

# Deep Learning Benchmarks on L1000 Gene Expression Data

by

Matthew B. A. McDermott

B.S., Harvey Mudd College (2014)

Submitted to the Department of Electrical Engineering and Computer  
Science

in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2019

© Massachusetts Institute of Technology 2019. All rights reserved.

**Signature redacted**

Author .....

Department of Electrical Engineering and Computer Science

January 31st, 2019

**Signature redacted**

Certified by .....

Peter Szolovits

Professor of Electrical Engineering and Computer Science

Thesis Supervisor

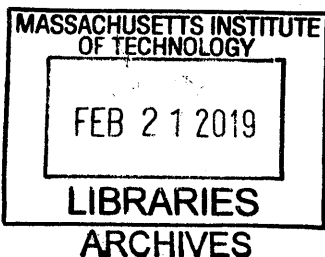
**Signature redacted**

Accepted by .....

Leslie A. Kolodziejski

Professor of Electrical Engineering and Computer Science

Chair, Department Committee on Graduate Students





# Deep Learning Benchmarks on L1000 Gene Expression Data

by

Matthew B. A. McDermott

Submitted to the Department of Electrical Engineering and Computer Science  
on January 31st, 2019, in partial fulfillment of the  
requirements for the degree of  
Master of Science in Computer Science and Engineering

## Abstract

Gene expression data holds the potential to offer deep, physiological insights about the dynamic state of a cell beyond the static coding of the genome alone. I 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 an empirical methodological foundation, including published benchmarks and well characterized baselines.

In this work, we lay that foundation by profiling a battery of classifiers against newly defined biologically motivated classification tasks on multiple L1000 gene expression datasets. In addition, on our smallest dataset, a privately produced L1000 corpus, we profile per-subject generalizability to provide a novel assessment of performance that is lost in many typical analyses.

We compare traditional classifiers, including feed-forward artificial neural networks (FF-ANNs), linear methods, random forests, decision trees, and  $K$  nearest neighbor classifiers, as well as graph convolutional neural networks (GCNNs), which augment learning via prior biological domain knowledge. We find GCNNs offer performance improvements given sufficient data, excelling at all tasks on our largest dataset. On smaller datasets, FF-ANNs offer greatest performance. Linear models significantly underperform on all dataset scales, but offer the best per-subject generalizability.

Ultimately, these results suggest that structured models such as GCNNs can represent a new direction of focus for the field as our scale of data continues to increase.

Thesis Supervisor: Peter Szolovits

Title: Professor of Electrical Engineering and Computer Science



## Acknowledgments

Neither I nor this work would not be where it is without the help and support of many people.

Most directly, this work was made directly possible by the experimental support of Professor Roy Perlis's and Professor Steve Haggarty's research groups at Massachusetts General Hospital (MGH), Professor Isaac Kohane of Harvard Medical School (HMS) and the broader N-Grid CEGS Collaboration<sup>1</sup>. In particular, Dr. Jennifer Wang, Dr. Wen-Ning Zhao, Dr. Steven D. Sheridan, all of MGH, were integral to this work, as was Dr. Susanne Churchill of HMS.

Both within and beyond this direct collaboration, I have been extremely fortunate in my advisor, Professor Peter Szolovits, who has been exactly the kind of advisor I needed during my studies, offering excellent advice and critical guidance, while simultaneously allowing me to explore and make my studies my own.

I've also benefited enormously from the support and camaraderie of Professor Szolovits' research group, the members of which have all been invaluable in shaping my work.

Last but certainly not least, this research was funded in part by grants from the National Institutes of Health (NIH): National Institute of Mental Health (NIMH) grant P50-MH106933 and National Human Genome Research Institute (NHGRI) grant U54-HG007963.

### Prior Publication

This work was presented at BioKDD 2018,<sup>2</sup> and is under review for IEEE TCBB SI: BioKDD 2018.

---

<sup>1</sup><https://n-grid.org/>

<sup>2</sup> Matthew B.A. McDermott, Jennifer Wang, Wen-Ning Zhao, Steven D. Sheridan, Peter Szolovits, Isaac Kohane, Stephen J. Haggarty, and Roy H. Perlis. Deep Learning Benchmarks on Gene Expression Data. *Presented at BioKDD*, August, 2018. [https://github.com/mmcdermott/LINCS\\_Deep\\_Learning\\_Benchmarks](https://github.com/mmcdermott/LINCS_Deep_Learning_Benchmarks)



# Contents

<b>1</b>	<b>Introduction</b>	<b>13</b>
1.1	Introduction . . . . .	13
1.2	Gene Expression Data . . . . .	16
1.2.1	The Biology . . . . .	16
1.2.2	Measuring Gene Expression/Transcriptomics . . . . .	16
1.2.3	Measurement Techniques . . . . .	17
1.2.4	Experimental Pipelines . . . . .	19
1.3	Machine Learning on Gene Expression Data . . . . .	19
1.3.1	Traditional Analyses . . . . .	19
1.3.2	Neural Representation Learning . . . . .	20
1.3.3	Neural Classification & Regression . . . . .	21
1.4	Structured Models via Graph Convolutional Networks . . . . .	21
1.4.1	Regulatory Graphs . . . . .	21
1.4.2	Graph Convolutional Networks in Theory . . . . .	22
1.4.3	Graph Convolutional Networks in Practice . . . . .	22
<b>2</b>	<b>Methodology</b>	<b>25</b>
2.1	Datasets . . . . .	25
2.1.1	Curated Views of the Public LINCS Corpus . . . . .	25
2.1.2	MGH NeuroBank Corpus . . . . .	28
2.2	Models . . . . .	31
2.2.1	Classifiers Tested . . . . .	31
2.3	Genetic Regulatory Networks Considered . . . . .	33

2.4	Experiments . . . . .	34
2.4.1	Hyperparameter Search & Technical Setup . . . . .	34
<b>3</b>	<b>Results &amp; Discussion</b>	<b>39</b>
3.1	LINCS Corpus . . . . .	39
3.1.1	Full Corpus . . . . .	39
3.1.2	Prostate Only Corpus . . . . .	41
3.2	MGH NeuroBank Corpus . . . . .	42
3.2.1	Raw Performance Results . . . . .	42
3.2.2	Generalization Experiments . . . . .	43
<b>4</b>	<b>Conclusion</b>	<b>47</b>
4.1	Summary . . . . .	47
4.2	Future Work . . . . .	48



# List of Figures

1-1	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: [11, 38]. . . . .	17
1-2	The L1000 technique is cheaper, but requires novel additional technical pre-processing compared to other measurement technologies. Each step in this flow induces more technical variability. In this work, the normalized, pre-aggregation data is referred to as “Level 4” data, whereas the post-aggregation data is referred to as “Level 5” data. This figure omits imputation of the full transcriptome as we never use imputed genes in this work. See [40] for full details. . . . .	18
1-3	Gene expression corpora are often produced by cloning a small number of pluripotent cellular sources, then perturbing and profiling those clones. Image: [11]. . . . .	19
1-4	The regulatory relationships between L1000 landmark genes, as determined according to [32]. 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. . .	23

2-1 A flowchart representing the various decision points when curating our views of the full LINCS corpus. . . . . 29

2-2 The dataset creation pipeline for the MGH NeuroBank Corpus. These data were created by our collaborators at the Center for Quantitative Health (dir. Professor Roy Perlis) and Chemical Neurobiology Laboratory (dir. Professor Stephen Haggarty) at Massachusetts General Hospital. Of particular note are Drs. Jennifer Wang, Wen-Ning Zhao, and Stephen D. Sheridan. . . . . 30

# List of Tables

2.1	Population Statistics for our Datasets and Tasks. . . . .	31
3.1	Performance (mean $\pm$ standard deviation) for the full, tissue-heterogenous LINCSeq corpus. . . . .	40
3.2	Performance (mean $\pm$ standard deviation) on the prostate LINCSeq corpus and MOA prediction task. . . . .	42
3.3	Performance (mean $\pm$ standard deviation) on the perturbation identity task on the MGH NeuroBank Corpus. . . . .	43
3.4	Per (Non-BD) Subject Generalization Accuracy (mean $\pm$ standard deviation) on the MGH NeuroBank Corpus. . . . .	45
1	Optimal Hyperparameters for LINCSeq Full MOA. . . . .	50
2	Optimal Hyperparameters for LINCSeq Full Subtype. . . . .	51
3	Optimal Hyperparameters for LINCSeq Full Primary Site. . . . .	52
4	Optimal Hyperparameters for LINCSeq Prostate Only MOA. . . . .	53
5	Optimal Hyperparameters for MGH NeuroBank Corpus Level 4. . . . .	54
6	Optimal Hyperparameters for MGH NeuroBank Corpus Level 5. . . . .	55



# Chapter 1

## Introduction

### 1.1 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, and 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 [40].

Simultaneous with the increasing availability of gene expression data, deep learning techniques have grown vastly more powerful and popular—showing advances in image processing [19, 13, 24], natural language processing [27, 16, 39], and speech recognition/generation [34, 28], among other fields. In some limited areas, these advances have also translated into the biomedical domain—for example, in analyzing mass spectrometry spectra [42], DNA sequences [45], amino acid sequences [30, 43, 5], or biomedical images [13, 19].

However, among non-sequential, non-imaging modalities, such as gene expression data, “deep” learning methods generally remain limited to simple, unstructured, shallow modeling techniques. In particular, while large-scale benchmarks such as the

ImageNet challenge<sup>1</sup> and the existence of an underlying mathematical structure have fueled the development of convolutional neural networks (CNNs) for image processing or recurrent neural network (RNNs) for sequential analysis, bioinformaticians are limited to unstructured feed-forward artificial neural networks (FF-ANNs), which are known to be relatively inefficient learners [26].

In this work, we aim to lay a foundation that will help deep learning succeed for gene expression data as it has in these other domains by providing a fixed definition of success via benchmarks and offering a potential avenue for using structure to create more intelligent modeling approaches. In particular, we define three biologically motivated benchmarking tasks over two curated views<sup>2</sup> 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 genes, 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 perform very well, but require large amounts of data, excelling at all tasks on our largest dataset, but under-performing FF-ANNs on our smaller datasets. Of other methods, FF-ANNs perform best, followed consistently by linear classifiers, then random forests, then decision trees. KNN classifiers perform very well on our larger datasets, nearly matching FF-ANNs, but they underwhelm on our smaller datasets.

Gene expression datasets often contain many samples spanning a very small set of subjects, as a single subject’s gene expression profile may be taken many times under varying conditions (e.g., drugs, etc.). As such, a pronounced risk when modelling gene

---

<sup>1</sup> ImageNet is a dataset containing millions of labeled images; its associated challenge tasks computer vision researchers to design algorithms to identify the objects in these images among a fixed set of categories. Many see ImageNet as a critical seed to the current deep learning boom [17, 37]

<sup>2</sup> See [https://github.com/mmcdermott/LINCS\\_Deep\\_Learning\\_Benchmarks](https://github.com/mmcdermott/LINCS_Deep_Learning_Benchmarks)

expression data is that you will learn a model specific to the very limited population expressed in your data. With datasets such as the LINCS data, which often only have the equivalent of one or two subjects (i.e., cell lines) per tissue type, assessing the extent of this overfitting can be difficult, and many works merely report per-sample performance metrics (allowing the model to train and test on the same set of subjects). In this work, we use our private, smaller corpus to assess per-subject generalizability by training on a restricted set of subjects and testing on a held-out subject. We find that all methods struggle to generalize to unseen subjects, showing performance drops ranging from 10 to 18 percent of their per-sample accuracies.

In sum, in this work we make the following contributions:

1. We establish biologically meaningful classification benchmarks at deep learning scale on the largest publicly available gene expression dataset. This is important because absent a shared, consistent view of the data and definition of success, 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 subjects to assess population-level generalizability.

## 1.2 Gene Expression Data

### 1.2.1 The Biology

The cellular system is governed at the root by the genome: the sequence of DNA base pairs that encode all information necessary for the cell's development and day to day functioning. In order to transmute 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. In this way, we can think of the genome as providing a function library for the cell system, and the proteins present (i.e., the expressed genes) as the actual mechanisms behind cellular functioning. A cell's gene expression profile thus captures a view into the dynamic state of the cell and offers insight far beyond the fixed picture of the DNA 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 coded by 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 what transcription and how) is a topic of intense study.

### 1.2.2 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 1-1). Transcriptomic gene expression is far more easily measured and we will focus on this modality in this work.

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



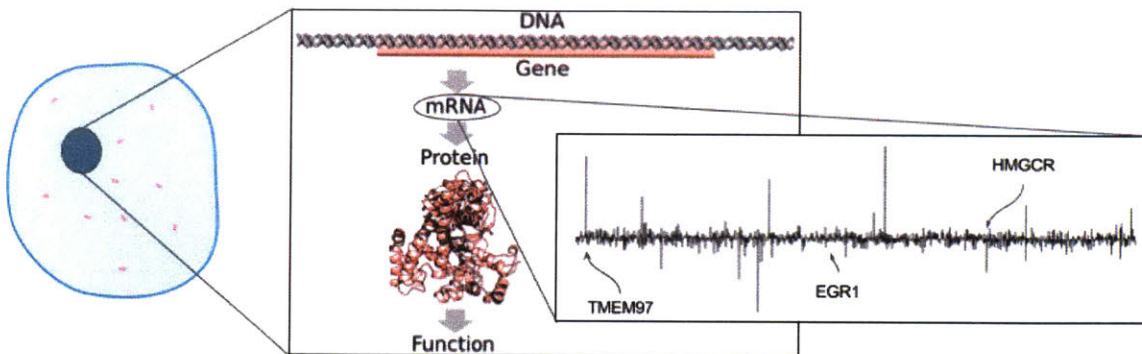


Figure 1-1: Transcriptomics data is measured by quantifying the mRNA produced during transcription. The output of this process is a vector with each dimension quantifying the expression of a particular gene. Both technical (e.g., misplaced reads) and biological (e.g., tissue type) factors add variance to these data. Images: [11, 38].

of the increased availability of data.

### 1.2.3 Measurement Techniques

Transcriptomics data itself can be measured by many techniques, including RNA-Seq, single-cell RNA-Seq (sc RNA-Seq), and the L1000 platform, which we focus on here. The L1000 platform [40] 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 approximately 1.3M samples, available on GEO at accession number GSE92742.<sup>3</sup>

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:

<sup>3</sup> <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE92742>

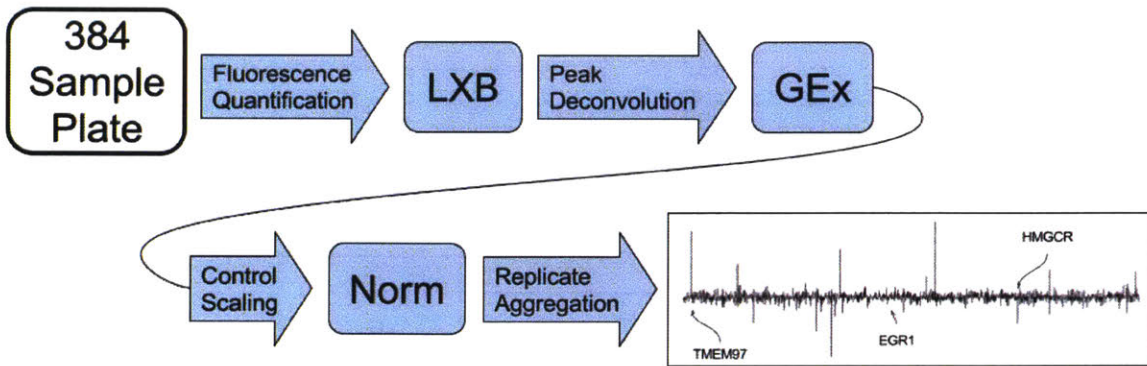


Figure 1-2: The L1000 technique is cheaper, but requires novel additional technical pre-processing compared to other measurement technologies. Each step in this flow induces more technical variability. In this work, the normalized, pre-aggregation data is referred to as “Level 4” data, whereas the post-aggregation data is referred to as “Level 5” data. This figure omits imputation of the full transcriptome as we never use imputed genes in this work. See [40] for full details.

**Level 4 (a.k.a. Roast)** Level 4 data is fully normalized and z-scored, and presented at the level of one profile per sample. From a machine learning perspective, this is what you would expect to work with when thinking of “raw L1000 output.”

**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 [40] 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 machine learning. 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.

See Figure 1-2 for a graphical representation of a subset of the L1000 technical pre-processing pipeline.

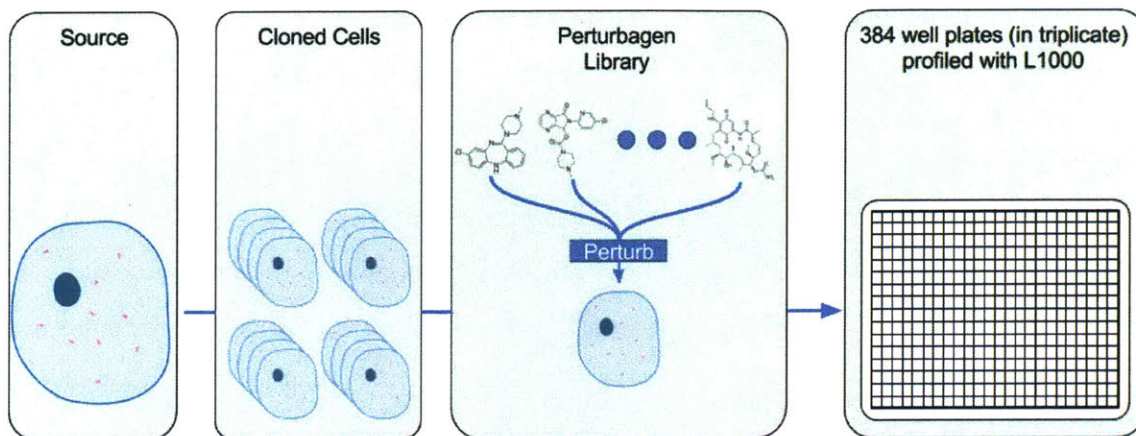


Figure 1-3: Gene expression corpora are often produced by cloning a small number of pluripotent cellular sources, then perturbing and profiling those clones. Image: [11].

## 1.2.4 Experimental Pipelines

In general, experimental pipelines producing large corpora of gene expression data work by acquiring some base cellular sample in pluripotent form, either patient derived or via a stock cell line, cloning that cell line extensively, then perturbing a number of samples and profiling them (Figure 1-3). 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.

## 1.3 Machine Learning on Gene Expression Data

### 1.3.1 Traditional Analyses

Traditional analyses on these data focus on statistical or geometric tests for differential gene expression [8], gene set enrichment analyses (GSEA) [41], and (for the L1000 platform specifically) signature based analyses [2, 40]. Some have also used tensor decomposition/completion to disentangle cell-type from perturbagen effects [22, 23], and explored traditional classifiers for adverse drug event prediction [44].

### 1.3.2 Neural Representation Learning

Other authors have used neural network models to build embeddings of gene expression data. In [14], the authors use a twin network architecture to represent gene expression profiles as 100 dimensional bar-codes. Comparing their representations to the standard representation (raw z-scores) and a representation based on gene set enrichment analysis (GSEA) they find that their perturbation barcodes consistently identify replicates, samples generated from perturbagens with shared targets, and show better clustering overlap with structural properties. Their architecture is a two layer deep feed-forward neural network, and notably uses inter-replicate variability (a sign of the noise within the technique) as a mechanism to help train their network to learn embeddings that natively differentiate between meaningful biological variation and confounding noise. However, their approach is aimed towards unsupervised representation learning, and thus only can provide soft benchmarking utility, as compared to a supervised task (though this may be more useful in some practical settings).

In [6], the authors use a 4 layer deep sparse autoencoder to analyze binarized yeast differential gene expression microarray data (e.g. their model took as input one feature per gene, with value 1 if that gene was differentially expressed in that sample, and value 0 if not). Their 4 layer architecture is designed to map to known biological levels of processing, and post-training analyses suggested good overlap between transcription-factor mediated regulatory relationships and the connections trained by their network between the first input layer and the first hidden layer.

In [31], 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. These augmentations allowed them to significantly reduce the parameter space relative to a fully connected, dense network of equal width.

### 1.3.3 Neural Classification & Regression

In [1], authors use a FF-ANN to classify profiles into categories based on the therapeutic effect of the generating perturbation. Researchers have also explored neural techniques for extrapolating the L1000 set of landmark genes to the full transcriptome. In [7] and [33], the authors explore gene expression extrapolation, albeit in the latter work as a test case for a semi-supervised model architecture rather than a goal in itself. In both cases, FF-ANN models are able to realize significant performance gains over linear models, again demonstrating that nonlinear relationships play a significant role on this modality.

## 1.4 Structured Models via Graph Convolutional Networks

### 1.4.1 Regulatory Graphs

As stated in Section 1.2.1, 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, with genes forming the vertices of this graph and edges between genes representing regulatory relationships between those two genes. Regulatory graphs are often represented as directed graphs, with direction representing the direction of the regulatory interaction, but in this work we will simplify them to undirected graphs, having them instead simply flag a regulatory relationship in either direction between two genes. We visualize one such regulatory graph in Figure 1-4.

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 are dynamic, changing 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.

### 1.4.2 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 defined over a graph by repeatedly featurizing the data over local neighborhoods within 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. 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.

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 algorithms. In practice, many operations are approximated, which affects the power of these models.

### 1.4.3 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, [25] explored node classification on knowledge and citation graphs. In this vein, GCNNs have also been used in several biological tasks. For example, [20] classifies proteins viewed as nodes in varying tissue-specific protein protein interaction graphs, [12] learns representations of molecular

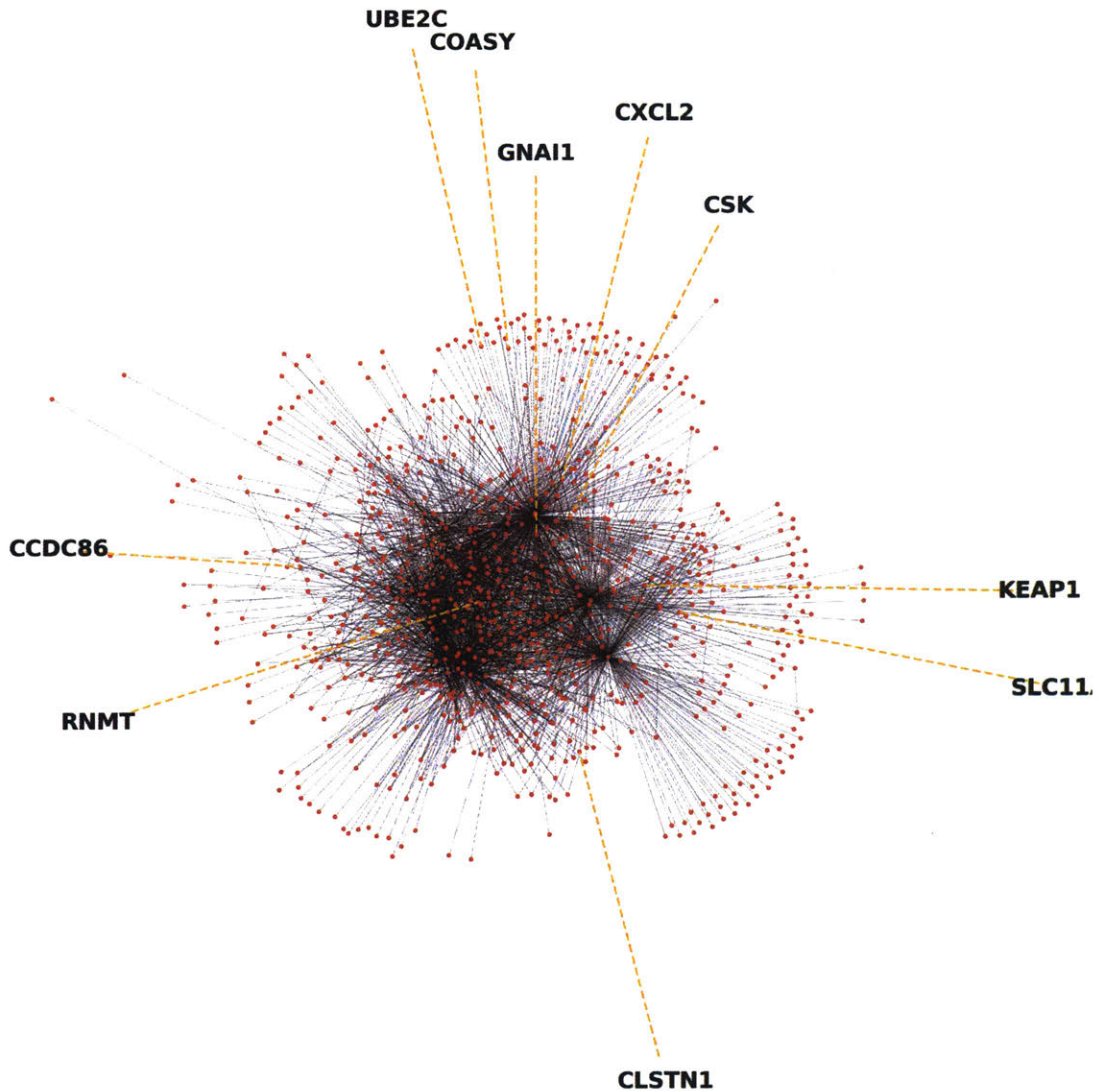


Figure 1-4: The regulatory relationships between L1000 landmark genes, as determined according to [32]. Nodes (red dots) are genes and edges between them represent known or suspected regulatory interactions. Note that many genes only have one known edge connecting them to much denser clusters within the center of the graph. This may reflect biological processes, or that some proteins are studied much more than others.

compounds interpreted as unique graphs with vertices determined by atoms and edges by bonds, and [15] learns representations of graphs defined by protein amino acid sequences for protein interface prediction.

Note that in these node classification tasks the authors are making predictions about nodes in a graph, rather than making predictions about a graph (such as a gene expression profile realized as a graph via a regulatory network, e.g. Figure 1-4). In the latter context, spectral methods are enticing; in fact, this picture is so appealing that many papers describing novel GCNN algorithms use this example to frame the impact of their ideas [9, 29, 4, 21]. 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.



# Chapter 2

## Methodology

Our principal goal here is to establish biologically meaningful benchmarking tasks for gene expression data, and to demonstrate the potential utility of structured methods that incorporate prior knowledge, across a variety of dataset sizes and levels of heterogeneity. To that end, in this section we will first profile the two datasets we will use and detail the benchmarking tasks we defined on each. Next, we will walk through the methods we test, paying close attention to the structured method we profile here, and finally we will detail our technical setup and experimental parameters.

### 2.1 Datasets

We will now detail the creation of both our views of the public LINCS corpus and the private MGH NeuroBank corpus. Full details and summary statistics for both corpora can also be found in Table 2.1.

#### 2.1.1 Curated Views of the Public LINCS Corpus

The full Level 4 LINCS dataset contains approximately 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 re-purposing 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 include compounds with only a single known MOA.

### **Dataset Curation Procedure**

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 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 lacked 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).

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 2.1. Note that there is significant class imbalance in this dataset—an unavoidable reflection of the corpus’s original makeup—but by filtering to a baseline number of examples per class we assert that there are at least a significant number of samples for every label, ensuring learning power. 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,<sup>1</sup> 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

---

<sup>1</sup>See [https://github.com/mmcdermott/LINCS\\_Deep\\_Learning\\_Benchmarks](https://github.com/mmcdermott/LINCS_Deep_Learning_Benchmarks)

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-subject*, *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.

A flowchart of our dataset curation process can also be seen in Figure 2-1.

### 2.1.2 MGH NeuroBank Corpus

Our private corpus of L1000 data was measured on a collection of subject-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 individuals, two healthy control subjects, one with Bipolar Disorder, and two with Schizophrenia (all diagnostic labels are DSM-IV diagnoses confirmed by structured clinical interview). All individuals' 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-subject generalizability, by training a model on only four of the five subjects, then testing on the data for the

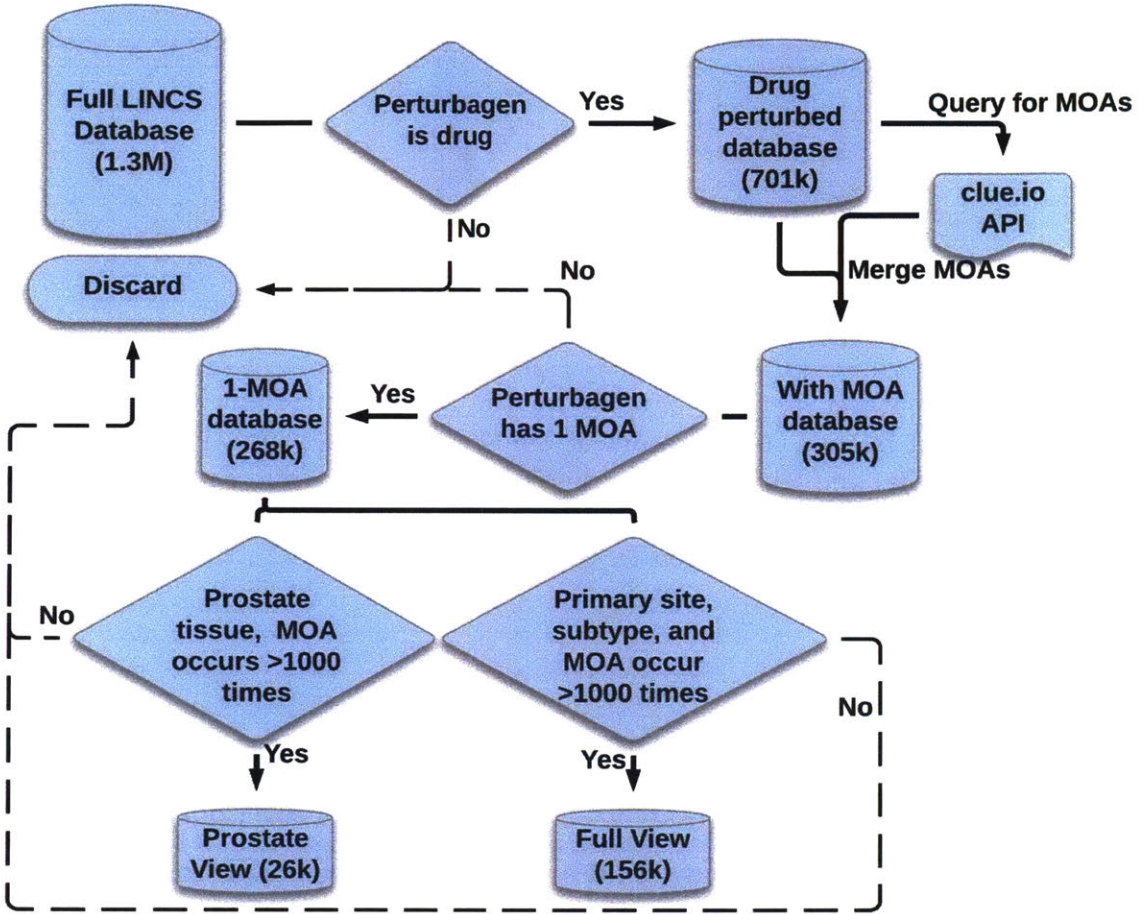


Figure 2-1: A flowchart representing the various decision points when curating our views of the full LINCS corpus.

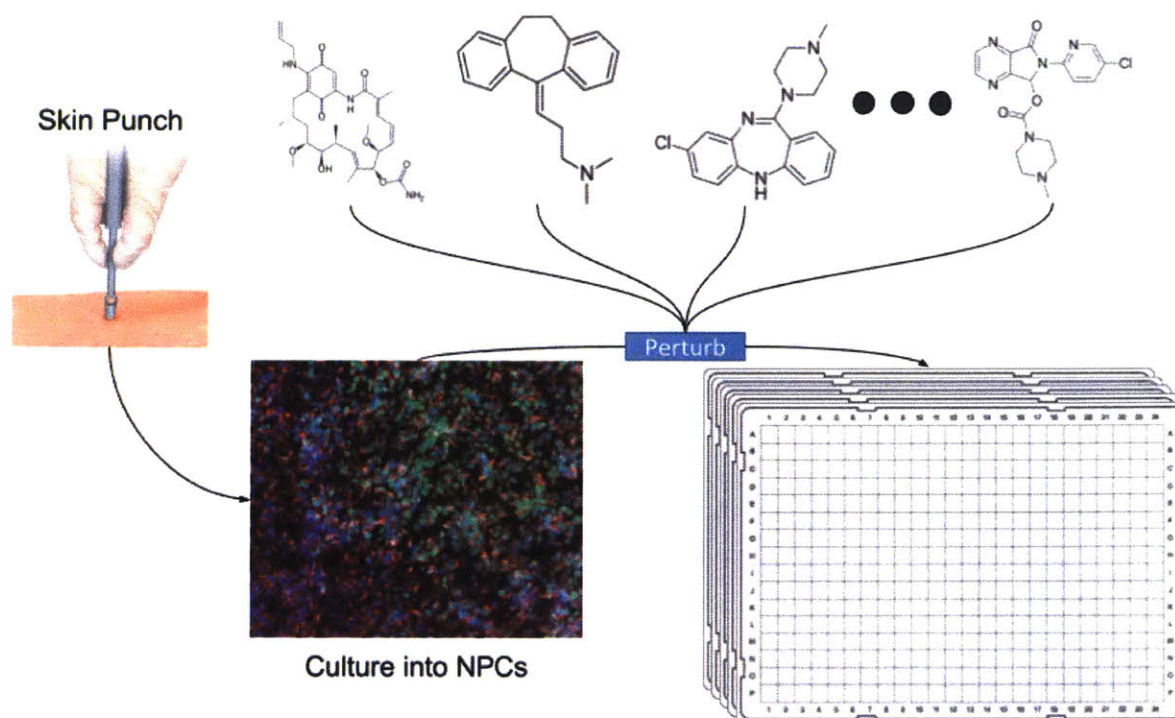


Figure 2-2: The dataset creation pipeline for the MGH NeuroBank Corpus. These data were created by our collaborators at the Center for Quantitative Health (dir. Professor Roy Perlis) and Chemical Neurobiology Laboratory (dir. Professor Stephen Haggarty) at Massachusetts General Hospital. Of particular note are Drs. Jennifer Wang, Wen-Ning Zhao, and Stephen D. Sheridan.

fifth subject. Per-subject generalizability is an important, oft-overlooked element of performance in this domain—many studies which rely only on data from the LINCS public corpora, for example, are often working with only one to two subjects per tissue type, which means their expected generalizability would likely be worse and the magnitude of the problem is difficult to assess. Our experiments here will provide some estimate of the performance delta that should be observed when generalizing to new subjects.

A graphical representation of the data creation pipeline can also be seen in Figure 2-2.

Table 2.1: Population Statistics for our Datasets and Tasks.

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)
LINCS (Prostate Only)	MOA	49	DMSO (25,638)	IKK Inhibitor (828)
	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)

## 2.2 Models

We compare a variety of standard classifiers, all (save GCNNs) implemented via `scikit-learn` [35] for maximal reproducibility and ease of use. GCNNs, as previously stated, were implemented via the method of [9].

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 1.4.

### 2.2.1 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 because they do not assume any particular structure of their input and are thus least constrained, they are relatively inefficient learners. Some postulate that this inefficiency is due to simply their larger parameter overhead; however, the full reason is not yet known. 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 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. Index construction, often a computationally intensive task on large datasets, was done via either brute force search, the construction of a `KDTree`, or the construction of a `Ball Tree`, as determined by `scikit-learn`'s `'algorithm=auto'` setting.

**Decision trees** Decision trees are low powered, but extremely mechanistically interpretable. Implemented via the `DecisionTreeClassifier` class.

**Graph Convolutional Neural Networks (GCNNs)** GCNNs allow us to inject prior biological knowledge in the form of a genetic regulatory network into a neural network, offering structural efficiency improvements and domain appropriate bias.

In this work, our GCNN is built using the spectral approach defined by [9]. We encourage interested readers to refer to the primary source for full details regarding this algorithm, but we provide a brief explanation of the method here. In particular, this method of graph convolutional processing approximates localized filters in the graph Fourier space via polynomials of the graph Laplacian. As follows from the graph theoretical nature of the Laplacian, restricting the order of these polynomials yields a localized radius of effect when impacting on the featurization of each graph node. These polynomials are realized in an efficient manner by relying on the stable recurrence relation of the Chebyshev polynomials, which form an orthogonal basis of a relevant Hilbert space and have been used historically in graph signal analysis for approximate wavelet analysis. Ultimately, this yields a means of producing fast, localized, graph convolutional filters. Graph pooling is implemented via the coarsening



phase of the Gracus multilevel clustering algorithm [10].

We use the code of [9] with minor modifications to support multi-component graphs. We considered a number of potential regulatory graphs, both tissue specific and tissue independent.

**Other Classifiers Considered** We also tested Naïve Bayes classifiers, Gaussian Processes Classifiers, Quadratic Discriminant Analysis, Boosted methods via Ada-boost, and Kernel Support Vector Classifiers, but these classifiers were removed from our experimental lineup for reasons varying from poor performance, non-insightful new results, computational intensity, or combinations thereof.

## 2.3 Genetic Regulatory Networks Considered

We considered a number of possible graphs, including those constructed from our data (a gene-gene pairwise correlation graph) and graphs pulled from the literature. For our literature sourced graphs, we will offer brief summaries of the source works here, but interested readers should refer to the primary sources for full 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 1-4.

Our tissue-independent regulatory network is a network of transcription-factor and micro-RNA mediated regulatory relationships summarized from 25 literature defined external datasets [32].<sup>2</sup> This graph is unweighted.

Our tissue-dependent regulatory network is built from a probabilistic model of tissue-specific gene-gene correlations [18].<sup>3</sup> We considered a number of possible tissues, profiling both relevant tissues (neuron for MGH NeuroBank and prostate gland for the prostate specific LINCS dataset) and irrelevant tissues (tooth, pancreas, skin fibroblast) to help differentiate whether any performance gains observed with these graphs were due to the appropriate tissue specificity or simply due to this style of graph construction being superior. These graphs were all weighted, with edge weights

---

<sup>2</sup>Networks available for download here: <http://www.regnetworkweb.org/download.jsp>

<sup>3</sup>Networks available for download here: <http://hb.flatironinstitute.org/download>

estimating the confidence in the true existence of that edge, determined via a probabilistic model. Ultimately, no tissue specific graph outperformed the tissue-independent graph of [32], so the distinction between relevant or irrelevant tissues proved negligible.

Last but not least, we considered a learned graph, whose (weighted) adjacency matrix was determined via correlation coefficients between genes in our data. This graph is by default fully connected, so to induce effective sparsity, we removed all edges corresponding to correlative relationships with a  $p$  value of less than 0.05. This graph also offered no performance advantage over the tissue-independent network in early experiments, so we removed it from our analyses and focused solely on the literature-defined graphs.

When working with any weighted graphs, we culled all edges with confidence below a cutoff threshold, which was tuned with all other hyperparameters. We treated all graphs as undirected, allowing them to capture merely a notion of regulatory interaction rather than any directed up- or down-regulation. This is certainly a simplification, and exploring more complex representations of regulatory graphs is definitely a promising area of future studies, but using undirected graphs here yields significant technical simplifications for this work enabling these graphs to work natively within our chosen graph convolutional framework.

## 2.4 Experiments

### 2.4.1 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. In addition to random search, we also rotated the discovered optimal hyperparameters across tasks during various stages of the search procedure and made certain manual tweaks in pursuit of obtaining strong performance metrics for

all models, particularly baseline methods. 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 rotated progressively through a very limited fixed set of number of epochs for computational reasons. Additionally, GCNNs only supported a single optimizer, whereas FF-ANNs offered several options. 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.

For our benchmarking tasks, 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.<sup>4</sup> Additionally, the optimal hyperparameters for all methods across all datasets and tasks can be found in the Appendix.

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<sup>5</sup> in our test set results below, as some tasks present

---

<sup>4</sup>See [https://github.com/mmcdermott/LINCS\\_Deep\\_Learning\\_Benchmarks](https://github.com/mmcdermott/LINCS_Deep_Learning_Benchmarks)

<sup>5</sup>The *F1* score on a binary classifier is the harmonic mean of the classifier’s precision and recall. The *macro-F1* score is an unweighted average of the F1 score of each class separately. Gener-

significant class imbalance. We chose these two metrics to offer, first, a comparatively understandable metric (accuracy) which allows for a clear baseline measure (majority class performance) but is often overly forgiving for tasks with large class imbalance, and second, a less overt, but still commonly used, metric which compensates for class imbalance. We chose not to use AUC as it is less immediately understandable than accuracy while also not accounting for class imbalance as directly as macro-F1, and to avoid having too many evaluation metrics and thereby diluting our comparisons. 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 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 cause them to outperform 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,

---

ally, *macro-F1* will offer a more conservative measure of performance for tasks with strong class imbalance.

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.<sup>6</sup>

---

<sup>6</sup>See [https://github.com/mmcdermott/LINCS\\_Deep\\_Learning\\_Benchmarks](https://github.com/mmcdermott/LINCS_Deep_Learning_Benchmarks)



# Chapter 3

## Results & Discussion

### 3.1 LINCS Corpus

#### 3.1.1 Full Corpus

Final results are shown in Table 3.1. 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. Observed differences between mean performance of any pair of classifiers were statistically significant ( $p \leq 0.05$ ).

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. Our hyperparameter search method also changed the distance metric underlying the KNN method. Across all tasks and datasets, the optimal distance metric was the “Canberra” distance, defined via

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

Using this distance metric induced performance gains over correlative and traditional, Euclidean distance measures. The Canberra distance is traditionally used for integer valued vectors and we are unsure why it would be preferred here. The KNN algorithm could choose uniformly between one of four classes of distance metrics, independently across each of our six dataset/task combinations. This means that if the choice of distance were independent of ultimate performance, the repeated use of the Canberra distance as our highest performing

Table 3.1: Performance (mean  $\pm$  standard deviation) for the full, tissue-heterogenous LINCS corpus.

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



distance metric would be expected with somewhere between 0.02% chance (presuming the ideal choice of parameters is independent from which dataset/task is under consideration) and 25% chance (presuming the ideal choice of parameters shares is totally deterministic across datasets/tasks). We have performed no deeper analyses to determine if this apparent distance metric preference is statistically significant, or to investigate why it might be so.

Linear classifiers robustly performed well. On the MOA task, hyperparameter search selected a logistic regression model (via the log loss in `scikit-learn`), whereas on the Subtype and Primary Site tasks, the optimal setting used a `modified_huber` loss, which is a smooth loss that is tolerant to outliers.

Random forests 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 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. Some might postulate that this is perhaps due to a poor search space of some critical hyperparameters; we intentionally ensured our hyperparameter search space was very broad, especially over these critical parameters. For number of trees, we searched over an equal mixture of Poisson distributions centered at 50, 200, and 400, respectively, and the optimal hyperparameters (shown in the appendix) showed a mix over this entire range. All regularization parameters were also included in our search space.

### 3.1.2 Prostate Only Corpus

Final results for prediction of prostate MOA are shown in Table 3.2. All classifier comparisons were statistically significant ( $p = 0.05$ ). Here, FF-ANNs perform best, though GCNNs are quite competitive. Note that GCNNs still preferred tissue non-

Table 3.2: Performance (mean  $\pm$  standard deviation) on the prostate LINCS corpus and MOA prediction task.

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

specific regulatory graphs, rather than prostate specific graphs. 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.

As indicated in Section 2.2.1, we tested both tissue-specific and tissue-independent regulatory graphs. Surprisingly, on the prostate corpus, the GCNN performed better using the tissue independent regulatory network than it did using the prostate specific regulatory graph. This may indicate that our tissue-specific graphs suffer from some unknown problem, or that tissue-independent graphs simply perform better overall.

Similar to the full system MOA task, the optimal linear model here was a logistic regression model.

## 3.2 MGH NeuroBank Corpus

### 3.2.1 Raw Performance Results

Final results for perturbation identification on the MGH NeuroBank corpus are shown in Table 3.3. 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.

Table 3.3: Performance (mean  $\pm$  standard deviation) on the perturbagen identity task on the MGH NeuroBank Corpus.

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

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 2.1). 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.

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.

### 3.2.2 Generalization Experiments

We also used the MGH NeuroBank Corpus to assess population level generalizability, by training on four of our subjects and testing on the fifth subject. As the MGH Neu-

roBank Corpus contains only one subject with Bipolar Disorder, we do not ever test on this subject’s data—absent more examples of any subject data in this diagnostic category, we would not expect a classifier to generalize well to this subject. 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 3.4. All methods showed a notable drop in accuracy on unseen subjects, 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 subjects. Note, though, that the MGH NeuroBank corpus only contains 5 total subjects to begin with, so it may be the case that these numbers would improve significantly were we to have even a only marginally larger subject pool. Note that on a dataset like LINCS, which is much larger and thus more amenable to higher-capacity learning yet has relatively fewer cellular sources (and with those cellular sources often differing by tissue type or primary diagnosis no less), it is reasonable to imagine that this observed population specific overfitting could foreseeably be even worse than what we observe on the MGH dataset—this point is critical given that this dataset has been used historically for many machine learning investigations with clinically generalizable aspirations, unlike our work where the tasks are designed to aid primarily in method development.

Table 3.4: Per (Non-BD) Subject Generalization Accuracy (mean  $\pm$  standard deviation) on the MGH NeuroBank Corpus.

Classifier Name	Accuracy	Macro F1
GCNN	47.7 $\pm$ 6.78	48.9 $\pm$ 7.40
FF-ANN	<b>48.7 <math>\pm</math> 7.85</b>	<b>50.1 <math>\pm</math> 8.34</b>
KNNs	37.9 $\pm$ 5.39	39.0 $\pm$ 6.68
Linear Classifier	44.1 $\pm$ 4.03	44.7 $\pm$ 4.21
Random Forest	38.8 $\pm$ 5.37	38.3 $\pm$ 6.76
Decision Tree	22.0 $\pm$ 3.85	21.8 $\pm$ 3.59



# Chapter 4

## Conclusion

### 4.1 Summary

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 com-

petitive performance, but feed-forward artificial neural networks continue to perform well, as do linear models.

**Assess population level generalizability** We demonstrate that subject 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 subject.

## 4.2 Future Work

There are several notable directions for future work. First, a notable absent classifier is a self-normalizing neural network (SNN) [26]. Introduced in late 2017, SNNs 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, [29, 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*<sup>1</sup> which has several large databases of protein-protein interactions found experimentally through yeast two-hybrid screening methods [36, 15]. Additionally, incorporating directional information in our regulatory graphs would also enable significantly more nuanced processing. 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 subjects, drugs, or even measurement technologies. More investigation into what drove the success of GCNNs here, perhaps by running dataset size ablation experiments, would also help clarify their strengths. Similarly,

---

<sup>1</sup> <http://interactome.baderlab.org/about/>



more investigations into the failings of random forest models or the relative strengths of differing distance metrics would also be informative.

Table 1: Optimal Hyperparameters for LINCS Full MOA.

GCNN	regularization	1.09e-2
	num_epochs	350
	Fs	[9]
	M	[137, 49]
	Ks	[7]
	batch_size	92
	pool	apool1
	learning_rate	1.23e-3
	decay_steps	405
	decay_rate	9.91e-1
	dropout	6.98e-1
	momentum	8.79e-1
	ps	[2]
	FF-ANN	activation
alpha		1.69
power_t		3.30e-1
learning_rate_init		1.09e-1
hidden_layer_sizes		[955]
learning_rate		adaptive
momentum		8.64e-1
early_stopping		True
nesterovs_momentum		True
KNNs		weights
	metric	canberra
	n_neighbors	12
Linear Classifier	penalty	l1
	l1_ratio	4.06e-1
	alpha	1.23e-3
	loss	log
	n_jobs	-1
	tol	1.00e-5
	learning_rate	invscaling
	eta0	3.17e-4
power_t	1.84e-1	
Random Forest	max_depth	100
	max_leaf_nodes	None
	criterion	gini
	n_estimators	211
	min_samples_split	2
	min_weight_fraction_leaf	1.27e-6
	min_impurity_decrease	1.70e-5
min_samples_leaf	1	
Decision Tree	max_features	None
	criterion	entropy
	max_depth	10
	splitter	best
	min_samples_leaf	2
	min_impurity_decrease	1.23e-3
	min_samples_split	2
max_leaf_nodes	None	
min_weight_fraction_leaf	2.08e-3	

Table 2: Optimal Hyperparameters for LINCFS Full Subtype.

GCNN	regularization	5.42e-3
	num_epochs	300
	pool	mpool1
	M	[150, 150, 14]
	batch_size	88
	Fs	[43]
	momentum	9.73e-1
	learning_rate	2.95e-3
	ps	[2]
	decay_steps	362
	decay_rate	9.76e-1
	Ks	[8]
FF-ANN	dropout	4.54e-1
	activation	relu
	nesterovs_momentum	True
	hidden_layer_sizes	[997]
	learning_rate	invscaling
	early_stopping	False
	learning_rate_init	5.53e-2
	power_t	2.26e-1
	alpha	8.20e-1
	momentum	8.67e-1
KNNs	metric	canberra
	weights	uniform
	n_neighbors	1
Linear Classifier	learning_rate	invscaling
	alpha	3.63e-1
	power_t	1.14e-1
	li_ratio	7.37e-1
	penalty	l2
	eta0	9.91e-5
Random Forest	loss	modified_huber
	criterion	gini
	max_depth	25
	min_samples_leaf	1
	max_leaf_nodes	500
	n_estimators	411
	min_weight_fraction_leaf	4.33e-4
	min_samples_split	2
	min_impurity_decrease	3.64e-5
	Decision Tree	min_impurity_decrease
criterion		entropy
min_weight_fraction_leaf		1.12e-2
max_leaf_nodes		100
min_samples_leaf		1
min_samples_split		2
max_depth		5
splitter		best
max_features	None	

Table 3: Optimal Hyperparameters for LINC5 Full Primary Site.

GCNN	regularization	3.08e-3
	num_epochs	350
	Fs	[[41]]
	batch_size	68
	M	[135, 12]
	ps	[[2]]
	decay_steps	380
	momentum	9.45e-1
	learning_rate	3.13e-3
	pool	apool1
	decay_rate	9.89e-1
	Ks	[[5]]
	dropout	5.62e-1
FF-ANN	learning_rate_init	5.53e-2
	activation	relu
	momentum	8.67e-1
	nesterovs_momentum	True
	learning_rate	invscaling
	power_t	2.26e-1
	early_stopping	False
	hidden_layer_sizes	[997]
	alpha	8.20e-1
	KNNs	n_neighbors
metric		canberra
weights		uniform
Linear Classifier	learning_rate	invscaling
	l1_ratio	7.85e-1
	power_t	8.51e-2
	loss	modified_huber
	penalty	l2
	eta0	2.99e-6
	alpha	4.94e-1
Random Forest	criterion	entropy
	max_depth	100
	min_samples_leaf	1
	min_weight_fraction_leaf	3.87e-4
	n_estimators	401
	max_leaf_nodes	None
	min_samples_split	4
	min_impurity_decrease	2.89e-4
	max_depth	100
Decision Tree	min_samples_leaf	1
	min_weight_fraction_leaf	2.53e-3
	min_samples_split	2
	criterion	gini
	min_impurity_decrease	7.73e-5
	max_features	250
	splitter	best
max_leaf_nodes	None	

Table 4: Optimal Hyperparameters for LINCS Prostate Only MOA.

GCNN	regularization	4.00e-3
	num_epochs	200
	Fs	[25]
	batch_size	55
	M	[168, 14, 9]
	Ks	[15]
	ps	[2]
	pool	mpool1
	learning_rate	5.00e-3
	decay_steps	415
	decay_rate	9.50e-1
	momentum	9.70e-1
	dropout	5.00e-1
FF-ANN	learning_rate	invscaling
	nesterovs_momentum	True
	hidden_layer_sizes	[997]
	learning_rate_init	5.53e-2
	momentum	8.67e-1
	early_stopping	False
	alpha	8.20e-1
	power_t	2.26e-1
	activation	relu
KNNs	metric	canberra
	weights	distance
Linear Classifier	n_neighbors	13
	eta0	3.17e-4
	ll_ratio	4.06e-1
	tol	1.00e-5
	penalty	l1
	learning_rate	invscaling
	alpha	1.23e-3
	n_jobs	-1
	power_t	1.84e-1
	loss	log
Random Forest	min_samples_split	2
	criterion	entropy
	min_weight_fraction_leaf	6.01e-5
	min_samples_leaf	2
	max_depth	None
	min_impurity_decrease	3.68e-4
	max_leaf_nodes	None
n_estimators	53	
Decision Tree	min_impurity_decrease	1.46e-3
	min_samples_leaf	1
	min_weight_fraction_leaf	1.81e-4
	max_depth	25
	max_features	250
	criterion	entropy
	max_leaf_nodes	None
	min_samples_split	2
splitter	best	

Table 5: Optimal Hyperparameters for MGH NeuroBank Corpus Level 4.

GCNN	regularization	2.65e-2
	decay_steps	410
	learning_rate	1.01e-2
	pool	mpool1
	momentum	8.14e-1
	num_epochs	350
	batch_size	25
	M	[138, 60]
	ps	[[2]]
	decay_rate	9.98e-1
	Ks	[[26]]
	Fs	[[31]]
	dropout	6.22e-1
FF-ANN	hidden_layer_sizes	[976]
	alpha	1.16
	power_t	3.21e-1
	activation	relu
	learning_rate_init	4.05e-1
	early_stopping	False
	momentum	9.07e-1
	tol	1.00e-5
	nesterovs_momentum	True
	learning_rate	invscaling
	n_neighbors	6
KNNs	metric	canberra
	weights	distance
Linear Classifier	learning_rate	invscaling
	tol	1.00e-5
	n_jobs	-1
	power_t	1.84e-1
	penalty	l1
	eta0	3.17e-4
	loss	log
	l1_ratio	4.06e-1
	alpha	1.23e-3
Random Forest	max_depth	25
	max_leaf_nodes	500
	min_weight_fraction_leaf	4.33e-4
	min_samples_split	2
	min_samples_leaf	1
	n_estimators	411
	criterion	gini
Decision Tree	min_impurity_decrease	3.64e-5
	min_samples_split	2
	max_leaf_nodes	None
	criterion	gini
	min_impurity_decrease	7.73e-5
	min_weight_fraction_leaf	2.53e-3
	min_samples_leaf	1
	max_features	250
max_depth	100	
splitter	best	

Table 6: Optimal Hyperparameters for MGH NeuroBank Corpus Level 5.

GCNN	regularization	5.00e-2
	decay_steps	400
	learning_rate	1.00e-3
	pool	apool1
	momentum	9.00e-1
	num_epochs	350
	batch_size	20
	M	[100, 60]
	ps	[[2]]
	decay_rate	9.60e-1
	Xs	[[7]]
	Fs	[[25]]
	dropout	5.00e-1
FF-ANN	hidden_layer_sizes	[946, 193]
	alpha	1.11
	power_t	8.87e-1
	early_stopping	False
	learning_rate_init	9.86e-1
	nesterovs_momentum	True
	learning_rate	constant
	momentum	8.76e-1
	activation	relu
	n_neighbors	7
KNNs	metric	canberra
	weights	distance
Linear Classifier	learning_rate	invscaling
	tol	1.00e-5
	n_jobs	-1
	power_t	1.84e-1
	penalty	l1
	eta0	3.17e-4
	loss	log
	l1_ratio	4.06e-1
	alpha	1.23e-3
Random Forest	max_depth	25
	max_leaf_nodes	500
	min_weight_fraction_leaf	4.33e-4
	min_samples_split	2
	min_samples_leaf	1
	n_estimators	411
	criterion	gini
Decision Tree	min_impurity_decrease	3.64e-5
	min_samples_split	2
	max_depth	10
	criterion	entropy
	min_impurity_decrease	1.23e-3
	max_leaf_nodes	None
	min_weight_fraction_leaf	2.08e-3
	min_samples_leaf	2
	max_features	None
splitter	best	





# Bibliography

- [1] Alexander Aliper, Sergey Plis, Artem Artemov, Alvaro Ulloa, Polina Mamoshina, and Alex Zhavoronkov. Deep Learning Applications for Predicting Pharmacological Properties of Drugs and Drug Repurposing Using Transcriptomic Data. *Molecular Pharmaceutics*, 13(7):2524–2530, July 2016.
- [2] David B Allison, Xiangqin Cui, Grier P Page, and Mahyar Sabripour. Microarray data analysis: from disarray to consolidation and consensus. *Nature Reviews. Genetics*, 7(1):55–65, January 2006.
- [3] James Bergstra and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.
- [4] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond Euclidean data. *arXiv:1611.08097 [cs]*, November 2016. arXiv: 1611.08097.
- [5] Renzhi Cao, Colton Freitas, Leong Chan, Miao Sun, Haiqing Jiang, and Zhangxin Chen. ProLanGO: Protein Function Prediction Using Neural Machine Translation Based on a Recurrent Neural Network. *Molecules*, 22(10):1732, October 2017.
- [6] Lujia Chen, Chunhui Cai, Vicky Chen, and Xinghua Lu. Learning a hierarchical representation of the yeast transcriptomic machinery using an autoencoder model. *BMC Bioinformatics*, 17(1):S9, January 2016.
- [7] Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, and Xiaohui Xie. Gene expression inference with deep learning. *Bioinformatics*, 32(12):1832–1839, June 2016.
- [8] Neil R. Clark, Kevin S. Hu, Axel S. Feldmann, Yan Kou, Edward Y. Chen, Qiaonan Duan, and Avi Ma’ayan. The characteristic direction: a geometrical approach to identify differentially expressed genes. *BMC Bioinformatics*, 15:79, March 2014.
- [9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3844–3852. Curran Associates, Inc., 2016.

- [10] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11), 2007.
- [11] domdomegg. A simple diagram of an unspecialised animal cell without labels, January 2016.
- [12] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2224–2232. Curran Associates, Inc., 2015.
- [13] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, February 2017.
- [14] Tracey M. Filzen, Peter S. Kutchukian, Jeffrey D. Hermes, Jing Li, and Matthew Tudor. Representing high throughput expression profiles via perturbation barcodes reveals compound targets. *PLOS Computational Biology*, 13(2):e1005335, February 2017.
- [15] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein Interface Prediction using Graph Convolutional Networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6533–6542. Curran Associates, Inc., 2017.
- [16] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional Sequence to Sequence Learning. *arXiv:1705.03122 [cs]*, May 2017. arXiv: 1705.03122.
- [17] Dave Gershgorn. The data that transformed AI research—and possibly the world, July 2017.
- [18] 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. Understanding multicellular function and disease with human tissue-specific networks. *Nature Genetics*, 47(6):569–576, June 2015.
- [19] Varun Gulshan, Lily Peng, Marc Coram, Martin C. Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, Ramasamy Kim, Rajiv Raman, Philip C. Nelson, Jessica L. Mega, and Dale R. Webster. Development and Validation of a Deep

- Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA*, 316(22):2402–2410, December 2016.
- [20] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1025–1035. Curran Associates, Inc., 2017.
- [21] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep Convolutional Networks on Graph-Structured Data. *arXiv:1506.05163 [cs]*, June 2015. arXiv: 1506.05163.
- [22] 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. Cell-specific prediction and application of drug-induced gene expression profiles. In *Biocomputing 2018*, pages 32–43. WORLD SCIENTIFIC, October 2017.
- [23] Victoria Hore, Ana Viñuela, Alfonso Buil, Julian Knight, Mark I. McCarthy, Kerrin Small, and Jonathan Marchini. Tensor decomposition for multiple-tissue gene expression experiments. *Nature Genetics*, 48(9):1094–1100, September 2016.
- [24] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv:1710.10196 [cs, stat]*, October 2017. arXiv: 1710.10196.
- [25] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907 [cs, stat]*, September 2016. arXiv: 1609.02907.
- [26] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-Normalizing Neural Networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 971–980. Curran Associates, Inc., 2017.
- [27] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics.
- [28] Yaniv Leviathan and Yossi Matias. Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone, May 2018.
- [29] Ron Levie, Federico Monti, Xavier Bresson, and Michael M. Bronstein. CayleyNets: Graph Convolutional Neural Networks with Complex Rational Spectral Filters. *arXiv:1705.07664 [cs]*, May 2017. arXiv: 1705.07664.

- [30] Yu Li, Sheng Wang, Ramzan Umarov, Bingqing Xie, Ming Fan, Lihua Li, Xin Gao, and John Hancock. DEEPre: sequence-based enzyme EC number prediction by deep learning. *Bioinformatics*, 34(5):760–769, March 2018.
- [31] Chieh Lin, Siddhartha Jain, Hannah Kim, and Ziv Bar-Joseph. Using neural networks for reducing the dimensions of single-cell RNA-Seq data. *Nucleic Acids Research*, 45(17):e156–e156, September 2017.
- [32] Zhi-Ping Liu, Canglin Wu, Hongyu Miao, and Hulin Wu. RegNetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. *Database*, 2015, January 2015.
- [33] Matthew B. A. McDermott, Tom Yan, Tristan Naumann, Nathan Hunt, Harini Suresh, Peter Szolovits, and Marzyeh Ghassemi. Semi-Supervised Biomedical Translation With Cycle Wasserstein Regression GANs. In *Thirty-Second AAAI Conference on Artificial Intelligence*, April 2018.
- [34] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. *arXiv:1609.03499 [cs]*, September 2016. arXiv: 1609.03499.
- [35] 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. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, October 2011.
- [36] 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, Xiping 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 Rombout, John Rasla, Patrick Reichert, Viviana Romero, Elieen Ruysinck, Julie M. Sahalie, Annemarie Scholz, Akash A. Shah, Amitabh Sharma, Yun Shen, Kerstin Spirohn, Stanley Tam, Alexander O. Tejada, 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. A Proteome-Scale Map of the Human Interactome Network. *Cell*, 159(5):1212–1226, November 2014.

- [37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, December 2015.
- [38] Thomas Shafee. 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.
- [39] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style Transfer from Non-Parallel Text by Cross-Alignment. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6830–6841. Curran Associates, Inc., 2017.
- [40] 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. A Next Generation Connectivity Map: L1000 Platform And The First 1,000,000 Profiles. *bioRxiv*, page 136168, May 2017.
- [41] 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. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550, October 2005.
- [42] Ngoc Hieu Tran, Xianglilan Zhang, Lei Xin, Baozhen Shan, and Ming Li. De novo peptide sequencing by deep learning. *Proceedings of the National Academy of Sciences*, 114(31):8247–8252, August 2017.
- [43] Sheng Wang, Jian Peng, Jianzhu Ma, and Jinbo Xu. Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields. *Scientific Reports*, 6:18962, January 2016.

- [44] Zichen Wang, Neil R. Clark, and Avi Ma'ayan. Drug-induced adverse events prediction with the LINCS L1000 data. *Bioinformatics*, 32(15):2338–2345, August 2016.
- [45] Jian Zhou and Olga G. Troyanskaya. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods*, 12(10):931–934, October 2015.