Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

Mixed-Layer Deep Modeling of Genotypes and Cross-Tissue Expression Uncovers *Trans*-Eqtls

Shuo Yang[1], Dana Pe'er[2] and Itsik Pe'er[1,3,*]

[1]Computer Science Department, Columbia University, New York, 10027, USA,
[2]Computational Biology Program, Memorial Sloan Kettering Cancer Center, New York, 10065, USA and
[3]Systems Biology Department, Columbia University, New York, 10032, USA

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Modeling genetics of gene expression had been effective at highlighting *cis*-eQTLs, variants that control nearby transcripts. Yet, incorporation of long-range effects has been hampered by unfavorable statistical considerations. On the other end, expression alone has been modeled across tissues by decomposition into contributing factors, without any connection to genetics.
Results: We develop **MI**xed-**L**ayer **A**nalysis of **G**enetics and **E**xpression (MILAGE), a model that combines direct effects of *cis*-SNPs on nearby transcripts with *trans*-effects that control global factors of expression in a tissue-specific pattern. We develop judicious initialization of the model, followed by gradient descent learning. We present GPU-based implementation of the learner to enable computational feasibility in this otherwise intractably-large parameter space. We show the model to explain $> 59\%$ of test-set variation in GTEx data. The inferred genetically-regulated factors are consistent with expected tissue similarity.
Key words: *trans*-eQTLs, factor analysis, neural network, tensor decomposition, deep learning
Availability: The source code is available at https://github.com/morrisyoung/eQTL_MILAGE
Contact: itsik@cs.columbia.edu

1 Introduction

Efforts to study genetic association have largely failed to find large effects of common genetic variants on clinical human traits. In contrast, many such variants had been effectively discovered as modulating gene expression (Brem *et al.* (2002); Rockman and Kruglyak (2006); Cookson *et al.* (2009)). Such expression Quantitative Trait Loci (eQTLs) have been well characterized and shown to most obviously include large effects in *cis*, that are often shared across multiple tissues (Michaelson *et al.* (2009)). Large scale efforts to characterize eQTLs across tissues, and most prominently the Genotype-Tissue Expression Project (GTEx, Lonsdale *et al.* (2013)) had cataloged both *cis*-SNPs (Ardlie *et al.* (2015); Aguet *et al.* (2016)) as well as *trans*-effects of variants (Jo *et al.* (2016)). More elaborate models for gene expression (Gao *et al.* (2013); Gao *et al.* (2016); Zhao *et al.* (2016)) utilize the correlation structure in transcriptional patterns to model and predict expression.

Yet, current studies and methods suffer from notable limitations. First, analysis of variants in *trans* seeks effects of a single variant on a single gene, rather than seeking pervasive genomewide effects. Secondly, only linear *trans*-effects are considered. Moreover, *cis*- and *trans*-effects are considered only separately. Finally, methods struggle to scale up to the whole genome, and are forced to rely on preprocessing/pruning to resolve the computational constraints, especially for *trans* analysis.

In this work, we build **MI**xed-**L**ayer **A**nalysis of **G**enetics and **E**xpression (MILAGE) to tackle the problem. We show our modeling details to be effective in describing expression within the tensor of data across tissues times individuals times genes.

2 Methods

2.1 Modeling

We introduce and evaluate predictive graphical models (Fig. 1) of three types: a straightforward multiple linear model (ML), a nonlinear neural network model (NN), and a tensor-decomposition linear predictive model (TM). NN extends ML by allowing non-linear effects in genetic regulation of gene expression. TM is linear, like ML, but improves model complexity by more agressively assuming low rank across gene co-expression factors.

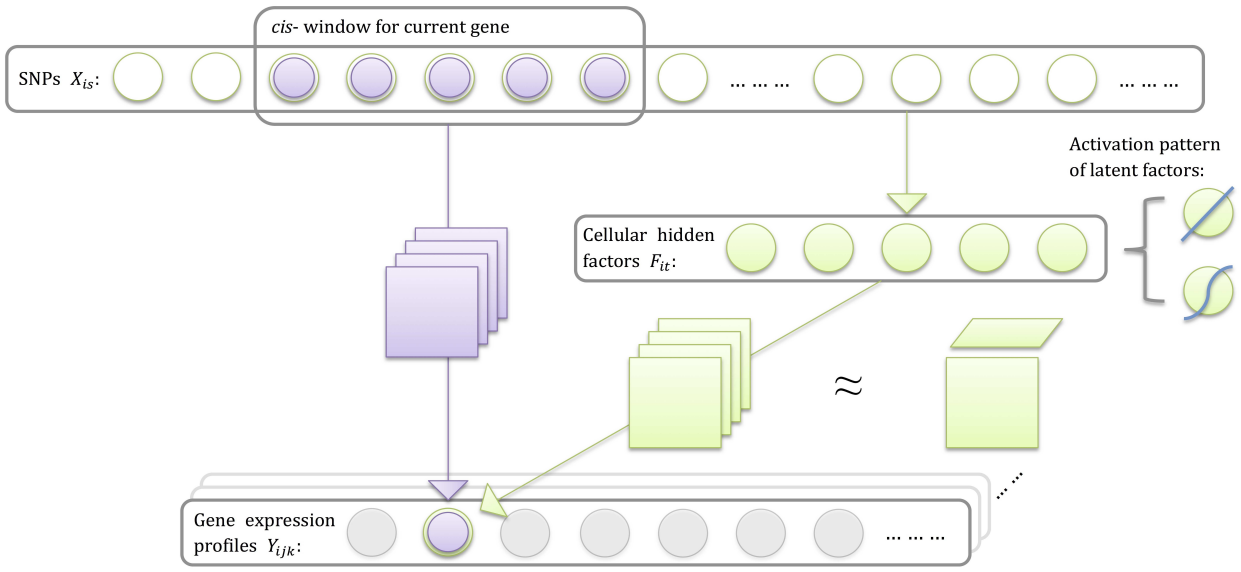

Fig. 1. MIxed-Layer Analysis of Genetics and Expression (MILAGE) to model expression tensors along with genetics across individuals and tissues. Gene expression is explained by cis-SNPs of each gene in a tissue-specific fashion (different purple pages), combined with genome-wide trans-effects through cellular hidden factors which has linear or nonlinear activation pattern. Cellular hidden factors perform regulation to gene expression in a tissue-specific fashion (different green pages), or in a tensor fashion where its tensor product with gene factor matrix and tissue factor matrix (the two extra green pages) constructs the factor effects for gene expression.

2.1.1 Linear and nonlinear factor modeling

We introduce notations to specify indices across the multiple dimensions of this problem as follows. The four dimensions that involve the input data include individuals indexed by $i = 1, \ldots, I$, genes indexed by $j = 1, \ldots, J$, tissues indexed by $k = 1, \ldots, K$ and SNPs indexed by $s = 1, \ldots, S$. The input data includes the observed variables $\mathbf{X} = [X_{is}]$, which is the genotype matrix across individuals and SNPs, and $\mathbf{Y} = \left[Y_{ijk}\right]$, which is the expression phenotype tensor across individuals, genes and tissues. Both inputs may have missing datapoints, and while $\mathbf{X}$ is in practice close to complete, $\mathbf{Y}$ is typically missing datapoints at the same scale as having them. Furthermore, $\mathbf{Y}$ typically has structured missingness, with vectors along the gene axis being either near-complete or completely missing for a particular individual-tissue pair (i, k), and such pairs being further structured by tissues with correlated missingness patterns across individuals. This is due to some tissues being syatematically inaccessible for some recruited individuals, e.g. brain from living individuals.

The hidden variables in the model to be learned include parameters of five types:

$$\Theta = [\omega_{kjs}, \alpha_{ts}, a_t, \beta_{kjt}, b_{kj}]$$

ω is the tensor of association signals between *cis*-SNPs and genes, which is tissue-specific. Proximity of a *cis*-SNP to its target transcript is modeled by each gene j having a restricted subset of SNPs, $S_j \subset \{1, ..., S\}$, for which ω_{kjs} is allowed to be nonzero, i.e. $\forall s \notin S_j : \omega_{kjs} = 0$. For convenience, SNPs are indexed by their order along the chromosome, so the set of allowed *cis*-SNPs is an interval $S_j = \{S_{low(j)}, ..., S_{hi(j)}\}$. α is the matrix of association signals between genome-wide SNPs and cellular hidden factors indexed by t, and these factors are individual specific without imposed tissue specificity, so may be relevant at any suset of the tisues. a is the vector of mean factor effects beyond SNPs effects. β is the matrix of association signasl between cellular factors and genes, independently across various tissue types. b is the vector of mean tissue effects beyond factor effects.

This gives rise to the following formulation for the gene expression:

$$\widetilde{Y}_{ijk} = \sum_{s=low(j)}^{hi(j)} \omega_{kjs} \cdot X_{is} + \sum_{t=1}^{T} \beta_{kjt} \cdot F_{it} + b_{kj} + \text{noise}$$

For the ML model $F_{it} = \sum_s \alpha_{ts} \cdot X_{is} + a_t$ implicitly encodes pathways activated by SNPs for a particular individual. In contrast, for the NN model $F_{it} = \text{sigmoid}(\sum_s \alpha_{ts} \cdot X_{is} + a_t)$, making the full structure a mixed model of linear regression and one-hidden-layer neural network with the logistic function as the activation function (Fig. 1).

We define the least-square error:

$$L'(\mathbf{X}, \mathbf{Y}, \Theta) = \frac{1}{2} \sum_i \sum_j \sum_k (\widetilde{Y}_{ijk} - Y_{ijk})^2$$

Thus the objective function (loss function) to be minimized by $L1$-regularizing the inferred hidden variables:

$$L(\mathbf{X}, \mathbf{Y}, \Theta) = L'(\mathbf{X}, \mathbf{Y}, \Theta) + \lambda_1 \sum_k \sum_j \sum_s |\omega_{kjs}| + \lambda_2 \sum_t \sum_s |\alpha_{ts}| + \lambda_3 \sum_k \sum_j \sum_t |\beta_{kjt}|$$

The $L1$ regularization terms with penalty strength parameters $\lambda_{\{1,2,3\}}$ in the loss function sparsifies the model parameters, eliminating small noise and false positive signals.

2.1.2 Tensor predictive modeling

The idea of tensor decomposition is not new to genetic studies (Hore *et al.* (2016)), but to our knowledge, our work the first to propose a uniform framework connecting genetics and gene expression profiles through factors in a joint decomposition. Also, our work goes an extra modeling step by explicitly incorporating the *cis*-regulation into the joint decomposition. This element of our models makes the eQTLs more interpretable on the one hand, while on the other allows the network modeling part to be more focused on genome-wide *trans*-effects.

Here we first introduce the genetic effects beyond *cis*-regulation in our tensor predictive modeling. Specifically, tensor predictive model assumes the expression tensor decomposable into factors, each of which reflects per-individual contributions $[F_{it}]$, per-gene contributions $[V_{jt}]$, and per-tissue contributions $[W_{kt}]$. To give some extra degree of freedom for modeling gene dimension, we allow bias for gene factor matrix through parameter d_j. Tensor predictive model further assumes the individual component $[F_{it}]$ is directly imputed by genetic information of these individuals, with coefficient α_{ts} and mean effect a_t.

Overall, the hidden variables to be learned include:

$$\Theta_{hidden} = [\alpha_{ts}, a_t, V_{jt}, d_j, W_{kt}]$$

Combining the *cis*-regulation (using the same indexing as above), we have the following formulation for the gene expression:

$$\widetilde{Y}_{ijk} = \sum_{s}^{S_j} \omega_{kjs} \cdot X_{is} + \sum_{t}^{T} F_{it} \cdot V_{jt} \cdot W_{kt} + d_j + \text{noise}$$

where $F_{it} = \sum_s \alpha_{ts} \cdot X_{is} + a_t$ is the imputation of individual factors from SNPs (Fig. 1).

We define the least-square error:

$$L'(\mathbf{X}, \mathbf{Y}, \Theta) = \frac{1}{2} \sum_i \sum_j \sum_k (\widetilde{Y}_{ijk} - Y_{ijk})^2$$

Thus the objective function (loss function) to be minimized by $L1$-regularizing the inferred hidden variables:

$$\begin{aligned} L(\mathbf{X}, \mathbf{Y}, \Theta) = L'(\mathbf{X}, \mathbf{Y}, \Theta) &+ \lambda_1 \sum_k \sum_j \sum_s |\omega_{kjs}| \\ &+ \lambda_2 \sum_j \sum_t |V_{jt}| + \lambda_3 \sum_k \sum_t |W_{kt}| \\ &+ \lambda_4 \sum_t \sum_s |\alpha_{ts}| \end{aligned}$$

The $L1$ regularization terms with penalty strength parameters $\lambda_{\{1,2,3,4\}}$ in the loss function have the same effects as before.

2.2 Inference and implementation

We have two stages in terms of development of these models. In the first stage, we derived the solvers of these models and implemented them on GPU to scale up. In the second stage, we utilized the increasingly popular machine learning library – *TensorFlow*[1], which has both great scalability and great flexibility, and is derivation-free for gradient descent based solvers. TensorFlow helped us a lot in running our models on real data, but here we'll still discuss our early-stage efforts, since they involve some very basic principles and useful techniques for solving models and scaling them up with GPU, which might be interesting to a very broad audience. We'll discuss in details how we solve the neural network model, as the other two are analogous to this and more straightforward.

The straightforward algorithm to solve this model is gradient descent (GD). Since the *trans*-part of our model is just a one-hidden-layer neural network, we use backpropogation to calculate the gradient of relevant parameters, which is α (with intercept a). For the least-squares part of the loss function (L'), we can compute partial derivatives with respect to the three types of model parameters as follows:

$$\frac{\delta L'}{\delta \omega_{kjs}} = \frac{1}{N} \sum_{i}^{N_k} (\widetilde{Y}_{ijk} - Y_{ijk}) \cdot x_{is}$$

$$\frac{\delta L'}{\delta \beta_{kjt}} = \frac{1}{N} \sum_{i}^{N_k} (\widetilde{Y}_{ijk} - Y_{ijk}) \cdot F_{it}$$

$$\frac{\delta L'}{\delta \alpha_{ts}} = \frac{1}{N} \sum_k \sum_i^{N_k} x_{is} \cdot F_{it} \cdot (1 - F_{it}) \sum_j (\widetilde{Y}_{ijk} - Y_{ijk}) \cdot \beta_{kjt}$$

where N_k is the number of data points from tissue k and N is the total number of data points in this incomplete expression tensor.

The $L1$ penalty term in our model is not formally differentiable when the relevant parameter is zero. We standardly abuse notation by defining the derivative to be zero at that point.

The scale of our model is very large, since we need to consider the whole-genome genetic effects of eQTLs. This increases the number of candidate *cis*-SNPs that need to be considered across all genes. It further increases the size of the linear system of genome-wide association to the given number of latent factors. Thus solving the GD requires special care to the implementation. We resolve the computational issue by General Purpose GPU computing (GPGPU). The difference between traditional CPU architecture and that of a GPU is that a GPU could utilize many cores to perform massively parallel computing. GPUs have recurrently benefited recent rapid developments in deep learning research (Oh and Jung (2004); Chellapilla *et al.* (2006); Raina *et al.* (2009); Cireşan *et al.* (2010)), as well as more general purpose scientific computing across various areas (Coccioni *et al.* (2011); Alerstam *et al.* (2008); Manavski and Valle (2008); Boyer and Baz (2013)). We thus took advantage of the computing power of GPUs to make our previously infeasible inference procedure practical. In practice, we use Compute Unified Device Architecture (CUDA) C/C++, which is a parallel computing platform and application programming interface (API) model for GPU programming. Our code and tested software is publicly available[2].

2.3 Initialization

2.3.1 Linear and nonlinear factor modeling

Initialization is very critical in our modeling, due to the strong biological context, and scarcity of data compared to the complexity of our model. We need to initialize wisely in order to avoid overfitting and make the learned results more biologically meaningful and relevant. Here we propose the following strategy to achieve that. We will discuss in details on how to initialize the neural network model, since the multiple linear model uses the same strategy but a relatively simpler version, which we will discuss later.

1. Initialize the association signals ω of *cis*-SNPs to all genes by solving linear systems through regularized regression, or more specifically group-LASSO (Yuan and Lin (2006)) in a multiple-tissue context, to encourage both sparsity and tissue similarity for *cis*-regulation.
2. Subtract the inferred *cis*-effects of our initialized *cis*-model part out from the expression profiles to get the residuals.
3. Spread the 3D expression tensor (here the residual tensor) into a 2D $Sample \times Gene$ matrix, ignoring the tissue label and individual label of each sample.

[1] https://www.tensorflow.org/

[2] https://github.com/morrisyoung/eQTL_MILAGE

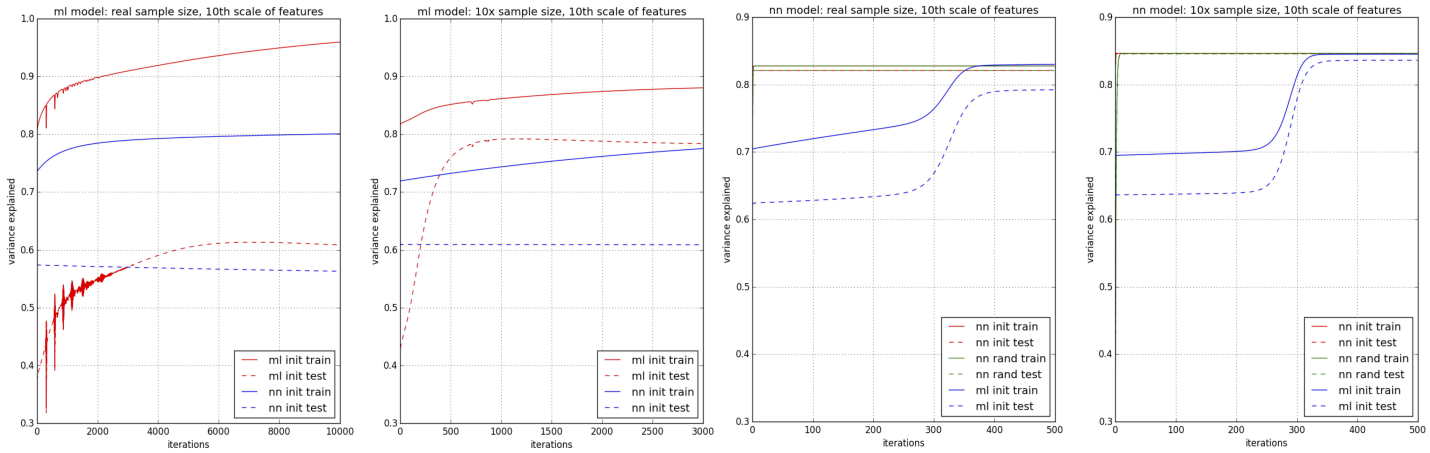

Fig. 2. We tested the performance of multiple linear model (ml) and neural network model (nn) on data with different nature with simulations. We simulated the data with ml and solved it with ml and nn (left two), and also simulated with nn (right two) and solved it with the two models. For each setting, we simulated the data with real sample size and 10X sample size, but fixing the feature dimensions (number of SNPs and number of genes) as 10th of the real data. All experiments were done only for the trans- factor part, without simulating and solving the cis- part.

4. Perform principal component analysis (PCA) on the $Sample \times Gene$ matrix.
5. Compute average sample factors from PCA for each individual across all tissues, to average out tissue effects and get tissue-unaware individual factors.
6. Scale individual factors into range $[0.1, 0.9]$. This step is needed because these values are the output of a sigmoid function, which needs to be in range $(0, 1)$, and scaling it to a narrower range around 0.5 avoids saturation of the logistic function.
7. Initialize association signals β between factors and genes by solving linear systems between the initialized, scaled factors and tissue-specific expression profiles with group-LASSO, to encourage both sparsity and tissue similarity for factor regulation.
8. Compute input factors for all individuals by passing the scaled individual factors through an inverse sigmoid function.
9. Initialize the association signals α between genome-wide SNPs with these input factors by solving a linear system with LASSO.

There are several critical components to elaborate here:

First of all, the biological content of this modeling problem needs significant sparsity for the model. We thus always use regularized regression model, LASSO (or group-LASSO), as a solver for linear systems to encourage sparsity. Secondly, the reason we initialize *cis* part first and use residuals to initialize *trans* part is that, *cis*-eQTLs have been well studied to contain more known signals for gene expression regulation. Thirdly, when initializing ω and β using group-LASSO with an incomplete tensor $\mathbf{Y}$ in Step 1 and Step 7 above, we first need to impute the expression profiles to make the tensor complete. This is required by the group-LASSO solver setting, which only supports complete datasets. In practice, we fill in missing values across the incomplete $\mathbf{Y}$ using the mean value of all available samples of $\mathbf{Y}$. Finally, for initializing the α matrix, we only use the *cis*-SNPs of active genes (non-zero β parameters) in each factor to initialize the parameter of SNPs within that factor, again with LASSO. Throughout our inference, we restrict analysis to variants previously implicated as significant eQTLs of all genes (Aguet *et al.* (2016) and Jo *et al.* (2016)).

The matrix α of SNP association with factors is supposed to be very sparse. This is encouraged twofold. First, the gene factors β are already sparsified through previous group-LASSO initialization, so there should be only limited number of genes within each factor thus limited number of *cis*-SNPs that might affect the entire factor. Secondly, the LASSO solver will further sparsify these candidate SNPs to leave only some of them to actually have an effect. We apply this inference layer by layer, first initializing β, then initializing α based on the underlying assumption of our network mode, which is that *trans*-eQTL signals can be summarized by a small number of factors. We require each SNP to have a local *cis*-effect on some gene in order to have distant *trans*-effects for other genes. This assumption might be too constrained for general deep models of phenotypes, but biologically it is reasonable to assume that even a *trans* signal across a factor is due to a *cis*-effect on one of its genes. Furthermore, considering a small number of candidate SNPs for α ensures sparsity.

For the ML model, we don't need to do the above Step 6 and Step 8, since the direct linear pathway from SNPs to genes through factors has no logistic activation function and the range of pre- and post-activation is not a numerical concern.

In all our initialization, PCA, LASSO and group-LASSO solvers are from *scikit-learn*[1] Python library.

2.3.2 Tensor predictive modeling

For initializing the TM model, we use the same idea to sparsify the model parameters. However, approximating the expression 3D tensor with low-rank matrices needs to be taken care of specially. Here we show the steps:

1. Do the *cis*-regression and substraction as in previous section, and get the spread expression residual tensor with shape $Sample \times Gene$.
2. Perform PCA on this expression residual matrix, and refold the sample factor loading matrix into a smaller incomplete 3D tensor with shape $Tissue \times Individual \times Factor$, as $Y_{\{k\},\{i\},\{t\}}$.
3. Apply incomplete PCA (Stacklies *et al.* (2007)) with a single principal component for each $Y_{\{k\},\{i\},t}$ and for all of them across all factors, to get tissue factor matrix $W_{\{k\},\{t\}}$ and individual factor matrix $F_{\{i\},\{t\}}$.
4. Use the corresponding individual factor and tissue factor for each sample in expression matrix to construct the linear system between them (Hadamard product, $F_{i,\{t\}} \circ W_{k,\{t\}}$) and the expression sample ($Y_{i,k,\{j\}}$), and solve the linear system by LASSO to construct sparse gene expression factor matrix $V_{\{j\},\{t\}}$.
5. Solve the linear system between SNPs and individual factors, with the same sparsity strategy utilized in previous section.

[1] http://scikit-learn.org/stable/

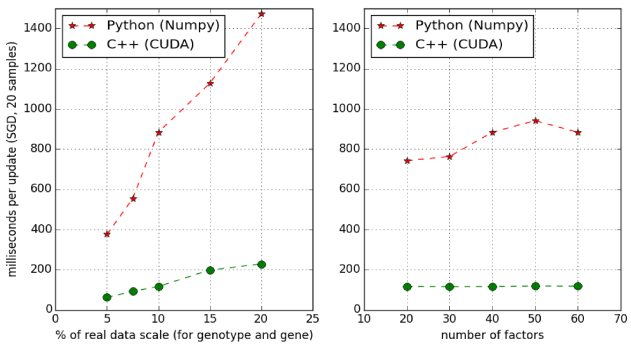

Fig. 3. We simulate data and benchmark the speed of GPU (CUDA C/C++) implementation and Python (Numpy) implementation. The left panel shows how the speed scales with the size of the dataset (S and J), when the number of factors (T) is fixed to 40. The right panel shows how the speed scales with the number of factors, with the size of dataset fixed as 10% of real data scale. Here we assume the real dataset has scale of $S = 1,000,000$, $J = 20,000$, which has the same order of magnitude as the real data from GTEx in our following real-data analysis.

The key difference here from the ML and NN model is that, we need to approximate the individual factor matrix and tissue factor matrix with multiple times of PCA approximations. The first time in Step 2 would allow us to analyze their variance in the new factor dimension, and the second time in Step 3 would further condense the variance of each factor from individual aspect and tissue aspect into their own dimensions. The whole process would allow us to keep the variance from the factors, and make the followup linear system (between the compound individual factor and tissue factor, with gene expression matrix) more aligned with the original 3D tensor decomposition (approximation). Also, we utilized incomplete PCA here since this setting involves incomplete tensor to be approximated by low-rank structures. We don't pre-impute the incomplete tensor here as we did for ML and NN, since the incomplete PCA method (Stacklies *et al.* (2007)) has already provided us a feasible and reliable solution to handle that.

3 Results

3.1 Simulation

We first conducted simulation analysis. We simulated the data treating the model as a generative one, simulating all parameters from standard normal distribution. While this simulation oversimplifies aspects of the distribution of real data, its purpose is to investigate the scalability of our models, and their capability of capturing signals on differently simulated data.

3.1.1 Benchmarking of CUDA implementation

Simulation allowed benchmarking on our computing platform, the Department of Systems Biology Information Technology (DSBIT) computing cluster at Columbia University, in which Nvidia Tesla M2090 GPUs and sufficient CPU memory (∼100GB per node) are available. In addition to our GPU implementation, we further benchmarked a highly optimized Python (*Numpy*[1]) implementation, in which both multi-threading and deeply-optimized CPU libraries for linear algebra are integrated. The comparison between the platforms in Figure 3 demonstrates the superiority of our GPU implementation when the dataset is very large.

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

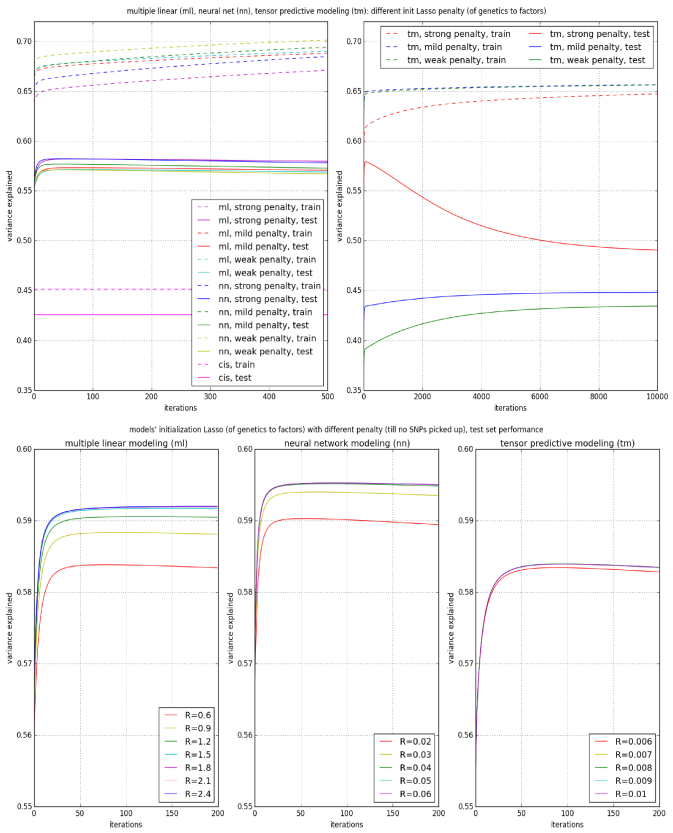

Fig. 4. Top: variance explained by cis- and the other three models; we use the overfitting signal from test set as the practical training stopping criteria; bottom: fine-tuning initialization LASSO (from genetics to factors) penalty parameters and its effects on the test set convergence of three models.

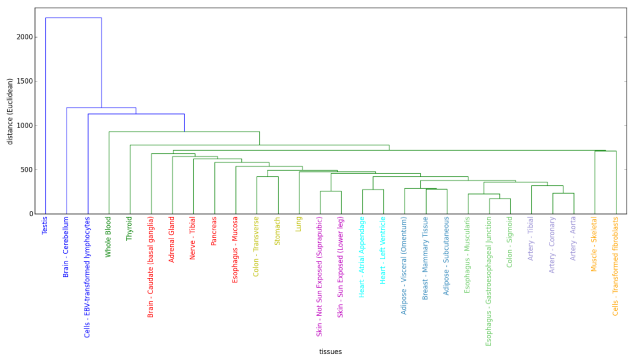

Fig. 5. Hierarchical clustering of tissue factors. Each tissue k is represented by a vector of association effects which is the β_k splayed.

3.1.2 Linear and nonlinear generative models

We simulated data from linear model (multiple linear) and nonlinear model (neural network) as generative models, and solved them in both linear and nonlinear ways. We intended to see how linear and nonlinear models could capture the signal in data with the same or different natures, when different sample availability were presented (Fig. 2). From the results we can see, when the data has a straight linear nature, the linear solver could always achieve a better likelihood value on both training set and testing set with a large margin to the nonlinear neural network opponents. The nonlinearly simulated data could be solved well by nonlinear solver under different

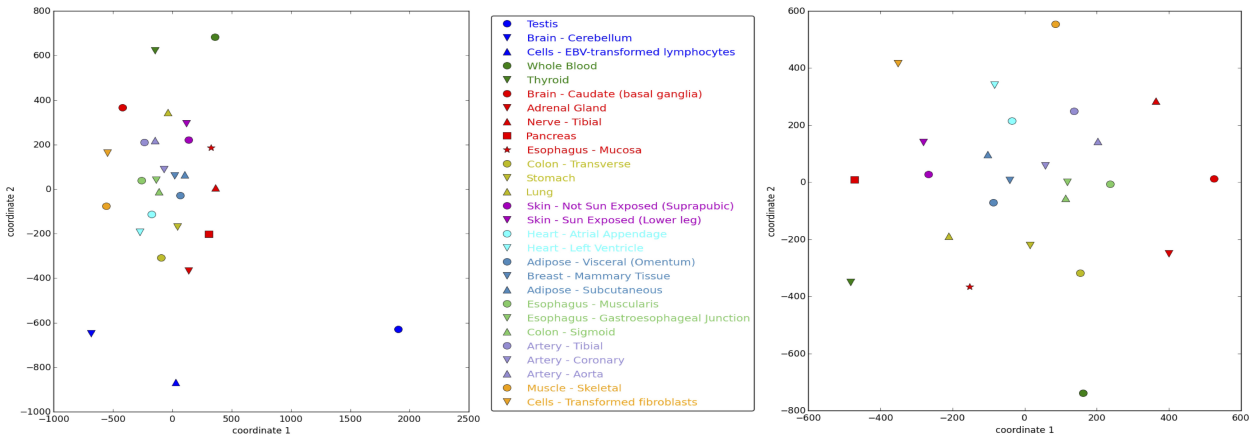

Fig. 6. Multi-Dimentional Scaling (MDS) of tissue regilation factors. Data is shown for all $K = 28$ tissues (left) as well as with the three outliers removed (right).

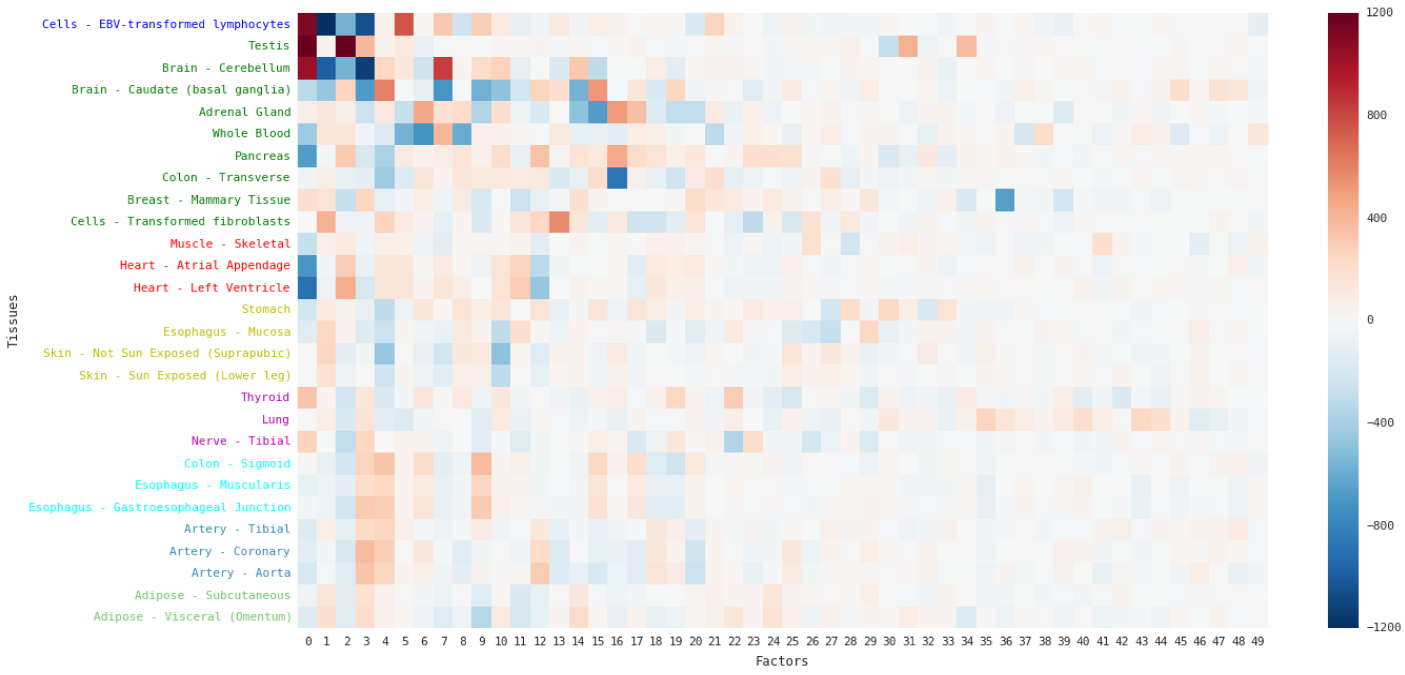

Fig. 7. Tissue activation matrix of factors in tensor predictive model.

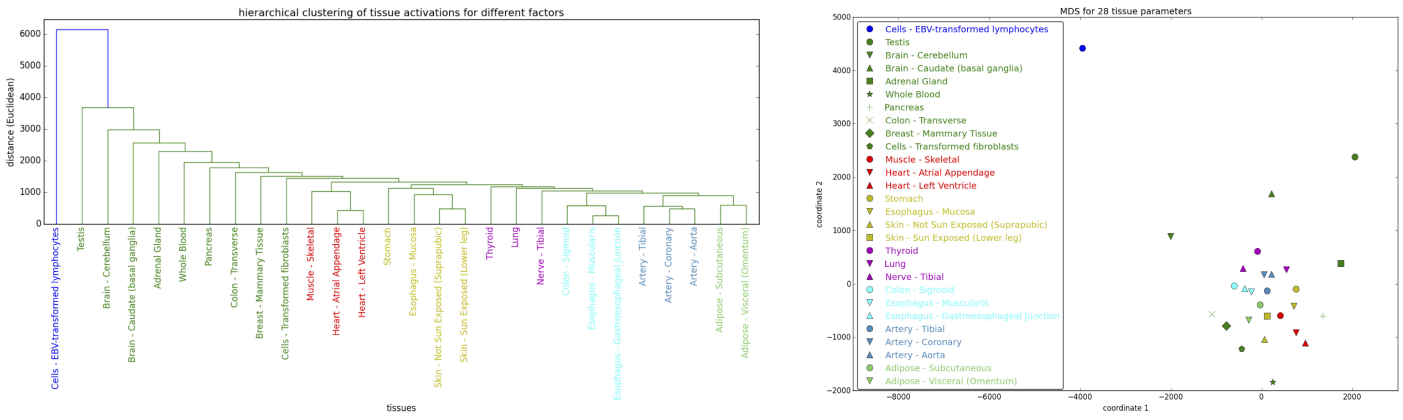

Fig. 8. Hierarchical clustering and MDS for tensor activation pattern of factors from tensor predictive model.

gene set (size)	overlap (ratio)	log *p*val	log FDR *q*val
Brain - Cerebellum, factor No.2			
synapse (754)	53 (0.0703)	-52.3	-48.1
synapse_part (610)	45 (0.0738)	-45.0	-41.1
neuron_part (1265)	56 (0.0443)	-44.6	-40.6
MODULE_11 (540)	40 (0.0741)	-39.9	-36.4
MODULE_100 (544)	39 (0.0717)	-38.3	-34.7
Testis, factor No.4			
sexual_reproduction (730)	31 (0.0425)	-26.3	-22.2
GNF2_CCNA1 (66)	16 (0.2424)	-26.2	-22.2
GNF2_MLF1 (87)	17 (0.1954)	-26.0	-22.2
multi_organism_reproductive_process (891)	31 (0.0348)	-23.7	-20.1
male_gamete_generation (486)	24 (0.0494)	-21.8	-18.7
Adrenal Gland, factor No.15			
small_molecule_metabolic _process (1767)	48 (0.0272)	-27.9	=23.7
sterol_metabolic_process (123)	19 (0.1545)	-25.0	-21.1
steroid_metabolic_process (237)	20 (0.0844)	-20.9	-17.2
metabolism_of_lipids_lipoproteins (478)	24 (0.0502)	-19.7	-16.1
WEST_ADRENOCORTICAL _TUMOR_DN (546)	25 (0.0458)	-19.5	=16.0

Table 1. We input the top 200 genes from each tissue's most activated factor, and compute overlap with MSigDB gene sets of all categories. The above table shows the top 5 enriched (or overlapped) gene sets returned by GSEA web interface. We can see these tissues in their representative factors all have very consistent functional gene sets overlapped with them, which indicates the effectiveness of these learned tissue-factors in terms of biological functions.

data availability, though the linear model could also approximate the data well especially for the training set given sufficient rounds of training (that is because we only have one hidden layer in our generative model, which is not deep in terms of nonlinearity). This gives us some insights regarding whether a linear or nonlinear model could better explain the data in our followup real-data analysis.

3.2 Real data analysis

3.2.1 Dataset and preprocessing

MILAGE is built to handle datasets like GTEx (Lonsdale *et al.* (2013)) for modeling diversity of expression across individuals and tissues. This imposes critical data size requirements. Specifically, GTEx data involves dozens of tissues across hundreds of individuals. Yet, much of the GTEx tensor is incomplete, making the actual size data more manageable that a full tensor would have been. We consider *eQTL tissues* where the per-tissue number of samples is $\geq$100 from GTEx. This leaves $K = 28$ tissues across which $I =_{total}= 449$ individuals have genotype data and expression data. We curtail another dimension of the computation bykeeping only the $J = 19,425$ autosomal transcripts expressed at $\mathrm{R}PKM \geq 0.1$ across $\geq 50\%$ of the samples. Finally, we address the longest dimension, which is genetic information along the genome. We use GTEx (version phs000424.v6.p1) genotypes for common SNPs (MAF≥ 0.01). We consider for the genotype matrix $\mathbf{X}$ the imputed genotype data for each SNP and individual. This data is available in dosage format, as real number in $[0, 2]$ that reports the expected number of non-reference alleles, rather than discrete genotype $\{0, 1, 2\}$, which is unknown. This representation conveys the uncertainty of imputation setting elements in $\mathbf{X}$ to $x = 0 \cdot P_0 + 1 \cdot P_1 + 2 \cdot P_2$, where $P_{0,1,2}$ are respectively the posterior probabilities for genotype $\{0, 1, 2\}$ ascribed by imputation. We further restrict analysis to all the significantly associated *cis*- SNPs and *trans*- SNPs across all genes from recent GTEx analytical efforts (see Aguet *et al.* (2016) and Jo *et al.* (2016)) as our pre-analyzed candidate SNPs. This still leaves us $S = 2,445,192$ sites for analysis. This total number of SNPs is distributed across genes so they have average count of 2817 (range $[0, 31873]$) candidate *cis*-SNPs within 1 million base pairs up- and down- stream of transcription start site. We use candidates for the respective gene in the *cis* part of our model.

For the gene expression data, we randomly draw 75% of individuals ($I_{train} = 336$) with their 4270 cross-tissue samples as the training set, and use the left ($I_{test} = 113$ individuals, 1424 samples) as the testing set, and we then stick with these sets during the whole evaluation process. We normalize the gene expression RPKM values across the training set using Z-statistics among all tissue samples. We further map expression profiles of the testing set into Z values using the same moments from the training set.This means there is now leakage of information from test samples into the model, while keeping the normalized expression values close to their Z-score interpretation.

We manually set the number of factors to be $T = 400$. This is consistent with the number of factors used by Gao *et al.* (2016) in a less expressive model. We thus expect fewer factors to be required, and thus hope for the learning algorithm to identify many of the factors as redundant. We rely on the regularized initialization of the factor tensor β to zero out the effects of such tensors, thus removing them from consideration also post-initialization.

3.2.2 Variance explained through different models

We evaluate the ability of MILAGE to explain variance in real data in four ways – *cis* pathway alone, and the other three variants of MILAGE (Fig. 4). We first consider only the *cis* part of our model, as *cis*-eQTLs were well studied for over a decade and have observable signals regulating gene expressions. Our results show that upon initialization the linear regression model for *cis*-SNPs could only explain 45.1% of the training set variance and 42.6% of that of the testing set. Moreover, neither of these benchmarks could be more than negligibly improved by further training with GD. This is consistent with known qualities of the regularized regression model (LASSO solver). Secondly, we consider the combined *cis*- and *trans*-model, with the three variants of MILAGE – ML, NN and TM. Since the *cis* part does not improve by learning, we keep it as it is, and use it to extract residuals of the expression tensor $\mathbf{Y}$ that still need to be explained. We train the *trans* part of these models on these residuals, and evaluate the total variance explained by both parts. The combined model performs significantly better than *cis* part of our model, despite much overlap in the variance eQTL-SNPs explain through both parts of the model. We observe that for all three models, a stronger penalty to the initialization of the model (for the wide linear connection of genetics and factors) brings a worse training set final result, while a better testing set result. This both shows the consistency with our understanding on regularization (better generalization on test set), and the insight that for a better overall likelihood less priority should be given to specific set of candidate SNPs in terms of their joint genetic effects to a factor. We further observe that the ML model performs very similar to the NN model, in both the trend of convergence and the final variance explained, while the TM model seems very sensitive to the initialization, and a worse initialization could even make the starting variance explained on test set worse than the *cis* part of the model. Combined with the simulation analysis conducted above (Section 3.1.2, Fig. 2), we conclude here straight linear effects are not the best interpretation of the genome-wide genetic effects for regulating gene expression, since we don't have significant better performance of linear model compared to the nonlinear model as we observed in simulation (Fig. 2). We also conclude that there are some nonlinearity since the model training of NN indeed further improves the model, however this nonlinearity is not as simple as what we simulated (Fig. 2) but rather

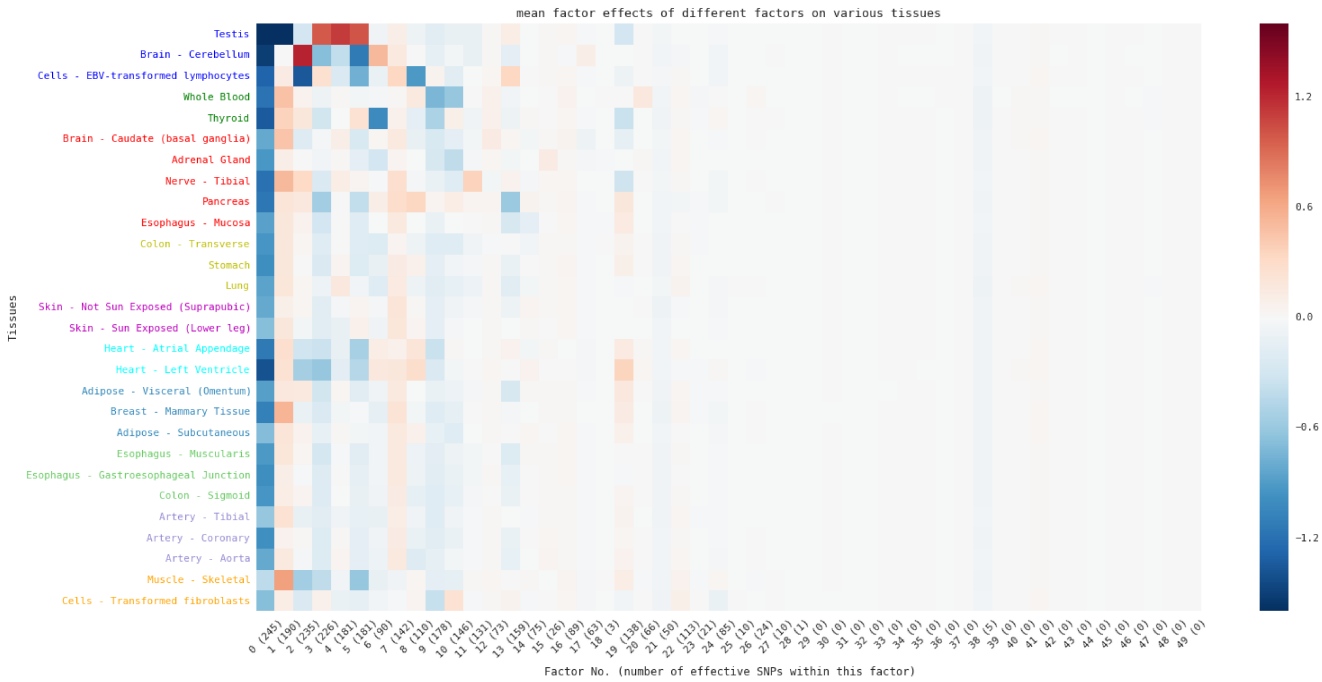

Fig. 9. The mean activation value for tissues of first 50 factors . The SNP associations are very sparse, and only limited number of SNPs has non-zero effects. In practice, we use ϵ as the threshold for non-zero SNPs, since there are some noisy effects among the SNPs. We saw when $\epsilon = 0.000001$ it would be able to eliminate all these small noise, while smaller values will quickly increase the total number of effective β several magnitude larger.

complicated among all the SNPs interacted with each other. For the TM model, it turns out with a good initialization (here it's a strong penalty for the wide linear system of genetics to factors to pick up or prioritize no SNPs), the model could achieve test set variance explained as good as its direct opponent ML model.

In Figure 4, we also show more details regarding the fine-tuned initializations of different models. Since we realized that solving the wide linear model from genetics to factors with LASSO was sensitive to the final convergence and overall likelihood of our learned model, we tried different LASSO penalty parameters to see how the training process vary accordingly. Since the candidates for this wide linear model are some heuristic SNPs to each factor (*cis*-SNPs of genes in that factor), this essentially tested how such heuristics would affect our finally learned model. It turns out that when no heuristics are inserted (strongest LASSO penalty to pick up no candidate SNPs), all the three models could explain the data better (better generalization on testing set). This clearly shows the ubiquity and complexity of the whole-genome *trans*-effects to co-expressed genes that any SNP could have some weak signal towards factors, and our inappropriate assumption that *trans*-eQTL must have effects through gene-gene interactions and a *trans*-eQTL must be a *cis*-eQTL of some genes. Through a series experiments here, our best result is from NN model, with no candidate SNPs prioritized in the initialization, which achieves a 59.5% variance explained on test set.

3.2.3 Tissue specific regulations reveal functional similarity

We applied hierarchical clustering for the tissue-specific regulation of gene expression profiles from factors in neural network model. We explored multiple hierarchical clustering methods, observing highly similar results (not shown). We use Weighted Pair Group Method with Arithmetic Mean (WPGMA) in our analysis (Fig. 5). Related tissues are similar in terms of their SNP-affected expression factors: the triplet artery tissues, the triplet of adipose/mammary tissues, as well as the pairs of heart and skin tissues are each clustered together. This indicates the closer relationship of functional regulations of these tissues. It is important to distinguish this results from just expression patterns being similar within these clusters, as these factors are the target of SNP regulation rather than just added contributions to tissue-expression of all individuals. Finally, from the clustering, we can identify three very significant outlier tissues – testis, cerebellum and EBV-transformed lymphocytes. Testis in particular is an extreme outlier.

Based on above observation, high dimensional hierarchical relationship of tissues is consistent with prior biological knowledge. We further applied Multidimensional Scaling (MDS) to examine whether this type of distance relationship could be preserved in a low-dimension space with nonlinear mapping. Figure 6 highlights the outlier tissues as dominating MDS of all tissues. When these are removed, similarities between related tissues are better visible.

In the TM model, tissue activation pattern of factors also indicates tissue similarity well. We demonstrate how the tissue activation pattern of factors from TM look like in Figure 7. We did the same hierarchical clustering and MDS on the parameter matrix, showed in Figure 8. Note that, the activation pattern in the tensor predictive model is not the same as the regulation parameters from multiple linear model or neural network model. It provides another layer of hierarchy representing how each tissue loads the common co-expression factors, rather than what the specific co-expression factors are in each tissue. This difference comes from the natural difference between tensor modeling and multiple linear modeling or its nonlinear version.

3.2.4 Factor and enrichment analysis

We further studied factors and their functional enrichment in neural network model.

We considered the mean β across all genes for each tissues and factor, $\widehat{\beta_{kt}} = \frac{\sum_j \beta_{kjt}}{J}$ (Fig. 9). Interestingly, we find factor#3, factor#4 and

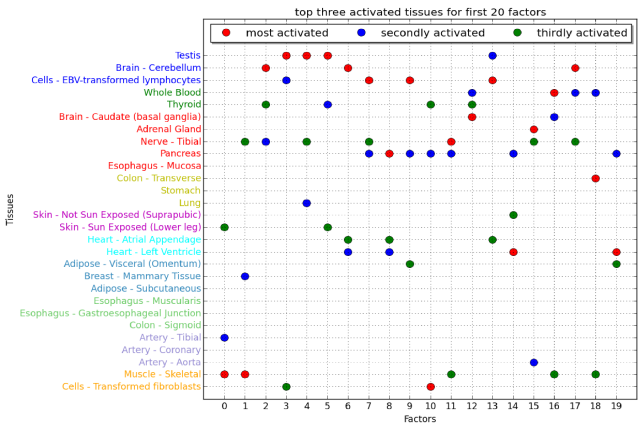

Fig. 10. The top three activated tissues for first 20 factors.

factor#5 are all strongly activated for testis, and factor#1 is strongly deactivated for the same tissue. This explains well the significant ourlier effect of this tissue seen in Figure 5. The more significant factors have uneven $\widehat{\beta}$ across tissues, each highlighting their top tissues (Fig. 10).

To investigate factors play a role in balancing the functional signals for specific tissues, we considered gene set enrichment analysis based on the β_{kjt} weights these factors ascribe to particular genes for $k = \text{testis}$ and $t = 1, 3, 4, 5$. We observe the top genes for these testis-associated factors (from 200 to 1000, decreasing-order for factor #3, #4 and #5, and increasing-order for factor #1) all have significant enrichment for some sexual-function gene sets (e.g., GO_SEXUAL_REPRODUCTION, GO_MALE_GAMETE_GENERATION). Furthermore, direct comparison of the most negatively weighted genes for $t = 1$ and the most positively weighted genes for $t = 4$ showed 105/200 or 868/1000 of them to intersect.

Next, we considered the learned α matrix which associates all SNPs with the specified factors. We find that each factor is only associated with limited number of significant causal SNPs in our model, as listed in Figure 9. This indicates the very sparse nature of the SNP factor associations, or more generally the limited amount of signal from *trans*-effects.

Finally, we conduct enrichment analysis of top genes in these tissues with respect to gene sets from the Molecular Signatures Database v5.2 (Subramanian *et al.* (2005)). We observe overlapping enrichment sets between the tissues, associating such tissues with some relevant molecular functions. We further identify the dominating functions of different factors. See Table 1 for the results.

We further build **M**ulti-**T**issue **G**ene **S**et **E**nrichment **A**nalysis tool[1] (MTGSEA), a complementary tool to MILAGE. MTGSEA reports the enrichment score (ES) of input gene sets for different factors in various tissues, with the method used in GSEA (Subramanian *et al.* (2005)). Different from the original ES calculation method, here we input a list of ordered genes (by the β values of these genes under the specified factor) with their β values, rather than actual gene expression profiles with phenotypes as required by GSEA software. However this is consistent with GSEA, as the β matrix is essentially used to characterize how each gene associated with this factor, providing a similar metric as the correlation number of gene with phenotype used in GSEA. In addition, we calculate a p-value to quantify the null chances to observe the calculated ES among the best ES calculated from all gene sets from different random shufflings. This randomization procedure aims to eliminate the correlation effects of different gene sets. Last but not least, in order to report only significantly enriched gene sets for different factors and tissues, we set the False Discover Rate (FDR) to 0.1 in our analysis. See Table 2 for the results. Similarly to the above overlap enrichment analysis but in a different and more systematic quantification fashion, we can identify some functionally relevant gene sets for different tissues (here we show Brain - Cerebellum factor #2 and #5, positive and negative enrichment respectively).

[1] https://github.com/morrisyoung/eQTL_MILAGE

gene set	factor 2(+)		factor 5(-)	
	ES	p	ES	p
GNF2_TM4SF2	.968	**	.972	**
calcium_ion_regulated_exocytosis_of _neurotransmitter	.922	**	.889	.01
GNF2_AF1Q	.886	.001	.912	**
syntaxin_1_binding	.877	.004	.869	.039
GNF2_RAB3A	.872	.005	.902	.004
GNF2_MAPT	.872	.005	.886	.013
GNF2_DNM1	.868	.006	.884	.013
GNF2_RTN1	.862	.015	.881	.016
glutamate_secretion	.859	.019	.849	.163
MODULE_274	.856	.025	.846	.207
botulinum_neurotoxicity	.851	.037	NS	
neurotransmitter_release_cycle	.849	.044	.853	.135
presynaptic_active_zone	.847	.05	.867	.042
transmission_of_nerve_impulse	.844	.064	.841	.304
RORIE_TARGETS_OF_EWSR1_FLI1 _FUSION_DN	.841	.081	NS	
positive_regulation_of_calcium_ion _dependent_exocytosis	.837	.098	.877	.025
exocytic_vesicle_membrane	.834	.116	.863	.061
cerebellar_cortex_formation	.828	.156	NS	
calcium_ion_regulated_exocytosis	.825	.181	.845	.239
excitatory_postsynaptic_potential	.825	.188	.863	.061
presynaptic_process_involved_in_synaptic _transmission	.821	.231	NS	
synaptic_transmission_glutamatergic	.819	.255	NS	
neurotransmitter_transport	.818	.263	NS	
sodium_channel_activity	.814	.331	NS	
regulation_of_neurotransmitter_receptor _activity	NS		.869	.039
insulin_synthesis_and_processing	NS		.865	.049
walking_behavior	NS		.863	.06
adult_walking_behavior	NS		.863	.06
synaptic_vesicle_recycling	NS		.863	.065
synaptic_vesicle_cycle	NS		.849	.169
postsynaptic_membrane_organization	NS		.841	.297
BIOCARTA_NOS1_PATHWAY	NS		.837	.354

Table 2. ES enrichment analysis in factor activated in brain, and a factor deactivated in brain showing similar enrichment patterns.$**$:$p < .001$; NS: FDR>0.1; *small italics*: GO annotation; small letters: Reactome annotation

4 Discussion

Modeling eQTLs poses significant analytical challenges due to the dimensions of the problem and the interconnectedness of its components. Yet, the combination of genetic variation data and molecular level measurements which they affect is a powerful one. In this work we showed how to jointly model *cis*- and *trans*-effects of eQTLs in various ways, and how to scale

up the model on whole-genome data. We show that our analysis reveals not only lists of eQTLs, but rather enrichment of networks to functional modules.

MILAGEFF opens the door and provides insights to various research questions, in terms of both the inference process itself, as well as the biology to be inferred. One key modelling decision to be made is whether linear models are sufficient to describe such data, or whether nonlinear terms are essential. We've showed in our analysis that there are indeed irreplaceable nonlinear effects for the factor-fashion *trans*-eQTLs, and a nonlinear model performs better than its linear opponent. Another such issue concerns the assumed independence of tissue parameters on one another, as opposed to potentially assuming the existence of tissue-independent factors (Hore *et al.* (2016)). We also showed in our real-data analysis that, a tensor-decomposition fashion model assuming the existance of tissue-independent co-expression factors has less performance on training set, while surprisingly it holds test set performance almost as good as a factor model with tissue-dependent parameters. This indicates the expressiveness of a simpler tensor-decomposition model, and on the other hand supplements the analysis of such tensor modeling for co-expression networks (Hore *et al.* (2016)) in terms of their tissue-dependency or tissue-independency nature. In terms of the biology, there is strong rationale for the incorporation of functional data, such as epigenetics, that might provide informative priors regarding SNP involvement in regulation. We leave such prior incorporation into our modeling to future investigations. Separately, we would want to use the learned model to impute gene expression profiles from an independent study, in order to assist further downstream biomedical analysis and evaluate performance with respect to other methods like PrediXcan (Gamazon *et al.* (2015)).

Acknowledgements

We thank the GTEx consortium for high quality, highly accessible data.

Funding

S.Y. and D.P. have been supported by NIH Pioneer award DP1HD084071 and NCI R01CA164729. I.P has been supported by NSF CCF1527498 and U54 CA209997.

References

Aguet, F. et al (2016). Local genetic effects on gene expression across 44 human tissues. *bioRxiv*.

Alerstam, E., Svensson, T. and Andersson-Engels, S. (2008). Parallel computing with graphics processing units for high-speed monte carlo simulation of photon migration. *Journal of Biomedical Optics*, **13**(6), 060504–060504–3.

Ardlie, K.G. et al (2015). The genotype-tissue expression (gtex) pilot analysis: Multitissue gene regulation in humans. *Science*, **348**(6235), 648–660.

Boyer, V. and Baz, D.E. (2013). Recent advances on gpu computing in operations research. In *2013 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum*, pages 1778–1787.

Brem, R.B. et al (2002). Genetic dissection of transcriptional regulation in budding yeast. *Science*, **296**(5568), 752–755.

Chellapilla, K., Puri, S. and Simard, P. (2006). High Performance Convolutional Neural Networks for Document Processing. In G. Lorette, editor, *Tenth International Workshop on Frontiers in Handwriting Recognition*, La Baule (France). Université de Rennes 1, Suvisoft. http://www.suvisoft.com.

Cireşan, D.C. et al (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, **22**(12), 3207–3220.

Cococcioni, M., Grasso, R. and Rixen, M. (2011). Rapid prototyping of high performance fuzzy computing applications using high level gpu programming for maritime operations support. In *2011 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, pages 17–23.

Cookson, W. et al (2009). Mapping complex disease traits with global gene expression. *Nat Rev Genet*, **10**(3), 184–194.

Gamazon, E.R. et al (2015). A gene-based association method for mapping traits using reference transcriptome data. *Nat Genet*, **47**(9), 1091–1098.

Gao, C., Brown, C.D. and Engelhardt, B.E. (2013). A latent factor model with a mixture of sparse and dense factors to model gene expression data with confounding effects. *ArXiv e-prints*.

Gao, C. et al (2016). Context specific and differential gene co-expression networks via bayesian biclustering. *PLOS Computational Biology*, **12**(7), 1–39.

Hore, V. et al (2016). Tensor decomposition for multiple-tissue gene expression experiments. *Nat Genet*, **48**(9), 1094–1100.

Jo, B. et al (2016). Distant regulatory effects of genetic variation in multiple human tissues. *bioRxiv*.

Lonsdale, J. et al (2013). The genotype-tissue expression (gtex) project. *Nat Genet*, **45**(6), 580–585.

Manavski, S.A. and Valle, G. (2008). Cuda compatible gpu cards as efficient hardware accelerators for smith-waterman sequence alignment. *BMC Bioinformatics*, **9**(2), S10.

Michaelson, J.J., Loguercio, S. and Beyer, A. (2009). Detection and interpretation of expression quantitative trait loci (eqtl). *Methods*, **48**(3), 265 – 276. Global approaches to study gene regulation.

Oh, K.S. and Jung, K. (2004). GPU implementation of neural networks. *Pattern Recognition*, **37**(6), 1311 – 1314.

Raina, R., Madhavan, A. and Ng, A.Y. (2009). Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 873–880, New York, NY, USA. ACM.

Rockman, M.V. and Kruglyak, L. (2006). Genetics of global gene expression. *Nat Rev Genet*, **7**(11), 862–872.

Stacklies, W. et al (2007). pcamethods–a bioconductor package providing pca methods for incomplete data. *Bioinformatics*, **23**(9), 1164.

Subramanian, A. et al (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, **102**(43), 15545–15550.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **68**(1), 49–67.

Zhao, S. et al (2016). Bayesian group factor analysis with structured sparsity. *Journal of Machine Learning Research*, **17**(196), 1–47.