Deep Learning Benchmarks on Gene Expression Data

Matthew B.A. McDermott
Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology
mmd@mit.edu

Jennifer Wang
Center for Quantitative Health, Massachusetts General Hospital
jennifer.wang@mgh.harvard.edu

Wen Ning Zhao
Chemical Neurobiology Laboratory, Massachusetts General Hospital
wnzhao@mgh.harvard.edu

Steven D. Sheridan
Center for Quantitative Health, Massachusetts General Hospital
ssheridan2@partners.org

Peter Szolovits
Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology
psz@mit.edu

Isaac Kohane
Department of Biomedical Informatics, Harvard University
isaac_kohane@harvard.edu

Stephen J. Haggarty
Chemical Neurobiology Laboratory, Massachusetts General Hospital
shaggarty@mgh.harvard.edu

Roy H. Perlis
Center for Quantitative Health, Massachusetts General Hospital
rperlis@mgh.harvard.edu

ABSTRACT

Gene expression data holds the potential to offer deep, physiological insights about the state of a cell beyond the static coding of the genome alone. We believe that realizing this potential requires specialized machine learning methods capable of using underlying biological structure, but the development of such models is hampered by the lack of published benchmark tasks and well characterized baselines.

In this work, we establish such benchmarks and baselines by profiling a battery of classifiers against biologically motivated classification tasks on two curated views of a large, public gene expression dataset (the LINCS corpus) and one privately produced gene expression dataset. We provide these two curated views of the public LINCS dataset and biologically motivated benchmark tasks to enable direct comparisons to future methodological work and help spur iterative deep learning method development on this important data modality.

In addition to profiling a battery of traditional classifiers, including linear models, random forests, decision trees, K nearest neighbor (KNN) classifiers, and feed-forward artificial neural networks (FF-ANNs), we also test a method novel to this data modality: Graph Convolutional Neural Networks (GCNNs), which allow us to incorporate prior biological domain knowledge.

We find that GCNNs can be highly performant, but require large amounts of data, whereas FF-ANNs consistently perform well. Non-neural classifiers across all tasks and datasets are dominated by linear models and KNN classifiers.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; *Supervised learning*; *Neural networks*; • **Applied computing** → **Computational transcriptomics**; **Transcriptomics**; *Biological networks*;

BioKDD'18, August 2018, London, UK
2018.

1 INTRODUCTION

Gene expression data offers a view beyond the static genome into the dynamic workings of the cell. The potential utility of this data modality is staggering—spanning drug development, personalized medicine, and biomedical research. Additionally, biologists have accrued a mass of domain knowledge regarding how gene expression is regulated, providing extensive, if complicated and uncertain, structure around these data. Further, the availability of large-scale, heterogeneous gene expression datasets is rapidly on the rise, fueled both by falling costs and development of new gene expression profiling technologies.

Despite these advances, machine learning methods for these data are relatively simple and unstructured. Researchers have explored variations of feed-forward neural networks for specialized purposes, but the lack of a rigid empirical methodological foundation, including published benchmarks, slows novel method development.

In this work, we lay that foundation: we examine several biologically relevant supervised classification tasks on datasets ranging in scale and heterogeneity, including two curated views[1] of the public L1000 LINCS dataset and one privately produced gene expression dataset. On each task, we profile K nearest neighbor (KNN) classifiers, decision trees, random forests (RFs), linear classifiers, and two neural classifiers: feed-forward artificial neural networks (FF-ANNs) and graph convolutional neural networks (GCNNs). GCNNs generalize the notion of convolutional neural networks (CNNs) onto data structured over arbitrary graphs and allow us to use prior biological knowledge, namely regulatory relationships between pairs of gene, to more intelligently model these data. To the best of our knowledge, this is the first work that uses these techniques to classify gene expression profiles.

We find that GCNNs can be performant, but require large amounts of data, excelling at all tasks on our largest dataset, but underperforming FF-ANNs on our smaller datasets. Of other methods, FF-ANNs perform best, followed consistently by linear classifiers,

[1] See https://github.com/mmcdermott/LINCS_Deep_Learning_Benchmarks

then random forests, then decision trees. KNN classifiers perform very well on our larger datasets, nearly matching FF-ANNs, but are computationally intensive and underwhelm on our smaller datasets.

Gene expression datasets often contain many samples spanning a much smaller set of patients, as a single patient's gene expression profile may be taken many times under varying conditions (e.g., drugs, etc.). As such, an important, distinct question from traditional performance measures (e.g., per-sample accuracy) is per-patient accuracy (i.e., generalization to unseen patients). We assess this on our private corpus and find that all methods struggle to generalize to unseen patients, showing performance drops ranging from 10 to 18 percent of their per-sample accuracies.

In this work we make the following contributions:

(1) We establish biologically meaningful classification benchmarks on the largest publicly available gene expression dataset. This is important because absent a shared, consistent view of the data and task definitions enabling new methods to be easily compared against the state of the art, deep learning method development is severely hampered.
(2) We profile a number of classifiers on these tasks, including non-neural methods and two variants of neural networks, one of which incorporates prior biological knowledge and, to the best of our knowledge, has never been profiled on this data modality.
(3) We profile these same classifiers on a similar task on a smaller, privately produced gene expression corpus to assess which techniques work well in data-starved environments.
(4) We assess how well these techniques transfer to unseen patients to assess population-level generalizability.

2 BACKGROUND & RELATED WORK

2.1 Gene Expression Data

2.1.1 The Biology. The cellular system is governed by the genome: the sequence of DNA base pairs that encode all information necessary for the cell's development and functioning. In order to process DNA into useful cellular work, the cell first *transcribes* genes into messenger RNA (mRNA), which is then shuttled towards cellular organelles that *translate* mRNA sequences into proteins: amino-acid built macromolecules that carry out all of the necessary functions of the cell. A cell's gene expression profile quantifies how actively these genes are being *expressed* (i.e., transcribed and translated into proteins) and thus provides a view into the dynamic state of the cell beyond the fixed picture offered by the genome alone.

A single cell's gene expression patterns will vary over time and in response to environmental conditions, such as exposure to drugs. The expression of proteins within the DNA is mediated by a host of factors, including other proteins in the cellular environment and external factors, and is critical to cell function. Understanding the genetic regulatory network (i.e., which factors govern transcription of what and how) is a topic of intense study.

2.1.2 Measuring Gene Expression/Transcriptomics. Gene expression can be quantified in many ways. Two broad categories of gene expression data are *proteomics*, which directly measures the quantities of produced proteins within the cell, and *transcriptomics*, which measures the quantities of produced mRNA transcripts within the cell (Figure 1). Transcriptomic gene expression is far more easily measured and will be our focus in this work.

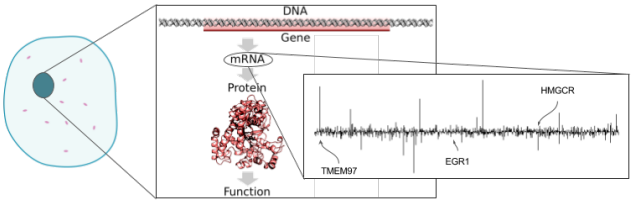

Figure 1: Transcriptomics data is measured by quantifying the mRNA produced during transcription. The output of this process is a vector with each dimension quantifying the expression of a particular gene. Both technical (e.g., misplaced reads) and biological (e.g., tissue type) factors add variance to these data. Images: [9, 26]

Note that there is not a direct correspondence between these two measurement techniques. Protein production is heavily regulated post-transcription, and in using transcriptomic data, we ignore these additional layers of biological processing in favor of the increased availability of data.

2.1.3 Measurement Techniques. Transcriptomics data itself can be measured by many techniques, including RNA-Seq, single-cell RNA-Seq, and the L1000 platform, which we focus on here. The L1000 platform [27] is notably cheaper per-sample than other transcriptomics techniques, which has enabled the creation of large scale public datasets, such as the LINCS dataset, which was produced with the L1000 platform and contains ~1.3M samples, available on GEO at accession number GSE92742.[2]

However, this low price point sacrifices some data quality and coverage. Rather than quantifying the full transcriptome, the L1000 platform only directly measures the expression levels of 978 "landmark genes" and requires several additional layers of processing which add their own sources of technical variability. From this directly measured subset, the L1000 technique also uses a linear model to impute the remaining genes' expression levels, but we ignore those inferred genes in our analyses and use only the landmark genes.

L1000 data is often used at one of two levels of pre-processing:

Level 4 (a.k.a. Roast). Level 4 data is fully normalized, plate-controlled, and z-scored, and presented at the level of one profile per sample.

Level 5 (a.k.a. Brew). Level 5 data takes the Level 4 data and aggregates samples under identical technical conditions into a single averaged view of that profile (see [27] for full details). This process reduces variance, but also dataset size. Typically datasets are reduced to roughly $\frac{1}{3}$ of their original size (L1000 experiments are often performed "in triplicate," with three identical experimental plates being prepared so that all samples are run under identical conditions at least three times). This variance reduction is useful for traditional bioinformatics, but it is not clear how helpful it should be for deep learning methods which generally prefer to automatically learn how to extract features from the most raw view of data

[2] https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE92742

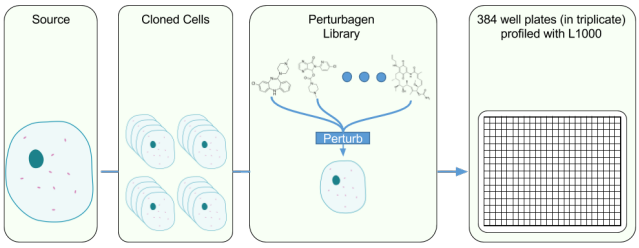

Figure 2: Gene expression corpora are often produced by cloning a small number of cellular sources, then perturbing and profiling those clones. Image: [9].

possible. We would like our classifiers to be able to fully account for the technical variability inherent between repeated measurements, but using Level 5 data would deprive us of that opportunity while costing a significant number of input samples. On the other hand, Level 5 data may be of higher quality.

2.1.4 Experimental Pipelines. In general, experimental pipelines producing large corpora of gene expression data work by acquiring some base cellular sample, either patient derived or via a stock cell line, cloning that cell line extensively, then perturbing a number of samples and profiling them (Figure 2). In this way, these datasets often have many more samples than cellular sources. This can lead to population-specific over-fitting, where a model specializes only to the population within the corpus and, despite generalizing to unseen samples within the corpus, the model will fail to generalize to unseen cellular sources.

Experimental pipelines also often show highly skewed distributions of perturbagen frequency, as common perturbagens may be profiled across many cell sources but more niche perturbagens will be profiled in isolation only on some smaller subset of cell lines. This problem is even more extreme considering "control" substances, such as DMSO, which are often profiled many more times than other compounds to provide a rigid baseline. As a result, attempting to evaluate machine learning models across perturbagens can be difficult as one must account for these biases in the dataset.

2.2 Machine Learning on Gene Expression Data

2.2.1 Traditional Analyses. Traditional analyses on these data focus on statistical or geometric tests for differential gene expression [7], gene set enrichment analyses (GSEA) [28], and (for the L1000 platform specifically) signature based analyses [2, 27]. Some have also used tensor decomposition/completion to disentangle cell-type from perturbagen effects [16, 17], and explored traditional classifiers for adverse drug event prediction [29].

2.2.2 Neural Representation Learning. Other authors have used neural network models to build embeddings of gene expression data. In [11], the authors use a twin network architecture to represent gene expression profiles as 100 dimensional bar-codes. They actually use the inherently high technical variability of this modality as a learning signal, by training their network to learn an embedding that minimizes distances between replicated samples. In [5], the authors use a sparse autoencoder to analyze binarized yeast differential gene expression microarray data. Post-training analyses found overlaps between transcription-factor mediated regulatory relationships and the connections trained by their network between the first two layers. In [21], the authors explore neural network mediated dimensionality reduction for single cell RNA-Seq data, augmenting traditional networks by adding nodes to the first hidden layer according to known transcription factor or protein-protein interactions, and only connecting input gene nodes to those regulatory or interaction nodes as dictated by prior biological knowledge.

2.2.3 Neural Classification & Regression. In [1], authors use a FF-ANN to classify profiles into categories based on the therapeutic effect of the generating perturbagen. Researchers have also explored neural techniques for extrapolating the L1000 set of landmark genes to the full transcriptome. In [6] and [23], authors use a 3 hidden layer feed forward network to perform gene expression extrapolation from the L1000 landmark set.

2.3 Structured Models via Graph Convolutional Networks

2.3.1 Regulatory Graphs. As stated in Section 2.1.1, gene expression is regulated by complex processes and is a topic of intense study. What we do know of gene expression regulation is often envisioned as a graph (such as in Figure 3) with genes forming the vertices of this graph and edges between genes representing regulatory relationships between those two genes. We visualize one such regulatory graph in Figure 3.

Many of these relationships are only suspected, and as biologists have yet to study all possible interactions between sets of genes, these graphs are biased towards representing commonly studied proteins. Additionally, regulatory relationships themselves depend on cell type and, even within a single cell, they change in response to perturbations and environmental conditions, among other factors. Nonetheless, these "regulatory graphs" present at least a partial encoding of the biological understanding of relationships between different genes, and we use them here to augment neural classifiers with domain knowledge via GCNNs. Regulatory graphs are usually directed, but in this work we consider them as undirected graphs for simplicity.

2.3.2 Graph Convolutional Networks in Theory. GCNNs are extensions of CNNs onto data defined over arbitrary graphs. Qualitatively, we can think of these networks as attempting to analyze data whose features are nodes in a graph by repeatedly summarizing the features within local neighborhoods of the graph, before aggregating those features into higher level signals spanning larger regions of the graph. This is directly analogous to how convolutional neural networks for image processing learn featurizations of local patches of the image, then pool those signals over larger windows.

There are two main strategies to generalize a CNN to other domains: the spectral approach, which generalizes the notion of a Fourier transform onto a graph via the graph Laplacian, and the locality approach, which uses the idea of processing data defined in local patches via neighborhoods in the graph more directly. GCNNs must also generalize the notion of "pooling" onto graphs, which they generally do via graph clustering algorithms, using the resulting node clusters to determine pooling neighborhoods.

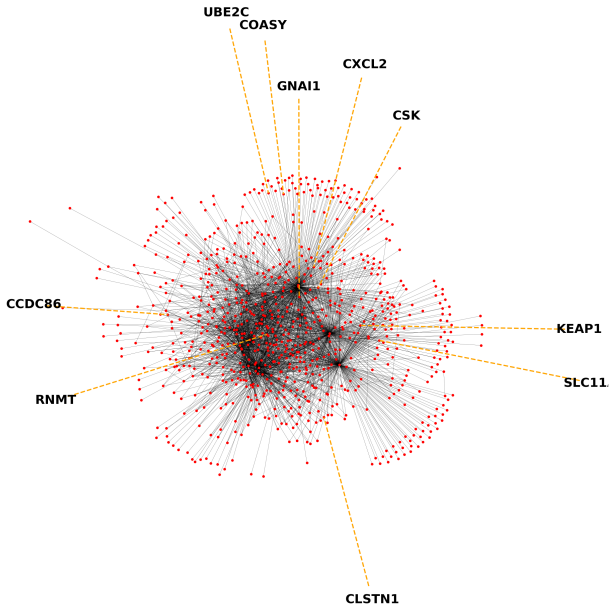

Figure 3: The regulatory relationships between L1000 landmark genes, as determined according to [22]. Nodes (red dots) are genes and edges between them represent known or suspected regulatory interactions. Note that many genes only have one known edge connecting them to much denser clusters within the center of the graph. This may reflect biological processes, or that some proteins are studied much more than others.

GCNNs promise to bring the normalization obtained via weight sharing over consecutive convolution and pooling operations to features defined over any arbitrary graph, but they present their own challenges. Both local and spectral methods present computational challenges, and efficient graph pooling algorithms run afoul of NP-hard graph clustering problems. In practice, many operations are approximated, which affects the power of these models.

2.3.3 Graph Convolutional Networks in Practice. Graph convolutional networks are often used in forming predictions at the node level, or in classifying whole graphs. For example, [18] explored node classification on knowledge and citation graphs. In this vein, GCNNs have also been used in several biological tasks. For example, [14] classifies proteins viewed as nodes in varying tissue-specific protein protein interaction graphs, [10] learns representations of molecular compounds interpreted as unique graphs with vertices determined by atoms and edges by bonds, [30] classifies polypharamacological interactions as edges of a drug and protein interaction graph, and [12] learns representations of graphs defined by protein three dimensional structure for protein interface prediction.

These node classification tasks differ from our context, where we wish to make predictions over a set of gene expression profiles, each of whose individual feature dimensions (the expression level of a particular gene) can be seen as a node on a static graph (a regulatory network such as Figure 3). Spectral methods are enticing for use in this context. In fact, this picture is so appealing that many papers describing novel GCNN algorithms use this example to frame the impact of their ideas [4, 8, 15, 20]. However, to the best of our knowledge, no work yet has profiled how these ideas actually serve on gene expression data in practice. We fill that lack here, and profile the work of [8], with minor technical modifications[3] to support multi-component graphs, on gene expression data backed by both tissue-independent and tissue-specific genetic regulatory networks culled from the literature.

3 METHODS

3.1 Datasets

3.1.1 Curated Views of the Public LINCS Corpus. The full Level 4 LINCS dataset contains ~1.3 M gene expression profiles over 76 cell lines, ranging in frequency from VCAP, profiled over 200,000 times to NCIH716 with only 43 samples. Each cell line is profiled in diverse conditions—for example, within prostate tissue (the most frequently sampled tissue type) over 40,000 unique perturbagens were tested (including both drugs and genetic knockout or over-expression perturbagens), many sampled only a single time. To be clear, each sample in this dataset is a complete gene expression profile over the landmark genes—i.e., it is a 978 dimensional vector where each number quantifies the expression level of a particular gene in the genome.

On this dataset, we formed three supervised learning tasks:

Primary Site Predicting primary site (e.g., "breast tissue" or "large-intestine") forces the classifier to examine deviations within a gene expression profile indicative of the tissue type, and would have applications to quality control within cell differentiation pipelines. Primary site is cell-line specific.

Subtype Subtype (e.g. "malignant melanoma" or "myoblast") is also cell-line specific and speaks to disease state and provides another way of aggregating the many disparate cell lines within LINCS into useful predictive categories.

MOA Predicting drug mechanism of action (MOA, e.g. "ATPase inhibitor" or "Sodium channel blocker") speaks to drug repurposing and discovery applications and aggregates many disparate perturbagens into meaningful predictive categories. However, note that though we treat this as a standard multi-class classification problem, in reality many drugs have multiple known MOAs, a distinction we ignore here for simplicity. To ensure this simplifying assumption adds minimal noise to our classification task, we only exclusively include compounds with only a single known MOA.

We chose to reduce the LINCS dataset to a single curated view simultaneously suitable for all three of these tasks rather than forming a separate view per task. This causes us to lose some samples which only meet inclusion criteria for a subset of our tasks, but it is much more convenient to work with and disseminate. In that pursuit, we reduced the dataset to only those samples perturbed by compounds (not genetic knock-out or over-expression perturbations), and further only those samples perturbed by compounds with a single known MOA. We further restricted the dataset to

[3] Our version of this code-base is available at https://github.com/mmcdermott/cnn_graph

only those samples corresponding to MOAs, primary sites, and subtypes that occurred more than 1000 times within the overall dataset, to ensure sufficient training examples for all classes for our classifiers. We performed these filtering steps independently—i.e., we removed all gene expression profiles belonging to a class in any of our three tasks that 1000 full examples at the start. This resulted in some few classes in some of our tasks having fewer than 1000 examples (because, at the beginning of the process, they had over 1000 measurements, but after removing some samples due to their class membership for another task, the class then had fewer than 1000 measurements). At these scales, we do not think that the lost samples significantly affect learning.

This formed one curated view of our data, and three classification tasks. One qualm some might have with this dataset is that it is very heterogeneous in terms of cell type—perhaps it is better to classify samples only derived from a single tissue type. To that end, we also formed a dataset containing only samples from prostate tissue (chosen as it was the most frequently sampled tissue type). As in our full dataset, here we restrict the samples to only those perturbed by compounds with a single known MOA that occurred at least 1000 times. This formed our "Prostate Only" dataset, on which we predict MOA only.

Full final dataset sizes, heterogeneity (among cell type) statistics, task statistics (e.g., class imbalance, number of classes) are shown in Table 1. We have made both of these datasets (though derived from fully public data), along with the cross-validation folds used in all of our experiments, publicly available,[1] so that others can most easily compare novel methodologies against our benchmarks.

We do not claim that these benchmark tasks or views of the data are the best benchmarks available. But these *are* biologically meaningful benchmarks on an important data modality that currently has *none*. We hope that as future methods evolve to better suit this methodology, we can also derive better benchmark tasks. Note here that we do not mean to claim that no machine learning tasks have been used on this modality previously, but rather that no set of systematized, very large sample size tasks for methodology development currently exist.

Given the very large ratio of samples to cellular sources (e.g. 156k to 36) and the very large skew in perturbagen frequency (e.g. DMSO accounting for approximately 1/6th of all data), as well as the lack of independence between perturbagen and cell type, we measure all accuracies on these datasets as *per-sample* accuracy, not *per-patient*, *per-drug*, or even *per-experimental condition* (as different experimental conditions are repeated to varying degrees). This means that our results on these data should not be interpreted to speak to true generalization outside the LINCS covariate space, but rather should be viewed only in their capacity to enable rigorous methodological comparisons.

3.1.2 MGH NeuroBank Corpus. Our private corpus of L1000 data was measured on a collection of patient-derived neural progenitor cells, which were perturbed with one of 60 different small-molecule bioactives at varying doses. Some of these compounds are known to have consistent gene-expression signatures (e.g., HDAC inhibitors), whereas others have known clinical utility but a less well understood transcriptomic profile (e.g., clozapine), and still others were unknown on all counts.

These cells come from a population of five patients, two healthy control subjects, one patient with Bipolar Disorder, and two patients with Schizophrenia. All patients' cells were treated with the same compounds. On this data, we predict perturbagen identity. Note that each perturbagen was profiled at one of several doses, which we ignore here. We also use this dataset to profile how well classifiers do on Level 4 vs. Level 5 data and make a first attempt at assessing per-patient generalizability, by training a model on only four of the five patients, then testing on the data for the fifth patient.

Full details for this corpus are also found in Table 1.

3.2 Models

We compare a variety of standard classifiers, all implemented via `scikit-learn` [24] for maximal reproducibility and ease of use.

In the interest of space, we will not provide a primer on each of the standard methods mentioned below in this work, but instead make clear why they were chosen to benchmark for this task and indicate which `scikit-learn` class was used to implement them. For a description of GCNNs see Section 2.3.

Classifiers Tested.

Feed-forward artificial neural network (FF-ANN) classifiers FF-ANNs are a common, powerful, non-linear modelling technique, and were used in many of the prior works on gene expression data. However, partly due to their lack of any particular structure, they are relatively inefficient learners. Implemented via the `MLPClassifier` class.

Linear classifiers Linear classifiers, subsuming both logistic regression (LR) and support vector classifiers (SVCs), are extremely common across all domains, including traditional bioinformatics analyses, and are interpretable. Implemented via the `SGDClassifier` class.

Random forests Random forests are not as commonly used in traditional bioinformatics use cases, but are thought to often provide a compelling non-neural but still non-linear baseline. They are composed of many bagged random decision trees. Implemented via the `RandomForestClassifier` class.

K nearest neighbors classifiers KNN methods are not as commonly used for classification, but are commonly used in this domain for clustering analyses, and we hope that investigating their performance here can help inform further choices for those and other analyses in these domains. They also shed some light on appropriate distance metrics. Implemented via the `KNeighborsClassifier` class.

Decision trees Decision trees are low powered, but extremely interpretable. Additionally, decision trees offer an interpretability closer to a mechanistic view than, for example, a linear classifier—e.g., were one able to find a performant decision tree classifier for diagnostic tasks, it would define immediately rule based differentiation methods between different disease states and prompt follow-on studies to determine if such rules were indicative of causal relationships. Implemented via the `DecisionTreeClassifier` class.

We also tested Naïve Bayes classifiers, Gaussian Processes Classifiers, Quadratic Discriminant Analysis, Boosted methods via Adaboost, and Kernel Support Vector Classifiers, but these classifiers were removed from our experimental lineup for reasons varying

Dataset Statistics:

Dataset	Number of Samples	# Cell Lines	Most Frequent Cell Line	Least Frequent Cell Line
Full LINCS	156,461	36	MCF7 (26,546)	NCIH716 (8)
Prostate Only LINCS	25,565	2	PC3 (13,625)	VCAP (11,940)
MGH NeuroBank (Level 4)	5602	5	N/A (1133)	N/A (1109)
MGH NeuroBank (Level 5)	1894	5	N/A (380)	N/A (377)

Task Statistics:

Dataset	Task	# Classes	Most Frequent Class	Least Frequent Class
LINCS (Full)	Primary Site	12	Prostate (43,686)	Ovary (415)
	Subtype	14	Adenocarcinoma (53,245)	Embryonal Kidney (1384)
	MOA	49	DMSO (25,638)	IKK Inhibitor (828)
LINCS (Prostate Only)	MOA	9	DMSO (8833)	Serotonin Receptor Antagonist (1029)
MGH NeuroBank (Level 4)	Perturbagen	60	DMSO (383)	Ruboxistaurin (78)
MGH NeuroBank (Level 5)	Perturbagen	60	DMSO (130)	Ruboxistaurin (27)

Table 1: Population statistics for our various datasets and tasks. The MGH NeuroBank Corpus was profiled on both Level 4 and Level 5 data (Section 2.1.3). Though DMSO is oversampled in the MGH NeuroBank Corpus, other perturbagens were sampled approximately uniformly. As MGH NeuroBank data uses patient derived cells, it uses no "cell lines."

from poor performance, non-insightful new results, computational intensivity, or combinations therein. None matched the performance of any neural model in early experiments.

Lastly, we tested GCNNs—in particular, the spectral approach defined by [8]. We use their provided code with minor modifications to support multi-component graphs. We considered a number of potential regulatory graphs, both tissue specific and tissue independent. Our tissue-independent regulatory networks were obtained via published resource [22][4] and our tissue-dependent regulatory networks were obtained via the published work [13].[5] Interested readers should refer to the primary sources to determine the details of the graph constructions—for our purposes it suffices to note that they are constructed to capture known or suspected genetic regulatory relationships as in Figure 3.

We considered a neuron graph for the MGH NeuroBank tissue, and a prostate gland graph for the prostate only LINCS dataset. The tissue independent graph has edges determined from the literature, and is unweighted and undirected. In contrast, the tissue-specific graphs come with edge weights determined via an estimated confidence in the true existence of that edge, determined via a probabilistic model. When working with this weighted graph, we culled all edges with confidence below a cutoff weight, which was tuned with all other hyperparameters.

3.3 Hyperparameter Search & Technical Setup

Hyperparameters for all classifiers were determined by a random search [3] over all possible parameters and tasks, including over the number and sizes of hidden layers for FF-ANNs and number of graph convolution layers/filter sizes/pooling sizes, loss types, etc. One notable disparity in the hyperparameter space searched is that the Scikit Learn FF-ANNs do not support dropout (only L2 regularization, which was included in our search), whereas the GCNNs do. To compensate for this potential bias, we took the optimal FF-ANN models found via the hyperparameter search and re-implemented them in Keras, as identically as possible, then performed a miniature grid-search over dropout within these models. This procedure induced a mild performance gain, but not enough to upset the observed model ordering on any tasks where GCNNs performed the best. We also did not hyperparameter optimize over batch size for FF-ANNs, but we did optimize over learning rate, a heavily related parameter, and we also tested several smaller batch sizes with our final models to ensure that we were not biasing the results against this baseline.

For GCNNs, we notably did not hyperparameter search over the number of epochs, but used a fixed number of epochs for computational reasons. Additionally, GCNNs only supported a single optimizer, whereas FF-ANNs offered several options in terms of optimizer. The search process was, however, run over various considered graphs, as well as over the graph edge weight cutoff, which we used to cull irrelevant edges from our graphs.

A full list of all hyperparameters tested, the distributions used to back our random search, and the final, chosen hyperparameters are available with our provided code.[1]

This random search was performed over 10 fold cross validation on the full LINCS dataset, and 15 fold cross validation on the private L1000 dataset (as that dataset is smaller, it warrants additional folds to improve accuracy). In each case, one fold was held out for testing, one for hyperparameter optimization, and the remaining used for training. The hyperparameter search optimized for mean accuracy over all folds, though we also report macro-F1 in our test set results below, as some tasks present significant class imbalance. For all results, statistical significance was assessed using paired t-test across all folds, followed by Benjamini-Hochberg multiple tests FDR adjustment within experimental conditions.

As different classifiers required different amounts of computational time to run, we did not run all classifiers for the same number of samples—this induces a mild bias towards the fastest running classifiers, as they will have had the opportunity to test additional hyperparameter settings. We did, however, ensure that we measured at least 60 samples for the standard FF-ANN classifier and linear models to ensure that we did not conclude any model better

[4]Networks available for download here: http://www.regnetworkweb.org/download.jsp
[5]Networks available for download here: http://hb.flatironinstitute.org/download

than those traditionally strong baselines simply due to lack of appropriate sampling. Graph convolutional networks, being highly computationally intensive, in particular on the larger datasets, were under-sampled compared to the other methods—it is possible that with more compute time their performance would improve. Note the direction of this bias: *were more samples to improve the performance of the GCNN methods further, it would only strengthen the performance gap observed on the largest datasets, and potentially render them more performant than the simpler models on our smaller datasets.* Because this bias is in favor of our baselines, rather than the more exotic, structured GCNN models, we feel comfortable still reporting these results even though they may improve later.

For our data-flush regimes (the tasks over the full and prostate only LINCS datasets), we used only the Level 4 data. This data is less processed, but presents 3 times as much data as the analogous Level 5 data. Note that had we used Level 5 data, our filtering procedure eliminating classes with less than 1000 examples would have eliminated many classes and made the overall task much easier. For our data-sparse tests (the task on our private L1000 corpus), we tested methods on both datasets, wondering whether in this data-sparse regime, the more processed data might prove more valuable than the relatively small increase in dataset size. Additionally, as in neither dataset on the MGH corpus did we filter out infrequent classes (given the dataset size, all classes are infrequent by our standards for the full LINCS data), this change from Level 5 to Level 4 can be done more transparently than on the full LINCS datasets.

Along with our code, the results of these hyperparameter searches are all publicly available.[1]

4 RESULTS & DISCUSSION

4.1 LINCS Corpus

4.1.1 Full Corpus. Final results are shown in Table 2. Accuracies and macro F1s are reported averaged across unseen test folds, using hyperparameters found via a separate validation fold. Included in the results are those obtained using a majority class classifier, which simply predicts the most frequent class with probability equal to that found in the training set. This was tested across the same folds and is reported here to ground all other reported results and variances.

We note that on all of the tested tasks, GCNNs perform best, by notable margins in accuracy and macro F1 on both primary site and subtype prediction. The margin of accuracy in MOA prediction is smaller, but still statistically significant. KNNs performed surprisingly well on all three tasks, offering competitive performance even with the FF-ANNs. Investigations of why they performed so well revealed two findings:

(1) KNN classifiers strongly prefer traditional distance metrics (e.g., Euclidean) over correlative based "distance metrics." This is notable because correlation is often used as a signal of biological similarity on these data, which may be contraindicated by these results
(2) Beyond even euclidean distance metrics, optimal hyperparameters on all tasks (including those on the other corpora) used the "Canberra" distance, defined via

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sum_i \frac{|\boldsymbol{x}_i - \boldsymbol{y}_i|}{|\boldsymbol{x}_i| + |\boldsymbol{y}_i|}.$$

This distance metric is traditionally used for integer valued vectors and we are unsure why it would be preferred here. We have not performed analyses to determine if this apparent distance metric preference is statistically significant.

Task	Classifier Name	Accuracy	Macro F1
Primary Site	GCNN	**93.9 ± 0.28**	**90.5 ± 0.82**
	FF-ANN	90.6 ± 0.44	85.6 ± 0.97
	KNNs	89.6 ± 0.30	87.2 ± 0.61
	Linear Classifier	60.9 ± 0.50	47.6 ± 0.63
	Random Forest	57.2 ± 0.48	40.2 ± 0.77
	Decision Tree	44.4 ± 0.70	24.7 ± 2.22
	Majority Class	27.9 ± 0.16	3.63 ± 0.02
Subtype	GCNN	**93.5 ± 0.34**	**91.7 ± 2.1**
	FF-ANN	90.5 ± 0.30	88.5 ± 0.54
	KNNs	89.8 ± 0.13	90.2 ± 0.27
	Linear Classifier	62.6 ± 0.62	56.3 ± 1.06
	Random Forest	51.7 ± 0.37	22.3 ± 0.49
	Decision Tree	41.1 ± 0.21	18.4 ± 0.62
	Majority Class	34.0 ± 0.21	3.62 ± 0.02
MOA	GCNN	**46.4 ± 0.35**	**31.6 ± 0.65**
	FF-ANN	45.9 ± 0.43	29.6 ± 0.60
	KNNs	43.5 ± 0.50	29.5 ± 0.58
	Linear Classifier	39.1 ± 0.29	20.6 ± 0.39
	Random Forest	32.3 ± 0.40	11.5 ± 0.31
	Decision Tree	28.7 ± 0.31	8.5 ± 0.29
	Majority Class	16.4 ± 0.16	0.57 ± 0.005

Table 2: Performance results attained for various classifiers on the full, heterogenous public LINCS corpus. All classifier comparisons were statistically significant ($p = 0.05$).

Classifier Name	Accuracy	Macro F1
GCNN	67.7 ± 0.76	46.0 ± 0.42
FF-ANN	**68.3 ± 0.60**	**50.4 ± 0.71**
KNNs	66.5 ± 0.71	46.2 ± 0.89
Linear Classifier	63.8 ± 0.52	42.6 ± 1.03
Random Forest	60.4 ± 0.48	37.4 ± 0.41
Decision Tree	53.2 ± 1.16	32.6 ± 0.91
Majority Class	34.54 ± 0.05	5.71 ± 0.01

Table 3: Performance results attained for various classifiers on the prostate LINCS corpus and MOA prediction task. Graph convolutional neural networks preferred non-specific regulatory graphs. All classifier comparisons were statistically significant ($p = 0.05$).

Linear classifiers robustly performed well, whereas random forest and decision trees both yielded underwhelming results, particularly with respect to Macro F1. One hypothesis as to why this may be is that Random Forests were less sampled in the hyperparameter search than linear models. Alternatively, these results may suggest that absolute feature values are less meaningful in our data than

Classifier Name	Level 5		Level 4	
	Accuracy	Macro F1	Accuracy	Macro F1
GCNN	46.0 ± 9.90	44.0 ± 10.8	54.6 ± 3.94	56.4 ± 3.94
FF-ANN	**63.2 ± 10.3**	**62.7 ± 10.8**	**57.3 ± 4.12**	**58.9 ± 4.00**
KNNs	46.9 ± 8.13	44.7 ± 9.15	44.9 ± 3.74	45.7 ± 3.61
Linear Classifier	52.3 ± 9.61	51.4 ± 10.0	49.1 ± 3.98	50.2 ± 3.63
Random Forest	48.0 ± 8.96	44.7 ± 9.15	43.2 ± 4.87	42.7 ± 4.75
Decision Tree	26.7 ± 8.07	25.6 ± 7.45	27.0 ± 2.02	26.4 ± 1.79
Majority Class	7.56 ± 2.37	0.23 ± 0.07	6.88 ± 0.77	0.21 ± 0.02

Table 4: Performance results attained for various classifiers on the perturbagen identity prediction task on the MGH NeuroBank Corpus. Results were *not* statistically significantly different at $p = 0.05$ between the Level 5 data and Level 4 data for any classifier save the GCNN. All within-level classifier comparisons were statistically significant ($p = 0.05$) save between Level 5 GCNNs and RF, GCNNs and KNNs, and KNNs and RFs.

are relationships between feature values—an idea that meshes well with the fact that this dataset is very heterogenous with respect to cell (e.g., tissue) type, and the same expression level of any individual gene may mean very different things in different tissue types. We have performed no deeper analyses to investigate these hypotheses and make no strong claims as to their validity absent further experiments.

4.1.2 Prostate Only Corpus. Final results for prediction of prostate MOA are shown in Table 3. Here, FF-ANNs perform best, though GCNNs are quite competitive. Again, KNNs perform well. Here, RFs and Decision Trees still under-perform the other methods, but perform better with respect to macro F1 than they do on the more heterogeneous full LINCS corpus, suggesting again that perhaps they may be more appropriate on more homogeneous data sources.

On these data, GCNNs still prefer the tissue independent regulatory networks over prostate specific graphs. This may indicate that our tissue-specific graphs are of low quality, or that tissue-independent graphs are simply more performant overall.

4.2 MGH NeuroBank Corpus

4.2.1 Raw Performance Results. Final results for perturbagen identification on the MGH NeuroBank corpus are shown in Table 4.

Here, FF-ANNs lead in performance by a wide margin compared to other methods. We interpret their strong success here relative to GCNNs to be indicative of a strong need for very large datasets for the GCNN models. Recall that this dataset is significantly smaller than our other datasets (see Table 1). This intuition is supported by two observations: 1) the apparent slope in GCNN performance relative to dataset size is quite steep, exceeding at all tasks on the largest dataset, nearly matching on the prostate only dataset, and failing by a large margin here, and 2) GCNNs show a statistically significant preference for the larger Level 4 data, whereas no other classifier cares between the two modalities in a statistically significant manner.

It is also possible that GCNNs are less appropriate on this corpus than on the larger corpora due to this dataset's strong neural focus. Or, it may be that GCNNs are most appropriate in heterogeneous datasets spanning many cell types. Ultimately, additional experimentation is needed to understand why they do worse here.

Classifier Name	Accuracy	Macro F1
GCNN	47.7 ± 6.78	48.9 ± 7.40
FF-ANN	**48.7 ± 7.85**	**50.1 ± 8.34**
KNNs	37.9 ± 5.39	39.0 ± 6.68
Linear Classifier	44.1 ± 4.03	44.7 ± 4.21
Random Forest	38.8 ± 5.37	38.3 ± 6.76
Decision Tree	22.0 ± 3.85	21.8 ± 3.59

Table 5: Generalization accuracy results on the MGH NeuroBank Corpus. Results measured on a held-out patient set, aggregated over all non-BD patients.

Among the other classifiers, linear classifiers perform well, followed by KNNs and RFs, then, much worse, by decision trees. No classifier save GCNNs shows a statistically significant preference for Level 5 data over Level 4 data, but all save GCNNs do show a (again, statistically *insignificant*) preference for Level 5 data in terms of absolute measure.

4.2.2 Generalization Experiments. We also used the MGH NeuroBank Corpus to assess population level generalizability, by training on four of our patients and testing on the fifth patient. As the MGH NeuroBank Corpus contains only one patient with Bipolar Disorder, we do not ever test on this patient's data—absent more examples of any patient data in this diagnostic category, we would not expect a classifier to generalize well to this patient. Including their results causes a mild but consistent drop in mean generalization accuracy across almost all classifiers tested. We report all results here using Level 4 data as no classifier statistically significantly preferred Level 5, but the relative drops in performance observed were similar for that modality.

Results for this experiment are shown in Table 5. All methods showed a notable drop in accuracy on unseen patients, ranging from a 10.2% drop for Linear Classifiers to an 18.5% drop for Decision Trees (percentages taken of per-sample accuracies, not raw percentage points). This indicates a definite unmet need for either a) more diverse datasets or b) novel methods able to better generalize to unseen patients. Note, though, that the MGH NeuroBank corpus only contains 5 total patients to begin with, so it may be the case that these numbers would improve significantly were we to have even a only marginally larger patient pool.

5 CONCLUSION

In this work we aimed to make the following contributions:

Establish biologically meaningful benchmark tasks for gene expression data. With the curation of the full and prostate-specific views of the LINCS dataset and specification of the Primary Site, Subtype, and MOA tasks, we meet this goal.

Provide robust benchmarks. We provide benchmarks on the tasks defined above for 6 different types of classifiers. We establish that graph convolutional neural networks, which incorporate prior biological knowledge via genetic regulatory graphs, perform very well when dataset size is very large, and feed-forward artificial neural networks offer good performance across all dataset sizes. Additionally, we profile non-neural classifiers, including K nearest neighbor methods, random forests, linear classifiers and decision trees. K nearest neighbor methods provide surprisingly strong performance in data rich environments using the Canberra distance.

Assess how these classifiers function in data-scarce regimes. We profile these same classifiers on a similar task on the smaller, privately produced MGH NeuroBank corpus. Here, we find that graph convolutional neural networks no longer offer competitive performance, but feed-forward artificial neural networks continue to perform well, as do linear models.

Assess population level generalizability. We demonstrate that patient level generalizability remains an important challenge in this domain. Linear classifiers generalize best, losing only 10.2% of their per-sample accuracy, while Decision Trees generalize worst, losing 18.5%. It is important to note that we were only able to assess this on our smallest dataset, the MGH NeuroBank Corpus, as differing cell lines represented too divergent demographic conditions in the full LINCS dataset, so this may simply be a reflection of the small dataset size, or indicative of a more chronic problem due to the fact that gene expression corpora contain many samples per patient.

6 FUTURE WORK

There are several notable directions for future work. First, a notable absent classifier is a self-normalizing neural network (SNNN) [19]. Introduced in late 2017, SNNNs have demonstrated improvements in a battery of different tasks and warrant inclusion here. Other types of classifiers capable of using graph structures would also warrant inclusion. Additionally, there are other graph convolutional networks one could use, [14, 20], as well as other sources for our regulatory graphs. One notable contender in that domain is *HuRI: The Human Reference Protein Interactome Mapping Project*[6] which has several large databases of protein-protein interactions found experimentally through yeast two-hybrid screening methods [12, 25]. Finally, we would also like to establish other types of machine learning benchmark tasks, most notably clustering tasks, or other tasks that can better assess generalizability across patients, drugs, or even measurement technologies.

ACKNOWLEDGMENTS

This research was funded in part by grants from the National Institutes of Health (NIH): National Institute of Mental Health (NIMH) grant P50-MH106933, National Human Genome Research Institute (NHGRI) grant U54-HG007963.

[6] http://interactome.baderlab.org/about/

REFERENCES

[1] Alexander Aliper, Sergey Plis, Artem Artemov, Alvaro Ulloa, Polina Mamoshina, and Alex Zhavoronkov. 2016. Deep Learning Applications for Predicting Pharmacological Properties of Drugs and Drug Repurposing Using Transcriptomic Data. *Molecular Pharmaceutics* 13, 7 (July 2016), 2524–2530. https://doi.org/10.1021/acs.molpharmaceut.6b00248

[2] David B Allison, Xiangqin Cui, Grier P Page, and Mahyar Sabripour. 2006. Microarray data analysis: from disarray to consolidation and consensus. *Nature Reviews. Genetics* 7, 1 (Jan. 2006), 55–65. https://www.nature.com/articles/nrg1749

[3] James Bergstra and Yoshua Bengio. 2012. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research* 13, Feb (2012), 281–305. http://www.jmlr.org/papers/v13/bergstra12a.html

[4] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2016. Geometric deep learning: going beyond Euclidean data. *arXiv:1611.08097 [cs]* (Nov. 2016). http://arxiv.org/abs/1611.08097 arXiv: 1611.08097.

[5] Lujia Chen, Chunhui Cai, Vicky Chen, and Xinghua Lu. 2016. Learning a hierarchical representation of the yeast transcriptomic machinery using an autoencoder model. *BMC Bioinformatics* 17, 1 (Jan. 2016), S9. https://doi.org/10.1186/s12859-015-0852-1

[6] Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, and Xiaohui Xie. 2016. Gene expression inference with deep learning. *Bioinformatics* 32, 12 (June 2016), 1832–1839. https://doi.org/10.1093/bioinformatics/btw074

[7] Neil R. Clark, Kevin S. Hu, Axel S. Feldmann, Yan Kou, Edward Y. Chen, Qiaonan Duan, and Avi Ma'ayan. 2014. The characteristic direction: a geometrical approach to identify differentially expressed genes. *BMC Bioinformatics* 15 (March 2014), 79. https://doi.org/10.1186/1471-2105-15-79

[8] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. *arXiv:1606.09375 [cs, stat]* (June 2016). http://arxiv.org/abs/1606.09375 arXiv: 1606.09375.

[9] domdomegg. 2016. A simple diagram of an unspecialised animal cell without labels. (Jan. 2016). https://commons.wikimedia.org/wiki/File:Simple_diagram_of_animal_cell_(blank).svg

[10] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. 2015. Convolutional Networks on Graphs for Learning Molecular Fingerprints. *arXiv:1509.09292 [cs, stat]* (Sept. 2015). http://arxiv.org/abs/1509.09292 arXiv: 1509.09292.

[11] Tracey M. Filzen, Peter S. Kutchukian, Jeffrey D. Hermes, Jing Li, and Matthew Tudor. 2017. Representing high throughput expression profiles via perturbation barcodes reveals compound targets. *PLOS Computational Biology* 13, 2 (Feb. 2017), e1005335. https://doi.org/10.1371/journal.pcbi.1005335

[12] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. 2017. Protein Interface Prediction using Graph Convolutional Networks. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 6533–6542. http://papers.nips.cc/paper/7231-protein-interface-prediction-using-graph-convolutional-networks.pdf

[13] Casey S. Greene, Arjun Krishnan, Aaron K. Wong, Emanuela Ricciotti, Rene A. Zelaya, Daniel S. Himmelstein, Ran Zhang, Boris M. Hartmann, Elena Zaslavsky, Stuart C. Sealfon, Daniel I. Chasman, Garret A. FitzGerald, Kara Dolinski, Tilo Grosser, and Olga G. Troyanskaya. 2015. Understanding multicellular function and disease with human tissue-specific networks. *Nature Genetics* 47, 6 (June 2015), 569–576. https://doi.org/10.1038/ng.3259

[14] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 1025–1035. http://papers.nips.cc/paper/6703-inductive-representation-learning-on-large-graphs.pdf

[15] Mikael Henaff, Joan Bruna, and Yann LeCun. 2015. Deep Convolutional Networks on Graph-Structured Data. *arXiv:1506.05163 [cs]* (June 2015). http://arxiv.org/abs/1506.05163 arXiv: 1506.05163.

[16] Rachel Hodos, Ping Zhang, Hao-Chih Lee, Qiaonan Duan, Zichen Wang, Neil R. Clark, Avi Ma'ayan, Fei Wang, Brian Kidd, Jianying Hu, David Sontag, and Joel Dudley. 2017. Cell-specific prediction and application of drug-induced gene expression profiles. In *Biocomputing 2018*. WORLD SCIENTIFIC, 32–43. http://www.worldscientific.com/doi/abs/10.1142/9789813235533_0004 DOI: 10.1142/9789813235533_0004.

[17] Victoria Hore, Ana Viñuela, Alfonso Buil, Julian Knight, Mark I. McCarthy, Kerrin Small, and Jonathan Marchini. 2016. Tensor decomposition for multiple-tissue gene expression experiments. *Nature Genetics* 48, 9 (Sept. 2016), 1094–1100. https://doi.org/10.1038/ng.3624

[18] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907 [cs, stat]* (Sept. 2016). http://arxiv.org/abs/1609.02907 arXiv: 1609.02907.

[19] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-Normalizing Neural Networks. *arXiv:1706.02515 [cs, stat]* (June 2017). http://arxiv.org/abs/1706.02515 arXiv: 1706.02515.

[20] Ron Levie, Federico Monti, Xavier Bresson, and Michael M. Bronstein. 2017. CayleyNets: Graph Convolutional Neural Networks with Complex Rational Spectral Filters. *arXiv:1705.07664 [cs]* (May 2017). http://arxiv.org/abs/1705.07664 arXiv: 1705.07664.

[21] Chieh Lin, Siddhartha Jain, Hannah Kim, and Ziv Bar-Joseph. 2017. Using neural networks for reducing the dimensions of single-cell RNA-Seq data. *Nucleic Acids Research* 45, 17 (Sept. 2017), e156–e156. https://doi.org/10.1093/nar/gkx681

[22] Zhi-Ping Liu, Canglin Wu, Hongyu Miao, and Hulin Wu. 2015. RegNetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. *Database* 2015 (Jan. 2015). https://doi.org/10.1093/database/bav095

[23] Matthew B.A. McDermott, Tom Yan, Tristan Naumann, Nathan Hunt, Harini Suresh, Peter Szolovits, and Marzyeh Ghassemi. 2018. Semi-supervised Biomedical Translation with Cycle Wasserstein Regression GANs. In *Association for the Advancement of Artificial Intelligence*. New Orleans, LA. http://www.marzyehghassemi.com/wp-content/uploads/2018/01/semi-supervised-CWR-GAN_Ghassemi.pdf

[24] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (Oct. 2011), 2825âĹŠ2830. http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

[25] Thomas Rolland, Murat Taşan, Benoit Charloteaux, Samuel J. Pevzner, Quan Zhong, Nidhi Sahni, Song Yi, Irma Lemmens, Celia Fontanillo, Roberto Mosca, Atanas Kamburov, Susan D. Ghiassian, Xinping Yang, Lila Ghamsari, Dawit Balcha, Bridget E. Begg, Pascal Braun, Marc Brehme, Martin P. Broly, Anne-Ruxandra Carvunis, Dan Convery-Zupan, Roser Corominas, Jasmin Coulombe-Huntington, Elizabeth Dann, Matija Dreze, Amélie Dricot, Changyu Fan, Eric Franzosa, Fana Gebreab, Bryan J. Gutierrez, Madeleine F. Hardy, Mike Jin, Shuli Kang, Ruth Kiros, Guan Ning Lin, Katja Luck, Andrew MacWilliams, Jörg Menche, Ryan R. Murray, Alexandre Palagi, Matthew M. Poulin, Xavier Rambout, John Rasla, Patrick Reichert, Viviana Romero, Elien Ruyssinck, Julie M. Sahalie, Annemarie Scholz, Akash A. Shah, Amitabh Sharma, Yun Shen, Kerstin Spirohn, Stanley Tam, Alexander O. Tejeda, Shelly A. Trigg, Jean-Claude Twizere, Kerwin Vega, Jennifer Walsh, Michael E. Cusick, Yu Xia, Albert-László Barabási, Lilia M. Iakoucheva, Patrick Aloy, Javier De Las Rivas, Jan Tavernier, Michael A. Calderwood, David E. Hill, Tong Hao, Frederick P. Roth, and Marc Vidal. 2014. A Proteome-Scale Map of the Human Interactome Network. *Cell* 159, 5 (Nov. 2014), 1212–1226. https://doi.org/10.1016/j.cell.2014.10.050

[26] Thomas Shafee. 2015. Protein coding genes are transcribed to an mRNA intermediate, then translated to a functional protein. RNA-coding genes are transcribed to a functional non-coding RNA. (PDB: 3BSE, 1OBB, 3TRA) Annotated version of not uploaded yet. (April 2015). https://commons.wikimedia.org/wiki/File:DNA_to_protein_or_ncRNA.svg

[27] Aravind Subramanian, Rajiv Narayan, Steven M. Corsello, David D. Peck, Ted E. Natoli, Xiaodong Lu, Joshua Gould, John F. Davis, Andrew A. Tubelli, Jacob K. Asiedu, David L. Lahr, Jodi E. Hirschman, Zihan Liu, Melanie Donahue, Bina Julian, Mariya Khan, David Wadden, Ian Smith, Daniel Lam, Arthur Liberzon, Courtney Toder, Mukta Bagul, Marek Orzechowski, Oana M. Enache, Frederica Piccioni, Alice H. Berger, Alykhan Shamji, Angela N. Brooks, Anita Vrcic, Corey Flynn, Jacqueline Rosains, David Takeda, Desiree Davison, Justin Lamb, Kristin Ardlie, Larson Hogstrom, Nathanael S. Gray, Paul A. Clemons, Serena Silver, Xiaoyun Wu, Wen-Ning Zhao, Willis Read-Button, Xiaohua Wu, Stephen J. Haggarty, Lucienne V. Ronco, Jesse S. Boehm, Stuart L. Schreiber, John G. Doench, Joshua A. Bittker, David E. Root, Bang Wong, and Todd R. Golub. 2017. A Next Generation Connectivity Map: L1000 Platform And The First 1,000,000 Profiles. *bioRxiv* (May 2017), 136168. https://doi.org/10.1101/136168

[28] Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov. 2005. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences* 102, 43 (Oct. 2005), 15545–15550. https://doi.org/10.1073/pnas.0506580102

[29] Zichen Wang, Neil R. Clark, and Avi Ma'ayan. 2016. Drug-induced adverse events prediction with the LINCS L1000 data. *Bioinformatics* 32, 15 (Aug. 2016), 2338–2345. https://doi.org/10.1093/bioinformatics/btw168

[30] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. 2018. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* (2018).