Gene representations for gene regulatory network inference

Llan Almendariz

[1]Bioinformatics Group, Wageningen University Research, Droevendaalsesteeg 1, Wageningen 6708 PB, the Netherlands

Abstract

Gene Regulatory Networks (GRN) describe the relationships between Transcription Factors (TFs) and target genes. Complete GRNs are important for understanding the molecular and functional processes. As experimental GRN inference remains laborious, various computational methods have been developed for GRN inference. Supervised link-prediction models learned from a prior GRN to predict missing links, the prior GRN is often attributed with a secondary data sources that describe the genes in the network. Link-prediction models learn node representations based on their *topology* in the network and the secondary data source. As *topology* constitutes structural features such as node connectivity and centrality, it is not clear from what component the performance is derived, *topology* or the secondary data source. This project investigates expression, coexpression and functional gene representations for GRN inference demonstrated in *Arabidopsis thaliana*. It was found that the gene representations improved GRN inference, however, a great reliance on *topological* information was observed. By testing model performance on unseen genes using various datasplits that constituted transductive, semi-transductive and inductive test settings, it was found that the gene representations varied in their utility for GRN inference. Furthermore, results show that testing on genes seen during training (transductive) positively impacts performance compared to unseen genes (inductive), due to exploitation of the *topological* information. Thus the inductive test setting provides a more difficult prediction problem, and a better evaluation method for gene representations. Furthermore, key areas of improvement are proposed to enhance the learning of robust gene representations for inductive prediction.

Key words: Gene Regulatory Network, *Arabidopsis thaliana*, deep learning, inductive, transductive, link-prediction

Introduction

Gene regulatory networks (GRNs) describe the relationships between genes in a biological system, generally conceptualizing genes as nodes and regulatory relationships as edges(Mercatelli et al., 2020). Regulation of gene expression is primarily done by Transcription Factors (TFs), proteins that repress or activate transcription of their target gene by binding to regulatory regions. Whether a gene can be regulated by a TF depends on various factors e.g. chromatin structure (Li and Reinberg, 2011), and TF binding sites (Georgakopoulos-Soares et al., 2023).

Genome wide screening for TF binding sites is primarily done using ChIP-seq that isolates the chromatin bound to a TF using an antibody capture system, however, this method is laborious and requires expensive antibody engineering (Nakato and Sakata, 2021). Another common method is the yeast one-hybrid systems that identifies DNA-protein interactions through a bait-prey system located in yeast cells (Hollingsworth and White, 2004). While this method is able to screen many target sequences against interaction with a TF, it is hindered by high rate of false positives (Fuxman Bass et al., 2016). Furthermore, both methods rely on a priori knowledge about the TFs and targets when preparing the experiment that is often unavailable in lesser studied organisms. This causes the experimental construction of a GRN to move at a slow pace, and therefore to date, *Arabidopsis thaliana* has 1,717 identified TF loci, but only 324 have at least one target gene assigned (Jin et al., 2017).

Various computational methods have been proposed to predict regulatory relationships using machine learning (ML) (Badia-i Mompel et al., 2023; Mercatelli et al., 2020). Traditional methods such as Aracne (Margolin et al., 2006) and GENIE3 (Huynh-Thu et al., 2010) aim to reverse engineer the GRN from the data by selecting the most probable combination of TFs for each gene (Hecker et al., 2009; Huang et al., 2009). These methods can be characterized as unsupervised feature selection algorithms, as they reconstruct the network given the expression data, and do not require a prior GRN for learning or extracting features. On the one hand this can be considered an asset, as this allows the algorithm to be used for GRN inference of lesser studied organisms, where a GRN is often unavailable. On the other hand a drawback, as they do not take advantage of existing knowledge of regulatory interactions.

More novel methods tend to use a supervised link-prediction framework to exploit the existing GRN, and better discern the positive and negative TF-target pairs (Chen and Liu, 2022). This framework relies on learning from known interactions (i.e. links) for the prediction of new interactions. Since TFs and targets are conceptualized as nodes in the network, these models learn a node representation that consists of two components: (**i**) *topology*, i.e. node connectivity and centrality (Zhang and Luo, 2017), and (**ii**) a secondary data source e.g. expression data. *Topological* node information has been shown to be informative for GRN inference, and link-prediction in general (Hao et al., 2020; Ghasemian et al., 2020).

In GRN inference using link-prediction, *topological* information is usually complemented by secondary data source(s) to improve performance. For instance, expression data (Patel and Wang, 2015), coexpression data (Du et al., 2019), and Gene Ontology (GO) data (Ieremie et al., 2022).

Among these data sources, expression and coexpression data are used to represent a gene by its expression levels among samples, and coexpression neighbors, respectively (Zhang et al., 2020b; Du et al., 2019). Both gene representations are similar, but the coexpression representation trades expression values for a selection of correlated neighbors. These representation enable learning expression patterns specific to regulation, however, do not incorporate functional gene information.

The GO data database contains functional gene information structured into a knowledge graph, where terms describing biological processes (BP), molecular functions (MF) and cellular component (CC) of genes are represented as nodes and hierarchical relationships between terms as edges. Currently, the database contains a total of 7,247 ontology terms. This information can be exploited to learn functional similarities between genes (Berardini et al., 2004).

Despite the ability of traditional ML methods to combine these secondary data sources, deep learning techniques are more efficient in this aspect. Key advantages over ML techniques are automatic feature extraction, and transformation of the data into a lower-dimensional space while capturing non-linear patterns (Xia et al., 2021). This makes deep learning more practical for combining data sources than ML.

Various deep learning models such as scGREAT Wang et al. (2024) and GENELink (Chen and Liu, 2022) have been proposed to combine and exploit node features. These models, use a supervised link-prediction framework to learn gene representations for GRN inference, demonstrating significant improvements over traditional methods (Wang et al., 2024). The scGREAT model, developed for human and mouse data, combines a gene symbol embedding generated using the Large Language Model (LLM) BioBERT Lee et al. (2020) with an expression representation. However, as these models are able to exploit the network structure, the performance derived from the gene representations cannot be discerned from the performance derived from *topological* node information.

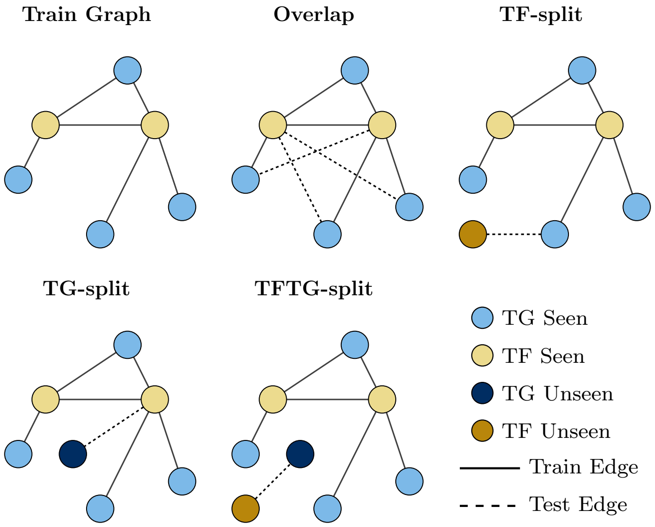

Figure 1: Link-prediction models are trained on a prior Gene Regulatory Network (GRN) (**Train Graph**) and evaluated in three test settings (i.e. datasplits). **Overlap**: a transductive test, where test nodes were seen during training, **TF-split** and **TG-split**: semi-transductive tests, where the test network contained unseen TFs or targets, respectively, and **TFTG-split**: an inductive test, where the test network contained only unseen nodes.

The reliance of *topological* node information can be avoided by creating specific test settings, namely, transductive, semi-transductive, and inductive. GRN inference models are typically evaluated in a transductive test setting, where nodes (TFs and targets) were already seen during training (Figure 1-**Overlap**). The model can thus exploit the *topological* node information. Conversely, in a inductive test setting, nodes seen during training are not in the test set (Figure 1-**TFTG-split**), removing the ability to exploit *topological* node information. The semi-transductive test setting contains both unseen and seen nodes in the test set. This setting translates to an instance where only TFs nodes were unseen during training (Figure 1-**TF-split**), and an instance where the target gene nodes were unseen (Figure 1-**TG-split**). These test settings should be taken into account in the evaluation of gene representations and GRN inference.

This research investigates three gene representations, based on (**i**) expression data, (**ii**) coexpression neighbor data, and (**iii**) GO data for *A. thaliana* GRN inference, a well studied plant model organism. The GRN inference model follows a supervised link-prediction framework for learning gene representations using a deep learning model. The utility for GRN inference of the gene representations were thoroughly evaluated in test settings displayed in Figure 1. I report a major reliance on *topological* information that is inherent to link-prediction models, contextualizing the performance gain derived from gene representation. Furthermore I investigate whether combining gene representations is beneficial for GRN inference and compare model performance with that of scGREAT across the test settings.

Methods

Computational gene regulatory network (GRN) inference is a well studied problem in bioinformatics that aims to predict how some genes regulate other genes. The deep learning model in this study follows a supervised link-prediction framework, thus predicting missing links from existing links (Getoor and Diehl, 2005). Gene representations constructed from (co)expression and GO data are used to construct numerical vectors that serve as input to the model. Optimization occurs such that the vectors encode important features for GRN inference.

Gene representation design principles

To infer a regulatory network using deep learning, the construction of a robust gene embedding (i.e. a representation) is essential, as this serves as the basis for predicting interactions. This study evaluates three gene representations, namely: **GO** embedding, **expression** embedding, and **coexpression** embedding. The set of genes is denoted by $\mathcal{G} = \{g_1, g_2, \ldots, g_m\}$, where the i^{th} gene $g_i \in \mathcal{G}$ corresponds to a gene annotated by Araport11(Lamesch et al., 2012). The total number of genes m is 37,336.

The information from the GO database (Berardini et al., 2004) was encoded in a binary vector $\mathbf{b}_i \in \{0,1\}^d$, where b_i denotes the binary embedding vector of length d for gene g_i. This binary embedding vector is a multi-hot encoded vector where each position relates to a term in the set of GO terms, that is denoted as $\mathcal{T} = \{t_1, t_2, \ldots, t_{N_{\text{terms}}}\}$. If GO-term t_j is assigned to the gene, the j^{th} element of $\mathbf{b}_i$ will be 1, otherwise 0. The number of GO-terms (N_{terms}) can vary based on evidence level constraints (see below); however, unless otherwise stated, all GO-terms were used.

The expression matrix was collected from plantrnadb.com (Yu et al., 2022; Zhang et al., 2020a), which contains over 28,000 gene expression samples and includes 37,336 genes. The data was collected from various experiments and is given as Fragments Per Kilobase per Million (FPKM) normalized values. Sample quality was determined by the proportion of uniquely mapped reads to the reference genome, and samples with a percentage below 90 were excluded. As the computational load depends on the number of samples, an inclusion criteria was set based on the ecotype and genotype. These terms describe a distinct population of *A. thaliana* adapted to a specific environment, and the genetic composition. Since Col-0 has been extensively sequenced, and used as a reference strain, only these samples were included. Post filtering, 1,343 samples remained. The $m \times n$ expression matrix is denoted as $\mathbf{E}$, where m corresponds to the number of genes and n to the number of samples; the values in $\mathbf{E}$ are z-score transformed counts. The expression representation was constructed by encoding the expression values over the n samples into dense vector $\mathbf{x}_i$, that corresponds to the i^{th} row in $\mathbf{E}$, and has length n.

The expression matrix $\mathbf{E}$ was used to construct a correlation matrix $\mathbf{C}$ of size $m \times m$. The correlation threshold was set at 0.6 to ensure that a sufficient number of genes had coexpression neighbors, and thus could be represented. The neighbors for a gene g_i are defined as those g_j for which $C_{ij} > 0.6$. Thus, for each gene g_i a set of neighbors was harvested from the correlation matrix. The Gene2vec (Du et al., 2019) model was used to embed each g_i for which a set of neighbors is available. The Gene2Vec model is based on the Word2Vec algorithm (Mikolov et al., 2013) and embeds a gene into a lower-dimensional space by optimizing the likelihood of observing its neighboring genes. The idea is that similar genes are mapped together in embedding space, thereby capturing biologically meaningful information. The embedding vector is denoted by $\mathbf{c}_i$ and has length d set at 1,000. The length of the coexpression representation was chosen to have the same order of magnitude as the expression and GO representations for better comparability.

Random representations & vector length

Link-prediction models can exploit *topological* node information and gene representations for GRN inference, making it difficult to isolate the contribution of each to the performance. To disentangle their effects, a base-line was calculated using randomly generated representations, that isolate the performance of *topological* node information. For each gene representation described in Section 2.1 a dummy **random** representation was created, specifically:

- random GO representation contains integers uniformly drawn from the set $\{0,1\}$ (binary), with the same length of the original GO representation.
- random (co)expression representation contains values draw uniformly from the interval $[0,1]$ (dense), and has the same length as the original (co)expression representation.

Vector length of the random representation can affect the performance, since a longer vector can take on more unique representations. To study this effect, the model were trained and tested with random representations of various lengths. Binary and dense random representations were tested, with the length ranging between 1 and 2000 positions.

Assessment of the GO evidence codes

Since the GO database contains gene annotations that are based on diverse evidence types (see Table 6), the quality of the supporting evidence may influence the utility of the GO representation for GRN inference. Specifically, uncurated annotations (IEA), and computationally derived annotations (ISS, ISM, IMP, IBA, RCA) are more likely to contain erroneous annotations, since they are not experimentally verified (Berardini et al., 2004). To examine this effect, the two extra GO representations were constructed beside the previously described GO representation. These, were constructed as mentioned previously, but, excluded from uncurated annotations, or computationally derived annotations.

For each the representation the coexpression (G2V) model as described in Section 2.4, with a modified input size, to be compatible with the length of the two extra GO representations. This was necessary as the exclusion of evidence types led to a smaller set of GO terms $\mathcal{T}$, and thus a shorter vector length. It is important to note that the G2V model is not to be confused with the Gene2vec model described in Du et al. (2019).

Network architecture

To leverage the gene representations described above (Section 2.1) for GRN inference, multiple deep learning models were designed. To evaluate the individual performance of the gene representations the architectures in Figure 2 were used, the combined performance was evaluated using the architecture in Figure 3. This was done to examine the usefulness and robustness of the representations for inferring interactions both in isolation and in concert. Additionally, a transformer module was tested in

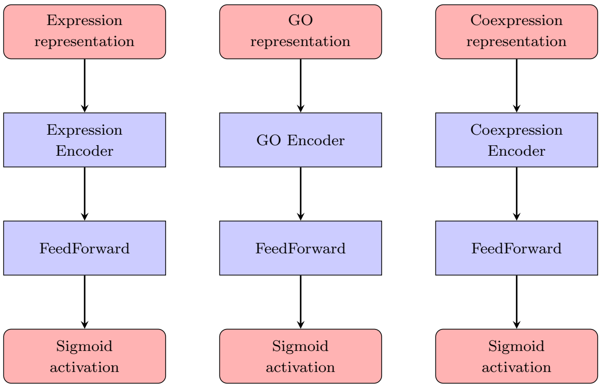

Figure 2: Architecture used for predicting interactions based on individual gene representations.

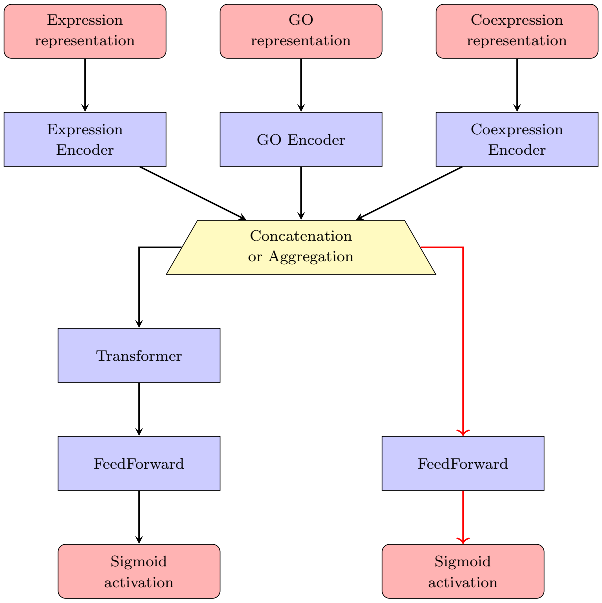

Figure 3: Model architecture used to combine gene representations, vector concatenation or addition was used to aggregate (trapezium) representations after encoding. In the combined setting a (i) linear model (red arrow) and (ii) transformer model were used (black arrow).

the fully combined setting to examine whether the use of attention mechanism would benefit prediction performance.

The input to the model is a multidimensional array (i.e. tensor) of size $b \times c \times e \times s$, where b is the batch size and c is 2 corresponding to 2 genes being processed in parallel. The number of representations is denoted by e, their length by s.

Each representation has its own encoder module to learn representation specific feature information and transform the representation to a lower dimension of size s'. The number of layers, parameters, and activation functions can be viewed in Table 1. The Rectified Linear Unit (ReLU) was used to capture non-linear patterns. The encoder layers were designed to be similar among the representations for better comparability; only the coexpression (G2V) encoder differs from the and expression (EXP) and GO (GO) encoders with respect to the number of parameters, which might reduce intercomparability.

The feedforward architecture can be seen in Table 1 and predicts based on the embeddings of two genes. To this end, the two embeddings are both flattened before entering the feedforward module, to match the dimensions $b \times s'$. The output is a scalar logit value, representing the raw likelihood of interaction. The sigmoid activation function is used to normalize the logit to a probability value.

In the combined setting $e > 1$ and embeddings need to be aggregated into a single vector to be compatible with the feedforward module. Two means of aggregation were tested: concatenation ($e \times s'$), and addition (s'), the values in brackets are the resulting lengths.

Table 1. Encoder and feedforward module configurations: the numbers of linear layers, parameters, and activation functions for the Linear Model.

Layer	Expression Encoder	GO Encoder	G2V Encoder	Feed-forward
Input Size	1343	7247	1000	*combined size*
Linear 1 in	1343	7247	1000	*combined size*
Linear 1 out	1280	1280	768	128
Activation 1	ReLU	ReLU	ReLU	ReLU
Linear 2 in	1280	1280	768	128
Linear 2 out	1024	1024	512	64
Activation 2	ReLU	ReLU	ReLU	ReLU
Linear 3 in	1024	1024	512	64
Linear 3 out	100	100	100	1
Activation 3	–	–	–	Sigmoid

Ground truth GRN pre-processing

A ground truth GRN was retrieved from the Plantregmap (Jin et al., 2015) and pre-processed to an edge table containing three columns: TF, target, a one indicating interaction. The TFs that were self-regulating were removed to reduce the complexity as a consequence of self-loops. Furthermore, the table was filtered to include only genes that could be represented using all three gene representations. This was done as the GO, expression and coexpression representations varied in the number of genes they could represent, namely: 25,385, 37,336 (i.e. all genes), and

32,181 respectively. After filtering, the ground truth network contained 1,213 positive edges that constituted the final ground truth network.

Data partitioning strategy and test settings

To train and evaluate the models, the ground truth GRN was partitioned into three datasets, namely:

- train set for training the model.
- validation set for tracking model performance during training.
- test set for evaluating the model post training.

Different partitioning strategies were used to create the four train/test datasplits that represent the test settings displayed in Figure 1, namely:

- **Overlap**: TF and target nodes can occur in the train and test sets.
- **TF-split**: TF test nodes cannot occur in the train set.
- **TG-split**: target gene test nodes cannot occur in the train set.
- **TFTG-split** no test nodes can occur in the train set.

The train/test ratio was 3:1, and the validation set was subsequently acquired from the train set with a train/validation ratio of 9:1. The train set of each datasplit was downsampled such that they were the same size to ensure a fair comparison. Variation in size was due to the different constraints by the partitioning strategies. Furthermore, the *TFTG-split* validation set was so small (< 50 examples) that it was combined with the test set.

Negative edges were selected according to the principle of Hard Negative Sampling (HNS), that couples as many false targets to every TF as it has positive targets (Robinson et al., 2021). This negative sampling technique aims to select edges that are difficult to be distinguished from positive edges, providing more informative training edges (Lai et al., 2024). Furthermore, this method ensures that the data is class-balanced. The four datasplits were illustrated by means of a venn diagram in Figure 7, and the resulting distribution of the train/validation/test sets can be viewed in Table 4.

Training setting

The network is optimized end-to-end by minimizing the BinaryCrossEntropy Loss (BCELoss) using the Adam optimizer. Training was done for a maximum of 50 epochs, but was terminated early if the loss did not decrease by more than 0.01 over the period of 2 epochs. The parameters for the early stopping module were constant in all experiments. The model was trained with a learning rate of 10^{-4}, a batch size of 32, and shuffling of the training data, unless otherwise stated.

Evaluation metrics

Performance was measured using the Area Under the Receiver Operating Characteristic (AUROC) score as the primary metric. The AUROC represents the discriminatory capacity between true and false positives. This metric was used as the data was class balanced, limiting any bias towards the majority class. The Area Under the Precision-Recall Curve (AUPRC) was calculated as well. The AUROC and AUPRC were calculated over the course of ten training runs and the mean scores were reported together with their standard errors (SEMs) calculated using Bessel's correction ($n - 1$). Additionally, the mean precision, recall, F1-score and accuracy were reported. The train and validation loss were recorded over the epochs to track the convergence. Furthermore, the accuracy was recorded over the epochs and plotted alongside the train and validation loss, to contextualize the practical impact of convergence on the number of correct predictions.

Table 2. Random parameter search was conducted for the following parameter combinations. The configuration space had size 1,701 of which 1,344 were explored.

Parameter	Tested Values
Embed Size	25, 50, 100, 200, 400, 600, 800
Encoder Layers	1, 2, 3, 4, 5, 6, 7, 8, 9
FFOut Layers	1, 2, 3, 4, 5, 6, 7, 8, 9
Activation	ReLU
Aggregation	concatenation
Learning Rate	$10^{-3}, 10^{-4}, 10^{-5}$

Parameter optimization

Hyperparameter configurations were optimized by sampling the configuration space listed in Table 2. Of the 1,701 possible configurations, 1,344 were evaluated in the inductive test setting represented by the *TFTG-split* dataset over the course of five train and test runs. The hyperparameters were chosen to investigate whether model complexity, as a function of the number of layers and trainable parameters, would improve GRN inference. The hyperparameter configuration space was explored by employing a random search strategy, to have a higher probability of finding the global optimum considering the time, in comparison to a grid search strategy (). The random search took approximately 8 hours on an NVIDIA GeForce RTX 3060 12 GB using NVIDIA-SMI 570.86.16 Driver Version 570.86.16 and CUDA Version 12.8.

scGREAT: data and training

The scGREAT (Wang et al., 2024) model served as an additional baseline for the architecture and GRN data from this research, based on human data. Evaluation of scGREAT deviated from the paper (Wang et al., 2024), this study used the four test settings settings outlined in Figure 1, and the random representations for the expression and Biovector gene representations. The random representations were generated using the same method as the (co)expression random representation described in Section 2.2. The human embryonic stem cells (hESC) dataset from BEELINE (Pratapa et al., 2020), which was included with scGREAT as a demo file, was partitioned according to the strategies described in Section 2.6. To ensure reproducibility, we adopted the parameters from scGREAT's demo, which accurately reproduced the AUROC scores reported in their publication (Wang et al., 2024): a batch size of 32, hidden embedding size of 768, 2 layers, 4 attention heads, and a learning rate of 10^{-5}. Training was conducted for a maximum of 50 epochs across 10 independent runs, with early stopping (delta=0.01, patience=2). This allowed us to directly compare scGREAT's performance in the three test settings with our own model and data partitioning approach.

Dimension reduction for exploring the embedding space

To investigate whether the (co)expression gene representations capture biologically relevant signals, the embedding space was explored. The high-dimensional gene representation vectors were projected into two dimensions, using t-distributed Stochastic

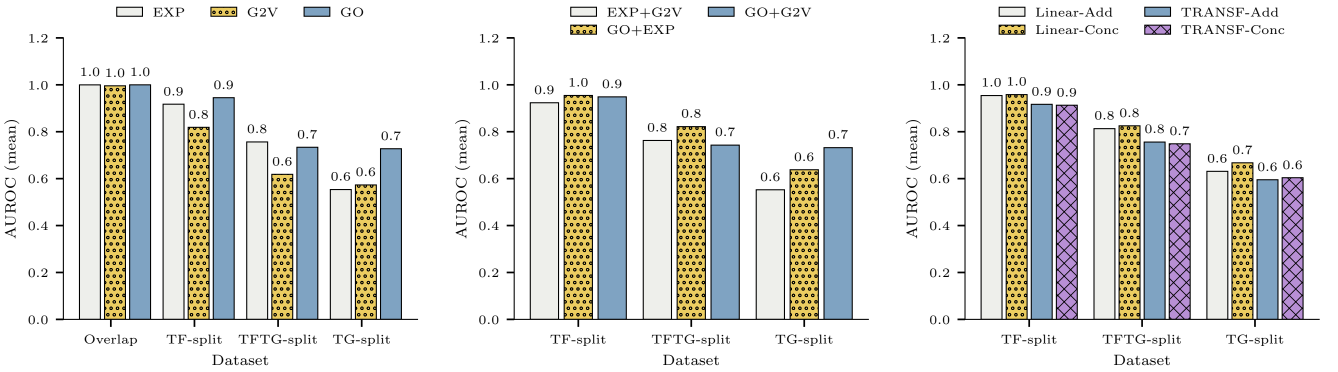

Figure 4: Model performance as a function of the AUROC score for **left**: prediction based on individual gene representations, **middle**: paired gene representation, and **right**: combined gene representation through vector concatenation (-Conc) and addition (-Add) in linear model (Linear) and transformer model (TRANSF). The representations are denoted by **EXP**: expression; **G2V**: coexpression; and **GO**: GO. It was found that GO generally scored high over the datasets while the EXP representation consistently performed better in the *TFTG-split* setting than all other representations.

Neighbor Embedding (t-SNE). This is done by preserving local relationships between datapoints by representing their similarities as probabilities, and creating similar probability distributions in a lower dimensional space (Maaten and Hinton, 2008). The t-SNE dimension reduction was conducted using the scikit-learn python package, with default parameters, except perplexity that was set at 300. As the data contained a large number of genes, a larger perplexity was used. this increases the number nearest neighbors used, and better preserves the global structure.

Results

GRN inference using individual gene representations

To examine the individual performance of each gene representation for GRN inference, the models in Figure 2 were tested across the four datasplits. Figure 4-left panel shows the performance of the expression (EXP), coexpression (G2V), and GO representation (GO). The AUROC score on the *overlap* datasplit was completely saturated (AUROC=1) for all gene representations, indicating that the model is perfectly able to predict all missing links. However, comparing the AUROC score to base-line, reveals that the relative gain is practically zero (Table 3). The *overlap* dataset, thus, does not provide any information on the usefulness of the gene representations as the AUROC score was already very high when using random representations (Figure 5). The general AUROC score and other metrics can be viewed in Table 10 and 11.

The performance in the other datasplits does improve over base-line, with improvements ranging from 0.05 to 0.25. The largest improvements were achieved in the *TFTG-split* dataset, where the EXP representation has the highest relative AUROC score. The *TF-split* dataset shows smaller improvements but a higher absolute AUROC score; here the GO representation gives the highest score. The relative AUROC score of the EXP representation is significantly lower in the *TG-split* compared to the GO representation. The G2V representation lags behind the GO and EXP representations for every dataset, except *overlap*.

The GO representation was further investigated, as the supporting evidence of annotations could influence the utility of the GO representation. The supporting evidence form the basis on which annotations are assigned to genes. Two extra GO representations were created by removing automatically inferred annotations and computationally inferred annotations (Table 6). The AUROC score was found to marginally improve as a consequence of removing automatically inferred annotations compared to using all annotations (Table 12 and 13). Removing computationally inferred annotations did not improve performance.

In summary, it was found that GO and EXP are the most informative gene representations, with GO more consistent across all datasets. The EXP representation achieves a marginally better AUROC score than the GO representation in the *TFTG-split*. Performance on the *overlap* dataset is almost completely saturated and is therefore not shown in Figure 2 middle and right panel. Furthermore, removing automatically inferred annotations only marginally improved the utility of the GO representations for GRN inference.

Table 3. Relative AUROC and AUPRC scores for **individual** gene representation. Scores were contrasted against scores achieved on random representations (actual - random).

		ROC		PRC	
		(mean)	(SEM)	(mean)	(SEM)
Dataset	Model				
Overlap	EXP	0.001	0.000	0.001	0.000
	G2V	-0.004	0.000	-0.007	0.000
	GO	0.000	0.000	0.000	0.000
TF-split	EXP	0.155	-0.003	0.149	-0.004
	G2V	0.077	-0.007	0.024	-0.006
	GO	0.125	-0.001	0.080	-0.002
TFTG-split	EXP	0.250	-0.005	0.256	-0.005
	G2V	0.114	-0.003	0.099	-0.004
	GO	0.221	-0.008	0.287	-0.009
TG-split	EXP	0.075	-0.001	0.045	0.001
	G2V	0.061	-0.004	0.074	-0.003
	GO	0.229	-0.004	0.254	-0.005

Combining representations improves GRN inference performance

As each gene representation encodes different information, the combination of representations could improve GRN inference. Figure 4-left shows that the GO representation gives more stable performance across the datasets, while the EXP representation achieves a higher score in the *TFTG-split*. To investigate whether combining gene representations improves GRN inference, the models outlined in Figure 3 were evaluated.

The gene representations were initially paired using addition, and were found to perform differently across the datasplits (Figure 4-middle). The GO+EXP pair was found to perform better in *TF-split* and *TFTG-split*, while the GO+G2V performed better in *TF-split*. However, compared to individual representations, the AUROC relative performance gain of GO+EXP in *TFTG-split* was only marginal, increasing from 0.25 (EXP) to 0.32 (GO+EXP) over base-line.

Subsequently, the representations were combined all together using addition or concatenation. This was done to further explore the performance of combined representations, and the impact of the aggregation method on the performance. Furthermore, a transformer-based model was evaluated to investigate whether the attention mechanism benefits GRN inference.

The linear model using vector concatenation was found to perform better in all datasplits compared to all other models (individual, paired and combined), achieving a relative AUROC of 0.325 (Table 16). Vector concatenation was found to be advantageous compared to vector addition for the linear model but not necessarily for the transformer model (Figure 4-right panel). The performance gain of concatenation over addition was most notable in *TG-split*, with a relative AUROC score of 0.1 and 0.06 respectively (Table 16). The transformer model did not benefit from the longer vector length as a consequence of concatenation, the effect of concatenation over addition was ambiguous in the four datasets.

In summary, it was found that combining gene representations boosts performance. Furthermore, vector concatenation resulted in better performance for the linear model, but not necessarily in the transformer model. Lastly, a linear model consistently outperformed a transformer model across the datasplits.

Data partition strategy influences base-line performance

Since the performance in the *overlap* datasplit was completely saturated (Figure 4-left panel), the data partitioning strategies were evaluated. This was done by substituting gene representations with randomly generated representations, separating the performance derived from *topological* node information and gene representations.

Figure 5 shows that all models achieve a perfect (AUROC=1) score in the *overlap* datasplit solely using random representations. Performance decreases in *TF-split* to approximately 0.75 AUROC score, and is random (AUROC=0.5) in *TFTG-split* and *TG-split*. The AUROC scores deviated among the gene representations in all datasplits except *overlap*. The performance achieved by random gene representations constitute base-line performance, all metrics can be viewed in detail in Table 8 and 9; these values were used to calculate the improvement over the base-line.

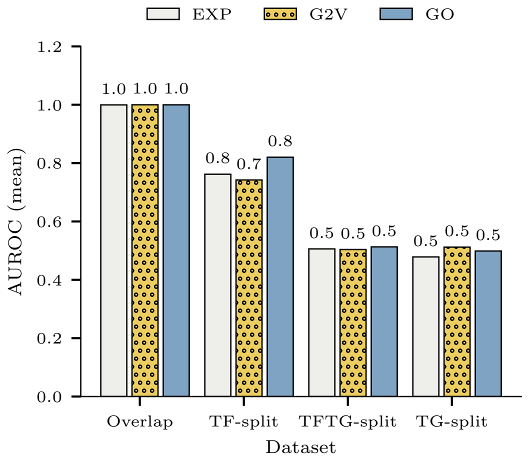

Figure 5: Model performance using random representations is not random (AUROC=0.5) in the *overlap* datasplit and *TF-split*.

Gene representation vector length affects base-line performance

As the baseline performance among the random gene representations deviated (Table 8), I reasoned that a longer random gene representation allows the model better differentiate between genes. Therefore, various lengths for random representations were evaluated across the datasplits.

It was found that base-line performance was influenced by vector length in the *overlap* datasplit and *TF-split*, but not in *TG-split* and *TFTG-split* (Figure 14 and 15). Furthermore, the AUROC grew at a higher rate with binary vectors (Figure 14), compared to the dense vectors (Figure 15). Suggesting that differences in base-line performance and performance in general gene between representation could be partially attributed to vector length and the type of vector.

Data partitioning strategy influences train and test networks

The GRN ground truth network is partitioned into a train and test networks, the structure of these networks are displayed in Figure 12. The structural difference between the train and test networks was most apparent in *TG-split* (Figure 12D). The train network was comprised of a relatively well-connected subnetwork surrounded by many smaller disconnected subnetworks, often comprised of a single TF. Conversely, the test network formed a dense structure similar to a “hairball”, with no disconnected subnetworks. *TF-split* (Figure 12B) yielded similar train and test networks, both comprised of a dense network with disconnected subnetworks. The *TFTG-split* (Figure 12C) training and test networks were both fully connected, but the test network was less dense. The *overlap* (Figure 12A) datasplit had the most dense training and test networks.

Alongside structural differences, the datasets also differed in terms of size, number of positive and negative examples and number of unique TFs and targets (Table 4). A slight class imbalance was only observed in the *overlap* dataset and constituted 42 and 58 positive-to-negative label difference in the training and test set respectively. The *overlap* datasplit had the largest test set, followed by the *TG-split* and *TF-split* datasets. The *TFTG-split*

Table 4. Breakdown of the training, test and validation subsets across the four datasplits. Edge count distributions (postive/negative labels) and node uniqueness metrics (TFs/targets), quantify the datasplit characteristics.

Dataset	Overlap			TF-split			TFTG-split			TG-split		
Subset	Train	Test	Val	Train	Test	Val	Train	Test	Val	Train	Test	Val
Edges	1012	824	166	1012	620	174	1012	238	238	1012	634	162
Neg. edges	485	441	83	503	310	87	506	119	119	494	317	81
Pos. edges	527	383	83	509	310	87	506	119	119	518	317	81
Unique TFs	240	290	72	171	83	24	167	57	57	241	161	73
Unique targets	368	279	92	818	532	167	331	125	125	771	145	39

had the smallest test size (n=238) and was also used as validation set. The number of unique TFs and targets was largest in the *overlap* dataset and lowest in the *TFTG-split* set. The *TF-split* contained a relatively large number of Targets compared to TFs, in the train and test sets. The *TG-split* contained a relatively large number of of unique targets in the training set while in the test set the number of unique TFs exceeded the number of unique targets.

In summary, the four network partitioning strategies result in different edge count distributions and node distributions over the train and test sets. As a consequence, the train and test network structure can be significantly different. This influence the extent to which *topological* node information generalizes from the train network to the test network.

Parameter optimization

To investigate whether model increasing the number of layers hidden size improves GRN inference, a parameter search was conducted. The configuration space outlined in Table 2 was explored using a random search strategy. Parameter optimization was found to boost performance in *TFTG-split*, but only marginally, the AUROC score improved from 0.825 to 0.857 post optimization (Figure 16 and 20). The top 25 percent best performing configurations, tended to have a 8 total layers (average 8.4), with a smaller encoder module (average 3.134), and larger feedforward module (average 5.020). The hidden layer size was approximately 300 on average, but with a high standard deviation (SD) of 270. The relationship between performance and the hyperparameters was explored further by plotting hyperparameters as a grid-heatmap. Figure 16 shows that the AUROC score varies across different parameter combinations. The top-left panel shows that the performance deteriorates by increasing the depth beyond 10 total layers. The encoder and feedforward depth can vary with minimal impact on the performance as long as the aforementioned total depth limit is not exceeded. The learning rate of 0.0001 is primarily dependent on the hidden embedding size (bottom-left panel). In summary, parameter optimization was found to only marginally optimize performance in *TFTG-split*.

Performance comparison against scGREAT

To contrast the optimized model, scGREAT (Wang et al., 2024) was used as an additional base-line alongside the base-line established using random representations. The optimized model was found to perform better than scGREAT, compared to the respective baseline on every dataset, except for the *TG-split* dataset. The relative AUROC score compared to (random) base-line in *TFTG-split* was 0.35 and 0.17 for the optimized model and scGREAT respectively (Table 18).

Table 5. Descriptive statistics of the top 25 percent performing configurations.

	(n) Encoder Layers	(n) FF Layers	Hidden size	LR	AUROC	Total Layers
count	336	336	336	336	336	336
mean	3.134	5.244	291	1e-4	0.815	8.4
std	1.737	2.275	271	1e-4	0.015	2.9
min	1	2	25	1e-4	0.790	3
25%	2	3	50	1e-4	0.802	6
50%	3	5	200	1e-4	0.814	8
75%	4	7	600	1e-4	0.827	10
max	8	9	800	0.001	0.858	16

If we explore the loss trajectories it is visible that in *TG-split*, the optimized model is overfitting, as well as in the *overlap* datasplit (Figure 10). In general, the optimized model does not converge anymore beyond 2 epochs. The loss trajectory of scGREAT shows more variance over the epochs and the train loss tends to spike up at 10 epochs, resulting in a reduction in validation loss. Furthermore, the scGREAT model tends to take longer to converge compared to the optimized model. However, it must be noted that scGREAT predicts on human data, and the optimized model on *A. thaliana* data, making direct comparison difficult. The accuracy tends to be relatively stable over the epochs and does not respond significantly to the drop in validation loss. This suggests that while the model approximates the data as function of loss, this convergence is not resulting in a significant increase in accuracy.

The loss trajectories on random representations show that the optimized model is not able to converge (Figure 10), the average loss per epoch is stable at 50. This results in random performance, with an AUROC of 0.5 (Figure 18). This is in stark contrast to the performance achieved with the unoptimized models (Figure 5). However, reducing the learning rate from 0.001 to 0.0001, re-instates the model's ability to perform on random representations (Figure 11).

The scGREAT model was found to give higher than random base-line performance on random gene representations, in all datasplits, except *TFTG-split*. On average the model achieved an AUROC score of 0.8 across the datasplits (except in *TFTG-split*). Using original gene representations, the model does converge but still tends to overfit in the *overlap* datasplit and *TG-split*. In summary, the learning trajectories indicate that the neural networks converge rather quickly on the training set and in some cases tend to overfit. Model optimization has not improved this phenomena. Additionally, the higher than random base-line performance was reproduced in scGREAT.

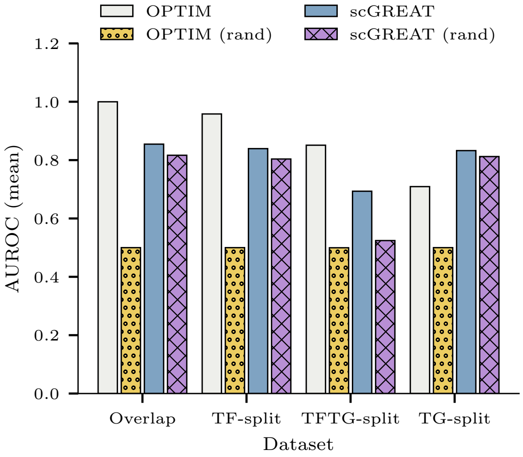

Figure 6: AUROC scores for the optimized model across the four datasplits compared against the performance achieved using random representations (rand) and against scGREAT.

t-SNE projections of the (co)expression representations

To explore the embedding space, and investigate whether biologically relevant signals were captured by the gene representations, t-SNE dimension reduction was conducted. t-SNE projections, visualized in Figure 17, demonstrate that both the expression and coexpression representations capture underlying biological relationships. The t-SNE graphs share a general structure; notably, Transposable Element (TE) genes (orange) tend to cluster together. Furthermore, the coexpression representation shows a distinct snake-like distribution in its t-SNE space.

Discussion

This study aimed to develop and evaluate gene representations for GRN inference in *A. thaliana*. Three gene representations: GO, expression, and coexpression, were developed and evaluated in four datasplits: *Overlap* (transductive), TF and *TG-split* (semi-transductive) and *TFTG-split* (inductive). It was found that framing GRN inference as a link prediction task resulted in a heavy reliance on *topological* node information. This suggests that the model is learning to exploit the existing network structure instead of learning biological relationships encoded in the gene representations.

Link-prediction models were found to achieve extremely high performance using random representations in the *overlap* datasplit, and in *TF-split*, realizing AUROC scores up to 0.99 and 0.7 respectively (Table 8). The use of random representations deprive the model from biological information encoded in the original gene representations. The model is thus able to exploit *topological* node information learned during training, for predicting links in the test network, resulting in a higher than random base-line performance.

In *TFTG-split* the base-line performance was random, as the nodes were completely separated between the train and test networks. This meant that the learned *topological* node information during training could not be generalized to the test network. This datasplit is therefore more suitable for evaluating the performance gain of the gene representations. The *TG-split* did contain *overlap* of target gene nodes between the train and test network, however, base-line performance was random. This contradicts the premise that complete separation is required to remove the reliance on *topological* node information. However, the structural differences between the train and test networks (Figure 12D), hindering the ability to generalize *topological* node information to the test network.

The reliance on topological information was further made evident in the scGREAT model, if the same data partition strategy (Figure 1) was used on their human data (Figure 6), and their gene representations were replaced with random representations. The scGREAT model achieved an AUROC of 0.8 on all datasplits, except *TFTG-split* (Table 18). Furthermore, the train and test network structures were shown to be similar across datasplits. This could explain why the performance in *TG-split* was higher than random, while this was not observed in the *A. thaliana TG-split*. This implies that network structural similarity is a prerequisite for the model to generalize learned *topological* node information to the test network.

This exploitation of topological information has been reported in literature by Turki and Wang, where it alone provided a solid basis for GRN inference in yeast and *E. coli*. This supports the premise that *topological* information is leveraged by ML models, making it unclear what the contribution of the gene representation is to GRN inference.

TFTG-split was found to be more informative for evaluating the performance derived from the gene representations, as the model could not rely on *topological* node information. However, most proposed link-prediction models are only evaluated in a transductive (i.e. *overlap* datasplit) test setting (Pratapa et al., 2020; Zhang et al., 2020b; Chen and Liu, 2022). In *TFTG-split* the test network contained unseen nodes, this represent an inductive test setting. This test setting have been used to predict Protein-Protein Interactions (PPIs) by Hamilton et al. using their GraphSAGE framework. They confirm that inductive tests indeed were more difficult (F1=0.61) than transductive tests, and elaborate that inductive tests require the model to align *topological* node information with secondary information **(e.g. expression data)**. Their reasoning, suggests that inductive tests still learn *topological* information, but require a more sophisticated integration with secondary information to be able to generalize *topological* information to the test network.

To generate inductive, transductive, and semi-transductive test settings, different partitioning strategies need to be employed. The ground truth network needs to be partitioned such that no nodes follow the rules outlined in Figure 1. This was found to greatly reduce the data available for training and testing. This effect is speculated to cause the difference between the training and test networks in the *TG-split* dataset in Figure 12D. The networks of the human data did not exhibit this difference in structure, which can be explained by the larger size of the ground truth network in combination with the the smaller number of TFs.

To address these challenges, a more sophisticated approach for partitioning data could be used. Linear programming could provide a method for optimizing the number of edges over the train/val/test networks given the constraints imposed as result of the test settings in Figure 1, however, it does not necessarily guarantee structural similarity (Dantzig, 1963). This points to a fundamental problem in network partitioning: find k subnetworks such that the number of edges between them is minimized. Various network partitioning methods have already been developed, and can be used to maximize compliance to the constraints (Condon and Karp, 1999), while preserving the network structure. This could improve the the model's ability to learn a robust gene representation, and align *topological* node information with secondary information.

Optimizing the partitioning of the data could also improve the model's ability to converge, as this was found to be a consistent problem. All tested model architectures (Figure 2 and 3) tended to converge after two epochs of training (Figure 10 and 9). This rapid convergence often came at the expense of performance on the validation set, even resulting in overfitting on the *overlap* datasplit and TG-split.

The performance of the individual gene representation was found to differ across the datasplits. The *TFTG-split* showed that generalizable expression patterns were learned, with the expression representation outperforming the GO and coexpression representations(Table 10). The GO representation consistently achieved high performance across all datasets. Its utility for GRN inference was not found to be significantly dependent on the type of evidence code used for its construction (Table 12). However, a minor improvement in AUROC score was observed if automatically inferred annotations (IEA) were excluded. The coexpression achieved the lowest performance overall (Table 10), which can be explained by the loss of information that occurs when the correlation matrix is thresholded to select neighbors. The results show that gene representations demonstrate variable results across datasplits, and thus differ in their utility for GRN inference in the respective datasplit.

It was found that combining gene representations, specifically the GO and expression representations, improved GRN inference(Table 14 and 16). The pairing the GO or expression representation with the coexpression representation, achieved lower performance except in *TG-split* (Table 14). The GO+G2V representation performed better on the *TG-split* datatset compared to the GO+EXP representation. This suggests that expression representation is not able to generalize to the test network in *TG-split.* Combining all representations improved performance only marginally (Table 16) compared to GO+EXP paired representation. Additionally, the method of vector aggregation was found to marginally affect the performance. In general, vector concatenation provided was better than vector addition for predicting regulatory interactions (Table 16).

For the construction of (co)expression representations it was decided to limit the expression samples to the *A. thaliana* Col-0 eco- and genotype, and a relatively high cut-off was set at 90 percent uniquely mapped reads. This reduced the number of samples from 28 thousand to 1,343. Future research could include all samples, as a larger and more diverse dataset could improve the robustness of the gene representations.

Increasing the number of layers and the hidden embed size, was not found to improve GRN inference necessarily (Figure 16). Parameter optimization showed that optimal performing models balanced encoder and decoder size, with maximal performance at a combined size of ten (Figure 5 and 20). Furthermore, a higher AUROC score tended to be achieved with a (hidden) embedding size of 400. The optimized model had marginally improved performance over the combined (un-optimized) model (Table 18 and 16), but did not seem to leverage *topological* information anymore. However, if the learning rate was adapted to from 0.001 to 0.0001 the model was able to perform well on random representations.

Conclusion

This study aimed to develop and evaluate gene representations for GRN inference in *A. thaliana.* Three gene representations: GO, expression, and coexpression, were developed and evaluated in four datasplits: *Overlap* (transductive), TF and *TG-split* (semi-transductive) and *TFTG-split* (inductive). By substituting the representations with random embeddings, it was found that framing GRN inference as a link-prediction task resulted in a heavy reliance on *topological* node information. Suggesting that the model is learning to exploit the existing network structure instead of learning biological relationships encoded in the gene representations. Furthermore, this phenomena was demonstrated in the scGREAT model (Wang et al., 2024).

To combat this, performance was evaluated in *TG-split*, as this setting was found to remove the ability to exploit *topological* node information. The degree to which *topology* could be exploited in the datasplits was investigated using random representations. Furthermore, the expression representation was able to learn a gene representation that generalized to unseen genes, in *TFTG-split.* The GO representation performed better in the *overlap* datasplit, and in *TF* and *TG-split*, where nodes were already seen. The coexpression embedding was found to be inferior to the GO and expression representations, potentially due to the loss of information that occurs during the neighbor selection. Additionally, it was found that combining representations improved GRN inference.

Generation of the test settings required a significant portion of the data to be excluded. This could be improved by using linear programming (Dantzig, 1963) and/or existing network partitioning methods (Condon and Karp, 1999). This loss of data is also speculated to have resulted in rapid convergence, and in some cases, overfitting. Furthermore, the number of expression samples used for the construction of the (co)epxression representations was greatly reduced to reduce the computational load, potentially impacting performance.

This study identified three areas of improvement: combining *topological* information with secondary data sources, improving data partitioning, and utilizing all available expression samples in the plantrna database (Yu et al., 2022; Zhang et al., 2020a). Improvements in these areas would allow the model to learn more robust gene representations that generalize better to unseen genes.

Appendix

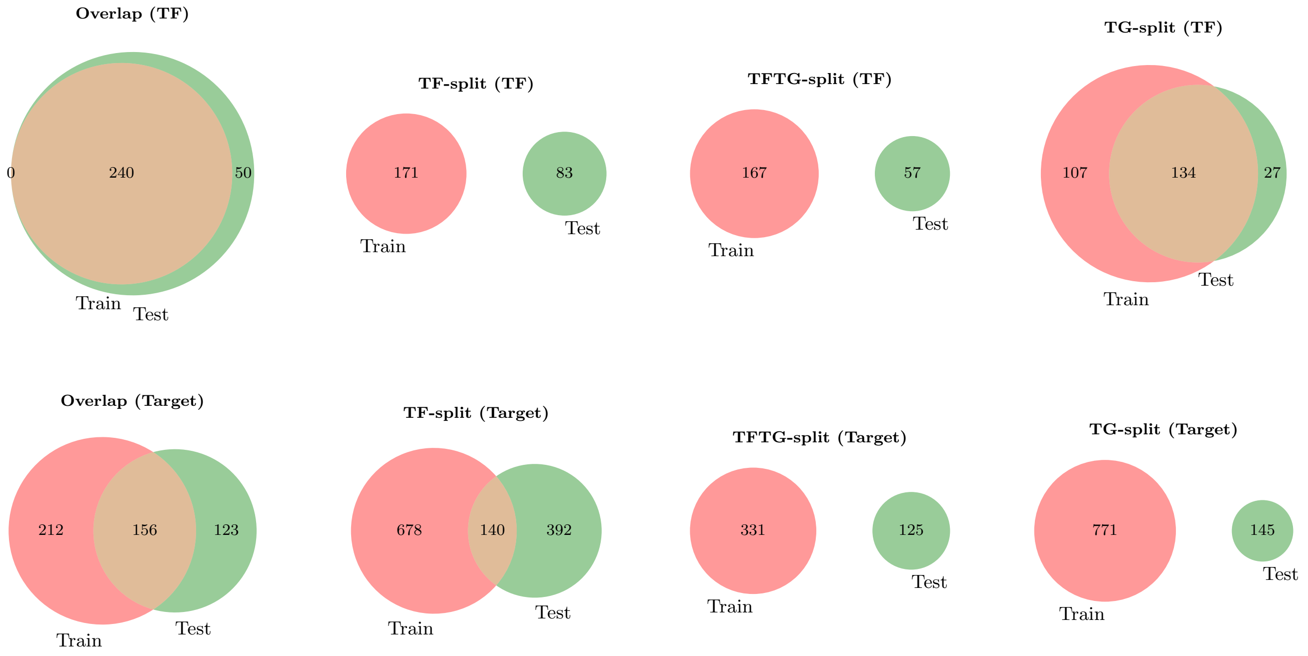

Figure 7: Overlap in TFs and TGs differ acrosss the datasplits to model the three test setting the context of regulatory gene prediction as outlined in Figure 1.

Table 6. Categories, descriptions of evidence codes, and the number of annotations of the respective evidence code. Each code represents a method by which biological function or relationships are inferred. Experimental and computational evidence codes are most prominent followed by automically-generated annotations.

	Category	Description	count
Code			
ISS	Computational	Inferred from sequence or structural similarity.	10887
IPI	Experimental	Inferred from physical interaction (e.g., protein-protein interactions).	27339
ISM	Computational	Inferred from sequence model (e.g., HMM profiles).	37754
IEA	Automatically-Generated	Inferred from electronic annotation (automatically generated, not human-reviewed).	38876
IMP	Experimental	Inferred from mutant phenotype.	40065
IBA	Computational	Inferred from biological aspect of ancestor.	60897
ND	Curatorial	No biological data available.	20874
IDA	Experimental	Inferred from direct assay (e.g., enzyme assays, microscopy).	32390
IGI	Experimental	Inferred from genetic interaction.	8999
TAS	Author Statement	Traceable author statement (from a published paper with a traceable source).	12744
HDA	High-Throughput	Inferred from high-throughput direct assay.	25947
IEP	Experimental	Inferred from expression pattern.	9178
IC	Curatorial	Inferred by a curator.	246
HEP	High-Throughput	Inferred from high-throughput expression pattern.	950
NAS	Author Statement	Non-traceable author statement (from a paper without a traceable source).	1422
RCA	Computational	Inferred from reviewed computational analysis.	159

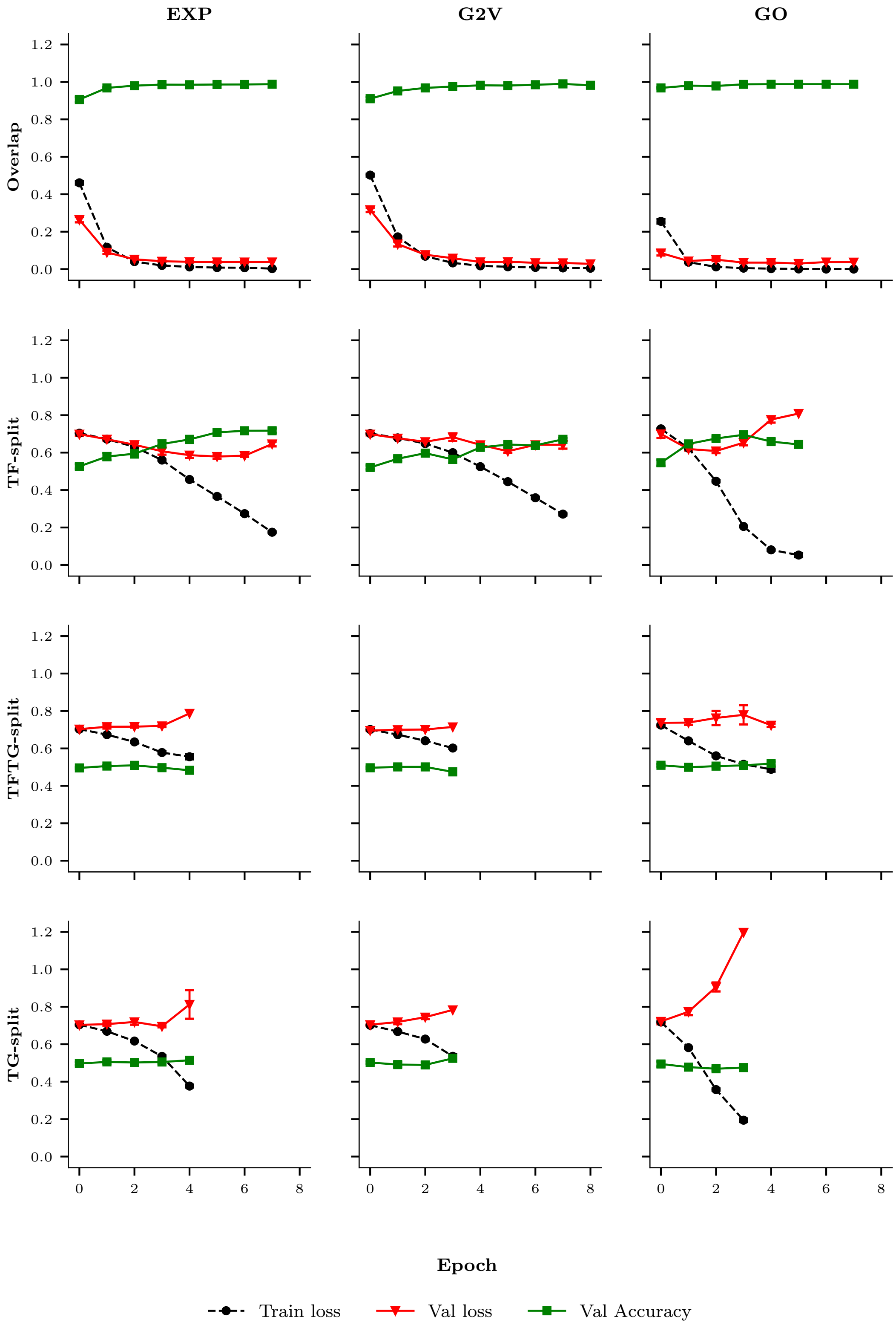

Figure 8: The train loss, validation loss and validation accuracy for **random** representations over the epochs. The scores were recorded for each datasplits, and show a higher than random base-line performance. Furthermore, a general trend of overfitting that is major in the *TG-split* datasplit is visible.

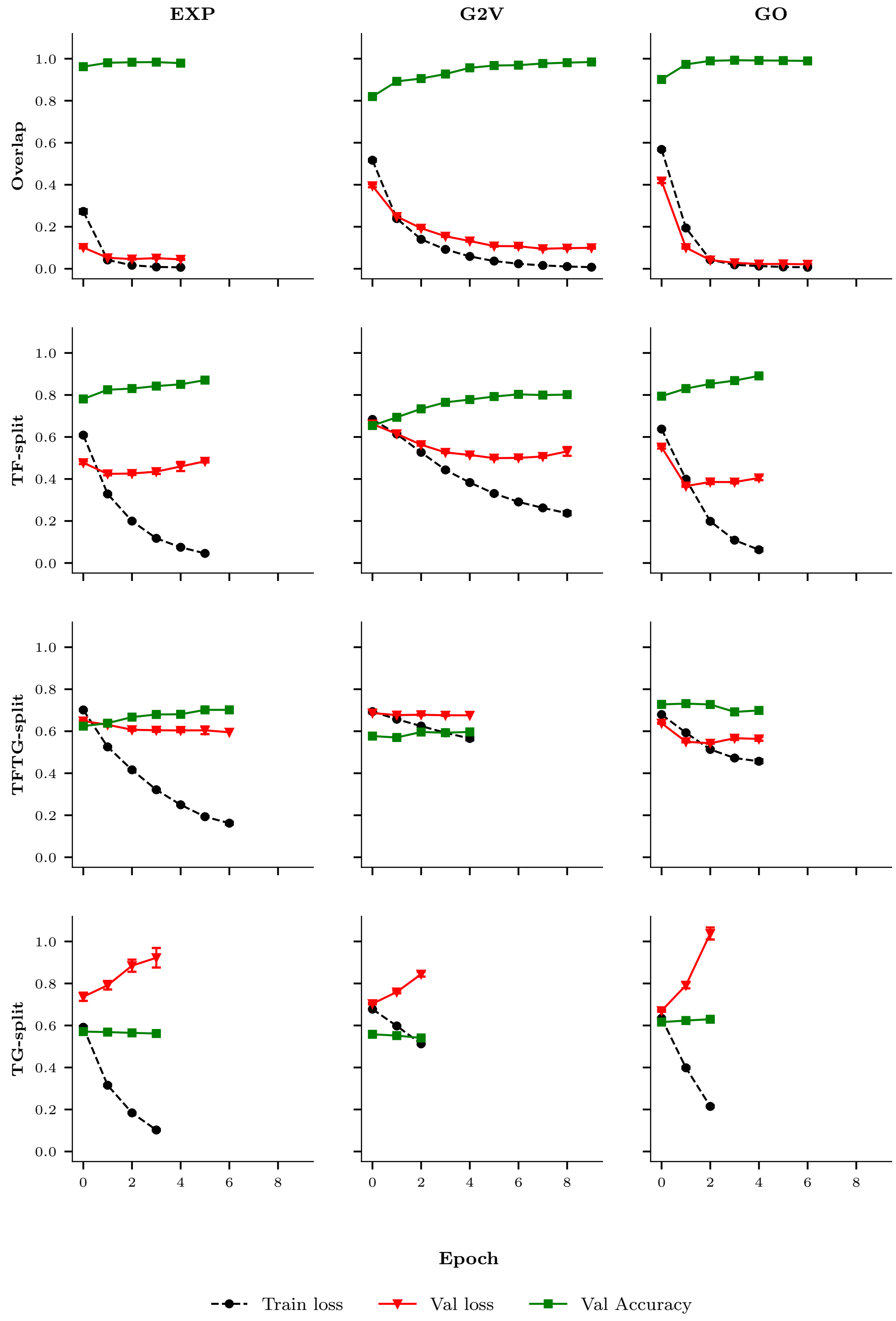

Figure 9: The train loss, validation loss and validation accuracy for each **individual** gene representation over the epochs. The scores were recorded for each datasplit and show a general trend of overfitting that is major in *TG-split*.

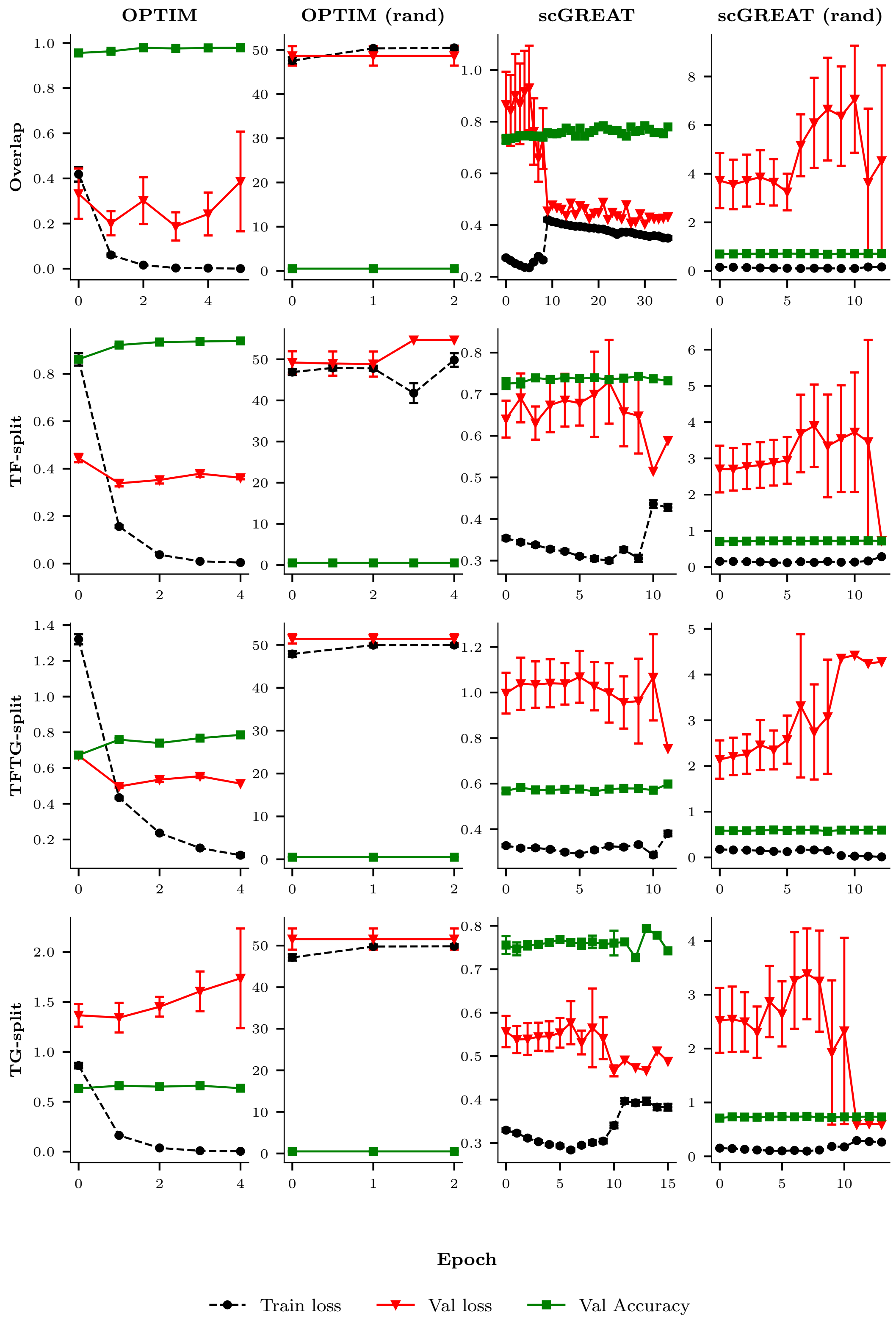

Figure 10: The train loss, validation loss, and validation accuracy for the **optimized** model (OPTIM) predicting on all gene representations compared to (**i**) random embeddings: OPTIM (rand), (**ii**) scGREAT transformer model (human data but same method for train and test split), and (**iii**) scGREAT random embeddings: scGREAT (rand). The optimized model does not show the higher than random base-line performance in the any of the datasplits. Additionally the optimized model shows less sign of overfitting compared to the models shown in Figure 9.

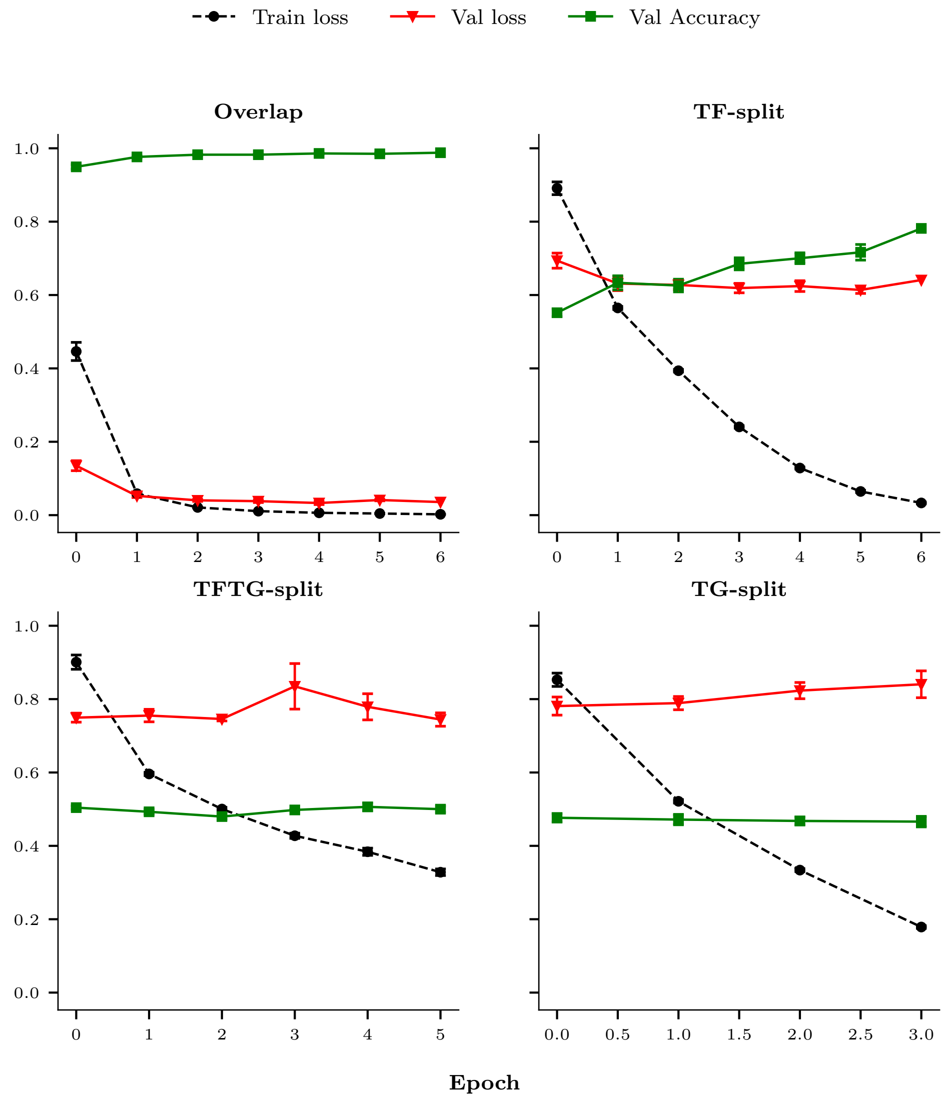

Figure 11: The train loss, validation loss, and validation accuracy, for the optimized model trained using random representations and with a learning rate of 0.0001. The model does converge on the data as opposed to the results in Figure 10.

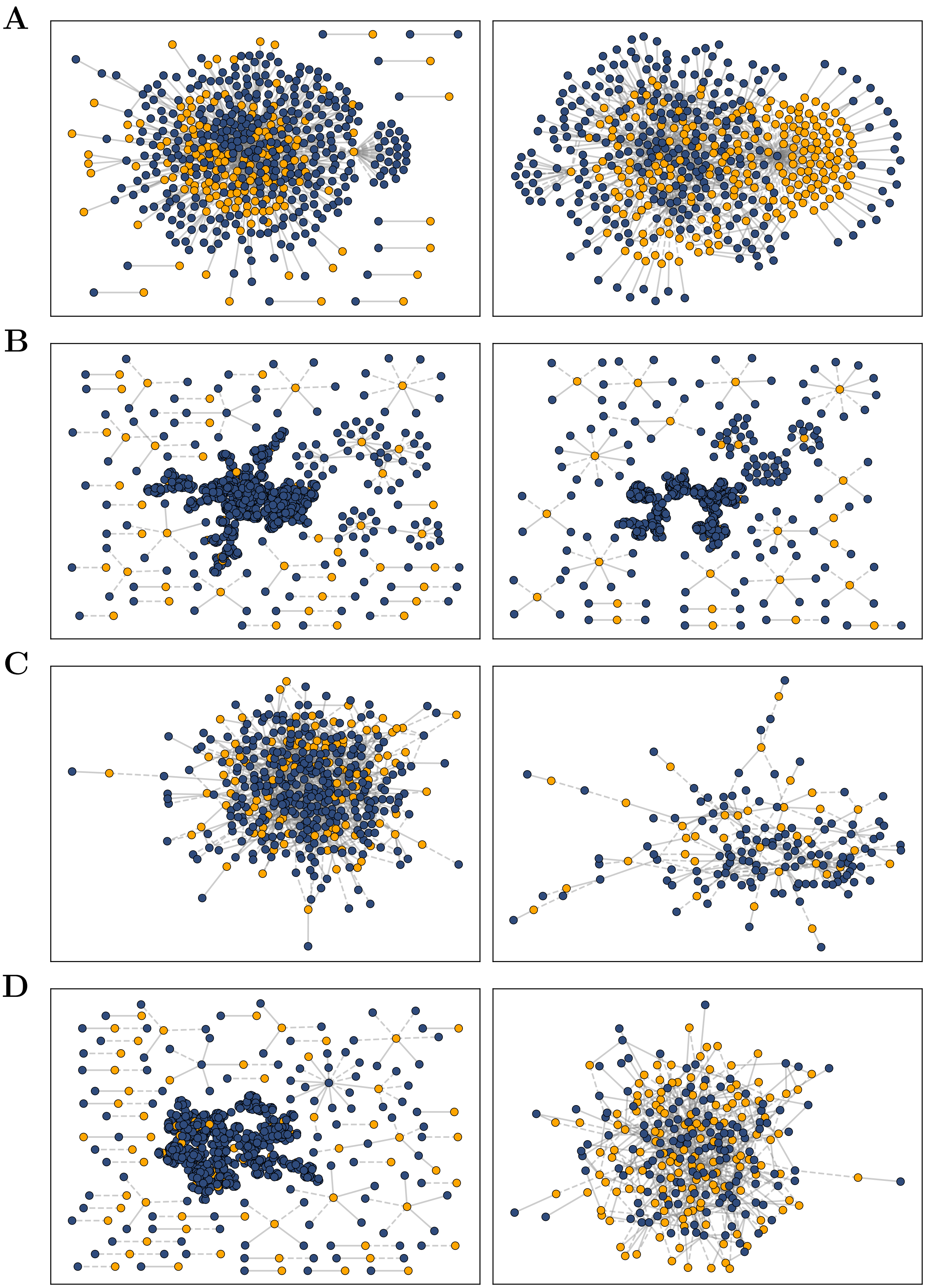

Figure 12: Train and test sub-networks for the four different datasets of *A. thaliana* GRN, **A**: *overlap*, **B**: *TF-split*, **C**: *TFTG-split*, and **D**: *TG-split*. Nodes are colored yellow for TFs and Targets blue, edges are straight line for interaction (positive, 1) and dashed for no interaction (negative, 0).

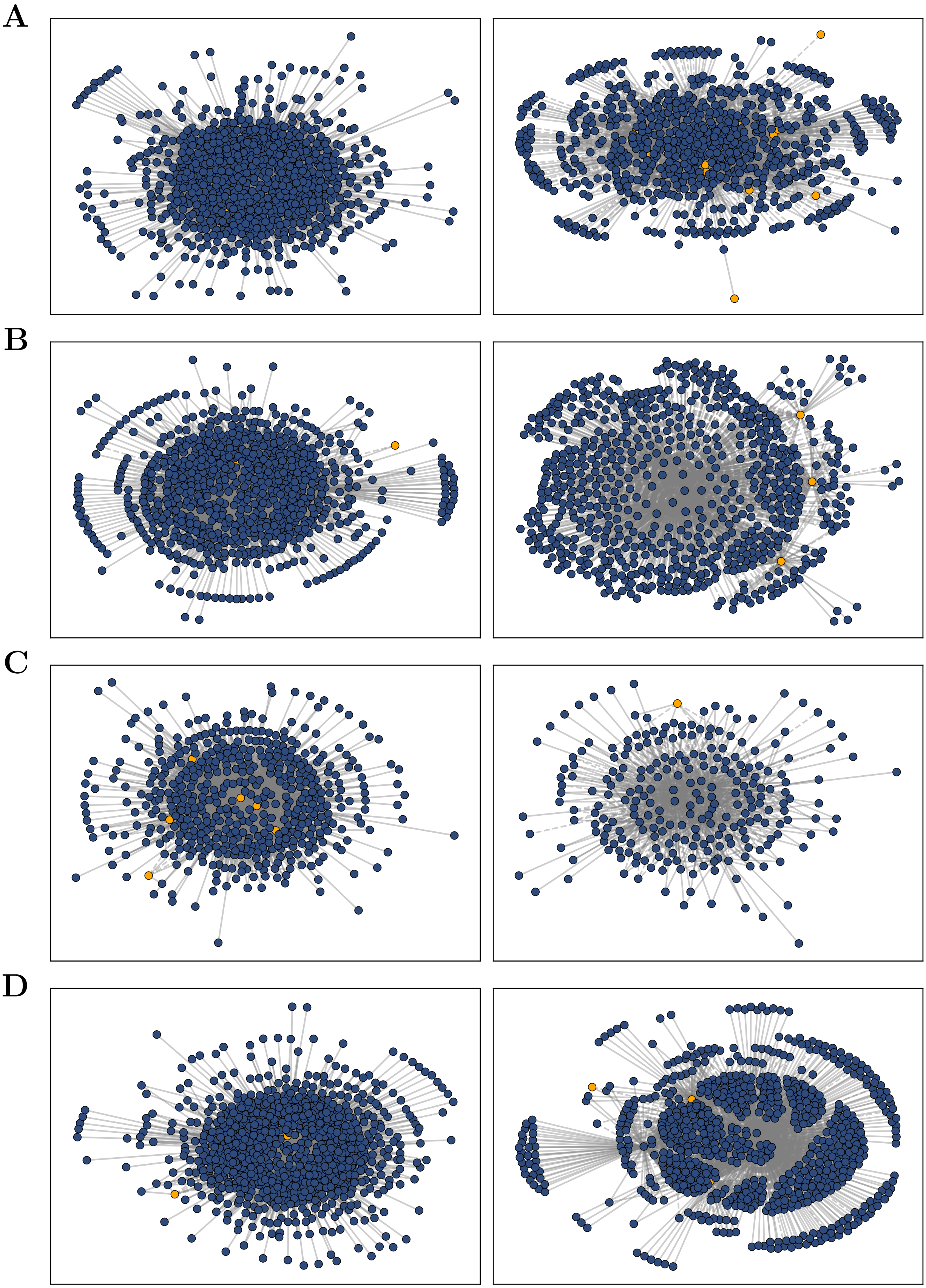

Figure 13: Train and test sub-networks for the four different datasets made from the scGREAT human GRN, **A**: *overlap*, **B**: *TF-split*, **C**: *TFTG-split*, and **D**: *TG-split*. Nodes are colored yellow for TFs and Targets blue, edges are straight line for interaction (positive, 1) and dashed for no interaction (negative, 0).

Table 7. Distribution of gene types in the expression dataset, and their median and mean expression values. Gene types were broken down as, **mRNA** (messenger RNA), **TE gene** (transposable element gene), **lncRNA** (long non-coding RNA), **aslncRNA** (antisense long non-coding RNA), pseudogene, **pre-tRNA** (precursor transfer RNA), **snoRNA** (small nucleolar RNA), **snRNA** (small nuclear RNA), and **rRNA** (ribosomal RNA).

		genes	counts	counts
		(n)	(med)	(mean)
mRNA	protein coding	27655	9.907	19.576
TE gene	TE gene	3901	0.000	0.318
lncRNA	non-coding RNA	2444	0.000	0.374
aslncRNA	non-coding RNA	1037	0.517	3.959
pseudogene	pseudogene	927	0.000	1.294
pre-tRNA	other RNA	689	0.000	0.797
snoRNA	other RNA	287	1.091	5.168
other RNA	other RNA	221	0.632	8.584
snRNA	other RNA	80	0.000	1.595
asRNA	other RNA	78	0.000	0.998
rRNA	other RNA	15	23.071	49.340

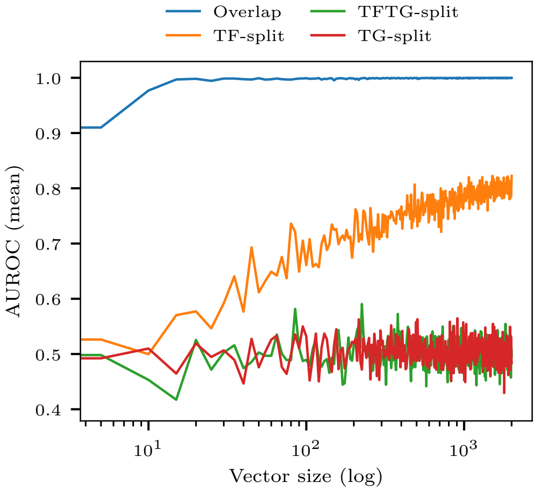

Figure 14: The AUROC score (y-axis) depends on vector length, AUROC scores were achieved using random generated binary representations of variable length (x-axis).

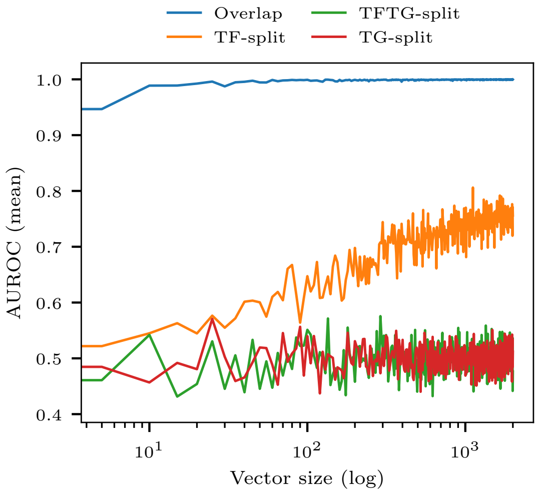

Figure 15: Continuous random vectors with higher number of dimensions enable the the the network to better differentiate between genes.

Table 8. AUROC and AUPRC (mean, SEM) for a **random** gene representations. For different models across datasplits.

			ROC		PRC
		(mean)	(SEM)	(mean)	(SEM)
Dataset	Model				
Overlap	EXP	0.999	0.000	0.999	0.000
	G2V	1.000	0.000	1.000	0.000
	GO	1.000	0.000	1.000	0.000
TF-split	EXP	0.762	0.005	0.780	0.006
	G2V	0.742	0.009	0.768	0.009
	GO	0.820	0.002	0.871	0.002
TFTG-split	EXP	0.506	0.012	0.516	0.011
	G2V	0.504	0.008	0.514	0.011
	GO	0.513	0.010	0.521	0.010
TG-split	EXP	0.478	0.007	0.500	0.007
	G2V	0.512	0.008	0.517	0.007
	GO	0.499	0.006	0.495	0.007

Table 9. Performance on **random** created gene representations as function of: Precision, Recall, F1-score, and Accuracy, for different models across datasplits.

		Precision	Recall	F1-score	Accuracy
Overlap	EXP	0.986	0.996	0.991	0.992
	G2V	0.988	0.997	0.992	0.993
	GO	0.987	1.000	0.993	0.994
TF-split	EXP	0.676	0.681	0.679	0.678
	G2V	0.612	0.734	0.667	0.634
	GO	0.709	0.754	0.731	0.722
TG-split	EXP	0.488	0.368	0.420	0.491
	G2V	0.500	0.556	0.527	0.500
	GO	0.511	0.384	0.439	0.508
TFTG-split	EXP	0.506	0.632	0.562	0.508
	G2V	0.498	0.306	0.379	0.499
	GO	0.506	0.518	0.512	0.506

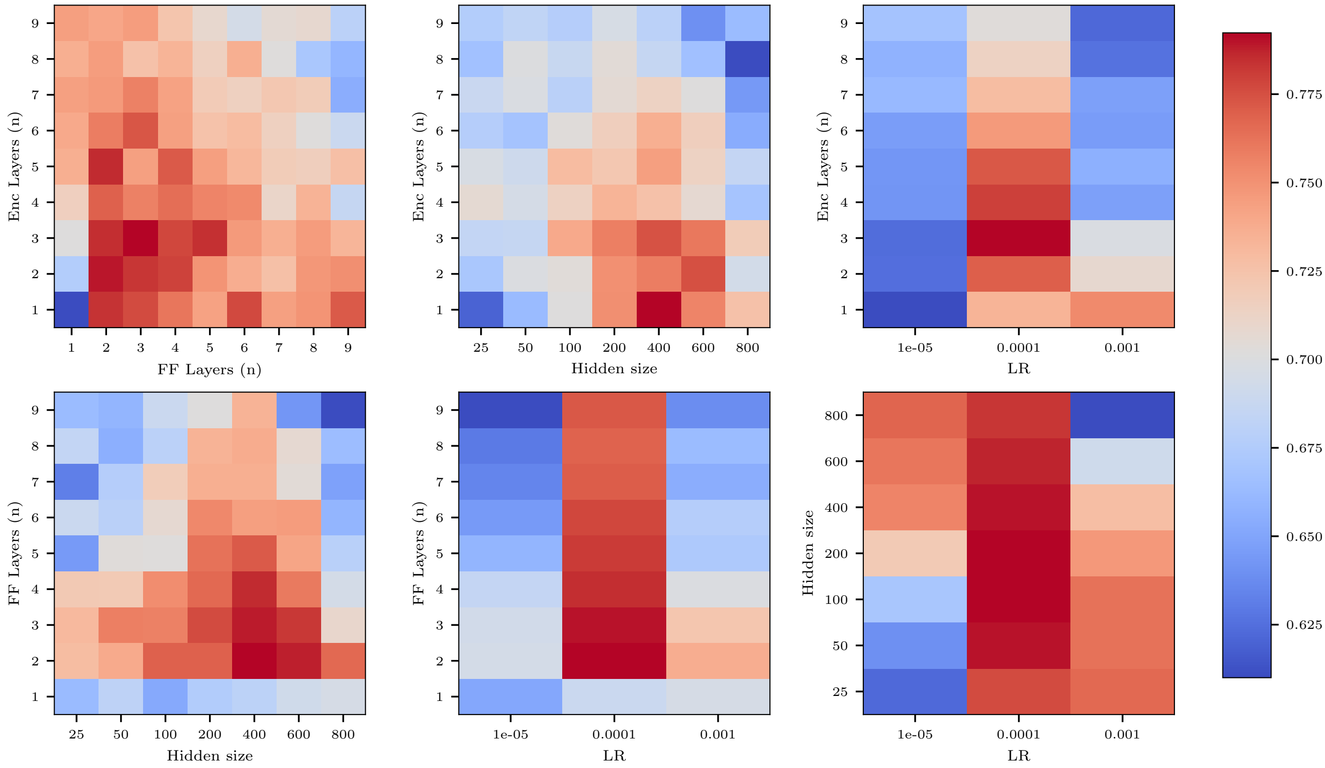

Figure 16: Configurations were tested using a random search technique and visualized by the AUROC (color gradient). The axis show the hyperparameters that were tested, namely, encoder layer, feedforward layers, hidden size, and learing rate (LR).

Table 10. ROC and PRC (mean and SEM) for **individual** gene representations across datasplits.

			ROC		PRC
		(mean)	(SEM)	(mean)	(SEM)
datasplit	Model				
Overlap	EXP	1.000	0.000	1.000	0.000
	G2V	0.996	0.000	0.993	0.000
	GO	1.000	0.000	1.000	0.000
TF-split	EXP	0.917	0.002	0.929	0.002
	G2V	0.819	0.002	0.792	0.003
	GO	0.945	0.001	0.951	0.000
TFTG-split	EXP	0.756	0.007	0.772	0.006
	G2V	0.618	0.005	0.613	0.007
	GO	0.734	0.002	0.808	0.001
TG-split	EXP	0.553	0.006	0.545	0.008
	G2V	0.573	0.004	0.591	0.004
	GO	0.728	0.002	0.749	0.002

Table 11. Precision, Recall, F1-score, and Accuracy for **individual** gene representations across datasplits.

		Precision	Recall	F1-score	Accuracy
Overlap	EXP	0.991	0.992	0.991	0.992
	G2V	0.990	0.990	0.990	0.990
	GO	0.997	0.988	0.993	0.993
TF-split	EXP	0.863	0.823	0.843	0.846
	G2V	0.743	0.754	0.749	0.747
	GO	0.869	0.875	0.872	0.871
TG-split	EXP	0.536	0.677	0.598	0.545
	G2V	0.524	0.732	0.611	0.534
	GO	0.572	0.844	0.682	0.606
TFTG-split	EXP	0.690	0.670	0.680	0.684
	G2V	0.592	0.611	0.601	0.595
	GO	0.710	0.629	0.667	0.686

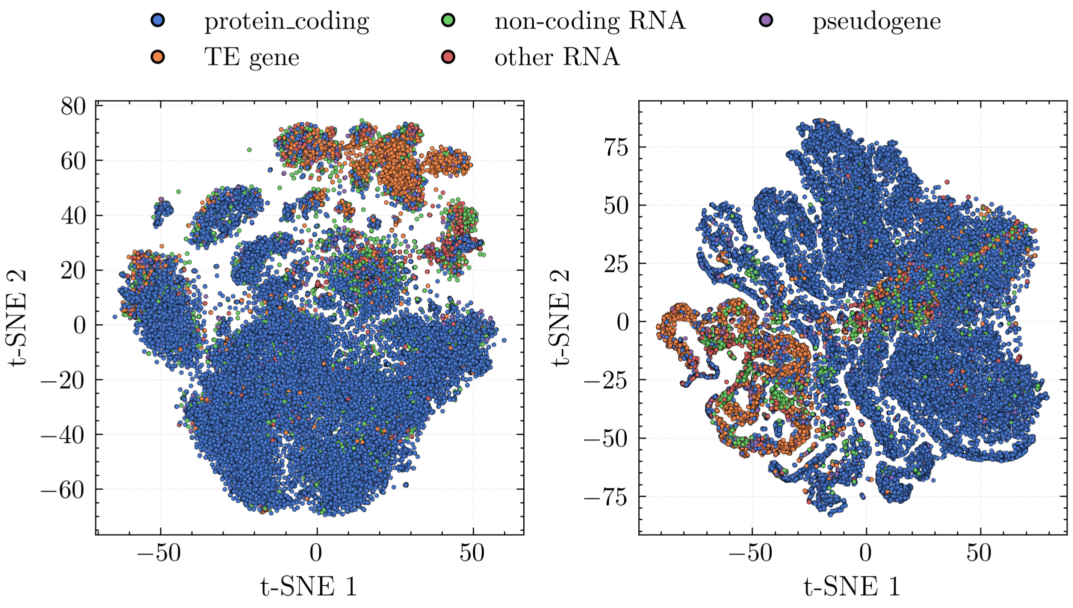

Figure 17: t-SNE dimension reduction of the gene expression values (left) and gene coexpression embeddings (right). Data points were colored according to the gene type.

Table 12. Effect of filtering on **GO-term evidence codes** on AUROC and AUPRC shows to be only marginally present. The following situations were tested: (**All GO-terms**) GO representation made from all available evidence codes, (**wo/comp**) without the computationally derived evidence codes, and (**wo/IEA**) without the automatically inferred evidence codes.

		Precision	Recall	F1-score	Accuracy
Overlap	All GO-terms	0.998	0.987	0.992	0.993
	w/o Comp	0.993	0.987	0.990	0.991
	w/o IEA	0.997	0.986	0.992	0.992
TF-split	All GO-terms	0.874	0.908	0.891	0.882
	w/o Comp	0.869	0.878	0.874	0.865
	w/o IEA	0.865	0.900	0.882	0.873
TG-split	All GO-terms	0.563	0.858	0.679	0.592
	w/o Comp	0.547	0.876	0.674	0.572
	w/o IEA	0.569	0.854	0.683	0.600
TFTG-split	All GO-terms	0.727	0.614	0.665	0.689
	w/o Comp	0.709	0.619	0.661	0.680
	w/o IEA	0.742	0.625	0.679	0.702

Table 13. Effect of filtering on **GO-term evidence codes** on Precission, Recall, F1-score, and accuracy, shows to be only marginally present. The following situations were tested: (**All GO-terms**) GO embeddings made from all available evidence codes, (**wo/comp**) without the computationally derived evidence codes, and (**wo/IEA**) without the automatically inferred evidence codes.

			ROC		PRC
		(mean)	(SEM)	(mean)	(SEM)
Dataset	Model				
Overlap	All GO-terms	1.000	0.000	1.000	0.000
	w/o Comp	1.000	0.000	1.000	0.000
	w/o IEA	1.000	0.000	1.000	0.000
TF-split	All GO-terms	0.952	0.001	0.959	0.001
	w/o Comp	0.940	0.001	0.947	0.001
	w/o IEA	0.949	0.001	0.957	0.001
TFTG-split	All GO-terms	0.730	0.001	0.808	0.001
	w/o Comp	0.711	0.002	0.787	0.002
	w/o IEA	0.734	0.002	0.811	0.001
TG-split	All GO-terms	0.725	0.001	0.753	0.001
	w/o Comp	0.670	0.001	0.712	0.002
	w/o IEA	0.722	0.002	0.751	0.003

Table 14. AUROC and AUPRC (mean and SEM) for **paired** gene representations.

			ROC		PRC
		(mean)	(SEM)	(mean)	(SEM)
datasplit	Model				
Overlap	EXP+G2V	1.00	0.00	1.00	0.00
	GO+EXP	1.00	0.00	1.00	0.00
	GO+G2V	1.00	0.00	1.00	0.00
TF-split	EXP+G2V	0.92	0.00	0.93	0.00
	GO+EXP	0.95	0.00	0.96	0.00
	GO+G2V	0.95	0.00	0.95	0.00
TFTG-split	EXP+G2V	0.76	0.01	0.77	0.01
	GO+EXP	0.82	0.00	0.85	0.01
	GO+G2V	0.74	0.00	0.81	0.00
TG-split	EXP+G2V	0.55	0.00	0.55	0.00
	GO+EXP	0.64	0.01	0.63	0.01
	GO+G2V	0.73	0.00	0.76	0.00

Table 15. Performance metrics on **paired** gene representations as function of: Precision, Recall, F1-score, and Accuracy, for different models across datasplits.

		Precision	Recall	F1-score	Accuracy
Overlap	EXP+G2V	0.992	0.996	0.994	0.995
	GO+EXP	0.992	0.994	0.993	0.994
	GO+G2V	0.999	0.999	0.999	0.999
TF-split	EXP+G2V	0.862	0.831	0.846	0.849
	GO+EXP	0.906	0.875	0.890	0.892
	GO+G2V	0.901	0.862	0.881	0.884
TG-split	EXPG2V	0.531	0.665	0.590	0.539
	GO+EXP	0.551	0.841	0.666	0.578
	GO+G2V	0.556	0.882	0.682	0.589
TFTG-split	EXP+G2V	0.685	0.697	0.691	0.688
	GO+EXP	0.729	0.743	0.736	0.733
	GO+G2V	0.736	0.629	0.679	0.702

Table 16. AUROC and AUPRC (mean, SEM) metrics for **combined** representations (all) separated by concatenation method and contrasted to a transformer based model.

			ROC		PRC
		(mean)	(SEM)	(mean)	(SEM)
Dataset	Model				
TF-split	Linear-Add	0.954	0.001	0.963	0.001
	Linear-Conc	0.958	0.000	0.966	0.000
	TRANSF-Add	0.917	0.006	0.929	0.007
	TRANSF-Conc	0.913	0.006	0.930	0.006
TFTG-split	Linear-Add	0.813	0.006	0.841	0.006
	Linear-Conc	0.825	0.005	0.859	0.004
	TRANSF-Add	0.756	0.012	0.797	0.009
	TRANSF-Conc	0.749	0.014	0.783	0.017
TG-split	Linear-Add	0.631	0.004	0.630	0.009
	Linear-Conc	0.667	0.004	0.671	0.007
	TRANSF-Add	0.595	0.008	0.581	0.011
	TRANSF-Conc	0.604	0.011	0.594	0.016

Table 17. Performance metrics for **combined** representations (all) separated by concatenation method and contrasted to a transformer based model. Performance is expressed as a function of Precision, Recall, F1-score, and accuracy.

		Precision	Recall	F1-score	Accuracy
Overlap	Linear-Add	0.993	0.996	0.994	0.995
	Linear-Conc	0.992	0.997	0.995	0.995
	TRANSF-Add	0.988	0.991	0.989	0.990
	TRANSF-Conc	0.988	0.986	0.987	0.988
TF-split	Linear-Add	0.907	0.870	0.888	0.890
	Linear-Conc	0.906	0.881	0.893	0.895
	TRANSF-Add	0.873	0.835	0.854	0.857
	TRANSF-Conc	0.852	0.839	0.846	0.847
TG-split	Linear-Add	0.549	0.814	0.656	0.573
	Linear-Conc	0.551	0.882	0.678	0.582
	TRANSF-Add	0.536	0.781	0.636	0.553
	TRANSF-Conc	0.543	0.787	0.643	0.563
TFTG-split	Linear-Add	0.734	0.708	0.721	0.726
	Linear-Conc	0.759	0.729	0.744	0.749
	TRANSF-Add	0.651	0.734	0.690	0.670
	TRANSF-Conc	0.629	0.753	0.686	0.655

Table 18. AUROC and AUPRC of the **Optimized** (OPTIM) model and **scGREAT** model in experimental setting and using random representations indicated with a "*" symbol. scGREAT's human label data is divided using the same data partitioning strategies as in *A. thaliana*.

			ROC		PRC
		(mean)	(SEM)	(mean)	(SEM)
datasplit	Model				
Overlap	OPTIM	1.000	0.000	1.000	0.000
	OPTIM (*)	0.500	0.000	0.732	0.000
	scGREAT	0.855	0.002	0.872	0.003
	scGREAT (*)	0.817	0.005	0.832	0.005
TF-split	OPTIM	0.958	0.001	0.965	0.001
	OPTIM (*)	0.500	0.000	0.750	0.000
	scGREAT	0.839	0.007	0.864	0.007
	scGREAT (*)	0.804	0.006	0.819	0.007
TFTG-split	OPTIM	0.851	0.004	0.879	0.003
	OPTIM (*)	0.500	0.000	0.700	0.050
	scGREAT	0.693	0.007	0.725	0.009
	scGREAT (*)	0.524	0.001	0.513	0.004
TG-split	OPTIM	0.709	0.004	0.706	0.007
	OPTIM (*)	0.500	0.000	0.750	0.000
	scGREAT	0.833	0.004	0.858	0.004
	scGREAT (*)	0.812	0.005	0.835	0.006

Table 19. Performance the **Optimized** (OPTIM) model and **scGREAT** model in experimental setting and using random representations indicated by a "*" symbol. scGREAT's human label data is divided using the same data partitioning strategies as in *A. thaliana*.

		Precision	Recall	F1-score	Accuracy
Overlap	OPTIM	0.994	0.994	0.994	0.994
	OPTIM (*)	0.465	0.400	0.430	0.507
	scGREAT	0.762	0.753	0.757	0.756
	scGREAT (*)	0.742	0.736	0.739	0.738
TF-split	OPTIM	0.908	0.874	0.891	0.893
	OPTIM (*)	0.500	0.700	0.583	0.500
	scGREAT	0.839	0.633	0.722	0.756
	scGREAT (*)	0.788	0.618	0.693	0.726
TG-split	OPTIM	0.567	0.855	0.682	0.601
	OPTIM (*)	0.500	0.600	0.545	0.500
	scGREAT	0.738	0.715	0.726	0.730
	scGREAT (*)	0.730	0.713	0.721	0.724
TFTG-split	OPTIM	0.779	0.749	0.763	0.768
	OPTIM (*)	0.500	0.700	0.583	0.500
	scGREAT	0.617	0.651	0.634	0.627
	scGREAT (*)	0.515	0.593	0.551	0.521

Table 20. Top 15 configurations of encoder, feedforward, and hidden size as function of AUROC score trained at a learning rate of 0.0001.

	Encoder	FF	Hidden	LR	Total	AUROC
	Layers (n)	Layers (n)		size	Layers	
1	1	2	400	0.0010	3	0.8577
2	3	4	400	0.0001	7	0.8500
3	4	2	800	0.0001	6	0.8493
4	3	4	800	0.0001	7	0.8487
5	2	2	800	0.0001	4	0.8471
6	2	3	800	0.0001	5	0.8461
7	2	3	600	0.0001	5	0.8458
8	3	7	400	0.0001	10	0.8444
9	3	3	600	0.0001	6	0.8443
10	5	4	400	0.0001	9	0.8440
11	1	2	50	0.0010	3	0.8440
12	3	3	400	0.0001	6	0.8438
13	1	3	100	0.0010	4	0.8436
14	3	7	600	0.0001	10	0.8433
15	3	5	800	0.0001	8	0.8430

Acknowledgments

I would like to express my heartfelt gratitude to everyone who took the time to read this thesis and provide valuable feedback. A special thanks goes to Dick de Ridder and Rens Holmer for their insightful contributions. This project has taught me the importance of critically evaluating both my own work and that of others, as well as the perseverance required to overcome numerous challenges along the way. I am particularly grateful to Enzo den Ouden and Aaron Pieter Korn for proofreading this thesis and offering a fresh perspective on my findings.

References

Badia-i Mompel, P., Wessels, L., Müller-Dott, S., Trimbour, R., Ramirez Flores, R. O., Argelaguet, R., and Saez-Rodriguez, J. (2023). Gene regulatory network inference in the era of single-cell multi-omics. *Nature Reviews Genetics*, 24(11):739–754. Publisher: Nature Publishing Group.

Berardini, T. Z., Mundodi, S., Reiser, L., Huala, E., Garcia-Hernandez, M., Zhang, P., Mueller, L. A., Yoon, J., Doyle, A., Lander, G., Moseyko, N., Yoo, D., Xu, I., Zoeckler, B., Montoya, M., Miller, N., Weems, D., and Rhee, S. Y. (2004). Functional annotation of the Arabidopsis genome using controlled vocabularies. *Plant physiology*, 135(2):745–755. Place: United States.

Bergstra, J., Bergstra, J., Bengio, Y., and Bengio, Y. Random Search for Hyper-Parameter Optimization.

Chen, G. and Liu, Z.-P. (2022). Graph attention network for link prediction of gene regulations from single-cell RNA-sequencing data. *Bioinformatics*, 38(19):4522–4529.

Condon, A. and Karp, R. (1999). Algorithms for graph partitioning on the planted partition model. *Random Struct. Algorithms.*

Dantzig, G. B. (1963). Linear Programming and Extensions. Technical report, RAND Corporation.

Du, J., Jia, P., Dai, Y., Tao, C., Zhao, Z., and Zhi, D. (2019). Gene2vec: distributed representation of genes based on co-expression. *BMC Genomics*, 20(1):82.

Fuxman Bass, J. I., Reece-Hoyes, J. S., and Walhout, A. J. (2016). Gene-Centered Yeast One-Hybrid Assays. *Cold Spring Harbor protocols*, 2016(12):pdb.top077669.

Georgakopoulos-Soares, I., Deng, C., Agarwal, V., Chan, C. S. Y., Zhao, J., Inoue, F., and Ahituv, N. (2023). Transcription factor binding site orientation and order are major drivers of gene regulatory activity. *Nature Communications*, 14(1):2333. Publisher: Nature Publishing Group.

Getoor, L. and Diehl, C. P. (2005). Link mining: a survey. *SIGKDD Explor. Newsl.*, 7(2):3–12.

Ghasemian, A., Hosseinmardi, H., Galstyan, A., Airoldi, E. M., and Clauset, A. (2020). Stacking models for nearly optimal link prediction in complex networks. *Proceedings of the National Academy of Sciences*, 117(38):23393–23400. Publisher: Proceedings of the National Academy of Sciences.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hao, Y., Cao, X., Fang, Y., Xie, X., and Wang, S. (2020). Inductive Link Prediction for Nodes Having Only Attribute Information. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 1209–1215. arXiv:2007.08053 [cs].

Hecker, M., Lambeck, S., Toepfer, S., van Someren, E., and Guthke, R. (2009). Gene regulatory network inference: Data integration in dynamic models—A review. *Biosystems*, 96(1):86–103.

Hollingsworth, R. and White, J. H. (2004). Target discovery using the yeast two-hybrid system. *Drug Discovery Today: TARGETS*, 3(3):97–103.

Huang, J., Du, Y., Quan, G., and Zhu, D. (2009). A Protein-Phenotype Mutual Information Based Identification of Human Disease Genes. In *2009 International Conference on Computational Intelligence and Software Engineering*, pages 1–4.

Huynh-Thu, V. A., Irrthum, A., Wehenkel, L., and Geurts, P. (2010). Inferring Regulatory Networks from Expression Data Using Tree-Based Methods. *PLOS ONE*, 5(9):e12776. Publisher: Public Library of Science.

Ieremie, I., Ewing, R. M., and Niranjan, M. (2022). TransformerGO: predicting protein–protein interactions by modelling the attention between sets of gene ontology terms. *Bioinformatics*, 38(8):2269–2277.

Jin, J., He, K., Tang, X., Li, Z., Lv, L., Zhao, Y., Luo, J., and Gao, G. (2015). An Arabidopsis Transcriptional Regulatory Map Reveals Distinct Functional and Evolutionary Features of Novel Transcription Factors. *Molecular Biology and Evolution*, 32(7):1767–1773.

Jin, J., Tian, F., Yang, D.-C., Meng, Y.-Q., Kong, L., Luo, J., and Gao, G. (2017). PlantTFDB 4.0: toward a central hub for transcription factors and regulatory interactions in plants. *Nucleic Acids Research*, 45(D1):D1040–D1045.

Lai, R., Chen, R., Han, Q., Zhang, C., and Chen, L. (2024). Adaptive Hardness Negative Sampling for Collaborative Filtering. arXiv:2401.05191 [cs].

Lamesch, P., Berardini, T. Z., Li, D., Swarbreck, D., Wilks, C., Sasidharan, R., Muller, R., Dreher, K., Alexander, D. L., Garcia-Hernandez, M., Karthikeyan, A. S., Lee, C. H., Nelson, W. D., Ploetz, L., Singh, S., Wensel, A., and Huala, E. (2012). The Arabidopsis Information Resource (TAIR): improved gene annotation and new tools. *Nucleic Acids Research*, 40(D1):D1202–D1210.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Li, G. and Reinberg, D. (2011). Chromatin higher-order structures and gene regulation. *Current Opinion in Genetics & Development*, 21(2):175–186.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605.

Margolin, A. A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R. D., and Califano, A. (2006). ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. *BMC Bioinformatics*, 7(1):S7.

Mercatelli, D., Scalambra, L., Triboli, L., Ray, F., and Giorgi, F. M. (2020). Gene regulatory network inference resources: A practical overview. *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms*, 1863(6):194430.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781 [cs].

Nakato, R. and Sakata, T. (2021). Methods for ChIP-seq analysis: A practical workflow and advanced applications. *Methods*, 187:44–53.

Patel, N. and Wang, J. T. L. (2015). Semi-supervised prediction of gene regulatory networks using machine learning algorithms. *Journal of Biosciences*, 40(4):731–740.

Pratapa, A., Jalihal, A. P., Law, J. N., Bharadwaj, A., and Murali, T. M. (2020). Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature methods*, 17(2):147–154.

Robinson, J., Chuang, C.-Y., Sra, S., and Jegelka, S. (2021). Contrastive Learning with Hard Negative Samples. arXiv:2010.04592 [cs].

Turki, T. and Wang, J. T. L. (2015). A New Approach to Link Prediction in Gene Regulatory Networks. volume 9375, pages 404–415, Cham. Springer International Publishing. Book Title: Intelligent Data Engineering and Automated Learning – IDEAL 2015 Series Title: Lecture Notes in Computer Science.

Wang, Y., Chen, X., Zheng, Z., Huang, L., Xie, W., Wang, F., Zhang, Z., and Wong, K.-C. (2024). scGREAT: Transformer-based deep-language model for gene regulatory network inference from single-cell transcriptomics. *iScience*, 27(4):109352.

Xia, F., Sun, K., Yu, S., Aziz, A., Wan, L., Pan, S., and Liu, H. (2021). Graph Learning: A Survey. *IEEE Transactions on Artificial Intelligence*, 2(2):109–127. Conference Name: IEEE Transactions on Artificial Intelligence.

Yu, Y., Zhang, H., Long, Y., Shu, Y., and Zhai, J. (2022). Plant Public RNA-seq Database: a comprehensive online database for expression analysis of ˜45000 plant public RNA-Seq libraries. *Plant biotechnology journal*, 20(5):806–808. Place: England.

Zhang, H., Zhang, F., Yu, Y., Feng, L., Jia, J., Liu, B., Li, B., Guo, H., and Zhai, J. (2020a). A Comprehensive Online Database for Exploring 20,000 Public Arabidopsis RNA-Seq Libraries. *Molecular Plant*, 13(9):1231–1233.

Zhang, J. and Luo, Y. (2017). Degree Centrality, Betweenness Centrality, and Closeness Centrality in Social Network. pages 300–303. Atlantis Press. ISSN: 1951-6851.

Zhang, S., Li, X., Lin, Q., Lin, J., and Wong, K.-C. (2020b). Uncovering the key dimensions of high-throughput biomolecular data using deep learning. *Nucleic Acids Research*, 48(10):e56.