Analysis of Gene Interaction Graphs for Biasing Machine Learning Models

Paul Bertin 1 2 Mohammad Hashir 1 2 Martin Weiss 1 2 Geneviève Boucher 2 3 Vincent Frappier 1 2 Joseph Paul Cohen 1 2

Abstract

Gene interaction graphs aim to capture various relationships between genes and can be used to create more biologically-intuitive models for machine learning. There are many such graphs available which can differ in the number of genes and edges covered. In this work, we attempt to evaluate the biases provided by those graphs through utilizing them for ‘Single Gene Inference’ (SGI) which serves as, what we believe is, a proxy for more relevant prediction tasks. The SGI task assesses how well a gene’s neighbors in a particular graph can ‘explain’ the gene itself in comparison to the baseline of using all the genes in the dataset. We evaluate seven major gene interaction graphs created by different research groups on two distinct datasets, TCGA and GTEx. We find that some graphs perform on par with the unbiased baseline for most genes with a significantly smaller feature set.

1. Introduction

A major challenge in using machine learning on gene expression data is overcoming the curse of dimensionality. The number of examples in most datasets, being typically much smaller than the number of genes, leads to the issue of spurious correlations. Using features from all the genes can be problematic as there are multiple ways that genes can be associated with each other to form groups of interacting genes. Given the limited number of examples, the model would tend to learn the noise and spurious correlations in place of any biologically relevant patterns. Thus, one needs a systematic approach that incorporates biological knowledge to select or weight meaningful features from genetic data and ensures the validity of the model.

Many groups have developed a number of gene-interaction graphs, structuring domain knowledge from different areas of molecular biology. Gene interaction graphs can be used with machine learning algorithms as a proxy for biological intuition to leverage decades of biology research (Zhang et al., 2017). These graphs can act as a biological prior on machine learning techniques to automate feature importance and selection. For example, network-based linear regression (Li & Li, 2008; Min et al., 2016) regularizes the weights of a linear model based on the connectivity of the nodes found in an interaction graph. Preliminary work by (Dutil et al., 2018) found that the same can be done for non-linear models and remarked that the quality of these graphs may impact their potential in developing general models which would be useful in the majority of tasks where gene expression or single-nucleotide polymorphism data is the input. Of course, these graphs were not developed as an input for ML applications, so it would not be surprising that they are not optimal to aid ML algorithms.

In this work, we propose a method to quantitatively evaluate the feature selections provided by gene-interaction graphs. Following a Single Gene Inference (SGI) evaluation approach (Dutil et al., 2018; Chen et al., 2016; Subramanian et al., 2017), we construct a task that compares the performance of a non-linear model (a multilayer perceptron) using only first degree gene neighbors against a model that uses the full gene set. With these experiments, we aim to measure the value of the bias provided to the ML model by the several feature selections by assessing how a gene’s neighbors in a graph capture the signal necessary to predict the expression level of that gene and we believe that this task could be a proxy for more clinically applicable tasks. Note that we do not claim to assess the intrinsic value of the different graphs beyond this specific task.

While regularization and feature selection are common tools for dealing with overfitting on high dimensional data, we decided to focus on feature selection. Compared to regularization, feature selection is a way to bias the model in a hard way which could potentially achieve poorer performance. But feature selection provides the model with greater interpretability, which is of primary importance when deal-
ing with genomic data as genomics is a domain where we have relatively limited intuition compared to images or text. The interpretability of the model could provide a ‘research gradient’ to biologists allowing them to focus on specific subgroups of genes, which could lead to a fruitful feedback loop between biological experiments and machine learning predictions. Interpreting those models could also help in generating new hypotheses that may be validated with experiments, thus helping biologists.

2. Materials and methods

The Single Gene Inference (SGI) task was first formulated in Dutil et al. (2018), which was inspired by (Chen et al., 2016) and (Subramanian et al., 2017). It involves predicting the expression level of a gene using the expressions of other genes available. This task allows us to evaluate different feature selections provided by different interaction graphs. More precisely, we use a gene-interaction graph to perform feature selection for the SGI task and then evaluate the performance with this selection. The gene graph serves as a form of a priori biological knowledge for the task. By using the expressions of only the genes connected directly to the target gene in the graph to infer it, the target gene’s first-degree neighborhood gets employed as a ‘biology-biased feature selection’. The SGI task is only a proxy for more relevant tasks such as the prediction of clinical attributes and to our knowledge it hasn’t demonstrated to be inherently useful for biologists.

The goal of these experiments is to assess how efficiently the first-degree neighborhoods in the graph capture the signal and concurrently eliminate the noise. The graph-biased feature selection inherently evaluates the quality of the graph in the context of SGI such that the graph contains all of the edges that are needed to explain a given node while still avoiding spurious correlations. We made several simplifications in order to be able to use all graphs in a similar manner, for example we did not take into account weighted edges but considered them as present or absent. We do not claim to evaluate the intrinsic value of the different graphs, but rather the usefulness of their set of edges to provide a good feature selection for SGI (and hopefully more biologically relevant tasks). Moreover, we hypothesize that second order (or higher) relationships in the graphs wouldn’t be relevant for prediction, that is to say the conditional probability of the target gene’s expression given all other genes would depend only on its first degree neighbors.

We are interested in graphs that contain a good proportion of those genes as nodes. As the final purpose is to use the graph to induce a bias in ML models, it is important to have a graph with sufficient coverage of all genes present in a dataset as opposed to a small subset of well-studied genes. To give an idea, the typical order of magnitude of the number of genomic features in the datasets we use is 20k. The different graphs can have different number of nodes which can impact their performance.

We restrict ourselves to a binary classification task to simplify interpretation of the results. We convert the target gene to a categorical variable based on whether it is over or under the mean expression for that gene, representing “overexpressed” or “underexpressed” respectively. Then, we fit the model for inferring the target gene by extracting the neighbors of the target gene in the graph and using their expression values as the input. Subsequently, the AUC is computed on a hold-out test set. If the target gene has no neighbors in the graph, an AUC of 0.5 is assigned without fitting any model as the absence of any neighbors indicates the absence of any input features making the prediction a random guess. We perform these experiments for several graphs and datasets, and all the genes in each dataset.

Model configuration Based on how Dutil et al. (2018) demonstrated the existence of non-linear signal in genetic data, we utilize a multilayer perceptron (MLP) in our experiments. The MLP has a single hidden layer with a ReLU activation and 16 neurons. The binary cross-entropy loss was used with an Adam optimizer and a learning rate of 0.001 for the FunCoup, Hетионет, and fully-connected graphs and 0.0007 for the rest on all datasets. The weight decay parameter was set to $1e^{-5}$. These hyperparameters were obtained with a hyperparameter search over the different graphs and datasets. We fit the model for 3 trials for each combination of a gene, graph and dataset to have a robust evaluation.

Datasets We use the TCGA (Weinstein et al., 2013) and the GTEx (Lonsdale et al., 2013) datasets, both of which are public and well-studied. The TCGA PANCAN database spans multiple tissues and measures 20,530 gene expressions for 10,459 samples; most samples come from cancer biopsies but many healthy examples are also included. The GTEx dataset consists of samples from only healthy subjects and has a higher amount of genomic features (34,218 genes) but only for 2,921 samples. We normalize both datasets by the mean (gene-wise) for our analysis. To ensure small overlapping of example sets between trials, we used 3000 samples for TCGA and 1500 samples for GTEx with equal splits between the training, testing and validation sets. The data was randomly sampled for every trial but remained the same for every gene regardless of graph.

Graphs We evaluate six graphs covering a variety of relationships in the genome, namely GeneMania (Warde-Farley et al., 2010), RegNetwork (Liu et al., 2015), Hетионет (Himmelstein et al., 2017), FunCoup (Ogris et al., 2018), HumanNet (Hwang et al., 2019) and StringDB (Szklarczyk et al., 2019). For Hетионет, we combine the Interaction, Co-
Analysis of Gene Interaction Graphs for Biasing ML Models

variation and Regulation sub-graphs to create an undirected composite graph. StringDB contains different sub-graphs depending on the interaction type; we use the co-expression graph and the entire graph in our analysis. We also generate a separate graph based on the Landmark genes (Subramaniam et al., 2017). Those 978 genes were chosen to optimally recover the observed connections seen in the pilot Connectivity Map dataset. We build a graph in which each gene in a given dataset is connected to the set of the 978 landmark genes which are themselves connected together, forming a clique. We refer to this graph as the Landmark graph. Finally, we compute the unbiased baseline which we call the ‘fully-connected graph’, as it can be represented as a graph where each gene has all the other genes as first-degree neighbors. As the different graphs have different distributions of degrees, we define a metric called connectedness of a graph to have a clearer picture. It is the ratio of the number of edges in a graph to the number of edges in the fully-connected graph, which is dataset-dependent.

3. Results

3.1. Performance of the biased models

The distribution of the AUCs, averaged across trials, for all the graphs is visualized in Figure 1. For both datasets, there is a considerable spike at 0.5 which is mostly due to the fact that we assign a 0.5 AUC to the genes not present in the graph and some graphs have a very small proportion of genes in the datasets, especially for the GTEx dataset. However, some part of the peak at 0.5 AUC does correspond to genes present in the graph and for which the trained model actually achieved an AUC of 0.5, which could be due to either bad convergence or too restrictive feature selection. We report the mean AUCs along with other statistics in Table 1. We chose to compute mean AUCs over both the whole dataset and the set of covered genes (which differs between graphs). To assess the robustness of the evaluation, we report the per-gene standard deviation of AUC across the three trials in Table 1.

3.2. Comparison with the fully connected graph

We would like to assess whether the graph-biased models perform better than the baseline. For each gene in a graph, we subtract the AUC achieved with its first degree neighborhood from the AUC achieved with all genes (baseline). Each difference is an average over three trials. The distribution of those differences is summarized in Figure 1. A lot of graphs achieve poorer performance than the baseline. On the other hand, some graphs, namely StringDB and Landmark, achieve reasonable performance. Note that these two graphs have the highest coverage and connectedness. Their corresponding distributions have a significant amount of mass in the positive region, meaning these models perform
We studied several existing gene interaction graphs to assess how well they can be used to provide feature selection for the SGI task. We found that large graphs such as StringDB and Landmark perform on par with the fully-connected graph most of the time, while having significantly smaller number of edges. This suggests that using the entire gene set better than the baseline a significant portion of the time, even if the margin is small. However, those distributions still have a long tail in the negative, meaning that using StringDB or Landmark hurts performance a lot for some genes. Those failure cases might be the consequence of missing edges in the graphs. To confirm this hypothesis, we plot the distribution of per-gene AUC differences with respect to the number of neighbors of genes in Figure 3. Highly negative AUC differences seem to correspond to genes with a relatively small number of neighbors.

**Figure 3.** Covered Genes AUC improvement with respect to the number of neighbors, for StringDB (all) and TCGA. Color indicates density. Averaged over 3 trials.

### Table 1. Statistics for each graph and dataset. **Coverage** is the percentage of genes in the dataset that are represented as nodes in the graph. **Connectedness** represents the percentage of edges in the graph compared to the fully connected graph. The **Covered Genes AUC** and **All Genes AUC** refer to the average AUC achieved by the graph, respectively, on only its covered genes and on the entire dataset (after adding uncovered genes with an AUC of 0.5). The **Improvement** is computed with respect to the fully-connected baseline for both the covered genes and all genes. **Per-gene AUC STD** refers to the per-gene standard deviation of AUCs across the three trials (averaged over all genes for a given graph and dataset). All uncertainties are computed as the standard deviation across the three trials.

|                      | Fully Connected | Genemania  | RegNet | HNetV2 | Hetio | FunCoup | StringDB (all) | Landmark |
|----------------------|----------------|------------|--------|--------|-------|---------|----------------|----------|
| **Coverage (%)**     | 100            | 79         | 35     | 87     | 86    | 82      | 92             | 100      |
| **Connectedness (%)**| 100            | 0.06       | 0.06   | 0.13   | 0.11  | 1.34    | 1.39           | 4.82     |
| **Per-gene AUC STD** | 0.017          | 0.010      | 0.005  | 0.011  | 0.013 | 0.011   | 0.011          | 0.013    |
| **All Genes AUC (+2e-4)** | 0.782 | 0.636    | 0.590  | 0.699  | 0.640 | 0.702   | 0.717          | 0.788    |
| **All Genes improvement (+2e-4)** | -0.146 | -0.193 | -0.083 | -0.143 | -0.081 | -0.011 | 0.006         | 0.006    |
| **Covered Genes AUC (+2e-4)** | 0.782 | 0.673 | 0.753 | 0.730 | 0.663 | 0.749 | 0.802 | 0.788 |
| **Covered Genes improvement (+2e-4)** | -0.126 | -0.056 | -0.062 | -0.130 | -0.045 | **0.013** | **0.006** |

**TCGA**

- **Coverage (%)**: 100, 48, 21, 52, 52, 49, 55, 100
- **Connectedness (%)**: 100, 0.02, 0.02, 0.04, 0.04, 0.47, 0.49, 2.86
- **Per-gene AUC STD**: 0.036, 0.012, 0.005, 0.012, 0.015, 0.013, 0.011, 0.012
- **All Genes AUC (+2e-4)**: 0.734, 0.599, 0.564, 0.647, 0.599, 0.630, 0.685, 0.748
- **All Genes improvement (+2e-4)**: -0.222, -0.170, -0.212, -0.135, -0.105, -0.050, 0.014
- **Covered Genes AUC (+2e-4)**: 0.734, 0.707, 0.802, 0.781, 0.693, 0.764, 0.837, 0.748
- **Covered Genes improvement (+2e-4)**: -0.116, -0.027, -0.037, -0.126, -0.054, **0.021**, **0.014**

**GTEx**

- **Coverage (%)**: 100, 79, 35, 87, 86, 82, 92, 100
- **Connectedness (%)**: 100, 0.06, 0.06, 0.13, 0.11, 1.34, 1.39, 4.82
- **Per-gene AUC STD**: 0.017, 0.010, 0.005, 0.011, 0.013, 0.011, 0.012, 0.013
- **All Genes AUC (+2e-4)**: 0.782, 0.636, 0.590, 0.699, 0.640, 0.702, 0.771, 0.788
- **All Genes improvement (+2e-4)**: -0.146, -0.193, -0.083, -0.143, -0.081, -0.011, **0.086**, **0.006**
- **Covered Genes AUC (+2e-4)**: 0.782, 0.673, 0.753, 0.730, 0.663, 0.749, **0.802**, **0.788**
- **Covered Genes improvement (+2e-4)**: -0.126, -0.056, -0.062, -0.130, -0.045, **0.013**, **0.006**

4. Conclusion

We studied several existing gene interaction graphs to assess how well they can be used to provide feature selection for the SGI task. We found that large graphs such as StringDB and Landmark perform on par with the fully-connected graph most of the time, while having significantly smaller number of edges. This suggests that using the entire gene set to predict the expression of a gene tends to include mostly uninformative features as the expression of most genes can be explained with a fraction of the gene set. However, using those graphs may result in a poor performance compared to the baseline for some genes (as observed for genes with relatively small number of neighbors in Figure 3) which suggests that the biologically relevant information for inferring the gene was not present in the graph as a first-degree relationship. Thus, the additive value of using those graphs as is to provide feature selection for the SGI task appears to be limited. Efficiently taking advantage of those interaction-graphs in other tasks while accounting for their incomplete coverage remains an interesting direction of research.

We chose to perform our evaluation on the complete dataset as opposed to the set of genes graphs cover. This is because our goal is to find the most general graph that can be used for a variety of ML tasks. Our analysis might not clearly represent the intrinsic value of the biological knowledge these graphs contain. For instance, one could try to compare the feature selections provided by the different graphs with random feature selections having similar statistics. Those analyses are left for future work.

**Acknowledgements**

We thank Francis Dutil and Mandana Samiei for their useful code and comments. This work is partially funded by a grant from the Fonds de Recherche en Sante du Quebec and the Institut de valorisation des donnees (IV ADO). This work utilized the supercomputing facilities managed by Mila, NSERC, Compute Canada, and Calcul Quebec. We also thank NVIDIA for donating a DGX-1 computer used in this work.
Analysis of Gene Interaction Graphs for Biasing ML Models

References

Chen, Yifei, Li, Yi, Narayan, Rajiv, Subramanian, Aravind, and Xie, Xiaohui. Gene expression inference with deep learning. Bioinformatics, jun 2016. doi: 10.1093/bioinformatics/btw074.

Dutil, Francis, Cohen, Joseph Paul, Weiss, Martin, Derevyanko, George, and Bengio, Yoshua. Towards Gene Expression Convolutions using Gene Interaction Graphs. In International Conference on Machine Learning (ICML) Workshop on Computational Biology (WCB), 2018.

Himmelstein, Daniel Scott, Lizee, Antoine, Hessler, Christine, Brueggerman, Leo, Chen, Sabrina L, Hadley, Dexter, Green, Ari, Khankhanian, Pouya, and Baranzini, Sergio E. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. eLife, September 2017. doi: 10.7554/eLife.26726.

Hwang, Sohyun, Kim, Chan Yeong, Yang, Sunmo, Kim, Eiru, Hart, Traver, Marcotte, Edward M, and Lee, Insuk. HumanNet v2: human gene networks for disease research. Nucleic acids research, January 2019. doi: 10.1093/nar/gky1126.

Li, Caiyan and Li, Hongzhe. Network-constrained regularization and variable selection for analysis of genomic data. Bioinformatics, 2008. doi: 10.1093/bioinformatics/btn081.

Liu, Zhi-Ping, Wu, Canglin, Miao, Hongyu, and Wu, Hulin. RegNetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. Database: The Journal of Biological Databases and Curation, 2015. doi: 10.1093/database/bav095.

Lonsdale, John, Thomas, Jeffrey, Salvatore, Mike, Phillips, Rebecca, Lo, Edmund, Shad, Saboor, Hasz, Richard, Walters, Gary, Garcia, Fernando, Young, Nancy, and Others. The genotype-tissue expression (GTEx) project. Nature genetics, 2013.

Min, Wenwen, Liu, Juan, and Zhang, Shihua. Network-regularized Sparse Logistic Regression Models for Clinical Risk Prediction and Biomarker Discovery. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2016. doi: 10.1109/TCBB.2016.2640303.

Ogris, Christoph, Guala, Dimitri, Kaduk, Mateusz, and Sonnhammer, Erik L L. FunCoup 4: new species, data, and visualization. Nucleic Acids Research, jan 2018. doi: 10.1093/nar/gkx1138.

Subramanian, Aravind et al. A Next Generation Connectivity Map: L1000 Platform and the First 1,000,000 Profiles. Cell, nov 2017. doi: 10.1016/j.cell.2017.10.049.

Szklarczyk, Damian, Gable, Annika L, Lyon, David, Junge, Alexander, Wyder, Stefan, Huerta-Cepas, Jaime, Simonovic, Milan, Doncheva, Nadezhda T, Morris, John H, Bork, Peer, Jensen, Lars J, and Mering, Christian von. STRING v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. Nucleic Acids Research, jan 2019. doi: 10.1093/nar/gky1131.

Warde-Farley, David, Donaldson, Sylva L., Comes, Ovi, Zuberi, Khalid, Badrawi, Rashad, Chao, Pauline, Franz, Max, Grouios, Chris, Kazi, Farzana, Lopes, Christian Tannus, Maitland, Anson, Mostafavi, Sara, Montojo, Jason, Shao, Quentin, Wright, George, Bader, Gary D., and Morris, Quaid. The GeneMANIA prediction server: biological network integration for gene prioritization and predicting gene function. Nucleic Acids Research, 2010. doi: 10.1093/nar/gkq537.

Weinstein, John N, Collisson, Eric A, Mills, Gordon B, Shaw, Kenna R Mills, Ozenberger, Brad A, Ellrott, Kyle, Shmulevich, Ilya, Sander, Chris, and Stuart, Joshua M. The Cancer Genome Atlas Pan-Cancer analysis project. Nature genetics, 2013. doi: 10.1038/ng.2764.

Zhang, Wei, Chien, Jeremy, Yong, Jeongsik, and Kuang, Rui. Network-based machine learning and graph theory algorithms for precision oncology. npj Precision Oncology, dec 2017. doi: 10.1038/s41698-017-0029-7.