It's all in the Genes

Application of AI to Discover Novel Binding of Small Molecules Based on Gene Expression Profiles

Dweepa Honnavalli[1], Kavya Varma[1], Susmitha Shankar[2], Gowri Srinivasa[3], and Prashath Athri[3']

[1,3]**PES University, Bangalore, India**
[2,3']**Amrita University, Bangalore, India**

Corresponding author:
Dweepa Honnavalli[1]

Email address: dweepa.prasad@gmail.com

ABSTRACT

The paper is concerned with the application of AI techniques in the field of drug repurposing. It uses the gene expression data in the level 5 LINCS dataset to represent different drugs and through deep learning techniques generates embeddings that inherently represent the functional properties of the drug. The paper makes use of two models for the purposes of generating embeddings. One based on the densely connected network and the second using the triplet network. Once generated, the embeddings are tested to determine their ability to discern the drug identity of the gene expression as well as the ATC class.

1 INTRODUCTION

Drug repurposing (or drug repositioning) is the process of finding a new medical utility for a drug (CC). Drug molecules often interact with not just the predetermined target but multiple other targets leading to unintended effects. Drug repurposing seeks to extricate the numerous drug-target interactions that can materialize and exploit these to reroute drugs to other utilities.

Prior to the approval or rejection of a new compound for medicinal use (CL), it undergoes rigorous testing which sheds light on the drug's pharmacology, formulation, dose, and potential toxicity (CC). The conclusions of this testing may result in drugs being approved or rejected either due to side effects it produces or failure in proving efficacy. The detailed study and analysis of a drug provides comprehensive information on the compound that becomes the foundation for drug repurposing. Accumulated data is used to discover novel applications of the pre-analysed drug reducing the otherwise delayed R&D timeframe for a new compound.

Increasing growth and standardisation in the field of genomics have resulted in a surge of data relating to drug-target interactions, gene expressions and adverse effects all of which are pertinent to drug repurposing. Computational biology can be employed to analyze and extract information from the extensive data available.

Target Based Similarity is an approach to drug repurposing where compound-protein interactions are analyzed to infer similar binding sites. An example of such data is the gene expression profile which is obtained by applying drugs to RNA strands and observing proteins released (CB). Gene expressions contain within it, embedded information on drug-target interactions that are effective in drug repurposing. Two drugs that can be used for similar applications will have some inherently common patterns within these gene expressions. Deep neural networks can be used to extract relevant embedded information and decide the biological utilities of a drug.

Drug repurposing capitalises on the fact that drugs in the market have already passed the FDA and other safety testing on humans. In order for a brand new drug to reach the market, it needs a lot of time and monetary investment to develop and more importantly, pass safety testing in order to reach production. In order to speed up the development, therapeutic biotech companies have adopted Drug Repositioning to reduce the lengthy R&D timeframe and leverage the previously conducted drug safety profile.

There exists millions of chemical compounds each with a specific set of properties that give it an appropriate use case. Finding drugs correctly suited for different requirements is generally considered a job for chemical laboratories where compounds are tested for various properties. It is difficult to accomplish this without extensive testing because drugs cannot be considered similar despite superficial similarities such as common functional groups. Two drugs with similar structures may not be suitable for the same application.

A solution for determining similarities between drugs stems from examining gene expressions. A gene expression is obtained by applying drugs to RNA strands and observing proteins released. Two drugs that can be used for similar applications will have some inherently common patterns within these gene expressions.

The problem statement for this project is to try to extract these inherent patterns using deep learning. Deep neural networks will be used to embed these gene expressions such that the embeddings of different drugs will signify how similar they are thus extracting compounds with similar applications without laboratory testing.

2 BACKGROUND AND LITERATURE SURVEY

Drug repositioning, of late, has gained interests of many biologists, chemists and drug companies. Around 30% of the drugs reaching the market today after FDA approval are repurposed [1]. There have been many different computational approaches underataken to repurpose drugs for new indications [2]. Broadly, these computational approaches fall under one of two categories:

- Target Based Similarity: Infers similar binding sites by assessing the compound–protein interactions.
- Disease Based Similarity: Associate drugs to new indications, by comparing the characteristics and similarities of different diseases.

Target based approaches are often coupled with techniques that infer similarity based on functional, chemical or structural properties of the perturbagen; perturbagen barcodes [3]; or through resulting adverse effects. We propose a method that is target based and involves deducing similarity based on inherent functional properties.

Most of the methods mentioned above (including our proposed method) are closely reliant on external datasets. To validate functional similarities of drugs in our paper, we make use of datasets such as *ATC (Anatomical Therapeutic Chemical) Classification System* which is used for the classification of drugs according to the organ system they act on (CC). We also use the *Drugbank* dataset [4] to bridge the gap between the ATC Classification system and the drugs present in our gene expression dataset.

The gene expression dataset under consideration is the the NIH Library of Integrated Network-Based Cellular Signatures Program [5]. The L1000 assay measures mRNA transcript abundance of 978 ”landmark” genes from human cells. Measurements of these 978 ”landmark genes” are applied to an inference algorithm to infer the expression of 11,350 additional genes in the transcriptome. 1.6 million profile expressions are obtained by measuring mRNA abundance in perturbed cells. The physical structure of the dataset is depicted in Table 1.

The dataset is divided into levels with different levels representing different aspects.

- Level 1: These consists of various ’Luminex’ graphs, where the x axis represents time and the y axis represents fluorescence, generated for different profiles.

.	profile 1	profile 2	...	profile 1.6 million
gene 1	+0.80	-0.04	...	+0.20
gene 2	-0.50	-0.95	...	+0.34
...	...	...	...	...
gene 978	0.75	+0.18	...	-0.07

Table 1. Representation of L1000 Assay

- Level 2: Gene expression values of 978 "landamrk genes" are obtained after deconvolution of Level 1 data.
- Level 3: This dataset has inferred information for 12K genes and 1.3 million profiles form level 2 data through a series of steps involving standardisation, normalisation and inferring the remaining 11k genes through a multiplication with a weighted matrix.
- Level 4: The dataset indicates the up and down regulation in the gene expression on adding perturbagen. After getting the standard gene expression values from the control genes by average all the control probe values for all genes and average control probes for each gene, we perform a Z score like operation to give us the regulated value for the table.
- Level 5: The replicates present in the level 4 data are consolidated which reduces the dimension of the data.

For the purpose of this problem, Level 5 data is considered. Overview of the dataset is shown in Table 2.

Number of perturbagens	2107
Number of samples per perturbagen on an average	42
Dimension of each sample	11,350
Total number of samples	118,500

Table 2. Overview of Level-5 L1000 Assay

Donner et. al [6] propose a novel approach in their paper which utilises a 'Dense Network' to generate embeddings trained using their drug identities and tested in a method of external evaluation against various drug classification markers like ATC classes, CHembl targets etc. A 'Dense Network' as described in the paper authored by Gao Huang [7] is a densely connected network where the output of each layer is an input to all the succeeding layers. They have also used a different activation function called SELU [8] which is a self normalising activation function that trains faster requires no external normalising layers. The paper authored by Donner et. al presents work that attempts to 'classify' drugs into different buckets which assume that all drugs corresponding to a bucket are similar to each other. We attempt to contrast this approach with our 'Triplet Network' by *separating* and not *classifying* different drugs to capture inherent similarities and differences without bucketing.

A 'Triplet Network' [9] is a network that is motivated by the 'Siamese Network' [10] which consists of identical twin networks that accept distinct inputs but are joined by a similarity function at the end. The Triplet Network augments the Siamese Network by extending a third identical network which accepts a distinct input and joins the other two networks at the end with a similarity function. Much of the work utilising the functionalities of the Siamese and the Triplet network are focused on image recognition and adjunct fields. These networks have an inherent capability to separate dissimilar inputs and appose similar inputs. Our work attempts to use this capability to clarify existing patterns of similarity along with finding new patterns within gene signatures to aide in discovering drugs that can be repurposed.

3 METHODOLOGY

Figure 1. Block Diagram of Proposed Method

The proposed method depicted in Fig. 1 involves cleaning and preprocessing gene expression data and learning a suitable embedding using a Deep Neural network approach, the results of which are then validated through internal (same dataset) and external validation (different datasets) and compared using different accuracy metrics.

3.1 Preprocessing

The original data (12k x 118k) is cleaned by reducing the number of columns by removing control genes. Each gene expression profile is the impact of a perturbagen at a particular time, concentration and other external factors on a probe of a cell line. Control genes are those sequences that have been determined as unchanging on addition of any perturbagen in any of the genes. There are 80 such control probes which are represented as 80 columns in the dataset. Each gene has all the control probes. The columns which represent these control probes are removed.

Out of 11,350 genes 978 of them are deemed as "landmark genes" which can represent 80% of data held by all the genes. The dimensions of the data are thus reduced to 978.
The dataset is then annotated by adding the perturbagen identity as labels to the columns.

The data is now in a usable format, which is a matrix of 978 rows and approximately 118k standardised columns- each column contributes as one input sample to the deep learning network.

3.2 Approaches

Generating embeddings requires learning intricate patterns of gene expression data to model similarities and dissimilarities. Two main deep learning models have been attempted to solve the problem of embedding gene expression data- Densenet and TripletNet.

3.2.1 DenseNet

Network This approach aims to create an embedding network that takes as input the gene expression corresponding to a perturbagen $[978x1]$ and outputs a representational embedding of this perturbagen of length e. The embedding network has a depth d and k number of neurons per layer. This network implements the densely connected network [7] where the input to each layer is the concatenation of the input to the previous layer and the output of the previous layer. This propagation of each layer's inputs introduces original inputs into each layer allowing for fewer neurons in a layer. The network uses the activation function SELU [8]. The self normalising nature of this activation function makes for faster training of the network. The last layer of the network is the size of the embedding (e). The normalized output from this layer is considered the embedding representing the perturbagen.

Loss Calculation The loss is calculated using an embedding matrix which represents the ideal embedding value for each perturbagen. It is a trainable matrix, is randomly initialised and is of size $[cxe]$ where c is the number of drug classes and e is the embedding length. During the training process, the similarity of the generated embedding with the embedding matrix is computed. The softmax cross entropy loss is calculated using these similarity values and this loss is minimized using an Adam Optimiser with a learning rate 0.005.

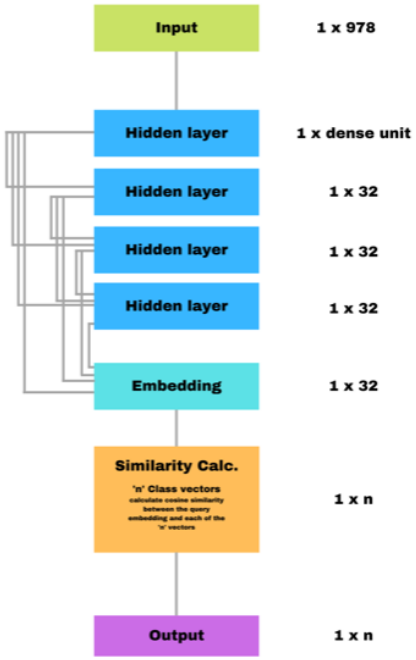


Figure 2. DenseNet Architecture

Model Configurations The parameters varied to obtain different configurations were the depth of the network, the number of parameters in the entire network and the length of the embedding. All models were trained for 100 epochs.

- Depth: 4, 8, 16, 24, 32
- Total number of parameters: 100K, 200K, 400K, 600K
- Embedding Length: 6, 8, 16, 24, 32

3.2.2 TripletNet

The Triplet Network approach involves three identical fully connected neural networks that use the same weights while working on three different input vectors concurrently. The three inputs to the network are referred to as 'Anchor', 'Positive', and 'Negative'.

- **Anchor.** The query gene expression.
- **Positive.** The gene expression which belongs to the same perturbagen class as the Anchor.
- **Negative.** The gene expression which does not belong to the same perturbagen class as the Anchor.

Dataset Generation An input to the TripletNet consists of 3 gene signatures- Anchor, Positive and Negative. These triplets are generated from the original LINCS level 5 dataset by sampling Anchor, Positive and Negative samples from each perturbagen. For each perturbagen identity, 50 permutations of Anchor, Positive and Negative are generated i.e 50 sets of Anchor, Positive and Negative.
The resulting dataset looks like this:

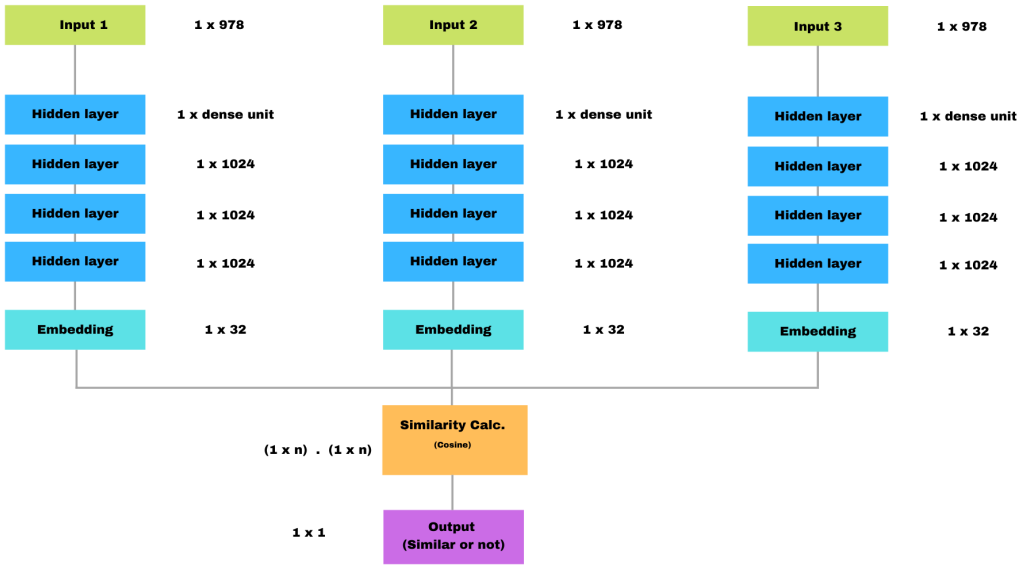


Figure 3. TripletNet Architecture

$$\begin{matrix} [Anchor & Positive & Negative] \\ [Anchor & Positive & Negative] \\ \dots & \dots & \dots \end{matrix}$$

Architecture The three inputs go through identical networks which result in three embeddings corresponding to each of the inputs. The architecture of the individual network is as follows:

- Number of layers: 3, 4, 5
- Number of neurons per layer: 512, 1024
- Activation: ReLU
- Dropout: 0.5, 0.9

Loss The loss function being employed by the network is a two-stage loss. Similarity between Anchor and Positive is calculated by minimising cosine distance between the two. Dissimilarity between Anchor and Negative is calculated by maximising cosine distance between the two.

$$\cos\theta = \frac{A.B}{|A|.|B|} \tag{1}$$

Two different Adam [11] optimisers are employed for this purpose- one minimises the distance, the other maximises.

3.3 Performance Evaluation

Performance evaluation is divided into two stages: Drug Identity Evaluation and Drug Class Evaluation.

3.3.1 Drug Identity Evaluation

In order to evaluate our trained network, we evaluated how well the the embeddings were able to retain the identities of the perturbagens. Query gene expressions belonging to the same perturbagens should output similar embeddings. The entire set of perturbagens are divided into 90% train and 10% test perturbagens. Gene expressions corresponding to these perturbagens contributed to the datasets. The network is trained

on the expression profiles stemming from the 'train' set of perturbagens and evaluated on how well it clusters expression profiles from the unseen(test) perturbagens. Each held-out profile was used as a query, and the quantiles and AUC values was estimated.

The metrics used for evaluation are:

1. *Recall by quantile:* Graph of quantile(0 to 1) vs Recall. Quantiles are described in section 3.3.3.
2. *ROC curve:* TPR vs FPR; True positive rate is the fraction of predicted positives out of all the samples that are actually positive. (TP/ TP+FN) and False positive rate is the fraction of predicted positives out of all the samples that are actually negative. (FP/FP+TN)
3. *Median quantile:* The quantile of the median value (Ideally is a low value)

3.3.2 Drug Class Evaluation

Once the embeddings are generated for all 118k gene expressions, perturbagen embeddings are extracted by averaging the embeddings of all the gene expressions belonging to that particular perturbagen.
In order to validate the embeddings procured by the penultimate layer of the network, we enlist the help of external datasets such as ATC (Anatomical Therapeutic Chemical) Classification System and Drugbank. In the ATC classification system, the active substances are divided into different groups according to the organ or system on which they act on. ATC is a hierarchical dataset where drugs are classified at five different levels. We concern ourselves only with the highest (first) level of classification that describes 14 classes of drugs.

1. Ailmentary tract and metabolism
2. Blood and blood forming organs
3. Cardiovascular system
4. Dermatologicals
5. Genito Urinary System and Sex Hormones
6. Systemic hormonal preparations, excl. sex hormones and insulin
7. Antineoplastic and immunomodulating agents
8. Antinfectives for systemic use
9. Musculo-skeletal system
10. Nervous system
11. Antiparasitic products, insecticides and repellents
12. Respiratory system
13. Sensory organs
14. Various

The dataset is prepared by mapping perturbagens to the ATC class using the help of the metadata present in the LINCS dataset as well as the Drugbank dataset [5]. Due to inconsistencies and replication in the metadata, complete mapping from perturbagen id to ATC class was not successful. $658/2107(31\%)$ of the perturbagens were mapped to their respective ATC Level 1 classes.

The distribution of classes in Fig. 4 showcases that the class distribution is skewed and gives an extremely unbalanced data set. All further predictions have to be done on the balanced side of the dataset or through some form of data balancing like upsampling or downsampling. Despite accounting for this skewed nature of the data, the paucity of data makes the results derived from these mappings less reliable.

The generated perturbagen embeddings were used to classify the drugs into the 14 ATC Level 1 classes. This classification was attempted via a variety of methods like KNN, SVM, ANN and clustering. The results of KNN are shown in Table 7.

3.3.3 Quantiles

'Quantiles' are a means of quantifying the performance of our network in generating similar embeddings for similar perturbagens and dissimilar embeddings for dissimilar ones. It is carried out using a query gene expression and a candidate set. The candidate set contains a set of positive candidates 'P' and negative candidates 'N'. The criteria for classifying candidates is as follows:

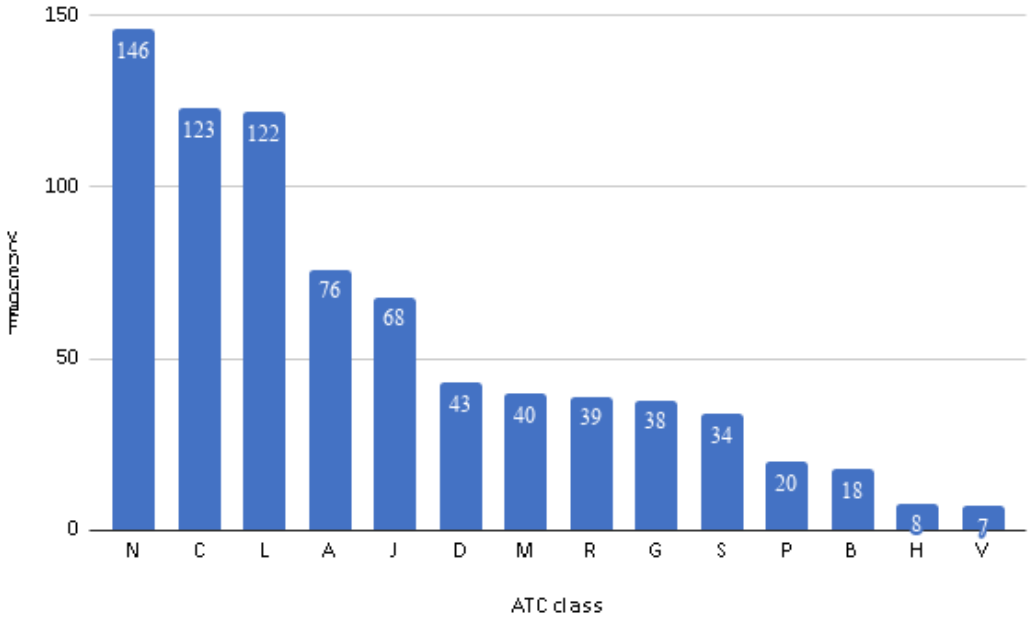


Figure 4. Distribution of ATC classes in Perturbagens

- *Positive.* Candidates belonging to the same functional group as the query perturbagen. perturbagen. Denoted by $\{p_i\}_{1 \leqslant i \leqslant P}$ where P is the total number of positive candidates for a specific query.
- *Negative.* Candidates belonging to different functional groups as the query perturbagen. Denoted by $\{n_j\}_{1 \leqslant j \leqslant N}$ where N is the total number of negative candidates for a specific query.

For each query (each row in dataset), similarity is calculated with all the embeddings in the candidate set(positive and negative). For each candidate $1 \leq i \leq P$ in P, the rank of that candidate is computed with all the negative scores. The rank is computed using this formula:

$$q_i = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{2}(sgn(n_j - p_i) + 1)$$

where $sgn(x)$ is the sign function.

The term 'recall' is used to measure the fraction of positives retrieved by the total number of positives.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

Top-x recall *r(x)* is defined as the number of positives ranked higher(having lower quantile) than the 'x' quantile where $x \in [0,1]$ i.e the fraction of negatives having higher similarities.

$$r(x) = \frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{2}(sgn(x - q_i) + 1)$$

where Q is the total number of positives over all queries, and the vector q of size Q is the concatenation of all the quantile vectors from the individual queries.

4 RESULTS AND DISCUSSION

4.1 DenseNet

Shown in Fig.5 are the results of the drug identification evaluation if the embeddings generated by the dense network on unseen data.

DenseNet													
Architecture of the Network				Results									
Number of Parameters	Depth	K	Length of Embedding Vector	Top -x Recall							Median Quantile Value	AUC
				0.0001	0.001	0.01	0.05	0.1	0.2	0.3		
97920	8	12	16	0.25661823	0.3108492	0.48553872	0.71204174	0.80985352	0.8873998	0.92138222	0.04800391	0.92324216
98352	32	3	8	0.18312384	0.2401648	0.41459067	0.65669429	0.77167649	0.86286667	0.9066963	0.05906855	0.90913053
98304	24	4	16	0.07529133	0.12142338	0.31820037	0.63486117	0.77001112	0.86778452	0.91041558	0.05460492	0.90907009
97920	8	12	32	0.088080374	0.13187511	0.301432976	0.555437334	0.709388983	0.877214938	0.924736887	0.063431021	0.905261595
98352	32	3	8	0.166532263	0.227214805	0.401369469	0.649235182	0.763932988	0.85171839	0.891664898	0.065783468	0.900293327

Figure 5. Drug Identification Results for DenseNet

4.2 TripletNet

Triplet												
Architecture of the Network					Results							
Number of Layers	Number of Neurons per Layer	Length of Embedding Vector	Dropout Rate	Number of Samples Per Perturbagen	Top -x Recall						Median Quantile Value	AUC
					0.0001	0.001	0.01	0.05	0.1	0.2	0.3		
Full dataset													
3	512	8	0.9	50	0.09312252	0.12865365	0.19474213	0.32635045	0.47198874	0.69412826	0.83026692	0.16799841	0.8207297
3	512	32	0.5	50	0.19969126	0.20334364	0.22174564	0.3189562	0.42367086	0.62250638	0.77469987	0.17859037	0.81291526
3	512	32	0.5	50	0.17577755	0.19357372	0.2169581	0.30603484	0.41333338	0.63711541	0.77777412	0.17884911	0.81275668
3	512	8	0.9	50	0.05235384	0.08049883	0.16203218	0.32292114	0.46853613	0.68528021	0.80906879	0.17512955	0.80838294
3	256	16	0.9	50	0.07911241	0.11044826	0.1786395	0.30697473	0.40926713	0.61978187	0.78338816	0.18592932	0.80679088
Unseen perturbagens													
3	256	32	0.9	50	0.00711934	0.0134208	0.05315201	0.19808921	0.3596452	0.63880304	0.80819066	0.18152965	0.80219774
3	256	8	0.9	50	0.00578657	0.01142236	0.04444215	0.16854288	0.31906714	0.60972066	0.81694549	0.19217465	0.80047967
3	256	32	0.9	50	0.01515919	0.0223768	0.05909813	0.20147527	0.36373617	0.64195666	0.81338814	0.18464008	0.8003947
3	512	16	0.9	50	0.01929744	0.02513852	0.05962016	0.19333259	0.34851336	0.61950595	0.80386233	0.18405507	0.80005888
3	512	16	0.9	50	0.013166	0.0202259	0.05918977	0.19626682	0.35667196	0.62357947	0.80246613	0.1876757	0.7948281

Figure 6. Drug Identification Results for TripletNet

The results from TripletNet (Fig. 6) manifest in two ways: first set of results correspond to the strength of the embeddings generated by a particular network on the full dataset (118k gene expression signatures) and the second set of results correspond to the strength of the embeddings generated on gene expressions corresponding to unseen perturbagens.

4.3 Drug Classification Results

Shown in Fig. 7 is the KNN results of drug classification on different embeddings of generated by the the two approaches. Shown in Table 3 is the KNN results of drug classification on the original data, the upsampled data and downsampled data for various number of classes.

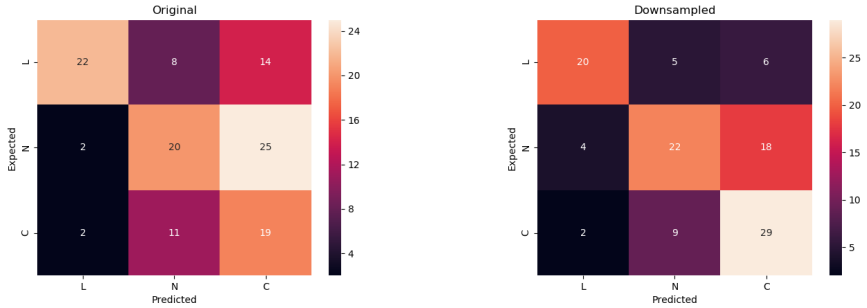
DenseNet: ATC Level 1 Classification (KNN)							
Architecture of the Network				Results			
Number of Parameters	Depth	K	Length of Embedding Vector	Number of Neighbours			
				7	8	9	10
98208	16	6	32	0.2689969605	0.2720364742	0.2583586626	0.2674772036
97920	8	12	16	0.273556231	0.2689969605	0.2568389058	0.2522796353
98304	24	4	16	0.2537993921	0.2629179331	0.2674772036	0.2598784195
97920	8	12	32	0.2568389058	0.2629179331	0.2629179331	0.2705167173
98304	24	4	32	0.2629179331	0.2598784195	0.2477203647	0.2537993921

Figure 7. Results for KNN for Drug Classification

KNN results for Top X classes		
Top X Classes	Original	Downsampling
Top 3	49.59%	61.74%
Top 4	33.10%	55.05%
Top 6	31.79%	41.38%
All	20.18%	25.24%

Table 3. KNN results for Top X classes

4.4 Confusion Matrices

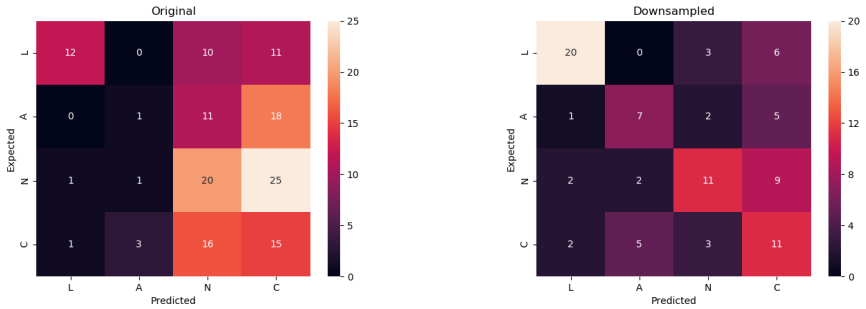


(a) Original **(b)** Downsampled

Figure 8. Confusion Matrix Top 3 classes

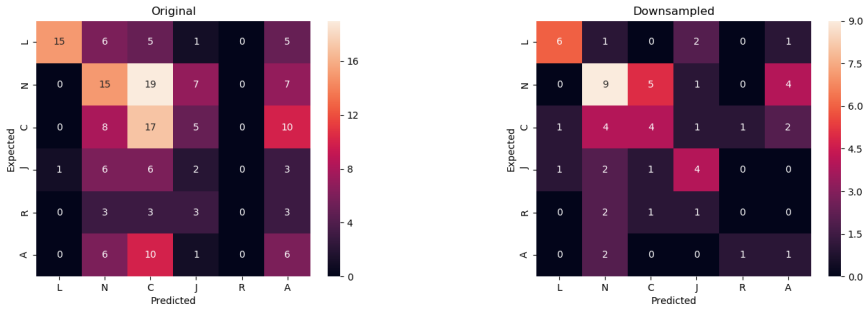
5 CONCLUSION & FUTURE WORK

The two approaches undertaken generate embeddings with a good ability to discern the drug identity but their ability to classify the drug into the ATC level 1 class is inconclusive due to both lack of mapped data as well as an inherent imbalance in the classes present. The significant improvement on change of sampling shows that the embeddings work well and results can be improved with more data.
The approach can be rerun on larger dataset such as the level 5, phase 1 or the level 4 data of the L1000 assay. Another aspect of future work is combining various classification values like the ATC class and other identifiers while generating the embeddings and not just the perturbagen identity. The performance of both the triplet network and the dense network is comparable and can lend itself well to combined network with the overall architecture of the triplet network with each of the parallel outputs as densely connected networks.



(a) Original **(b)** Downsampled

Figure 9. Confusion Matrix Top 4 classes



(a) Original **(b)** Downsampled

Figure 10. Confusion Matrix Top 6 classes

REFERENCES

[1] Craig Shimasaki. Understanding biotechnology business models and managing risk.

[2] Philippe Sanseau and Jacob Koehler. Editorial: Computational methods for drug repurposing. *Briefings in Bioinformatics*, 12(4):301–302, 07 2011.

[3] 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):1–19, 02 2017.

[4] Guo AC Lo EJ Marcu A Grant JR Sajed T Johnson D Li C Sayeeda Z Assempour N Iynkkaran I Liu Y Maciejewski A Gale N Wilson A Chin L Cummings R Le D Pon A Knox C Wilson M Wishart DS, Feunang YD. Drugbank 5.0: a major update to the drugbank database for 2018.

[5] 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 C. Smith, Daniel Lam, Arthur Liberzon, Courtney Toder, Mukta Bagul, Marek Orzechowski, Oana M. Enache, Federica Piccioni, Sarah A. Johnson, Nicholas J. Lyons, Alice H. Berger, Alykhan F. Shamji, Angela N. Brooks, Anita Vrcic, Corey Flynn, Jacqueline Rosains, David Y. Takeda, Roger Hu, Desiree Davison, Justin Lamb, Kristin Ardlie, Larson Hogstrom, Peyton Greenside, 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. *Cell*, 171(6):1437–1452.e17, November 2017.

[6] Yoni Donner, Stéphane Kazmierczak, and Kristen Fortney. Drug repurposing using deep embeddings of gene expression profiles. *Molecular Pharmaceutics*, 15(10):4314–4325, 2018. PMID: 30001141.

[7] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected

convolutional networks, 2016.

[8] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks, 2017.

[9] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network, 2014.

[10] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, page 737–744, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[11] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.