    --bind $WORKDIR \
    --bind $SCRATCH:/temp \
    $COURSEDIR/containers/genespace_latest.sif Rscript scripts/genespace.R
        /path/to/genespace/workingDirectory
```

Step 4: Explore and Visualize Results

After running GENESPACE, you will have a `pangenome_matrix.rds` file in your working directory. This file contains the orthogroup assignments of genes after taking into consideration the syntenic relationships among the genomes analyzed. Core, accessory and specific orthogroups and genes can be identified based on their presence across the accessions.

- **Core genes / Core orthogroups**: present in all Accessions

- **Accessory genes / Accessory orthogroups**: present in some but not all Accessions
- **Unique genes / Unique orthogroups**: present in only one Accessions

Make a summary table of the number of core, accessory and unique orthogroups and genes for your accession.

> ### Guiding questions:
>
> **Orthogroup Analysis**: How many orthogroups are shared between the accessions and the reference genome? How many are unique to each?

> ### Note:
>
> Knowing the genes that are present in unique vs shared orthogroups can provide insights into the evolutionary relationships between the accessions and the reference genome. It also allows you to ask questions about the functional categories (GO terms) that are enriched in shared vs unique orthogroups.

Step 5: Visualize Synteny

Dotplots and syntenic maps are a useful tool for assessing and improving the quality of nonreference genome asemblies, and also for visualizing structural rearrangements between species (deletions, duplications, insertions, translocations, inversions) and within species (retained old duplicated genomic regions; relicts of past whole genome duplication events).

Dotplots

For visualizing pairwise synteny, GENESPACE produces:

- **All hits** above a score threshold (`.rawHits.pdf`).
- **Syntenic anchor hits** by syntenic block ID (`syntenicHits.pdf`).

Reference GENESPACE Guide for more details.

Synteny Maps (Riparian Plots)

Riparian plot is the primary method to visualize syntenic relationships among >2 species. Each chromosome is a rounded rectangle polygon. Each syntenic block, phased and colored by the reference genome chromosomes (or contigs), are visualized as braids. Refer to the Riparian Guide for instructions.

> **Guiding questions:**
>
> Do you see any major structural rearrangements between accessions?