Machine-learning approaches to explore functional fate of duplicate gene prediction models in a model plant

M2 bioinformatics mention Genomics Informatics and Mathematics for Health and Environment
Université Évry Paris-Saclay

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

university year 2024–2025
internship report

Samuel Ortion

Supervizors:	
Carène Rizzon	UÉVE, LaMME
Marie Szafranski	ENSIIE, LaMME
Emmanuelle Lerat	CNRS, LBBE
Franck Samson	INRAE, LaMME
University tutor:	
Farida Zehraoui	UÉVE, IBISC
Reviewer:	
Massinissa Hamidi	UÉVE, IBISC

Abstract Gene duplication is a common phenomenon all around the tree of Life. It is often the case that more than half the genes of a species can still be identified as duplicated. One particularly interesting question about these genes is related to how they diverge and acquire new functions: some duplicated genes keep the original function, whereas some other genes' functions change. During this internship we aimed to distinguish between these two cases based on genomics information on duplicated gene pairs using a machine learning approach. We first computed a set of descriptive variables based on genomics data for *Arabidopsis thaliana* and gathered ground truth labels from the literature. We performed an analysis of the descriptors and found a discrepancy between the labeled subset and the whole set of duplicated genes. Finally, we trained simple logistic regression models and identified descriptors having a predictive capacity.

Keywords Duplicate genes, functional redundancy, Arabidopsis, machine learning

Contents

List of Figures

List of Tables

List of Algorithms

Glossary

allopolyploidy Polyploidy where the supplementary genetic material comes from another species (12)

autopolyploidy Polyploidy where the supplementary genetic material comes from the same species (12)

functional redundancy Both duplicate gene keep the same function (14)

machine learning A field of study mixing computer science and statistical inference to build algorithms that can learn from data (24, *see* statistical learning)

neofunctionalization A duplicate gene acquire a new function (14)

ohnolog Gene duplicated during a whole genome duplication (*see* WGD)

ortholog Homolog gene whose divergence started at a speciation event (*see* homolog)

plasmid A circular, mobile DNA element in bacteria. (18)

polyploidization Getting three sets of chromosomes or more (12)

pseudogene Gene no longer expressed that accumulates mutations (14)

pseudogenization Loss of a gene by accumulation of mutation (14)

singleton A gene with a single copy (26)

statistical learning (24, *see* machine learning)

subfunctionalization Each duplicate gene share one part of the original duplicated gene function (14)

Abbreviations

cDNA complementary DNA (14)

GFF Gene Feature Format (i)

HMM Hidden Markov Models (21)

HPC High Performance Computing (ii)

PPI Protein-Protein Interaction (5, 22, 23, 28, 29)

SOAK Same/Other/All K-fold cross-validation (ii)

TAG Tandemly Arrayed Genes (i, 12, 36, 39)

TE Transposable Element (23, 36, 40)

WGD Whole Genome Duplication (12, 39)

Acknowledgments

I want to thank my internship supervisors for the very helpful advises and heading given to the subject.

I would like to thank the interns who joined me in the room 406 and brought to this place a bit more animation, little squabbles too, but more importantly, provided numerous subjects of conversation about each others study matter.

I would like to thank the organizers of the laboratory seminars and PhD candidates seminars, and invited speakers for sharing with us some very interesting topics; even though I have to admit that I did not understand a broad part of the mathematical aspects of them.

Finally, I would like to thank all members of the laboratory for the great ambiance there.

The LaMME and LBBE laboratories

My internship took place in the *Laboratoire de Mathématiques et Modélisation d'Évry* (LaMME). The LaMME laboratory gather researchers from the *Université Évry Paris-Saclay* (UÉVE), the *École Nationale Supérieure d'Informatique pour l'Industrie et l'Entreprise* (ENSIIE), the *Centre National de la Recherche Scientifique* (CNRS) and the *Institut National de Recherche pour l'Agriculture, l'Alimentation et l'Environement* (INRAE). Researchers are splitted in three teams: Analysis and Partial Differential Equations, Probability and Financial Mathematics and Statistics and Genome, the team I joined. This team works on statistical learning, graphical models, multiple testing, association studies, change points detection, duplicate genes and transposable elements.

I was supervized by Carène Rizzon, Marie Szafranski, and Franck Samson from the LaMME. I was also remotely supervized by Emmanuelle Lerat from the *Laboratoire de Biométrie et Biologie Évolutive* (LBBE).

The LBBE laboratory is under CNRS and *Université Lyon 1* supervision. It uses mathematical modeling and computer science for evolutionary ecology, genomics and health-related studies.

1. Introduction

1.1. What are duplicate genes?

In 1970, in his book *Evolution by Gene Duplication*, Susumu Ohno proposed that the duplication of genes could be a major motor of species evolution (Ohno 1970), by introducing new materials on which the evolution could build a greater diversity and new gene functions.

1.1.1. Gene duplication mechanisms

Several molecular mechanisms can lead to the duplication of a gene. We review them in the following sections. The duplication mechanisms are schematically represented in Figure 1.1.

Whole genome duplication and polyploidization

A Whole Genome Duplication (WGD) occurs when the entire genome is duplicated. This can be caused by an abnormal meiosis, when unreduced gametes are fertilized. In general polyploidization is the process by which a species acquires two copies of its chromosomes, or even more. Two kinds of polyploidization are differentiated depending on the origin of the supplementary set of chromosomes. In the case of autopolyploidy it comes from a cell of the same species, whereas in the case of allopolyploidy this set of chromosomes comes from two different species.

Multiple polyploidization events occurred in Eukaryotes. It occurred more often in plants than in animals or fungi (Otto et al. 2000). In human, the triploid syndrome is very often associated with a lethal phenotype in utero (*Orphanet: Triploidy Syndrome* 2025), and is thus absent in the adult population (Otto et al. 2000). Even though polyploid events are rare and often lethal, especially in animals, some of them are concomitant with wide species radiation events (Van de Peer et al. 2009).

Uneven crossing over and tandemly arrayed genes

A crossing-over occurs when two homologous chromosomes exchange a fragment of their chromatids during cell division. An uneven crossing-over, when both shared fragments do not have the same length, leads to the duplication of a contiguous region of a chromosome. This process, also known as ectopic recombination, induces the duplication of one chromosome region disposed directly after the original region, in tandem. The array of genes duplicated by this means and disposed in a cluster on the same chromosome are thus called Tandemly Arrayed Genes (TAG).

Transposable elements mediated gene duplication

Transposable elements (TEs) mobility constitute a major motor of genome reorganization. We distinguish two main types of TEs: the DNA transposons and the retrotransposons. Both kind of TEs may be a source of gene duplication.

Transduplication Autonomous DNA transposons contain a gene coding for the transposase. This enzyme may replicate the transposon via a mechanism known as “cut and paste”. In this mechanism, the transposase extracts the sequence of the transposon from the chromosome (“cut”) and introduces it back into the genome, at another locus (“paste”). The transposase may duplicate another gene while doing so.

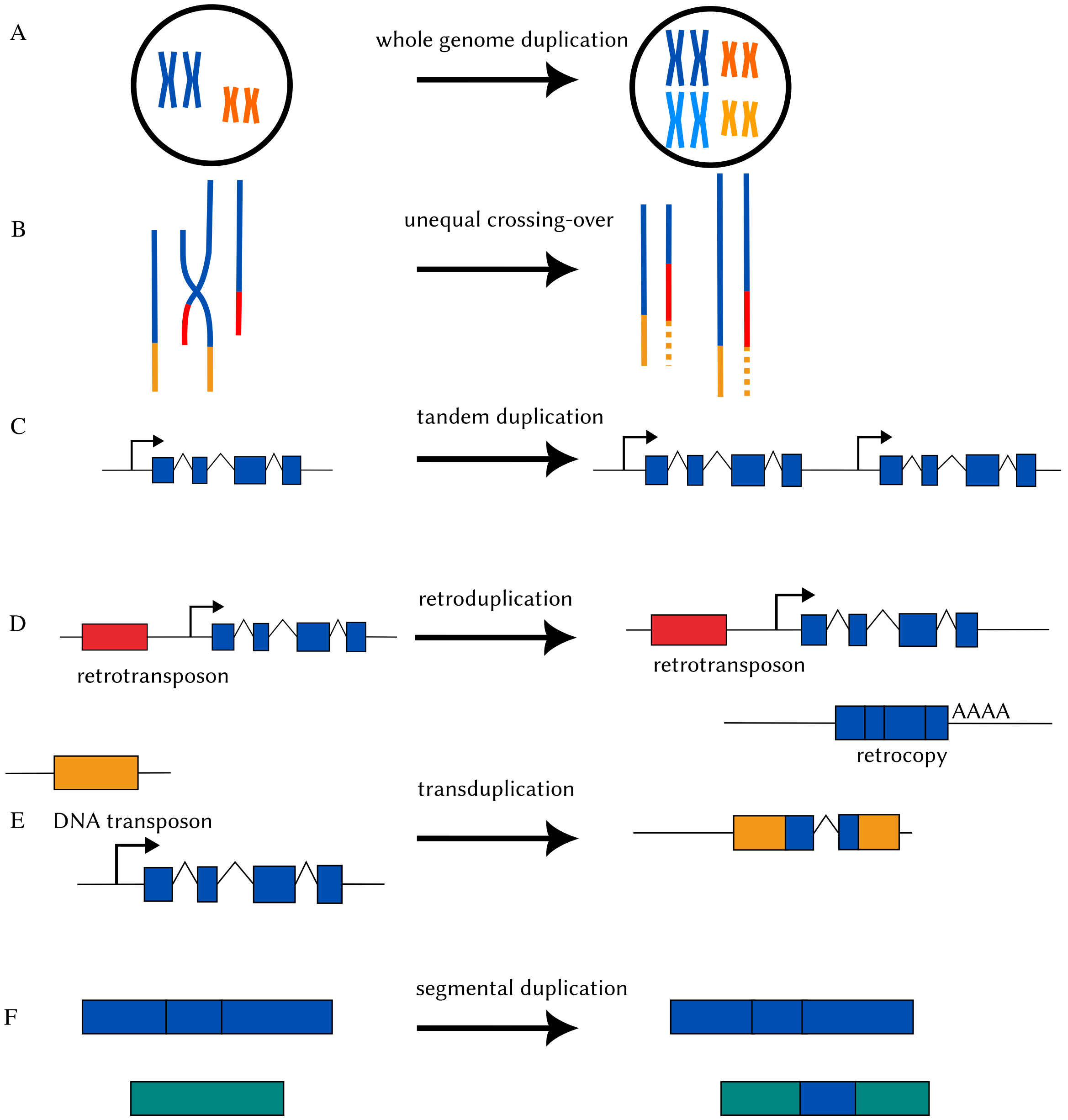


Figure 1.1.: Different types of duplications. (A) Whole genome duplication. (B) An unequal crossing-over leads to a duplication of a fragment of a chromosome. (C) In tandem duplication, two (set of) genes are duplicated one after the other. (D) Retrotransposons enable retroduplication: an RNA transcript is reverse transcribed and inserted back without introns and with a polyA tail in the genome. (E) A DNA transposon 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. (2020) (fig. 1)).

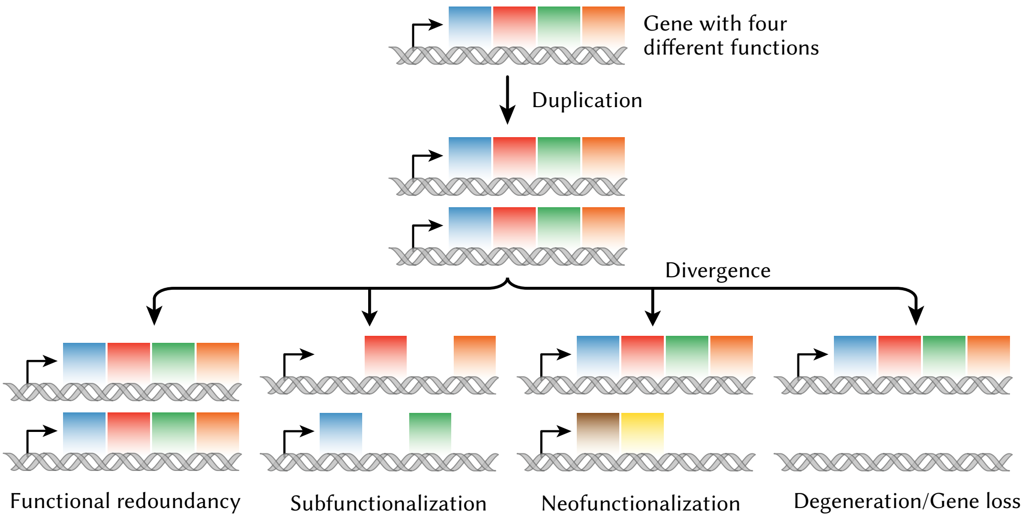


Figure 1.2.: Fate of duplicate genes. An original gene with four functions is duplicated. Its two copies may both keep the original functions (functional redundancy). The original functions may split between the different copies (subfunctionalization). One of the copies may acquire a new function (neofunctionalization). It may also degenerate and lose its original functions (pseudogenization). (Adapted from Smedlib, CC BY-SA 4.0, via Wikimedia Commons.)

Retroduplication A retrotransposon encodes a reverse transcriptase enzyme. Transcribed messenger RNA of this kind of TE may be reverse transcribed into its complementary DNA (cDNA). This cDNA may be introduced back into the genome at another locus, a process known as "copy and paste", because the original transcribed sequence remains unaltered. This process may apply to another gene transcript as well, when the reverse transcriptase acts off-target. In this case the new duplicate gene copy is flanked with the poly-A tail of its matured messenger RNA and it has lost its intronic regions, that would have been excised during messenger RNA maturation.

Segmental duplication and low copy repeats

Low copy repeats are long stretches of DNA, at least 1 kb long, with a high identity score. The mechanisms by which they were duplicated is not well deciphered, but TEs may well be involved, as, at least in the fruitfly *Drosophila melanogaster*, a lot of TEs are in the neighborhood of the segmental duplication. The fact that the low copy repeats are not adjacent on the chromosomes does not corroborate the hypothesis that segmental duplication would be due to an uneven crossing-over (Fiston-Lavier et al. 2007).

1.1.2. Evolutionary fate of duplicate genes

A duplicate gene may encounter different fates after duplication. Due to the relaxation of the selective pressure on one of the copy, one gene may accumulate deleterious mutations. These deleterious mutations can render the gene dysfunctional. It may no longer be expressed. Acquiring even more mutations, that would not be subjected to a negative selective pressure, it may become a pseudogene, and soon be indiscernible from non-coding sequence (pseudogenization). On the other hand, both duplicate genes may conserve the original function (functional redundancy). The duplicate genes may also lose one part of the original gene function. Both gene remaining functions would then be complementary and ensure the original function maintenance (subfunctionalization). Last but not least, because one of the duplicates can occupy the original function, the other may be under a less severe conservation pressure. This can enable more freedom for gene divergence. The gene may thus acquire a new function (neofunctionalization), and provide a new target for evolution.

1.2. Our objective for this internship

The aim of this internship is to explore machine learning approaches that predict whether two duplicate genes have still the same function. In particular, we wanted to explore the role of the TEs in the diversification of gene functions.

During this exploration, we encountered these four milestones:

1. Extension of the reference dataset with more recent data, and new descriptive variables
2. Exploration of the distributions of the descriptors, in which we identified a discrepancy between the labeled subset and the whole genome dataset
3. Development of a machine learning model with model adaptation
4. Comparison of the performance of our model with the published models

We did not achieved all these goals yet.

1.3. *Arabidopsis thaliana* as a model plant for genomics

The thale cress (*Arabidopsis thaliana*) is a small flowering plant belonging to the *Brassicaceae* family (including mustard, or cabbage for instance). It is present in almost all continents, living in a wide range of habitats: crop fields, forest clearing, disturbed soils and anthropized environments.

Figure 1.3.: A picture of a thale cress, *Arabidopsis thaliana*, in the street at Clermont-Ferrand, France (CC-By-SA Fabrice Rubio via PlantNet[1]).

Because thale cress has a small size and reproduces quickly (several generations per year), it is a privileged model organism for genetic studies on plants. The limited size of its genome (approximately 135 Mbp) and haploid chromosome number of five further facilitated the sequencing of its genome. Indeed, Arabidopsis was the first plant to be sequenced in the early 2000s (The Arabidopsis Genome Initiative 2000). The Arabidopsis genome sequencing project enabled more studies. It gave early insight on TEs, duplicate genes, structural and functional gene annotations in plants.

2. Material and methods

2.1. Source data

We made our analyzes on the plant model organism *Arabidopsis thaliana*. We used the Ensembl Plant TAIR10 release of *A. thaliana* genome[1].

The following is a list of the input files we used from Ensembl Plant:

- `Arabidopsis_thaliana.TAIR10.pep.all.fa.gz` – Protein sequences in compressed FASTA format.
- `Arabidopsis_thaliana.TAIR10.cds.all.fa.gz` – Protein coding sequences in compressed FASTA format.
- `Arabidopsis_thaliana.TAIR10.60.gff3.gz` – Gene features in compressed GFF3 format.

2.2. Identification of duplicate gene pairs

To identify the duplicate gene pairs we used the pipeline "FTAG-Finder" ported to `snakemake` (Köster et al. 2012) during my Master 1 internship last year in LaMME (Ortion 2024). This pipeline has been built by successive interns working under the supervision of Carène Rizzon and Franck Samson at LaMME since 2014 (Bouillon 2016; Normand 2017; Jasmin 2016; Correa et al. 2021) and was originally written for the Galaxy plateform (Blankenberg et al. 2010).

The workflow boils down to these main steps:

1. Run an all-against-all BLASTp alignment on the longest protein isoforms from a species proteome,
2. Merge the BLASTp hits,
3. Filter the BLASTp hits based on cumulated coverage (here we chose a threshold of >60%) and cumulated similarity percentage (we chose a threshold of >30%),
4. Extract a protein homology graph, using the maximum `bitscore` as a homology metrics,
5. Run the Markov clustering algorithm on this graph to extract partition of gene vertices corresponding to duplicate genes families,
6. Enumerate all gene pairs within these families.

The Snakemake implementation of the FTAG-Finder workflow is open-source, released under the permissive MIT license and available on GitLab (`https://gitlab.com/sortion/FTAG-Finder/`). A bit more details on FTAG-Finder is presented in section A.1

2.3. Functional redundancy

2.3.1. Assessing functional redundancy with double gene knock-outs

To assess the functional redundancy of duplicate genes, biologists perform a gene knock-out. A gene knock-out is a modification of the genome sequence that disable the expression of a gene. After the gene knock-out, they observe the phenotype of the mutant organism. If when only one of the duplicate genes is knocked-out, the phenotype is still viable, but the phenotype is no longer viable when both duplicate genes are knocked-out, then the genes are considered functionally redundant.

[1] `https://plants.ensembl.org/Arabidopsis_thaliana/Info/Index`

Gene duplication may introduce a functional redundancy: two genes may encode proteins having a similar function. In this case the knock-out of only one of the duplicate genes is not sufficient to introduce a loss of function, the remaining intact gene being able to support the function. A double knock-out is needed to assess whether the duplicate genes share redundant functions.

2.3.2. How to induce a gene knock-out in plants? The example of the transfer DNA method

To induce a gene knock-out in plants, the biologists may use the Transfer DNA (T-DNA) method. The T-DNA comes from a bacterial plasmid, that is capable of introducing itself in a plant's genome. The DNA transfer mechanism can be used at our advantage when the insertion is targetting a specific gene locus, thus disrupting the gene. Figure 2.1.

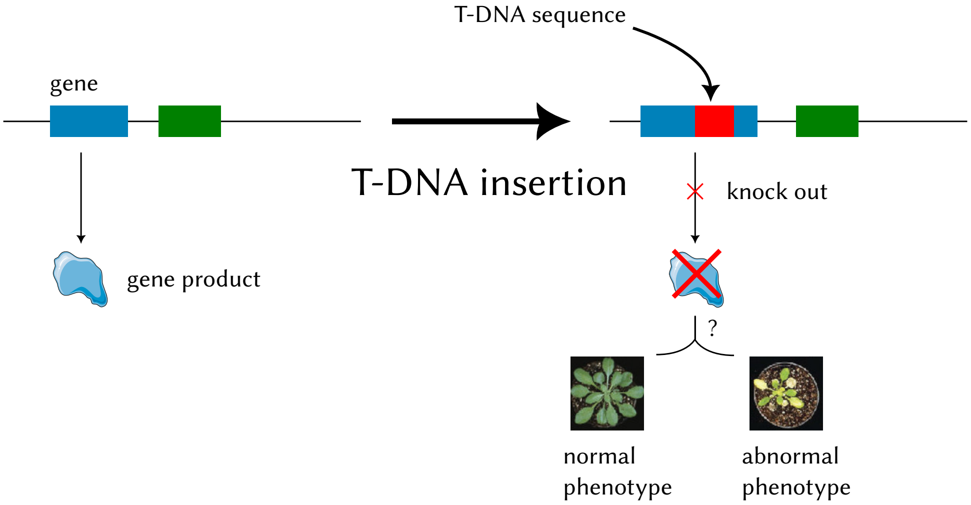


Figure 2.1.: Transfer DNA mediated knockout principle. On Transfer DNA insertion in a gene, the gene is no longer expressed. This may lead to two possible outcomes: either the plant still has a normal phenotype, develops and reproduces normally, or it may have an abnormal phenotype.

2.3.3. Expert functional redundancy gene annotation

A gene knock-out experiment is expensive, and a lot of genes do not show a visible phenotype change when knocked-out. Some gene pairs identified as functionally redundant were so identified, not with a thorough double knock-out experiment but via an expert annotation, in the data sources we considered.

2.4. Origins of the dataset prediction labels

The aim of our model is to predict whether two duplicate genes have acquired different functions. We gathered ground truth data from four sources published in the literature (Bolle et al. 2013; Ezoe et al. 2020; Lloyd et al. 2012; Meinke 2019).

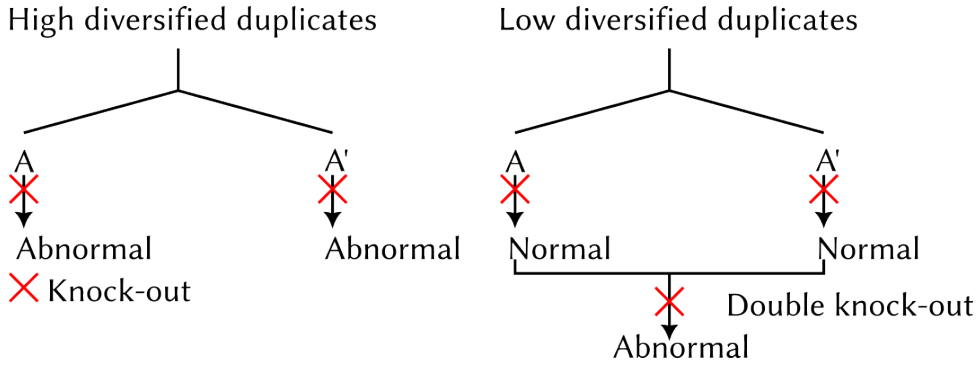


Figure 2.2.: How are the label determined. When a single gene knockout on each duplicate gene induces an abnormal phenotype, we consider that the genes have a low degree of diversification. On the other hand, when a double-knockout is necessary to induce an abnormal phenotype, we consider than the genes have a high degree of diversification.

In the dataset, we encode the labels as follows:

$$Y = \begin{cases} 0 & \text{low degree of functional divergence} \\ 1 & \text{high degree of functional divergence} \end{cases}$$

A low degree of functional divergence means the duplicate genes share the same function whereas, a high degree of functional divergence means they have different functions.

2.5. Extension of the dataset labeled set

The labeled set of duplicate genes is the union of all labeled gene pairs found in Bolle et al. (2013), Lloyd et al. (2012), Meinke (2019), and Ezoe et al. (2020). Recent papers aiming at building a prediction model (Ezoe et al. 2020; Cusack 2020) used data dating from 2013. I added more recent gene pairs, coming from (Meinke 2019). When a single pair is found to be associated with two different labels in the literature, we make the choice to simply discard this pair from the dataset. This choice is motivated by the ambiguity associated with such pairs.

Table 2.1.: Counts detailing the source of the labels.

Source	Labeled pairs	Newly acquired data	Redundant (0)	Divergent (1)	Ambiguous
Ezoe et al. (2020)	574	-	111	454	9[2]
Bolle et al. (2013)	50	43	50	0	4
Meinke (2019)	51	33	46	0	5
Total	651	77	207	454	9

2.6. Extraction of the features of the dataset

Most of the descriptors used to predict the functional redundancy was previously extracted for *Arabidopsis thaliana* by Seanna Charles, a previous intern in the laboratory (Charles 2024). I built my version of the reference dataset on top of her work, extending her dataset by including more labels from the literature and introducing new genomic-based descriptors.

2.6.1. Transposable element environment

Based on the gene feature annotations in the Arabidopsis TAIR10 genome, Seanna Charles extracted two metrics for the transposable element environment of each gene. Within a window of 2000 bp around the gene,

[2] Same as total, as the others sources are concordant otherwise.

the transposable element coverage is defined as

$$\text{Coverage} = \frac{\sum_{i=1}^{n} \textsc{End}(\text{TE}_i) - \textsc{Start}(\text{TE}_i) + 1}{\textsc{End}(\text{gene}) - \textsc{Start}(\text{gene}) + 1} \times 100, \tag{2.1}$$

where n is the number of transposable elements within the 2 kb window. The transposable element density is also defined as

$$\text{Density} = \frac{n}{(\textsc{End}(\text{gene}) - \textsc{Start}(\text{gene}) + 1) - \sum_{i=1}^{n} \textsc{End}(\text{TE}_i) - \textsc{Start}(\text{TE}_i) + 1} \times 1000. \tag{2.2}$$

The two metrics can then be used in a multivariate model, where each gene is associated with both metrics. We can also build a joint descriptor using both information.

Paired descriptor Based on the quantitative metrics describing the transposable elements of genes we classified each genes into three categories: TE-free, TE-poor, and TE-rich. TE-free genes have a null TE-density and coverage, meaning they do not have any TEs in their close neighborhood. Then, TE-poor and TE-rich classes are differentiated using a k-medoids clustering on the remaining genes. Figure 2.3 represents the k-medoids clustering of the whole genome, except TE-free genes. The genes are associated with their closest medoids, and the genes associated with the medoid with the lowest TE-density and TE-coverage is classified as TE-poor, the others being classified as TE-rich.

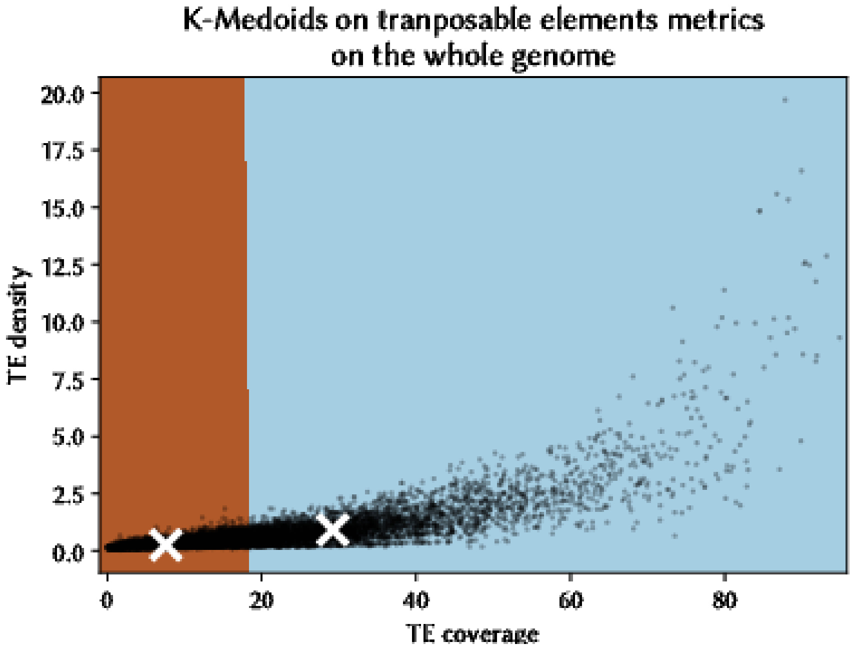


Figure 2.3.: K-medoid clustering ($k = 2$) on transposable element density and coverage. White crosses represent the medoid positions. Flat tints correspond to the area linked to each medoid.

Moreover, we associated each pair of duplicate genes with the pair of TE-environments formed with each of them (e.g., TE-free – TE-free, TE-free – TE-rich, ...).

2.6.2. Gene expression data

RNA-seq-based expression data has been used to measure the difference of gene expression between two duplicate genes. To do so, Seanna Charles downloaded `fastq` files from the CatDB database. The identifiers of the experiments are listed in her report. She used FastP to pre-process the reads (Chen et al. 2018) and kallisto to count the number of transcripts per genes (Bray et al. 2016).

The gene expression divergence is measured with a Manhattan distance d_m on the series of gene expression measure on each experiment i and gene g e_{gi}.

$$d_m = \frac{1}{2}\sum_{k=1}^{n}\left|\frac{e_{1k}}{\sum_{i=1}^{n} e_{1i}} - \frac{e_{2k}}{\sum_{i=1}^{n} e_{2i}}\right|.$$

To compute this metrics in an more efficient manner, I rewrote her python script into a Rust program, using the `pola.rs` crate (a Rust library implementing columnar data frames).

The Manhattan distance ranges from 0 to 1. Values close to 1 corresponds to gene pairs with very different gene expression profiles in the series of experiments, tissues and organs considered.

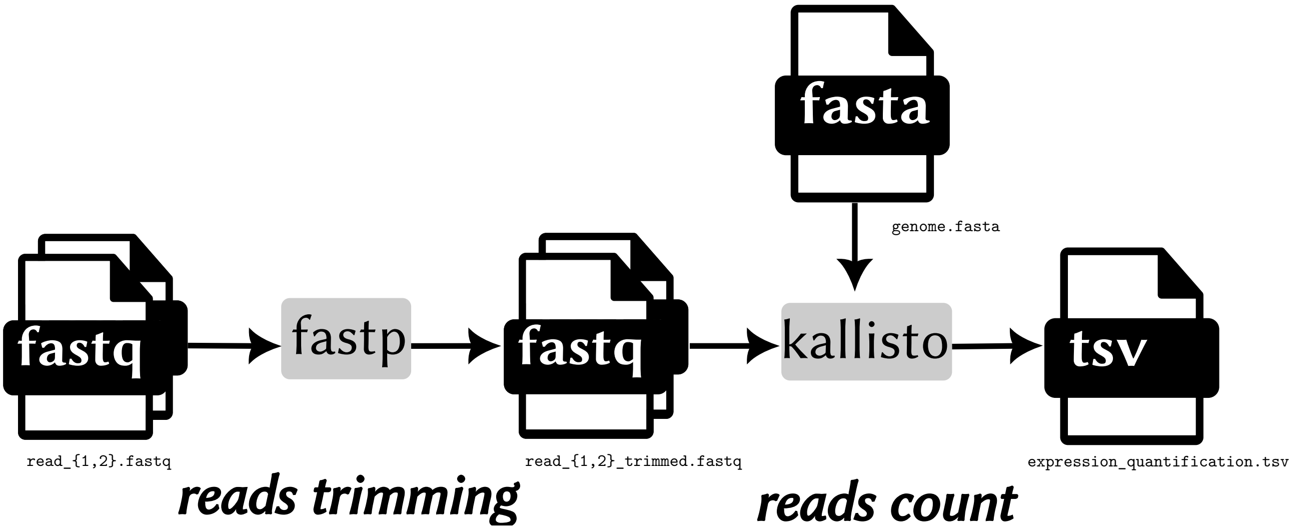


Figure 2.4.: A diagram of the pipeline for the quantification of the gene expression in a given condition, from RNA-seq reads in FASTQ format

2.6.3. Age of duplication and selection pressure

Each amino-acid of a protein is encoded by a codon of three nucleotides. As there exists more codons than amino-acids, different codons may encode the same amino-acid. When a mutation in a coding sequence changes a codon, two possible outcomes are possible: the encoded amino-acid may change, or stay the same. We call the former a *non-synonymous* mutation and the latter a *synonymous* mutation. A synonymous site is a nucleotide position which can be subject to a synonymous mutation. The number of synonymous mutations computed between two homologous sequences over the total number of synonymous sites (either denoted by d_s, or k_s) can be viewed as a measure of the age of duplication, if we consider the mutation rate to be constant over time. Indeed synonymous mutations do not affect the protein sequence and thus the fitness of the individual. On the other hand, a non-synonymous mutation modifies the peptide sequence. We denote by d_n or K_a the rate of non-synonymous mutation over the total number of non-synonymous sites. The ratio of $\omega = K_a/K_s$ corresponds to the selection pressure on the gene.

There exists multiple methods to compute the K_a and K_s. We used the Yang and Nielsen model YN00 (Z. Yang et al. 2000) implemented in the PAML package (Ziheng Yang 1997; Z. Yang 2007).

2.6.4. Pfam domains

Pfam annotations are functional domains identified in protein sequences thanks to sequence profiles. These sequence profiles are Hidden Markov Models (HMM) that represent the domain motifs in the proteic sequence. Using HMMer with option `--cut_tc`, we identified all Pfam domains on TAIR10 first isoforms of each gene annotated in Araport11 (a reannotation of the TAIR10 genome) (Cheng et al. 2017). This work has been done previously by Franck Samson, for the database GBOT[3] developed in the laboratory (Samson et al. 2024). We simply extracted the annotations using a PostgreSQL query.

We defined a descriptor based on the Pfam annotations of each duplicate gene pair in our dataset. Let Pfam_1 and Pfam_2 be the multisets of Pfam domain annotations of gene 1 and 2. We chose a multi-set representation, *i.e.* a set whose elements can appear several times, because a single Pfam domain can be found multiple times on the same protein sequence. The descriptor $\text{Pfam}_{\text{descriptor}}$ is defined as a Jaccard index.

[3] GBOT: `http://stat.genopole.cnrs.fr/server/gbot/index.html`

A Jaccard index, measuring the similarity between two sets A and B is defined as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{2.3}$$

which, applied to the sets of Pfam annotations on the duplicate proteins gives

$$\begin{aligned} Pfam_{\text{Jaccard}} &= J(Pfam_1, Pfam_2) \\ &= \frac{|Pfam_1 \cap Pfam_2|}{|Pfam_1 \cup Pfam_2|} \end{aligned} \tag{2.4}$$

I developed a C++ program to compute this descriptor.

The Pfam$_{\text{Jaccard}}$ ranges from 0 to 1, where 1 corresponds to a gene pair having all Pfam domains in common.

2.6.5. Protein-protein interactions (PPI)

Two proteins may form proteic complexes. These complexes can be detected *in vitro* using affinity experiments.

The network of protein-protein interactions brings a wealth of information about the environment of a protein and its function in the cell. It is indeed assumed that proteins with a similar neighborhood in the PPI network would have a similar place in the metabolism of the cell. Thus a study of the neighborhood of the duplicate genes in the PPI graph is expected to give some information on their relative functions.

A curated PPI network with 11,402 different interacting proteins (vertices) totaling 80,971 interactions (edges) was provided to us by Marie-Hélène Mucchielli-Giorgi from the IPS2 laboratory (Institute of Plant Sciences of Paris-Saclay). Based on this network, the Jaccard index between two proteins A, B is defined as

$$\text{PPI Jaccard}_{A,B} = \frac{|\mathcal{N}(A) \cap \mathcal{N}(B)|}{|\mathcal{N}(A) \cup \mathcal{N}(B)|}, \tag{2.5}$$

where $\mathcal{N}(v)$ is the neighborhood of vertex v in the PPI graph, namely, the proteins interacting with v, and $|.|$ is the cardinal number of a set. In the subsequent plots, this value is represented as a percentage.

I rewrote Seanna Charles's Python script, after discovering a mistake that made the Jaccard index percentages go beyond 100.

In the PPI network we considered, we did not made the distinction between each protein isoform and considered all proteins interacting with any isoform associated with a particular gene identifier.

Knowing our model organism *Arabidopsis thaliana* is fairly well known, we can consider that a missing edge in the interaction graph is not due to a lack of information, but rather corresponds to a real absence of interaction between the two proteins.

The PPI Jaccard values range from 0 to 100. Values near 100 correspond to a gene pair whose proteins share most of interacting neighbors in the graph.

2.6.6. Gene family sizes

A gene family with a lot of gene copies is expected to have more diversified members. Thus, we kept the size of the gene families as a descriptor.

Figure 2.5 shows the distribution of *Arabidopsis thaliana* proteome version TAIR10 protein families sizes. We see the typical exponential degrowth of this distribution: most of the families has no more than two genes, whereas the more the family size increases, the less families this size corresponds to.

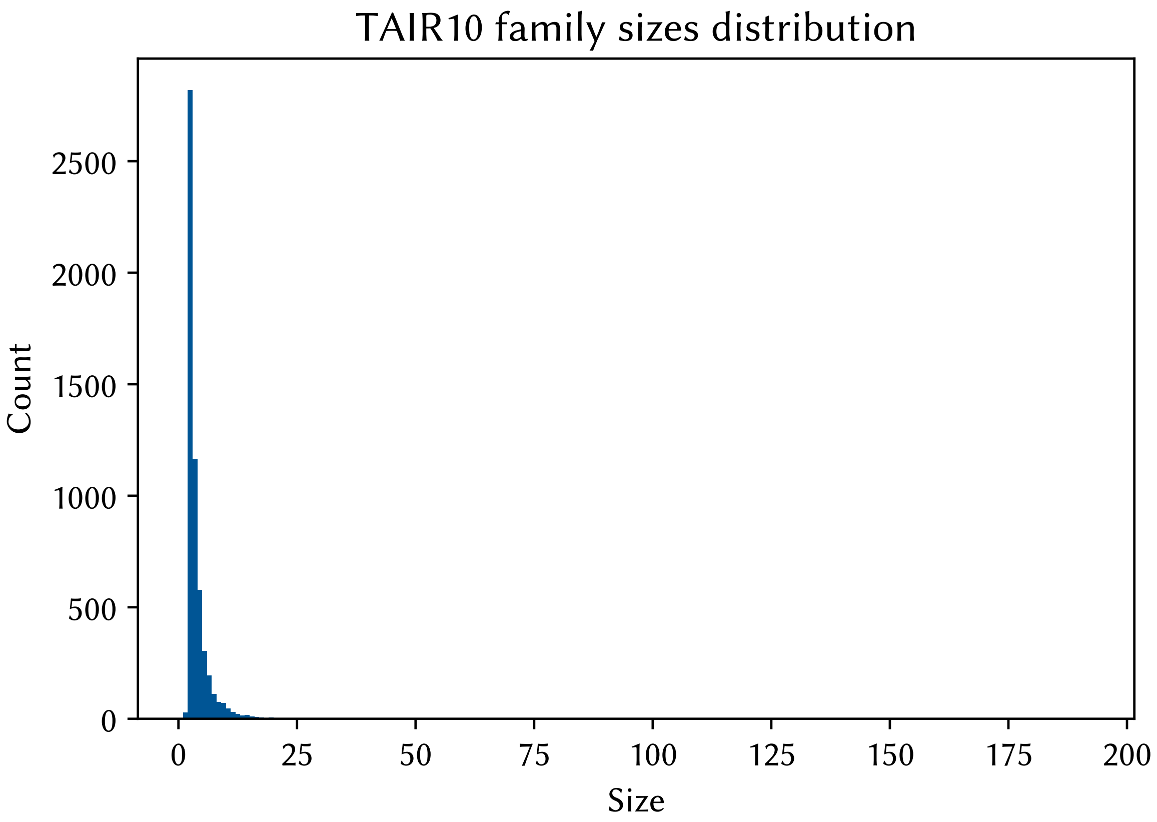


Figure 2.5.: Distribution of *Arabidopsis thaliana* TAIR10 duplicate genes families sizes.

2.6.7. Summary of the features

- K_a – rate of non-synonymous mutation over non-synonymous sites
- K_s – rate of synonymous mutation over synonymous sites (age of duplication)
- K_a/k_s – selective pressure
- $\Delta_{\text{expression}}$ – conservation of gene expression
- Pfam_i, $i \in \{1, 2\}$ – number of pfam domains for gene 1 and 2
- $\text{Pfam}_{\text{Jaccard}}$ – jaccard index between Pfam annotations of proteins
- $\text{Pfam}_{\text{Union}}$ – union between Pfam annotations of proteins
- $\text{Pfam}_{\text{Inter}}$ – intersection between Pfam annotations of proteins
- $\text{PPI}_{\text{interactang},i}$ – number of interacting proteins with protein i in the PPI network
- $\text{PPI}_{\text{jaccard}}$ – jaccard index on PPI data between proteins
- $\text{TE}_{\text{coverage},i}$ – coverage of transposable elements in the vicinity of the gene i
- $\text{TE}_{\text{density},i}$ – density of transposable elements in the vicinity of the gene i
- $\text{TE}_{\text{number},i}$ – number of transposable elements in the vicinity of the gene i
- $\text{TE}_{\text{environment},i} \in \{\text{TE-free, TE-poor, TE-rich}\}$ – classes representing the Transposable Element (TE) environment in the vicinity of the gene i
- $\text{TE}_{\text{environment pair}}$ – sorted concatenation of the classes of each gene of a pair

- $\text{TE}_{\text{environment class}} \in \{\text{free, half, rich}\}$ – qualification of the pairwise TE-environment categories in three classes, free for TE-free – TE-free pairs, half for pair with one TE-free gene and rich when both genes are either TE-rich or TE-poor
- family_size – size of the homologous gene family

Table 2.2.: Detail of the domain of constructed quantitative descriptive variables and the interpretation of the values

Descriptor	Domain	Signification
K_a	$\mathbb{R}_+$	the higher the value, the more constrained is the sequence
K_s	$\mathbb{R}_+ \ (< 5)$	the higher the value, the older the duplication
${}^{K_a}/_{K_s}$	$\mathbb{R}_+$	values above 1 correspond to adaptative selection, values below 1 corresponds to purifying selection, the farther from 1, the stronger the selective pressure
$\Delta_{\text{expression}}$	$[0, 1]$	higher values correspond to more divergent gene expression profiles
$\text{Pfam}_{\text{Jaccard}}$	$[0, 1]$	higher values correspond to gene pairs with more Pfam protein domain shared
$\text{PPI}_{\text{Jaccard}}$	$[0, 1]$	higher values corresponds to gene pairs with more common interacting proteins

2.7. A brief introduction to statistical learning and to the algorithms involved in our exploratory analysis

Statistical learning, or machine learning is a set of methods aiming to learn a function on a set of p attributes, also called *explanatory variables*, or *descriptors* to a variable we want to predict.

We have some data, that is a set of *individuals*, or *objects* described by their attributes:

$$\mathbf{X} = (x_1, ..., x_j, ..., x_p) \in \mathcal{X},$$

where x_j is an observation of variable j for the individual, and $\mathcal{X}$ is a variable space, generally $\mathbb{R}^p$.

The objective of a machine learning algorithm is to provide a function $f : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{Y}$ is the space in which the value we want to predict lies.

2.7.1. Supervised and unsupervised learning

In supervised learning, we know in advance the label associated with the individuals in the training dataset. On the other hand, in an unsupervised learning, the learning does not use any previously labeled data. The later case is often referred to as clustering. It aims to gather the data entries into groups for further analyses. In our case, we are facing a supervised learning problem, as we do have a label describing the functional redundancy in our training dataset.

2.7.2. Classification and regression

Two main categories of learning algorithms exist. One category corresponds to the algorithms that learns a function mapping the explicative variables into a category, a task known as classification. Another category corresponds to algorithms learning a function mapping the explicative variables to a number, a task known as regression. In our case, the task of determining whether a pair of duplicate genes is functionally redundant is a classification task.

2.7.3. Training and test datasets

To be able to assess the performance of a model, a typical machine learning workflow involves the separation of the dataset into two subsets: a training set and a test set. The learning algorithm is ran on the training set and the performance of the model is assessed by comparing the prediction of the models with the expected labels on the test set.

2.7.4. K-Medoids clustering

The K-medoids clustering is very similar to the K-means algorithm, except that instead of selecting the average position of the cluster of points to update the centroids, the centroids are selected among points present in the dataset. The problem of K-medoids clustering being NP-hard (Hsu et al. 1979), heuristics has been proposed to solve it efficiently. One of them is the Partitioning Around Medoids (PAM) algorithm (Kaufman et al. 1990).

Algorithm 1 Partitioning Around Medoids (PAM)

1. (BUILD) Initialize greedily k of the n data points as the medoids: (i) first select the point that minimizes the distance to all other points, and (ii) iteratively add the $k - 1$ points that minimizes the sum of distances from the medoids to their closest neighbors.
2. Associate each point to the closest medoid.
3. (SWAP)

```
repeat
    for each medoid $m$ and each other data point $o$ do
        compute the cost change obtained in case of a swap of $m$ with $o$
        if the change is the one that minimize the cost then
            remember this $o, m$ combination as $o_{\text{best}}, m_{\text{best}}$
    swap $o_{\text{best}}$ with $m_{\text{best}}$
until The cost of the configuration no longer decrease.
```

2.8. Source code

The source code of the tools developed throughout the internship as well as analysis notebooks is available at `https://gitlab.com/sortion/lamme2025/`.

2.9. Tools version

We ran our statistical analyzes with the R programming language version 4.5.0. Most of the machine learning experiments was done on Python 3.11 with scikit-learn version 1.6.1.

3. Results

3.1. General statistics on the gathered dataset

Table 3.1.: Summary counts for *Arabidopsis thaliana*

Proteins	27,416
Duplicates	19,776 (64%)
Singletons	7640 (27%)
Gene Families	5910
Intra-Family Pairs	75,721

Table 3.1 summarizes some statistics on the duplicate genes found in *Arabidopsis thaliana* as detected by the FTAG-Finder pipeline (see section 2.2).

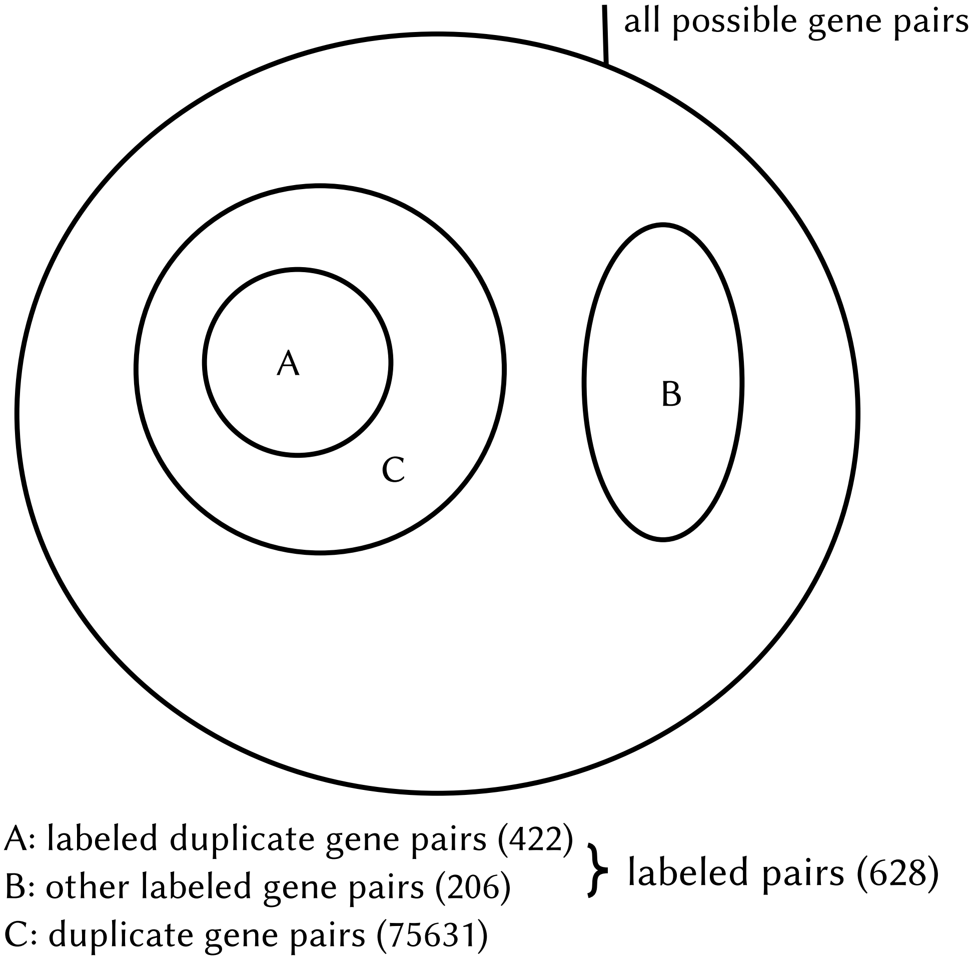

all possible gene pairs
A
C
B
A: labeled duplicate gene pairs (422)
B: other labeled gene pairs (206)
labeled pairs (628)
C: duplicate gene pairs (75631)

Figure 3.1.: A Venn diagram that represents the whole set of possible gene pairs, among which we can find the duplicate gene pairs, including labeled gene pairs. There is also the set of gene pairs that are labeled, but which do not appear in the list of duplicate genes identified by our pipeline FTAG-Finder.

The whole gathered dataset contains different set of genes. The main part corresponds to the set of duplicate genes, identified with FTAG-Finder. Among the whole set of gene pairs of the dataset, we know the ground truth label for a small fraction of them. Of theses labeled gene pairs, some are duplicate gene pairs (set A) and other are non-duplicate (i.e., there is a functional redundancy for this gene pair in the literature, but we do not consider it as duplicate), this corresponds to set B. The set C corresponds to the whole set of duplicate genes. Figure 3.1 depicts the relation between these three sets of gene pairs as a Venn diagram.

Table 3.2.: Label counts on the gathered dataset

A	**duplicate gene pairs**	cardinal
$A(1)$	functionally divergent pairs	454
$A(0)$	functionally redundant pairs	174
B	**non-duplicate gene pairs**	cardinal
$B(1)$	functionally divergent pairs	200
$B(0)$	functionally redundant pairs	6

The proportion of the label in the dataset is uneven. Table 3.2 references the counts of functionally divergent pairs and functionally redundant pairs. As can be read from this table, most gene pairs from the dataset are functionally divergent. This discrepancy is dramatic for the set of labeled genes that we did not identify as duplicate.

3.2. Analyses of the descriptive variables

We begin our analyses by a descriptive exploration of the distributions of the explanatory variables in the whole set of duplicate gene pairs, compared to the set of labeled pairs. We are interested in the discrepancies of these distributions between the whole set of duplicate gene pairs and the subset of labeled gene pairs. We are also interested in the discrepancies between the distributions of descriptors of functionally redundant pairs and functionally divergent pairs. For the comparison between the whole set of duplicate genes and the subset of labeled genes, we compare the distributions of descriptors evaluated for gene pairs taken from set C versus set $A \cup B$ ($A \subset C$).

3.2.1. Lengths of the proteins: duplicate proteins tend to be longer than singletons

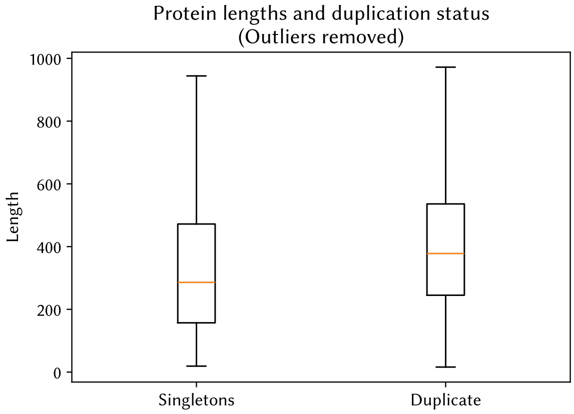
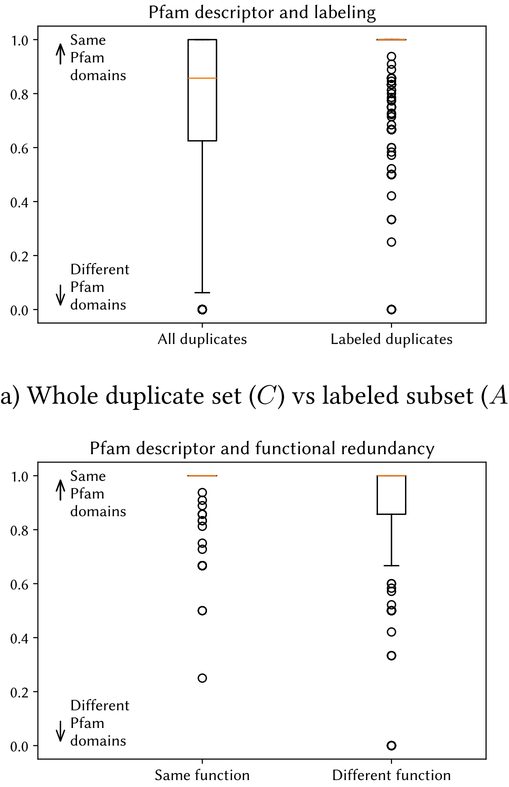


Figure 3.2.: Boxplot of duplicate genes and singleton protein lengths. Duplicate genes protein lengths tend to be greater than singleton protein lengths.

We run a non-parametric Wilcoxon – Mann-Whitney test on the series represented as boxplots in Figure 3.2. The value of the U statistics is $U = 174840980$. The p-value is $3 \cdot 10^{-234}$. $p < 0.05$, so at level $\alpha = 0.05$, we reject the null hypothesis: the singleton proteins tend to be significantly shorter than duplicate proteins.

3.2.2. Pfam domains



(a) Whole duplicate set (C) vs labeled subset (A)

(b) Duplicate with same ($A(0)$) versus different functions ($A(1)$)

Figure 3.3.: Boxplots of Jaccard index on Pfam annotations

A labeled duplicate gene pair has more Pfam domains in common than unlabeled duplicate gene pairs Figure 3.3a shows a boxplot of Pfam annotation Jaccard index between the whole set of duplicate genes and the labeled subset.

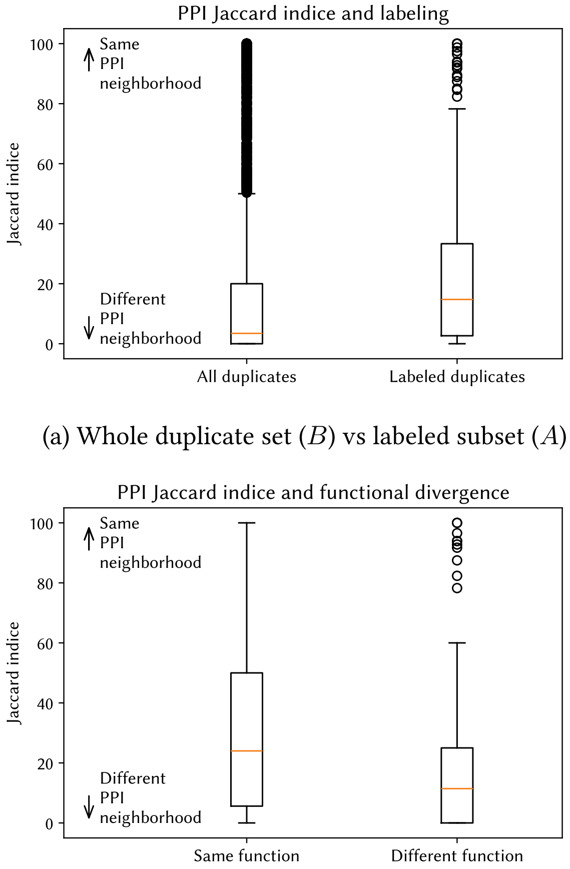
We run a Wilcoxon-Mann-Whitney test (W = 18996073, p-value < $2.2 \cdot 10^{-16}$). The p-value is below $\alpha = 0.05$, to at level 5%, we reject the null hypothesis: a labeled duplicate gene pair tends to have a larger Pfam annotation Jaccard index, thus to share more Pfam annotations, than a gene pair taken from the whole set of duplicate genes.

A gene pair with the same function has more Pfam domains in common than a gene pair with different functions As depicted in Figure 3.3b, the series of Pfam domains Jaccard index seems to be higher in gene pairs with the same function than gene pairs with different functions. This is corroborated with a Wilcoxon – Mann-Whitney test (W = 21013, p-value = $2.975 \cdot 10^{-05}$). At $\alpha = 0.05$, we reject the null hypothesis, the Jaccard index on Pfam annotations is higher when the gene pairs share the same function (i.e., they also share more Pfam annotations).

3.2.3. PPI

The PPI network contains 11,402 nodes (proteic genes) and 80,971 edges (interactions). Among these genes, we count 8,674 duplicate genes and 2,728 singletons.

A duplicate gene has more PPI interactants than singleton genes (W = 87158695, p-value < $2.2 \cdot 10^{-16}$.



(a) Whole duplicate set (B) vs labeled subset (A)

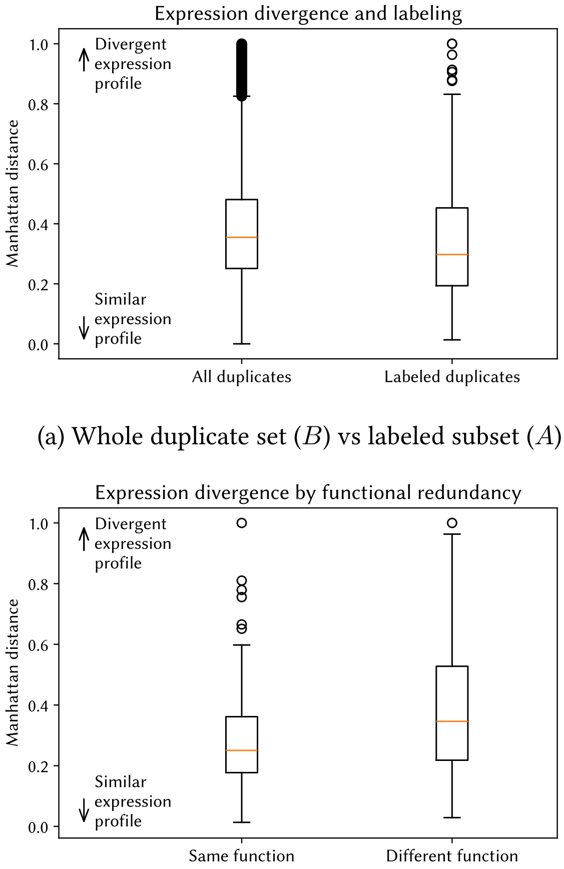
(b) Duplicate with same function ($A(0)$) versus duplicate with different function ($A(1)$)

Figure 3.4.: Boxplots of Jaccard index on neighbors in the PPI network

A labeled duplicate gene pair have more PPI interactants in common than unlabeled duplicate gene pair Figure 3.4a depicts the boxplots of PPI network neighbor Jaccard index between two genes of the whole set of duplicate genes versus the set of labeled duplicate genes. A Wilcoxon – Mann-Whitney test shows that the series are significantly different (W = 9412513, p-value < $2.2 \cdot 10^{-16}$).

A gene pair with the same function have more PPI interactants in common than a gene pair with different functions Again, we find a discrepancy between the Jaccard index between the neighborhood of genes in the PPI network. A pair of duplicate genes that still have the same function tend to have a significantly more similar neighborhood in the PPI network than a pair of duplicate genes with different functions (Wilcoxon, W = 24610, p-value = $5.616 \cdot 10^{-7}$).

3.2.4. Expression divergence



(a) Whole duplicate set (B) vs labeled subset (A)

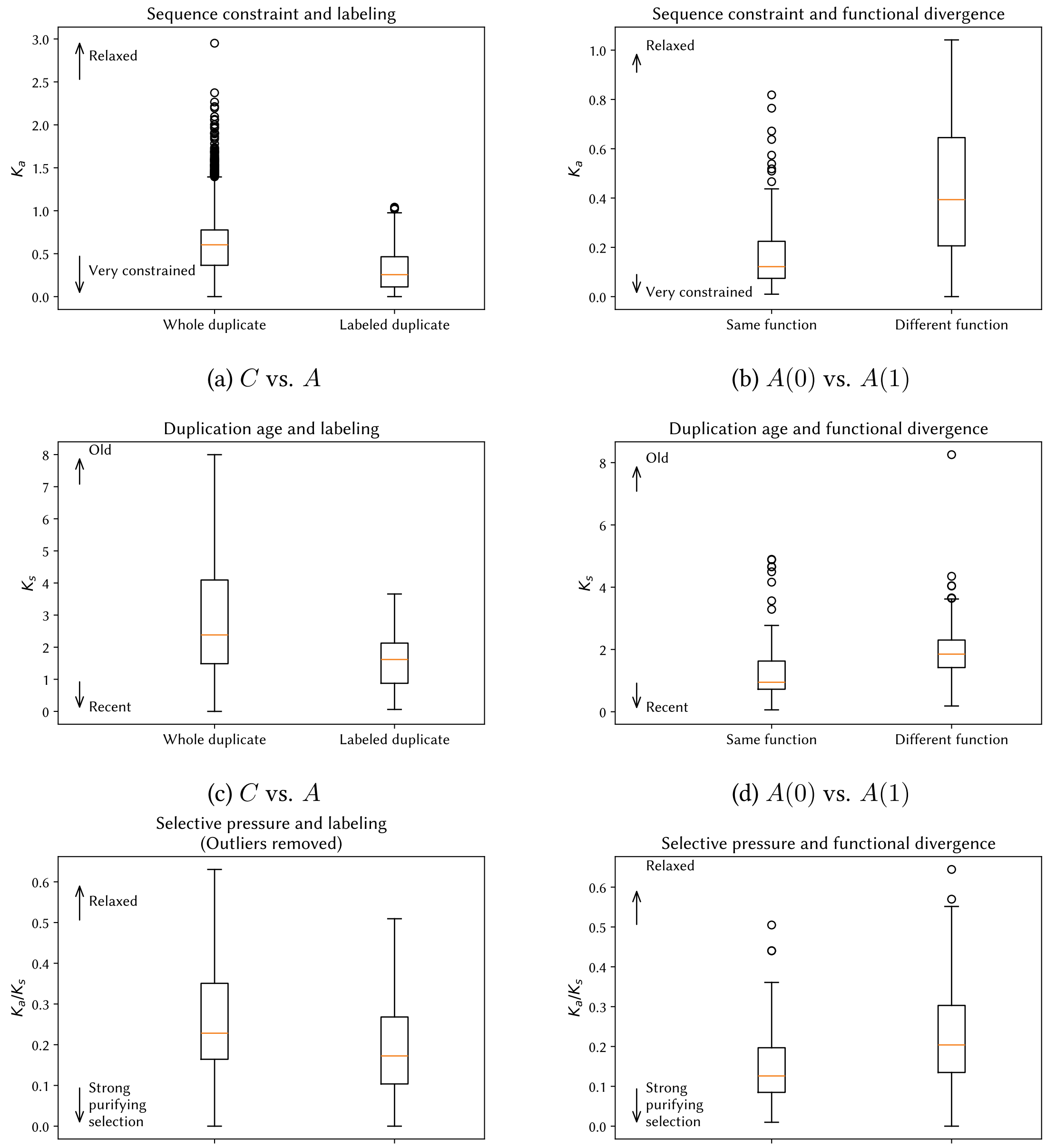
(b) Duplicate with same function ($A(0)$) versus duplicate with different function ($A(1)$)

Figure 3.5.: Boxplots of expression divergence (Manhattan distances) on RNA-seq data

A labeled duplicate gene pair have a more similar expression profile than non labeled duplicates Figure 3.6c depicts the boxplots of RNA-seq expression divergence between two genes of the whole set of duplicate genes versus the set of labeled duplicate genes. A Wilcoxon – Mann-Whitney test shows that the series are significantly different (W = 6949448, p-value = $6.939 \cdot 10^{-07}$).

A gene pair with the same function has a more similar expression profile than a gene pair with different functions Again, we find a discrepancy between the expression divergence between a gene pair with the same function and a gene pair with different functions. A pair of duplicate genes that still have the same function tends to have a significantly closer gene expression profile than a pair of duplicate genes with different functions (Wilcoxon, W = 11108, p-value = 0.001233).

3.2.5. Sequence conservation



(a) C vs. A

(b) $A(0)$ vs. $A(1)$

(c) C vs. A

(d) $A(0)$ vs. $A(1)$

(e) C vs. A

(f) $A(0)$ vs. $A(1)$

Figure 3.6.: Boxplots of sequence conservation metrics.

Figure 3.6 depicts the boxplots of K_a, K_s, and their ratio $\omega = {}^{K_a}/_{K_s}$.

Sequence constraint (K_a)

The sequence constraint, measured by K_a, is higher in labeled pairs of duplicate genes than any pair of duplicate genes (W = 6843428, p-value < $2.2 \cdot 10^{-16}$). Moreover, genes from a labeled pair have more constrained sequences than a pair of genes with different functions (W = 6977.5, p-value < $2.2 \cdot 10^{-16}$).

Duplication age (K_s)

Similarly, a duplicate gene pair tends to have been duplicated at a more recent time when it is labeled (W = 8810125, p-value < $2.2 \cdot 10^{-16}$). Again, the functional divergence of a duplicate gene pair is linked with the duplication age of this pair: a duplicate gene pair tends to be older when the genes have divergent functions

(W = 9146, p-value < $2.2 \cdot 10^{-16}$). This result is expected. Indeed, a gene pair that has been duplicated a long time ago have had more time to diverge than a gene pair resulting from a more recent duplication event.

Degree of evolutionary constraint ($\omega = {}^{K_a}/_{K_s}$)

The ratio of these metrics, a proxy of the evolutionary constraint on the proteins, is also linked to the labeling status and the label itself. An ω value ranging between 0 and 1 corresponds to a purifying selection, namely, a selection limiting protein sequence variations. The less the value of ω the stronger the purifying selection is. When ω goes beyond 1, it corresponds to an adaptive selection pressure (which allows protein sequence variation), and the stronger this adaptive pressure is, the higher the ω is valued. As depicted in Figure 3.6e and Figure 3.6f, the selective pressure is purifying ($\omega < 0.6$). A duplicate gene pair with a label tends to be submitted to a stronger purifying selection (W = 9299560, p-value < $2.2 \cdot 10^{-16}$). A similar observation can be made with the functional redundancy. A gene pair with the same function tends to be submitted to a stronger purifying selection.

3.2.6. Descriptor distribution discrepancy between functionally divergent and functionally redundant gene pairs

The B subset, corresponding to labeled gene pairs we do not consider as duplicate, is mostly labeled as divergent. We want to know if the descriptive variables for this set of similar to set A of labeled duplicate gene pairs.

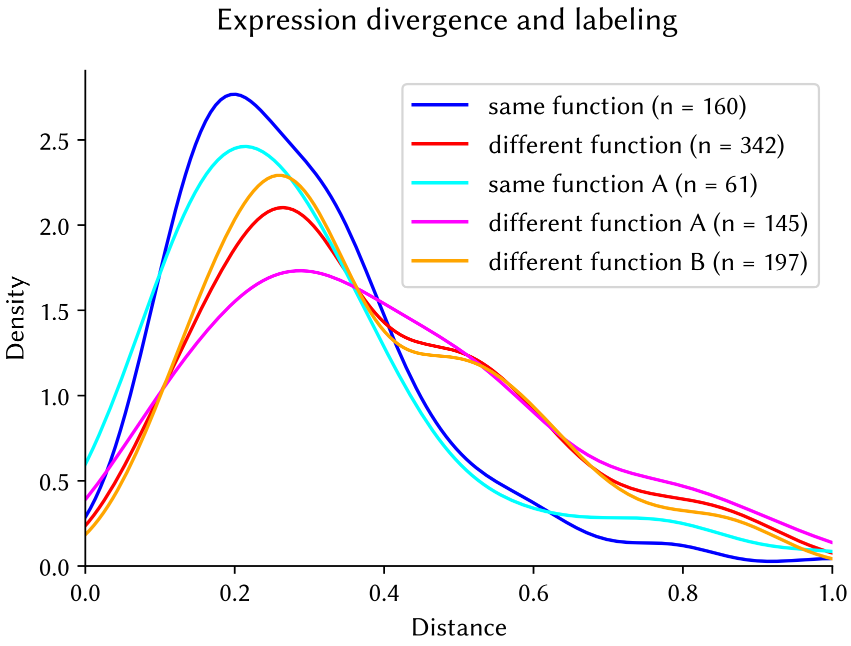


Figure 3.7.: Gene expression divergence distributions with respect to the duplication status and the label of the gene pairs.

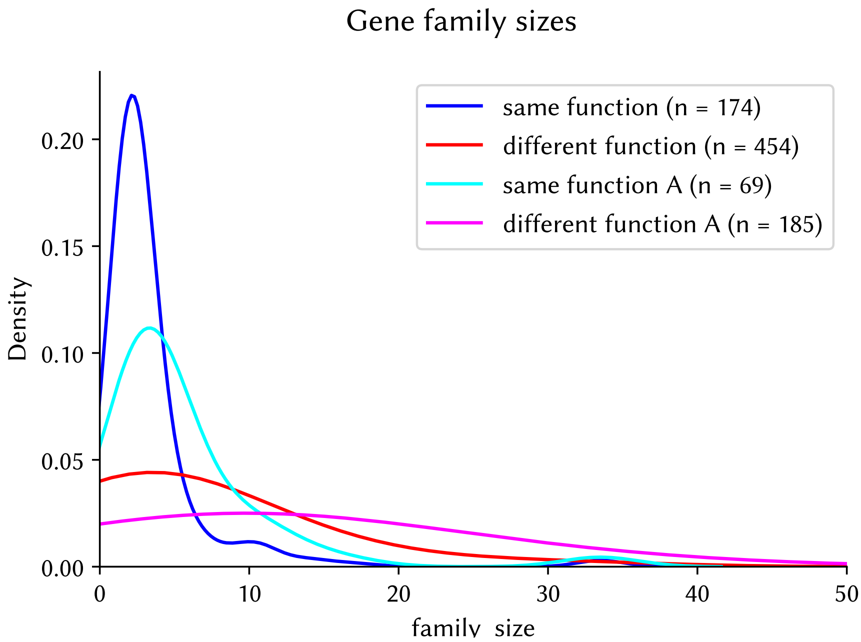


Figure 3.8.: Gene family sizes distributions with respect to the duplication status and the label of the gene pairs.

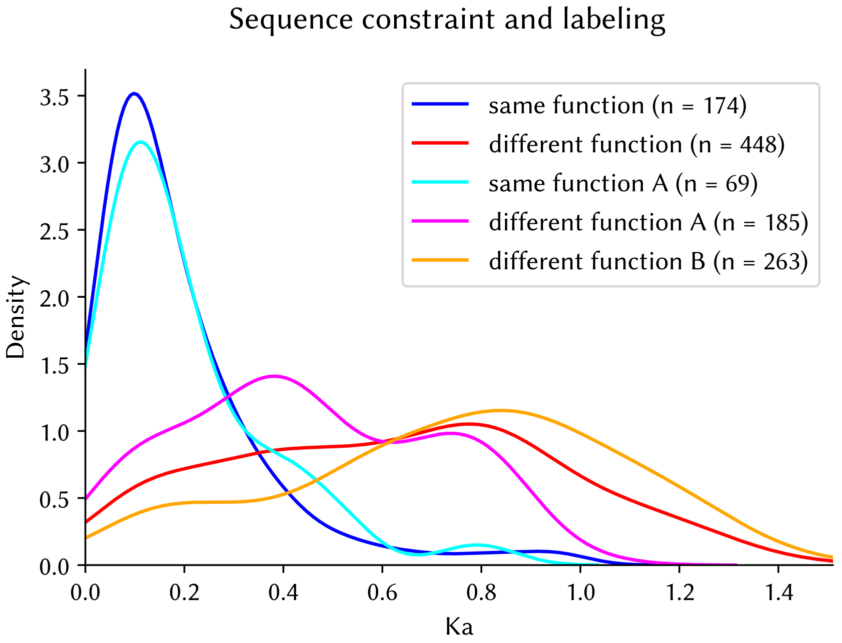


Figure 3.9.: K_a distributions with respect to the duplication status and the label of the gene pairs.

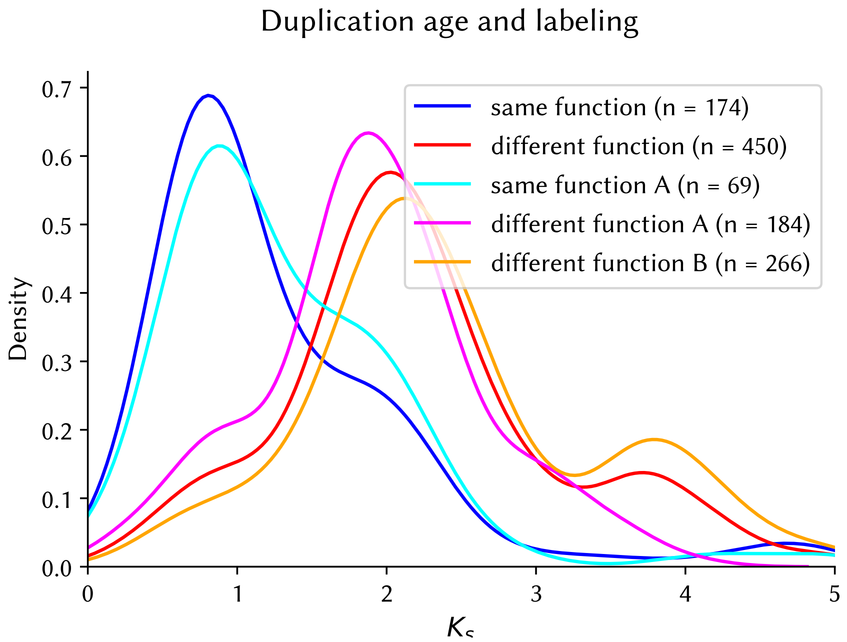


Figure 3.10.: K_s distributions with respect to the duplication status and the label of the gene pairs.

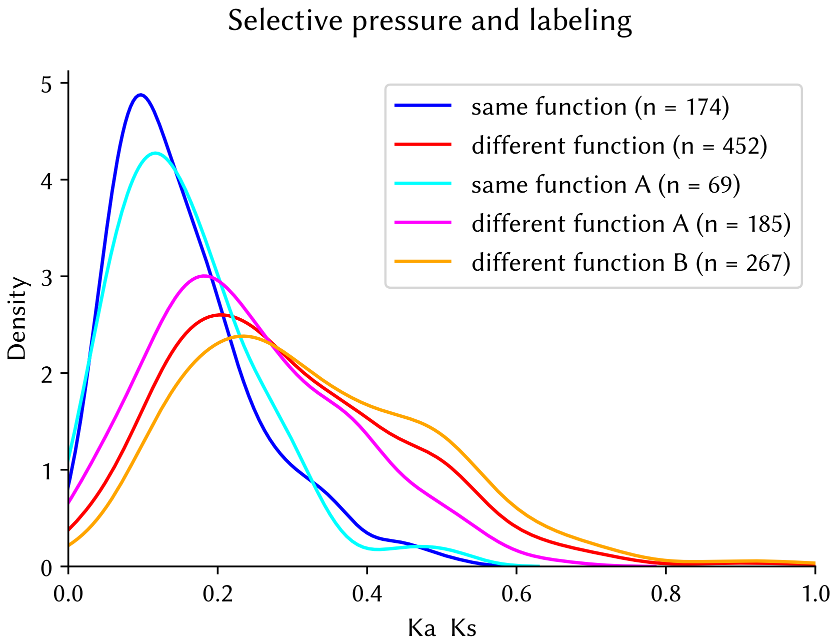


Figure 3.11.: K_a/K_s distributions with respect to the duplication status and the label of the gene pairs.

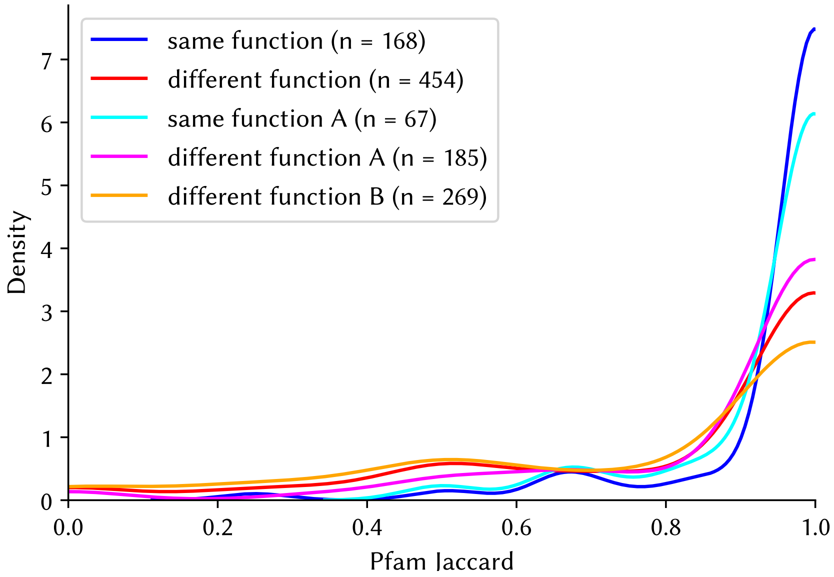


Figure 3.12.: Pfam Jaccard index distributions with respect to the duplication status and the label of the gene pairs.

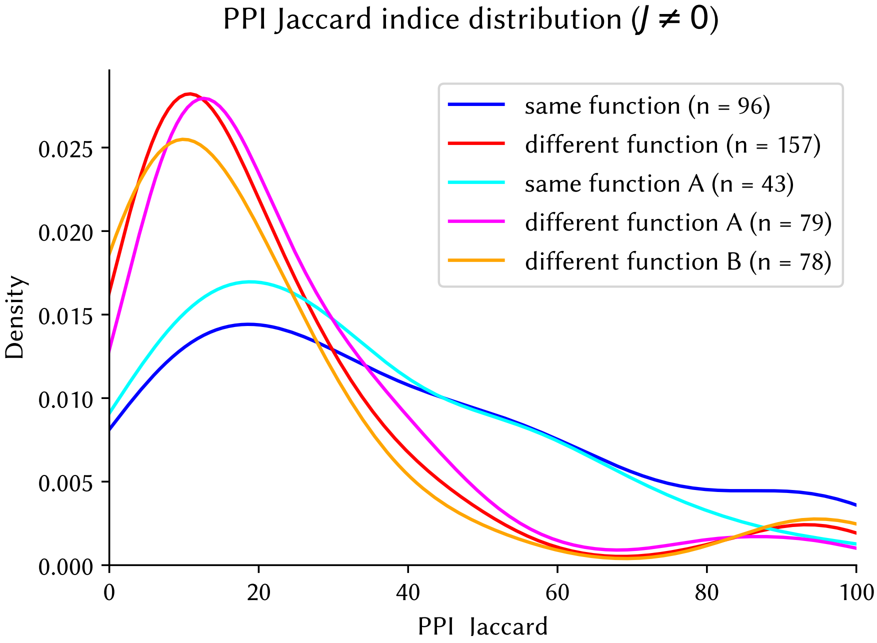


Figure 3.13.: PPI Jaccard index distributions with respect to the duplication status and the label of the gene pairs.

Figure 3.7, Figure 3.8, Figure 3.9, Figure 3.10, Figure 3.11, Figure 3.12, and Figure 3.13 illustrates the discrepancies in distribution duplicate gene pairs labeled as functionally redundant and gene pairs labeled as functionally divergent, and, within the same label category, between duplicate gene pairs and other gene pairs gathered from the literature. The non-duplicate gene pairs distributions (set B-1) is not shown due to its insignificant size ($n = 6$). All descriptor but the expression divergence shows a significantly different distribution between functionally divergent duplicate gene pairs and functionally redundant non-duplicate gene pairs (wilcoxon p-value < 0.05). This provides use with yet another argument to discard the gene pairs we consider as non-duplicate from the dataset used to train the predictive model, even if It reduces a little bit the number of training instances, as we aim to predict the functional divergence of *duplicate* gene pairs.

3.2.7. Analysis of Transposable Element environment variable with respect to the other variables

Transposable elements environment and duplication status of a gene

Table 3.3.: Contingency table of gene duplication status and TE-environment

	duplicate	**singleton**
TE-free	14585	6709
TE-poor	3189	1268
TE-rich	1712	991

Table 3.3 is a contingency table gathering the count of genes that are either duplicates or singletons and characterized by their environment in TEs. A χ^2 test of independence ($\chi^2 = 52.592$, p-value = $3.799 \cdot 10^{-12}$) indicates that there is a significant association between the duplication status of the genes and their environment in TE (at level $\alpha = 0.05$). Based on the residuals, using the "rule of $\pm$ 2", we can see that the TE-poor environment is under-represented in singletons, whereas the TE-rich environment is over-represented in singletons, compared to duplicated genes.

Transposable element environment of a duplicate gene pair

Table 3.4.: Table of expected counts for TE-environment pairs and observed counts for the homogeneity χ^2 test

	observed	**expected**
TE-free_TE-free	44133	43276.17
TE-poor_TE-poor	2070	1850.23
TE-rich_TE-rich	1458	574.19
TE-free_TE-poor	16968	17896.48
TE-free_TE-rich	7761	9969.73
TE-poor_TE-rich	2248	2061.45

To test if there is an over-represented pair of TE-environment pair, I run a χ^2 homogeneity test. The expected and observed counts is shown in table 3.4. The χ^2 test of homogeneity ($\chi^2 = 706.54$, p-value $< 2.2 \cdot 10^{-16}$) indicates that there is a statistically significant over-represented TE-environment pair. By analyzing the residuals, we can identify the TE-free – TE-free pairs are more frequent than expected, idem for TE-poor – TE-poor and TE-rich – TE-rich; and that gene pairs composed of different TE environments are less frequent than expected in the observed count, especially for TE-rich and TE-free, which symbolizes the most extreme divergence in TE-environment between a pair of genes. This suggests that duplicate gene pairs tends to have the same TE-environment.

Transposable element environment with respect to the co-localization of a gene pair

We identify three co-localization classes of a duplicate gene pairs: "Chromosome", when both gene belongs to the same chromosome, "Different chromosomes", when the genes belong to different chromosomes and "TAG1", when the genes belong to the same TAG with at most 1 spacer gene belonging to another family.

A χ^2 test of independence ($\chi^2 = 37.885$, p-value = $5.934 \cdot 10^{-9}$) on the contingency table of co-localization of genes and whether they belong to the same TE-environment class indicates that the colocalization and the TE-environment is significantly linked (at level $\alpha = 0.05$). Analyzing the residuals, we find that the gene pairs belonging to the same TAG1 (thus being located closely on the genome) tends have the same TE environment. This result is expected as the distribution of TEs is not even across the genome. It is indeed expected that genes from the same pair located next to each other share the same genomic environment, and thus TE environment in particular.

Table 3.5.: Count of colocalization of duplicate gene pairs

Co-localization	Gene pairs
Chromosome	17287
Different chromosome	53769
TAG1	4663

Table 3.6.: Count of co-localization of gene pair belonging to either the same TE environment category or a different one

	Same TE environment	Different TE environment
Different chromosome	33635	19451
Chromosome	10970	6075
TAG1	3055	1451

3.3. Machine learning with logistic regression model

3.3.1. Protocol

The dataset considered is the set of labeled duplicate gene pairs (set A) only. The whole set A is splitted into a training dataset (80% of the set A, with $n_{\text{train}} = 337$) and a test dataset (20% of the set A, with $n_{\text{test}} = 85$). The train-test split is stratified in order to obtain sets of data with a similar amount of 0 and 1 labels. To gain a better robustness in the results, I ran the split and training / testing phases 100 times.

The training and test datasets are pre-processed as follows:

- Ordinal encoding of the TE_environment_class qualitative descriptor
- Scaling of each variable with scikit-learn's StandardScaler
- Imputation of missing values with Histogram Gradient Boosting model predicting the missing values of a descriptor based on the other descriptors

3.3.2. Independent univariate logistic regression models

In a first approach to predict the functional redundancy of duplicate genes, I trained multiple independent univariate logistic regression models.

Taken alone, all but one descriptor (TE_environment_class) has a p-value below $\alpha = 0.05$, so at confidence level α, we reject the null hypothesis that the variables are unrelated to the predicted label, and conclude

Table 3.7.: Results of univariate logistic regression models. The p-values shown, as well as the accuracy is the average over 100 experiments.

descriptor	average_p_value	std_p_value	average_accuracy	std_accuracy
family_size	0.000176802	0.000131459	0.622941	0.0549142
$\text{PPI}_{\text{Jaccard}}$	$1.01427 \cdot 10^{-05}$	$2.06754 \cdot 10^{-5}$	0.660824	0.0491324
$\text{Pfam}_{\text{Jaccard}}$	0.000963043	0.00123416	0.608941	0.0524057
K_s	$4.80421 \cdot 10^{-9}$	$1.99526 \cdot 10^{-8}$	0.731647	0.0381835
K_a	$1.26611 \cdot 10^{-19}$	$5.24494 \cdot 10^{-19}$	0.726706	0.04225
K_a/K_s	$2.56443 \cdot 10^{-8}$	$8.14748 \cdot 10^{-8}$	0.672824	0.0478954
$\Delta_{\text{expression}}$	$3.15074 \cdot 10^{-5}$	$7.68782 \cdot 10^{-5}$	0.620353	0.043889
$\Delta_{\text{expression}}/K_s$	0.0772591	0.0969267	0.593529	0.0513722
TE_environment_class	0.680273	0.198611	0.599176	0.0432984

that all descriptor have a predictive capacity, except the TE_environment_class. This being said, however, the accuracy reached with these simple models are not very good, ranging from 60% to 70%.

3.3.3. Multivariate logistic regression

In a second approach, I trained a single logistic regression model with all descriptors. The average accuracy of this model across 100 experiments is 0.728 with standard deviation 0.046. Table 3.8 reports the p-value associated with each descriptor, averaged across the 100 experiments and the corresponding standard deviations.

Table 3.8.: Results of multivariate logistic regression models. The results come from a repetition of 100 different train-test splits. The average accuracy is 0.728 with standard deviation 0.046.

descriptor	**average_p_value**	**std_p_value**
family_size	0.000176306	0.000113934
$\mathrm{PPI}_{\mathrm{Jaccard}}$	$9.73047 \cdot 10^{-6}$	$3.27523 \cdot 10^{-5}$
$\mathrm{Pfam}_{\mathrm{Jaccard}}$	0.00103855	0.00164059
K_s	$2.93697 \cdot 10^{-9}$	$1.1007 \cdot 10^{-8}$
K_a	$9.74364 \cdot 10^{-20}$	$3.37293 \cdot 10^{-19}$
${}^{K_a}/_{K_s}$	$1.49097 \cdot 10^{-8}$	$3.13808 \cdot 10^{-8}$
$\Delta_{\mathrm{expression}}$	$2.01039 \cdot 10^{-5}$	$3.95739 \cdot 10^{-5}$
${}^{\Delta_{\mathrm{expression}}}/_{K_s}$	0.0874627	0.0884115
TE_environment_class	0.684642	0.207522

Interpretation of the logistic regression results The model accuracy is not strongly different from the best univariate logistic model (with K_s as unique descriptor). It is even a little bit less accurate. Compared to the logistic regression built by Ezoe et al. (2020), we find that K_a, K_s and ${}^{K_a}/_{K_s}$ are the most prominent descriptors. The new descriptors we introduced ($\Delta_{\mathrm{expression}}$, $\mathrm{Pfam}_{\mathrm{Jaccard}}$, $\mathrm{PPI}_{\mathrm{Jaccard}}$) are significantly linked with the functional divergence. Only the transposable elements class descriptor does not appear to contribute significantly to the model.

4. Discussion

The main contribution of this work is the addition of more recent gene redundancy data and the improvement on a reference dataset for *Arabidopsis thaliana* functional fate prediction.

We found that previously published models relied on a set of "duplicate" gene pairs that contains gene pairs we do not consider as such.

The experiments of double-knockout of genes being expensive, an in silico method to predict the functional redundancy is interesting, to eventually target experiments on expected functionally redundant genes.

The dataset is small The literature contains about 600 labels for gene redundancy. This is less than 1% of all possible duplicate gene pairs. Of course, if we knew the true label for every duplicate gene pairs, the interest of a predictive model would be rather limited, but this restrained dataset makes any machine learning models prone to biases and generalization difficulties.

On the choice of gene pairs considered for knock-out experiments The choice of gene pairs biologists considered for their knock-out experiments is not random. It depends on their particular interests and the possibility to see a macroscopic phenotype change on knock-out. This bias the set of labeled duplicate genes, and so, is expected to bias any machine learning trained on this data too.

On the length of duplicate proteins We found that duplicate proteins tend to be longer than singleton proteins in *Arabidopsis thaliana*. A similar result have been found in *Homo sapiens* (Vance et al. 2023), and yeast (He et al. 2005). This can be interpreted by the manifestation of the retention mechanisms of duplicate genes: longer genes, with more complexity (more proteic domains, for instance), have more chance to be retained after duplication (He et al. 2005).

The family sizes is linked to the duplication mode. For instance gene pairs coming from WGD usually belong to a gene family with this pair of gene only (Blanc et al. 2004; Wang et al. 2011). This has an impact on the function of the genes. WGD duplicates retained during evolution are involved in specific functions, Kassahn et al. (2009) found that WGD genes are "enriched for function in signaling, transcription, calcium ion transport, and metabolism". This functions are expected to remain stable, contrary to stress-response genes, often present in large clusters of genes, still present as TAG for instance. This stress-response genes are highly variable for plants to be able to adapt to varying environment. Thus, the family size can be linked to the functional divergence in two different manner: (i) by being linked to the mode of duplication, and so to the particular evolution constraints of the genes that are over-represented among this mode and (ii) by being linked to the age of family; a family with a large number of members would have had to have originated a longer time ago, if we expect a relatively constant rate of gene duplication through time. Such a long time would consequently enable more mutation to occur and new functions to be acquired, inducing a functional divergence of gene pairs belonging to this family.

Pfam domains and functional divergence Functionally diverged genes is expected to have different protein domains, because the protein domains is involved in the biochemical function of the proteins. The biochemical function of the proteins can be considered as the building blocks for the biological functions we observe indirectly via the visible phenotype of the plant.

Indeed, we find that the Jaccard indice of Pfam domain annotations is closer to 0 when gene are functionally divergent, which means that functionally divergent proteins share less protein domains, and thus ought to have more different biochemical functions.

Sequence conservation and functional divergence We found that K_a and ${}^{K_a}/_{K_s}$ have a link with the functional divergence of gene pairs. This observation is concordant with (Ezoe et al. 2020).

The ratio of expression divergence over the age of duplication, is an analog of the RE/K_s predictor present in (Ezoe et al. 2020) where RE is the correlation between expression data, and not the Manhattan distance. We could try to compute correlation instead of distance to obtain a descriptor closer to the one used in (Ezoe et al. 2020).

On the encoding of the TE-environment Maybe the ordinal encoding of the TE-environment class is not adapted to properly encode the pairwise encoding of transposable element environment. A one-hot encoding may be better suited for this. However we will have to explore how to assess the strength of the link between this qualitative variable when encoded so. A comparison of model log-likelihood might be useful.

Why is the TE-environment not helpful for functional divergence prediction We expected that transposable element environment would be involved enough in the functional divergence of duplicate gene pairs to be of strong value in prediction. However, we did not find a strong link between this variable and the functional divergence. This may be explained by the strong preponderance of TE-free environment, as was also found in *Drosophila melanogaster* (Benmehdia 2023). Maybe, in a species with more TE, such as maize, *Zea mays*, the TE environment would have a greater predictive capability.

This problem is overly-simplified compared to the biological reality Representing the functional redundancy as a Boolean variable is evidently an over-simplification of the reality. There exists a continuum between non-viable phenotype and a normal phenotype. It is not well-defined, however, as there exists no scale representing the phenotype 'abnormality'. Thus a binary representation can be considered a fair-enough compromise, considering the scarcity of the phenotype data available.

On the duplication status of labeled gene pairs considered as non-duplicate gene pairs No less than 200 of the 206 gene pairs of the labeled set from the literature (mostly gathered in Ezoe et al. (2020)) are labeled as genes with different functions. This is not unexpected. Indeed It is not so absurd that unrelated genes taken from a species proteome do not have the same function. This evidently raises the question of the choice of gene pairs the former papers made. We can made several hypothesizes for this choice, namely, gene pairs with related biological functions that phytologists wanted to compare or gene pairs taken among the genes whose knock-out renders the phenotype of *Arabidopsis* non-viable.

The method we used to identify duplications can also be one factor explaining why there are labeled gene pairs we consider as non-duplicate. Indeed, the method can returns false negative: gene pairs considered as non-duplicate whereas they are in fact duplicate. However, Seanna and I tested this hypothesis by doing a BLASTp alignment of "non-duplicate" labeled gene pairs with the proteome of *A. thaliana* and no hits were found at all. This consolidates our belief in a choice of gene pairs not only made on a sequence homology basis.

Using the mode of duplication as a descriptive variable The mode of duplication (whole genome duplication, segmental duplication, TE mediated duplication and tandem duplication) could add a wealth of information for the prediction of the functional redundancy of the duplicated genes. Indeed, the expression divergence of duplicate genes has a bias depending on the mode of duplication (Wang et al. 2011) in *Arabidopsis thaliana*. Other descriptors (such as the K_a measured in the same study) could have a similar behavior against a different mode of duplication, and have an impact on the functional redundancy of duplicate genes.

Take better account of correlations between descriptors Because many descriptors are correlated, It would be better to use machine learning methods that handle this correlations such as elastic net or group lasso. Some experiments have been made towards this goal with no probant results yet. This has to be further explored.

The need to explore further the domain adaptation methods

There is a strong discrepancy between the training dataset descriptor distribution and the whole duplicate genes dataset for which we want to perform our prediction. This renders difficult the generalization of our models to the prediction of the functional redundancy of unseen duplicate gene pairs. To alleviate this issue, we tried to apply standard domain adaptation methods (such as the methods implemented in the Python package ADAPT[1]) and the Weighted Elastic Net Domain Adaptation (WENDA) method already proposed for a regression task in biology (Handl et al. 2019)).

These methods give us hope to enhance our prediction capabilities on the whole set of duplicate genes. However, the very limited size of the training dataset limits considerably the extent to which these methods can be effective.

The choice we made to use the A subset only to train and test our models limits this difficulty. However, we will have to face this issue when we will try to adapt the model to new species, such as *Drosophila melanogaster*, a fruit fly, or *Arabidopsis lyrata* a close relative to the plant we used in our tests *Arabidopsis thaliana*.

Conclusion and perspectives During this internship, I continued an exploration towards the construction of a machine learning model capable to predict the functional redundancy of duplicate gene pairs in *Arabidopsis thaliana*. There is still a lot of work to do to obtain a reliable model, as we currently have an error rate of 30% with our best model, and are unsure of its prediction capability for randomly chosen gene pairs (e.g. duplicate gene pairs not present in the labeled dataset).

We hope that one could, in a near future, be able to build a model able to predict the functional redundancy of duplicate genes for other species, such as *Arabidopsis lyrata* or *Drosophila melanogaster*, or even *Malus domestica* Golden, the apple tree, for which there is a great potential for agronomic trait selection for prediction of functional redundancy and study of duplicate stress-gene clusters, for instance. However, this would require a preliminary work to determine ground truth labels, because, to the best of our knowledge, there exists no such data for these organisms; double knock-out experiments having not been widely performed in these species.

[1] https://adapt-python.github.io/adapt/

Bibliography

Altschul, Stephen F. et al. (Oct. 5, 1990). “Basic Local Alignment Search Tool.” In: *Journal of Molecular Biology* 215.3, 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).

Benmehdia, Assia (2023). *Analysis of the Genomic Structure of* Drosophila Melanogaster*: Is the Evolution of Duplicated Genes Related with Their Environment in Transposable Elements?* Master 2 internship report. Laboratoire de Mathématiques et Modélisation d’Évry, p. 61.

Blanc, Guillaume and Kenneth H. Wolfe (July 2004). “Functional Divergence of Duplicated Genes Formed by Polyploidy during Arabidopsis Evolution.” In: *The Plant Cell* 16.7, pp. 1679–1691. ISSN: 1040-4651. DOI: 10.1105/tpc.021410. PMID: 15208398.

Blankenberg, Daniel et al. (Jan. 2010). “Galaxy: A Web-Based Genome Analysis Tool for Experimentalists.” In: *Current Protocols in Molecular Biology* Chapter 19, Unit 19.10.1–21. ISSN: 1934-3647. DOI: 10.1002/0471142727.mb1910s89. PMID: 20069535.

Bolle, Cordelia et al. (July 2013). “GABI - DUPLO: A Collection of Double Mutants to Overcome Genetic Redundancy in *Arabidopsis Thaliana*.” In: *The Plant Journal* 75.1, pp. 157–171. ISSN: 0960-7412, 1365-313X. DOI: 10.1111/tpj.12197. URL: https://onlinelibrary.wiley.com/doi/10.1111/tpj.12197 (visited on 01/13/2025).

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

Bray, Nicolas L et al. (May 2016). “Near-Optimal Probabilistic RNA-seq Quantification.” In: *Nature Biotechnology* 34.5, pp. 525–527. ISSN: 1087-0156, 1546-1696. DOI: 10.1038/nbt.3519. URL: https://www.nature.com/articles/nbt.3519 (visited on 05/09/2025).

Buchfink, Benjamin, Klaus Reuter, and Hajk-Georg Drost (Apr. 2021). “Sensitive Protein Alignments at Tree-of-Life Scale Using DIAMOND.” In: *Nature Methods* 18.4, 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).

Camacho, Christiam et al. (Dec. 15, 2009). “BLAST+: Architecture and Applications.” In: *BMC bioinformatics* 10, p. 421. ISSN: 1471-2105. DOI: 10.1186/1471-2105-10-421. PMID: 20003500.

Charles, Séanna (2024). “Preliminary Study on the Prediction of Functional Divergence of Duplicated Genes in Arabidopsis Thaliana. Creation of a First Reference Data Set.”

Chen, Shifu et al. (Sept. 1, 2018). “Fastp: An Ultra-Fast All-in-One FASTQ Preprocessor.” In: *Bioinformatics* 34.17, pp. i884–i890. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty560. URL: https://doi.org/10.1093/bioinformatics/bty560 (visited on 05/09/2025).

Cheng, Chia-Yi et al. (Feb. 2017). “Araport11: A Complete Reannotation of the Arabidopsis Thaliana Reference Genome.” In: *The Plant Journal: For Cell and Molecular Biology* 89.4, pp. 789–804. ISSN: 1365-313X. DOI: 10.1111/tpj.13415. PMID: 27862469.

Correa, Margot et al. (May 7, 2021). “The Transposable Element Environment of Human Genes Differs According to Their Duplication Status and Essentiality.” In: *Genome Biology and Evolution* 13.5. 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).

Csardi, Gabor and Tamas Nepusz (2006). “The Igraph Software Package for Complex Network Research.” In: *InterJournal* Complex Systems, p. 1695. URL: https://igraph.org.

Csárdi, Gábor et al. (2025). *igraph: Network Analysis and Visualization in R.* manual. DOI: 10.5281/zenodo.7682609. URL: https://CRAN.R-project.org/package=igraph.

Cusack, Siobhan Anne (2020). “Modeling and Prediction of Genetic Redundancy in *Arabidopsis Thaliana* and *Saccharomyces Cerevisiae*.” Michigan State University.

Ezoe, Akihiro, Kazumasa Shirai, and Kousuke Hanada (Dec. 8, 2020). “Degree of Functional Divergence in Duplicates Is Associated with Distinct Roles in Plant Evolution.” In: *Molecular Biology and Evolution*

38.4. Ed. by Michael Purugganan, pp. 1447–1459. ISSN: 1537-1719. DOI: 10.1093/molbev/msaa302. URL: https://academic.oup.com/mbe/article/38/4/1447/6025181 (visited on 01/13/2025).

Fiston-Lavier, Anna-Sophie, Dominique Anxolabehere, and Hadi Quesneville (Oct. 1, 2007). “A Model of Segmental Duplication Formation in Drosophila Melanogaster.” In: *Genome Research* 17.10, pp. 1458–1470. ISSN: 1088-9051, 1549-5469. DOI: 10.1101/gr.6208307. PMID: 17726166. URL: http://genome.cshlp.org/content/17/10/1458 (visited on 03/24/2025).

Handl, Lisa et al. (July 15, 2019). “Weighted Elastic Net for Unsupervised Domain Adaptation with Application to Age Prediction from DNA Methylation Data.” In: *Bioinformatics* 35.14, pp. i154–i163. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz338. URL: https://doi.org/10.1093/bioinformatics/btz338 (visited on 04/23/2025).

He, Xionglei and Jianzhi Zhang (June 7, 2005). “Gene Complexity and Gene Duplicability.” In: *Current Biology* 15.11, pp. 1016–1021. ISSN: 0960-9822. DOI: 10.1016/j.cub.2005.04.035. PMID: 15936271. URL: https://www.cell.com/current-biology/abstract/S0960-9822(05)00435-5 (visited on 06/20/2025).

Hocking, Toby Dylan et al. (Oct. 11, 2024). *SOAK: Same/Other/All K-fold Cross-Validation for Estimating Similarity of Patterns in Data Subsets*. DOI: 10.48550/arXiv.2410.08643. arXiv: 2410.08643 [stat]. URL: http://arxiv.org/abs/2410.08643 (visited on 02/17/2025). Pre-published.

Hsu, Wen-Lian and George L. Nemhauser (Nov. 1979). “Easy and Hard Bottleneck Location Problems.” In: *Discrete Applied Mathematics* 1.3, pp. 209–215. ISSN: 0166218X. DOI: 10.1016/0166-218X(79)90044-1. URL: https://linkinghub.elsevier.com/retrieve/pii/0166218X79900441 (visited on 03/25/2025).

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

Kassahn, Karin S. et al. (Aug. 2009). “Evolution of Gene Function and Regulatory Control after Whole-Genome Duplication: Comparative Analyses in Vertebrates.” In: *Genome Research* 19.8, pp. 1404–1418. ISSN: 1088-9051. DOI: 10.1101/gr.086827.108. PMID: 19439512. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2720184/ (visited on 06/20/2025).

Kaufman, Leonard and Peter J. Rousseeuw (Mar. 8, 1990). *Partitioning Around Medoids*. 1st ed. Wiley Series in Probability and Statistics. Wiley. ISBN: 978-0-471-87876-6. DOI: 10.1002/9780470316801. URL: https://onlinelibrary.wiley.com/doi/book/10.1002/9780470316801 (visited on 03/24/2025).

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

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

Lloyd, Johnny and David Meinke (Mar. 1, 2012). “A Comprehensive Dataset of Genes with a Loss-of-Function Mutant Phenotype in Arabidopsis.” In: *Plant Physiology* 158.3, pp. 1115–1129. ISSN: 0032-0889. DOI: 10.1104/pp.111.192393. URL: https://doi.org/10.1104/pp.111.192393 (visited on 01/13/2025).

Meinke, David W. (2019). “Genome-Wide Identification of EMBRYO-DEFECTIVE Genes Required for Growth and Development in Arabidopsis.” In: *New Phytologist* 226.2, pp. 306–325. ISSN: 1469-8137. DOI: 10.1111/nph.16071. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/nph.16071 (visited on 01/16/2025).

Mölder, Felix et al. (Apr. 19, 2021). *Sustainable Data Analysis with Snakemake*. DOI: 10.12688/f1000research.29032.2. F1000Research: 10:33. URL: https://f1000research.com/articles/10-33 (visited on 03/26/2024). Pre-published.

Normand, Kévin (2017). “Development of CreationList, a Tool Dedicated to the Analysis of TAGs and Integration in Galaxy.” Internship defense.

Ohno, Susumu (1970). *Evolution by Gene Duplication*. Berlin, Heidelberg: Springer Berlin Heidelberg. 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).

Orphanet: Triploidy Syndrome (2025). URL: https://www.orpha.net/en/disease/detail/3376 (visited on 02/25/2025).

Ortion, Samuel (2024). *Further Development on FTAG Finder.* Master 1 internship report. Laboratoire de Mathématiques et Modélisation d'Évry.

Otto, Sarah P and Jeannette Whitton (Dec. 2000). "POLYPLOID INCIDENCE AND EVOLUTION." In: *Annual Review of Genetics* 34.1, pp. 401–437. ISSN: 0066-4197, 1545-2948. DOI: 10.1146/annurev.genet.34.1.401. URL: https://www.annualreviews.org/doi/10.1146/annurev.genet.34.1.401 (visited on 02/25/2025).

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

Samson, Franck and Sébastien Aubourg (June 25, 2024). "GBOT One Flew over the Ortholog's Nest." In: Jobim. URL: https://hal.inrae.fr/hal-04831777 (visited on 05/13/2025).

Scrucca, Luca et al. (2023). *Model-Based Clustering, Classification, and Density Estimation Using mclust in R.* Chapman and Hall/CRC. ISBN: 978-1-032-23495-3. DOI: 10.1201/9781003277965. URL: https://mclust-org.github.io/book/.

Steinegger, Martin and Johannes Söding (Nov. 2017). "MMseqs2 Enables Sensitive Protein Sequence Searching for the Analysis of Massive Data Sets." In: *Nature Biotechnology* 35.11, pp. 1026–1028. ISSN: 1546-1696. DOI: 10.1038/nbt.3988. URL: https://www.nature.com/articles/nbt.3988 (visited on 06/12/2025).

The Arabidopsis Genome Initiative (Dec. 2000). "Analysis of the Genome Sequence of the Flowering Plant Arabidopsis Thaliana." In: *Nature* 408.6814, pp. 796–815. ISSN: 1476-4687. DOI: 10.1038/35048692. URL: https://www.nature.com/articles/35048692 (visited on 01/16/2025).

Van de Peer, Yves, Steven Maere, and Axel Meyer (Oct. 2009). "The Evolutionary Significance of Ancient Genome Duplications." In: *Nature Reviews Genetics* 10.10, pp. 725–732. ISSN: 1471-0064. DOI: 10.1038/nrg2600. URL: https://www.nature.com/articles/nrg2600 (visited on 04/19/2024).

Van Dongen, Stijn and Cei Abreu-Goodger (2012). "Using MCL to Extract Clusters from Networks." In: *Bacterial Molecular Networks.* Ed. by Jacques Van Helden, Ariane Toussaint, and Denis Thieffry. Vol. 804. New York, NY: Springer New York, pp. 281–295. ISBN: 978-1-61779-360-8 978-1-61779-361-5. DOI: 10.1007/978-1-61779-361-5_15. URL: http://link.springer.com/10.1007/978-1-61779-361-5_15 (visited on 04/11/2024).

Vance, Zoe and Aoife McLysaght (Oct. 6, 2023). "Ohnologs and SSD Paralogs Differ in Genomic and Expression Features Related to Dosage Constraints." In: *Genome Biology and Evolution* 15.10. Ed. by Yves Van De Peer, evad174. ISSN: 1759-6653. DOI: 10.1093/gbe/evad174. URL: https://academic.oup.com/gbe/article/doi/10.1093/gbe/evad174/7287110 (visited on 05/09/2025).

Wang, Yupeng et al. (Dec. 2, 2011). "Modes of Gene Duplication Contribute Differently to Genetic Novelty and Redundancy, but Show Parallels across Divergent Angiosperms." In: *PLOS ONE* 6.12, e28150. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0028150. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0028150 (visited on 06/02/2025).

Yang, Z. (Apr. 18, 2007). "PAML 4: Phylogenetic Analysis by Maximum Likelihood." In: *Molecular Biology and Evolution* 24.8, pp. 1586–1591. ISSN: 0737-4038, 1537-1719. DOI: 10.1093/molbev/msm088. URL: https://academic.oup.com/mbe/article-lookup/doi/10.1093/molbev/msm088 (visited on 06/11/2025).

Yang, Z. and R. Nielsen (Jan. 2000). "Estimating Synonymous and Nonsynonymous Substitution Rates under Realistic Evolutionary Models." In: *Molecular Biology and Evolution* 17.1, pp. 32–43. ISSN: 0737-4038. DOI: 10.1093/oxfordjournals.molbev.a026236. PMID: 10666704.

Yang, Ziheng (1997). "PAML: A Pprogram Package for Phylogenetic Analysis by Maximum Likelihood." In: *CABIOS APPLICATIONS NOTE* 13.5, pp. 555–556. URL: http://abacus.gene.ucl.ac.uk/ziheng/pdf/1997YangCABIOSv13p555.pdf.

A. Appendix

A.1. More details on the FTAG-Finder pipeline on Snakemake

The FTAG-Finder pipeline is a workflow designed to find the families of paralogous proteic genes for a given species, based solely on its proteome, and to detect the Tandemly Arrayed Genes (TAG) in this proteome, using the positions of the genes. The workflow includes also a module to generate diverse lists of genes enabling facilitated statistical analysis on sets of genes.

Snakemake is a workflow engine written in Python (Köster et al. 2012; Mölder et al. 2021). Initially targetting bioinformatics analysis, it includes several capabilities that enables easier large scale data analysis workflows. For instance, Snakemake can submit jobs on high performance computing clusters via Slurm. It includes features enhancing the reproducibility of the computation, by fixing the software dependencies versions, via conda environments definition or a Docker / apptainer container.

The source code of the Snakemake version of FTAG-Finder is available on `https://gitlab.com/sortion/FTAG-Finder`.

A.1.1. Workflow steps

Step 1. **Proteome alignment all-against-all** The user can chose either the legacy `blastall` tool (Altschul et al. 1990), the current NCBI BLAST+ `blastp` tool (default choice) (Camacho et al. 2009), `diamond` (Buchfink et al. 2021), for a faster computation, or `mmseqs2` (Steinegger et al. 2017). This step takes as input the species proteome in FASTA format, and outputs the BLASTP hits in BLAST output format 7.

Step 2. **BLASTp hits sorting** A python scripts sorts the hits according to alignment position on the proteome.

Step 3. **Merging of BLASTp hits (optional)** A C program, named `mergeBlast` takes the sorted blast outputs and merges the hits that overlaps and computes the cummulated alignment coverage (the proportion of the sequence represented in the alignment) and alignment similarity percentage.

Step 4. **Extraction the homology graph** Based on the merged blast hits TSV file, a python scripts extracts a weighted graph in ABC format (column 1 and 2 corresponding to the adjacent proteic gene names, and the last third column corresponds to the weights measured by the highest bitscore among the BLASTp hits between the two genes). In this phase, the user can choose whether to keep the longest protein isoform only.

Step 5. **Clustering of the homology graph** To identify the gene families, the homology graph is clustered using either the Markov Cluster algorithm (Van Dongen et al. 2012) Walktrap algorithm ((Pons et al. 2005)), implemented with R packages `igraph` (Csárdi et al. 2025; Csardi et al. 2006) and `mclust` (Scrucca et al. 2023). A simple python script is also provided in case the user prefer to use a single-linkage clustering algorithm. This step results in communities of genes homologous one with the others, which constitutes what we call gene families.

Step 6. **Identification of the TAG** A python scripts then takes the position of genes – that can be automatically extracted from a Gene Feature Format (GFF) file, on the one hand, and on the other hand the clustering of genes into gene families, and identifies the genes belonging to the same TAG array. A TAG array of definition d being defined as an array of homologous genes separated by at most d unrelated genetic features (other proteic genes, or non-coding RNA genes for instances).

Step 7. **Generation of gene lists** Several python scripts are also provided to ease the generation of lists of genes of the following kinds:

- Intra-family pairs: pairs of genes belonging to the same family;

- Intra-TAG pairs: pairs of genes belonging to the same genes;
- Local pairs: pairs of genes separated by exactly n spacers, belonging to the same family;
- Successive pairs: pairs of genes separated by exactly n spacers, regardless the family they belong to.

The workflow has been tested on GNU/Linux (debian 12 Bookworm) and MacOS 15 Seqoia, as well as on a High Performance Computing (HPC) environment with Slurm on the IFB-core cluster.

A.2. A method to assess the differences between two training datasets

k-fold cross-validation To gain a better confidence on the test accuracy measure, one can use a k-fold cross-validation. Each data instance is assigned an index varying from 1 to k. Then the model is trained and tested k times each time with a training dataset corresponding to all the data instances but those indexed by k and tested on these data instances taken apart.

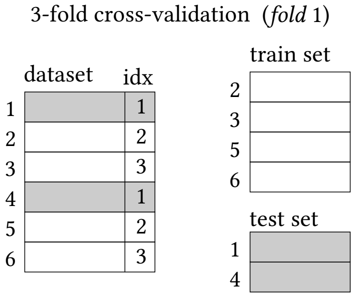


Figure A.1.: k-fold cross-validation

Same/Other/All K-fold cross-validation The Same/Other/All K-fold cross-validation (SOAK) method allows to run a k-fold cross-validation on subset of the data (Hocking et al. 2024). Each data instance is assigned, along with the fold index, an indicator for the subset the instance belongs. Then, the SOAK method proceeds in three steps: given two data subsets A and B: first, in the *same* step, it trains and test the model on k times, one time per fold. In the *other* step, it trains the model on the A subset and test on the fold coming from the B subset (hence the name). In the last *all* step, the model is trained on the whole training dataset, regardless the subsets A and B and tested using cross-validation as well.

This approach enables a comparison between the distribution of the descriptors and labels between the two data subset. Indeed, one of the underlying hypotheses of machine learning is that there exists a probability distribution that links both explanatory variable and predicted variable, and which is shared by unseen data. If there is a huge drop in performance on the other step it means that the model cannot generalize well on the data instance of subset B if it learned from subset A, hence that subset A and B do not share the same distribution.

A.3. A port of the SOAK method on Python

Two main programming languages are used among the community of machine learning: R and Python. Both have a huge set of bespoke libraries and implemented algorithms. However, some specific tools are implemented in only one of them. This is the case of SOAK (Hocking et al. 2024) and WENDA (Handl et al. 2019), which are unfortunately implemented in R and Python ecosystems respectively. In order to use the SOAK cross-validation procedure along with WENDA domain adaptation model, I wrote a Python version of SOAK, named `soakpy`. I made this function freely available on GitLab under the same license (GNU GPL v3) as the original `mlr3resampling` R package. The url of the repository is `https://gitlab.com/sortion/soakpy`.

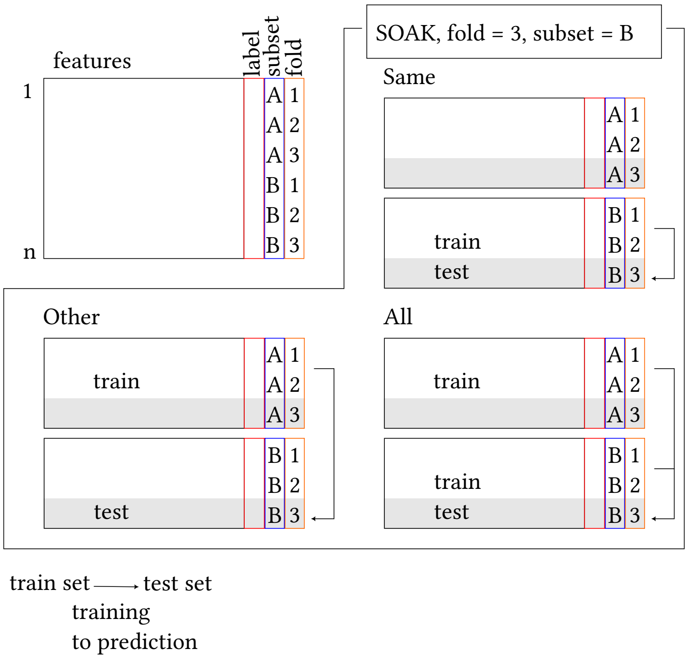


Figure A.2.: SOAK (Same/Other/All k-fold cross-validation) procedure scheme. The arrows indicate the learning process, from the training set to the prediction test set used in the phase. (Reproduced from (Hocking et al. 2024))