Target Predictions using LINCS Data

Yan Xia
Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
yxia1@andrew.cmu.edu

Abstract

Identifying the binding targets of small molecules is an essential process in drug discovery and development. The two conventional approaches include high throughput screening (HTS) and computational structural docking. HTS suffers from its expensive cost and time-consuming procedure, while the computational methods reply on simplifying assumptions that often leads to less accurate results. In this project, we developed machine learning based approaches to efficiently predict drug targets using the massive LINCS data. We extracted meaningful features from the LINCS data and integrated them with information from other genomic data, and build a random forest based classifier that achieves remarkable prediction accuracy. Our strategy provide an fast and efficient way of predicting drug targets, and can naturally serve as a pre-pruning step for the computationally expensive structural based approaches.

1 INTRODUCTION

A major computational and experimental challenge is mapping small molecules to their protein targets. To date, most studies have relied on high throughput screening (HTS), *i.e.*, testing millions of compounds against a single target [2, 9]. While this expensive and time-consuming modality can sometimes be effective at identifying active compounds in vitro, only a tiny fraction of the 100,000 predicted protein-protein interactions (PPIs), which are the direct or indirect targets of most drugs, have been disrupted in HTS experiments. More importantly, from the point of view of drug development, most *in vitro* assays do not provide any context regarding drug activity in the cell. Another common strategy is computational structural docking, in which the small molecules are docked onto multiple positions on the molecular structure of a protein, and the fitness of interaction are estimated through complex biophysical computations and molecular dynamic simulation [7, 13]. While the computational methods are generally less costly and easier than HTS, to be accurate these methods require extensive computational time for each pair of drug-protein being tested and thus do not scale to the large number of possible interactions that needs to be studied. Specifically, it is not feasible to perform extensive and complete docking for all the genes in human genome.

In this project, we aim to use the Library of Integrated Cellular Signatures (LINCS)[1] to predict the drug targets through machine learning approaches. LINCS currently contains gene expression profiles following knockdowns (KD) and treatments in multiple cell lines. Our prediction strategy follows the following key hypothesis of this project: *the gene expression pattern of treating cells with a given drug is similar to that of knocking down the drug targets in the same cell line*. The rationale (depicted in Figure 1) is that most drugs bind to their targets and inhibit the targets' cellular

[1] http://www.lincsproject.org/

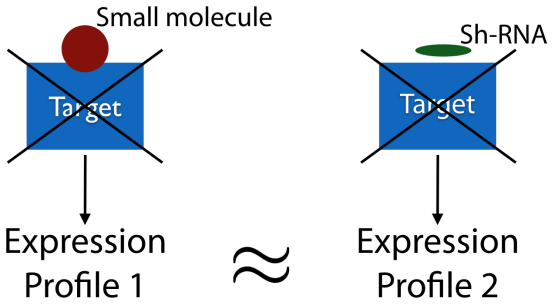

Figure 1: The hypothesis of this project: The effect of inhibiting a target by a small molecule is similar to knocking down the same target with sh-RNA.

functioning. The effect of drug-binding, therefore, should be similar to simply knocking down the cellular level of that target. For this reason, proteins that show high knockdown correlation with the drug likely lie on a pathway that is directly affected by the drug and would thus serve as good candidates for direct targets for the drug. This approach provides a valuable alternative to *in vitro* screening. By focusing on the pathways targeted by different drugs and small molecules using the LINCS data and knockdown gene expression profiles, we should be able to predict a more biologically relevant set of targets for each small molecule as well as potential off-target interactions.

A key challenge when analyzing these massive datasets is to integrate results across experiments and cell types and to further integrate the data with additional data sources. Therefore, determining drug targets from the LINCS data is not trivial. First, the data is noisy and so several false positives and false negatives can arise in each experiment. Moreover, most drugs are not intended to be active for most cell types and so it is not even clear if the intended target(s) are active in the cells that are being profiled. Even when the cell is correct and the results are accurate, several indirect targets may appear to be affected (for example, those upstream/downstream of the immediate target) making it hard to determine which of the differentially expressed genes proteins are the direct targets of the drug. Finally, drugs often act at the protein level and so can have little impact on the direct readouts (mRNAs) of their targets making it hard to identify such targets using gene expression data.

Rather than directly focusing on genes that are differentially expressed (DE) following drug treatment (which, as mentioned above is not likely to lead to accurate set of targets), our hypothesis leads to an indirect approach to match small molecules and their protein targets. Specifically, we examine the set of DE genes following drug treatment and a similar set following knockdown of a specific protein and compare them in order to identify a target for the molecule. In addition, we would combine the condition and cell type specific LINCS data with other genomics data (mainly protein-protein interaction and localization data) to further improve the set of predicted targets across cells.

The final prediction strategy of this project is based on random forest [8, 12], which is especially suitable for dealing with missing data. In our case, while we have KD and treatment experiments for many genes and molecules across several cells, relatively few genes and molecules have been tested in all cells. Thus, to train a classifier for predicting targets we would need to develop methods that can handle missing data in the classification process and still yield accurate results.

Our strategy can easily be combined with the computational docking methods. The correct targets can be enriched to the top 100 genes using our methods, this means that the computational docking methods only need to focus on these 100 genes instead of all genes in human genome. By focusing on less genes, we can perform much more extensive docking for each gene and can most likely obtain a even more enriched list (e.g. top 10), which can easily be tested by molecular biologist using biological assays.

2 DATA AND METHODS

2.1 Data Sources

LINCS LINC is an NIH program that generates gene expression profiles across multiple cell lines and perturbational types at a massive scale. To date, LINCS has generated over 1 billion data points of gene expression profiles (over 150 gigabytes of data) containing small-molecules and genetic gain- (cDNA) and loss-of-function (sh-RNA) constructs across multiple cell types.

Specifically, the dataset contains experiments profiling the effects of 20,143 small-molecule compounds (including known drugs and pathway-specific tool compounds). In addition, there are 22,119 genetic constructs for over-expressing genes (gain-of-function) or knocking-down (KD) genes (loss-of-function). These constructs were designed to affect genes encoding targets of FDA-approved drugs, drug-target pathway members, and targets associated with disease. These perturbing agents are tested on 18 different cell types, which were selected from diverse lineages which span established cancer cell lines, immortalized (but not transformed) primary cells and both cycling and quiescent cells.

The gene expression profiles were measured using the L1000 assay[2]. This is a bead-based assay in which the raw fluorescence transcriptional responses corresponding to certain cell-perturbagen combinations are measured. In order to increase the throughput of profiling, this assay does not directly measure all of the ˜20,000 genes in the human genome. Instead, it measures a set of 978 so-called "landmark genes" and the expression values of other genes were computationally imputed from them. This reduced representation is possible because of the high correlation of gene expression. The landmark genes are carefully chosen to be minimally redundant and can capture approximately 80% of the information.

The raw data collected from L1000 assay were processed through a 4-stage computational pipeline which converts raw fluorescence intensity into differential gene expression signatures:

- Level 1: Raw, unprocessed flow cytometry data.
- Level 2: Gene expression values per 1,000 genes after de-convolution.
- Level 3: Gene expression profiles of both directly measured landmark transcripts plus imputed genes. Normalized using invariant set scaling followed by quantile normalization.
- Level 4: Signatures with differentially expressed genes computed by robust z-scores for each profile relative to population control.

All data of all 4 stages are available in LINCS, and we use the level-4 signature values in this project. The data processing of LINCS was done using l1ktool[3].m

ChEMBL ChEMBL is an open large-scale bioactivity database [6]. We retrieved the records of all FDA-approved drugs using ChEMBL web service API[4]. These records contain the designed targets for the drugs and the synonyms and unique chemical ID for them. Using these information, we can cross reference these drugs in LINCS.

Protein-Protein Interaction BioGRID[3] and HPRD[11] curated set of physical and genetic interactions including interactions, chemical associations, and post-translational modifications from publications. We retrieve all the records corresponding to protein-protein interactions(PPI) from these data sources and converted the obtained PPI to adjacency list representation.

Gene Ontology We obtained the cellular localization of genes from the Gene Ontology Consortium [4]. The Gene Ontology (GO) project is a collaborative effort to address the need for consistent

[2] `http://support.lincscloud.org/hc/en-us/sections/200437157-L1000-Assay`
[3] `http://code.lincscloud.org/`
[4] `https://www.ebi.ac.uk/chembl//ws/home_old`

Localization	Assigment
Cell Membrane	External
Endosome	Internal
Secreted	External
Cytoplasm	Internal
Nucleus	External
Chromosomes	Internal
Mitochondria	Internal
ER	Internal
Lysosome	Internal
Golgi	Internal
Peroxisome	Internal
Ribosome	Internal
Microsome	Internal
Endomembrane	Internal
Cytoskeleton	Internal
Centrosome	Internal
Vesicle	External
Vacuole	External
Membrane	External
Cell	Internal

Table 1: Possible cellular localizations retrieved from GO and their assignment.

descriptions of gene products across databases. The GO database provides web services to query genes in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner[5]. We further assign the locations as either "intracellular" (inside of cell) and "extracellular" (outside of cell). The detailed assignment can be found in Table 1.

2.2 Naming Conventions and Notations

A target gene is transcribed and translated to protein, which the drugs bind to. Therefore, "target" in the text may refer to either a gene or a protein depending on the context.

- The LINCS signature values measure the change of genes. In this context, targets refer to genes.
- Knockdown experiment decrease the expression of target genes and reduce the amount of target proteins in cell.
- The drug binds to target proteins.

Before we describe the procedure of constructing the validation dataset and features, we first lay out the symbols and notations used in the later text here.

[5] `http://geneontology.org/page/go-enrichment-analysis`

Symbol	Meaning
d	Index for a drug
c	Index for a cell line
g	Index for a gene
N_D	Total number of drugs
N_C	Total number of cell lines
C_d	The set of cell line indices for drug d
P_d	The set of protein target indices for drug d
G_c	The set of knockdown gene indices for cell line c
N_{dc}	Number of experiments for applying drug d to cell line c
N_{gc}	Number of experiments for knocking out gene g in cell line c
$\mathbf{\Delta}$	Drug-response data
$\mathbf{\Gamma}$	Gene-knockdown data
$\mathbf{\Psi}$	Control data
$\mathbf{\Omega}$	Full feature data
$\mathbf{X}_d$	Training data derived from drug d
$\mathbf{y}_d$	Training label derived from drug d
$\boldsymbol{\nu}_d$	Negative (non-target) genes for drug d

Table 2: Symbols and notations used in the project

2.3 Building a Validation Dataset from LINCS

2.3.1 Choosing Small Molecules and Cell Lines

To examine our main hypothesis, we build a validation dataset that includes small molecule and KD experiments that satisfies the following:

- The cellular target of a small molecule is known *a priori*.
- A small molecule has been applied to multiple cell lines.
- The knockdown of the target gene are also available for the same cell lines.

Specifically, this validation dataset includes LINCS experiments corresponding to the following small-molecule response and gene knockdown experiments.

Small molecules Our hypothesis requires that we know the correct targets for each drug *a priori* in order to evaluate the predictions from our methods. Therefore, we include only FDA-approved drugs in the validation dataset because their intended targets were well established and documented.

We retrieved the reported targets and other meta-information of all FDA-approved drugs using the ChEMBL, and then cross-referenced these drugs in LINCS using their primary product names, synonyms, canonical SMILES strings and standard InChIKey. We have identified 1031 out of around 1300 FDA-approved drugs tested in LINCS.

Cell lines We need to extract signature values of both small molecule and gene knockdown experiments from different cell lines. Our hypothesis requires that a cell line has (1) knockdown experiments for many genes, and (2) the targets of drugs are in the set of knocked-down genes. Guided by these two requirements, we queried the meta-information of LINCS signatures and selected seven cell lines to be included in the validation dataset and their information are shown in Table 3.

Cell line	Drug	Knockdown	Control
A549	188	11947	52
MCF7	180	12031	54
VCAP	175	13225	56
HA1E	172	11968	53
A375	143	11696	58
HCC515	129	7828	52
HT29	96	10185	52

Table 3: Seven Cell lines are included in the validation dataset. The number of drugs, knockdown genes and control experiment are shown. For a given cell line, we only include drugs that have their target knockdown experiments available in that cell line.

Not all of the drugs are applied to all the cell lines in LINCS. Therefore, we only include drugs that have been applied to 4 or more cell lines, and finalized 152 drugs for the validation cell lines (Table 4). There are 29 drugs in the validation dataset that have been applied to all 7 cell lines. We used these drugs to evaluate the predictive power of individual features, which is further discussed below.

7 Cell lines	6 Cell lines	5 Cell lines	4 Cell lines	Total
29	30	42	51	152

Table 4: The number of drugs for combinations of cell lines included in the validation dataset

2.3.2 Extracting Experiments from LINCS

After determining the subsets of small molecules and cell lines, we obtained the associated experiment identifiers known as "distil ID" from LINCS meta-information. We included only the reproducible distil IDs known as "Gold" IDs.

We then extracted the corresponding signature values from LINCS using the L1000 Analysis Tools (l1ktools) [6]. We chose only to extract the signature values of 978 "landmark" genes, because their expression were directly measured, and the values of other genes were imputed from the data of these landmark genes.

Drug-response experiments There exists multiple experiments (distil IDs) corresponding to a combination of drug d and cell line c (applying drug d to cell line c). Denote the N_{dc} as the number of experiments for the combination d, c. We extracted a matrix of signature values of size $978 \times N_{dc}$ (number of landmark genes $\times$ number of experiments) per combination. We next take the median of signature values across different experiments, and obtain a 987×1 signature vector per combination. The overall drug-response data $\mathbf{\Delta}$, therefore, is implemented as a MATLAB structure with $D = 152$ entries, each containing the following fields.

$$\begin{aligned} \texttt{name:} &\quad \texttt{PertID}_d \text{ (string)} \\ \texttt{cells:} &\quad \texttt{Cells}_{C_d} \text{ ($|C_d| \times 1$ string array)} \\ \texttt{signature:} &\quad \Delta_{d\cdot\cdot} \text{ ($978 \times |C_d|$)} \end{aligned}$$

where $\texttt{PertID}_d$ is the unique internal identifier of a small molecule in LINCS. $\Delta_{d\cdot\cdot}$ contains the expression values of drug d across C_d different cell lines. The $\texttt{Cells}_{C_d}$ field contains cell line names corresponding to the column of $\mathbf{\Delta}_{d\cdot\cdot}$.

Gene knockdown experiments We follow the similar protocol to extract the signature values of gene knockdown experiments. Denote N_{gc} as the number of experiments for the combination of gene g and cell line c (knocking down gene g in cell line c). Then, for each combination of g and c we extracted signature values of size $978 \times N_{gc}$. After taking the medians across different experiments, we obtain a 978×1 vector per combination. The

[6] https://github.com/cmap/l1ktools

overall gene knockdown data $\mathbf{\Gamma}$ has $C = 7$ entries and each entry contains the following fields:

$$\begin{aligned} \texttt{name:} \quad & \texttt{Cells}_c \text{ (string)} \\ \texttt{genes:} \quad & \texttt{Symbols}_{G_c} \text{ ($|G_c| \times 1$ string array)} \\ \texttt{signature:} \quad & \mathbf{\Gamma}_{c\cdot\cdot} \text{ ($978 \times |G_c|$)} \end{aligned}$$

where $\texttt{Cells}_c$ is the name of a cell indexed by c. $\mathbf{\Gamma}_{c\cdot\cdot}$ contains the signature values of the knockdown of genes in cell line c. The $\texttt{Symbols}_{G_c}$ field is a subset of gene symbols corresponding to the column identifiers of $\mathbf{\Gamma}_{c\cdot\cdot}$ under the HGNC naming scheme.

Control experiments We also extracted the signatures of control experiments. The signature values for each cell line were extracted and we obtained a 978×1 vector after taking the medians. We denote the overall control experiment data as $\mathbf{\Psi}$. $\mathbf{\Psi}$ is of size $978 \times C$ and implemented with the following format:

$$\begin{aligned} \texttt{name:} \quad & \texttt{Cells}_c \text{ (string)} \\ \texttt{control:} \quad & \mathbf{\Psi}_{\cdot c} \text{ (978×1)} \end{aligned}$$

where $\mathbf{\Psi}_{\cdot c}$ is the signature column vector for a cell line c.

2.4 Extracting and Integrating Features from Different Data Sources

2.4.1 Correlation feature

The correlation feature, denoted as f_{cor} is constructed as follows:

For each drug d in $\mathbf{\Delta}$

- Denote T_d as the intersection of gene symbol indices for cells in C_d. *i.e.*
$$T_d = \bigcap_{c \in C_d} G_c$$
- Obtain the knock-down signature values of T_d from $\mathbf{\Gamma}$. Denote this data matrix as $\mathbf{\Gamma}_{C_d \cdot T_d}$ which is of size $|C_d| \times 978 \times |T_d|$, where for each cell line in C_d there is a signature matrix of size $978 \times |T_d|$.
- Compute the Pearson's correlation between $\mathbf{\Delta}_{d\cdot\cdot}$ ($978 \times |C_d|$) and $\mathbf{\Gamma}_{C_d \cdot T_d}$ ($|C_d| \times 978 \times |T_d|$). Specifically, for each cell line $c \in C_d$, we compute the correlation between $\mathbf{\Delta}_{d\cdot c}$ and $\mathbf{\Gamma}_{c \cdot T_d}$, and obtain a correlation vector of size $|T_d|$. This is the correlation between the responses of the cells to the drug treatment and their response to the gene KD. Each entry in this vector is the correlation of 978 landmark genes of the drug d in one cell line ($\mathbf{\Delta}_{d\cdot c}$) and a knockdown of gene g in the same cell line ($\mathbf{\Gamma}_{c\cdot g}$). In other words, if we collect these correlation vectors for all cell lines in C_d and denote the overall correlation feature as f_{cor} has the following definition:
$$f_{cor}(d, g, c) = \text{corr}\left(\mathbf{\Delta}_{d\cdot c}, \mathbf{\Gamma}_{c\cdot g}\right) \quad \forall g \in T_d$$
The correlation feature for one drug d, *i.e.* $f_{cor}(d, \cdot, \cdot)$, has a dimension of $|T_d| \times |C_d|$.

2.4.2 Cell selection feature

The cell selection feature , denoted as f_{CS}, is computed as follows.

- For each drug d in $\mathbf{\Delta}$ ($\mathbf{\Delta}_{d\cdot\cdot}$):
  - For each cell line c in C_d:
    - compute the correlation between $\mathbf{\Delta}_{d\cdot c}$ and $\mathbf{\Psi}_{\cdot c}$
$$f_{CS}(d, c) = \text{corr}\left(\mathbf{\Delta}_{d\cdot c}, \mathbf{\Psi}_{\cdot c}\right)$$

In other words, $f_{CS}(d, \cdot)$ produces a $|C_d| \times 1$ vector, each entry corresponds to the correlation between the drug-response and control experiments for one cell line in C_d. This feature is used to determine the relevance of the drug to the cell type being studied.

2.4.3 PPI correlation score

The "PPI correlation Score", denoted as f_{PC} is constructed as follows:

For each drug d, we first obtain T_d, the intersection of gene symbol indices, as before. Then for each cell line c in C_d, we sort T_d in descending order using the correlation values $f_{cor}(d, \cdot, c)$, and we denote the sorted gene symbol indices for cell line c as $\sigma_c(T_d)$.

We then construct the PPI correlation feature f_{PC} as follows:

- For each knockdown gene g in T_d: Obtain the set of neighbor gene symbol indices from PPI adjacency list, and denote it as B_g.

$$f_{PC}(d, g, c) = \frac{|B_g \cap \sigma_c(T_d)_{1:100}|}{|B_g \cap \sigma_c(T_d)| + 50}$$

- f_{PC} feature is of the same dimension as f_{cor}, which is $|T_d| \times |C_d|$.

In other words, $f_{PC}(d, g, c)$ reflects the fraction of gene g's binding partners that is more correlated with drug d in the context of cell line c. We use 50 as the pseudo-count to penalize hub proteins which have substantially more neighbors than others.

2.4.4 PPI expression score

We compute two types of PPI expression scores, denoted as $f_{PE_{max}}$ and $f_{PE_{avg}}$, as follows:

- For a drug d:
  - For each knockdown gene g in T_d:
    Obtain N_g as above.
    - For each cell line c, find the set of signature values for the neighbors: $\mathbf{\Delta}_{d,N_g,c}$ (size: $|N_g| \times 1$). Then, the two PPI expression scores computed as

$$f_{PE_{max}}(d, g, c) = \max\left(\mathbf{\Delta}_{d,N_g,c}\right)$$
$$f_{PE_{avg}}(d, g, c) = \text{avg}\left(\mathbf{\Delta}_{d,N_g,c}\right)$$

2.4.5 Feature data structure

We combined the features for all drugs in a MATLAB structure $\mathbf{\Omega}$. $\mathbf{\Omega}$ has D entries, and each entry $\mathbf{\Omega}^{(d)}$ has the following fields:

`Name:`	$\texttt{PertID}_d$ (string)
`Targets:`	P_d (targets for d)
`Cells:`	C_d ($\|C_d\| \times 1$ string array)
`Genes:`	T_d (common genes across G_c)
`Correlation:`	$f_{cor}(d, \cdot, \cdot)$ ($\|T_d\| \times \|C_d\|$)
`PPI Correlation:`	$f_{PC}(d, \cdot, \cdot)$ ($\|T_d\| \times \|C_d\|$)
`Max PPI Expression:`	$f_{PE_{max}}(d, \cdot, \cdot)$ ($\|T_d\| \times \|C_d\|$)
`Avg PPI Expression:`	$f_{PE_{avg}}(d, \cdot, \cdot)$ ($\|T_d\| \times \|C_d\|$)
`cell selection:`	$f_{CS}(d, \cdot)$ ($\|C_d\| \times 1$)

There are a total of $D = 152$ drugs in $\mathbf{\Omega}$, and the number of drugs with different $|C_d|$ are summarized in Table 4.

2.5 Methods

2.5.1 Criterion of successful classification

Due to the intrinsic noise from the data, we define a successful classification for a drug if any of its correct targets is enriched into the top K ranked genes, where K can be either 50 or 100.

2.5.2 Single Feature

The evaluation of single features was performed using the drugs that have been applied on all 7 cell lines. There are 29 of these drugs from $\mathbf{\Omega}$. We sort the common genes T_d descendingly for a drug d and a cell line c using an individual feature $f(d, \cdot, c)$, where f is either f_{cor} or f_{PC}. Denote $\sigma_d(g, c)$ as the ranking of a gene $g \in T_d$ in the context of cell line c. Then, we define the overall ranking of a gene $\sigma_d(g)$ to be the best ranking across all seven cell lines, *i.e.* $\sigma_d(g) = \min(\sigma_d(g, c))$ for $c \in C_d$.

2.5.3 Constructing training dataset

Next, we wish to learn and evaluate classifiers that predict drug target using all features from the feature dataset $\mathbf{\Omega}$, . Therefore, we first construct training data (design matrix $\mathbf{X}$ and its associated labels $\mathbf{y}$) from the feature dataset $\mathbf{\Omega}$.

For each drug d in $\mathbf{\Omega}$, we select the rows corresponding to the targets in P_d from the other feature matrices and concatenate them into a row vector. The same cell selection vector is appended to every row of targets. These rows are assigned with a positive label 1. We then randomly sampled 100 non-target genes (denoted as $\boldsymbol{\nu}_d$) and construct the row vectors the same way as the target genes, and these rows are assigned with a negative label 0. In other words, the training matrix and label vector constructed from a drug d are of the following format.

$$
\begin{array}{c} \mathbf{X}_d \\ \\ \begin{bmatrix} f_{cor}(d, P_{d1}, \cdot) & f_{PC}(d, P_{d1}, \cdot) & f_{PE_{max}}(d, P_{d1}, \cdot) & f_{PE_{avg}}(d, P_{d1}, \cdot) & f_{CS}(d, \cdot) \\ f_{cor}(d, P_{d2}, \cdot) & f_{PC}(d, P_{d2}, \cdot) & f_{PE_{max}}(d, P_{d2}, \cdot) & f_{PE_{avg}}(d, P_{d2}, \cdot) & f_{CS}(d, \cdot) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{cor}(d, P_{dm}, \cdot) & f_{PC}(d, P_{dm}, \cdot) & f_{PE_{max}}(d, P_{dm}, \cdot) & f_{PE_{avg}}(d, P_{dm}, \cdot) & f_{CS}(d, \cdot) \\ f_{cor}(d, \boldsymbol{\nu}_{d1}, \cdot) & f_{PC}(d, \boldsymbol{\nu}_{d1}, \cdot) & f_{PE_{max}}(d, \boldsymbol{\nu}_{d1}, \cdot) & f_{PE_{avg}}(d, \boldsymbol{\nu}_{d1}, \cdot) & f_{CS}(d, \cdot) \\ f_{cor}(d, \boldsymbol{\nu}_{d2}, \cdot) & f_{PC}(d, \boldsymbol{\nu}_{d2}, \cdot) & f_{PE_{max}}(d, \boldsymbol{\nu}_{d2}, \cdot) & f_{PE_{avg}}(d, \boldsymbol{\nu}_{d2}, \cdot) & f_{CS}(d, \cdot) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{cor}(d, \boldsymbol{\nu}_{d100}, \cdot) & f_{PC}(d, \boldsymbol{\nu}_{d100}, \cdot) & f_{PE_{max}}(d, \boldsymbol{\nu}_{d100}, \cdot) & f_{PE_{avg}}(d, \boldsymbol{\nu}_{d100}, \cdot) & f_{CS}(d, \cdot) \end{bmatrix} \end{array}
\quad
\begin{array}{c} \mathbf{y}_d \\ \\ \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \end{array}
$$

where $m = |P_d|$, which is the total number of targets for drug d. Therefore, the training matrix $\mathbf{X}_d$ for drug d is of size $(m + 100) \times 5\,|C_d|$, and label vector $\mathbf{y}$ has length $m + 100$.

2.5.4 Logistic Regression and Random Forest

We used MATLAB's `lassoglm` package to train a lasso-regularized logistic regression model. We applied MATLAB's `TreeBagger` package for random forest based models. Leave-One-Out Cross Validation (LOOCV) was performed to evaluate the performance of both methods.

2.5.5 Code Repository

All the relevant code for this project can be found at `https://github.com/xiayan/linc_target_pred`.

3 Results

3.1 Designing Features using the Signature Profiles from LINCS

First, we describe the rationale of each feature extracted from LINCS and other auxiliary genomic datasets. The detailed process of generating these features are included in Data and Methods.

The correlation feature captures the main hypothesis, *i.e.*, the gene expression pattern of treating cells with a given drug is similar to that of knocking down the drug targets in the same cell line. We compute the Pearson's correlation of the signature values between a drug response and a gene knockdown experiment in the same cell lines.

The correlation from some cell lines may be more useful than the others since we do not expect drugs to be effective in all cell lines. The cell selection features are designed capture this cell line differences. We reason that if a drug has more effect on one cell line, the gene expression pattern for that drug and cell line combination should be very different than that of the control experiment. This hypothesis implies that the signature vector of a more effective cell to a drug should have lower correlation with the control vector.

We also build features that take into account of the protein-protein interaction information. In addition to having high correlation with the signatures of the target knockdown experiment, we believe that the signatures of drug-response experiments should also show high correlation with the knockdown experiments of its target's binding partners.

PPI expression score is another PPI-related feature that considers the signature values directly. Specifically, we think that the knockdown of the target can lead to higher changes of the signature values for that target's partners. We compute two types of PPI expression scores.

3.2 Predictive Power of Individual Features

We evaluate the predictive power of individual features. This process allows us to confirm that the features we construct are meaningful, and when combined together, can potentially lead to better predictions.

The correlation feature captures the main hypothesis of this work. To evaluate it, we rank the genes by the correlation feature alone across the cell lines, and select the final rankings to be the best one across all cell lines. From this procedure, it is very possible that multiple genes have the same ranking (e.g., 7 genes can all have 1 as the final rankings, each for a different cell for the same drug) and we break ties randomly. We note that the decision of taking the best ranking is consistent with the assumption that small molecule are not always effective in all of the cell lines. We use the drugs that have been applied to all 7 cell lines ($|C_d| = 7$, 29 out of 152 drugs in $\mathbf{\Omega}$). This choice allows a fair comparison, since no drug has missing features.

The correlation feature correctly enriched the targets of 8 drugs to the top 100 which is top 3% (100 out of 3104 genes). The mean ranking of the best ranked targets of all drugs is 800.59 (Table 5). We compare the performance of correlation feature with a baseline in which the we use random genes as the targets of drugs and repeated the same evaluation procedure. The random experiment only classified two drugs correctly, and has the mean ranking of merely 1365.00 which is much worse than that of using the correlation feature. It is therefore evident from these results that the correlation feature has significantly better predicative power than random.

The cell selection feature builds on top of the correlation feature and enables us to focus on correlations in relevant cells rather than treating all cell lines equally. After applying a drug to different cell lines, we expect the relevant cell lines to demonstrate significantly dissimilar gene expression profiles from those in the untreated state. Therefore, to evaluate the effect of incorporating the cell selection feature, we perform the following steps:(1) for a given drug d, we determine the most relevant cell line $c = \text{argmin}_c f_{CS}(d, c)$, *i.e.*, the cell line has the lowest correlation with the control signature profile; (2) use the ranking of that cell line as the final ranking for genes, instead of taking the best ranking across 7 cell lines. Even though the final ranking for each gene cannot be better than selecting the best from all 7 cell lines, it is possible that the average ranking decreases because

there are no tied rankings in this case. The results shows that including the cell selection feature indeed lead to the decrease of average ranking (776.86), though the number of correctly classified drugs also decreased to 6. Nonetheless, this procedure is designed to evaluate the predicative power of cell selection feature, instead of an actual procedure of performing classification. It is clear from these results that cell selection feature helps to improve the overall performance and it is beneficial to combine it with other features as we will discuss below.

The PPI correlation feature is another feature that utilizes the correlation between KD and drug treatment. It represents our hypothesis that the signature profiles of targets' binding partners also have high correlation with the drug-response signature profile. The evaluation procedure of the PPI correlation feature is similar to that of th correlation feature. We rank the genes using their PPI correlation values and select the best ranking across 7 cell lines as the final ranking for each gene. The result shows that PPI correlation feature along correctly classifies 10 drugs and the mean ranking is further decreased to 724.31 (Table 5).

Overall, the evaluation of individual features reveals that the features that we extracted and integrated from LINCS and the auxiliary datasets have significant predictive powers. This finding motivates us to use them together in single classification models, such as logistic regression and random forest, and it most likely will further improve the classification performance.

3.3 Classification using Logistic regression and Random Forest

In our first attempt, we trained a logistic regression model [12] to utilize all the features in classification. The evaluation is performed using the same 29 drugs that have been applied to all 7 cell lines. To estimate the performace of the model, we used Leave-One-Out Cross Validation (LOOCV) for each drug. Specifically, we train a lasso-regularized logistic regression model using the features from 28 drugs and apply it to compute the probability for each gene being the target given the features in the left-out drug. We then rank the genes using the predicted probability and examine the best ranked target for the held-out drug. Through this process, logistic regression classified 11 out of 29 drugs successfully (Table 5). Furthermore, logistic regression improves the average ranking of all drugs to 712.83.

Logistic regression learns a linear decision boundary and assumes that the examples are independent. This is not generally true in our case since the many drugs target proteins in the common, well-established signaling pathways [10]. For example, GPCR-targeting drugs represent 30 to 40 percent of marketed pharmaceuticals. Therefore, in our second attempt, we used random forest which is able to learn more sophisticated decision boundaries and performs automatic variable selections [5]. We followed the same LOOCV procedure and trained a random forest regressor with 5000 decision trees using features from 28 drugs, and it was then applied to all the genes for held-out drug. Applying random forest regression resulted in much better performance. 16 out of 29 drugs (55%) are now successfully classified, and the average ranking is improved to 471.45 (Table 5).

These results confirm that including all features leads to better classification and demonstrates the superior performance of random forest. Coincidentally, random forest is also especially suitable for dealing with missing features, and this allows us to extend our analysis to all 152 drugs in $\mathbf{\Omega}$.

3.4 Extending Random forests to Drugs with Missing Features

The overall goal of this project is to predict the targets of small molecules that most likely have not been applied to all 7 cell lines, it is highly desirable that our method can handle missing data (*i.e*. cells for which experiments were not performed) robustly so that it is applicable to more small molecules.

To this end, we have developed two methods to deal with different C_d combinations and extended the random forest model to all the drugs in $\mathbf{\Omega}$. In the first method we simply build the random forest "on-the-fly". For a given drug i, we iterate through all other drugs in $\mathbf{\Omega}$ and test if a drug d has a cell line collection that is compatible with that of drug i. In other words, we test if $C_i \subseteq C_d$ and if so we extract the features of corresponding cell lines in C_i from $\mathbf{\Omega}^{(d)}$ and include them in the training

Drug	Random	Cor	CS	PC	LR	RF
vinorelbine	310	126	1318	128	28	88
dexamethasone	1498	1891	943	284	757	157
dasatinib	2325	1009	222	94	182	532
vincristine	1979	473	386	439	456	37
mycophenolate-mofetil	564	1100	2986	1263	3064	3086
amlodipine	995	1338	1801	2439	3037	650
lovastatin	1712	72	2078	811	1334	55
clobetasol	2194	820	157	21	38	65
calcitriol	2514	1059	221	2938	1299	252
flutamide	919	2604	2806	69	702	647
prednisolone	2382	1439	787	206	257	23
nifedipine	940	1225	1285	1465	3037	2249
vemurafenib	1042	1	1	82	22	2
glibenclamide	29	1415	409	2028	1300	366
digoxin	2376	73	118	1470	732	44
bortezomib	1882	1	2	1	24	5
vinblastine	1612	515	100	56	38	2
digitoxin	573	89	216	430	79	50
losartan	645	489	770	988	735	1931
pitavastatin	1855	1976	1117	1036	1632	373
digoxin	69	521	194	776	208	64
hydrocortisone	303	312	58	72	29	17
paclitaxel	2299	74	47	121	79	19
lovastatin	988	1	1587	735	128	100
irinotecan	1742	1023	236	20	46	160
vincristine	1394	96	17	74	28	9
vinblastine	1359	490	1383	75	35	2
raloxifene	2080	2883	1172	1818	1114	2520
digoxin	1005	102	112	1066	252	167
Mean Ranking	1365.0	800.6	776.9	724.3	712.8	471.4
Top 100	2	8	6	10	11	16

Table 5: Performance of different methods on 29 drugs. Cor: correlation feature; CS: cell selection feature; PC: PPI correlation feature; LR: logistic regression; RF: random forest

		All	7 Cells	6 Cells	5 Cells	4 Cells
On-the-fly	Top 100	58	13	15	16	14
	Top 50	42	10	10	12	10
	Top 100 %	38%	45%	50%	38%	27%
	Top 50 %	28%	34%	33%	29%	20%
	Mean Ranking	767.3				
Two-level	Top 100	64	14	15	22	13
	Top 50	54	12	14	20	8
	Top 100%	42%	48%	50%	52%	25%
	Top 50%	36%	41%	47%	48%	16%
	Mean Ranking	718.2				

Table 6: Performance of two random forest models on all drugs. The number of drugs with targets ranked in top 100 and top 50 are shown. These total numbers are broken down to different cell line numbers. The percentage of these successful drugs are also reported.

data. After we include data for all compatible drugs we can use the training data to train and apply a random forest for the given drug i. We note that for any drug in $\mathbf{\Omega}$, there are at least 28 compatible drugs because 29 drugs have been applied to all 7 cell lines. However, the main disadvantage of this method is that we need to train separate random forest for every test drug.

In the second method we build the random forest in two steps and we denote it as the "two-level" random forest. In the first step, we randomly sample 4 cell lines from the total 7 cell lines (denote as C_i). In the second level, we iterate through all drugs in $\mathbf{\Omega}$ and if $C_i \subseteq C_d$ for a drug d, we extract the features corresponding to cell lines in C_i from $\mathbf{\Omega}^{(d)}$ and add them into the training data. After extracting all compatible features from $\mathbf{\Omega}$, we train a decision tree for C_i. We repeat this process for 3500 times, such that each combination of 4 cell lines have around 100 decision trees on average ($\binom{7}{4} = 35$). To apply this two-level random forest to a test drug t with cell line profile C_t, we iterate through these 3500 decision trees and test if $C_i \subseteq C_t$ for a decision tree i. For the decision trees that are compatible with drug t, we extract features corresponding to the cell lines in C_i from $\mathbf{\Omega}^{(t)}$ and apply decision tree i to it. The final value for each gene is the average of all compatible decision trees. Comparing to the "on-the-fly" method, the "two-level" method requires we train only once to obtain a random forest that is compatible with all drugs in the dataset.

The performance of both methods are summarized in Table 6. The "On-the-fly" random forest ranked the targets of 58 out of 152 drugs in the top 100 (38%), with 42 of them in top 50 (28%). The "Two-level" random forest leads to even better performance. 64 drugs (42%) are successfully classified in top 100 and 54 of them has targets ranked in top 50% (54%). We also compare these results with a random classifier, for which we repeated the same procedure for 20000 times but with randomly selected genes as drug targets. Figure 2 shows that the majority of these random trials has success percentages of approximately 7% when we use top 100 as the criterion, which is much worse than both methods. These encouraging result suggests that we can apply these two methods, especially the "two-level" random forest, to any small molecules that were applied to any 4 cell lines combinations in LINCS, and it is likely that we will enrich their correct targets to the top 100 list.

3.5 Gene Ontology Analysis of Targets

In the last stage of the project, we aim to investigate the biological differences between the correctly and incorrectly classified drugs using our "two-level" random forest. We believe such characterization can (1) reveal the scope of small molecules that our method is suitable for, and (2) give rise to additional features that can easily be incorporated into our random forest model.

We group the drugs whose targets were ranked among top 100 from the "two-level" random forest as "successful" (54 drugs total), and attempted to compare them with the rest of drugs, which were considered as "unsuccessful". To categorize these drug targets, we resorted to the Gene Ontology Enrichment Analysis tool (see Data and Methods for detail). Given a set of genes, this tool can find out which biological categories are over-represented (or under-represented). The tool supports many biological categories, and we are especially interested in the results from the "cellular com-

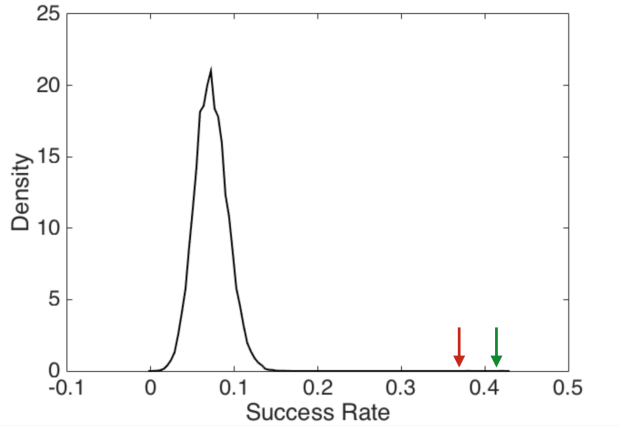

Figure 2: Comparing the random forest approaches with a random classifier. The red arrow indicates the success rate of on-the-fly random forest and the green arrow represents the two-level random forest.

	Cellular Component	p-value
Successful Targets	proteasome core complex	7.81E-37
	proteasome complex	1.1E-28
	proteasome alpha-subunit	5.68E-18
	cytosol	7.53E-12
	protein complex	1.88E-11
Failed Targets	transmembrane transporter complex	7.77E-15
	sodium-exchanging ATPase complex	4.42E-14
	cation-transporting ATPase complex	8.74E-13
	plasma membrane part	2.19E-11
	chloride channel complex	2.33E-9

Table 7: The cellular localization of successful and unsuccessful drug targets enriched by Gene Ontology

ponent", which compares the cellular localization between successful and failed targets. As shown in Table7, the successful targets mostly belong to the proteasome-related cellular components and they are intracellular. On the contrary, the failed targets are mostly associated with the transmembrane transporter complexes. This analysis reveals that our method tends to work with the drugs that have targets internal to the cell, while those that failed tended to have targets on the cell membrane. This observation is reasonable because the transmembrane targets are generally more difficult to characterize in biological assays, and therefore, their signature profiles may contain more noises in LINCS.

This findings motivate us to incorporate cellular component as an additional feature in our two-level random forest. We encode this feature by assigning 1 to the intracelluar genes and -1 to the extracellular ones (see Data and Methods for detail). We then run the two-level random forest again with this additional feature included. The result shows that the cellular component further improves the performance of two-level random forest. It increase the number of top 100 genes to 66 and top 50 genes to 55. The mean ranking is also improved significantly (from 718.2 to 615.3).

4 Discussion

In this project, we use the LINCS gene expression data and developed machine learning methods for this target prediction problem. Based on the main hypothesis that the gene expression profile of a drug-response experiment is correlated to that of the targets' knockdown experiment, we ex-

tracted the correlation feature from LINCS and integrate it with information from auxiliary datasets (such as PPI and cellular localization). We demonstrate that the features have significant predicative power and, when combined together in random forest models, they lead to remarkable prediction performance. In addition, the gene ontology analysis suggests that our method performs better with the intracellular targets.

It is still troublesome for a common laboratory to test all 100 proteins predicted from our approach. However, our method can serve as a valuable pre-pruning step for the more accurate computational structure based approaches that computationally dock the small molecule to the protein molecular structure and compute the interaction energies. Since our method can enrich the correct targets into the a 100 protein group, we need to perform much fewer docking experiments for the proteins in this list. To this end, we have collaborated with Prof. Carlos Camacho at the University of Pittsburgh to develop a pipeline to integrate our genomics-based predictions with detailed computational structure analysis. The detail of this collaboration is beyond the scope of this report.

It is worth noting that the accuracy of our methods is most likely underestimated. It is not uncommon for drugs to bind unintended "off-target" proteins in addition to their designed targets [1]. Therefore, drugs may actually bind to the proteins in our top 100 predictions. This observation reveals the key advantage of our methods when comparing to the conventional high-throughput screening (HTS). HTS is performed *in vitro*. Although we can obtain information about whether small molecules can bind to the protein, we do not know the behavior of small molecules in cell. LINCS experiments were performed *in vivo*, so they provide information of drug activity in the cell and when combining with structural approaches it also offer important insights on off-target interactions.

The Two-level random forest can be used on any small molecules in LINCS that have been to 4 or more cell lines in LINCS. We have identified 1598 such small molecules and completed the prediction for all of them. We are collaborating with biologists to test our predictions in biological assays.

References

[1] Andreas Bender, Josef Scheiber, Meir Glick, John W Davies, Kamal Azzaoui, Jacques Hamon, Laszlo Urban, Steven Whitebread, and Jeremy L Jenkins. Analysis of pharmacology data and the prediction of adverse drug reactions and off-target effects from chemical structure. *ChemMedChem*, 2(6):861–873, 2007.

[2] Konrad H Bleicher, Hans-Joachim Böhm, Klaus Müller, and Alexander I Alanine. Hit and lead generation: beyond high-throughput screening. *Nature Reviews Drug Discovery*, 2(5):369–378, 2003.

[3] Andrew Chatr-aryamontri, Bobby-Joe Breitkreutz, Rose Oughtred, Lorrie Boucher, Sven Heinicke, Daici Chen, Chris Stark, Ashton Breitkreutz, Nadine Kolas, Lara O'Donnell, et al. The biogrid interaction database: 2015 update. *Nucleic acids research*, page gku1204, 2014.

[4] Gene Ontology Consortium et al. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(suppl 1):D258–D261, 2004.

[5] Ramón Díaz-Uriarte and Sara Alvarez De Andres. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006.

[6] Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.

[7] Douglas B Kitchen, Hélène Decornez, John R Furr, and Jürgen Bajorath. Docking and scoring in virtual screening for drug discovery: methods and applications. *Nature reviews Drug discovery*, 3(11):935–949, 2004.

[8] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[9] Nathalie Malo, James A Hanley, Sonia Cerquozzi, Jerry Pelletier, and Robert Nadon. Statistical practice in high-throughput screening data analysis. *Nature biotechnology*, 24(2):167–175, 2006.

[10] John P Overington, Bissan Al-Lazikani, and Andrew L Hopkins. How many drug targets are there? *Nature reviews Drug discovery*, 5(12):993–996, 2006.

[11] TS Keshava Prasad, Renu Goel, Kumaran Kandasamy, Shivakumar Keerthikumar, Sameer Kumar, Suresh Mathivanan, Deepthi Telikicherla, Rajesh Raju, Beema Shafreen, Abhilash Venugopal, et al. Human protein reference database—2009 update. *Nucleic acids research*, 37(suppl 1):D767–D772, 2009.

[12] Yanjun Qi, Ziv Bar-Joseph, and Judith Klein-Seetharaman. Evaluation of different biological data and computational classification methods for use in protein interaction prediction. *Proteins: Structure, Function, and Bioinformatics*, 63(3):490–500, 2006.

[13] Gisbert Schneider and Uli Fechner. Computer-based de novo design of drug-like molecules. *Nature Reviews Drug Discovery*, 4(8):649–663, 2005.