Deep Large-Scale Multi-Task Learning Network for Gene Expression Inference

Kamran Ghasedi Dizaji[1], Wei Chen[3], Heng Huang[1,2] ⋆

[1]Department of Electrical and Computer Engineering, University of Pittsburgh
[2]Department of Biomedical Informatics, School of Medicine, University of Pittsburgh
[3]Department of Pediatrics, UPMC Children's Hospital of Pittsburgh, University of Pittsburgh

Abstract. Gene expressions profiling empowers many biological studies in various fields by comprehensive characterization of cellular status under different experimental conditions. Despite the recent advances in high-throughput technologies, profiling the whole-genome set is still challenging and expensive. Based on the fact that there is high correlation among the expression patterns of different genes, the above issue can be addressed by a cost-effective approach that collects only a small subset of genes, called landmark genes, as the representative of the entire genome set and estimates the remaining ones, called target genes, via the computational model. Several shallow and deep regression models have been presented in the literature for inferring the expressions of target genes. However, the shallow models suffer from underfitting due to their insufficient capacity in capturing the complex nature of gene expression data, and the existing deep models are prone to overfitting due to the lack of using the interrelations of target genes in the learning framework. To address these challenges, we formulate the gene expression inference as a multi-task learning problem and propose a novel deep multi-task learning algorithm with automatically learning the biological interrelations among target genes and utilizing such information to enhance the prediction. In particular, we employ a multi-layer sub-network with low dimensional latent variables for learning the interrelations among target genes (*i.e.* distinct predictive tasks), and impose a seamless and easy to implement regularization on deep models. Unlike the conventional complicated multi-task learning methods, which can only deal with tens or hundreds of tasks, our proposed algorithm can effectively learn the interrelations from the large-scale ($\sim$10,000) tasks on the gene expression inference problem, and does not suffer from cost-prohibitive operations. Experimental results indicate the superiority of our method compared to the existing gene expression inference models and alternative multi-task learning algorithms on two large-scale datasets.

1 Introduction

Characterizing the cellular status under various states such as disease conditions, genetic perturbations and drug treatments is a fundamental problem in biological studies. Gene expression profiling provides a powerful tool for comprehensive analysis of the

⋆ Corresponding Author. This work was partially supported by NSF IIS 1836938, DBI 1836866, IIS 1845666, IIS 1852606, IIS 1838627, IIS 1837956, and NIH AG049371.

cellular status by capturing the gene expression patterns. The recent advances in high-throughput technologies make it possible to collect extensive gene expression profiles in versatile cellular conditions, providing invaluable large-scale databases of gene expressions for various biomedical studies [10, 6]. For instance, Van *et. al.* recognized the effective genes on breast cancer by studying gene expression patterns of different patients [44]. Stephens *et. al.* analyzed the relations between and within different cancer types by investigating the correlations of gene expression data among distinct types of tumors [42]. Richiardi *et. al.* examined the gene expression data in a post meortem brain tissue, and showed correlation between resting-state functional brain networks and activity of genes [35]. Radical change in expression levels of several immune-related genes is identified in mice susceptible to influenza A virus infection using a microarray analysis [46]. The gene expression patterns in response to drug effects are also investigated on different tasks such as drug-target network construction [49] and drug discovery [34]. Moreover, the connection of single-gene mutations on some chromosomes and early onset of Alzheimer disease are examined in [3].

Despite recent developments on gene expression profiling, constructing large-scale gene expression archives under different experimental conditions is still challenging and expensive [31]. But previous studies have shown high correlations between gene expressions, indicating that the genes have similar functions in response to various conditions [32, 15, 38]. The clustering analysis of single cell RNA-Seq also shows similar expression pattern between intra-cluster genes across different cellular states [32]. Based on this fact, a small group of informative genes can be considered as the representative set of whole-genome data. The researchers in the Library of Integrated Network-based Cell-Signature (LINCS) program[1] used this assumption and employed principle component analysis (PCA) to choose $\sim$1000 genes, which contain $\sim$80% of the information in the entire set of genes. Note that profiling these $\sim$1000 genes, called landmark genes, instead of the whole-genomes drastically reduces the collection costs ($\sim$\$5 per profile) [33]. Hence, a cost-effective strategy in profiling of large-scale gene expressions data is to collect the landmark genes and predict the remaining genes (*i.e.* target genes) using a computational model.

The linear regression models with different regularizations are the first candidate models for predicting target genes. Later there were some attempts to use non-linear model to better capture the complex patterns of the gene expression profiles [14]. Deep models generally have shown remarkable flexibility in capturing the non-linear nature of biomedical data and high scalability in dealing with the large-scale datasets. Following the successful application of deep models on multiple biological problems [27, 1, 39, 41, 50], a few deep regression models have been also introduced for the gene expression inference problem [8, 12]. However, these deep regression models do not utilize the interrelations among the target genes. These models usually consist of multiple shared layers among the genes, followed by a specific layer for each gene at the top. Therefore, these models ignore the biological information related to the gene interactions in their training process which leads to their sub-optimal predictions.

To address the above challenges and utilize the interrelations between target genes to enhance the prediction task, we formulate the expression inference of target genes

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

from the landmark ones as a multi-task learning problem. Multi-task learning algorithms generally aim to improve the generalization of multiple task predictors using the knowledge transferred across the related tasks through a joint learning framework [7]. We consider each gene expression prediction as a learning task and employ the multi-task learning model to automatically learn the interrelations of all tasks (*i.e.* all target genes) and utilize such information to enhance the prediction. Although there are multiple studies in literature on designing multi-task learning algorithms for deep models [36], they are designed and applied to tens or hundreds of tasks, and are not effective and scalable to deal with large number of tasks like the gene expression inference problem with about 10,000 tasks.

In this paper, we propose a novel multi-task learning algorithm for training a deep regression network with automatically learning the task interrelations of the gene expression data. Our deep large-scale multi-task learning method, denoted as Deep-LSMTL, can effectively learn the task interrelations from a large number of tasks, and is also efficient without suffering from the cost-prohibitive computational operations. In particular, our Deep-LSMTL model learns tasks interrelations using subspace clustering of task-specific parameters. Considering this clustering as the reconstruction of each task parameters by linear and sparse combination of other task-specific parameters, Deep-LSMTL provides a seamless regularization on deep models by approximating the reconstruction loss in the stochastic learning paradigms (*e.g.* stochastic gradient descent). Deep-LSMTL employs a two-layer sub-network with low-dimension bottleneck to learn non-linear low-rank representations of task interrelations. Meanwhile, as a multi-task learning model, Deep-LSMTL can transfer asymmetric knowledge across the tasks to avoid the negative transfer issue, and enforce the task interrelations through the latent variables instead of the model parameters. All these advantages help Deep-LSMTL predict the target genes better than conventional approaches. We evaluate Deep-LSMTL with several deep and shallow regression models on two large-scale gene expression datasets. Experimental results indicate that our proposed algorithm has significantly better results compared to the state-of-the-art MTL methods and deep gene expression inference networks disregarding the neural network size and architecture. Furthermore, we gain insights into genes relations by visualizing the relevance of landmark and target genes in our inference model. The main contributions of this paper can be summarized as follows:

- Proposing a novel multi-task learning algorithm for training deep regression models, which is scalable to the large-scale tasks and efficient for the non-image data in the gene expression inference problem.
- Introducing a seamless regularization for deep multi-task models by employing a multi-layer sub-network with low-rank latent variables for learning the task interrelations.
- Outperforming existing gene expression inference models and alternative MTL algorithms by significant margins on two datasets regardless of network architectures.

The following sections are organized as follows. In Section 2, we briefly review the related works on gene expression inference and recent multi-task learning algorithms. In Section 3, we start with the general clustering-based multi-task learning method, and then propose our multi-task learning algorithm for deep regression models. Then,

we show the experimental results in Section 4, and evaluate the effectiveness of our algorithm in comparison with alternative models on multiple experimental conditions. We also plot some visualization figures to confirm the validity of our model. Finally, we conclude the paper in Section 5.

2 Related Work

2.1 Gene Expression Inference

Since archiving whole-genome expression profiles under various perturbations and biological conditions is still difficult and expensive [31], finding a way to reduce the costs while preserving the information is an important problem. The previous studies have shown that gene expressions are highly correlated, and even a small set of genes can contain rich information. For instance, Shah *et. al.* indicated that a random set of 20 genes contains $\sim$50% of the information of the whole-genome [38]. Moreover, the recent studies in RNA-seq confirm the assumption that a small set of genes is sufficient to indicate the comprehensive information throughout the transcriptome [32, 15].

In order to determine the set of most informative genes, researchers of the LINCS program collected *GEO* dataset[2] based on Affymetrix HGU133A microarrays, and analyzed the correlation of gene expression profiles. Given the total number of 12,063 genes, they calculated the maximum percentage of information that can be recovered by a subset of genes based on the comparable rank in the Kolmogorov-Smirnov statistic. According to the results of LINCS analysis, a subset of only 978 genes is able to recover 82% of the observed connections in the entire transcriptome [21]. These genes are landmark genes and can be utilized to infer the expression of remaining genes referred to target genes.

Considering the gene expression inference as a multi-task regression problem, the shallow models such as linear regression with ℓ_1-norm and ℓ_2-norm regularizations and K-nearest neighbors (KNN) are used to infer the target genes expression from the landmark ones [8, 12]. There are also a few attempts to use deep models on detecting and inferring gene expressions [8, 12, 23, 45]. Using the representation power of deep learning models, Chen *et al.* introduced a fully connected multi-layer perceptron network as a multi-task regression model for the gene expression inference [8]. They justified the effectiveness of their deep model by achieving better experimental results compared to shallow and linear regression models. Recently, Dizaji *et al.* introduced a semi-supervised model, called SemiGAN, based on generative adversarial networks (GAN) for the gene expression inference problem [12]. Assuming a set of landmark genes as the unlabeled data and a set of landmark and their corresponding target genes as the labeled data, SemiGAN learns the joint and marginal distributions of landmark and target genes, and then enhanced the training of a regression model using the estimated target genes for the unlabeled data as pseudo-labels. Although these deep inference models addressed the issue of insufficient capacity in shallow and linear regression models, they did not explore the task interrelations, which indicate the biological knowledge of genes, in their training process. Thus, we formulate the gene

[2] `https://cbcl.ics.uci.edu/public_data/D-GEX/`

expression inference problem as a multi-task learning and propose a new MTL method to explicitly learn the interrelations among the target genes in the learning framework and utilize these information to enhance the prediction results and also improve the generalization of our multi-task inference network.

2.2 Multi-Task Learning Algorithms

The main goal of multi-task learning is to enhance the generalization of multiple task predictors using the knowledge transferred across the related tasks in a joint training process [7]. The main assumption in MTL methods is that the parameters of multiple tasks lie in a low-dimensional subspace due to their correlation. Using this assumption, Argyriou *et. al.* aimed to have common features across tasks by imposing $\ell_{(2,1)}$-norm regularization on the feature matrix, and solved the convex equivalent of its objective function with this regularization [2]. Kang *et. al.* introduced a method to share the features only within group of related tasks rather than all tasks [20]. Because the strict grouping of tasks is infeasible in real-world problems, some studies suggested the overlapping groups of related tasks for sharing the parameters [24, 29]. Asymmetric multi-task learning (AMTL) provides a regularization loss by constructing the parameters of each task using the sparse and linear combination of other tasks' parameters, and penalizes the unreliable task predictors with higher loss to have less chance for knowledge transfer compared to the reliable task predictors with lower loss [25]. Furthermore, some works investigated the general idea of regularizing parameters using the task interrelations obtained via clustering-based approaches [43, 5, 11, 18].

The common form of adopting multi-task learning methods on deep neural networks is to share multiple layers among all tasks, and stack a specific layer for each task at the top. There are also some studies on designing the shared structure in deep multi-task models [48, 47, 37]. Lee *et. al.* extended AMTL to deep models (Deep-AMTFL) by allowing asymmetric knowledge transfer across tasks through latent features rather than parameters. [26]. Our MTL method for deep models differs from the previous studies, since it employs a multi-layer sub-network with low-dimension latent representations for learning task interrelations, providing an effective and scalable multi-task learning algorithm for the gene expression problem with a large number of tasks.

3 Deep Large-Scale Multi-Task Learning Network

In the problem of gene expression inference, we consider $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ as the training set with N samples, where $\mathbf{x}_i \in \mathbb{R}^D$ and $\mathbf{y}_i \in \mathbb{R}^T$ denoting the landmark and target gene expression profiles for the i-th sample respectively. T shows the number of target genes (*i.e.* output dimension) and D indicates the number of landmark genes (*i.e.* input dimension). Considering that $\mathbf{y}_i \in \mathbb{R}^T$, we have T regression tasks and our goal is to learn a multi-task regression model to estimate the target gene expressions from their corresponding landmark genes. Unless specified otherwise, we use the following notations throughout the paper. The lower and upper case letters denote the scalars (*e.g.* i, T), bold lowercase letters indicate vectors (*e.g.* $\mathbf{x}, \mathbf{w}$), the upper case letters represent matrices (*e.g.* $\mathbf{X}, \mathbf{W}$), and calligraphic letters indicate functions, sets and losses.

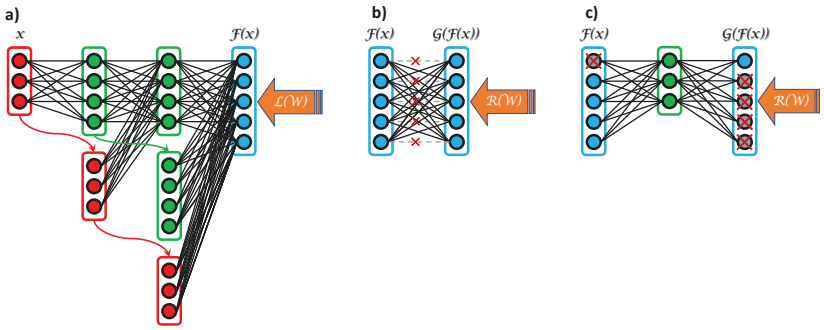


Fig. 1. Deep-LSMTL architecture. **a**) This figure illustrates the architecture of our DenseNet ($\mathcal{F}$), where each layer receives the features of all preceding layers as the input. The ℓ_1-norm loss ($\mathcal{L}$) is applied on the output of this network. **b**) This network indicates the shallow and linear $\mathcal{G}$ function used on Eq. (4). The crosses on some wights represents the zero diagonal elements constraint. **c**) This network shows the two-layer model $\mathcal{G}$ on Eq. (5), where β and $(1-\beta)$ filters are represented by the cross signs. The regularization loss ($\mathcal{R}$) is applied on the output of this layer.

3.1 Clustered Multi-Task Learning

Multi-task learning algorithms generally share the relevant knowledge among tasks by proposing a joint learning framework for the tasks. This joint learning framework usually contains a regularization term to improve generalization of the model as the following objective:

$$\min_{\mathbf{W}} \sum_{i=1}^{N}\sum_{t=1}^{T} \mathcal{L}(\mathbf{w}_t; \mathbf{x}_i, y_{it}) + \mathcal{R}(\mathbf{W}) \tag{1}$$

where the first term ($\mathcal{L}$) is the loss function applied separately on each task, and the second term ($\mathcal{R}$) is the regularization employed to enforce sharing the parameters according to the tasks relations. Note that $\mathbf{w}_t$ shows task-specified parameters as a column of $\mathbf{W} \in \mathbb{R}^{D\times T}$, if we assume a shallow regression network as our model. Although, the mean squared error (MSE) is the first choice for the loss in regression tasks, we empirically find out that the ℓ_1-norm loss function $\mathcal{L}(\mathbf{W}; \mathbf{x}_i, \mathbf{y}_i) = \|\mathbf{y}_i - \mathcal{F}(\mathbf{x}_i)\|_1$ is a better candidate in our objective, where $\mathcal{F}(\mathbf{x}_i) = \mathbf{W}\mathbf{x}_i$ is a regression model. There are also several studies in literature advocating ℓ_1-norm loss rather than MSE in different applications due to its robust performance in dealing with outliers and noisy data.

It has been shown that the shallow MTL models can be extended to deeper models by sharing a set latent features across all tasks as $\mathbf{W} = \mathbf{LS}$, where $\mathbf{L} \in \mathbb{R}^{D\times K}$ shows the shared parameters and $\mathbf{S} \in \mathbb{R}^{K\times T}$ denotes the task-specific weights [2, 24]. The same idea can be adopted in deep models to use multiple layers of shared features followed by a task-specific layer. The multi-layer perceptron (*i.e.* fully connected) network is the simplest form of a deep MTL model as $\mathcal{F}(\mathbf{x}) = \sigma(...\sigma(\sigma(\mathbf{x}\mathbf{W}^{(1)})\mathbf{W}^{(2)})...\mathbf{W}^{(L)})$, where the first $L-1$ layers are shared across all tasks and the last one is a task-specific layer. However, we employ a more efficient architecture for the shared layers by adopting the

densely connected convolutional network [16] in our inference model. Assuming the input for each layer as $\mathbf{x}^{(l)}$ where $l \in \{0, ..., L-1\}$, the output of our DenseNet is computed by $\mathbf{x}^{(l+1)} = \sigma([\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, ..., \mathbf{x}^{(l)}]\mathbf{W}^{(l+1)})$, where $[\mathbf{x}^{(0)}; , \mathbf{x}^{(1)}, ..., \mathbf{x}^{(l)}]$ represents the concatenation of features from all the previous layers. Figure 1(a) shows each layer of DenseNet receiving the features of all preceding layers as the input. The DenseNet has several advantages compared to multi-layer perceptron (MLP) such as reusing the features of previous layers, alleviating the vanishing-gradient issue in deep models, and reducing the number of parameters. The objective function in Eq. (1) can be written for our deep MTL network as follows:

$$\min_{\mathbf{W}^{(1)},...,\mathbf{W}^{(L)}} \sum_{i=1}^{N} \|\mathbf{y}_i - \mathcal{F}(\mathbf{x}_i)\|_1 + \mathcal{R}(\mathbf{W}^{(1)}, ..., \mathbf{W}^{(L)}). \tag{2}$$

To regularize the task-specific parameters, we can impose clustering-based constraints according to the task relations [43, 5, 11, 18, 25]. While the clustering constraints enforce the related tasks to share information and have similar parameters or features, they do not force all of the tasks to use shared features, and avoid the negative transfer issue where unrelated tasks adversely affect the features of correlated tasks [36]. Grouping the task-specific parameters using subspace clustering is an effective example of the clustering constraints. In the following equation, we replace the regularization term in Eq. (2) by the subspace clustering constraint:

$$\min_{\mathbf{W}^{(1)},...,\mathbf{W}^{(L)},\mathbf{V}} \sum_{i=1}^{N} \|\mathbf{y}_i - \mathcal{F}(\mathbf{x}_i)\|_1 + \lambda\|\mathbf{W}^{(L)} - \mathbf{W}^{(L)}\mathbf{V}\|_F^2 + \gamma\|\mathbf{V}\|_1 \tag{3}$$

where $\mathbf{V} \in \mathbb{R}^{T\times T}$ is the self-representation coefficient matrix with zero diagonal elements (*i.e.* $v_{tt} = 0$), showing the correlation among the T tasks. This regularization encourages the parameters of each task to be reconstructed by the linear and sparse combination of other tasks, and avoids the negative transfer issue by learning asymmetric similarity between the tasks.

In order to implement Eq. (3) in deep models seamlessly, we multiply the features of latest hidden layer into the second term loss. Since our last layer has linear activation function, we can reformulate the objective in Eq. (3) as:

$$\min_{\mathbf{W}^{(1)},...,\mathbf{W}^{(L)},\mathbf{V}} \sum_{i=1}^{N} \|\mathbf{y}_i - \mathcal{F}(\mathbf{x}_i)\|_1 + \lambda\|\mathcal{F}(\mathbf{x}_i) - \mathcal{G}(\mathcal{F}(\mathbf{x}_i))\|_F^2 + \gamma\|\mathbf{V}\|_1 \tag{4}$$

where $\mathcal{F}(\mathbf{X}_i) = [\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, ..., \mathbf{x}^{(L-1)}]\mathbf{W}^{(L)}$ is the prediction of our DenseNet model for the i-th sample, and $\mathcal{G}(\mathcal{F}(\mathbf{x}_i)) = \mathcal{F}(\mathbf{X}_i)\mathbf{V}$ can be considered as a layer stacked at the top of our DenseNet. The architecture of this layer is illustrated on Figure 1(b).

3.2 Deep Large-Scale Multi-Task Learning

The introduced model in the previous section has multiple drawbacks. First, it is not scalable to a large number of tasks. Specially, this is a critical issue in the gene expression inference problem as the number of target genes (*i.e.* output size) is very large

($\sim$10,000) and consequently the number of parameters in $\mathbf{V}$. Moreover, the shallow and linear layer $\mathcal{G}(.)$ might not capture the complex correlations among the tasks. In addition, while we know that the target genes expressions are highly correlated, there is no explicit constraint to learn a low-dimension manifold for the tasks relations.

In order to address the aforementioned issues, we introduce a new function for $\mathcal{G}(.)$ to better capture the tasks correlations in our MTL algorithm. To increase the capacity of $\mathcal{G}$ function, we replace the linear model with a two-layer network as $\mathcal{G}(\mathcal{F}(\mathbf{x}_i)) = \mathbf{V}^{(2)}\sigma(\mathbf{V}^{(1)}\mathcal{F}(\mathbf{x}_i))$, where $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)}$ are the first and second layer parameters respectively. Moreover, we are able to decrease the number of parameters in $\mathcal{G}$ by setting the number of units in its hidden layer smaller than the number of tasks. Specifically, while the shallow linear $\mathcal{G}$ function has T^2 parameters ($\sim 10^4 \times 10^4 = 10^8$), the proposed $\mathcal{G}$ contains $2TK$ free parameters, where $K \ll T$ ($\sim 2 \times 10^4 \times 100 = 2 \times 10^6$). In addition to addressing the scalability issue, a low-dimension bottleneck in $\mathcal{G}$ helps learning a low-rank representation for the tasks relations as shown in the hidden layer of Figure 1(c). The following equation shows the objective for the proposed method:

$$\min_{\mathbf{W}^{(1)},...,\mathbf{W}^{(L)},\mathbf{V}^{(1)},\mathbf{V}^{(2)}} \|\mathcal{F}(\mathbf{x}) - \mathbf{y}\|_1 + \lambda\|(1-\beta) \odot \big[\mathcal{F}(\mathbf{x}) - \mathcal{G}(\beta \odot \mathcal{F}(\mathbf{x}))\big]\|_F^2, \quad (5)$$

where β is a binary mask, and $\odot$ indicates the element-wise multiplication. The second term of this objective forces each task to be reconstructed by the other tasks, learning the relations among the tasks.

Note that reconstructing the output of each task using the other ones in the multi-layer $\mathcal{G}$ function is not as straight-forward as zeroing the diagonal elements of V in the subspace clustering constraint. To solve this problem, we use the random β mask to approximate the reconstruction process in stochastic learning approaches (*e.g.* SGD). In particular, we randomly mask one or a few tasks outputs in each training iteration (*e.g.* $\beta = [1, 0, 0, ..., 0]$), then compute the output of regularization sub-network by $\mathcal{G}(\beta \odot \mathcal{F}(\mathbf{x}))$, and finally apply the reconstruction loss only to the masked tasks via $(1-\beta)$ filter. Utilizing this approach, we seamlessly adopt the subspace clustering regularization in our deep low-rank MTL network.

4 Experiments

In this section, we evaluate our model compared to the alternative deep and shallow regression methods on multiple datasets. To do so, we first describe the experimental setups, compare Deep-LSMTL with the state-of-the-art models, and investigate the effectiveness of our MTL algorithm on neural networks with different architectures. Furthermore, we visualize the relevance of the landmark and target genes in the inference problem, providing insights into the learned knowledge in our model.

4.1 Experimental Setup

Datasets : In our experiments, we include the microarray-based *GEO* dataset, the RNA-Seq-based *GTEx* dataset and the 1000 Genomes (*1000G*) RNA-Seq expression data[3].

[3] `https://cbcl.ics.uci.edu/public_data/D-GEX/`

The original *GEO* dataset consists of 129,158 gene expression profiles corresponding to 22,268 probes (978 landmark genes and 21,290 target genes) that are collected from the Affymetrix microarray platform. The original *GTEx* dataset is composed of 2,921 profiles from the Illumina RNA-Seq platform in the format of Reads Per Kilobase per Million (RPKM). The original *1000G* dataset includes 2,921 profiles from the Illumina RNA-Seq platform in the format of RPKM.

Following the data pre-processing in [8], we remove duplicate samples, normalize joint quantile and match cross-platform data. In particular, we first remove duplicated samples. We then map the expression values in the *GTEx* and *1000G* datasets according to the quantile computed in the *GEO* data, after which the expression value has been quantile normalized from 4.11 to 14.97. Finally, we normalize the expression values of each gene to zero mean and unit variance. After pre-processing, 943 landmark genes and 9520 target genes remain in each profile. Our datasets contain 111,009 profiles in *GEO* dataset, 2,921 profiles in *GTEx* dataset and 462 profiles in the *1000G* dataset.

Following the experimental protocol in [8], we evaluate the methods under two different circumstances. First, we consider 80% of the *GEO* data for training, 10% of the *GEO* data for validation, and the other 10% of the *GEO* data for testing. Second, we use the same 80% of the *GEO* data for training, the *1000G* data for validation, and the *GTEx* data for testing. The second scenario is useful for validating the regression models on cross-platform prediction, since the training, validation and testing belong to the different distributions.

Alternative Methods : The most well-known linear inference model is the least square regression, which has the following objective function:

$$\min_{W} \sum_{i=1}^{n} ||\mathbf{x}_i\mathbf{W} - \mathbf{y}_i||_2^2 + \lambda||\mathbf{W}||_p^2 \,, \tag{6}$$

where $\mathbf{W}$ is the model parameters, and λ represents the regularization hyper-parameter. When $\lambda = 0$, we call the model as least square regression (LSR). But when $\lambda \neq 0$, we have two other linear models, LSR-L2 with ℓ_2-norm regularization (*i.e.* $p = 2$) and LSR-L1 with ℓ_1-norm regularization (*i.e.* $p = 1$). The regularization terms in LSR-L2 and LSR-L1 help the regression model to alleviate the overfitting issue. Our proposed algorithm employs the mean absolute error instead of the mean squared error for the regression loss, and also benefits from the ℓ_1-norm regularization but only in the $\mathcal{G}$ network parameters.

We also include the k-nearest neighbors (KNN) method as a baseline method, where the prediction of a given profile is calculated as the average of its k nearest profiles. In addition, we compare with two deep learning methods, D-GEX [8] and SemiGAN [12], for gene expression inference. Generally, D-GEX model uses a multi-layer perceptron neural network as the inference model, while SemiGAN is designed based on generative adversarial networks. However, our model utilizes a DenseNet architecture for its base network (i.e. $\mathcal{F}$).

We also adopt a few multi-task learning algorithms for training deep inference models in our problem. We review them in the following part very briefly, but refer the

readers to the original papers for more details. The CNMTL method aims to cluster the task-specific (*i.e.* last layer) parameters using regularizations based on the wights mean, and between-cluster and within-cluster variances as follows [18]:

$$\min_{\mathbf{W}} \sum_{i=1}^{N}\sum_{t=1}^{T}\mathcal{L}(\mathbf{w}_t,\mathbf{x}_i,\mathbf{y}_i)+\lambda_M\|\overline{\mathbf{W}}\|_F^2+\lambda_B\sum_{k=1}^{K}\|\overline{\mathbf{W}}_k-\overline{\mathbf{W}}\|_F^2 + \lambda_W\sum_{k=1}^{K}\sum_{j\in\mathcal{J}(k)}\|\mathbf{W}_j^{(L)}-\overline{\mathbf{W}}_k\|_F^2 \tag{7}$$

where the second term is the weights mean regularization with λ_M as the hyper-parameter and $\overline{\mathbf{W}}=1/T\sum_{t=1}^{T}\mathbf{W}_t^{(L)}$ as the average of last layer weights across tasks, the third term is the between-cluster variance regularization with λ_B as the hyper-parameter and $\overline{\mathbf{W}}_k$ as the average of last layer weights of the k-cluster, and the last term is the within-cluster variance regularization with λ_W as the hyper-parameter and $\mathcal{J}(k)$ representing a set of tasks belonging to the k-th cluster. Setting $\mathcal{L}(\mathbf{W};\mathbf{x}_i,\mathbf{y}_i)=\|\mathbf{y}_i-\mathcal{F}(\mathbf{x}_i)\|_1$, we have similar regression loss for the CNMTL model (and also all the following alternative models) for a fair comparison, but Deep-LSMTL uses different regularization (i.e. the second term in Eq. 5) than CNMTL.

The GO-MTL algorithm imposes ℓ_1-norm regularization on the task-specific parameters and Frobenius-norm regularization on the shared wights [24]:

$$\min_{\mathbf{W}} \sum_{i=1}^{N}\sum_{t=1}^{T}\mathcal{L}(\mathbf{w}_t,\mathbf{x}_i,\mathbf{y}_i)+\mu\|\mathbf{w}_t^{(L)}\|_1+\lambda\sum_{l=1}^{L-1}\|\mathbf{W}^{(l)}\|_F^2 \tag{8}$$

where, μ and λ are the regularization hyper-parameters. We also use ℓ_1-norm loss regularization in the $\mathcal{G}$ network parameters of Deep-LSMTL.

The AMTL method enforces each set of task-specific weights to be reconstructed by the linear combination of other tasks parameters using the following objective [26]:

$$\min_{\mathbf{W},\mathbf{V}} \sum_{i=1}^{N}\sum_{t=1}^{T}\alpha_t\mathcal{L}(\mathbf{w}_t,\mathbf{x}_i,\mathbf{y}_i)+\lambda\|\mathbf{W}^{(L)}-\mathbf{W}^{(L)}\mathbf{V}\|_2^2 \tag{9}$$

where, λ is the regularization hyper-parameter, and α_t is the coefficient representing the easiness level of the t-th task that makes the outgoing transfer from hard tasks less than the easy tasks. AMTL has similar objective to our model in regularizing the task-specific parameters, but Deep-LSMTL has more flexible two-layer sub-network $\mathcal{G}$ with a computationally less expensive and easy to apply regularization for deep models.

The AMTFL algorithm extends AMTL to regularize the features rather than the parameters [25]:

$$\min_{\mathbf{W},\mathbf{V}} \sum_{i=1}^{N}\sum_{t=1}^{T}\alpha_t\mathcal{L}(\mathbf{w}_t,\mathbf{x}_i,\mathbf{y}_i)+\mu\|\mathbf{w}_t^{(L)}\|_1+\gamma\|\mathbf{Z}-\sigma(\mathbf{Z}\mathbf{W}^{(L)}\mathbf{V})\|_F^2+\lambda\sum_{l=1}^{L-1}\|\mathbf{W}^{(l)}\|_F^2 \tag{10}$$

Table 1. Comparison of different inference models on *GEO* and *GTEx* datasets based on the MAE and CC evaluation metrics. The results of the shallow regression models in the first part and the previous deep inference networks in the second part are reported from the original papers or running their released codes. The MTL methods in the third part and our proposed models in the fourth part use densely connected architecture with different numbers of hidden units. Better results correspond to lower MAE values or higher CC values.

	Methods	*GEO* Dataset		*GTEx* Dataset	
		MAE	CC	MAE	CC
Shallow	LSR	0.3763±0.0844	0.8227±0.0956	0.4704±0.1235	0.7184±0.2072
	LSR-L1	0.3756±0.0841	0.8221±0.0960	0.4669±0.1274	0.7163±0.2188
	LSR-L2	0.3758±0.0842	0.8223±0.0959	0.4682±0.1233	0.7181±0.2076
	KNN	0.3708±0.0958	0.8218±0.1001	0.6225±0.1469	0.5748±0.2052
Deep	D-GEX	0.3204±0.0879	0.8514±0.0908	0.4393±0.1239	0.7304±0.2072
	SemiGAN	0.2997±0.0869	0.8702±0.0927	0.4223±0.1266	0.7443±0.2087
MTL	Deep-GO-MTL	0.2931±0.0934	0.8717±0.1075	0.4201±0.1391	0.7434±0.2153
	Deep-CNMTL	0.2946±0.0928	0.8704±0.1080	0.4199±0.1393	0.7401±0.2163
	Deep-AMTL	0.2942±0.0936	0.8719±0.1072	0.4238±0.1388	0.7368±0.2164
	Deep-AMTFL	0.2947±0.0930	0.8703±0.1081	0.4205±0.1390	0.7428±0.2154
Ours	DenseNet	0.2924±0.0945	0.8727±0.1070	0.4227±0.1388	0.7416±0.2156
	Deep-LSMTL	**0.2887±0.0949**	**0.8753±0.1062**	**0.4162±0.1390**	**0.7510±0.2166**

where, μ, λ and γ are the regularization hyper-parameters, α_t is the task easiness coefficient, and $\mathbf{Z}$ is the output of the last hidden layer. AMTFL aims to regularize its model using a reconstruction loss on the shared features (only the last hidden layer). Although, our regularization term can also be seen as a reconstruction loss, Deep-LSMTL applies the regularization on the predictions (not the last hidden layer features) and benefits from non-linear and easy to implement reconstruction sub-network.

Evaluation Metrics : We use mean absolute error (MAE) and concordance correlation (CC) as the evaluation metrics. Given the testing data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{M}$, for a certain model, we denote the predicted expressions as $\{\hat{\mathbf{y}}_i\}_{i=1}^{M}$. The MAE is then computed using

$$MAE_t = \frac{1}{M}\sum_{i=1}^{M}|\hat{y}_{it} - y_{it}|\,, \tag{11}$$

where MAE_t indicates the mean absolute error for the t-th task, y_{it} shows the ground truth expression value for the t-th target gene in the i-th testing profile, and $\hat{y}_{it}$ represents the corresponding predicted value. The definition of CC is

$$CC_t = \frac{2\rho\sigma_{\mathbf{y}_t}\sigma_{\hat{\mathbf{y}}_t}}{\sigma^2_{\mathbf{y}_t} + \sigma^2_{\hat{\mathbf{y}}_t} + (\mu_{\mathbf{y}_t} - \mu_{\hat{\mathbf{y}}_t})^2}\,, \tag{12}$$

where CC_t shows the concordance correlation for the t-th target gene. ρ is the Pearson correlation, and $\mu_{\mathbf{y}_t}$, $\mu_{\hat{\mathbf{y}}_t}$, and $\sigma_{\mathbf{y}_t}$, $\sigma_{\hat{\mathbf{y}}_t}$ are the mean and standard deviation of $\mathbf{y}_t$ and $\hat{\mathbf{y}}_t$ respectively. Note that in addition to the mean values of the absolute error and concordance correlation via $MAE_{mean} = 1/T\sum_{t=1}^{T} MAE_t$ and $CC_{mean} =$

$1/T \sum_{t=1}^{T} CC_t$, we report the standard deviation across the tasks for each inference model.

Implementation Details : In our model, we use a DenseNet structure with three hidden layers and $9,000$ hidden units on each layer. Leaky rectified linear unit [28] with leakiness ratio 0.2 is used as our activation function, and Adam algorithm [22] is employed as our optimization method. Moreover, we decrease our learning rates from 1×10^{-3} to 1×10^{-5} linearly from the first epoch to the maximum epoch 500. The batch size is set to 100. We also utilize batch normalization [17] as the layer normalization to speed up the convergence of the training process. The parameters of all layers are initialized by Xavier approach [13]. We also select the dropout probability, λ, and number of hidden units in subspace layer from $dropout^{set} = \{0.05, 0.1, 0.25\}$, $\lambda^{set} = \{0.1, 1, 10\}$, and $units^{set} = \{500, 1000, 2000\}$ respectively based on the validation results. We use Pytorch toolbox for writing our code, and run the algorithm in a machine with one Titan X pascal GPU.

4.2 Performance Comparison

We compare the performance of Deep-LSMTL with other models on *GEO* and *GTEx* datasets. As shown in Table 1, the alternative models are grouped as the shallow regression models in the first part, the previous deep regression networks in the second part, the MTL algorithms applied on deep regression models in the third part, and our DenseNet baseline and Deep-LSMTL network in the fourth part of the table. Regarding the MTL methods and our Deep-LSMTL network, we try to run the largest possible network with three hidden-layers on one GPU. The number of hidden-units for Deep-Go-MTL, Deep-CNMTL, Deep-AMTFL, Deep-AMTL and Deep-LSMTL are 8000, 4000, 5000, 7000 and 9000 respectively.

The MAE and CC results show that Deep-LSMTL significantly and consistently outperforms all of the alternative models on both *GEO* and *GTEx* datasets. As expected, Deep-LSMTL has large improvements against the shallow models, indicating the importance of deeper networks in capturing the complex nature of gene expression data. Deep-LSMTL also achieves better results than the existing deep inference models in the literature, proving the advantages of using the task interrelations in our MTL algorithm. Moreover, Deep-LSMTL not only shows better results compared to other MTL methods, but it also indicates the need for far less GPU memory than the other MTL methods.

Since the expressions of target genes are normalized, the direct comparisons of the errors may not be conclusive. In order to check if the improvement of Deep-LSMTL over the alternative models is statistically significant, we use the 5×2 cross validation method in [9]. In particular, we repeat 2-fold cross-validation of Deep-LSMTL and the best alternative model on *GEO* dataset (*i.e.* DenseNet) 5 times, and use a paired student's t-test on the MAE results. Based on the obtained p-values that is much less than 5%, we reject the null hypothesis that the results of the two models have the same distribution. Thus we can claim that Deep-LSMTL has statistically significant improvements compared to the other alternative models.

Table 2. Comparison of MTL algorithms for the gene expression inference problems on *GEO* and *GTEx* datasets. All of the models use a two-hidden layers DenseNet as their structure, but have different numbers of hidden units in each part of the table. Better results correspond to lower MAE value or higher CC value.

Methods	*GEO* Dataset		*GTEx* Dataset		# params	# units
	MAE	CC	MAE	CC		
Deep-GO-MTL	0.3087±0.0912	0.8602±0.1120	0.4264±0.1384	0.7347±0.2179	8.08×10^7	3000
Deep-CNMTL	0.3070±0.0912	0.8625±0.1104	0.4263±0.1390	0.7322±0.2188	8.08×10^7	
Deep-AMTL	0.3073±0.0912	0.8621±0.1105	0.4265±0.1385	0.7322±0.0000	1.71×10^8	
Deep-AMTFL	0.3088±0.0912	0.8599±0.1121	0.4263±0.1383	0.7346±0.2180	1.47×10^8	
Deep-LSMTL	**0.3034±0.0914**	**0.8626±0.1153**	**0.4258±0.1383**	**0.7377±0.2188**	9.98×10^7	
Deep-GO-MTL	0.3014±0.0922	0.8665±0.1099	0.4267±0.1388	0.7366±0.2178	1.7×10^8	6000
Deep-CNMTL	0.2992±0.0923	0.8696±0.1079	0.4260±0.1388	0.7345±0.2179	1.7×10^8	
Deep-AMTL	0.2999±0.0924	0.8688±0.1085	0.4262±0.1388	0.7351±0.2175	2.61×10^8	
Deep-AMTFL	0.3016±0.0922	0.8664±0.1100	0.4265±0.1387	0.7371±0.2172	2.94×10^8	
Deep-LSMTL	**0.2951±0.0927**	**0.8692±0.1089**	**0.4234±0.1391**	**0.7397±0.2174**	1.89×10^8	
Deep-GO-MTL	0.2983±0.0929	0.8693±0.1089	0.4268±0.1386	0.7376±0.2167	2.78×10^8	9000
Deep-AMTL	0.2972±0.0932	0.8713±0.1077	0.4268±0.1386	0.7367±0.2170	3.69×10^8	
Deep-LSMTL	**0.2919±0.0934**	**0.8717±0.1080**	**0.4201±0.1391**	**0.7439±0.2170**	2.97×10^8	

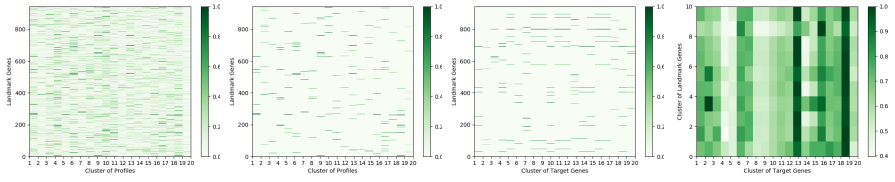
Table 3. MAE comparison of D-GEX and Deep-LSMTL on *GEO* and *GTEx* datasets, when the number of hidden layers varies from 1 to 3, and the number of hidden units are 3000, 6000 or 9000. The structure of both models are based on the MLP network.

Methods	*GEO* Dataset			*GTEx* Dataset			# hidden layers
	# hidden units			# hidden units			
	3000	6000	9000	3000	6000	9000	
D-GEX	0.3421±0.0858	0.3337±0.0869	0.3300±0.0874	0.4507±0.1231	0.4428±0.1246	0.4394±0.1253	**1**
	0.3377±0.0854	0.3280±0.0869	0.3224±0.0879	0.4586±0.1194	0.4446±0.1226	0.4393±0.1239	**2**
	0.3362±0.0850	0.3252±0.0868	0.3204±0.0879	0.5160±0.1157	0.4595±0.1186	0.4492±0.1211	**3**
Deep-LSMTL	0.3179±0.0901	0.3097±0.0903	0.3054±0.0903	0.4363±0.1368	0.4349±0.1369	0.4295±0.1380	**1**
	0.3086±0.0908	0.2985±0.0915	0.2944±0.0916	0.4338±0.1374	0.4321±0.1371	0.4289±0.1379	**2**
	0.3067±0.0913	0.2965±0.0922	**0.2927±0.0923**	0.4301±0.1379	0.4286±0.1373	**0.4253±0.1383**	**3**

4.3 Ablation Study

While the previous experiments confirm the effectiveness of Deep-LSMTL in dealing with large-scale tasks by fitting a larger network on one GPU compared to other MTL methods, we design another experiment to compare the MTL methods with the same structure. To do so, we consider the two-hidden-layer DenseNet architecture for all the MTL methods in three different settings with 3000, 6000, and 9000 hidden units. Table 2 shows the results of Deep-GO-MTL, Deep-CNMTL, Deep-AMTL, Deep AMTFL, and Deep-LSMTL on both *GEO* and *GTEx* Datasets. Note that there are still out-of-memory issues for Deep-CNMTL and Deep-AMTFL with 9000 hidden units. The results in Table 2 indicate better performance for Deep-LSMTL compared to the other MTL models on different architectures. Thus, Deep-LSMTL not only provides a better scalable model in our inference problem, it also shows better performance even when the base network structure is similar.

In addition to investigating the effectiveness of Deep-LSMTL on the different base network than DenseNet, we compare Deep-LSMTL and D-GEX with MLP structure in Table 3. We report the results for both models, where MLP network has one, two or three hidden layers and the hidden layers have 3000, 6000 or 9000 hidden units. Deep-



(a) (b) (c) (d)

Fig. 2. Visualization of the relevance score calculated for each landmark gene on GEO dataset. **a)** Relevance score of landmark genes *w.r.t.* cluster of profiles. We grouped the gene expression profiles into 20 clusters using K-means, and plot the contribution of each landmark gene to different clusters of profiles. **b)** Cleaned version of landmark gene score. For each profile cluster, only the top 20 landmark genes in (a) are kept for clear visualization. **c)** Relevance score of landmark genes *w.r.t.* cluster of target genes. We divide the 9520 target genes into 20 clusters via K-means, and demonstrate the contributions of cleaned landmark genes. **d)** Relevance score of landmark gene clusters *w.r.t.* cluster of target genes. The landmark genes are clustered into 10 clusters, and their contributions in predicting different clusters of target genes is plotted.

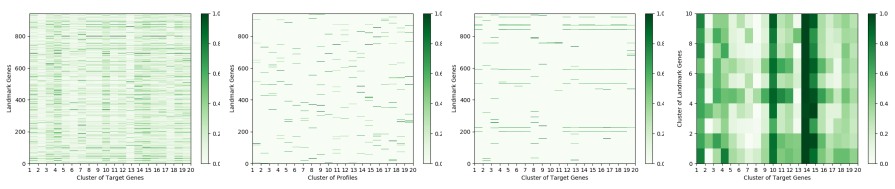
LSMTL again outperforms D-GEX in all architectures consistently, and confirms its capability regardless of the base network structure.

4.4 Visualization

We perform a qualitative study on Deep-LSMTL to show the role of different landmark genes in the gene expression inference problem. In order to plot visualization figures, we adopt the Layer-wise Relevance Propagation (LRP) [4] method to calculate the importance of landmark genes that is learned in our model. Figure 2 shows the results of Deep-LSMTL with DenseNet structure (in Table 1) on *GEO* dataset. First, we divide the gene expression profiles into 20 clusters and then use LRP to calculate the relevance score of landmark genes *w.r.t.* each profile cluster in Figure 2(a) and 2(b). These figures show that the landmark gene expression patterns are different for various profile groups, replicating the findings in previous cancer sub-type discovery and cancer landscape study that different groups of samples usually exhibit different expression patterns [40, 19].

Next, we analyze the relationship between landmark genes and target genes. We cluster the target genes into 20 groups and calculate the overall relevance score of landmark genes in the prediction of each target gene cluster in Figure 2(c). For the sake of better visualization, we also group the landmark genes into 10 clusters and display the association between landmark gene clusters and target gene clusters in Figure 2(d). We notice an apparent difference in the relevance patterns for different target gene clusters, yet some similarity among certain clusters. This finding has also been validated by the previous gene cluster analysis [30], where genes cluster information is related to the structure of biosynthetic pathways and metabolites.

We also visualize the predictions of our model on GTEx dataset in Fig. 3 similar to GEO dataset. The figures show similar patterns as the previous outcomes. However,



(a) (b) (c) (d)

Fig. 3. Visualization of the relevance score calculated for each landmark gene on GTX dataset. **a)** Relevance score of landmark genes *w.r.t.* cluster of profiles. We grouped the gene expression profiles into 20 clusters using K-means, and plot the contribution of each landmark gene to different clusters of profiles. **b)** Cleaned version of landmark gene score. For each profile cluster, only the top 20 landmark genes in (a) are kept for clear visualization. **c)** Relevance score of landmark genes *w.r.t.* cluster of target genes. We divide the 9520 target genes into 20 clusters via K-means, and demonstrate the contributions of cleaned landmark genes. **d)** Relevance score of landmark gene clusters *w.r.t.* cluster of target genes. The landmark genes are clustered into 10 clusters, and their contributions in predicting different clusters of target genes is plotted.

they are more notable because of training on GEO data and predicting on GTEx data, qualitatively confirming the capability of our proposed model in capturing the relations among genes even for cross-platform prediction.

5 Conclusion

In this paper, we proposed a novel multi-task learning algorithm for training deep regression models on the gene expression inference problem. Our proposed method efficiently exploits the task interrelations to improve the generalizations of the predictors. We introduced a regularization on our learning framework that is easy to implement on deep models and scalable to a large number of tasks. We validated our model on two gene expression datasets, and found consistent and significant improvements over all counterparts regardless of the base network architecture. Furthermore, we interpreted the role of landmark genes in the inference of target genes expression using visualization figures, providing insights into the information captured by our model.

References

1. Alipanahi, B., Delong, A., Weirauch, M.T., Frey, B.J.: Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. Nature biotechnology **33**(8), 831 (2015). https://doi.org/10.1038/nbt.3300
2. Argyriou, A., Evgeniou, T., Pontil, M.: Convex multi-task feature learning. vol. 73, pp. 243–272. Springer (2008)
3. Association, A., et al.: 2013 alzheimer's disease facts and figures. vol. 9, pp. 208–245. Elsevier (2013)

4. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7), e0130140 (2015)
5. Bakker, B., Heskes, T.: Task clustering and gating for bayesian multitask learning. vol. 4, pp. 83–99 (2003)
6. Brazma, A., Parkinson, H., Sarkans, U., Shojatalab, M., Vilo, J., Abeygunawardena, N., Holloway, E., Kapushesky, M., Kemmeren, P., Lara, G.G., et al.: Arrayexpress—a public repository for microarray gene expression data at the ebi. Nucleic acids research **31**(1), 68–71 (2003)
7. Caruana, R.: Multitask learning. vol. 28, pp. 41–75. Springer (1997)
8. Chen, Y., Li, Y., Narayan, R., Subramanian, A., Xie, X.: Gene expression inference with deep learning. Bioinformatics **32**(12), 1832–1839 (2016)
9. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. vol. 10, pp. 1895–1923. MIT Press (1998)
10. Edgar, R., Domrachev, M., Lash, A.E.: Gene expression omnibus: Ncbi gene expression and hybridization array data repository. Nucleic acids research **30**(1), 207–210 (2002)
11. Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. vol. 6, pp. 615–637 (2005)
12. Ghasedi Dizaji, K., Wang, X., Huang, H.: Semi-supervised generative adversarial network for gene expression inference. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1435–1444. ACM (2018)
13. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. pp. 249–256 (2010)
14. Guo, X., Zhang, Y., Hu, W., Tan, H., Wang, X.: Inferring nonlinear gene regulatory networks from gene expression data based on distance correlation. PloS one **9**(2), e87446 (2014)
15. Heimberg, G., Bhatnagar, R., El-Samad, H., Thomson, M.: Low dimensionality in gene expression data enables the accurate extraction of transcriptional programs from shallow sequencing. Cell systems **2**(4), 239–250 (2016)
16. Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L.: Densely connected convolutional networks (2017)
17. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning (ICML). pp. 448–456 (2015)
18. Jacob, L., Vert, J.p., Bach, F.R.: Clustered multi-task learning: A convex formulation. In: Advances in neural information processing systems (NIPS). pp. 745–752 (2009)
19. Kandoth, C., McLellan, M.D., Vandin, F., Ye, K., Niu, B., Lu, C., Xie, M., Zhang, Q., McMichael, J.F., Wyczalkowski, M.A., et al.: Mutational landscape and significance across 12 major cancer types. Nature **502**(7471), 333 (2013)
20. Kang, Z., Grauman, K., Sha, F.: Learning with whom to share in multi-task feature learning. In: International Conference on Machine Learning (ICML). pp. 521–528 (2011)
21. Keenan, A.B., Jenkins, S.L., Jagodnik, K.M., Koplev, S., He, E., Torre, D., Wang, Z., Dohlman, A.B., Silverstein, M.C., Lachmann, A., et al.: The library of integrated network-based cellular signatures nih program: system-level cataloging of human cells response to perturbations. Cell systems (2017)
22. Kingma, D., Ba, J.: Adam: A method for stochastic optimization (2014)
23. Kishan, K., Li, R., Cui, F., Yu, Q., Haake, A.R.: Gne: a deep learning framework for gene network inference by aggregating biological information. BMC systems biology **13**(2), 38 (2019)

24. Kumar, A., Daumé III, H.: Learning task grouping and overlap in multi-task learning. In: Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML). pp. 1723–1730. Omnipress (2012)
25. Lee, G., Yang, E., Hwang, S.: Asymmetric multi-task learning based on task relatedness and loss. In: International Conference on Machine Learning (ICML). pp. 230–238 (2016)
26. Lee, H., Yang, E., Hwang, S.J.: Deep asymmetric multi-task feature learning. In: Proceedings of the 35th International Coference on International Conference on Machine Learning (ICML) (2018)
27. Leung, M.K., Xiong, H.Y., Lee, L.J., Frey, B.J.: Deep learning of the tissue-regulated splicing code. Bioinformatics **30**(12), i121–i129 (2014)
28. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: International Conference on Machine Learning (ICML). vol. 30 (2013)
29. Maurer, A., Pontil, M., Romera-Paredes, B.: Sparse coding for multitask and transfer learning. In: International Conference on Machine Learning (ICML). pp. 343–351 (2013)
30. Medema, M.H., Kottmann, R., Yilmaz, P., Cummings, M., Biggins, J.B., Blin, K., De Bruijn, I., Chooi, Y.H., Claesen, J., Coates, R.C., et al.: Minimum information about a biosynthetic gene cluster. Nature chemical biology **11**(9), 625 (2015)
31. Nelms, B.D., Waldron, L., Barrera, L.A., Weflen, A.W., Goettel, J.A., Guo, G., Montgomery, R.K., Neutra, M.R., Breault, D.T., Snapper, S.B., et al.: Cellmapper: rapid and accurate inference of gene expression in difficult-to-isolate cell types. Genome biology **17**(1), 201 (2016)
32. Ntranos, V., Kamath, G.M., Zhang, J.M., Pachter, L., David, N.T.: Fast and accurate single-cell rna-seq analysis by clustering of transcript-compatibility counts. Genome biology **17**(1), 112 (2016)
33. Peck, D., Crawford, E.D., Ross, K.N., Stegmaier, K., Golub, T.R., Lamb, J.: A method for high-throughput gene expression signature analysis. vol. 7, p. R61. BioMed Central (2006)
34. Rees, M.G., Seashore-Ludlow, B., Cheah, J.H., Adams, D.J., Price, E.V., Gill, S., Javaid, S., Coletti, M.E., Jones, V.L., Bodycombe, N.E., et al.: Correlating chemical sensitivity and basal gene expression reveals mechanism of action. Nature chemical biology **12**(2), 109 (2016)
35. Richiardi, J., Altmann, A., Milazzo, A.C., Chang, C., Chakravarty, M.M., Banaschewski, T., Barker, G.J., Bokde, A.L., Bromberg, U., Büchel, C., et al.: Correlated gene expression supports synchronous activity in brain networks. Science **348**(6240), 1241–1244 (2015)
36. Ruder, S.: An overview of multi-task learning in deep neural networks (2017)
37. Ruder, S., Bingel, J., Augenstein, I., Søgaard, A.: Learning what to share between loosely related tasks (2017)
38. Shah, S., Lubeck, E., Zhou, W., Cai, L.: In situ transcription profiling of single cells reveals spatial organization of cells in the mouse hippocampus. Neuron **92**(2), 342–357 (2016)
39. Singh, R., Lanchantin, J., Robins, G., Qi, Y.: Deepchrome: deep-learning for predicting gene expression from histone modifications. Bioinformatics **32**(17), i639–i648 (2016)
40. Speicher, N.K., Pfeifer, N.: Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery. Bioinformatics **31**(12), i268–i275 (2015)
41. Spencer, M., Eickholt, J., Cheng, J.: A deep learning network approach to ab initio protein secondary structure prediction. IEEE/ACM transactions on computational biology and bioinformatics **12**(1), 103–112 (2015)
42. Stephens, P.J., Tarpey, P.S., Davies, H., Van Loo, P., Greenman, C., Wedge, D.C., Nik-Zainal, S., Martin, S., Varela, I., Bignell, G.R., et al.: The landscape of cancer genes and mutational processes in breast cancer. Nature **486**(7403), 400 (2012)
43. Thrun, S., O'Sullivan, J.: Discovering structure in multiple learning tasks: The tc algorithm. In: International Conference on Machine Learning (ICML). vol. 96, pp. 489–497 (1996)

44. Van't Veer, L.J., Dai, H., Van De Vijver, M.J., He, Y.D., Hart, A.A., Mao, M., Peterse, H.L., Van Der Kooy, K., Marton, M.J., Witteveen, A.T., et al.: Gene expression profiling predicts clinical outcome of breast cancer. nature **415**(6871), 530 (2002)
45. Wang, Z., He, Z., Shah, M., Zhang, T., Fan, D., Zhang, W.: Network-based multi-task learning models for biomarker selection and cancer outcome prediction. Bioinformatics (2019)
46. Yan, W., Wei, J., Deng, X., Shi, Z., Zhu, Z., Shao, D., Li, B., Wang, S., Tong, G., Ma, Z.: Transcriptional analysis of immune-related gene expression in p53-deficient mice with increased susceptibility to influenza a virus infection. BMC medical genomics **8**(1), 52 (2015)
47. Yang, Y., Hospedales, T.: Deep multi-task representation learning: A tensor factorisation approach. In: International Conference on Learning Representations (ICLR) (2017)
48. Yang, Y., Hospedales, T.M.: Trace norm regularised deep multi-task learning (2016)
49. Yıldırım, M.A., Goh, K.I., Cusick, M.E., Barabási, A.L., Vidal, M.: Drug-target network. Nature biotechnology **25**(10), 1119–1126 (2007)
50. Zhou, J., Troyanskaya, O.G.: Predicting effects of noncoding variants with deep learning–based sequence model. Nature methods **12**(10), 931 (2015)