Data-Centric Deep Learning for Homology Detection in Promoter Sequences Through a Nextflow Pipeline

Víctor Fernández Oliveras

MSc in Bioinformatics

Supervised by Cristina Araiz Sancho & Cedric Notredame

Academic tutor: Cinta Pegueroles Queralt

02 July 2026

UAB
Universitat Autònoma de Barcelona

CRG
Centre for Genomic Regulation

Table of Contents

Abstract

Homology detection in ncDNA remains challenging because these regions tend to evolve rapidly and lack additional conserved information that can guide the alignment, such as amino acid sequences in coding DNA. This thesis investigates whether a deep learning model can generate alternative promoter representations that better reflect functional similarity than nucleotide identity. As a proof of concept, promoter regions from human–mouse one-to-one orthologous kinase genes were paired with matched RNA-seq expression profiles, using downstream gene expression as a proxy for promoter function, relying on the assumption that evolutionarily close sequences are more likely to retain similar functions. A data-centric approach was adopted by optimizing the data preprocessing with the objective of capturing as much underlying biological signal in the data as possible. Normalization strategies and expression similarity metrics were benchmarked against the expected conservation of ortholog expression, identifying quantile normalization and cosine similarity as the best-performing methods. A Siamese Neural Network was trained with contrastive learning, aiming to place functionally related promoters closer in embedding space and unrelated promoters farther apart. Classical sequence alignment failed to distinguish functionally similar from dissimilar promoter pairs, whereas the learned embeddings showed a modest but statistically significant separation on unseen test data (4.1% of the maximum separation), indicating that the model learnt regulatory/evolutionary signal contained in the promoter sequences. Although the current model is not yet a robust homology detection tool, it provides a proof of concept for embedding-based representations of divergent regulatory DNA. The work also demonstrates that benchmarked preprocessing choices were essential for model performance, since a non-optimized preprocessing configuration failed to recover meaningful signal. All analyses were automated in a reproducible and scalable Nextflow pipeline.

Objectives and Research Questions

The objectives and research questions of this project combine a biological point of view and a methodological focus.

The primary objectives are:

- To generate evolutionarily-meaningful sequence representations from ncDNA sequences (promoters) to improve homology detection using a DL model.
- To develop a reproducible Nextflow pipeline that leverages biological ground-truths to optimize both data preprocessing and model performance, ensuring the capture of biological signals

This master thesis addresses the following research questions:

- Is there any remaining evolutionary signal in the promoters that a DL model can effectively learn to distinguish between homologous and non-homologous sequences?
- Does optimizing data preprocessing through systematic benchmarking significantly improve model performance compared to using standard methods?

Project Overview

Problem Statement & Rationale

This project addresses the problem of detecting evolutionary relationships between non-coding DNA (ncDNA), which is extremely useful for genome annotation or evolutionary studies. As a proof of concept, we study promoters, non-coding regulatory regions that control the initiation of transcription and contribute to the spatial, temporal and tissue-specific regulation of gene expression. Aligning ncDNA (e.g. promoters) is, however, substantially more difficult than doing so in coding sequence. Coding regions benefit from additional conserved layers to guide alignment, such as amino acid sequence and protein structure. In contrast, ncDNA is not translated and tends to evolve faster. As a result, sequence identity resulting from classical alignment is rarely informative of the evolutionary relationships of the sequences.

Project Objectives & Scope

The conceptual inspiration for this work comes from the strategy used by Foldseek: improving protein structure alignment by transforming complex three-dimensional information into a one-dimensional sequence representation that can be easily compared. In this work, we aimed to convert promoter sequences into alternative representations whose distance was more informative than nucleotide sequence identity.

The biological assumption underlying the project is that DNA sequences sharing a closer evolutionary origin (e.g. human-mouse one-to-one orthologs) are more likely to retain similar functions than those with less direct common ancestry (e.g. human-mouse non-orthologs). In this framework, the expression similarity of the downstream gene is used as an approximate measure of promoter function, and therefore, the expression similarity between downstream genes is used as a proxy for functional similarity of promoter pairs.

Aware of the high level of noise and technical biases present in expression data (which are further compounded by mixing different studies, individuals, tissues, and species), as well as the numerous strategies available to address them, this work also aimed to evaluate the importance of a data-centric approach in deep learning by optimizing the data preprocessing steps, rather than selecting standard methods.

Methodological Workflow

Two benchmarking steps were performed in parallel relying on the biological ground-truth of closely evolutionarily related genes tending to present a higher functional similarity. Quantile normalization was the metric that maximized the correlation between human and mouse expression in orthologous genes; cosine similarity was the best metric for separating orthologous and non-orthologous expression similarity distribution.

Gene pairs with the highest expression similarity were labeled as positive, representing functionally related promoter pairs, while pairs with the lowest expression similarity were labeled as negative, representing functionally unrelated pairs. These labeled pairs were used to train a Siamese Neural Network. The model received two promoter sequences as input and processed them through local pattern scanning and feature dependencies analysis, to finally generate one embedding (vector) for each promoter. Training was performed with contrastive learning, so

that embeddings from positive pairs were pulled together while embeddings from negative pairs were pushed apart.

Outcomes & Key Contributions

The main result of the project is that the learned promoter embeddings by the model provided a better distinction between functionally related and functionally unrelated promoter pairs than classical sequence alignment: 4.1% of the maximum separation vs no separation. The current model is not yet a robust homology detection tool, but it provides a proof of concept for using deep learning to generate evolutionarily-informative alternative sequence representations of promoters.

The results of this work demonstrated that well-founded design choices for data preprocessing (the data-centric approach) were essential. When the pipeline was run using non-optimized preprocessing choices, the model failed to learn any generalizable sequence patterns. This supports the idea that the data quality is crucial in determining whether the model learns useful biological signal or noise; and that the optimum data processing strategies can be task-specific.

All analytical steps were automated in a Nextflow pipeline. This automation ensures reproducibility by making the full workflow executable from defined inputs and parameters in controlled computational conditions. The pipeline is therefore both a methodological result of the project and the computational framework used to generate its biological findings.

Acronyms

3Di	Three-dimensional interaction
AI	Artificial Intelligence
ANN	Artificial Neural Network
bp	Base pairs
CLR	Centered Log-Ratio
CREs	Cis-regulatory elements
DE	Differential expression
DL	Deep learning
DNA	Deoxyribonucleic acid
ENCODE	Encyclopedia of DNA Elements
FASTA	Text format for nucleotide/protein sequences
GC content	Guanine-Cytosine percentage
GWAS	Genome-wide association studies
ID	Identifier
KS	Kolmogorov-Smirnov test
LLMs	Large Language Models
lncRNA	Long non-coding RNA
MET	Matched Expression Tissues
ncDNA	Non-coding DNA
PCA	Principal Component Analysis
PC1	Principal Component 1
PC2	Principal Component 2
PDB	Protein Data Bank
Pfam	Protein Family Database
QN	Quantile normalization
ReLU	Rectified Linear Unit function
RNA	Ribonucleic acid
RNA-seq	RNA sequencing
SNN	Siamese Neural Network
SSD	Sum of Squared Differences
TATA box	Core promoter sequence motif
TMM	Trimmed Mean of M-values
TPM	Transcripts per Million
TSS	Transcription start site
UTR	Untranslated region

1. Background

1.1. Non-Coding DNA Importance and Alignment

The non-coding genome importance is widely recognized nowadays in agrogenomics[1], medical genomics[2], phenotype prediction[3]… Despite this, most current genomic workflows and analyses still eminently focus on the coding genome[2]. The vast non-coding regulatory landscape includes cis-regulatory elements (CREs) such as promoters and enhancers, as well as transcript untranslated regions (UTRs) and splice sites[1]. Among all these regulatory elements, in this work we focus on promoters. Promoters represent the beginning of gene transcription in the human genome, integrating the signal of multiple regulatory elements to sharply modulate gene expression[2] with precise temporal[4] and tissue- or cell-specific[3,5] execution.

Comparative genomics relies heavily on sequence comparison through sequence alignment as a foundational methodology for homology detection[6,7], functional annotation[5,8], phylogenetic reconstruction[9] or read mapping[10]. Aligning ncDNA presents significant challenges compared to coding DNA. The primary obstacle is that ncDNA tends to be far less evolutionarily conserved than coding DNA, as it usually evolves faster due to lower functional constraints[8]. For instance, regulatory regions present some degree of tolerance to the exact order, orientation and spacing of motifs[5], or even their total number[8]. Secondly, coding regions present more highly conserved biological elements (the amino acid chain, the protein three-dimensional structure or molecular function…) (Fig. 1) which are used in sequence-alignment-based algorithms to guide the alignment and improve homology detection. However, van Kempen et al.[7] explain that "detecting distant evolutionary relationships from sequences alone remains challenging", referring to amino acid sequences. Since non-coding DNA (ncDNA) is not translated, it lacks these highly conserved layers. Given that alignment algorithms struggle even within constrained coding regions, the complexity of aligning ncDNA is substantially increased

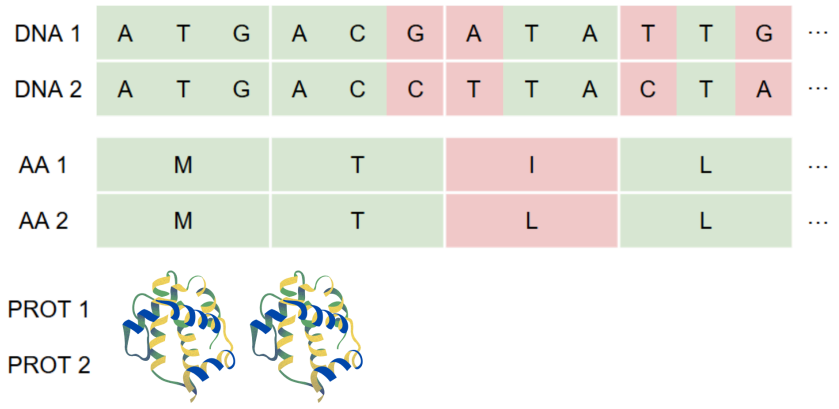

Figure 1: Schematic example of increasing conservation through the coding DNA biological layers: nucleotide sequence (4/9 mismatches), amino acid sequence (1/4 mismatches) and protein structure (identical).

1.2. Foldseek and Sequence Representation

When considering evolutionarily close sequences, protein structure similarity tends to be informative of the amino acid sequence identity, or vice versa. Informative, in this context, means that knowing the sequence similarity of a given sequence pair allows you to make an educated guess about the structural similarity. However, when this identity falls into the "twilight zone",

between 20-35%, the structure similarity signal gets blurred[11]. To overcome this sequence-level limitation, structural alignment tools look directly at spatial coordinates, but this approach is computationally demanding and scales poorly with massive databases[7]. Foldseek[7] addresses this scalability problem by discretizing complex three-dimensional structural data into a one-dimensional sequence representation, thereby transforming a computationally intensive spatial optimization problem into an efficient sequence alignment task.

Foldseek uses an Artificial Neural Network (ANN) that has learnt an alphabet composed of states called 3Di (three-dimensional interactions) which derive explicitly from spatial residue-residue proximity, and therefore they implicitly integrate complex structural, functional and evolutionary information. Once both the query and target protein structures are converted into their corresponding one-dimensional 3Di sequences, Foldseek aligns them using classical alignment algorithms[7].

Fundamentally, the underlying paradigm of this software is to transform a highly informative layer that is computationally difficult to align (three-dimensional spatial coordinates) into a representation that is easier to align (one-dimensional vector called embedding) while retaining the critical functional information of the structural level. This core idea inspires the framework of this project. Here, nucleotide sequence level that is highly difficult to align due to low evolutionary conservation (promoters) is transformed into an alternative representation (embedding) that integrates functional information derived from a higher informative biological layer (see 1.3. Gene Expression Profiles as Function for more) to enhance homology detection. See a summary of these ideas in Fig. 2. Recently, DNA language models have already been proven to capture these type of functional high-order sequence and evolutionary signals[2,3,5], for instance detecting TATA boxes, transcription start sites (TSSs) or polyadenylation sites[5].

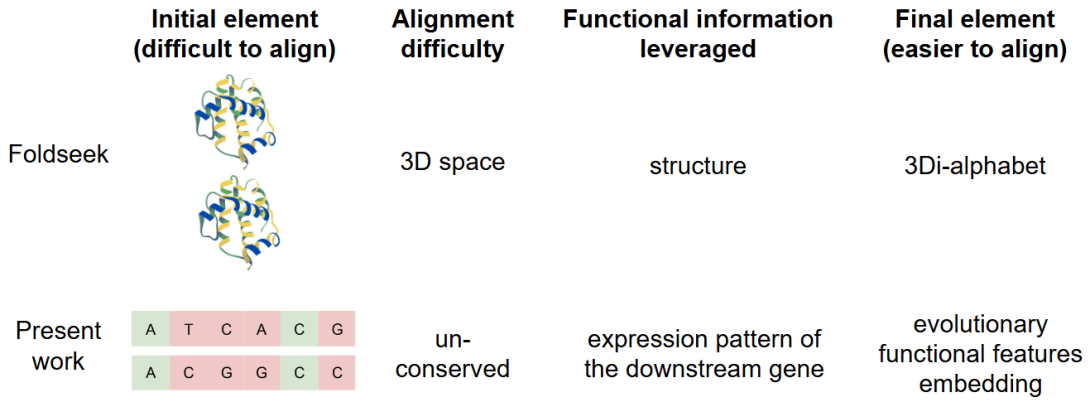

Figure 2: Parallelism between Foldseek and this project in problematic alignment, information leveraged and sequence representation. 3Di refers to 3D-interactions between residues.

While alternative sequence representations have been established to reduce computational overhead in large datasets, leveraging them to maximize signal retention in rapidly evolving regions is an entirely novel approach.

The idea of using alternative representations of nucleotide sequences for enhancing their alignment has already been used for addressing the computational demands of large DNA datasets. Both non-AI and AI-based approaches have been published:

- AI-free: Chen et al.[9] explain how "subsketching" methods use k-mers to "summarize a long sequence into a small set of representative fingerprints that can be rapidly compared in place of the original sequences for similarity estimation". The results of their work show a high

correlation between alternative representation similarity and original sequence similarity, suggesting that their "sketches" capture critical features of the sequence. However, this approach still presents inherent limitations, such as the difficulty in selecting an optimal value for the k-mer length parameter (k), as noted by the same authors.

- AI-based: Embed-Search-Align[10], Nucleotide Transformer[12] or Hyena-DNA[13] are some examples of deep learning models which enhance sequence alignment in large datasets by generating vectorial alternative representations of DNA to act as guides; currently rivaling conventional alignment-based algorithms[10].

However, to the authors' knowledge, leveraging these representations specifically to enhance signal retention in ncDNA or promoters has not yet been reported as a distinct objective.

1.3. Gene Expression Profiles as Function

Gene expression patterns are a fundamental property of a gene, as they are crucial to establishing tissue identity[4] and cellular context[2,3]. Promoters are in charge of regulating this expression[2,14] and therefore, gene expression profiles can be understood as an approximate measure of the biological function of a promoter. For these reasons, in this project, the expression profile of the downstream gene of each promoter is used as a proxy for promoter function to decide if a given pair of compared promoters is considered as functionally related or unrelated (Fig. 3). Profiles are measured using information of several tissues in order to accurately capture important biological features such as the tissue-specificity of the gene[15].

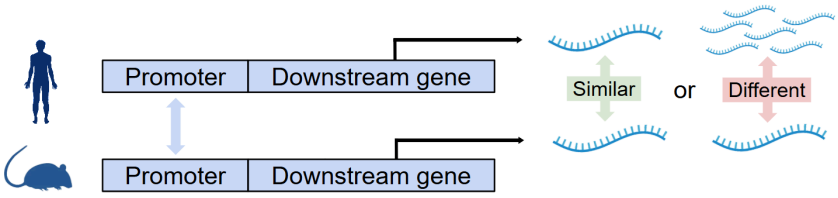

Figure 3: Diagram of our function similarity measurement. To evaluate the functional similarity of two promoters, we compare the expression profile of the gene regulated by each promoter. If it is similar, the pair of sequences is considered functionally related, otherwise it is considered functionally unrelated.

How to measure expression patterns and differences from RNA-seq data is not trivial. Several metrics, each treating expression information in slightly different ways with their positive aspects and its caveats, have been proposed[15]. See all the considered metrics for the benchmarking of this work in Table 3 (2.3.1. Metrics).

1.4. Orthologs Similarity as Baseline

Orthologs are genes that have diverged as a direct consequence of a speciation event. They are a subtype of homologous genes, as they originate from a single ancestral gene. Conceptually, homology is treated as a strict binary classification: two sequences either share a common ancestry or they do not. However, this discrete parameter can be transformed into a continuous one by treating it as the probability of being homologous. With this and under the Principle of Parsimony, it is conventionally assumed that biological sequences sharing a closer evolutionary origin are more likely to retain more equivalent functions[16]. Because here we use gene expression profiles as function for the promoters, we operate under the assumption that these profiles should be therefore fundamentally conserved between human-mouse one-to-one orthologs[15] (Fig. 4). Other types of orthologs are discarded for simplicity. We use this as a reference to evaluate

and benchmark the efficacy of various normalization and profiling methodologies in downstream analysis.

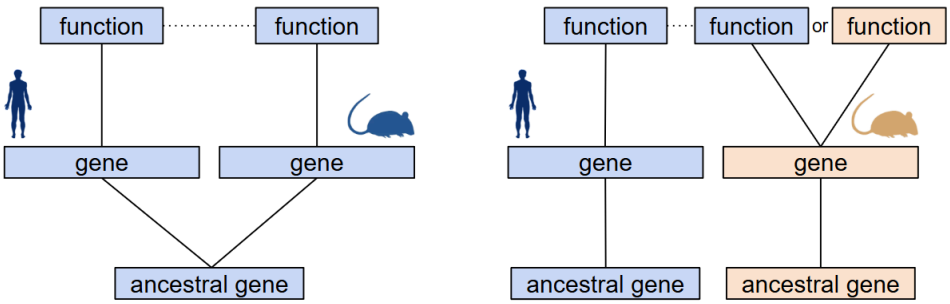

Figure 4: Diagram showing the relationship between common evolutionary ancestry and functional similarity, using genes as example. Mouse-human orthologs (left) are expected to present similar functions due to their shared evolutionary origin. Genes with no common ancestry relation (or too distant) will present either similar or dissimilar functions.

1.5. Normalization of the Data

The normalization of high-throughput biological data is a crucial step in the data processing procedure[17,18]. It is driven by two distinct reasons. From a computational perspective, models exhibit higher stability and performance when trained on data with certain mathematical properties, such as aligned distributions[19]. From a biological point of view, the primary objective is to preserve and reflect the underlying biological reality. This is the primordial perspective in this data-centric work:

"The better the data represents biology, the better the model will learn."

Orthologs expression data is translationally used across the whole project, and in consequence, technical variation should be eliminated, or at least minimized, in order to be used and compared across species[20,21]. This "variation" might include biases and/or differences in sequencing depth[4], library size[21], sequencing technology, read mapping[22], gene length or GC content[23], general noise, species-specific characteristics[20], etc. Moreover, each publicly available dataset or database could have been normalized following a different strategy[17]. And even without technical errors, one should be aware of the fact that the RNA-seq experimental outcomes are random variables following a certain distribution constrained by the compositional nature of the sequencing data[21], where the true biological state remains unknown[23]. All this makes the necessity for normalization something very clear.

The number of normalization methods is huge, each having optimal usage requirements[17,20]. The problem of deciding which normalization method is the best option for each analysis is still unresolved. A bad election might reduce performance in the downstream steps or even lead to errors and misinterpretations[17] or effects in non-intuitive ways[18]. Zhao et al.[17] quoted in 2020: "While the analyst (or biologist) is now aware that normalization is imperfect ["especially when the data does not meet the assumptions"], they are left non-the-wiser on best practices". Aware of this, here diverse widely-used normalization methods (described in 2.2.1. Methods) are evaluated on the orthologs functional similarity baseline to determine which of them all better represent the underlying biology of the data.

1.6. Deep Learning Modeling and Data-Centric Approach

Artificial intelligence (AI) and deep learning are widely applied across bioinformatics. Fundamentally, a deep learning model consists of a network of layers made of interconnected computational nodes (artificial neurons) that perform non-linear mathematical operations to extract patterns and statistical relationships from data, as an objective by itself or to perform predictions[24].

The weights of the connections between these nodes and the biases introduced by each of them represent the parameters of the network, which are optimized during training. Hyperparameters are features that are fixed (rather than optimized) during training[24], such as the learning rate or the percentage of randomly dropped out data in each iteration.

In this work, we implement a specific architecture known as a Siamese Neural Network[25] (SNN), utilized by prominent sequence-to-function models, such as PromoterAI[2]. A SNN takes two distinct inputs with a single label and processes them independently. The network generates an embedding for each entity of the pair and evaluates their similarity to calculate the loss (error) during training (Fig. 5). The objective of the model is pulling together embeddings of positively labeled pairs and pushing apart embeddings of negatively labeled pairs[25], an approach based on contrastive learning[26]. In this work, the input is a pair of promoter sequences and the label is 1 (functionally related, approximating homology) or 0 (non-functionally related, approximating non-homology), depending on the difference between their expression profiles.

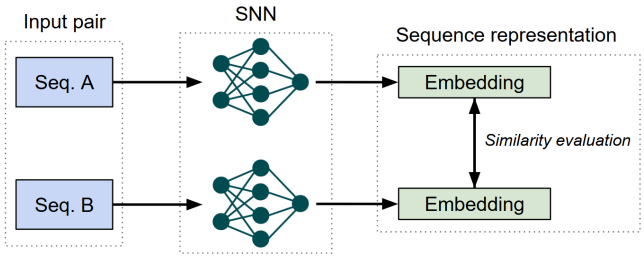

Figure 5: Diagram of a Siamese Neural Network (SNN) processing a pair of sequences. Each sequence enters to the network and is transformed into an embedding. The distance between these alternative vectorial representations is then compared to the label (positive or negative) to calculate the model performance.

To force these embeddings to capture the complex "syntax of regulation"[2] or "grammar of molecular evolution" from raw sequences, modern DL genomic models[2,3,10], and the one implemented in this work, rely on some fundamental architectural pillars to mimic the hierarchical behaviour of promoters, where local features are modulated by their broader genomic context[5]:

- **Convolution**: convolutional layers act as local feature detectors to scan for short, localized nucleotide patterns[3] like transcription factor binding motifs, which represent the "atomic units of gene expression"[5].
- **Attention**: attention mechanisms capture predictive-relevant dependencies between nucleotides and/or motifs without manual feature engineering[5]. Within these mechanisms, Transformer blocks hold substantial promise[10], with efficient scalability to detect long-range sequence interactions[3].
- **Pooling**: pooling layers compress multidimensional matrices. For sequence analysis, they can flatten the two-dimensional sequence feature maps into a one-dimensional embedding vector[10], reducing dimensionality and preserving the most prominent signals detected by the network.

Besides this network architecture, the performance of a model depends critically on the quality of the data[27], or in other words, as said, on how well these data represent biology. This data importance is being claimed by the scientific community. Jin[28] argues that "deep learning in biology requires a shift toward data-centric development" and "preprocessing and other data-related choices are not just supporting steps, but central drivers of model success". Whang et al.[27] emphasize "software engineering needs to be re-thought where data become a first-class citizen on par with code". For this reason, in this project we optimize the data normalization and expression profiling options, rather than treat them as secondary steps of the analysis.

1.7. Nextflow

Workflow management systems are essential in bioinformatics because modern omics analyses involve multiple software tools, intermediate files, parameter choices, computational environments...[29]. These factors can make analyses difficult to reproduce other researchers' results. In this context, reproducibility is not only a desirable feature, but a necessary condition for ensuring that data-based scientific results can be first of all verified, and then reused.

There exist several workflow managers, but the most important ones are Galaxy, Make/Snakemake and Nextflow[30]. Among them, Nextflow has become the preferred option, with its popularity increasing since the year it was created (Fig. 6).

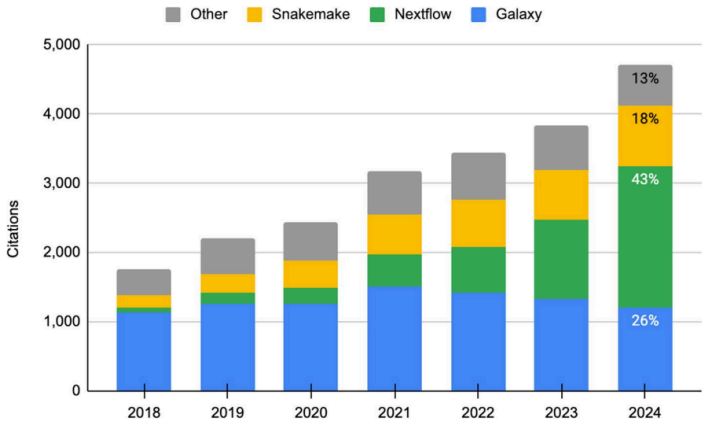

Figure 6: Number of citations for bioinformatics workflow managers between 2018 and 2024 in Google Scholar. Retrieved from Langer et al.[30].

This popularity is related to several practical advantages. Nextflow is designed to improve reproducibility, scalability and portability[29]. Its process-based and dataflow-oriented structure facilitates modular pipeline development, automatic task execution when inputs become available or containerization[28]. Moreover, Nextflow can be deployed in high-performance computing environments[30] and supports common batch schedulers such as SLURM[29], allowing pipeline tasks to be submitted to cluster infrastructures without rewriting the workflow logic.

The use of workflow managers has become essential for generating reproducible, large-scale scientific insights from complex biological datasets, and Nextflow is the preferred candidate for accomplishing this objective.

2. Materials & Methods

2.1. Data Selection and Gathering

This section describes the data acquisition process, ensuring the reproducibility of the downstream analysis. In case of any doubt, the scripts used for data gathering are available in this work's GitHub repository (https://github.com/victor-fdz/DeepBenchBioData), as well as the conda environment with the specific software versions.

2.1.1. Human-Mouse One-to-One Orthologs, Kinases

Initial gene selection was conducted using the Ensembl BioMart[31] interface (v116), with the GRCh38.p14 assembly. The gene set was restricted to kinase family (chosen because of their abundance and cross-tissue expression) by applying the Pfam ID PF00069. Results were filtered for human-mouse one-to-one orthologs (orthology_one2one flag) to simplify the analysis. After the filtering, 353 genes remained.

2.1.2. RNA-seq Data

Expression data for the selected genes was manually sourced from ENCODE[32] (v134.5-hotfix-1-g86e3e93a9), restricting to RNA-seq experiments. Due to the unbalanced availability of tissue samples between the two species, the mouse data was utilized as the primary constraint to determine tissue selection, with human data collected subsequently to match it. For each tissue fulfilling the filtering criteria in each species, a single sample was randomly selected and retrieved. Once both species data was retrieved, genes in which all the expression values were < 1 Transcripts per Million (TPM) were discarded, resulting in a final set of 309 genes.

In order to minimize confounding biological variables, mouse samples were filtered to adult (2–20 months of age) males. The search space in the human cohort was restricted to adult (45–80 years old) male non-diseased tissue samples. The specific sample ID for each tissue and species is available in Table 1.

Table 1: ENCODE sample identifier for the selected RNA sequencing experiments in human and mouse.

Tissue	Mouse sample	Human sample
Adrenal gland	ENCFF890GTZ	ENCFF783MER
Brain	ENCFF707MBR	ENCFF725USM
Muscle	ENCFF546IKJ	ENCFF296APA
Heart	ENCFF519HZD	ENCFF658ESI
Liver	ENCFF161ZNN	ENCFF677SKC

A minimum number of 5 tissues per gene was searched, as this number has been reported as the threshold for detecting the signal of tissue-specificity[15]. Moreover, testis tissue was discarded from the beginning because of its enrichment in expression outliers[15] and the large number of unique genes that it contains[4]. For the fifth tissue in mouse, although the hippocampus was available, it was discarded to avoid functional overlap with the selected brain sample (cerebral cortex). Kryuchkova and Robinson[15] reported in 2017 how considering closely functional tissues biases expression profile metrics. Instead, liver was chosen as a more functionally distant tissue. The liver sample met all criteria except for biological sex, which was defined as unknown.

2.1.3. Promoter Sequences, Gene Pairs & Alignment

Although the promoter length may vary in each gene, for the scope of this project promoters were defined with a fixed length (model requirement) of 500 bp (base-pairs) TSS-upstream region. These are expected to, at least, be enriched with non-coding promoter sequence[5]. Data was programmatically retrieved from Ensembl[33] (v116) and stored as FASTA files.

Each human gene was paired with each mouse gene, generating a 95481 pairs. All the pairs were globally aligned through Needleman-Wunsch algorithm with a -2 penalty for opening a gap and -0.5 for extending it, using the PairwiseAligner module of BioPython[34] (v1.87). Additionally, a negative control dataset was generated with the 95172 non-orthologous pairs (95481 - 309).

2.2. Normalization

2.2.1. Methods

A set of six normalization methods (Table 2) were applied to the retrieved TPM-normalized expression data in order to correct artifacts and experimental biases.

Table 2: Normalization methods used in the benchmarking of this project, with an example of a positive aspect and a disadvantage.

Name	Description	Advantage	Disadvantage
TMM, edgeR[22]	It removes genes with extreme log-fold-changes or expression levels, selects a reference sample based on upper quartile proximity, and scales read counts using weighted log-fold-changes.	Handles high dynamic ranges of expression values.	Very sensitive to filtering. Not designed for TPMs, but for read counts.
DESeq2[22]	Scales read counts using the geometric mean across samples.	Handles high dynamic ranges and reduces background noise.	Not designed for TPMs, but for read counts.
Quantile normalization (standard)[17]	Ranks expression values independently for every sample and assigns the average value of each rank position to all samples.	Completely aligns expression distributions across samples.	Very sensitive to filtering and aggressive in variability reduction.
Quantile normalization (sample-specific)[17]	Applies quantile normalization independently for each tissue.	Less aggressive in reducing variability between tissues.	Assumes no bias between tissues.
Log1p or $\log(x + 1)$[18]	Logarithmic transformation after adding a pseudocount to handle zero values. In this study, base 10 was used.	Filtering low-expressed genes is not necessary. Variance stabilization.	Reduces differences among low expression values.
CLR (Centered Log-Ratio)[23]	Uses the logarithm of the ratio between each expression value and the geometric mean of the sample.	Unit-agnostic.	Requires previous zero filtering or pseudocounts.

TMM and DESeq2 normalization methods expect raw read counts, not TPMs. Therefore, rounded TPMs were used as a sub-optimal but common approximation[22]. Additionally, data normalized by these two linear transformations was log-transformed afterwards to deal with extreme values. On the other hand, QN (both approaches) was implemented with a previous log-transformation, as their data transformation is non-linear.

2.2.2. Benchmarking

To integrate the multi-tissue information into a unified benchmarking framework, expression matrices were restructured so the original single entry for each gene pair was expanded into five distinct rows, with each row representing the expression of the gene pair in one single tissue. With this, we generated the two final random variables "Expression in Human" and "Expression in Mouse"(Fig. 7) that should correlate in orthologs.

		Human Expression Values					Mouse Expression Values				
Gene Human	Gene Mouse	Brain	Heart	Liver	Adrenal Gland	Muscle	Brain	Heart	Liver	Adrenal Gland	Muscle
Gen X_H	Gen X_M	B_H	H_H	L_H	A_H	M_H	B_M	H_M	L_M	A_M	M_M

↓

Gene Human	Gene Mouse	Expression in Human	Expression in Mouse	Tissue
Gen X_H	Gen X_M	B_H	B_M	Brain
Gen X_H	Gen X_M	H_H	H_M	Heart
Gen X_H	Gen X_M	L_H	L_M	Liver
Gen X_H	Gen X_M	A_H	A_M	Adrenal Gland
Gen X_H	Gen X_M	M_H	M_M	Muscle

Figure 7: Matrix restructure for multi-tissue benchmarking. A single gene pair entry is expanded into five entries, each representing the expression of the gene pair in one tissue, thereby generating the variables we wanted to evaluate: Expression in Human and Expression in Mouse.

In order to evaluate the dependency between our variables for each tissue in each gene pair, Pearson correlation (Equation 1) was selected as we aimed to evaluate positive monotonic absolute agreement in cross-species expression after normalization. To account for possible similarity inflation introduced by the normalization methods, the value was corrected by subtracting the correlation in non-orthologous pairs (expected as low because of their diversity in expression similarity) (Equation 2).

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} \tag{1}$$

$$\Delta\rho = \rho_{\text{Orthologs}} - \max\left(0, \rho_{\text{Non-Orthologs}}\right) \tag{2}$$

In Equation 2, the maximum between 0 and correlation in non-orthologs was used to avoid values >1 and/or inflating correlations because of spurious randomness.

Besides Pearson correlation, Spearman correlation was also considered as it is a standard alternative for measuring positive monotonic dependencies. However, it was discarded for two main reasons. Firstly, the normalization had the explicit goal of forcing the expression of the orthologs to be very similar in absolute values, rather than in relative position within the dataset. Secondly, de Siqueira et al.[35] showed that Spearman correlation performs better at detecting local correlations, which could be great to detect if there were expression outliers. However, the normalization is expected to correct for these genes and thus using Spearman correlation could be too permissive.

2.3. Gene Expression Profiling

2.3.1. Metrics

The seven gene expression profiling metrics included in the benchmarking of this project (Table 3) are divided in:

- **Pair-wise metrics**: evaluate matched-tissue information generating a single value for each gene pair.
- **Gene-wise metrics**: tissue-specificity metrics that offer a value for each species, so each gene pair has two values and their absolute difference is used for the benchmarking.

Table 3: Gene expression profiling metrics benchmarked. In all formulas, x_i represents the expression value for tissue i in first species, y_i represents the expression in the second species, and n represents the number of tissues.

Metric	Type	Description	Equation
Tau[15]	Gene-wise	Emphasis on the maximum expression value of the gene.	Equation 3
Gini Index[15]	Gene-wise	Measure of inequality in gene expression distribution across tissues.	Equation 4
Shannon entropy[15,36]	Gene-wise	Uncertainty of a gene's expression distribution.	Equation 5
SSD (Sum of Squared Differences)	Pair-wise	Squared sum of the differences between tissues in both conditions.	Equation 7
MET (Matched Expression Tissues)	Pair-wise	Number of tissues that match in expression status, either both expressed or both non-expressed.	Equation 6
Cosine similarity	Pair-wise	Angle-based similarity between the expression vectors.	Equation 8
Z-score + cosine similarity	Pair-wise	Cosine similarity after previous Z-score standardization.	Equation 9

$$\tau = \frac{\sum_{i=1}^{n}(1 - x_{i'})}{n-1}, \quad x_{i'} = \frac{x_i}{\max_{1 \le i \le n}(x_i)} \tag{3}$$

$$\text{Gini index} = \frac{n+1}{n} - \frac{2\sum_{i=1}^{n}(n+1-i)x_i}{n\sum_{i=1}^{n}x_i} \tag{4}$$

$$H = -\sum_{i=1}^{n} p_i \log_2(p_i), \quad p_i = \frac{x_i}{\sum_{i=1}^{n} x_i} \tag{5}$$

$$\sum_{i=1}^{n} ((x_i \geq 1 \wedge y_i \geq 1) \vee (x_i < 1 \wedge y_i < 1)) \tag{6}$$

$$\text{SSD} = \sum_{i=1}^{n} (x_i - y_i)^2 \tag{7}$$

$$\text{cosine similarity} = \frac{X \cdot Y}{\|X\| \, \|Y\|} \tag{8}$$

$$z_i = \frac{x_i - \mu}{\sigma}; \text{z-score cosine similarity} = \frac{Z_X \cdot Z_Y}{\|Z_X\| \, \|Z_Y\|} \tag{9}$$

2.3.2. Benchmarking

To evaluate the efficacy of each profiling strategy, the benchmarking workflow quantified the statistical separation between the distributions of true orthologous pairs and non-orthologous pairs. Under the evolutionary assumptions of this project, true orthologs should have highly similar profiles, whereas non-orthologs are expected to present more dispersion in values, with a lower average value among pairs. With this objective, a two-sample Kolmogorov-Smirnov (KS) test was applied. This test evaluates the null hypothesis that two independent samples are drawn from the same continuous distribution, without assuming a specific underlying parametric distribution. Its statistic KS indicates the longest vertical line between the cumulative distributions of the two samples.

2.4. Deep Learning Model

2.4.1. Labeling, Splitting and Encoding

Expression data was normalized and gene pairs were profiled with the best-performing methods identified in the benchmarking: Quantile normalization and cosine similarity. From the ~95000 gene pairs sorted by expression similarity, top 10000 pairs were labeled as positive (functionally related) and bottom 8000 were labeled as negative (functionally unrelated). The rest of the pairs remained with the label "undefined".

The original 309 orthologs dataset was split into training (70%), validation (15%) and testing (15%); so only the pairs in which both genes were assigned to the training set were used for training, and the same for validation and testing (Fig. 8); thereby preventing performance inflation by entity leakage.

Gene Human	Gene Mouse	Gene-Wise Splitting	Pair-Wise Splitting
gene A_H	gene B_M	Training	Training
gene B_H	gene C_M	Validation	Training
gene B_H	gene C_M	Validation	Validation

Figure 8: Splitting strategy to prevent entity leakage. In this three-pair example, pair-wise splitting allows mouse gene C to leak from training to validation sets. Gene-wise splitting prevents this by ensuring each gene of each species is restricted to a single set.

Sequences were encoded using basic one-hot-encoding, with A = [1, 0, 0, 0], C = [0, 1, 0, 0], T = [0, 0, 1, 0] and G = [0, 0, 0, 1].

2.4.2. Model, Contrastive-Loss Function and Hyperparameters

The model was implemented in PyTorch[37] (2.11.0) and trained by Adam optimizer utilizing a contrastive-loss function, a framework well-established for sequence classification tasks in genomics deep learning[10]. Contrastive-loss optimizes the network by minimizing the embedding distance between positive pairs while driving the embedding distance of negative pairs apart up to a specified margin[26]. The loss function is mathematically defined as:

$$L = hd^2 + (1 - h)\text{ReLU}(m - d)^2$$

where h is the binary homology label (1 for functional equivalence, 0 for functional difference) and m is the preset margin (fixed to 1), and:

$$d = 1 - \text{Cosine similarity}$$

where Cosine similarity ranges $[-1, 1]$ and therefore d ranges $[0, 2]$.

In Large Language Models (LLMs), cosine similarity is widely used to measure semantic similarity across vector spaces[10]. By analogy, this metric is employed here to evaluate functional evolutionary syntax similarity between the representation of the paired promoter sequences. This usage of cosine similarity is independent of its role as profiling metric.

After hyperparameter optimization through Optuna[38] (v4.8.0) for 300 trials, the hyperparameters evaluated were determined as: three parallel tracks with kernel sizes of 6, 10, and 20 nucleotides; learning rate of 0.0005, weight decay of 0 and dropout rate of 0.1.

2.5. Pipeline

The automated Nextflow (v26.04.3) pipeline developed in this study represents both a core methodological output of the project and the computational framework used to generate the biological results. A description of its characteristics is detailed in 3.5. Nextflow Pipeline. The specific production run of the pipeline used to generate the results of this work was Git commit ed453d9 with the following command:

```
nextflow run main_full_pipeline.nf
-c conf/full_pipeline.config
-profile local
--input data/new_expression_data.txt
--name final_results
--outdir results/final_results
--promoter data/kinases_promoter_alignment.tsv
--human_fasta data/promoter_kinases_human.fasta
--mouse_fasta data/promoter_kinases_mouse.fasta
--labeling rank_labeling
--split_mode anti_leakage
--val_frac 0.15
--test_frac 0.15
--n_pos 10000
--n_neg 8000
--epochs 40
```

```
--small_kernel_size 6
--medium_kernel_size 10
--large_kernel_size 20
--dropout 0.1
--learning_rate 0.0005
--weight_decay 0
```

3. Results & Discussion

3.1. Normalization

3.1.1. Benchmarking: Quantile Normalization

With our goal being the comparison of expression patterns across species, our first objective was to identify the most suitable metric for such comparison. This task is made especially complex because absolute expression values were obtained from different experiments, individuals and protocols; and with a single sample per tissue (see 2.1. Data Selection and Gathering). Technical biases introduced by these variations should be minimized by the normalization method. Therefore, we searched for the method that maximized the correlation between the expression in human and the expression in mouse in orthologs, as it is expected to be high due to their common origin and therefore tendency to similar function. We corrected the correlation subtracting the correlation in non-orthologs (very low correlation expected due to diverse functional similarity), yielding the $\Delta\rho$ (Equation 2). With this, we aimed to search the normalization method that better represented the underlying biology of the data.

The difference between the correlations in orthologs and non-orthologs showed that quantile normalization (QN) achieved the highest benchmarking performance, with a strong value of $\Delta\rho = 0.668$ when considering all the tissues together. This aligns with its role as a standard component of analytical pipelines for high-throughput genomic data[3,17] and represents a significant improvement over the original TPM-normalized data, which produced a $\Delta\rho$ of 0.194.

The diverse normalization methods presented shared tissue-wise patterns (Fig. 9). The brain exhibited the highest average correlation, followed by heart and adrenal gland, which maintained relatively high overall correlations. Muscle and liver exhibited the lowest values. These common tissue-wise patterns indicated that QN was not introducing any tissue-specific bias.

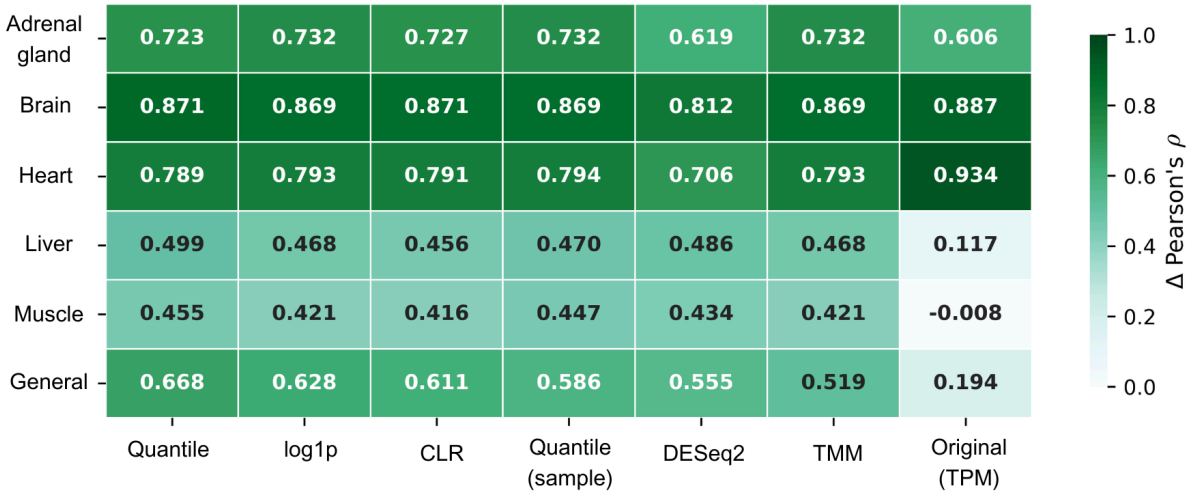

Figure 9: Heatmap showing the increment of Pearson correlation (orthologs - non-orthologs) in each tissue after each normalization method. "Original" refers to the initial TPM-normalized data. Sorted from left to right by decreasing performance in the benchmarking (using all tissues "General" value).

See the underlying visual relation of each gene pair for the selected method (QN) in the "General" tissue category (all combined), as well as the removal of outliers in Fig. 10. Find the rest of the plots in Fig. A1. In non-orthologs, just QN (sample-specific strategy) presented some

correlation ($\rho_{\text{Non-Orthologs}} = 0.130$). The rest of methods exhibited extremely low, even negative, correlations (Fig. A2).

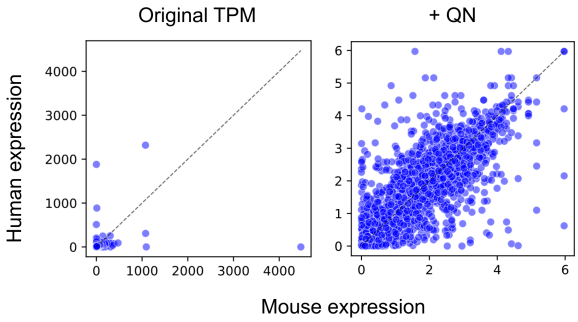

Figure 10: The graphs show very low correlation between mouse and human TPMs in the 5 considered tissues and a strong correlation after QN. Each dot is a gene pair.

3.1.2. Interpretation of the Expression Correlation

In order to interpret and understand the correlations resulting from this benchmarking, some discussion is needed here. For a given orthologous gene pair, the relationship between human and mouse expression measurements can be modeled through the following simple equation:

$$\text{Expression}_{\text{Human}} = \text{Expression}_{\text{Mouse}} \pm \text{Deviation}$$

Where the total deviation is a function of both biological variance and technical artifacts:

$$\text{Deviation} = \text{Variance}_{\text{Biological}} + \text{Error}_{\text{Technical}}$$

The biological variance element is driven by two main factors:

- **Dataset contamination**: the orthologous dataset functions fundamentally as an ortholog-enriched dataset, as we cannot ensure all gene pairs labeled as orthologs are in fact orthologs in real evolutionary history.
- **Evolutionary divergence**: true orthologs have evolved independently since the speciation event, allowing for functional divergence.

The primary objective of a normalization method is to minimize the technical error component. However, standard normalization techniques are rarely capable of eliminating all technical biases completely, and sometimes they even introduce method-specific mathematical artifacts. For this reason, we took advantage of the different expected $\text{Variance}_{\text{Biological}}$ element between orthologs (low, as their expression tends to be more conserved) and non-orthologs (high, as some of the pairs will offer similar expression and others will not) to minimize this potential error, through a corrected correlation $\Delta\rho$ (Equation 2). Consider an extreme, malicious case: a normalization method that forces every expression value to a constant value of 10. This would yield a perfect dependency between orthologs. Calculating $\Delta\rho$ controls for this artifact because the non-orthologous control dataset would also present complete dependency, reducing the net performance score to 0. This safeguard was particularly relevant for very aggressive methods such as sample-specific QN, which would have been the selected option if no correction was applied ($\rho_{\text{Orthologs}}$ was 0.715 and dropped to 0.586 after correction).

With the above in mind, we can better understand the correlation values used in this benchmarking. Each obtained correlation can be interpreted as:

$$\rho_{\text{Expression}_{\text{Human}},\text{Expression}_{\text{Mouse}}} = 1 - (\text{Biological}_{\text{Variance}} + \text{Technical}_{\text{Error}} - \text{Normalization}_{\text{Bias}})$$

Note that in this formula, $\text{Biological}_{\text{variance}}$, $\text{Technical}_{\text{Error}}$ and $\text{Normalization}_{\text{Bias}}$ refer to the variance introduced by these elements, not the absolute value of the previously explained elements.

The fact that the correlation obtained with QN was 0.668, so less than 1, is not considered as a failure of the method or a lack of performance. From the remaining quantity until 1, most of it is expected to be biological variance, with a smaller fraction being technical error (corrected by normalization) and nearly-negligible normalization bias (corrected by $\Delta\rho$). The exact proportion of each component is unknown, but the benchmarking framework was designed to minimize the latter two.

About the filtering strategy applied, we are aware of its potential effect on the results. Lin et al.[22] pinpoint that the filtering strategy and timing affect the results of normalization, especially in QN.

3.1.3. Data Structure Shift After Normalization

Our expression data should present variance because of real biological diversity of the genes' expression, rather than because of technical biases that introduce species- or tissue-specific variations. In order to evaluate if our data presented these biases and if so, normalization could correct them, a Principal Component Analysis (PCA) was executed to measure global variance distribution before and after QN.

In Fig. 11 one can observe how, before normalization, the data was divided in sub-populations, one per species, contrary to the scenario after normalization, where it forms a single cluster with no species stratification, showing how the normalization process removed the species-specific bias. QN induced an important magnitude and direction shift to the main contributors to the Principal Component 2 (PC2): brain and liver. This change coincides with the loss of the species-specific bias, suggesting that these tissues were the main contributors to the technical artifact observed between species. The distributions of the expression values in each tissue and species (Fig. 12) also show that QN aligned the distribution values across species and tissues, a data feature reported as a good-practice for deep learning[19]. This comparative data structure analysis strengthens the good results in cross-species expression correlation after QN.

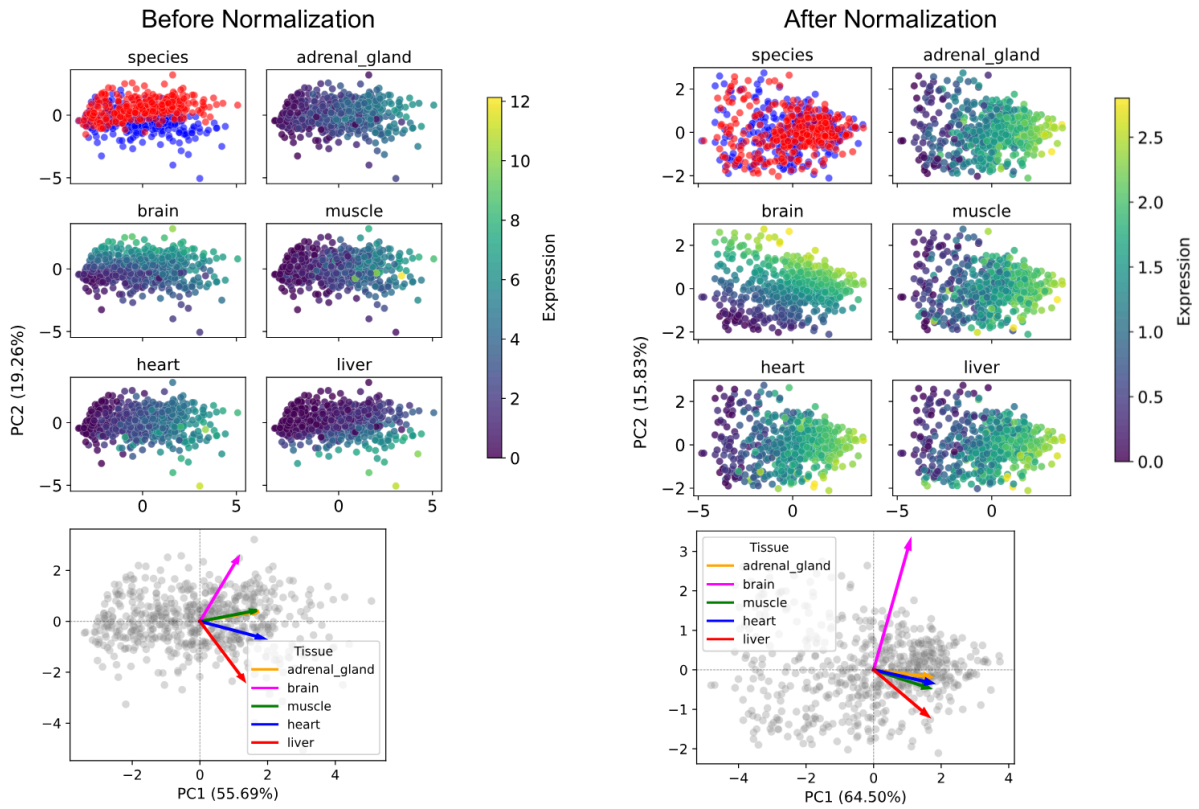

Figure 11: PCA of genes based on their expression values. Dots colored by species (red for mouse, blue for human) and by the expression value in each tissue. Initial TPM-normalization in the left and with extra QN in the right. Loadings of the PCA at the bottom, where each arrow represents the magnitude (length) and the direction (arrowhead) of the contribution of each tissue to the Principal Component.

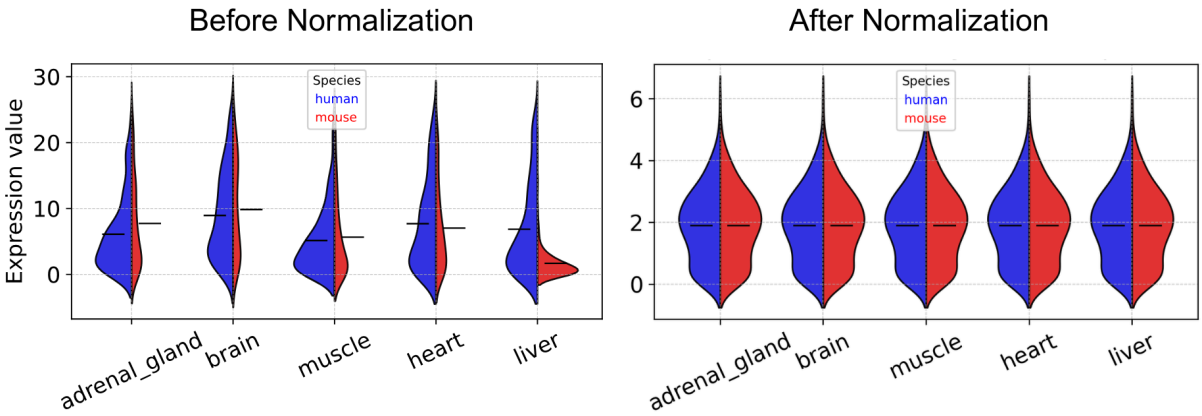

Figure 12: Violin plots showing the distribution of the tissue expression values by species and by tissue. Initial TPM-normalization in the left (genes with >30 TPMs were filtered out for visualization) and with extra QN in the right.

3.2. Gene Expression Profiling

3.2.1. Benchmarking: Cosine Similarity

With the objective of measuring the functional similarity of each pair of promoters in the most biologically reliable way, we evaluated the capacity of several expression profiling metrics to maximize the difference between the expression similarity values in orthologs (higher average

functional similarity expected due to their shared evolutionary origin) and in non-orthologs (lower average functional similarity expected due to more distant evolutionary origin). The Kolmogorov-Smirnov (KS) test was used to quantify the statistical separation between the two distributions, with a higher KS statistic indicating a greater separation.

Cosine similarity generated the most differentiated distributions between orthologous pairs (higher average cosine similarity, so higher average expression similarity) and non-orthologous pairs (lower average cosine similarity, so lower average expression similarity) (Fig. 13). It yielded the highest Kolmogorov-Smirnov statistic (KS = 0.257) and the highest statistical significance (p-value = 1.82e-18). The results for all the evaluated profiling metrics are summarized in Table 4 and graphically represented in Fig. A3.

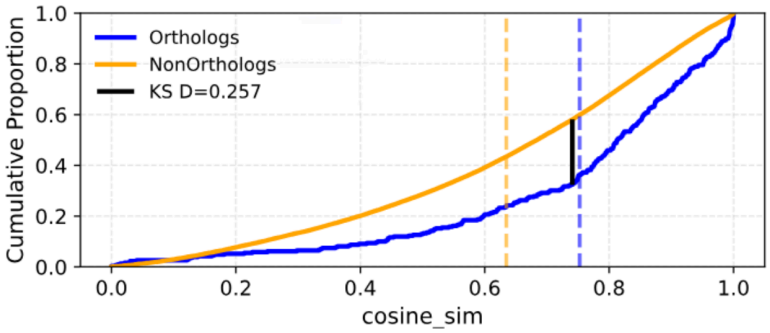

Figure 13: Cumulative plot for expression similarity values measured as cosine similarity for orthologs and non-orthologs. Discontinuous vertical lines show the mean value of each distribution and the black line shows the KS statistic (higher vertical difference between distributions).

Table 4: Kolmogorov-Smirnov test results for gene expression profiling metrics. Metrics are ranked by decreasing performance.

Metric	KS statistic	p-value
Cosine similarity	0.257	1.82e-18
Gini Index	0.233	3.21e-15
Shannon entropy	0.232	5.35e-15
SSD	0.222	7.44e-14
MET	0.192	1.92e-10
Cosine similarity + Z-score	0.169	3.77e-08
Tau	0.166	6.51e-08

3.2.2. Possible Sources of Confounding

Although cosine similarity showed the highest preservation of the underlying biological signal under our ground-truth, this relative best performance does not imply absolute accuracy. As with any other metric, cosine similarity possesses inherent mathematical limitations. An example inspired in Chen et al.[9]: consider two gene expression values in 2 species (A and B) made by 2 tissues; A = [1, 1] and B = [5, 5]. They have a perfect cosine similarity of 1.0, because both vectors project along the exact same spatial trajectory (Fig. 14), despite a severe difference in expression magnitude.

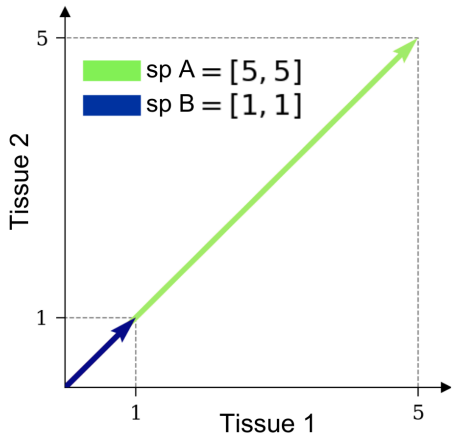

Figure 14: Example of possible confounding when using cosine similarity as a profiling metric. Two species with a different 2-tissue profile have the same cosine similarity value. Expression vectors are represented as arrows.

Furthermore, the non-orthologous dataset is expected to contain a broad dispersion of expression similarity values, including pairs with high similarity and pairs with low similarity. Therefore, with the Kolmogorov-Smirnov test we could have underestimated profiling metrics, because this test searches the maximum difference between the cumulative distributions, which can be influenced by the presence of highly similar pairs in the non-orthologs dataset. However, this underestimation would be common to all methods, so it does not represent a problem in the benchmarking process.

3.3. Sequence Representation Through Deep Learning

3.3.1. Embedding Distance Outperforms Sequence Similarity

In sequence identity, the average difference between functionally similar and functionally different pairs was −0.02, with higher similarity for negative pairs (48.55%) than for positive pairs (48.53%), showing the lack of capacity of sequence alignment to detect functional similarity in these data. Alignment score also failed to capture functional signal, yielding a lower score for positive promoter sequences (-26.26) than for negative promoter pairs (-25.80).

In order to generate the most informative alternative sequence representations of the promoters, or in other words, embeddings that are highly close in functionally related sequences and very distant in functionally different sequences, we built a SNN and trained it with the normalized and profiled data. Promoter sequences were used as input and their label (positive for functionally similar and negative for functionally distant) was defined after their expression similarity (see 2.4.1. Labeling, Splitting and Encoding for more details).

With the objective of enhancing model performance, an initial hyperparameter exploration was conducted to find a suitable set of hyperparameters, resulting in kernel sizes = {6, 10, 20}, learning rate = 0.0005, weight decay = 0 and dropout = 0.1. To generate the meaningful embeddings, the model was trained and then evaluated by calculating the difference between the average embedding distance in positive pairs and negative pairs. The distance between embeddings was calculated as $1 - \text{cosine similarity}$. This resulted in a difference of 1.380 in the training set, with a reduction to 0.070 in the validation set, indicating a clear overfitting to the training set (Table 5). However, even as the model fit training set-specific noise, it simultaneously captured biological signals that enabled some embedding separation between positive and negative pairs in the validation set.

Table 5: Absolute values and differences of average embedding distance in positive pairs and negative pairs in the 3 subdatasets (training, validation and testing).

Subdataset	Positive pairs	Negative pairs	Difference
Training	0.048	1.428	1.380
Validation	0.200	0.270	0.070
Testing	0.151	0.233	0.082

The generalization capacity of the model was tested by using the network to predict functional relations between unseen pairs of promoters, and then, measuring the previously mentioned difference. The application of the model to the test set offered similar results to the validation set, generating embeddings with slightly higher distances in negative pairs (0.233) than in positive pairs (0.151), with a difference of 0.082 (Fig. 15).

A one-sided t-test was performed to test higher average distance in negative pairs than in positive pairs, which was statistically significant (p-value = 3.79e-6). In relative values, a difference between average embedding distances of 0.082 represents 4.10% of the maximum achievable distance ($d \in [0, 2]$). This outcome reflects a completely new signal that was not detected by sequence alignment, as it did not yield higher average sequence identity in positive pairs than in negative ones. These results suggest that the network had learnt evolutionarily-meaningful generalizable patterns from the nucleotide sequences. Overall, the results of the model help distinguish between functionally similar and functionally distant promoters.

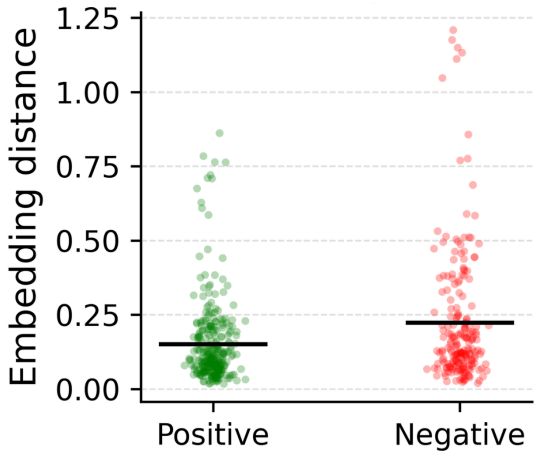

Figure 15: The graph shows the tendency of negative pairs (functionally unrelated promoters) towards higher embedding distances, and vice versa for positive pairs (functionally related promoters). Horizontal lines represent the averages.

As another proof of the utility of the learnt patterns during training, promoter pairs labeled as "undefined" (neither positive nor negative) presented an average embedding distance of 0.173, an intermediate value between positive (0.151) and negative pairs (0.233). This result is consistent with the expectations, as this group contains pairs with lower expression similarity than positive pairs, but higher than negative pairs. Moreover, this suggests a modest capacity to resolve a continuous gradient, beyond merely separating the extreme pairs.

3.3.2. Factors Limiting Higher Performance

The reasons for the model not achieving higher performance might be diverse. Three interesting possibilities are:

- **Lack of Context**: in eukaryotic genomes, gene expression regulation is heavily influenced by long-range chromatin interactions[5]. Because of the limited sequence context window processed by the current architecture (500 bp and no downstream context), the model cannot capture these distant dependencies. Despite this, state-of-the-art DNA Language Models had their best results in expression predictions mediated by variants proximal to the promoter (in PromoterAI[2]) and specifically close to the TSS (in AlphaGenome[3]). Moreover, larger promoters would imply a huge increase in computational resources, as Transformer blocks (used in the model) demand increases quadratically with the input length[10].
- **Scaling Laws and Data Curation**: deep learning scaling laws demonstrate that model performance improves with substantial increase in data size, computational resources and model parameters[39]. For instance, Foldseek[7] achieved high accuracy by leveraging a massive number of high-quality structural predictions from AlphaFold[40] and experimentally-resolved structures from Protein Data Bank[41] (PDB). In this sense, our model could lack of enough data or model size. Moreover, while this framework attempted to curate its data to the maximum extent, it might still be suboptimal.
- **Spectral Bias**: the training process suffered a rapid convergence within the first few epochs (Fig. A4). This suggests a strong influence of an inherent feature in deep learning models optimized with Adam, the spectral bias[42]. Rather than learning all features simultaneously, neural networks preferentially capture low-frequency, globally prominent patterns first[42]. The model's capacity to discover deeper complex insights could be limited by this phenomenon.

3.4. Importance of the Data-Centric Approach

In order to measure the surplus value of the optimized data preprocessing framework compared to a naive selection of random, previously known or literature-based methods, the complete analytical pipeline was re-executed by bypassing the benchmarking and enforcing a baseline preprocessing configuration of CLR normalization coupled with Gini Index as the expression profiling metric.

The model was retrained using this alternative dataset, with a previous hyperparameter optimization in order to make results comparable with the optimized data preprocessing framework. The average embedding distance for positive pairs was higher (0.333) than the average distance for negative pairs (0.310), contrary to the objective under our biological expectations.

This demonstrates that the input data failed to accurately reflect biological reality, illustrating how standard, unbenchmarked data processing choices can severely degrade downstream analysis or, in this case, even completely nullify the model's predictive power. Without the data-centric approach, our model would be completely unable to capture biological signal, making it completely useless for functional similarity detection.

3.5. Nextflow Pipeline

As a methodological outcome of this project, all the processing and analytical workflow was fully automated with a centralized Nextflow pipeline. A diagram of the pipeline is available in Fig.

16. The core scientific computing is written entirely in Python (v3.11.11) using standard data science libraries. The workflow operations, I/O management, and execution orchestration are handled by Nextflow, inheriting relevant features, such as argument tracking (every execution generates a permanent execution log) or the generation of a graphical execution report and information flow diagram.

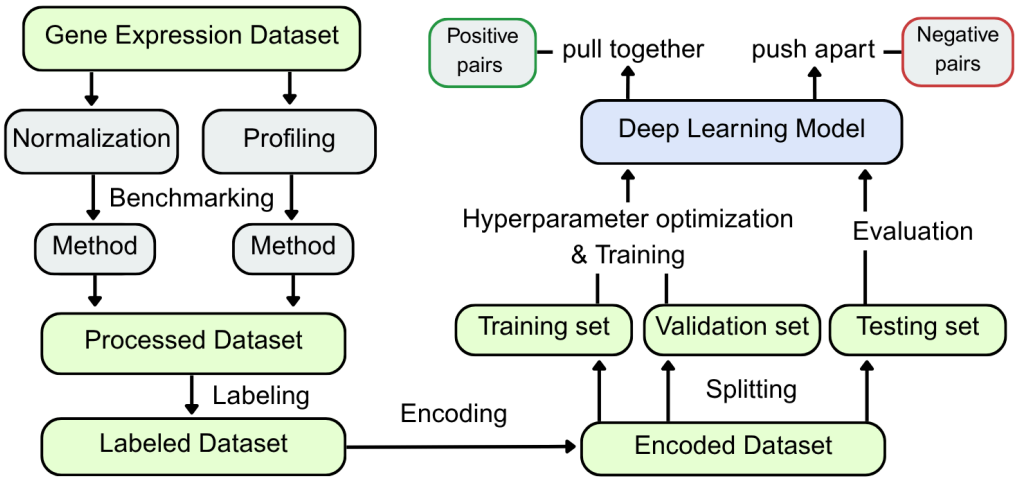

Figure 16: Diagram of the pipeline steps and information flow.

To ensure scalability, data normalization and expression profiling are benchmarked as independent modules, mitigating the growth of the combinatorial space so these pipeline steps can be extended with additional methods in a computationally cheaper way. The addition of new methods just requires the hard-coding of the new functions, following the naming examples and adding the names in a list (step by step detailed in the GitHub repository).

The flexibility of the pipeline is a core functionality for the user. Selecting a different correlation metric or reference tissue for benchmarking, deciding an unbalanced specific number of positive and negative pairs or changing the dimensions of the embeddings, are just some of the alternative options the pipeline offers. This pipeline can be leveraged to explore other expression datasets, tissues, species or gene families; whereas it is not designed as a general-purpose RNA-seq analysis tool.

Beyond its optimization utility, it can serve as a framework for studying how preprocessing variations alter the way a deep learning model learns, an area of active investigation within the genomics community[5,17].

Finally, although completely functional and reusable, the pipeline is still in a developing stage. Some of the features, modules and parameters can still be extended and improved. Furthermore, not all the steps of the workflow are optimized, only those with a distinguishable biological ground-truth. For instance, data labeling strategy was not optimized as its evaluation relies on topological optimization, which falls outside the scope of this project.

4. Limitations & Future Research

Some aspects limit the current work. Firstly, this analysis was restricted exclusively to the kinase protein family. The generalization capacity of the model across other protein families remains to be evaluated. In this line, we used promoters as a proof-of-concept of ncDNA, so this analysis should be extended to other regulatory regions to evaluate its applicability to functional ncDNA in general.

The pipeline remains an active development project and exhibits limitations. The optimization of the data preprocessing and the deep learning model was independent, as optimizing them simultaneously would represent a complex computational scalability problem. An interesting approach to this issue is the nf-core[43] pipeline STIMULUS (https://nf-co.re/deepmodeloptim/dev/), which is designed to address this challenge but is also at a development stage.

The performance of the model in generating informative embeddings is still moderate, maybe due to data quantity/quality constraints or architecture limitations. Although the generalization to unseen pairs of sequences suggests that the model is capturing genuine evolutionary syntax, implementing interpretability methods would be necessary to identify the specific patterns driving predictions.

The long-term objective of this methodology is to scale beyond simple binary classification. By refining the resolution of the sequence embeddings, this approach would aim to serve as the foundation for an embedding-based alignment tool through an alternative evolutionary alphabet capable of scoring evolutionarily-meaningful similarity between highly divergent DNA regions.

5. Conclusions

The Siamese Neural Network successfully generalized to unseen sequences, suggesting that the model captured evolutionary/functional syntax. The difference between average embedding distances in functionally similar and functionally different gene pairs remains modest (0.082) but statistically significant, as well as higher than in sequence identity (no difference). Although the model is still not a robust way to measure homology from sequence data, it offers an increase in the capacity to classify promoter sequence pairs in functionally related and unrelated compared to nucleotide sequence alignment. This represents a promising starting point for future research.

A comparison between optimized data processing (QN + cosine similarity) against naive method selection (CLR + Gini Index) demonstrates that informed benchmarked decisions guided by biological ground-truths are absolutely crucial to ensuring data integrity for model learning and meaningful downstream analytical results, thereby underscoring the importance of the data-centric approach in deep learning.

The Nextflow pipeline enables automatic optimization of specific data preprocessing decisions and model optimization. It ensures reproducibility of the results and scalability to new preprocessing strategies and data.

Key Findings

- The sequence representations obtained from the deep learning model outperformed sequence alignment by generating moderately more distant embeddings in functionally unrelated promoters than in functionally related ones.
- Although it has limited power, the model represents a promising starting point for future research, which could lead to a powerful genome annotation and phylogenetics analysis tool.
- The data-centric approach (optimization of the data preprocessing) was critical to ensure model performance.
- The Nextflow pipeline developed functions as a reproducible workflow that automates the optimization of the data preprocessing and the deep learning model.

Acknowledgements

After all the effort this work required, the first person I must thank is Cristina. Firstly, for setting up this great project, then for her support during the entire process and finally, for doing such a great job in her first supervision experience.

I also want to thank Cedric, for his invaluable scientific guidance and feedback and for creating a lab with such amazing people. To Ionas, Jose & Wangshen, thank you for the laughs, lunches together, football talks and great scientific discussions.

To the whole Dias & Frazer lab for their amazing suggestions on deep learning and, just as importantly, for an incredible beach volleyball tournament.

And last but not least, thanks to Cinta for her advice in our meetings during these months and for her interest in my well-being and project advancement.

Annex

All the code used in this work (deep learning model, Nextflow pipeline & more) is available in the GitHub repository: https://github.com/victor-fdz/DeepBenchBioData.

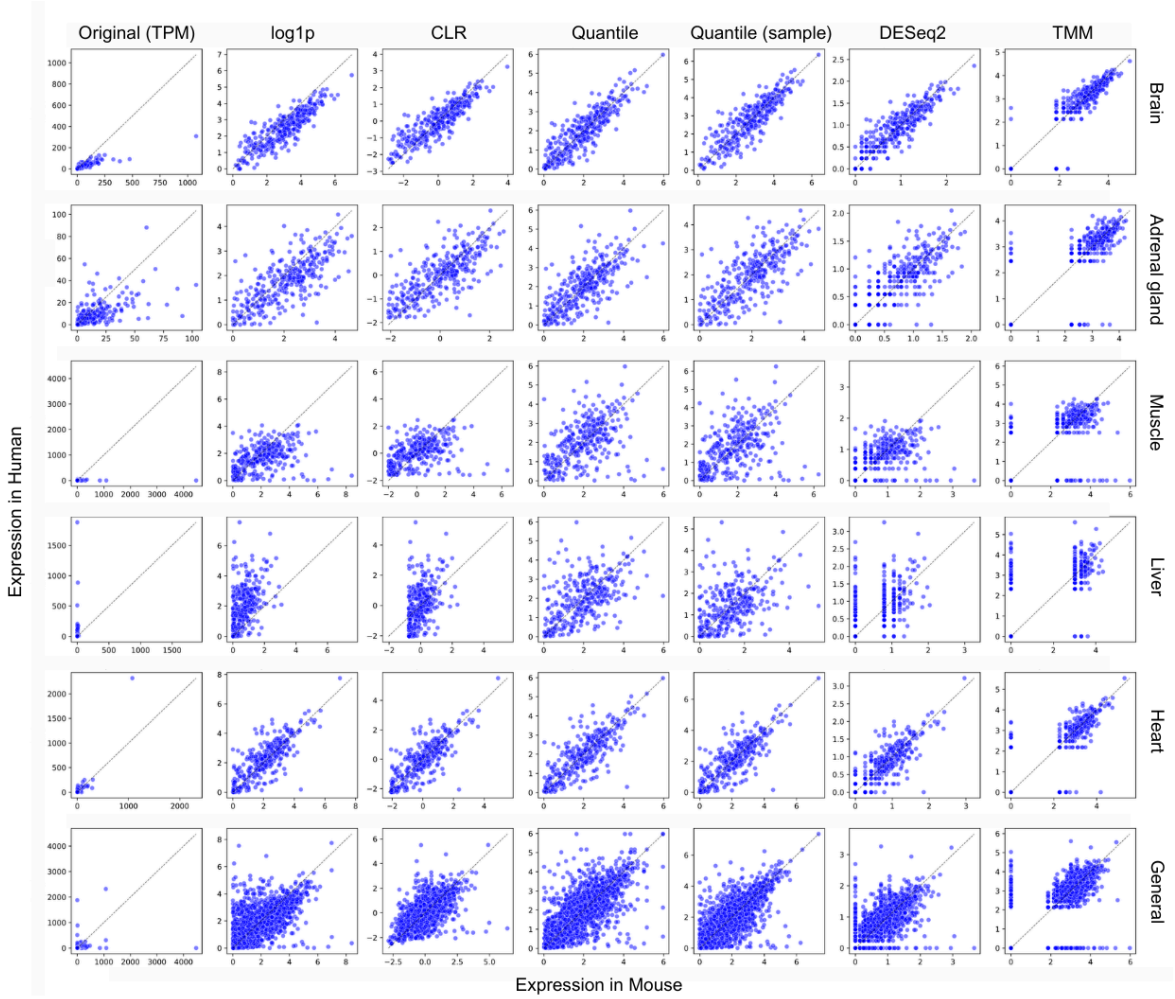

Figure A1: Scatter plots for every normalization in every specific tissue and general (all tissues combined). Mouse gene expression (X-axis) vs human gene expression (Y-axis) in the one-to-one orthologs dataset.

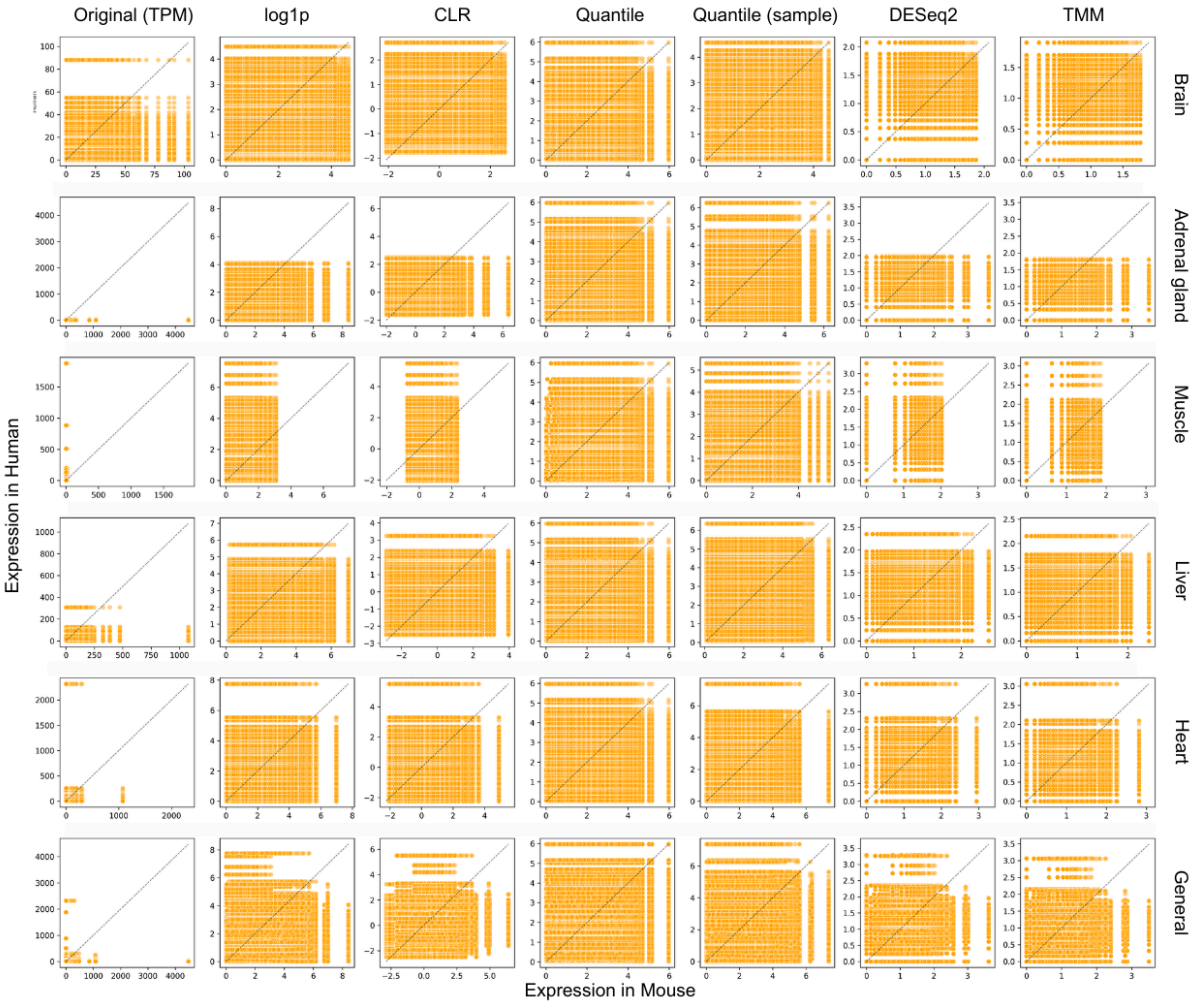

Figure A2: Scatter plots for every normalization in every specific tissue and general (all tissues combined). Mouse gene expression (X-axis) vs human gene expression (Y-axis) in the non-orthologous dataset.

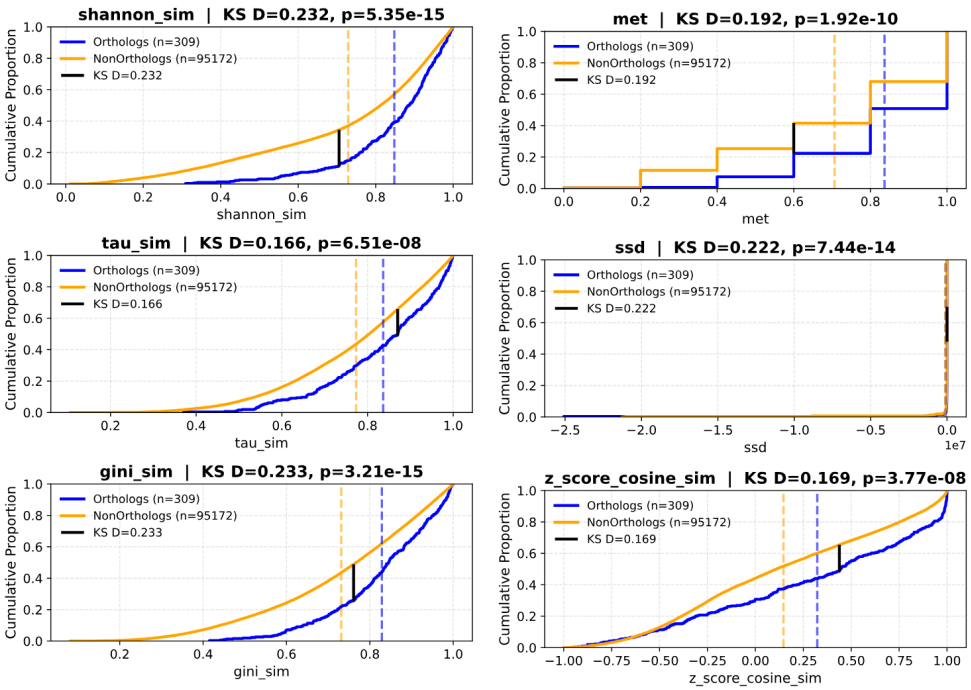

Figure A3: Cumulative distribution of orthologous and non-orthologous pairs for each expression similarity measure. The graphs show the Kolmogorov-Smirnov statistic as a vertical black line. Dotted vertical lines show the mean value of each distribution.

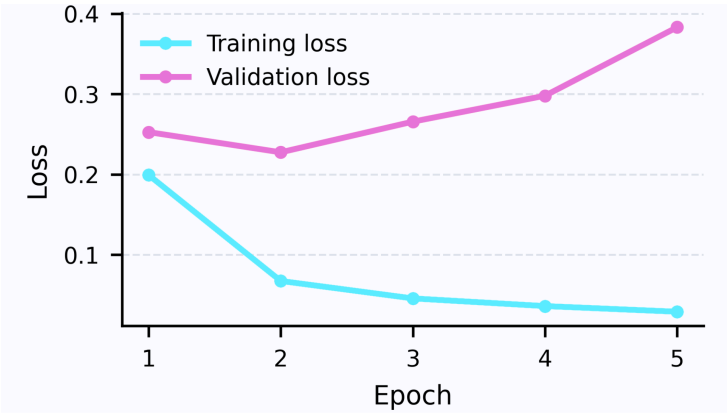

Figure A4: Loss dynamics during training. The reduction only occurs during the first two epochs. From that moment on, validation loss increases. Early-stopping restored model parameters from epoch 2.

Bibliography

1. Peña-Martínez E G, Rodríguez-Martínez J A. Decoding Non-coding Variants: Recent Approaches to Studying Their Role in Gene Regulation and Human Diseases. Vol. 16. IMR Press Limited; 2024.
2. Jaganathan K, Ersaro N, Novakovsky G, Wang Y, James T, Schwartzentruber J, et al. Predicting expression-altering promoter mutations with deep learning. Science. 2025;389(6760).
3. Avsec Ž, Latysheva N, Cheng J, Novati G, Taylor K R, Ward T, et al. Advancing regulatory variant effect prediction with AlphaGenome. Nature. 2026;649(8099):1206–18.
4. Aguet F, Ardlie K G. Tissue Specificity of Gene Expression. Current Genetic Medicine Reports. 2016;4(4):163–9.
5. Karollus A, Hingerl J, Gankin D, Grosshauser M, Klemon K, Gagneur J. Species-aware DNA language models capture regulatory elements and their evolution. Genome Biology. 2024;25(1).
6. Notredame C, Higgins D G, Heringa J. T-coffee: A novel method for fast and accurate multiple sequence alignment. Journal of Molecular Biology. 2000;302(1):205–17.
7. Kempen M van, Kim S S, Tumescheit C, Mirdita M, Lee J, Gilchrist C L, et al. Fast and accurate protein structure search with Foldseek. Nature Biotechnology. 2024;42(2):243–6.
8. Ponting C P. Biological function in the twilight zone of sequence conservation. BMC Biology. 2017;15(1):71.
9. Chen K, Pattar V, Shao M. Sequence similarity estimation by random subsequence sketching [Internet]. 2025. Available from: http://biorxiv.org/lookup/doi/10.1101/2025.02.05.636706
10. Holur P, Enevoldsen K C, Rajesh S, Mboning L, Georgiou T, Bouchard L S, et al. Embed-Search-Align: DNA sequence alignment using Transformer models. Bioinformatics. 2025;41(3).
11. Rost B. Twilight zone of protein sequence alignments. Vol. 12. 1999 pp. 85–94.
12. Dalla-Torre H, Gonzalez L, Mendoza-Revilla J, Carranza N L, Grzywaczewski A H, Oteri F, et al. Nucleotide Transformer: building and evaluating robust foundation models for human genomics. Nature Methods. 2025;22(2):287–97.
13. Nguyen E, Poli M, Faizi M, Thomas A, Birch-Sykes C, Wornow M, et al. HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution. 2023;. Available from: http://arxiv.org/abs/2306.15794
14. Haberle V, Stark A. Eukaryotic core promoters and the functional basis of transcription initiation. Nature reviews Molecular cell biology. 2018;19:621–37.
15. Kryuchkova-Mostacci N, Robinson-Rechavi M. A benchmark of gene expression tissue-specificity metrics. Briefings in Bioinformatics. 2017;18(2):205–14.
16. Langschied F, Iruegas R, Sikora M, Covino R, Ebersberger I. A Multi-level Perspective on the Evolution of Orthologs and Their Functions. Journal of Molecular Evolution. 2025;93(6):720–9.

17. Zhao Y, Wong L, Goh W W B. How to do quantile normalization correctly for gene expression data analyses. Scientific Reports. 2020;10(1).

18. Booeshaghi A S, Pachter L. Normalization of single-cell RNA-seq counts by log(x + 1)† or log(1 + x)†. Bioinformatics. 2021;37(15):2223–4.

19. Xu L, Xu M, Ke Y, An X, Liu S, Ming D. Cross-Dataset Variability Problem in EEG Decoding With Deep Learning. Frontiers in Human Neuroscience. 2020;.

20. Zhou Y, Zhu J, Tong T, Wang J, Lin B, Zhang J. A statistical normalization method and differential expression analysis for RNA-seq data between different species. BMC Bioinformatics. 2019;20(1).

21. Quinn T P, Erb I, Richardson M F, Crowley T M. Understanding sequencing data as compositions: An outlook and review. Bioinformatics. 2018;34(16):2870–8.

22. Lin Y, Golovnina K, Chen Z X, Lee H N, Negron Y L, Sultana H, et al. Comparison of normalization and differential expression analyses using RNA-Seq data from 726 individual Drosophila melanogaster. BMC Genomics. 2016;17(1).

23. Sancho C A. Exploring RNA-seq through ratios: An nf-core workflow for compositional data analysis. 2024.

24. Zou J, Huss M, Abid A, Mohammadi P, Torkamani A, Telenti A. A primer on deep learning in genomics. Nature Genetics. 2019;51(1):12–8.

25. Bromley J, Guyon I, LeCun Y, Sackinger E, Shah R. Signature Verification using a "Siamese" Time Delay Neural Network. Advances in Neural Information Processing Systems. 1993;6.

26. Hadsell R, Chopra S, Lecun Y. Dimensionality Reduction by Learning an Invariant Mapping. 2006.

27. Whang S E, Roh Y, Song H, Lee J G. Data collection and quality challenges in deep learning: a data-centric AI perspective. VLDB Journal. 2023;32(4):791–813.

28. Jin S. Dealing with noise and biases in omics: statistical methods, deep learning and workflows. 2026.

29. Tommaso P D, Chatzou M, Floden E W, Barj P P, Palumbo E, Notredame C. Nextflow enables reproducible computational workflows. Nature. 2017;35:316–9.

30. Langer B E, Amaral A, Baudement M O, Bonath F, Charles M, Chitneedi P K, et al. Empowering bioinformatics communities with Nextflow and nf-core. Vol. 26. BioMed Central Ltd; 2025.

31. Kinsella R J, Kähäri A, Haider S, Zamora J, Proctor G, Spudich G, et al. Ensembl BioMarts: A hub for data retrieval across taxonomic space. Database. 2011;2011.

32. Consortium T E P. The ENCODE (ENCyclopedia Of DNA Elements) Project. 2004.

33. Dyer S C, Austine-Orimoloye O, Azov A G, Barba M, Barnes I, Barrera-Enriquez V P, et al. Ensembl 2025. Nucleic Acids Research. 2025;53(D1):D948–57.

34. Cock P J A, Antao T, Chang J T, Chapman B A, Cox C J, Dalke A, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics. 2009;25(11):1422–3.

35. Siqueira Santos S de, Takahashi D Y, Nakata A, Fujita A. A comparative study of statistical methods used to identify dependencies between gene expression signals. Briefings in Bioinformatics. 2013;15(6):906–18.

36. Karaca Y, Moonis M. Chapter 14 - Shannon entropy-based complexity quantification of nonlinear stochastic process: diagnostic and predictive spatiotemporal uncertainty of multiple sclerosis subgroups [Internet]. Karaca Y, Baleanu D, Zhang Y D, Gervasi O, Moonis M, editors. Multi-Chaos, Fractal and Multi-Fractional Artificial Intelligence of Different Complex Systems. Academic Press; 2022. pp. 231–45. Available from: https://www.sciencedirect.com/science/article/pii/B9780323900324000183

37. Paszke A, Gross S, Massa F, Lerer A, Google J B, Chanan G, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. 2019.

38. Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: A Next-generation Hyperparameter Optimization Framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining [Internet]. Anchorage, AK, USA: Association for Computing Machinery; 2019. pp. 2623–31. (KDD '19). Available from: https://doi.org/10.1145/3292500.3330701

39. Kaplan J, McCandlish S, Henighan T, Brown T B, Chess B, Child R, et al. Scaling Laws for Neural Language Models. 2020;. Available from: http://arxiv.org/abs/2001.08361

40. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021;596(7873):583–9.

41. Berman H M, Westbrook J, Feng Z, Gilliland G, Bhat T N, Weissig H, et al. The Protein Data Bank [Internet]. Vol. 28. 2000. Available from: http://www.rcsb.org/pdb/status.html

42. Rahaman N, Baratin A, Arpit D, Draxler F, Lin M, Hamprecht F, et al. On the Spectral Bias of Neural Networks. In: Chaudhuri K, Salakhutdinov R, editors. Proceedings of the 36th International Conference on Machine Learning [Internet]. PMLR; 2019. pp. 5301–10. (Proceedings of Machine Learning Research; vol. 97). Available from: https://proceedings.mlr.press/v97/rahaman19a.html

43. Ewels P A, Peltzer A, Fillinger S, Patel H, Alneberg J, Wilm A, et al. The nf-core framework for community-curated bioinformatics pipelines. Nature Biotechnology [Internet]. 2020;38(3):276–8. Available from: https://doi.org/10.1038/s41587-020-0439-x