# Semi-Supervised Learning of the Electronic Health Record with Denoising Autoencoders for Phenotype Stratification

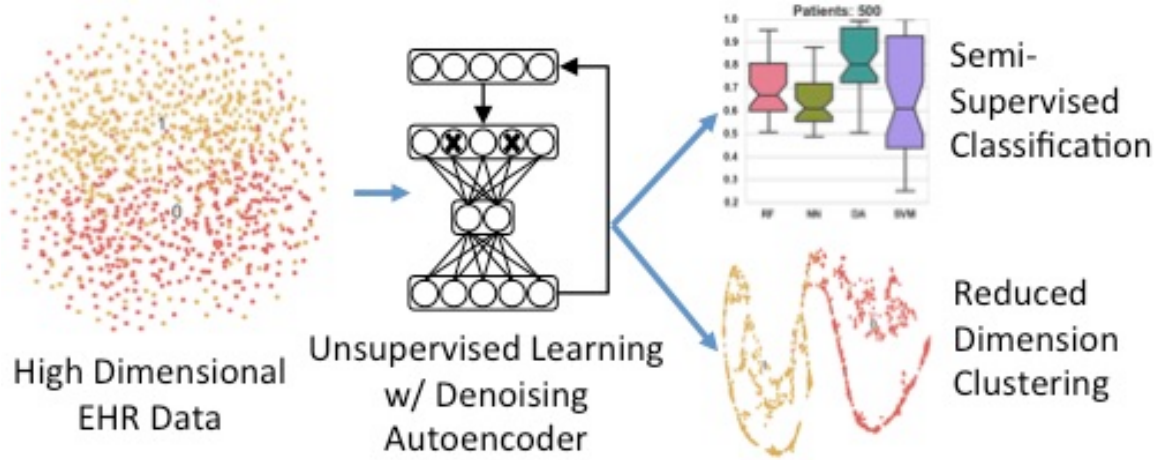
**Brett K. Beaulieu-Jones[1], Casey S. Greene[2]**
brettbe@med.upenn.edu
1 Graduate Group in Genomics and Computational Biology, Computational Genetics Lab, Institute for Biomedical Informatics, University of Pennsylvania
2 Department of Systems Pharmacology and Translational Therapeutics, Institute for Biomedical Informatics, Institute for Translational Medicine and Therapeutics, University of Pennsylvania

## ABSTRACT

Patient interactions with health care providers result in entries to electronic health records (EHRs). EHRs were built for clinical and billing purposes but contain many data points about an individual. Mining these records provides opportunities to extract electronic phenotypes that can be paired with genetic data to identify genes underlying common human diseases. This task remains challenging: high quality phenotyping is costly and requires physician review; many fields in the records are sparsely filled; and our definitions of diseases are continuing to improve over time. Here we develop and evaluate a semi-supervised learning method for EHR phenotype extraction using denoising autoencoders for phenotype stratification. By combining denoising autoencoders with random forests we find classification improvements across simulation models, particularly in cases where only a small number of patients have high quality phenotype. This situation is commonly encountered in research with EHRs. Denoising autoencoders perform dimensionality reduction allowing visualization and clustering for the discovery of new subtypes of disease. This method represents a promising approach to clarify disease subtypes and improve genotype-phenotype association studies that leverage EHRs.

**Keywords**: Electronic Health Record; Denoising Autoencoder; Unsupervised; Electronic Phenotyping , Patient Stratification

## GRAPHICAL ABSTRACT




## INTRODUCTION

Human diseases are complex and not perfectly understood. This means that while diseases are often considered as fixed phenotypes, many have evolving definitions and are difficult to classify. The electronic health record (EHR) is a popular source for electronic phenotyping to augment traditional genetic association studies, but there is a relative scarcity of research quality annotated patients [1]. Electronic phenotyping relies on either codes designed for billing or time intensive clinician review. This is an ideal environment for semi-supervised algorithms, particular those that perform unsupervised learning on many patients followed by supervised learning on a smaller, annotated, subset.

Denoising autoencoders (DAs) are a powerful tool to perform unsupervised learning [2]. DAs are trained similarly to artificial neural networks but taught to reconstruct an original input from an intentionally corrupted input. Through this training they find higher-level representations modeling the structure of the underlying data. By applying DAs in the EHR we sought to determine whether they could reduce the number of labeled samples required, construct non-billing code based phenotypes and elucidate disease subtypes for fine-tuned genetic association. Unsupervised learning can also help to recognize misdiagnosed patients in the form of outliers and is robust to changing disease definitions over time.

EHRs were designed for billing and clinical usage and not optimized for research. Despite this, EHRs have already proven an effective source of phenotypes in genetic association studies [3,4]. Initially, phenotypes were hand designed based on manual clinician review of patient records. These studies were limited by the time and cost inherent in manual review, but DAs can make use of unlabeled data [5]. Semi-supervised learning can be performed using DAs to perform unsupervised pre-training. After learning the structure of the data, the DA's hidden layer can be used as input to a traditional. This allows the DA to learn from all samples, even those without labels, and requires only a small subset to be annotated. Today, phenome-wide association studies (PheWAS) are the most prevalent example of EHR phenotyping, proving particularly effective at identifying pleiotropic genetic variants [6]. PheWASs often use algorithms based on the International Classification of Disease (ICD) codes to construct a phenotype. This coding system was designed for billing, not to capture human phenotypes. DA constructed features are combinations of all clinical data and may provide a more holistic view of a patient than billing codes alone.

Through extensive study, disease diagnoses are becoming more precise over time [7–11]. Cancers, for example, were historically typed by occurrence location and the efficacy of different treatment types. As the mechanisms of cancer are better understood, they are further categorized by their physiological nature. The progression of subtypes in lung cancer illustrates the change over time for a previously poorly defined disease [7]. Beginning with a single diagnosis based on occurrence in the lung, it was later differentiated as small cell lung cancer and non-small cell lung cancer [8,9]. Non-small cell lung cancer was then broken up into squamous cell carcinoma, adenocarcinoma, and large cell carcinoma. Today these subtypes continue to be broken up based on the genetic locations and pathways of associated risk variants [10]. Higher-level subtypes have been found to differ in cell size, cell shape, tumor origin site and chemical properties and it is becoming clear that genome-phenotype associations to these subtypes are a many-to-one relationship [11]. The unsupervised nature of DAs means that even as the definitions of a disease change, they would not need to be retrained. Such refinements are not limited to cancer subtyping. The ability to produce more homogenous phenotypes increased genotype to phenotype linkage in schizophrenia, bipolar disease [12], and Rett Syndrome

[13–16]. Furthermore, type 2 diabetes subtypes have been discovered using topological analysis of EHR patient similarity [17]. The dimensionality reduction possible with a DA makes clustering and visualization more feasible. Subtyping exposes disease heterogeneity and will contribute to physiological understanding of complex diseases.

DAs were initially introduced as a component in constructing the deep networks used in deep learning [18]. Deep learning algorithms have become the dominant performers in many classification problems including image recognition, speech recognition and natural language processing [19–24]. Deep learning techniques have recently been used with increasing popularity to solve biological problems including tumor classification, predicting chromatin structure and protein binding [2,25,26]. DAs showed strong performance early in the deep learning revolution but have been surpassed in most domains by convolutional neural networks or recurrent neural networks [18]. While complex deep networks such as convolutional neural networks have surpassed the performance of DAs in these areas, they rely on strictly structured relationships such as the relative positions of pixels within an image [21,27]. This structure is unlikely to exist in the EHR. In addition, convolutional neural networks and other deep networks (Deep belief networks, Recurrent Neural Networks etc.) are notoriously hard to interpret. DAs are easily generalizable, benefit from both linear and nonlinear correlation structure in the data, and contain accessible, interpretable, internal nodes [2]. Oftentimes the hidden layer is a “bottle-neck”, a much smaller size than the input layer, in order to force the autoencoder to learn the most important patterns in the data [27]. In patients diagnosed with the same disease, these important patterns may represent subtypes or other important patient clustering.

We evaluate DAs for phenotype construction using four simulation models of EHR data for complex phenotypes, modify DAs to effectively handle missingness in data and use the DA to create cluster visualizations that can aid in the discovery of subtypes of complex diseases.

# METHODS

We developed a denoising autoencoder approach that constructs phenotypes through unsupervised learning. To evaluate the DA, we created a simulation framework that represents multiple hidden factors that affect numerous potentially overlapping observed variables. We evaluated the reduced DA models against feature-complete representations with popular supervised learning algorithms. These evaluations covered both complete datasets, as well as the more realistic cases of incompletely labeled and missing data. Finally, we developed a technique that uses the reduced feature-space of the DA to visualize potential subtypes. Each of these is fully described in its own section below, full parameters included sweeps are available in the supplementary materials. Source code to reproduce each analysis is included in our repository (https://github.com/greenelab/DAPS) [28] and is provided under a permissive open source license (3-clause BSD). A docker build is included with the repository to provide a common environment to easily reproduce results without installing dependencies [29]. In addition, Shippable, a continuous integration platform, is used to reanalyze results and generate figures after each commit [30].

## Unsupervised Training

We used the Theano library [31,32] to construct a DA consisting of three layers, an input layer $x$, a single hidden layer $y$, and a reconstructed layer $z$ [18] (Figure 1A). Noise was added to the input layer through a stochastic corruption process, which masks 20% of the input values, selected at random, to zero.

The hidden layer y was calculated by multiplying the input layer by a weight vector $W$, adding a bias vector $b$ and computing the sigmoid (Formula 1). The reconstructed layer $z$ was similarly computed using tied weights, the transpose of $W$ and $b$ (Formula 2). The cost function is the cross-entropy of the reconstruction, a measure of distance between the reconstructed layer and the input layer (Formula 3).

$$y = s(Wx + b) \quad \text{(Formula 1)}$$

$$z = s(W'x + b') \quad \text{(Formula 2)}$$

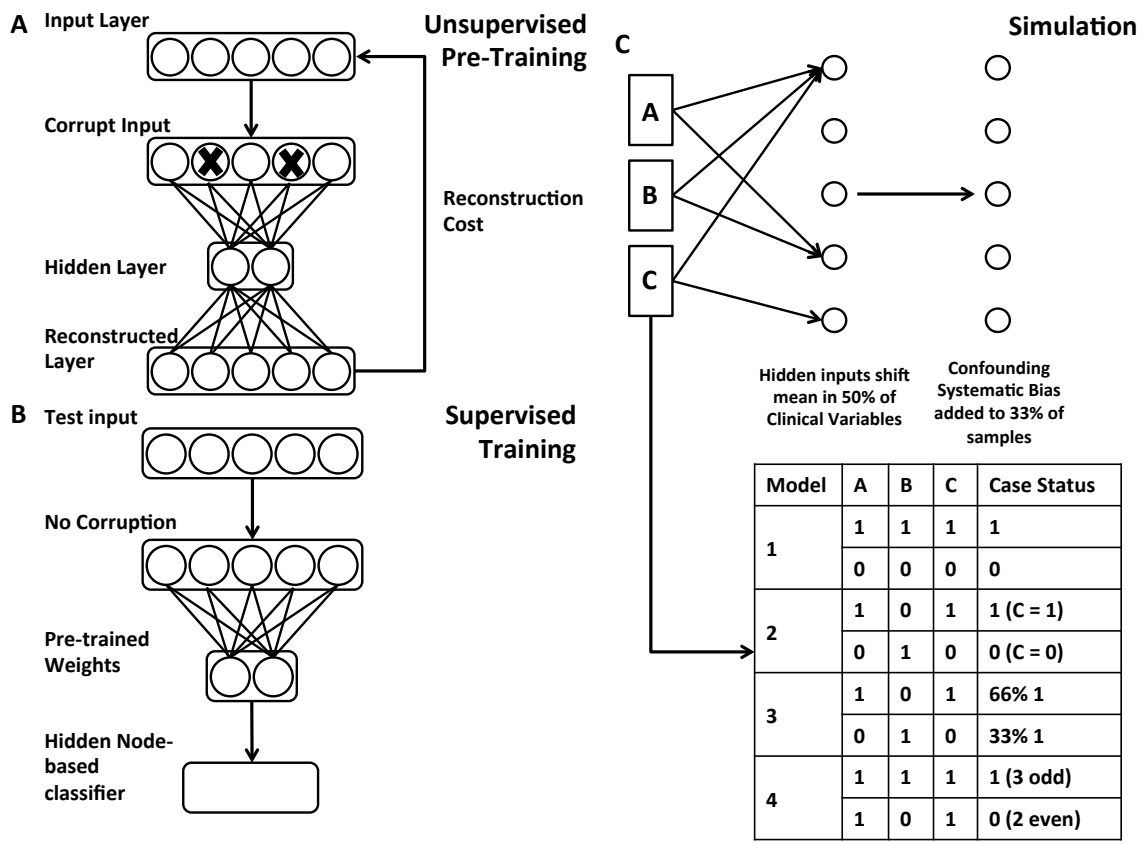
$$cost = -\sum_{k=1}^{d}[x_k \log(z_k) + (1 - x_k)\log(1 - z_k)] \quad \text{(Formula 3)}$$

Stochastic gradient descent was performed for 1000 training epochs, at a learning rate of 0.1. Hidden layers of two, four, eight and sixteen hidden nodes were included in the parameter sweep with a 20% input corruption level. Vincent et al. [18] provide a through explanation of training for DAs without missing data.

In the event of missing data, the cost calculation was modified to exclude missing data from contributing to the reconstruction cost. A missingness vector *m* was created for each input vector, with a value of 1 where the data is present and 0 when the data is missing. Both the input sample *x* and reconstruction *z* were multiplied by *m* and the cross entropy error was divided by the sum of the *m*, the number of non-missing features to get the average cost per feature present (Formula 4). This allowed the DA to learn the structure of the data from present features rather than imputation.

$$cost = -\sum_{k=1}^{d}[x_k \log(z_k)\ m_k + (1 - x_k)\log(1 - z_k)\, m_k] \,/\, count(m) \quad \text{(Formula 4)}$$

Full implementation and training details are available in the supplementary materials.




| Model | A | B | C | Case Status |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 1 | 1 (C = 1) |
| | 0 | 1 | 0 | 0 (C = 0) |
| 3 | 1 | 0 | 1 | 66% 1 |
| | 0 | 1 | 0 | 33% 1 |
| 4 | 1 | 1 | 1 | 1 (3 odd) |
| | 1 | 0 | 1 | 0 (2 even) |

**Fig. 1. A)** Network diagram of DAs used for unsupervised pre-training. **B)** Supervised classification occurs using the pre-trained DA hidden nodes as input to a traditional classifier. **C)** Simulation model with example cases and controls under each rule set.

## Supervised Denoising Autoencoder Classifier

To convert the DA to a supervised classifier, we first trained the DA in an unsupervised fashion (pre-training) (Fig 1A). We then applied a variety of traditional machine learning classifiers including, decision trees, random forests, logistic regression, nearest neighbors and support vector machines to the pre-trained unsupervised hidden layer values, $y$, of the DA (Figure 1B). Random forests applied to DA hidden nodes (DA+RF) were shown for all comparisons. Predictive performance was measured by comparing the AUROC using stratified 10-fold cross validation. The Scikit-learn library was used for the traditional classifiers [33]. The Support Vector Machine uses a radial basis function kernel.

## Simulation Framework

Patients were simulated from a set of abstract hidden input effects with a status of 1 or 0. These input effects shift the mean of 1 to N observed clinical variables chosen at random with replacement (Figure 1C). An example of a hypothetical condition a hidden input effect could represent is the familial hypercholesterolemia genotype. For a patient with the familial hypercholesterolemia genotype, the simulated clinical observations could represent increases in levels of total and low-density lipoprotein cholesterol, the deposition of cholesterol in extravascular tissues, corneal arcus and elevated triglyceride levels [34]. Some factors such as elevated triglyceride levels are not solely the result of the genetic predisposition and are related to environmental factors. Hypothetically additional hidden input effects on the same observed variable would represent these other factors. Because our goal is to evaluate methods for their ability to broadly capture these types of patterns, we generate randomized relationships between hidden and observed variables. This avoids overfitting our evaluation to specific phenotypes.

Next, a confounding systematic bias was added to a random subset (33%) of the patients as a source of additional noise to simulate the variance accompanying data created by physicians, labs, hospitals or other spurious effects.

Within the simulation combinations of hidden input effects determine case-control status under four models:

1. **All together/all relevant.** Individuals have the same value (0 or 1) for all hidden input effects. Controls have all hidden effects set to 0. Cases have all hidden effects set to 1. A model capturing any hidden input will be able to predict case/control status in this scenario.
2. **All independent /single effect relevant.** Individuals have 0 to N (specified per simulation) hidden input effects chosen at random. One arbitrary effect (the last one) is used to determine case-control status. In controls, this is 0. In cases, this is 1. A model capturing the relevant hidden input will be able to predict case/control status in this scenario.

3. **All independent/percentage based.** Individuals have 0 to N (specified per simulation) of hidden input effects chosen at random set to 1. The percentage of hidden input effects on represents the probability of the patient being a case. A model capturing more hidden effects will be able to more accurately predict case/control in this scenario.
4. **All independent/complex rule based.** Individuals have 0 to N (specified per simulation) of hidden inputs chosen at random set to 1. The sum of hidden effects determines case-control status (cases are even, controls are odd). A model must capture all hidden effects to successfully predict case/control in this scenario.

**Supervised Classification Comparison**

If successfully trained, the hidden layer of a DA, $y$, captures the first $n$ factors of variation in the data, where $n$ is the number of nodes in the hidden layer. To test whether the DA constructed useful features by learning the main factors of variation in the data we used the trained hidden layer as an input to a shallow classifier.

To do this, we first completed unsupervised pre-training of the DA with all of the simulated samples. The hidden layer values, $y$, were calculated for all samples using the trained DA without any corruption and fed in as the features to a random forest to form a supervised classifier.

Classification performance between DAs plus random forests (DA+RF) were compared against decision trees, random forests, nearest neighbors and support vector machines in a parameter sweep under each model (Table 1). Additional model parameters included in sweeps are included in the supplementary materials. All traditional classifiers were implemented with Scikit-learn [33]. Classification performance was compared using the AUROC.

**Table 1. Simulation Model 1 Parameter Sweep Specifications.**

| Parameter | Values |
|---|---|
| Observed Variables | 50, 100, 200, 400 |
| Effect Magnitude (x Variance) | 1, 2, 4 |
| Hidden Input Effects | 1, 2, 4, 8, 16 |
| Effected Observed Variables per Hidden Input Effect | 5, 10 |
| Unlabeled Patients | 10,000 |
| Labeled Patients | 100, 200, 500, 1,000, 2,000 |
| Systematic Bias | 0.1 applied to 0.33 of patients |
| DA Hidden Nodes | 1, 2, 4, 8 |

### Semi-Supervised Classification Comparison

The supervised classification comparison was repeated but with additional patients simulated and utilized during the unsupervised pre-training of the DA. The additional patients were simulated at the same 50% case, 50% control ratio but their labels were discarded after simulation. These additional patients were mixed with the original labeled patients and included in the unsupervised pre-training of the DA. The unlabeled samples were then discarded and the DA+RF was then provided the same, labeled, patient groups as the traditional classifiers. The labeled patient samples were run through the trained DA in the same manner as the unsupervised pre-training but without any corruption added to the data. The DA+RF and traditional classifiers were evaluated in a parameter sweep under each model using 10-fold cross validation.

### Missing Data Comparison

The semi-supervised classification comparison was repeated five times with, 0%, 10%, 20%, 30% and 40% of the data missing. Missing data was added at random per sample, depending on the specified percentage missing.

Throughout these trials, the cost calculation was modified to exclude missing data from the cost and allow the DA to learn without imputing values (Formula 4). The traditional classifiers were trained using mean imputation for missing data. Mean imputation is particularly well suited for the simulation models because the observations were drawn from normal distributions, potentially giving an advantage to the non-DA algorithms that would not be available in many real datasets.

As in the semi-supervised classification comparison trial, the DA+RF and traditional classifiers were evaluated under each model using 10-fold cross validation.

### Clustering and Visualization

To interpret and visualize results, patient populations were clustered using principal components analysis (PCA) and t-Stochastic Neighbor Embedding (t-SNE) of the trained DA's hidden nodes [35,36]. PCA and t-SNE were implemented with the Sci-kit learn library [33].

Ten thousand patients (5,000 cases, 5,000 controls) with four hidden effects were simulated under model 1. PCA followed by t-SNE was performed initially on the raw input for comparison and then on the hidden nodes of the DA after every 10 training epochs.

To test the ability to identify subtypes, we simulated 15,000 patients, 5,000 cases under model 1, 5,000 cases under model 2, and 5,000 controls. Input observations were compared to two, three and four-node DAs using PCA followed by t-SNE.

## RESULTS

### Case-Control DA Training Visualization

We trained a DA and visualized the training process using PCA and t-SNE. These visualization techniques offer intuition and the ability to examine the sub-clusters. Given 5,000 cases and 5,000 controls under simulation model 1, PCA and t-SNE alone did not yield defined clusters (Fig 2A). Figures 2B-F show the separation of cases from controls

as the DA is trained. One thousand epochs of training via stochastic gradient descent were found to be sufficient for the convergence of reconstruction cost and stabilization of visualizations within simulated data (Figure 2 E and F, Supplemental Figure 1).

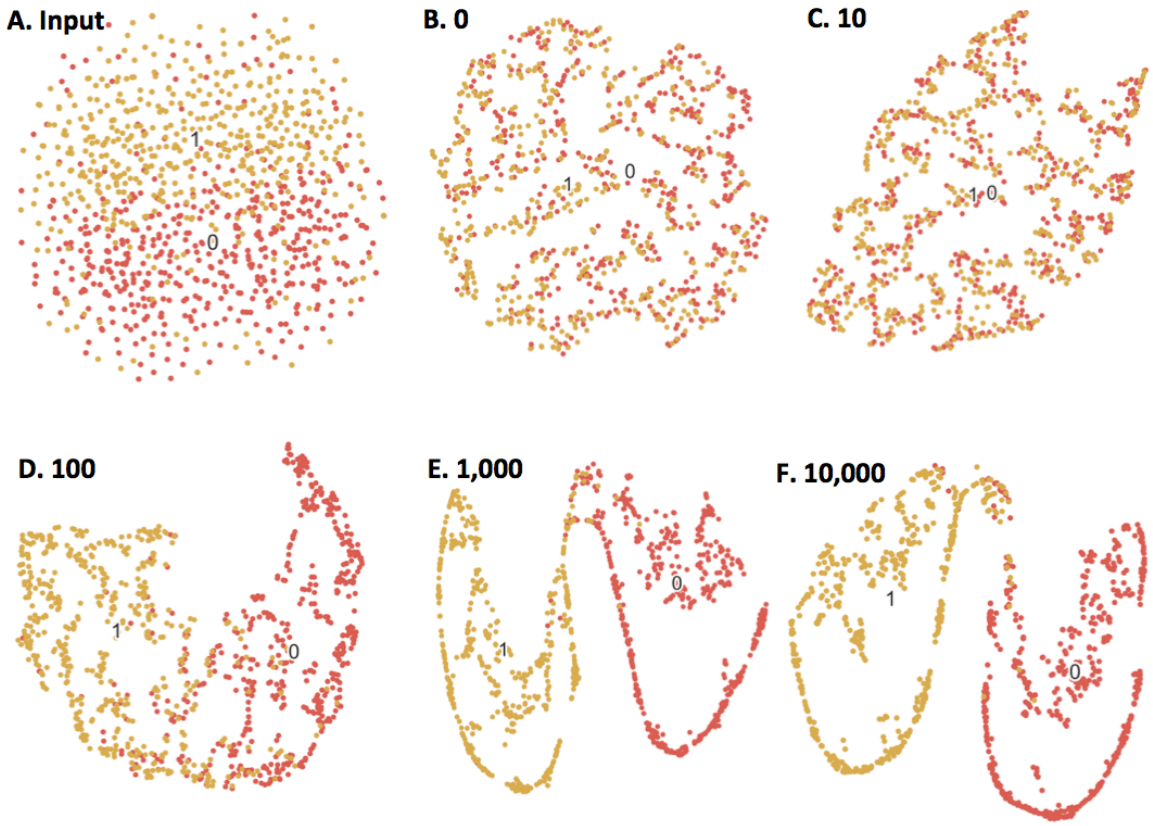



**Fig. 2** Case vs. Control clustering via principal components analysis and t-distributed stochastic neighbor embedding throughout the training of the DA. **A.)** Raw input **B.)** 0 training epochs **C.)** 10 training epochs **D.)** 100 training epochs **E.)** 1,000 training epochs **F.)** 10,000 training epochs.

**Fully Supervised Comparison**

To examine the ability of DAs to learn the structure of the data we compare the predictive ability of classification algorithms applied to the DA constructed through unsupervised training. Random forests demonstrated a strong balance of performance and stability, and were used for all comparisons (Supplemental Figure 2). We then compare the DA plus a random forest classifier (DA+RF) to the top performing classifiers on raw input data (Table 2).

**Table 2. Mean Receiver Operating Curve Area Under Curve by method under simulation model 1.**

| **Patients** | **DA+RF** | **Random Forest** | **Support Vector Machine** | **Decision Tree** | **Nearest Neighbors** |
|---|---|---|---|---|---|
| **100** | 0.618 | 0.653 | 0.504 | 0.599 | 0.635 |
| **200** | 0.637 | 0.610 | 0.449 | 0.589 | 0.608 |
| **500** | 0.677 | 0.690 | 0.663 | 0.617 | 0.642 |
| **1000** | 0.774 | 0.717 | 0.776 | 0.634 | 0.651 |
| **2000** | 0.755 | 0.736 | 0.862 | 0.643 | 0.658 |
| **Mean** | 0.692 | 0.681 | 0.651 | 0.616 | 0.639 |

Key trends emerged under each model; with few patients SVMs had AUCs indistinguishable from those expected from a random classifier. As one would expect, SVMs were top performers at when the number of patients was high. Random forest classification performance scaled steadily with patient count. The DA+RF performed similarly to the random forest, showing that a 2-node DA is able to capture at least one of the input hidden effects. Capturing any signal is sufficient to accurately classify simulation model 1.

**Semi-Supervised Comparison**

The full potential of the DA+RF is reflected in semi-supervised parameter sweep comparison for simulation model 1 (Fig 3A). The DA method's performance is high, even with very few labeled examples, when the sufficient unlabeled examples are available. Because of the extreme feature reduction, the traditional classifier on top of the DA is able to reach its learning capacity with very few labeled patients (Fig 3A). Efficient learning from labeled examples is critical in practical use cases because there are often few well-annotated cases due to the expense of clinician manual review. The 2-node DA plus random forest also showed strong performance in relation to an SVM when there were many observed clinical variables (Fig 3B) and when there were many hidden effects. The SVM again showed the highest performance at very high numbers (1000 or more) of labeled patients.

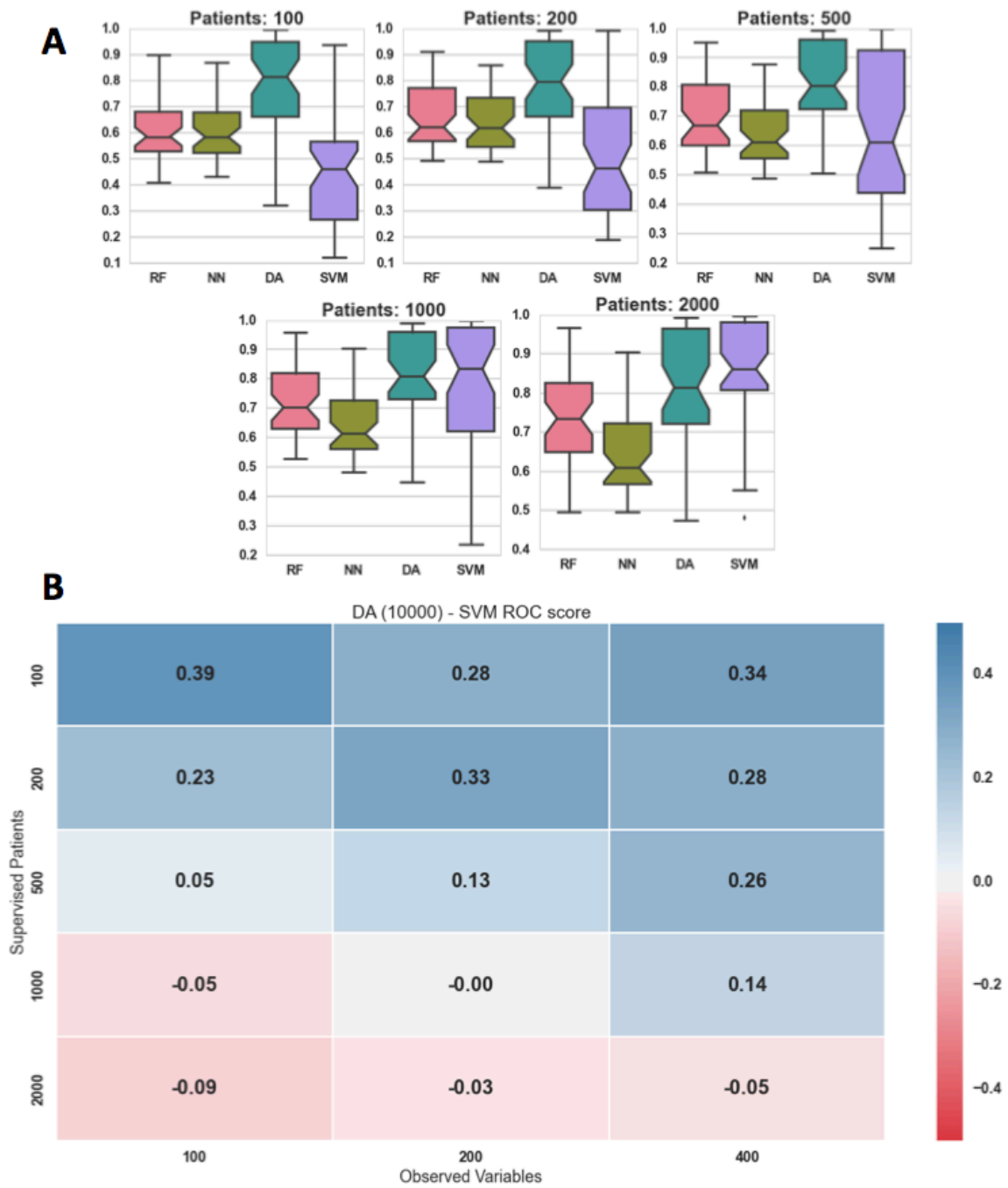



**Fig. 3 A)** Classification AUC in relation to the number of labeled patients under simulation model 1 (RF – Random Forest, NN – Nearest Neighbors, DA – 2-node DA + Random Forest, SVM – Support vector machine). Unsupervised pre-training of the 2-node DA was performed with 10,000 patients. **B)** Heat Map showing the difference of DA+RFC and SVM methods in relation to the number of labeled patients and observed variables under simulation model 1.

These patterns repeat across the other simulation models, with more complex models requiring more hidden nodes to adequately model the structure of the data. In simulation model 2 (Fig 4A), both 4 and 8 node DAs outperform the 2-node DA. In simulation model 3, the probabilistic manner of simulation means that even a perfect classifier would be unable to classify all samples, so methods were compared on this model using expected maximum predictive accuracy of a perfect classifier. In this model, case control odds were equal to the percentage of hidden input effects on. If there are 4 hidden input effects and 2 are on, the patient has a 50% chance of being a case and a 50% chance of being a control. A classifier cannot model this uncertainty and the maximum expected accuracy was calculated from a binomial distribution multiplied by the minority percentage as the best a classifier could do is learn the majority class. For example, in the case of 4 hidden effects the maximum expected accuracy is 68.75%. Under Model 3, the 4-node DA is the strongest performing, with median performance 5% better than the next best traditional classifier. Model 4 (Fig 4C, 4D) was the most difficult to classify as the classifier had to capture all of the hidden effects to be accurate. In several cases, no classifier did better than the expected performance of a random classifier. In fact, the SVM's average AUC over the entire sweep was indistinguishable from random performance. As expected, the 2-node performs worse than the 4 and 8-node DAs on model 4. The 2-node DA lacks sufficient dimensionality to capture more than 4 hidden input effects.

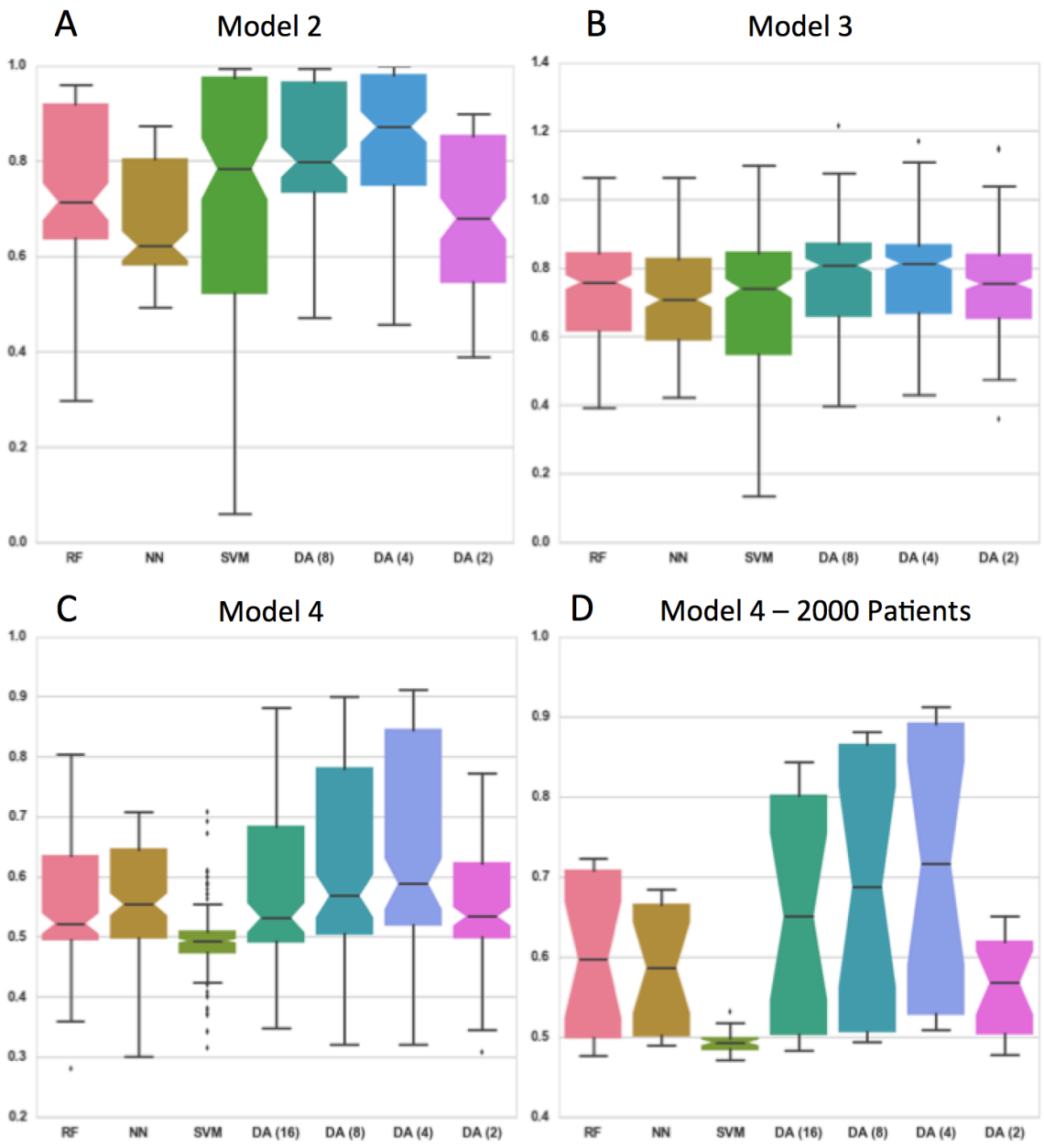



**Fig. 4 A)** Classification Accuracy of model 2 (1, 2, 4 and 8 effects). **B)** Classification AUC normalized to simulation model 2 expected max predictive accuracy (1, 2, 4 and 8 effects). **C)** Classification AUC of model 4 (1, 2, 4 and 8 effects). **D.)** Classification AUC of model 4 (parameter sweep results for 1, 2, 4 and 8 effects using only the parameter sets with 2,000 labeled patients)

**Semi-Supervised Missing Data Comparison**

Clinical records often have empty fields, so algorithms must be robust to missing data. We evaluated the DA's robustness in this situation. The DA is robust to missing data maintaining near-max classification performance across the missingness proportions tested (Fig5, Supp. Fig 3). For these simulation models, the mean imputation used for non-DA approaches is an ideal strategy. Figure 5B shows consistent performance between the DA and SVM even as the percent of data missing increases, suggesting that the DA is at least as robust as the ideal imputation method.

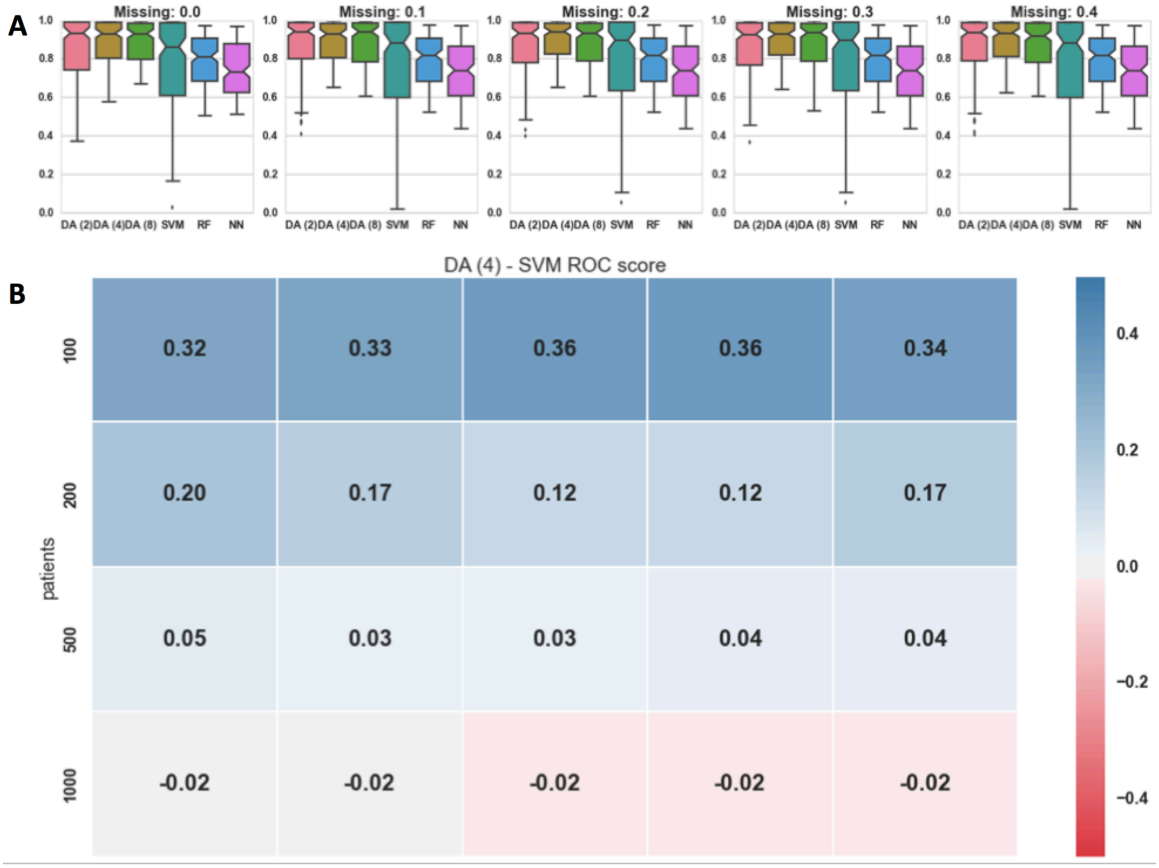



**Fig. 5. A)** Classification AUC in relation to the amount of missing data under simulation model 1. **B)** Heatmap showing difference of DA and SVM in relation to supervised patient count and percent of missing data.

## Simulated Subtype Clustering Visualization

We evaluated the DAPS' ability to perform patient stratification. To perform this analysis, we simulated 5,000 cases from each of two different models (1 and 2) to represent a disease with two subtypes. An additional 5,000 controls were simulated. We then visualized the DA constructed from this set of patients using PCA followed by tSNE. In the input data, the subtypes are relatively overlapping (Fig 6A). A DA with two nodes was also unable to separate this number of subtypes (Figure 6B). Visualizations constructed from DAs with three (Figure 6C) or four (Figure 6D) were able to effectively separate both subtypes of cases from each other and from controls.

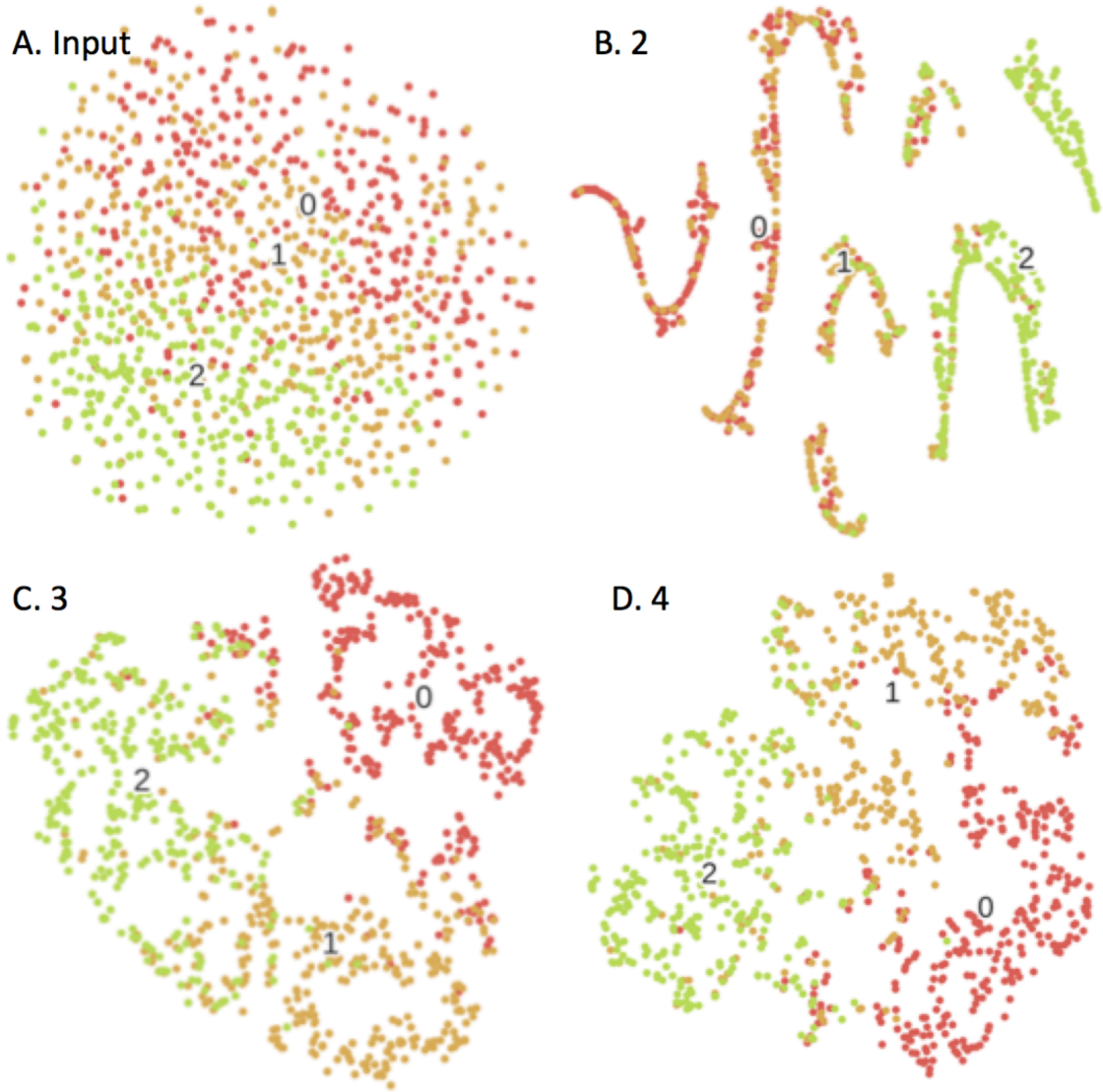



**Fig. 6** Case vs. Control clustering via principal components analysis and t-distributed stochastic neighbor embedding after training the DA for controls and cases generated from a combination of models 1 and 3. **A)**. Raw input. **B.)** 1,000 training epochs with 2 hidden nodes. **C.)** 1,000 training epochs with 3 hidden nodes. **D.)** 1,000 training epochs with 4 hidden nodes.

# DISCUSSION

In this study, we presented a semi-supervised learning approach using DAs to model patients in the EHR. Competitive supervised classification accuracy with a large degree of feature reduction indicates the DA successfully learned the structure of the high-dimensional EHR data. DAs are particularly well suited to the EHR because their unsupervised nature allows the formation of a semi-supervised classifier and the ability to utilize large un-annotated patient populations to improve classification accuracy. The dimensionality reduction of DAs allows clustering of the reduced feature set for the visualization and determination of subtypes. These clusters may reveal disease subtypes, fine-tuned targets for genotype-phenotype association. The DA models are easily de-constructible because they use a simple model for the traditional classifier with transparent node compositions that can be traced back to inputs. In addition, our method proposes a straightforward modification to the DA to enable it to process missing data without imputation.

PheWASs are a powerful tool to leverage the vast clinical data contained in the electronic health record but currently suffer from the reliance on billing codes or manual clinician annotation. Denny et al. [1] call out the need for increased accuracy in phenotype definition in the original PheWAS publication, particularly for rare phenotypes or phenotypes that do not directly correspond with a billing code. In addition, several studies have found increased genetic linkage via subtyping [12–16,37]. Li et al. [17] presented a powerful example of EHR subtyping of patients with type 2 diabetes using a similar methodology, but they utilized Ayasdi, a commercial, closed source topology data analysis software tool. Our method is built on free, open source libraries that will continue to be improved and our software is accessible for the research community.

DA nodes and clusters of nodes provide composite variables that may better approximate and represent the condition of the subject. These additional phenotype targets may provide more homogeneous targets for genotype associations. Beyond genotype to phenotype association, these visualizations may also help clinicians to understand the

level of heterogeneity for a specific disease and to make treatment associations among sub-clusters of patients.

Our work provides an important contribution but challenges remain. Most importantly, the transition from simulated data to real world clinical data requires an additional patient selection step, in order to find suitable patients to perform the unsupervised pre-training step. In addition, we assume a preprocessing step has already been performed to handle the compound structure present in the EHR. This step is necessary to transform categorical, free text, images and temporal data to suitable input for the DA.

Future work will focus on developing tools to examine and interpret constructed phenotypes (hidden nodes) and clusters. We anticipate high weights indicate important contributors to node construction revealing relevant combinations of input features. In addition, we will develop a framework for evaluating the significance of constructed clusters for genotype to phenotype association. Finally we will construct a scheme for determining optimal hyper parameter (i.e. hidden node count) selection.

## Acknowledgements

This work is supported by the Commonwealth Universal Research Enhancement (CURE) Program grant from the Pennsylvania Department of Health as well as the Gordon and Betty Moore Foundation's Data-Driven Discovery Initiative through Grant GBMF4552 to C.S.G. The authors would like to thank Dr. Jason H Moore for helpful discussions. The authors acknowledge the support of the NVIDIA Corporation for the donation of a TitanX GPU used for this research.

**References**


[1] J.C. Denny, L. Bastarache, M.D. Ritchie, R.J. Carroll, R. Zink, J.D. Mosley, et al., Systematic comparison of phenome-wide association study of electronic medical record data and genome-wide association study data, Nat. Biotechnol. 31 (2013) 1102–1111. doi:10.1038/nbt.2749.

[2] J. Tan, M. Ung, C. Cheng, C.S. Greene, Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders., Pacific Symp. Biocomput. 20 (2015) 132–43. http://www.ncbi.nlm.nih.gov/pubmed/25592575.

[3] S.J. Hebbring, S.J. Schrodi, Z. Ye, Z. Zhou, D. Page, M.H. Brilliant, A PheWAS approach in studying HLA-DRB1*1501., Genes Immun. 14 (2013) 187–91. doi:10.1038/gene.2013.2.

[4] J.C. Denny, D.C. Crawford, M.D. Ritchie, S.J. Bielinski, M. a. Basford, Y. Bradford, et al., Variants near FOXE1 are associated with hypothyroidism and other thyroid conditions: Using electronic medical records for genome- and phenome-wide studies, Am. J. Hum. Genet. 89 (2011) 529–542. doi:10.1016/j.ajhg.2011.09.008.

[5] J.S. Elkins, C. Friedman, B. Boden-Albala, R.L. Sacco, G. Hripcsak, Coding neuroradiology reports for the Northern Manhattan Stroke Study: a comparison of natural language processing and manual review., Comput. Biomed. Res. 33 (2000) 1–10. doi:10.1006/cbmr.1999.1535.

[6] A.L. Tyler, D.C. Crawford, S. a Pendergrass, The detection and characterization of pleiotropy: discovery, progress, and promise., Brief. Bioinform. (2015) bbv050–. doi:10.1093/bib/bbv050.

[7] W.D. Travis, E. Brambilla, M. Noguchi, A.G. Nicholson, K.R. Geisinger, Y. Yatabe, et al., International association for the study of lung cancer/american thoracic society/european respiratory society international multidisciplinary classification of lung adenocarcinoma, J Thorac Oncol. 6 (2011) 244–285. doi:10.1513/pats.201107-042ST.

[8] Kreyberg L., Histological lung cancer types. A morphological and biological correlation., Acta Pathol Microbiol Scand Suppl. (1962) 157:1–92.

[9] C.F. Mountain, Revisions in the International System for Staging Lung Cancer., Chest. 111 (1997) 1710–7. doi:10.1378/chest.111.6.1710.

[10] L. West, S.J. Vidwans, N.P. Campbell, J. Shrager, G.R. Simon, R. Bueno, et al., A novel classification of lung cancer into molecular subtypes, PLoS One. 7 (2012) 1–11. doi:10.1371/journal.pone.0031906.

[11] A.S. Crystal, A.T. Shaw, L. V Sequist, L. Friboulet, M.J. Niederst, E.L. Lockerman, et al., Patient-derived models of acquired resistance can identify effective drug combinations for cancer., Science. 346 (2014) 1480–6. doi:10.1126/science.1254721.

[12] A. Labbe, A. Bureau, I. Moreau, M.-A. Roy, Y. Chagnon, M. Maziade, et al., Symptom dimensions as alternative phenotypes to address genetic heterogeneity in schizophrenia and bipolar disorder, Eur. J. Hum. Genet. 20 (2012) 1182–1188. doi:10.1038/ejhg.2012.67.

[13] P. Chaste, L. Klei, S.J. Sanders, V. Hus, M.T. Murtha, J.K. Lowe, et al., A Genome-wide Association Study of Autism Using the Simons Simplex Collection: Does Reducing Phenotypic Heterogeneity in Autism Increase Genetic Homogeneity?, Biol. Psychiatry. 77 (2015) 775–784. doi:10.1016/j.biopsych.2014.09.017.

[14] F. Ariani, G. Hayek, D. Rondinella, R. Artuso, M.A. Mencarelli, A. Spanhol-Rosseto, et al., FOXG1 Is Responsible for the Congenital Variant of Rett Syndrome, Am. J. Hum. Genet. 83 (2008) 89–93. doi:10.1016/j.ajhg.2008.05.015.

[15] A. Bureau, A. Labbe, J. Croteau,, C. Mérette, Using disease symptoms to improve detection of linkage under genetic heterogeneity, Genet. Epidemiol. 32 (2008) 476–486. doi:10.1002/gepi.20320.

[16] M. Maziade, M.A. Roy, M. Martinez, D. Cliche, J.P. Fournier, Y. Garneau, et al., Negative, psychoticism, and disorganized dimensions in patients with familial schizophrenia or bipolar disorder: continuity and discontinuity between the major psychoses, Am J Psychiatry. 152 (1995) 1458–1463.

[17] L. Li, W. Cheng, B.S. Glicksberg, O. Gottesman, R. Tamler, R. Chen, et al., Identification of type 2 diabetes subgroups through topological analysis of patient similarity, 7 (2015) 1–16. doi:10.1126/scitranslmed.aaa9364.

[18] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, Proc. 25th Int. Conf. Mach. Learn. - ICML '08. (2008) 1096–1103. doi:10.1145/1390156.1390294.

[19] Y. Bengio, Learning Deep Architectures for AI, Found. Trends® Mach. Learn. 2 (2009) 1–127. doi:10.1561/2200000006.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., Going Deeper with Convolutions, arXiv Prepr. arXiv1409.4842. (2014) 1–12. doi:10.1109/ICCV.2011.6126456.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the Inception Architecture for Computer Vision, (2015). http://arxiv.org/abs/1512.00567.

[22] G. Hinton, L. Deng, D. Yu, G. Dahl, A.R. Mohamed, N. Jaitly, et al., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal Process. Mag. 29 (2012) 82–97. doi:10.1109/MSP.2012.2205597.

[23] Y. Bengio, R. Ducharme, P. Vincent, C. Janvin, A Neural Probabilistic Language Model, J. Mach. Learn. Res. 3 (2003) 1137–1155. doi:10.1162/153244303322533223.

[24] A. Graves, A. Mohamed, G. Hinton, Speech Recognition With Deep Recurrent Neural Networks, Icassp. (2013) 6645–6649. doi:10.1109/ICASSP.2013.6638947.

[25] J. Zhou, O.G. Troyanskaya, Predicting effects of noncoding variants with deep learning–based sequence model, Nat. Methods. 12 (2015) 931–934. doi:10.1038/nmeth.3547.

[26] Y. Park, M. Kellis, news and views Deep learning for regulatory genomics, Nat. Publ. Gr. 33 (2015) 825–826. doi:10.1038/nbt.3313.

[27] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408. doi:10.1111/1467-8535.00290.

[28] B.K. Beaulieu-Jones, Denoising Autoencoders for Phenotype Stratification (DAPS): Preprint Release. Zenodo, (2016). doi:10.5281/zenodo.46165.

[29] Docker, Docker, (2016). https://www.docker.com/.

[30] Shippable, Shippable, (2016). https://www.shippable.com.

[31] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, et al., Theano: new features and speed improvements, arXiv Prepr. arXiv …. (2012) 1–10. http://arxiv.org/abs/1211.5590.

[32] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, et al., Theano: a CPU and GPU math compiler in Python, in: 9th Python Sci. Conf., 2010: pp. 1–7. http://www-etud.iro.umontreal.ca/~wardefar/publications/theano_scipy2010.pdf.

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al., Scikit-learn: Machine Learning in Python, … Mach. Learn. …. 12 (2012) 2825–2830. http://dl.acm.org/citation.cfm?id=2078195\nhttp://arxiv.org/abs/1201.0490.

[34] F. Civeira, Guidelines for the diagnosis and management of heterozygous familial hypercholesterolemia., Atherosclerosis. 173 (2004) 55–68. doi:10.1016/j.atherosclerosis.2003.11.010.

[35] L. Van Der Maaten, G. Hinton, Visualizing Data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605. doi:10.1007/s10479-011-0841-3.

[36] G.E. Hinton, S.T. Roweis, Stochastic neighbor embedding, Adv. Neural Inf. Process. Syst.

(2002) 833–840. doi:http://books.nips.cc/papers/files/nips15/AA45.pdf.

[37] M. Manchia, J. Cullis, G. Turecki, G.A. Rouleau, R. Uher, M. Alda, The Impact of Phenotypic and Genetic Heterogeneity on Results of Genome Wide Association Studies of Complex Diseases, PLoS One. 8 (2013) e76295. doi:10.1371/journal.pone.0076295.

# Semi-Supervised Learning of the Electronic Health Record with Denoising Autoencoders for Phenotype Stratification (DAPS)

## SUPPLEMENTARY MATERIALS

## Parameter Sweep Specifications

**Supp. Table 1. Simulation Model 2 Parameter Sweep Specifications.**

| Parameter | Values |
|---|---|
| Observed Variables | 100 |
| Effect Magnitude (x Variance) | 5 |
| Hidden Input Effects | 1, 2, 4, 8 |
| Effected Observed Variables per Hidden Input Effect | 5, 10 |
| Unlabeled Patients | 10,000 |
| Labeled Patients | 100, 200, 500, 1000, 2000 |
| Systematic Bias | 0.1 applied to 0.33 of patients |
| DA Hidden Nodes | 2, 4, 8 |

**Supp. Table 2. Simulation Model 3 Parameter Sweep Specifications.**

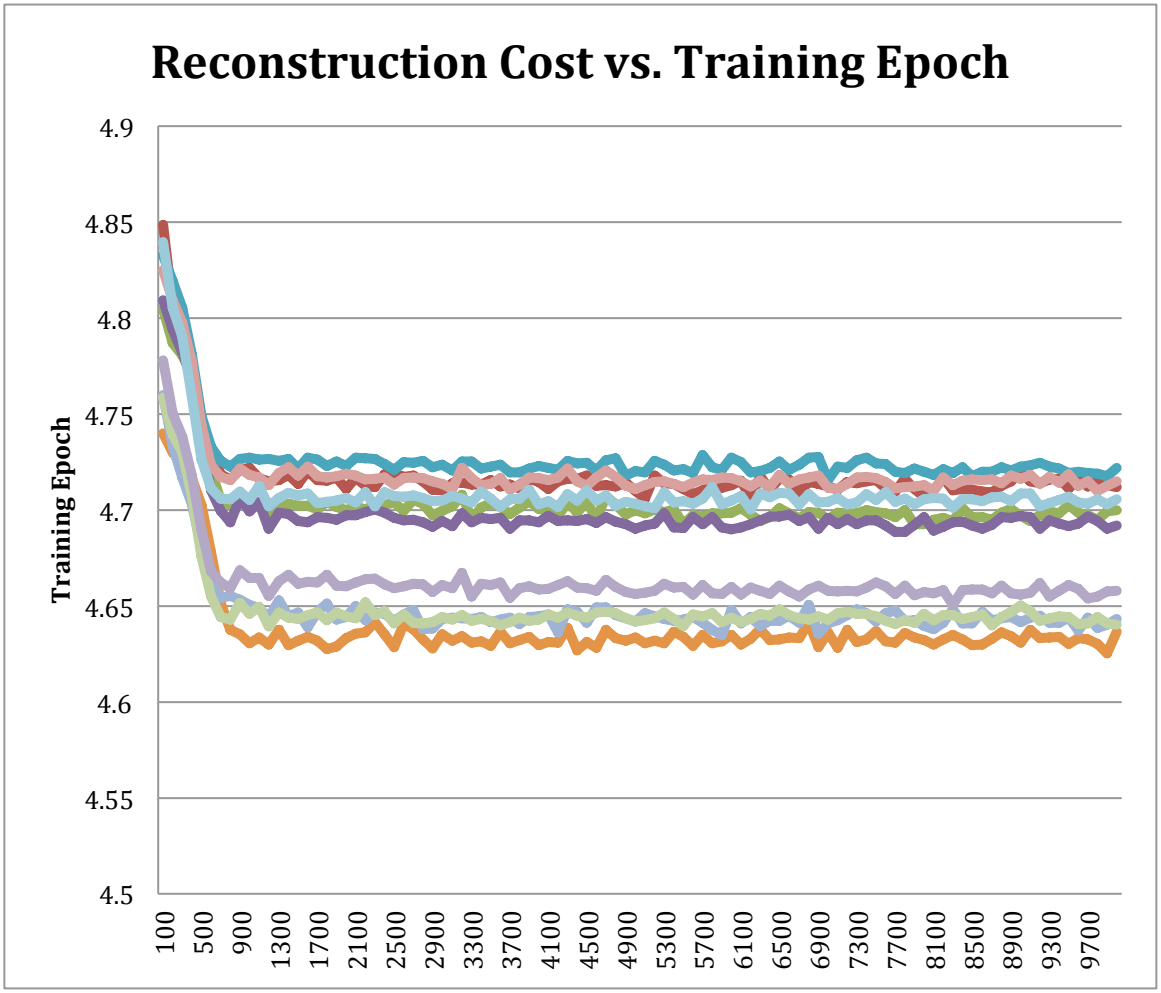
| Parameter | Values |
|---|---|
| Observed Variables | 100 |
| Effect Magnitude (x Variance) | 5 |
| Hidden Input Effects | 1, 2, 4, 8 |
| Effected Observed Variables per Hidden Input Effect | 10 |
| Unlabeled Patients | 10,000 |
| Labeled Patients | 100, 200, 500, 1000, 2000 |
| Systematic Bias | 0.1 applied to 0.33 of patients |
| DA Hidden Nodes | 2, 4, 8 |

**Supp. Table 3. Simulation Model 4 Parameter Sweep Specifications.**

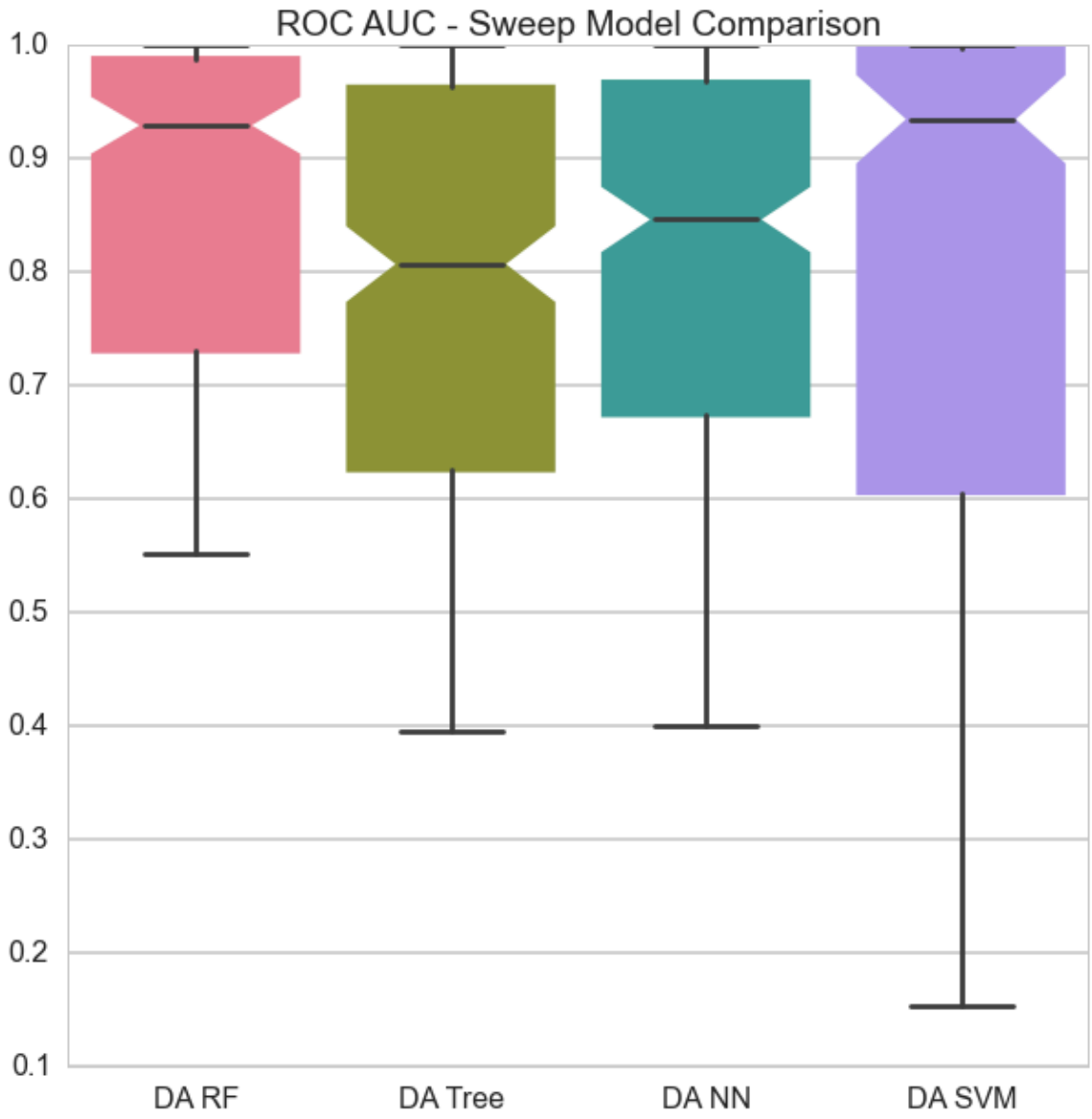
| **Parameter** | **Values** |
|---|---|
| Observed Variables | 100 |
| Effect Magnitude (x Variance) | 10 |
| Hidden Input Effects | 1, 2, 4, 8 |
| Effected Observed Variables per Hidden Input Effect | 5 |
| Unlabeled Patients | 10,000 |
| Labeled Patients | 100, 200, 500, 1000 |
| Systematic Bias | 0.1 applied to 0.33 of patients |
| DA Hidden Nodes | 2, 4, 8, 16 |

**Supp. Table 4. Missing Data, Simulation Model 1 Parameter Sweep Specifications.**

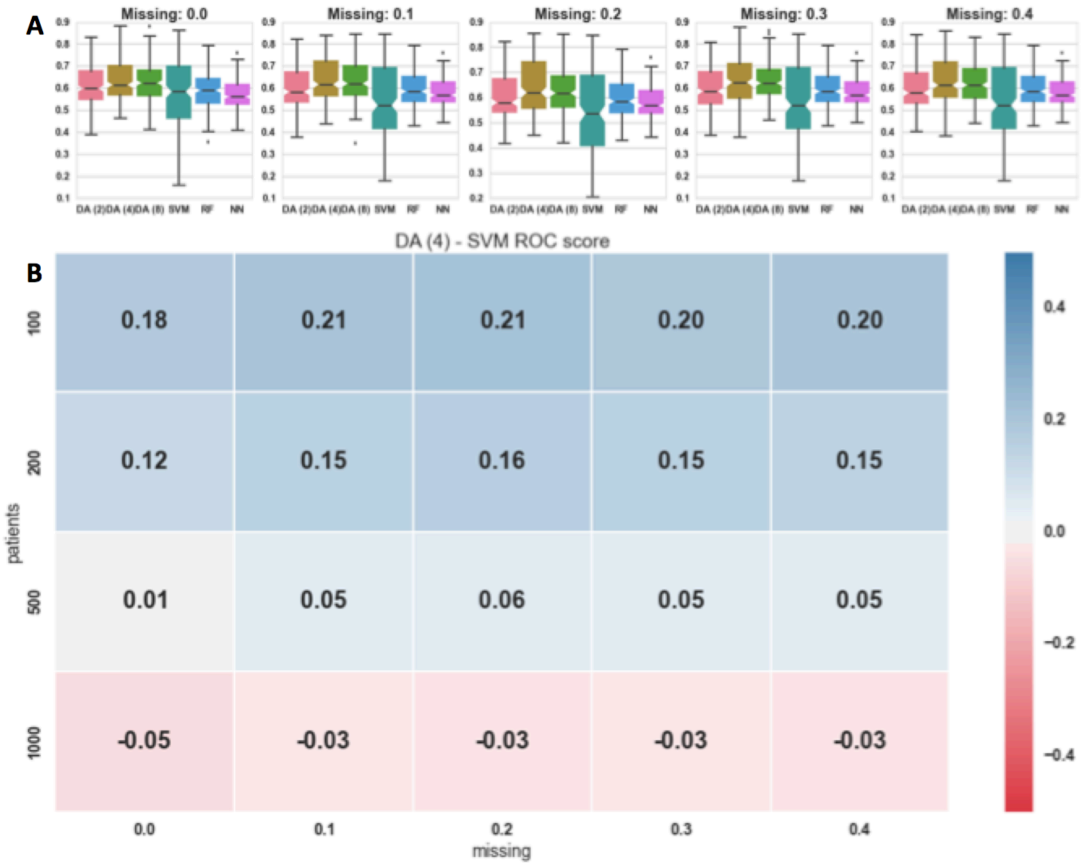
| **Parameter** | **Values** |
|---|---|
| Observed Variables | 100 |
| Effect Magnitude (x Variance) | 2 |
| Hidden Input Effects | 2, 4, 8, 16 |
| Effected Observed Variables per Hidden Input Effect | 10 |
| Unlabeled Patients | 10,000 |
| Labeled Patients | 100, 200, 500, 1000 |
| Systematic Bias | 0.1 applied to 0.33 of patients |
| DA Hidden Nodes | 2, 4, 8 |
| Missing Data | 0, 0.1, 0.2, 0.3, 0.4 |




**Supp. Figure 1.** Denoising Autoencoder Reconstruction Cost vs. Training Epochs




**Supp. Figure 2.** ROC AUC comparisons for traditional classifiers across model 1 with DA hidden nodes as inputs.




**Supp. Figure 3. A)** Classification AUC in relation to the amount of missing data under simulation model 3. **B)** Heatmap showing difference of DA and SVM in relation to supervised patient count and percent of missing data.