PertSpectra: Interpretable Matrix Factorization for Predicting Functional Impact of Genetic Perturbation Experiments

Seowon Chang[1], Anna Shcherbina[2], Tal Ashuach[2], Shahin Mohammadi[2], Stephanie See[2], Ninad Ranadive[2], Emily Fox[2], Navpreet Ranu[2]

1. Center for Computational and Molecular Biology, Brown University, Providence, Rhode Island, 02912, USA
2. Insitro, South San Francisco, California, 94080, USA
E-mail: seowon_chang@brown.edu, annashch@insitro.com, tal.ashuach@insitro.com

In drug discovery, measuring the effects of genetic perturbations is a powerful tool for studying unknown disease mechanisms, but biological interpretation of these effects, especially with the advent of screens involving combinatorial perturbations, remains challenging. To address limitations in current methodology we introduce PertSpectra, a guided triple matrix factorization that incorporates perturbation information and regularizes the model using a known gene-gene interaction graph prior to generate sparse, biologically relevant latent factors that capture perturbational effects. We evaluate PertSpectra on three single-cell RNA-seq datasets with both single and combinatorial genetic perturbations, measuring latent space interpretability, predictive ability on unseen combinations of observed perturbations, and stratification of functionally similar perturbations. We show that PertSpectra provides an integrated modeling approach to understanding combinatorial perturbation data in the context of drug discovery.

Keywords: Machine learning; perturb-seq; single cell RNA-seq; matrix factorization; combinatorial perturbations; functional impact; gene ontology

1. Introduction

Perturb-seq,[1] single-cell RNA sequencing performed on genetic perturbation screens, has emerged in recent years as a powerful tool in drug discovery for studying how targeted perturbations of individual genes affect cellular phenotype, providing insights into disease mechanisms. However, analyzing these experiments presents significant challenges, since the impact of each perturbation is often relatively small in comparison to the noise and sparsity in the data.[2] Furthermore, recent studies include multiple perturbations introduced into the same cells,[3] but few models attempt to untangle the effect of multiple perturbations. A variety of approaches have been proposed to tackle these challenges and assist in downstream tasks, such as interpreting perturbational effects, prediction of unseen combinations of perturbations, and predicting the cellular effects of unseen perturbations.

Autoencoder frameworks such as CPA[4] and GEARS[5] focus on capturing confounding variables such as cell type and batch effects to make predictions on unseen perturbations. sVAE+[6] and SAMS-VAE[7] are generative frameworks that incorporates a sparse shift mechanism, which capture interventions' sparse effects as explicit latent variables, improving the perturbation effect prediction. These models focus on predicting the transcriptional effects induced by perturbations, but quantifying the contributions of each gene to the latent dimensions remains difficult, making the latent spaces produced by these models challenging to interpret without extensive prior knowledge or expert annotations.

Other models prioritize learning an interpretable latent space, often by employing matrix factorization techniques,[8,9] which decomposes the observed cell-by-gene matrix into cell-by-factor and factor-by-gene matrices. This factor-by-gene matrix, denoted the gene loading matrix, captures the contribution of each individual gene to each learned factor in the latent space, thereby enabling biological interpretation of these factors. However, matrix factorization has several limitations in the context of interpreting perturb-seq data: (1) these models are largely designed for observational data, and need to be modified to support perturbational assays; (2) there are many ways of decomposing a matrix, and many models lack a biologically-informed inductive bias that would bias the decomposition towards biologically coherent factors, so these models often produce factors that remain difficult to interpret.

There are recent matrix factorization models that attempt to address some of these limitations. GSFA[10] is a Bayesian factorization method that incorporates perturbation information in a triple matrix factorization. Perturbation labels are encoded in a binary cell-by-perturbation design matrix, and the observed cell-by-gene matrix is decomposed into this cell-by-perturbation matrix and learned perturbation-by-factor and factor-by-gene matrices. Thereby GSFA learns an embedding space for perturbations, as opposed to individual cells. However, GSFA uses uninformative priors on the gene loading matrix that do not incorporate prior biological knowledge, so learned factors might not align with the underlying biology in the data. Additionally, GSFA is optimized with Gibbs-sampling, a computationally expensive algorithm that struggles with large-scale perturb-seq datasets.

Another recent development in this direction is Spectra,[9] a matrix factorization model designed for observational RNA-seq data that incorporates biologically informed inductive bias into the optimization procedure. Spectra maps the learned gene loading matrix to a gene-gene interaction graph, which is regularized by a known gene-gene interaction graph prior. In doing so, Spectra balances biological domain knowledge with data-driven relationships captured in the latent factors. However, Spectra does not support perturbation labels and was not designed for genetic screens.

Inspired by these contributions and motivated by the complementary limitations of existing approaches, here we introduce PertSpectra, a scalable guided triple matrix factorization method designed for genetic perturbation experiments, which leverages prior biological knowledge. We incorporate perturbational information as a fixed binary design matrix as in GFSA, and employ graph regularization on the gene loading matrix to guide the latent factors in a biologically-meaningful direction as in Spectra. Notably, this construction naturally supports additive combinatorial perturbations, as the perturbation design matrix can simply have multiple perturbations encoded for a given cell. PertSpectra is designed to (1) correctly predict both single and combinatorial perturbed cell expression profiles and (2) yield interpretable latent factors that yield insights into the functional impact of perturbations. Our main contributions can be summarized as:

- A triple matrix factorization formulation incorporating perturbation information as a fixed binary matrix.
- A functional protein graph[11] prior used during optimization to incorporate functional information into the learned latent factors.
- A set of qualitative and quantitative evaluations against baseline methods on three real-world perturb-seq datasets, one of which was generated in-house, to demonstrate state-of-the-art performance. We introduce an interpretability analysis as a part of these benchmarks.

We evaluate PertSpectra on three perturb-seq datasets including single and combinatorial perturbations, and demonstrate that Spectra produces biologically coherent, highly interpretable latent factors, while maintaining or improving performance on predictive tasks, compared with similar models. We benchmark PertSpectra against scETM, a single cell topic model, and GSFA, a perturbation-focused matrix factorization, to show the benefits of both incorporating perturbation information and inductive bias to model genetic perturbation experiments.

2. Background and Problem Setting

2.1. *Data and Preprocessing*

Three perturb-seq datasets were analyzed: Norman[3] (growth-pathway specific combinatorial knockouts in the K562 cell line), Replogle[2] (genome wide, CRISPRi in the K562 cell line), and an in-house dataset (combinatorial knockouts in the A549 cell line along the TNFa and IL1B mediated signaling along the NFκB pathway). The gene-by-sample matrices for each individual dataset were filtered to remove cells with low counts and genes with expressed in few cells and then log-normalized with scanpy[12] (full details in appendix A). The top 5000 highly variable genes across all samples were then selected. The identity of the perturbation for each cell was encoded in a binary matrix $\boldsymbol{H}$, where $\boldsymbol{H}_{ij} = 1 \iff$ gene j was perturbed in cell i. PertSpectra also incorporates prior knowledge from the StringDB[11] knowledge graph of known protein-protein interactions. We filtered to high-confidence, validated interactions and subset the graph to genes measured in the perturb-seq experiments (full details in appendix A).

Table 1. Evaluation datasets

Dataset	Type	# Cells	# Perts
In-house	Combinatorial	65648	182
Norman	Combinatorial	101484	235
Replogle	Single	105943	517

2.2. *Training Setup*

For the combinatorial perturbation datasets (in-house and Norman), we evaluated model performance on unseen combinations of observed singleton perturbations. Therefore, we held out 30% of combinatorial perturbations and reserved all cells that received those combinatorial perturbations for the test set. We then further divided the rest of the cells into 80-20 train/validation sets, proportionally to the perturbations received.

For the single perturbation datasets (Replogle), we wanted to evaluate model performance on unseen cells for seen perturbations. Therefore, data was divided into a 80-20 train/test split by cell, and the training split was further divided into a 80-20 train/validation split.

2.3. *Spectra*

As PertSpectra utilizes several components of Spectra, it stands to give a brief overview of the Spectra model.

A perturb-seq experiment yields an expression matrix $\mathbf{X}$ of dimensions $n \times g$, where n is the number of cells and g is the number of genes. Spectra learns a gene loading matrix $\boldsymbol{\theta}$ with dimensions $z \times g$, and cell loading matrix $\boldsymbol{\alpha}$ with dimensions $n \times z$ where z is the number of latent factors. The cell loading matrix $\boldsymbol{\alpha}$ is restricted to be strictly positive, under the assumption that a transcriptional profile of a cell is the sum of a set of active pathways, and negative pathway activation is difficult to interpret in a biological setting. This gives the decomposition

$$\mathbb{E}[\mathbf{X_{ij}}] = \boldsymbol{\alpha}_i^\intercal \boldsymbol{\theta}_j, \tag{1}$$

where $\mathbf{X_{ij}}$ is the expression of gene j in cell i, $\boldsymbol{\alpha}_i \in \mathbb{R}_+^z$ is the cell loading for cell i, and $\boldsymbol{\theta}_j \in \Delta^z$ (where Δ^z is the set of positive z-vectors that sum to 1) is the gene loading for gene j.

Spectra also addresses technical variation that may be present in the data. Certain genes that are involved in basic cellular functions are often highly expressed, while other important genes, such as transcription factors, may be lowly expressed. Matrix factorization methods may give higher weight to genes in the former group and less weight on the latter group, leading to poor latent factors. Spectra corrects for this by introducing a learned parameter ν_j as a gene-specific scaling factor, bounded from below by a hyperparameter δ. This allows the model to capture high expression and variability of specific genes, without increasing the weights of gene factors and giving too much or too little importance to specific genes. Thus, the base model of Spectra is given as

$$\mathbb{E}[\mathbf{X_{ij}}] = (\nu_j + \delta)\boldsymbol{\alpha}_i^{\intercal}\boldsymbol{\theta}_j, \tag{2}$$

The objective function to learn Spectra's parameters $\boldsymbol{\Theta} = \{\alpha, \theta, \nu\}$ is given by

$$\mathcal{L}(\boldsymbol{\Theta}) = \lambda\mathcal{L}_{recon}(\boldsymbol{\Theta}) + \mathcal{L}_{graph}(\boldsymbol{\Theta}), \tag{3}$$

where $\mathcal{L}_{recon}$ is the reconstruction loss, $\mathcal{L}_{graph}$ is the penalty term on the gene loading matrix $\boldsymbol{\theta}$, and λ a hyperparameter controlling the balance between the two loss terms. We give a brief overview of the intuition and formulation of the components of the loss function.

For $\mathcal{L}_{recon}$, Spectra uses the Poisson log likelihood

$$\mathcal{L}_{recon}(\boldsymbol{\Theta}) = -\mathbf{X}\log(\hat{\mathbf{X}}(\boldsymbol{\Theta})) + \hat{\mathbf{X}}(\boldsymbol{\Theta}), \tag{4}$$

where $\hat{\mathbf{X}}$ is Spectra's estimated gene expression from its learned parameters $\boldsymbol{\Theta}$. Spectra chooses this loss term as this function derived from the Poisson distribution has been widely used for modeling single cell RNA-seq counts.[8,13] Furthermore, it does not overly weight highly expressed or lowly expressed genes, as the loss scales according to the expression values of $\mathbf{X}$. Thus, this serves as a reasonable objective function for reconstructing the gene expression.

For $\mathcal{L}_{graph}$, we are first given a prior knowledge adjacency matrix $\mathbf{A}$, where edge $\mathbf{A}_{ij} = 1$ if there is a direct biological relationship between genes i and j. Spectra generatively defines its own gene-by-gene adjacency matrix as a weighted inner product of $\boldsymbol{\theta}$ with itself, with parameters that allow for random edge generation/deletion:

$$\mathbb{P}[\mathbf{A}_{ij} = 1] := (1-\kappa)(1-\rho)\langle\boldsymbol{\theta}_i, \boldsymbol{B}\boldsymbol{\theta}_j\rangle + \kappa(1-\rho), \tag{5}$$

where $\boldsymbol{B}$ is a learned scaling factor, κ is an edge creation rate, and ρ is an edge deletion rate. Spectra lets $\mathbb{P}[\mathbf{A}_{ij} = 1]$, or the the probability of an edge between genes i and j, be correlated to similarity of the factor distribution between the two genes captured by $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$. $\boldsymbol{B}$ is a scaling factor that prevents the regularization from discouraging discovery of new gene-gene relationships. If genes i and j show clear relationship from the data but not in the prior information, a simple inner product $\langle\boldsymbol{\theta}_i, \boldsymbol{B}\boldsymbol{\theta}_j\rangle$ would discourage the model from learning this new discovery. Thus, $\boldsymbol{B}$ prevents heavy bias towards the prior information graph. Furthermore, parameters κ and ρ allow for the model to add or delete edges if the expression data that drives $\boldsymbol{\theta}$ deviates from the relationships in the prior graph. Therefore, cliques and communities within this generated graph, that represent gene groups with shared latent factors, should reflect functional groupings of genes. By regularizing this factor-dependent gene-gene graph against a prior graph derived from experiments, Spectra constrains the learned $\boldsymbol{\theta}$ to yield biologically interpretable factors. Therefore, the graph regularization loss is the negative log likelihood:

$$\mathcal{L}_{graph}(\boldsymbol{\Theta}) = -\mathbf{A}_{ij}\log(\mathbb{P}[\mathbf{A}_{ij}=1]) - (1-\mathbf{A}_{ij})\log(1-\mathbb{P}[\mathbf{A}_{ij}=1]), \tag{6}$$

Spectra optimizes $\mathcal{L}(\boldsymbol{\Theta})$ using the Adam optimizer, with a learning rate schedule of [1.0, 0.5, 0.1, 0.01, 0.001, 0.0001] and maximum number of epochs set at 10,000.

3. Methods

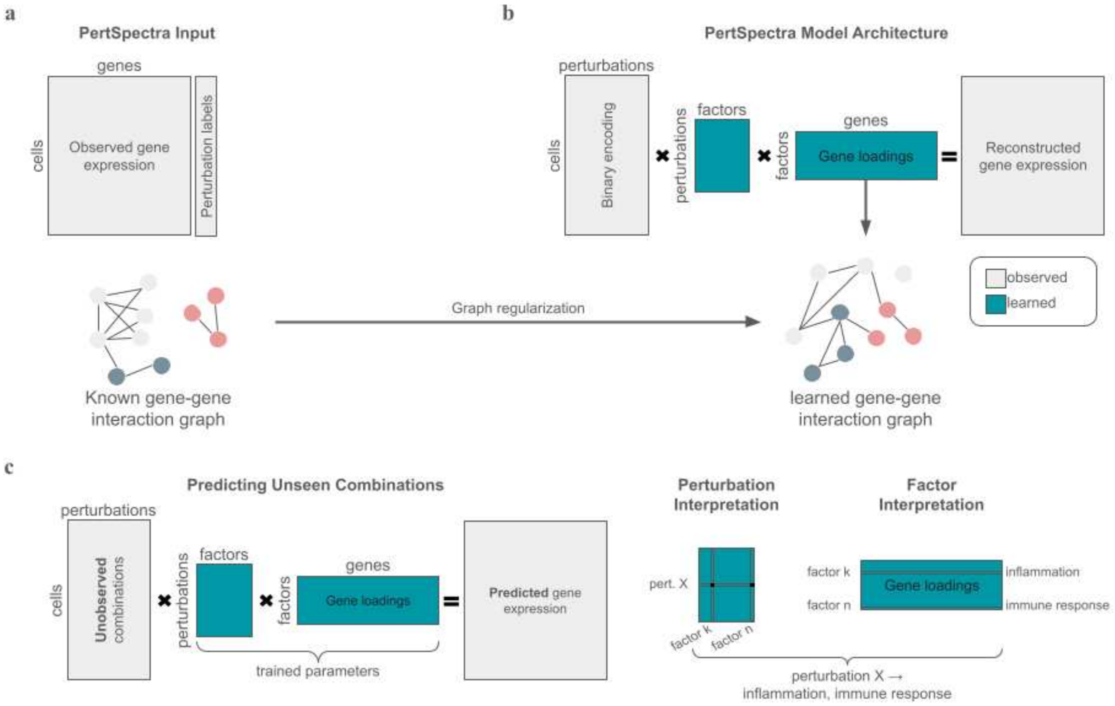


Fig. 1. a. Input for PertSpectra, b. Illustration of PertSpectra model, c. Downstream applications of a trained PertSpectra model

3.1. *PertSpectra Architecture and Training Procedure*

PertSpectra is a guided triplet matrix factorization model that incorporates both perturbation information from the experiment as well as biological knowledge graphs. Specifically, we modified Spectra, a single-cell matrix factorization method, to account for perturbation information. A perturbation sequencing experiment outputs a gene expression matrix $\mathbf{X}$ with dimensions $n \times g$ and perturbation labels vector $\boldsymbol{h}$ of length n, where n is the number of cells and g is the number of genes. We can transform h into a binary matrix $\boldsymbol{H}$ of dimensions $n \times p$, where p is the number of unique perturbations, such that $\boldsymbol{H}_{i,j} = 1 \iff \boldsymbol{h}_i = j$. Unlike

Spectra, which decomposes the gene expression matrix into two matrices, PertSpectra has the following factorization:

$$\mathbb{E}[\mathbf{X_{ij}}] = (\nu_j + \delta)\boldsymbol{H}_i^\intercal \boldsymbol{P}\boldsymbol{W}_j \tag{7}$$

where $\boldsymbol{H}_i \in \{0,1\}^p$ is the binary encoding of the known perturbation design vector for cell i, $\boldsymbol{P}_i$ is a $p \times z$ perturbation loading matrix, and $\boldsymbol{W}_j \in \Delta^z$ (where Δ^z is the set of positive z-vectors that sum to 1) is the gene loading vector for gene j, and z is the latent dimension.

Since the $\boldsymbol{H}$ matrix is fixed, PertSpectra learns estimates of the $\boldsymbol{P}$ and $\boldsymbol{W}$ matrices. We can then treat $\boldsymbol{W}$ as a loading matrix (analogous to Principal Components Analysis), which provides the ability to study each factor based on the loadings of the genes in $\boldsymbol{W}$. The graph regularization biases these loadings towards capturing known gene-gene relationships, further increasing the ability to interpret the results. Since the $\boldsymbol{P}$ matrix has dimensions $p \times z$, or factors per perturbation, this provides a per-perturbation relation to each factor. Therefore, this factorization gives us (1) biologically interpretable gene loadings and (2) an association of the latent factors to perturbations, rather than individual cells.

To optimize this factorization, we consider a reconstruction loss term between the predicted and observed normalized expression and a graph regularization penalty on $\boldsymbol{W}$. Formally, the objective function for learning PertSpectra's parameters $\boldsymbol{\Theta}$ is

$$\mathcal{L}(\boldsymbol{\Theta}) = \lambda \mathcal{L}_{recon}(\boldsymbol{\Theta}) + \mathcal{L}_{graph}(\boldsymbol{\Theta}). \tag{8}$$

As in the original Spectra model, we use Poisson log likelihood for the reconstruction loss:

$$\mathcal{L}_{recon} = -\mathbf{X}\log(\hat{\mathbf{X}}) + \hat{\mathbf{X}}, \tag{9}$$

where $\hat{\mathbf{X}}$ is PertSpectra's estimated gene expression from its learned parameters. To account for imbalance in the number of cells receiving a certain perturbation in the training set, we weight the reconstruction loss in the following manner: For cell i receiving perturbation $\boldsymbol{h}_i$, we weight its reconstruction loss by the reciprocal of the number of cells that received perturbation $\boldsymbol{h}_i$:

$$\boldsymbol{\eta}_i = \frac{1}{|C_i|}, \tag{10}$$

where $C_i = \{k : \boldsymbol{h}_k = \boldsymbol{h}_i, 1 \leq k \leq n\}$. Thus, the reconstruction loss $\mathcal{L}_{recon}(\boldsymbol{\Theta})$ becomes

$$\mathcal{L}_{recon}(\boldsymbol{\Theta}) = \boldsymbol{\eta}\big[-\mathbf{X}\log(\hat{\mathbf{X}}(\boldsymbol{\Theta})) + \hat{\mathbf{X}}(\boldsymbol{\Theta})\big], \tag{11}$$

where $\boldsymbol{\eta} \in \mathbb{R}_+^n$ is a vector of the cell weights as described above.

For the graph penalty loss, we use the graph regularization term used by Spectra. Given an input prior graph $\mathbf{A}$, we generatively defined PertSpectra's adjacency matrix as

$$\mathbb{P}[\mathbf{A}_{ij} = 1] := (1-\kappa)(1-\rho)\langle \boldsymbol{W}_i, B\boldsymbol{W}_j\rangle + \kappa(1-\rho), \tag{12}$$

with κ and ρ representing the same learned edge creation/deletion parameters as described in Spectra. Then, we define the graph penalty function as

$$\mathcal{L}_{graph}(\boldsymbol{\Theta}) = -\mathbf{A}_{ij}\log(\mathbb{P}[\mathbf{A}_{ij} = 1]) - (1-\mathbf{A}_{ij})\log(1-\mathbb{P}[\mathbf{A}_{ij} = 1]), \tag{13}$$

We optimize PertSpectra's objective function using AdamW, with a learning rate schedule of [1.0, 0.5, 0.1, 0.01, 0.001, 0.0001] and maximum number of epochs set at 10,000.

3.2. *Benchmark Model Training Procedure*

We benchmarked PertSpectra against two models designed specifically for interpretability of gene expression data: scETM and GSFA. scETM[14] is a topic model designed for learning a triplet factorization of cell-by-topic, topic-by-embedding, and embedding-by-gene matrices. The learned topics should be biologically relevant, and the matrix factorization yields interpretability for analyzing the learned topic and gene embeddings. While designed for observational single cell RNA-seq datasets, scETM's topics should provide similar interpretability analysis and biologically-relevant latent embeddings.

GSFA[10] is a Bayesian factorization model that, similarly to PertSpectra, factorizes the gene count matrix into cell-by-perturbation, perturbation-by-embedding, and embedding-by-gene matrices, where the first matrix is given from a perturb-seq experiment and the latter two matrices are learned. Unlike PertSpectra, GSFA imposes mixture priors on the loading matrices rather than a graph regularization, and parameter optimization is achieved via Gibbs Sampling.

We chose these models as benchmarks as a form of ablation for different aspects of our proposed model. scETM does not leverage perturbation information in learning the loading matrices, so improvement over this model would demonstrate the benefits of explicitly incorporating perturbation information in learning interpretable latent factors. GSFA is a similar guided factorization model, but does not incorporate prior knowledge in regularizing the learned factors; thus, improvement over GSFA would demonstrate the importance of leveraging prior information to yield more interpretable factors.

To train scETM, we used the gene by sample count matrix as input, as scETM does not account for perturbation information. The expression matrix was filtered with the same steps as mentioned above, and the counts were normalized as described by the scETM paper (raw count of the gene divided by the total counts in each cell). The train-test splits as described above for PertSpectra were used to split the data for scETM. To train GSFA, the expression matrix was normalized as described in their methodology. Because of memory constraints of the Gibbs Sampling algorithm, GSFA was trained on subsets of the in-house and Norman datasets. We downsampled the train-test splits used for PertSpectra, proportionally to the perturbation labels. GSFA was not able to train on the Replogle dataset, and we encountered persistent out-of-memory issues when training on more than 20,000 cells on an EC2 instance with 128 Gb of RAM and 32 vCPUs.

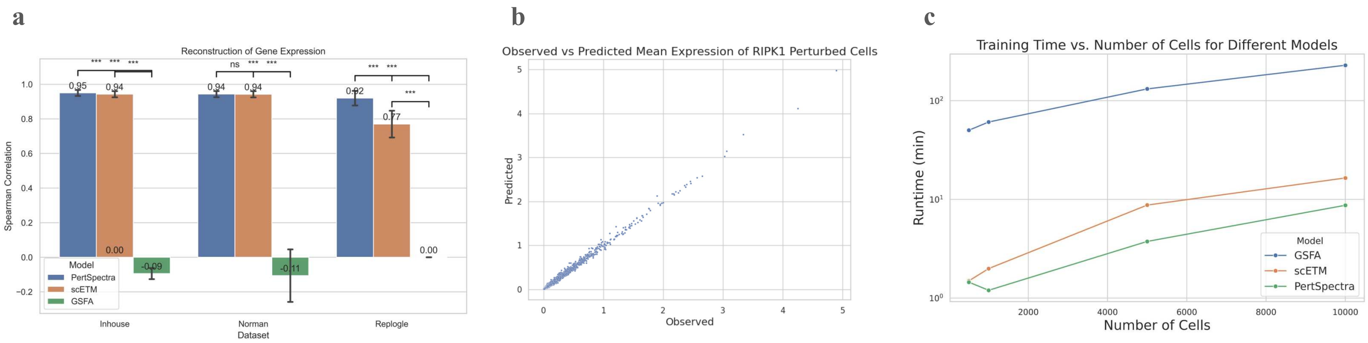


Fig. 2. a. Spearman correlation between predicted and observed mean gene expression by perturbation, b. Predicted vs observed mean expression for the perturbation RIPK1 (inhouse), c. Training runtime vs. number of cells across different models

4. Evaluation

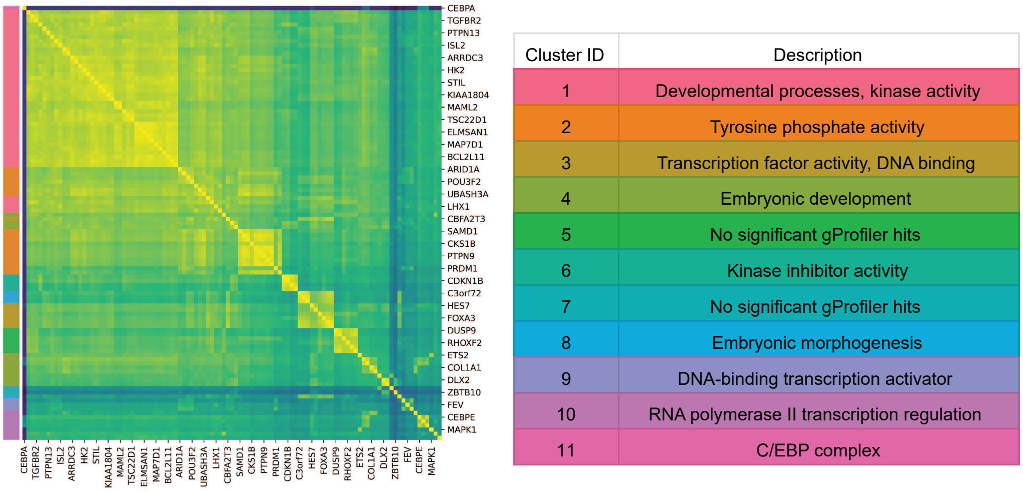
4.1. *Predicting Unseen Combinations of Perturbations*

We first evaluated the models' ability to reconstruct gene expression levels for unseen cells or unseen combinations of perturbations. We computed Spearman correlation on the observed vs. predicted mean normalized gene expression for each perturbation or combination. This measures the ability of the model to correctly predict the gene expression profile of cells with unseen combinations of seen perturbations for the combinatorial datasets (in-house and Norman), and the ability of the model to correctly predict the gene expression profiles of unseen cells with seen perturbations for the single perturbation datasets (Replogle). We see that PertSpectra and scETM perform comparably across the in-house and Norman datasets (mean correlations of 0.95 and 0.94 for in-house, 0.94 for Norman for both methods), and PertSpectra achieves better performance than scETM on the Replogle dataset (0.92 and 0.77, respectively), with the error bars representing the standard deviation of the distribution of correlations and significance shown with ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, and ns: no significance (Figure 2). GSFA's poor performance may be attributed either to the model's focus on interpretability rather than reconstruction, or due to the down-sampling procedure we used to avoid out-of-memory issues.

We also evaluated the training runtime of these models, as they varied drastically. We found that PertSpectra scaled best as experiment size (number of cells sequenced) increased, as it uses a gradient-based optimization with GPU capabilities. GSFA performed poorly in this metric, due to its use of Gibbs Sampling to optimize its parameters and lack of GPU integration.

4.2. *Biological Structure in the Perturbation Embedding*

Next, to evaluate how well the perturbation latent factors $\boldsymbol{P}$ capture the underlying biology in the data, we reasoned that perturbations that target functionally related genes would have similar latent factors. Importantly, the model does not receive information on which gene



Cluster ID	Description
1	Developmental processes, kinase activity
2	Tyrosine phosphate activity
3	Transcription factor activity, DNA binding
4	Embryonic development
5	No significant gProfiler hits
6	Kinase inhibitor activity
7	No significant gProfiler hits
8	Embryonic morphogenesis
9	DNA-binding transcription activator
10	RNA polymerase II transcription regulation
11	C/EBP complex

Fig. 3. Hierarchical clustering on distance matrix computed on PertSpectra's perturbation loading matrix shows meaningful biological structure

is targeted by each perturbation, and therefore cannot leverage the graph prior provided to influence the embedding space, ensuring that distances in the perturbation embedding are reflecting signals gleaned from the data, and not from the prior graph. As an initial exploratory task, we first qualitatively analyzed the $\boldsymbol{P}$ matrix from the Norman dataset. To do so, computed the pairwise Euclidean distance matrix, performed hierarchical clustering, and tested the resulting clusters for enrichment of genes that share a biological pathway, using Gene Ontology (GO) annotations and a software package gprofiler.[15] We then filter significant GO terms (p-values ¡ 0.05) and picked out the top 1 or 2 most common GO terms that show up for each cluster to indicate the prevalent biological functions assigned to that cluster. We see that biological functions clustered together, indicating that information about the perturbations' biological impacts were captured in the perturbation loading matrix structure.

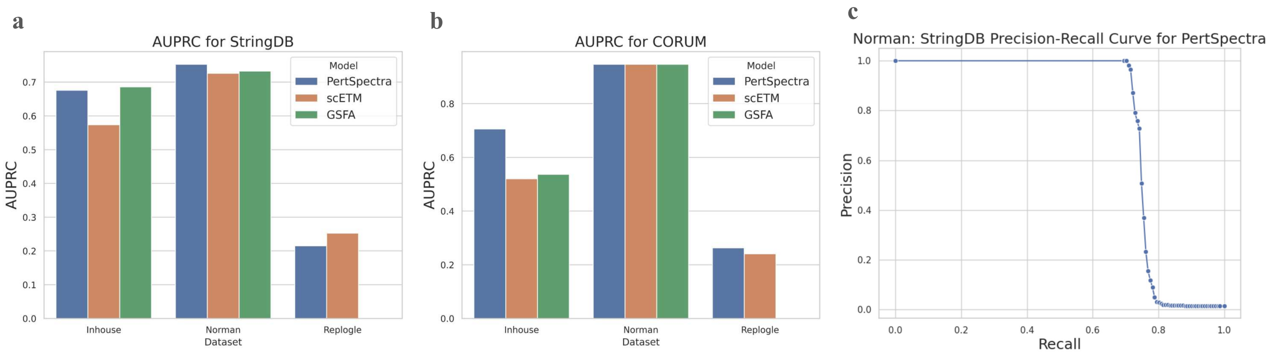


Fig. 4. a. Area under the Precision-recall curve (AUPRC) for StringDB known relationships, b. AUPRC for CORUM known relationships, c. Precision-recall curve for PertSpectra on the Norman dataset

Having established that our model is capturing biological information, we then conducted

a quantitative analysis comparing against scETM and GSFA. We used StringDB[11] and CORUM[16] as the true labels, where the pair of genes A and B have a label of 1 if that relationship exists in the database, and 0 otherwise. To generate pairwise similarity between perturbations from the models, we compute pairwise euclidean distance and normalize it to the range $[0, 1]$ using the learned perturbation loading matrices, either learned directly in the case of PertSpectra and GSFA ($\boldsymbol{P}$ and $\boldsymbol{\alpha}$, respectively), or for scETM the aggregated perturbation loading matrix. We take ($1-$ distance matrix) to generate pairwise similarity. Finally, we computed the AUPRC between pairwise similarity of perturbations in the learned latent space and prior gene-gene relationship labels from knowledge bases. We find that PertSpectra performs competitively across the three datasets, suggesting that the learned factors across the three models capture similar levels of biological relationships between the perturbations (Figure 4).
One limitation of this analysis is that because of the use of a similarity metric, rather than a classifier, to identify gene-gene relationship labels, we found that the distribution of the labels may not be well calibrated. For example, as seen in Figure 4c., the precision-recall curve is not well calibrated, indicating that this recall metric may be flawed. Therefore, analyzing the interpretability of the learned latent substructure required the following more in-depth analysis.

4.3. *Interpretability of Loading Matrices*

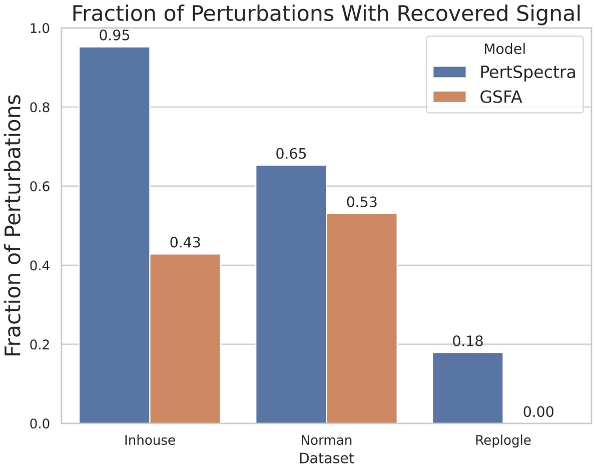


Fig. 5. Recovery of known biological functions from hypergeometric test interpretability analysis

We evaluated the interpretability of the loading matrices $\boldsymbol{P}$ and $\boldsymbol{W}$ with the following analysis. We reason that the matrix factorization learned "good" latent factors if 1) the gene loading weights indicate latent variables represent different biological functions and 2) the perturbation loadings associate perturbations to those biologically representative latent variables. From the $z \times g$ loading matrix $\boldsymbol{W}$, we can extract the biological functions represented by each latent factor by applying Gene Set Enrichment Analysis (GSEA)[17] on each loading vector $\boldsymbol{W}_i$ for $i \in 1, 2, ..., z$. GSEA is a statistical analysis that identifies over-represented sets of genes that correspond to biological phenotypes. Thus, since the loading weights $\boldsymbol{W}$ indicate the contribution of each gene to each loading vector, GSEA performed on the $\boldsymbol{W}_i$ captures the biological functions represented by each latent factor. We can then observe the $\boldsymbol{P}$ matrix, and for each perturbation retrieve the top k factors associated with each perturbation. Thus, we can observe the biological functions assigned by the model to each perturbation. To quantify the quality of recall of these biological functions, we can observe the overlap with a known, annotated set of gene sets that contain the perturbed gene. We then perform a hypergeometric test on the overlap of the model's biological functions and the known biological functions, with significance (p-value < 0.001) indicating that the model can recall the correct biological functions non-randomly. The hypergeometric test accounts for the sizes of the model's GO term set and the known GO term set, which controls for a naive model assigning arbitrary number of GO terms to a perturbation to get a good recall. For each perturbation, we can perform this hypergeometric test and calculate the fraction of perturbations in which a model can confidently recall gene sets. Full details of the analysis available in Appendix C.

In the first step of this analysis, scETM did not yield statistically significant GO terms for its factors, suggesting that scETM's gene loading matrix did not sufficiently capture known biological patterns. Thus, scETM could not be included in this analysis. In comparing PertSpectra to GSFA, we find significant improvement in perturbation recovery via the learned loading matrices (Figure 5). The fraction of perturbations that have significant p-values from the hypergeometric test is much higher for PertSpectra in both the in-house (0.95 vs 0.43) and Norman (0.65 vs 0.53) datasets. This gives further confidence to PertSpectra's ability to capture biological function in its latent factors.

5. Discussion and Future Work

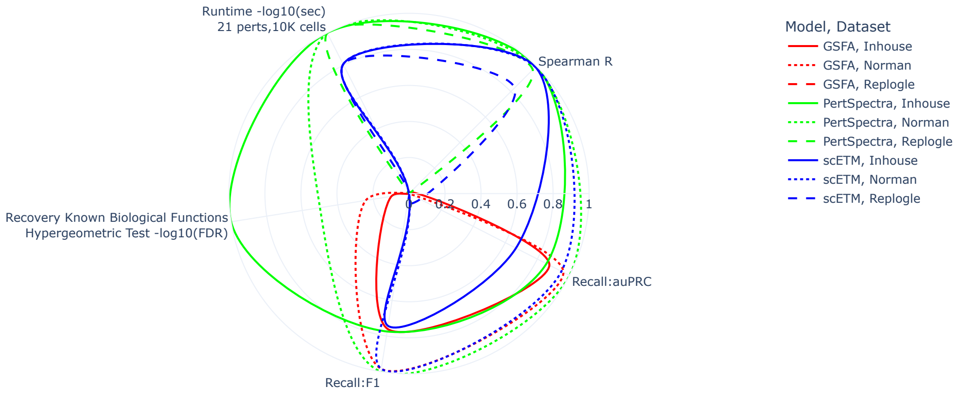


Fig. 6. Radar plot summarizing model performance across metrics/downstream analyses

In this paper, we introduce PertSpectra, a guided matrix factorization framework designed for perturb-seq data that incorporates biological domain knowledge to learn interpretable, biologically coherent representations of cellular response to perturbations. PertSpectra regularizes the gene loading matrix with a known gene-gene interaction graph prior, introducing crucial inductive bias to guide the factorization towards biologically coherent latent factors, thereby increasing the interpretability of the measure responses. Furthermore, PertSpectra's factorization incorporates a binary design matrix to encode the perturbation labels, a construction which naturally extends beyond single perturbations to capture additive effects of seen or unseen combinatorial perturbations. Overall, PertSpectra generates highly interpretable latent factors, accurately predicts associations between perturbations and biological processes, and scales seamlessly into larger datasets. In addition to interpretability, we demonstrate that PertSpectra is competitive with or outperforms other methods across a variety of datasets and tasks, including on held-out cells and unseen combinations of perturbations (Figure 6). Genetic perturbation experiments hold the potential for discovering key biological mechanisms driving cellular phenotypes - we hope that PertSpectra will aid in the computational analysis of these experiments to gain deeper insights into novel biology.

We note that while PertSpectra can associate perturbations with biological functions in the form of gene-set signatures, this relationship does not indicate a causal relationship and should not be construed as such. There is extensive work in learning causal mechanisms from perturb-seq data,[18–20] which is a separate effort from interpretability frameworks for identifying associations from perturbations to functional processes. Furthermore, due to the linear nature of matrix factorization, PertSpectra is limited to interpreting effects of single perturbations or additive effects of combinatorial perturbations. Further research is required for interpretation of non-additive synergistic combinations of perturbations, and would be important in several applications, including target discovery of combinatorial therapeutics.

6. Code Availability

The code for implementations, analysis, and figures are available at https://github.com/insitro/PertSpectra.

7. Data Availability

The model weights, precomputed metrics, the inhouse dataset, and reference datasets used in downstream analyses are available at s3://pert-spectra/ and on zenodo https://zenodo.org/records/14740509.

Acknowledgements

We thank Angela Pisco for discussions throughout the duration of this project which greatly improved this work. We also thank members of the Data Science and Machine Learning department at Insitro for providing constructive feedback on the results presented in this work. Disclosures: This work was performed while Seowon Chang was an intern at Insitro. Anna Shcherbina, Tal Ashuach, Shahin Mohammadi, Stephanie See, Ninad Ranadive, Emily Fox, and Navpreet Ranu are employees of Insitro.

References

1. A. Dixit *et al.*, Perturb-seq: Dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens, *Cell* **167**, 1853 (2016).
2. J. Replogle *et al.*, Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq, *Cell* **85**, 2559 (2022).
3. T. Norman *et al.*, Exploring genetic interaction manifolds constructed from rich single-cell phenotypes, *Science* **365**, 786 (2019).
4. M. Lotfollahi, A. Klimovskaia Susmelj, C. De Donno, L. Hetzel, Y. Ji, I. L. Ibarra, S. R. Srivatsan, M. Naghipourfar, R. M. Daza, B. Martin *et al.*, Predicting cellular responses to complex perturbations in high-throughput screens, *Molecular Systems Biology* , p. e11517 (2023).
5. Y. Roohani, K. Huang and J. Leskovec, Predicting transcriptional outcomes of novel multigene perturbations with gears, *Nature Biotechnology* (2023).
6. R. Lopez, N. Tagasovska, S. Ra, K. Cho, J. K. Pritchard and A. Regev, Learning causal representations of single cells via sparse mechanism shift modeling, *Conference on Causal Learning and Reasoning* (2023).
7. M. Bereket and T. Karaletsos, Modelling cellular perturbations with the sparse additive mechanism shift variational autoencoder, in *Advances in Neural Information Processing Systems*, 2023.
8. H. M. Levitin, J. Yuan, Y. L. Cheng, F. J. Ruiz, E. C. Bush, J. N. Bruce, P. Canoll, A. Iavarone, A. Lasorella, D. M. Blei and P. A. Sims, De novo gene signature identification from single-cell rna-seq with hierarchical poisson factorization, *Molecular Systems Biology* **15**, p. e8557 (2019).
9. R. Kunes *et al.*, Supervised discovery of interpretable gene programs from single-cell data, *Nature Biotechnology* **42**, 1084– (2024).
10. Y. Zhao, K. Luo, L. Liang, M. Chen and X. He, A new bayesian factor analysis method improves detection of genes and biological processes affected by perturbations in single-cell crispr screening, *Nature Methods* **20**, 1693– (2023).
11. D. Szklarczyk, R. Kirsch, M. Koutrouli, K. Nastou, F. Mehryary, R. Hachilif, A. L. Gable, T. Fang, N. T. Doncheva, S. Pyysalo, P. Bork, L. J. Jensen and C. von Mering, The STRING database in 2023: protein-protein association networks and functional enrichment analyses for any sequenced genome of interest, *Nucleic Acids Res.* **51**, D638 (January 2023).
12. F. A. Wolf, P. Angerer and F. J. Theis, SCANPY: large-scale single-cell gene expression data analysis, *Genome Biol.* **19** (December 2018).
13. E. Dann, N. C. Henderson, S. A. Teichmann, M. D. Morgan and J. C. Marioni, Differential abundance testing on single-cell data using k-nearest neighbor graphs, *Nature Biotechnology* **40**, p. 245–253 (September 2021).
14. Y. Zhao, H. Cai, Z. Zhang and Y. Li, Learning interpretable cellular and gene signature embeddings from single-cell transcriptomic data, *Nature Communications* **12** (2021).
15. U. Raudvere, L. Kolberg, I. Kuzmin, T. Arak, P. Adler, H. Peterson and J. Vilo, g:profiler: a web server for functional enrichment analysis and conversions of gene lists, *Nucleic Acids Research* **47**, p. W191–W198 (2019).
16. A. Ruepp, B. Brauner, I. Dunger-Kaltenbach, G. Frishman, C. Montrone, M. Stransky, B. Waegele, T. Schmidt, O. N. Doudieu, V. Stumpflen and H. W. Mewes, Corum: the comprehensive resource of mammalian protein complexes, *Nucleic Acids Research* **36**, p. D646–D650 (December 2007).
17. A. Subramanian *et al.*, Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles, *Proceedings of the National Academy of Sciences of the United States of America* **102**, 15545 (2005).
18. P. Brouillard, S. Lachapelle, A. Lacoste, S. Lacoste-Julien and A. Drouin, Differentiable causal

discovery from interventional data, in *Advances in Neural Information Processing Systems*, 2020.

19. R. Lopez, J.-C. Hütter, J. K. Pritchard and A. Regev, Large-scale differentiable causal discovery of factor graphs, in *Advances in Neural Information Processing Systems*, 2022.
20. A. Nazaret, J. Hong, E. Azizi and D. Blei, Stable differentiable causal discovery, *arXiv preprint arXiv:2311.10263* (2023).