Further development on FTAG-Finder

a pipeline to identify Gene Families and Tandemly Arrayed Genes

M1 GENIOMHE Mention Bioinformatique
Université Paris-Saclay – Université d'Évry val d'Essonne

Laboratoire de Mathématiques et Modélisation d'Évry,
Statistique et Génome

university year 2023–2024
internship report

Samuel Ortion

Supervizors:
Carène Rizzon
Franck Samson

University tutor:
Nathalie Boudet

1 Acknowledgement

I would like to thank my supervisors, Mrs Carène Rizzon and Mr Franck Samson for their support during the internship. Their expertises in comparative genomics and technical tools used in my internship proved to be especially useful throughout my work.

I would like to express my gratitude to every members of the Laboratoire de Mathématiques et Modélisation d'Évry for their warm welcome, and for the numerous discussion we had in lunch time.

In particular, I'm grateful for the warm welcome I received from my office neighbors, with whom we were able to discuss a number of subjects related to our interests and research.

Finally, I would like to thank my reviewer, Mrs Nathalie Boudet, for her proofreading.

Contents

List of Figures

List of Tables

List of Listings

Glossary

allopolyploidisation Polyploidisation with genetic material coming from a diverged species 7

autopolyploidisation Polyploidisation within the same species 7

orthologues Homologous genes whose divergence started at a speciation event 14

polyploidisation Mechanism leading to the acquisition of at least three versions of the same original genome in a species 7

polyspermy Fertilization of an egg by more than one sperm 7

pseudogene A gene-like sequence that lost its capacity to transcribe 9

retroduplication Duplication of a gene through retro-transcription of its RNA transcript 9

segment duplication DNA sequences present in multiple locations within a genome that share high level of sequence identity 9

subfunctionalization Fate of a duplicate gene which gets a part of the original gene function, the function being shared among multiple duplicates 10

Abbreviations

CLI Command Line Interface 25

DAG Directed Acyclic Graph 16, 18, 26

GFF General Features Format 13

GFF3 General Features Format version 3 25

HPC High Performance Computing 23

MCL Markov Clustering 19

TAG Tandemly Arrayed Genes 7, 10, 11, 14, 15, 18–20, 22

WGD Whole Genome Duplication 7

XML eXtensible Markup Language 15

2 Introduction

2.1 General notions on duplicate genes

Duplicate genes represent an important fraction of Eukaryotic genes: It is estimated that between 46% and 65.5% of human genes could be considered as duplicates. The estimate varies strongly depending on the criteria in use, because ancient duplication event may be hard to detect [5].

Back in the 70's, in his book *Evolution by Gene Duplication*, Susumu Ohno describes how gene duplication is involved in species evolution [19].

Gene duplication increases genetic variability. It is also involved in some speciation events.

2.1.1 Gene duplication mechanisms

Multiple mechanisms may lead to gene duplication. Their effects range from the duplication of the whole genome to the duplication of a fragment of a gene.

Whole genome duplication and polyploidisation

During an event of Whole Genome Duplication (WGD), the entire set of genes present on the chromosomes is duplicated (figure 2.1 (A)). WGD can occur thanks to polyspermy or in case of a non-reduced gamete. A striking example is probably *Triticum aestivum* (wheat) which is hexaploid due to hybridisation events [10].

We distinguish two kinds of polyploidisations, based on the origin of the duplicate genome: (i) Allopolyploidisation occurs when the supplementary chromosomes come from a divergent species. This is the case for the *Triticum aestivum* hybridisation, which consisted in the union of the chromosome set of a *Triticum* species with that of an *Aegilops* species. (ii) Autopolyploidisation consists in the hybridisation or duplication of the whole genome within the same species.

Unequal crossing-over

Another source of gene duplication relies on unequal crossing-over. During cell division, a crossing-over occurs when two chromatids exchange a fragment of chromosome. If the cleavage of the two chromatids occurs at different positions, the shared fragments may have different lengths. Homologous recombination of such uneven crossing-over leads to the incorporation of a duplicate region, as depicted in figure 2.1 (B, C). This mechanism leads to the duplication of the whole set of genes present in the fragment. These duplicate genes locate one set after the other: we call them Tandemly Arrayed Genes (TAG).

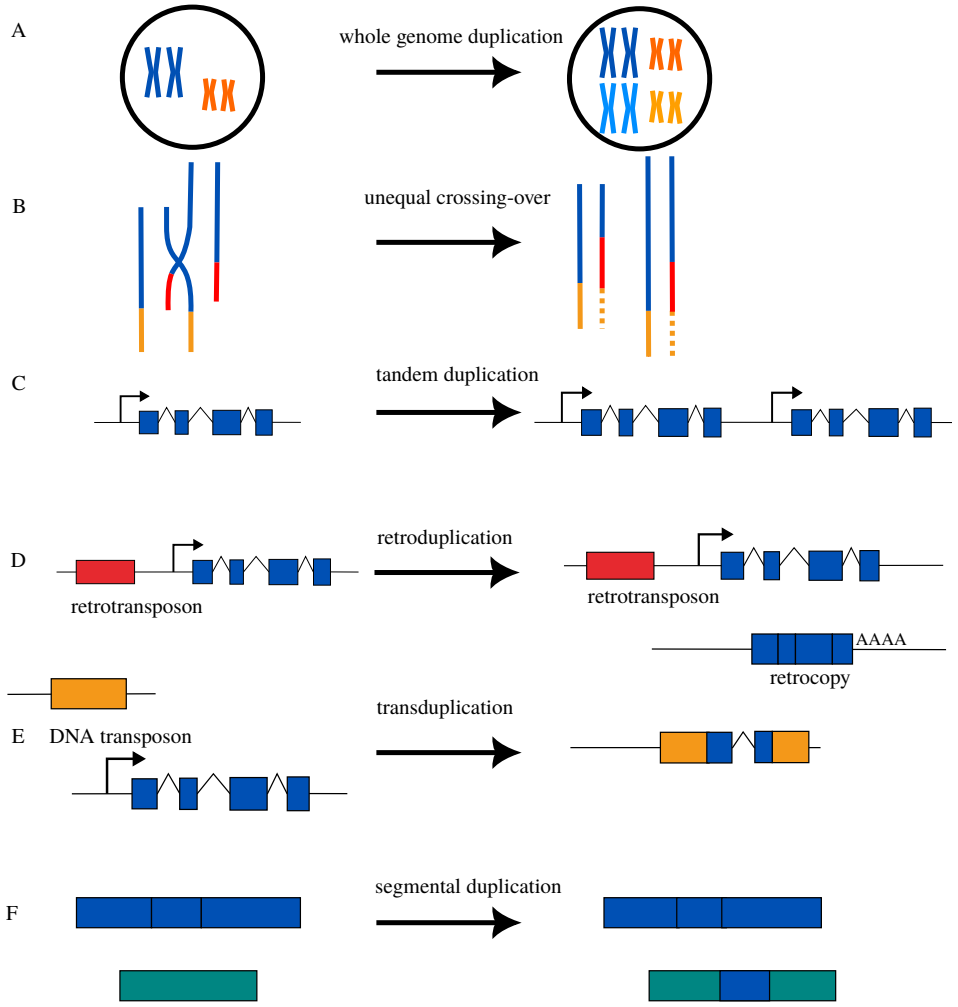


Figure 2.1.: Different types of duplications. (A) Whole genome duplication. (B) An unequal crossing-over leads to a duplication of a fragment of a chromosome. (C) In tandem duplication, two (set of) genes are duplicated one after the other. (D) Retrotransposons enable retroduplication: a RNA transcript is reverse transcribed and inserted back without introns and with a polyA tail in the genome. (E) A DNA transposons can acquire a fragment of a gene. (F) Segmental duplication corresponds to long stretches of duplicated sequences with high identity. (Adapted from Lallemand et al. [16] (fig. 1)).

Retroduplication

Transposable elements play a major role in genome plasticity, and enable gene duplication too. Retrotransposons, or RNA transposons are one type of transposable elements. They share similar sequences structures and replication mechanisms with retroviruses. Retrotransposons replicate in the genome through a mechanism known as "copy-and-paste". These transposons typically contain a reverse transcriptase gene. This enzyme proceeds in the reverse transcription of an mRNA transcript into its reverse complementary DNA sequence which can insert elsewhere in the genome. More generally, retroduplication refers to the duplication of a sequence through reverse transcription of a RNA transcript. Genes duplicated through retroduplication lose their intronic sequences and bring a polyA tail in their new locus (figure 2.1 (D)).

Transduplication

DNA transposons are another kind of transposable elements whose transposition mechanism can also lead to gene duplication. This type of transposable element moves in the genome through a mechanism known as "cut-and-paste". A typical DNA transposon contains a transposase gene. This enzyme recognizes two sites surrounding the donor transposon sequence in the chromosome resulting in a DNA cleavage and an excision of the transposon. Then, the transposase can insert the transposon at a new genome position. A transposon may bring a fragment of a gene during its transposition in the new locus (figure 2.1 (E)), leading to the duplication of this fragment.

Segment duplication

Finally, segment duplications, also called *low copy repeats*, are long stretches of DNA with high identity score (figure 2.1 (F)). Their exact duplication mechanism remains unclear [16]. They may come from an accidental replication distinct from an uneven cross-over or from a double-stranded breakage. Transposable elements may well be involved in the mechanism, as a high enrichment of transposable elements has been found next to duplicate segment extremities in *Drosophila* [16].

2.1.2 Fate of duplicate genes in genome evolution

Back in the 70s, in his book *Evolution by Gene Duplication*, Susumu Ohno proposed that gene duplication plays a major role in species evolution [19]. Indeed, gene duplication provides new genetic materials to build on a new phenotype while keeping a backup gene for the previous function. Duplicate genes evolve after duplication: they may be inactivated, and become pseudogenes, they may be deleted or conserved. If they are conserved, they may or may not acquire a new function. Figure 2.2 depicts the different possible fates of a duplicate gene.

Pseudogenization

As genome evolves, one duplicate gene may be inactivated and become pseudogene. This pseudogene keeps a gene-like structure which degrades as and when further genome modifications occur. Pseudogenes are no longer expressed.

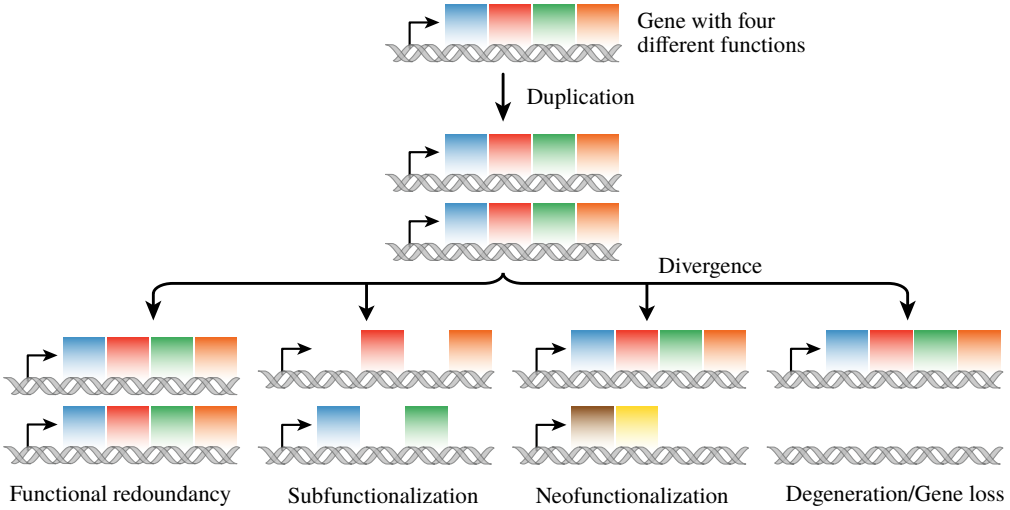


Figure 2.2.: Fate of duplicate genes. An original gene with four functions is duplicated. Its two copies may both keep the original functions (functional redoundancy). The original functions may split between the different copies (subfunctionalization). One of the copies may acquire a new function (neofunctionalization). It may also degenerate and lose its original functions (pseudogenization). (Adapted from Smedlib, CC BY-SA 4.0, via Wikimedia Commons.)

Neofunctionalization

After duplication, the new gene copy may gain a new function.

Subfunctionalization

Two duplicate genes with the same original function may encounter a subfunctionalization: each gene conserves only one part of the original function.

Functional redoundancy

Another possibility is that the two gene copies keep the ancestral function, resulting in a functional redoundancy. In this case the quantity of gene product may increase.

The study of TAGs raises particular interest because they are overrepresented among duplicate genes. They play a role in species evolution in reaction to stress. In this period of climate change it is important to understand the mechanisms that enables plants to adapt to a changing environment [7]. It could also enable the selection of plants of agronomic interest [6].

2.2 Existing tools

Multiple tools targeting duplicate gene detection have been made available on the Internet, but a lot of them are not open source, and do not last after their home laboratory abandoned the maintenance of these web services. [15] reviews these tools for each kind of duplicate genes.

Among these tools, McScanX targets the detection of tandemly arrayed genes [24]. However, this tool does not construct gene families.

FTAG Finder is a tool developed in the LaMME laboratory that aims to detect gene families and extract TAGs.

The aim of FTAG-Finder is to provide an open-source tool for fast analysis of families and tandemly arrayed genes on a single species proteome genome, providing multiple parameter choices to the user of the pipeline.

2.3 Objectives for the internship

My internship focuses on FTAG Finder.

2.3.1 Scientific questions

The underlying question of FTAG-Finder is the study of the evolutionary fate of duplicate genes in Eukaryotes. The tool enables the detection of duplicate genes in the genome of a single species, and more specifically the detection of Tandemly Arrayed Genes. For instance, the gene lists obtained from FTAG Finder could be used to answer questions such as "What is the impact of gene duplication and TAG on species evolution?", "Do genes belonging to a given TAG share the same orientation?". We might also get interested in the particular functions that genes belonging to a TAG preferentially perform.

2.3.2 Extend the existing FTAG-Finder Galaxy pipeline

Galaxy is a web-based platform for running accessible data analysis pipelines. It was first designed for use in genomics data analysis [9]. Initially created in 2016, by Bérengère Bouillon, the Galaxy integration of the FTAG-Finder pipeline received multiple successive modifications by GENIOMHE interns, under the supervision of Carène Rizzon and Franck Samson [2, 11, 18, 4]. The legacy FTAG-Finder Galaxy versions are currently deployed on the server of the *Laboratoire de Mathématiques et Modélisation d'Évry*[1].

One objective of my internship is to pursue this work to further enhance the Galaxy version of FTAG-Finder, and to include new features. Among these feature, a particular emphasis will be placed on adding (or fixing) tools to generate lists of gene pairs, building on top of work done by predecessors [13].

[1] The galaxy instance is available through `http://stat.genopole.cnrs.fr/galaxy`

2.3.3 Port FTAG-Finder pipeline on Snakemake

Another objective of my internship is to port FTAG-Finder on a workflow manager better suited to larger and more reproducible analysis.

We decided to use Snakemake.

Snakemake is a python-powered workflow manager based on rules *à la* GNU Make [14].

3 Material and methods

3.1 Data sources

I ran the FTAG-Finder pipeline on the proteome of *Arabidopsis thaliana*, *Drosophila melanogaster* and *Cyanistes caeruleus*.

- `TAIR10` release of *Arabidopsis thaliana* proteome and genomics features annotations, via Ensembl: `https://plants.ensembl.org/Arabidopsis_thaliana/Info/Index`
  - `Arabidopsis_thaliana.TAIR10.pep.all.fa.gz` – protein sequences in compressed FASTA format
  - `Arabidopsis_thaliana.TAIR10.59.gff3.gz` – genomic features annotation in compressed General Features Format (GFF) format
- Dmel release `r6.57` of *Drosophila melanogaster* proteome and genomics features annotations, via Flybase: `http://ftp.flybase.net/releases/FB2024_02/dmel_r6.57/fasta/`
  - `dmel-all-translation-r6.57.fasta.gz` – protein sequences in compressed FASTA format
  - `dmel-all-r6.57.gff.gz` – genomic features annotation in compressed GFF format
- `cyaCae2` release of *Cyanistes caeruleus* proteome and genomic features annotations, via Ensembl: `https://www.ensembl.org/Cyanistes_caeruleus/Info/Index`
  - `Cyanistes_caeruleus.cyaCae2.pep.all.fa.gz` – protein sequences in compressed FASTA format
  - `Cyanistes_caeruleus.cyaCae2.111.gff3.gz` – genomic features annotation in compressed GFF format

3.2 Methods

3.2.1 The duplicate gene detection method used in FTAG Finder

Different methods exist to detect duplicate genes. These methods depend on the type of duplicate genes they target and vary on computation burden as well as in the ease of use (for a review, see Lallemand et al. [16]).

The common step of all these method is the detection of paralog genes.

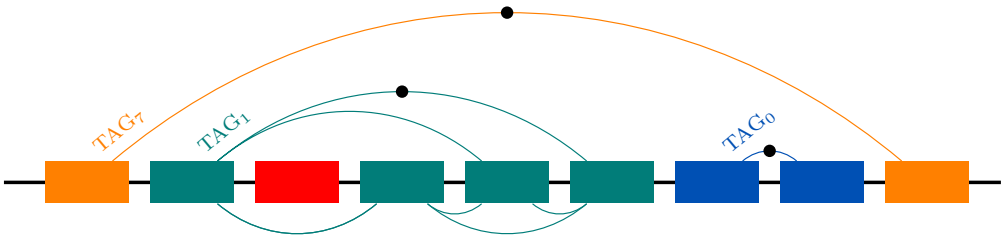


Figure 3.1.: Schematic representation of TAG definitions. Several genes are represented on a linear chromosome. The red box represent a singleton gene. Orange boxes represent a TAG with two duplicate genes seperated by 7 other genes (TAG_7). Four green boxes constitute a TAG, two genes are, at most, separated by one gene that do not belong to the same family (TAG_1). The two blue boxes represents a TAG with two genes next to each other (there is no spacer: TAG_0). The curved edges represent the homology links between each pair of genes within a TAG.

Paralog detection

Paralogs are homologous genes derived from a duplication event. We can identify them as homologous genes coming from the same genome, or as homologous genes between different species once we filtered out orthologues (homologous genes derived from a speciation event).

We can use two gene characteristics to assess the homology between two genes: gene structure or sequence similarity. The sequence similarity can be tested with a sequence alignment tool, such as `BLASTp` [1], `Psi-BLAST`, and `HMMER3` [12], or `diamond` [3]. These tools are heuristic algorithms, which means they may not provide the best results, but do so way faster than exact algorithms, such as the classical Smith and Waterman algorithm [22] or its optimized versions `PARALIGN` [21] or `SWIMM`.

The pipeline proceeds in three steps. First, it estimates the homology links between each pair of genes. Then, it deduces the gene families. Finally, it searches for TAG, relying on the position of genes belonging to the same family.

Estimation of homology links between genes

This step consists of establishing a homology relationship between each gene in the proteome.

In this step, FTAG-Finder uses `BLAST` (Basic Local Alignment Search Tool) [1] with an "all against all" search on the proteome.

Several `BLASTp` metrics can be used as a homology measure, such as bitscore, identity percentage, E-value or a variation on these. The choice of metrics can affect the results of graph clustering in the following step, and we should therefore choose them carefully [8].

The user may choose whether to keep a single protein isoform, or to keep all the isoforms in the homology graph. When the user keeps multiple isoforms before the graph clustering step, the isoforms are filtered after, to keep only one family per gene in the graph partition. This

possibility allows to detect links between large and small isoforms in the same search. The results are different depending on the choice of the user.

Identification of gene families

Based on the homology links between each pair of genes, we construct an undirected weighted graph whose vertices correspond to genes and edges to predicted homology links between them. We apply a graph clustering algorithm on the gene similarity graph in order to infer the gene families corresponding to densely connected communities of vertices. FTAG-Finder proposes three alternative algorithms for the graph clustering step: single linkage, Markov Clustering [23] or Walktrap [20]. The clustering step allows to predict more homology links: we assume that genes belonging to the same cluster all share homology links.

Detection of TAG

The final step of FTAG-Finder consists in the identification of TAG based on the gene families and the positions of genes on the chromosomes. For a given chromosome, the tool seeks genes belonging to the same family and located close to each other. The tool allows a maximal number of unrelatted genes between the homologous genes. Figure 3.1 is a schematic representation of some possible TAG positioning on a genome associated with their definition in this FTAG-Finder step. A gene belongs to a TAG of definition "TAG$_n$" if no more than n spacer (i.e., genes that do not belong to the same family) are located between them, as depicted in Figure 3.1.

3.3 Tools

3.3.1 Galaxy

Galaxy is a workflow manager. It was initially designed to provide an accessible interface to genomics analysis. It consists mostly in a web application offering wrappers around command line tools, associated with a job scheduler running a queue of jobs on the machine hosting the Galaxy instance. Listing 1 outlines a sample of a Galaxy tool definition eXtensible Markup Language (XML) file for running `blastp` on a proteome in FASTA format. Similarly to a Snakemake rule, a Galaxy tool XML file contains a section for the command to call the wrapped tool (`<command>`). It also contains a section that describes the output files (`<outputs>`). Perhaps more importantly, a section `<inputs>` defines the input files and possible offered tool options. The `<inputs>` section will be rendered as a HTML form in the Galaxy web interface.

Contrary to Snakemake, Galaxy does not infer the succession of jobs based on the input files of the tool XML definition file. This role is devoted to the human user, which might be a source of errors, even though Galaxy offers multiple features to help users to ensure the traceabillity of the files produced by the tools. This is arguably a potential shortcoming in an ideal of reproducible science, which might let technically skilled computer users to prefer the approaches proposed by alternatives such as Snakemake or Nextflow (particularly through initiatives like nf-core), which provide useful tool sets to enable bioinformaticians to shrink the human error factors in their data analysis workflows.

3.3.2 Snakemake

Snakemake is a workflow manager first released in 2012. The user defines rules in a domain specific language built on top of Python. These rules specify the required inputs and the expected outputs as well as the process that transforms the data. Based on such rules, Snakemake constructs a Directed Acyclic Graph (DAG) of jobs from the initial inputs to the desired processed outputs. Listing 2 depicts a typical Snakemake rule. In this example we focused on a rule to run `blastp` on a protein FASTA file.

```
<tool id="toy_example" name="Simple blastp" version="0.1.0">
    <description>Run blastp</description>
    <command>
    blastp -query $query -db ./blastdb -out $output -outfmt 6 -matrix
    ↪ $matrix
    </command>
    <inputs>
        <param format="fasta" name="query" type="data" label="Protein
        ↪ query sequences"/>
        <param type="text" name="matrix" value="BLOSUM62"/> [...]
    </inputs>
    <outputs>
        <data format="tabular" name="output" type="data"
        ↪ label="${query.name}.blastp.tsv"/>
    </outputs>
</tool>
```

Listing 1: A simplified example of a Galaxy tool wrapper to run `blastp`

```
rule blastp:
    input:
        query="{name}.fasta",
        db=[...]
    output:
        tsv="{name}.blastp.tsv"
    params:
        db="db/{name}",
        matrix="BLOSUM62"
    shell:
        """blastp -query "{input.query}" -db "{params.db}" \
            -outfmt "{params.format}" -out "{output.tsv}" \
            -matrix {params.matrix}"""
```

Listing 2: A simplified example of a Snakemake rule to run `blastp`

4 Results

4.1 Update of the Galaxy tools of FTAG-Finder

The legacy FTAG-Finder scripts were working on an old version of Galaxy, but no longer on the newest Galaxy platform (version 24).

I updated the Python scripts and the Galaxy tool definitions XML files to make them compatible with the Galaxy platform (version 24).

A screenshot of the Galaxy user interface is provided in Figure A.2 showing the Find-Homolog tool web form.

The updated scripts and tool definitions are available in the `galaxy` branch of the git repository (`https://gitlab.com/sortion/FTAG-Finder`).

To prove the proper behaviour of the Galaxy tools, I used the `planemo`[1] tool and wrote automatic test cases. These tests are written in the XML definition files. `planemo` is a tool dedicated to Galaxy tool development.

I also installed a Galaxy instance on a QEMU/KVM Debian 12 virtual machine, to be able to test the installation of Galaxy v24 on my computer. This galaxy version was not installed on the laboratory server. The installation of the virtual machine allowed me to test the v24 without waiting for an installation on the server of the laboratory.

4.2 Snakemake development and first tests

I modified the python scripts interfacing with Galaxy to integrate them into a SnakeMake powered pipeline. Then, I used this pipeline to detect the TAG of *Arabidopsis thaliana*, the Thale cress and *Cyanistes caeruleus*, the Blue Tit.

The code of the Snakemake port of FTAG-Finder is available at `https://gitlab.com/sortion/FTAG-Finder`. A more detailed listing of the modifications made on the pipeline is presented in Section A.1.

Snakefile is the name of the file containing multiple Snakemake rules, leading to a pipeline definition. An example of DAG built by Snakemake for the FTAG-Finder pipeline is depicted in Figure A.1.

The user defines the parameters of the pipeline in a YAML file, including the name of the run.

The user may launch the pipeline using the standard Snakemake way, specifying the YAML config file:

```
$ snakemake --cores all --snakefile=Snakefile
↪  --configfile=config.yaml
```

[1] https://planemo.readthedocs.io/en/latest/

The pipeline outputs multiple lists of genes to be used in subsequent statistical analysis on TAGs:

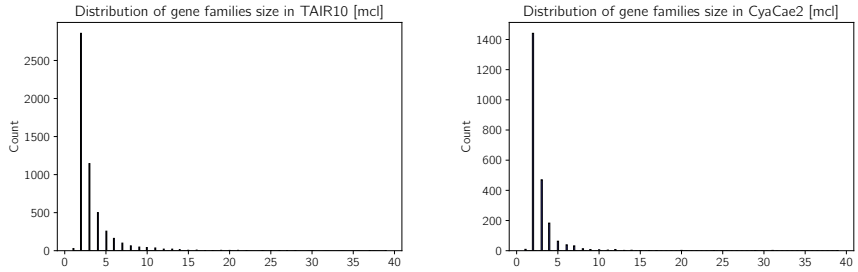
In the following, let us note the name of the run “”, we will note the name of the files accordingly.

- A two-columns file associating each gene to its family identifier: `results/<name>.<clustering method>.tsv`
- A set of random pairs: `results/lists/<name>/random_gene_pairs.tsv`;
- All pair of genes belonging to the same family: `results/lists/<name>/intra_family_pairs.tsv`;
- All pair of genes separated by at most n genes, regardless of the family they belong to: `results/lists/<name>/successive_gene_pairs.tsv`; and finally
- Pairs of genes belonging to the same TAG: `results/lists/<name>/local_gene_pairs.tsv`.

These outputs enable us to perform multiple statistical tests on TAG.

From the gene to family mapping file, I plotted the distribution of family size for two distant Eukaryotes, the Thale cress plant and the Blue Tit in Figure 4.1 to check that the result matches the expectation.

The selected parameters values for *Cyanistes caeruleus* are defined in Listing 3. The corresponding YAML config file is available at https://gitlab.com/sortion/lamme2024/-/blob/main/exp/mutual/config/20240527_CyaCae2.yaml?ref_type=heads.



(a) *Arabidopsis thaliana* (b) *Cyanistes caeruleus*

Figure 4.1.: Distribution of gene family size. The x-axis corresponds to the number of genes per family, and the y-axis corresponds to the number of family falling into this category, in *Arabidopsis thaliana* (a) and *Cyanistes caeruleus* (b). The gene families are inferred with a Markov Clustering (MCL) of the similarity graph of the proteins. The homology links used in this similarity graph have been estimated using the bitscore of the local alignment using blastp on the longest protein isoform.

Species	Families	Duplicate genes
Arabidopsis thaliana	5379	17784
Cyanistes caeruleus	2282	6492

Table 4.1.: Predicted families and gene counts for *Arabidopsis thaliana* and *Cyanistes caeruleus*

(a) *Arabidopsis thaliana* (b) *Cyanistes caeruleus*

Figure 4.2.: Photos of two relatively distant Eukaryotes. (a) A Thale cress, *Arabidopsis thaliana*, in the street at Clermont-Ferrand, France (CC-By-SA Fabrice Rubio via PlantNet[2]). (b) An Eurasian Blue Tit, *Cyanistes caeruleus*, foraging for insects in bulrushes at Courcouronnes, France (own work).

4.3 Biological analyses

4.3.1 Gene orientation concordance

Based on the TAG listed by the FTAG-Finder pipeline, we wanted to check wether the genes belonging to the same TAG shared the same orientation. We distinguish three cases: the orientation of two genes is considered as coherent, when they belong to the same strand, divergent when they head to the opposite strand, and convergent otherwise. Figure 4.3 depicts the three cases.

To test wether a concordance is privileged we perform a Fisher exact test on the contingency table reported in Table 4.2:

Let us denote by PST the pairs of successive genes belonging to a TAG and PSnT the pair of successive genes that does not belong to a TAG, following the convention initiated by [17].

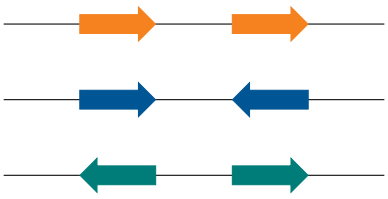


Figure 4.3.: Schematic representation of the three cases we consider for the orientations coherence test: coherent orientation in orange, convergent orientation in blue and divergent orientation in green

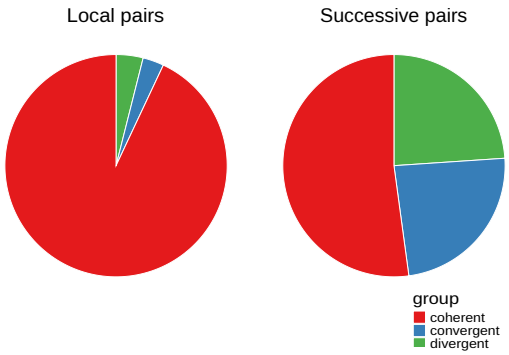


Figure 4.4.: Concordance of DNA strand of pairs of genes within the same family or not in TAIR10. Coherent genes are genes located on the same strand. Convergent and divergent gene pairs are located on different DNA strands: they are facing either towards the other or at the opposite direction. The proportion of concordance state is shown for two set of pair of genes located one next to the others with no spacer between them on the same chromosome and that are either the member of the same family (Local pairs) or not (Successive pairs).

	coherent	convergent	divergent
PST	1516	50	64
PSnT	12498	4913	4916

Table 4.2.: Orientation concordance Fisher exact test contingency table on TAIR10.

The hypotheses involved in the test are the following:

$$\begin{cases} (H_0) & \text{the proportion of each orientation concordance is similar in both groups} \\ (H_1) & \text{the proportion is different between PST and PSnT} \end{cases}$$

The p-value of the Fisher test is below $2.2 \cdot 10^{-16}$, so at level $\alpha = 0.05$, we reject (H_0): the proportion of convergent genes in TAG or not TAG differs. Moreover, the orientation of genes within a TAG is most often coherent.

We have found a result similar to [17] from which we borrowed the idea of the analysis.

We may interpret this result as follows: it looks like there is a selective pressure on the orientation of the genes, that keeps the copies of the genes within a TAG in the same orientation. This is maybe to share a similar gene regulation system.

5 Discussion and perspectives

The Snakemake version allows to run the whole pipeline using a single command line, without requiring user interaction between each step. It allows to avoid human mistakes. It enables the results of the workflow to be reproducible, provided the config file is shared. Contrary to Galaxy, Snakemake can be run on the user own computer or on a High Performance Computing (HPC) cluster. This allows to obtain results faster on a larger data input.

The first tests demonstrates that my Snakemake version of FTAG-Finder is working.

To confirm the correct behavior of my development, we could make a comparison of results obtained with Snakemake and those performed by Galaxy.

It would be required also to test the proper functionality of my work on the compute machines of the LaMME laboratory.

A change of version of the `igraph` R package rendered the Walktrap clustering method unusable in newer R versions. I tried to update the method names to the newer interface, but the clustering obtained does not fit the former ones. Further debugging would be needed in this area.

An evaluation of the performance of the tool might be interesting.

Based on the different lists of genes generated by this version of FTAG Finder, we could perform more statistical tests on TAG. For instance, we could perform an analysis of overrepresentation of particular Gene Ontology terms for genes belonging to a TAG as compared with the other genes.

We could also perform an analysis on the age of the duplication of the genes of the TAGs by doing an analysis of the Ka/Ks ratio. The Ka/Ks ratio corresponds to the number of nonsynonymous substitutions per non synonymous site (K_a) over the number of synonymous substitutions per synonymous site (K_s).

The Thale cress lives in very diverse habitat. Duplicate genes offers it a range of possibility to evolve and develop resistance to the sometimes polluted environment. Gene duplication enables the Thale cress to adapt more rapidly to changes in its environment and they often contains genes involved in the response to stress.

6 Conclusion

During my 2-months internship, I updated the Python scripts and the tool definitions so that they become compatible with the latest Galaxy version (v24) I successfully ported the Galaxy version of FTAG-Finder on Snakemake. I ran this pipeline on the IFB compute cluster on the proteome of *Arabidopsis thaliana*, *Drosophila melanogaster*, and *Cyanistes caeruleus*.

Using the list of genes outputted by the FTAG-Finder pipeline, I performed a Fisher exact test to check wether the TAG genes shared the same orientation.

Throughout this internship I extended my knowledge on Snakemake. I discovered the Galaxy tool.

During my internship, I developed a stronger knowledge on duplicate genes and on the method used in their analysis.

I enjoyed the multiple seminar that took place in the LaMME laboratory, which offered me an overview on diverse research topics.

References

[1] Stephen F. Altschul et al. “Basic Local Alignment Search Tool”. In: *Journal of Molecular Biology* 215.3 (Oct. 5, 1990), pp. 403–410. ISSN: 0022-2836. DOI: 10.1016/S0022-2836(05)80360-2. URL: https://www.sciencedirect.com/science/article/pii/S0022283605803602 (visited on 04/30/2023).

[2] Bérengère Bouillon. *Development and Tools Integration on the Bioinformatics Platform Galaxy for Gene Families and TAGs Determination*. Internship Report. Laboratoire de Mathématiques et Modélisation d’Évry, 2016.

[3] Benjamin Buchfink, Klaus Reuter, and Hajk-Georg Drost. “Sensitive Protein Alignments at Tree-of-Life Scale Using DIAMOND”. In: *Nature Methods* 18.4 (Apr. 2021), pp. 366–368. ISSN: 1548-7105. DOI: 10.1038/s41592-021-01101-x. URL: https://www.nature.com/articles/s41592-021-01101-x (visited on 03/28/2024).

[4] Séanna Charles. *Finalisation du pipeline FTAG (Families and TAG) Finder, un outil de détection des gènes dupliqués sous Galaxy*. Internship Report. Laboratoire de Mathématiques et Modélisation d’Évry, 2023.

[5] Margot Correa et al. “The Transposable Element Environment of Human Genes Differs According to Their Duplication Status and Essentiality”. In: *Genome Biology and Evolution* 13.5 (May 7, 2021). Ed. by Ellen Pritham, evab062. ISSN: 1759-6653. DOI: 10.1093/gbe/evab062. URL: https://academic.oup.com/gbe/article/doi/10.1093/gbe/evab062/6273345 (visited on 03/19/2024).

[6] *Dupliquer pour s’adapter ou comment accélérer l’évolution des plantes ? | CNRS Biologie*. Oct. 14, 2020. URL: https://www.insb.cnrs.fr/fr/cnrsinfo/dupliquer-pour-sadapter-ou-comment-accelerer-levolution-des-plantes (visited on 09/07/2024).

[7] Carine Géry and Evelyne Téoulé. “Cold Acclimation Diversity in Arabidopsis Thaliana: CRISPR/Cas9 as a Tool to Fine Analysis of Tandem Gene Arrays, Application to CBF Genes”. In: *Development Genes and Evolution* 232.5 (Dec. 1, 2022), pp. 147–154. ISSN: 1432-041X. DOI: 10.1007/s00427-022-00693-4. URL: https://doi.org/10.1007/s00427-022-00693-4 (visited on 09/07/2024).

[8] Theodore R. Gibbons et al. “Evaluation of BLAST-based Edge-Weighting Metrics Used for Homology Inference with the Markov Clustering Algorithm”. In: *BMC Bioinformatics* 16.1 (Dec. 2015), p. 218. ISSN: 1471-2105. DOI: 10.1186/s12859-015-0625-x. URL: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-015-0625-x (visited on 03/19/2024).

[9] Jeremy Goecks et al. “Galaxy: A Comprehensive Approach for Supporting Accessible, Reproducible, and Transparent Computational Research in the Life Sciences”. In: *Genome Biology* 11.8 (2010), R86. ISSN: 1474-760X. DOI: 10.1186/gb-2010-11-8-r86. pmid: 20738864.

[10] K. A. Golovnina et al. “Molecular Phylogeny of the Genus Triticum L”. In: *Plant Systematics and Evolution* 264.3 (Apr. 1, 2007), pp. 195–216. ISSN: 1615-6110. DOI: 10.1007/s00606-006-0478-x. URL: https://doi.org/10.1007/s00606-006-0478-x (visited on 03/27/2024).

[11] Fabien Jasmin. *Study of Tandemly Arrayed Genes Expression for Arabidopsis Thaliana*. Internship Report. Laboratoire de Mathématiques et Modélisation d’Évry, June 27, 2016.

[12] L. Steven Johnson, Sean R. Eddy, and Elon Portugaly. “Hidden Markov Model Speed Heuristic and Iterative HMM Search Procedure”. In: *BMC Bioinformatics* 11.1 (Aug. 18, 2010), p. 431. ISSN: 1471-2105. DOI: 10.1186/1471-2105-11-431. URL: https://doi.org/10.1186/1471-2105-11-431 (visited on 04/09/2024).

[13] NORMAND Kevin. “Development of CreationList, a Tool Dedicated to the Analysis of TAGs and Integration in Galaxy”. Internship defense. 2017.

[14] Johannes Köster and Sven Rahmann. “Snakemake–a Scalable Bioinformatics Workflow Engine”. In: *Bioinformatics (Oxford, England)* 28.19 (Oct. 1, 2012), pp. 2520–2522. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/bts480. pmid: 22908215.

[15] Tanguy Lallemand. “Évolution des gènes dupliqués chez le pommier : Identification et caractérisation de la dominance du sous-génome dans le génome de la pomme”. PhD thesis. Université d’Angers, Nov. 15, 2022. URL: https://theses.hal.science/tel-04081238 (visited on 03/30/2024).

[16] Tanguy Lallemand et al. “An Overview of Duplicated Gene Detection Methods: Why the Duplication Mechanism Has to Be Accounted for in Their Choice”. In: *Genes* 11.9 (Sept. 4, 2020), p. 1046. ISSN: 2073-4425. DOI: 10.3390/genes11091046. URL: https://www.mdpi.com/2073-4425/11/9/1046 (visited on 03/19/2024).

[17] Julie Lê-Hoang. *Etude transcriptomique des gènes dupliqués en tandem (TAG) chez Arabidopsis thaliana*. Internship Report. Laboratoire de Mathématiques et Modélisation d’Évry, 2017.

[18] Kévin Normand. *Rapport V6*. Internship Report. Laboratoire de Mathématiques et Modélisation d’Évry, Apr. 18, 2017.

[19] Susumu Ohno. *Evolution by Gene Duplication*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1970. ISBN: 978-3-642-86661-6. DOI: 10.1007/978-3-642-86659-3. URL: http://link.springer.com/10.1007/978-3-642-86659-3 (visited on 03/21/2024).

[20] Pascal Pons and Matthieu Latapy. *Computing Communities in Large Networks Using Random Walks (Long Version)*. Dec. 12, 2005. DOI: 10.48550/arXiv.physics/0512106. arXiv: physics/0512106. URL: http://arxiv.org/abs/physics/0512106 (visited on 03/30/2024). Pre-published.

[21] Torbjørn Rognes. “ParAlign: A Parallel Sequence Alignment Algorithm for Rapid and Sensitive Database Searches”. In: *Nucleic Acids Research* 29.7 (Apr. 1, 2001), pp. 1647–1652. ISSN: 0305-1048. DOI: 10.1093/nar/29.7.1647. URL: https://doi.org/10.1093/nar/29.7.1647 (visited on 04/09/2024).

[22] T. F. Smith and M. S. Waterman. “Identification of Common Molecular Subsequences”. In: *Journal of Molecular Biology* 147.1 (Mar. 25, 1981), pp. 195–197. ISSN: 0022-2836. DOI: 10.1016/0022-2836(81)90087-5. URL: https://www.sciencedirect.com/science/article/pii/0022283681900875 (visited on 04/29/2023).

[23] S. van Dongen. “A New Cluster Algorithm for Graphs”. In: R 9814 (Jan. 1, 1998). URL: https://ir.cwi.nl/pub/4604 (visited on 03/22/2024).

[24] Y. Wang et al. “MCScanX: A Toolkit for Detection and Evolutionary Analysis of Gene Synteny and Collinearity”. In: *Nucleic Acids Research* 40.7 (Apr. 1, 2012), e49–e49. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkr1293. URL: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkr1293 (visited on 05/06/2024).

A Annex

A.1 Contributions to FTAG-Finder

FTAG-Finder on Snakemake

I wrote the Snakemake version of FTAG-Finder. Doing so, I refactored a little bit the python scripts originally written for Galaxy by predecessor interns. These refactoring consisted mostly in offering a more versatile command line interface.

Additional scripts

The version of FTAG-Finder I inherited from predecessor interns required some input files to be manually crafted. I wrote several scripts that allows the Snakemake version of the pipeline to run from the released proteome file in fasta format and the feature annotations in General Features Format version 3 (GFF3) format down to the end results without manual edit.

These scripts consists in:

- A script to extract a protein identifier to gene identifier mapping from some standard FASTA header formats (i. e. TAIR, Ensembl and Flybase). This selection is obviously incomplete, but the scripts written may serve as a basis to support other headers conventions.
- A script that extracts the relevant parts of the GFF3 files containing positions of genomic features (genes, non coding RNA, chromosomes, etc.). The Find Homologs step previously required manual edit of the file.

Creation of special lists of genes

Fabien Jasmin described several possible statistical analyses that could be performed on Tandemly Arrayed Genes and special list of genes that would be required for such analyses [11]. Kévin Normand then wrote a python script that implement the creation list module and integrated this tool in the LaMME Galaxy instance [13]. Building on top of these works, I (re)wrote python scripts for the "Creation List" module, with modifications to better fit my Snakemake version of the pipeline, and my own taste. Specifically, I splitted the different creation list scripts into their own Python scripts, and I wrote a bespoke Command Line Interface (CLI) for each of them using the standard Python `argparse` module.

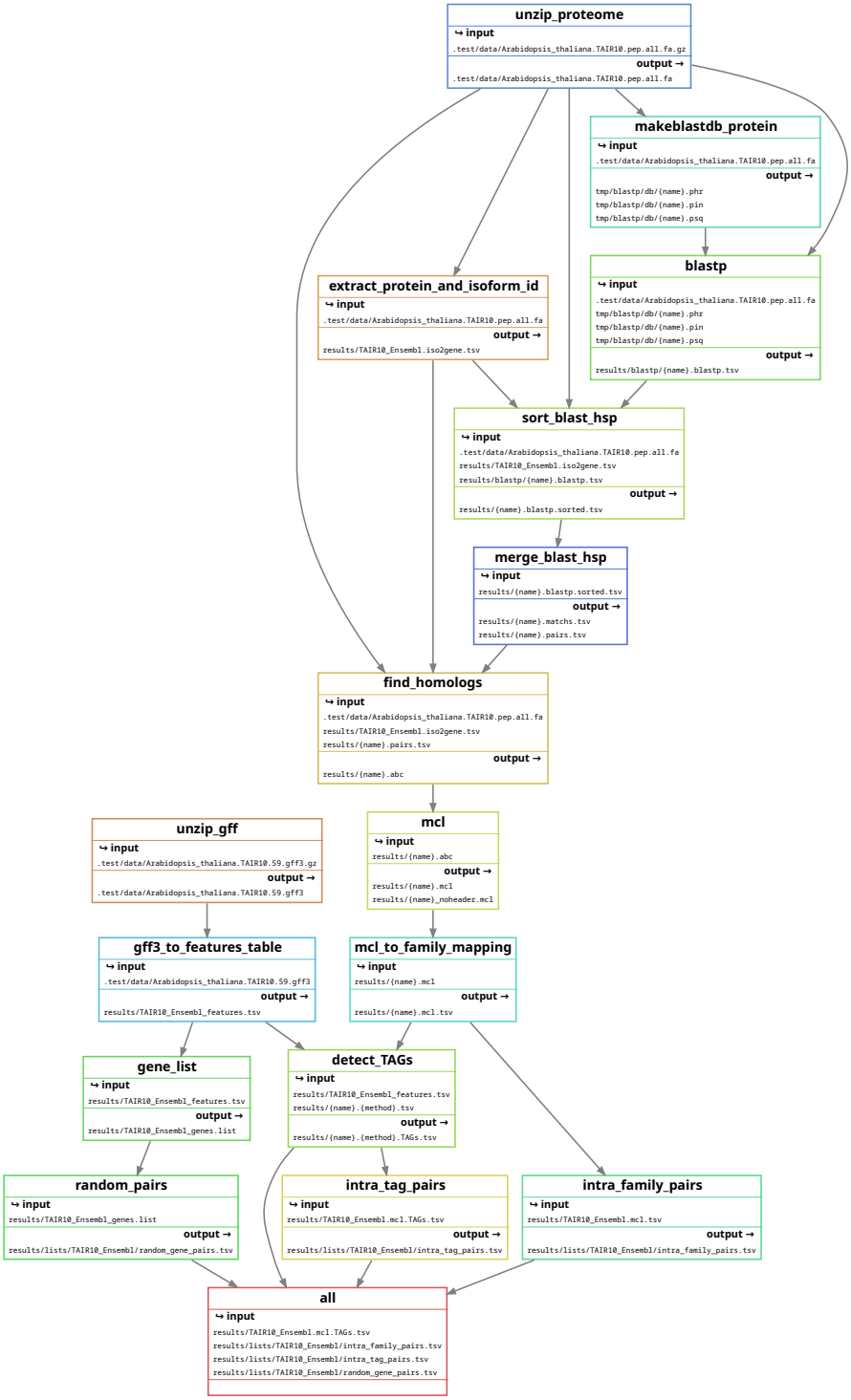


Figure A.1.: A example of DAG built by Snakemake on the FTAG-Finder pipeline. The final rule 'all' defines the desired final output files. Then, Snakemake infers the succession of jobs based on the input and output files defined in the rules. Based on this graph, the rules are either executed one after the other when a rule depends on the other or in parallel when the rules are independent.

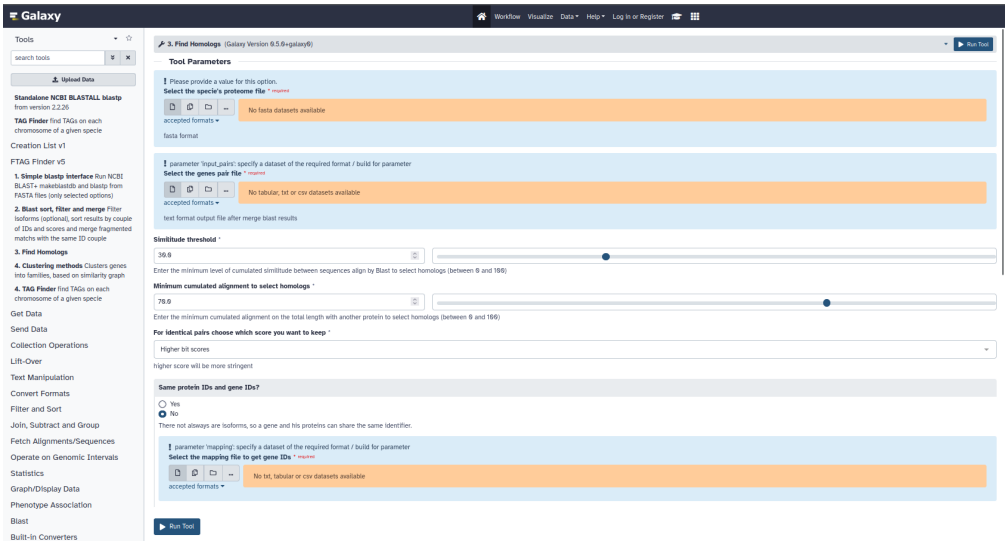


Figure A.2.: Screenshot of the Find-Homologs tool under Galaxy v24 web form

A.2 Supplementary figures

A.3 Parameters

```yaml
alignment_method: blastp # "blastp", "diamond" or "mmseqs2"
clustering_method: mcl # "sl" (single linkage), "mcl" or "Walktrap"
protein_id_is_gene_id: no
feature_id_is_gene_id: yes
geneid_proteinid_mapping: # Extract geneid-proteinid mapping from the input fasta file
  extract:
    format: Ensembl # "TAIR", "Ensembl" or "flybase"
blastp:
  evalue: 1.0
  seg: "yes" # make sure this is not simply the YAML symbol yes.
  matrix: BLOSUM62
  gapopen: 9
  gapextend: 1 # warning: this differs from the Galaxy version.
  threshold: 0.05 # warning: this differs from the Galaxy version.
  xdrop_gap: 0.0
sort_blast_hsp:
  filter_isoform: no
  isoform_alternative: "longest"
  filter_selection: "computed"
  # custom_list_filename: <file> # If filter_selection = "own"
merge_blast_hsp:
  merge: yes # TODO
  merge_amino_acid_allowed_overlap: 10 # 12 was too stringent for the test dataset
find_homologs:
  similarity_threshold: 30 # within 0 - 100
  cummulated_coverage_threshold: 70 #
  bitscore_selection: "higher" # "higher" or "lower" selected bitscores
mcl:
  inflation: 4
  pre_inflation: 3
  pre_inflation_max_bound: 3
  start_inflation: 3
  remove_loops: yes
  scheme: 4
detect_TAGs:
  definitions: [0, 1, 5, 10]
  coding: yes

```


Listing 3: Parameter values of the Snakemake workflow used to generate the families and TAG lists for *Cyanistes caeruleus* and *Arabidopsis thaliana*