CNNC: Convolutional Neural Networks for Co-Expression Analysis

Ye Yuan[1], Ziv Bar-Joseph[1,2*]

[1]Machine Learning Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA. [2]Computational Biology Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA.

*e-mail: zivbj@cs.cmu.edu

Abstract

Co-expression analysis has been extensively used in genomics studies and tools for over two decades. To date, most methods for such analysis are unsupervised and symmetric. Such methods cannot infer causality and are prone to both overfitting and false negatives resulting from differences between cells in bulk studies. Here we present a new, supervised method based on convolutional neural networks (CNNs) for co-expression analysis. We use a normalized histogram image of gene pair co-expression as the input to the CNN. Testing our method on several co-expression prediction tasks we show that it outperforms prior methods and that scRNA-Seq data leads to more accurate results when compared to bulk data. The method can be directly extended to integrate sequence and epigenetic data and to infer causal relationships.

Supporting website with software and data: https://github.com/xiaoyeye/CNNC.

Introduction

Co-expression analysis, which seeks to identify genes that are correlated or anti-correlated across a large number of samples or time points, has been a key research area of computational genomics for almost two decades[1-5]. In addition to the identification of pairs of related genes, co-expression analysis serves as an initial step in many of the most widely used computational methods for the analysis of genomics data including various clustering methods[6], network inference and reconstruction approaches[7-11], methods for classification based on genes expression[3] and many more.

Given its centrality for several downstream applications, much work has focused on improving the ability to infer correlated and anti-correlated genes. The most popular method is based on Pearson correlation analysis[12]. Such analysis focuses on shared trends rather than exact values. Other popular and widely used methods involve mutual information (MI)[13-15], nonparametric methods, for example Spearman correlation[16] methods based on alignment[17] and more[18-20].

While the above methods were shown to be useful in many applications, they also suffer from serious drawbacks. The first major issue is overfitting. Given the large number of genes that are profiled, and the often relatively small (at least in comparison) number of samples, several genes that are determined to be co-expressed may only reflect chance or noise in the data[21]. In addition, to date most co-expression analysis utilized bulk gene expression data (either array or RNA-Seq). In such data, correlations may be obscured by the different cell populations such that even if two genes appear highly correlated, it may be that they are actually never expressed in the same cell at the same time[22]. Finally, most of the widely used co-expression analysis methods are symmetric which means that each pair has only one co-expression value. While this is advantageous for some applications (for example, clustering) it may be problematic for methods that aim at inferring causality (for example, network reconstruction methods).

To address these issues we developed a new tool, CNNC which provides a

supervised way (that can be tailored to the condition / question of interest) to perform co-expression analysis. The method utilizes both bulk and single cell RNA-Seq (scRNA-Seq) data from tens of thousands of experiments, allowing us to overcome cell population confounders. Our method utilizes CNNs which we tailor for the gene expression analysis by representing input data for each pair of genes as an (image) histogram. The network is trained with positive and negative examples for the specific domain of interest (for example, known targets of a TF in a specific cell type, known pathways in a specific biological process etc.). Depending on the input data the training can be either symmetric or directed (for example, training the network to infer that TF A regulate gene B but not vice versa). To reduce overfitting CNNC determines specific thresholds based on the training for calling a pair correlated or anti-correlated or for inferring causality.

We applied CNNC to data from tens of thousands of single cell and bulk experiments. We noticed that scRNA-Seq data greatly improves performance when compared to bulk RNA-Seq. Using the same expression data to learn different CNNs (by varying the labels based on the specific domain the network was applied to) we show that CNNC outperforms prior co-expression analysis methods both for directly inferring interactions (including TF-gene and protein-protein interactions) and as a component in algorithms for the reconstruction of known pathways and clustering. Finally, we discuss the accuracy of the directionality predictions which are unique to CNNC and shown that these predictions provide important information for determining missing interactions in known pathways.

Results

We developed a general computational framework for supervised co-expression analysis (**Fig. 1**). CNNC is based on CNN which is used to analyze summarized co-expression histograms from pairs of genes from bulk and scRNA-Seq data. Given a relatively small labeled set of positive pairs (with either negative or random pairs serving as negative) the CNN learns to discriminate between interacting and / or causal pairs and negative pairs. Once trained the CNN can be used to predict co-expression scores for all gene pairs.

Learning a CNNC model

We used processed scRNA-Seq data that was collected from over 500 different studies representing a wide range of cell types, conditions etc[23]. All raw data was uniformly processed and assigned to a pre-determined set of more than 20,000 mouse genes (Methods). We also used bulk RNA-Seq RPKM data from Encyclopedia of DNA Elements (ENCODE) project[24], which contains 58 mouse tissues or cell types. For both datasets we first generated a normalized empirical probability distribution function (NEPDF) for each gene pair (genes *a* and *b*) based on their expression in the scRNA-Seq or bulk RNA-Seq data (**Fig. 1**, left). We calculated their normalized 2-dimension (2D) histogram and fixed its size at 32X32, where columns represent gene *a* expression levels and rows represent gene *b* such that entries in the matrix represent the (normalized) co-occurrences of these values. See Methods for details. Bulk and sc NEPDF were then either used separately or concatenated to form a combined NEPDF with dimension of 32X64. Next, the histogram matrix is used as input to a CNN which is trained using a N-dimension (ND) output label vector, where N depends on specific tasks. In our case N can either be 1 (interacting or not) or 3 in which case label 0 indicates that genes *a* and *b* are not interacting and label 1 (2) indicates that gene *a* (*b*) regulates gene *b* (*a*). Because of the final 'softmax' layer classification utilized by CNNs, for a three-label task CNNC's output is a vector consisting of three respective probabilities, [*p0, p1, p2*], which sum to 1. In general,

our CNN model consists of one 32X32 or 32X64 input layer, ten intermediate layers including six convolutional layers, three maxpooling layers, one flatten layer, and a final ND 'softmax' layer or one scalar 'Sigmoid' layer (**Methods** and **Supplementary Fig. 1**).

In addition to gene expression data, we can easily integrate other data types including Dnase-seq, PWM, etc. For this, we concatenated the additional information as a vector to the intermediate output of the gene expression data and continued with the standard CNN architecture. See **Methods** and **Supplementary Fig. 1** for complete details and **Supplementary Table 1** for information on training and run time.

Using CNNC to predict TF-gene interactions

Chromatin immunoprecipitation (ChIP)-seq has been widely used as a gold standard for studying cell-specific protein-DNA interactions[25]. Here we first evaluated CNNC's performance on regulator-target prediction based on data from the GTRD ChIP-seq database[26].

We extracted data from GTRD for 41 TFs for which ChIP-seq experiments were performed in mouse embryonic stem cell (mESC). To determine targets for each TF based on the ChIP-seq data from GTRD, we followed ref 27 and 28 [27, 28] and defined the promotor region as 10KB upstream to 1KB downstream from the transcription start site (TSS) for each gene. If a TF X has at least one detected peak signal with p-value smaller than 10^{-300} in or overlapping the promotor region of gene Y, we say that TF X regulates gene Y. We also used this data to compare CNNC with the two most popular co-expression analysis methods: Pearson correlation (PC) and mutual information (MI) and to compare the accuracy of predictions based on the sc and bulk RNA-Seq data. Since the prior methods used for comparison are symmetric, we focused here on the two labels setting (interacting or not). We later discuss causality inference on this data. To compare the methods and data types we performed leave-one-TF-out cross validation analysis. For each of the 41TFs, we trained CNNC with all other TFs and used the left out TF for testing (**Methods**).

Fig. 2 presents the results of this comparison analysis. First, we see that for all methods scRNA-Seq data (left column) provides much more information when compared to bulk (middle column). Note that while we had more scRNA-Seq profiles in our training set when compared to bulk experiments, these actually represent much fewer cells and conditions than those used in the bulk data. We have also tested the performance when using the same number of bulk and scRNA profiles (**Supplementary Fig. 2**). We found that even with this very small number of scRNA-Seq profiles (with much fewer cells than the bulk) CNNC performs better when using scRNA-Seq. These results support prior claims about convolution effects resulting from population of cells that make target inference harder when using bulk data[22, 29]. Still, bulk data did include some useful information and for all methods since the joint sc and bulk data performed best when compared to individual data type on its own. As for the methods themselves, for all types of input data, CNNC outperforms the other two methods. This is especially noticeable when using the scRNA-Seq (and combined) data where CNNC is 15% more accurate than MI and close to 20% more accurate than PC. The difference is even more pronounced for the top ranked predictions. Here, for CNNC we see almost no false negatives for the first 15% of ranked pairs (inset, **Fig. 2i**).

Data Integration further improves TF target gene prediction

The above analysis was only based on using expression values. However, as noted above, gene co-expression is often used as a component in more extensive procedures that often integrate different types of genomics data. To test how the use of the NN based method can aid such procedures we combined the co-expression values obtained by our method and the other methods with sequence and DNase hypersensitivity data. For sequence, we used PWMs for the TFs we tested from the Jaspar website[30]. We have also used Dnase-seq data for the same cell line from the mouse ENCODE project[24]. While there are several different methods for integrating expression, sequence and DNase data, since our main focus here is on the

co-expression analysis methods we used a simple strategy for processing the PWM and DNase data (**Methods**) which resulted in an additional 2D vector as input for each pair which we embedded to create a 512D vector (**Fig. 1 and Methods**). We next extended the CNN to utilize the additional data by concatenating it with the NEPDF's 512D vector in the flatten layer to form a 1024D vector as shown in **Fig. 1** and **Supplementary Fig. 1**.

Results, presented in **Fig. 2j**, show that these additional data sources indeed improve the ability to predict TF-gene interactions. However, as before a combined framework utilizing our CNNC method for co-expression analysis outperformed a method that used both MI and PC. Thus, the NN based approach can successfully replace other methods as a component in a more elaborated systems biology framework for inferring interactions.

CNNC can predict pathway regulator-target gene pairs

While TFs usually directly impact the resulting expression profile of their target genes (and so co-expression analysis seems like a natural option to study such interactions) several methods have also utilized RNA-Seq data to infer pathways that combine protein-protein and protein-DNA interactions[31-35]. To test whether CNNC can serve as a component in pathway inference methods we selected two representative pathway databases, KEGG[36] and Reactome[37] as gold standard and used these to train and test our co-expression framework. Since we are interested in causal relationships we only used directed edges with activation or inhabitation edge types and filtered out cyclic gene pairs where genes regulate each other mutually (to allow for a unique label for each pair). As for the negative data, here we limited the negative set to a random set of pairs where both genes appear in pathways in the database but do not interact. Here leave-one-gene-out cross validation strategy requires extremely large computing resources due to the large number of genes with outgoing directed edges (3,057 for KEGG and 2,519 for Reactome). Instead, we performed a three-fold cross validation where we kept the set of genes for which we predicted interactions

completely separated (so a gene in the test set does not have any interaction in the training set). The positive data was uniformly divided by the outgoing gene into three equal sized outgoing subsets, CNNC was trained using any two subsets and evaluated using the left subset, and then the test ROCs for each outgoing gene in the three subsets were calculated (**Methods**). Results are presented in **Fig. 3**. As can be seen, CNNC performs very well on the KEGG pathways (See **Supplementary Fig. 3** for the different folds) and also performs quite well on Reactome pathways (see also **Supplementary Fig. 3**). In contrast, the other co-expression analysis methods do not perform as well on these datasets (**Supplementary Fig. 4**).

Using CNNC for casualty prediction

So far we focused on general interaction predictions, which is what most symmetric co-expression analysis are aimed at. However, as discussed above CNNC can also be used to infer directionality by changing the output of the NN.

We next used CNNC to infer causal edges for all three datasets we studied (for TF-gene interactions causal relationships are clear, for the pathway database we only analyzed directed edges and so had the ground truth for that data as well). As can be seen in **Fig. 4**, when using the TF GTRD dataset, CNNC achieves a median AUROC of 0.8227 (**Fig. 4a**), with 32 of the 41 TFs obtaining an AUROC of more than 0.5 on this leave-one-TF-out classification task. Interestingly, as can be seen in **Supplementary Fig. 5**, when only using bulk RNA-Seq data performance on the GTRD data prediction is very weak. Thus, for the causality inference task scRNA-Seq data is the only one that can provide enough information. For KEGG, CNNC is very successful achieving a median AUROC of 0.9949 (**Fig. 4c**) (See **Supplementary Fig. 6** for the different folds). For Reactome (**Fig. 4e**) we see that the most confident predictions are correct, but beyond the top predictions performance levels off (See **Supplementary Fig. 6** for the different folds). To try to understand the process used by the NN to distinguish causal directions we plotted two NEPDF inputs to the NN (**Figs. 4g and 4h**) which were correctly predicted as two different labels (1 for **4g** and

2 for **4h**). As can be seen, in both inputs the two genes display partial correlations and there are places where both are up or down concurrently. However, the main difference between the histograms in **4g** and **4h** are cases where one gene is up and the other is not. In **4g** gene 2 is up while gene 1 is not indicating that the causal relationship is likely g1 -> g2. The opposite holds for **4h** and so the method infers that g2 -> g1 for that input. Thus, unlike the prior symmetric methods, the NEPDF that serves as input provides important clues that the NN can utilize to infer both interactions and causality.

Pathway application

Given the results for KEGG we asked whether we can use the CNNC method to infer missing edges in current pathways. There have been several attempts to utilize expression and other data to further refine known pathways and many of these are based on co-expression analysis[9, 19-21, 32, 33, 38, 39]. Since our method provides both direction and score we can extract all predicted directed edges above a certain score and compare the resulting pathway to the database pathway to see if any additional edges, that do not appear in the database, are predicted by our method. For this we focused on the interleukin 17 (IL-17) pathway from KEGG database, which plays crucial roles in inflammatory responses. We extracted 6 proteins and 4 directed edges from this pathway by only using directed edges with activation or inhabitation edge types and filtering out cyclic gene pairs (**Fig. 5a**). We applied CNNC trained on all database pathway edges that do not contain any of these 6 proteins. As can be seen in **Supplementary Fig. 7**, CNNC predicted 8 of the possible 30 edges for this path (15 pairs with two possible directions each). 4 of these 8 were the original 4 directed edges annotated in the database itself. The other 4 edges were not present in the KEGG as causal interactions for this pathway. However, all are either supported by their presence in other KEGG pathways or by recent publications (we reiterate that interactions for all six proteins in other KEGG pathways were excluded from the training data as mentioned above so these predictions are not contaminated by their

presence in other pathways in KEGG). Specifically, (traf6, nfkb1) and (map3k7, nfkb1) have been annotated as causal pairs in KEGG's 'Pertussis pathway' and 'RIG-I-like receptor signaling pathway'[36], respectively. (rela, nfkb1) is the known p50/p65 heterodimer of NF-κB[40]. As for the (nfkbia, traf6) pair, it was found that traf6's binding to MAP3K7 activates ikbkb which in turn phosphorylates nfkbia[41].

CNNC output as a similarity matrix for clustering

To evaluate CNNC's performance on the downstream analysis, we used it to generate a gene-gene similarity matrix. We next used this matrix as an input for a hierarchical clustering algorithm.

We extracted the top 2,000 (top 1,000, see **Supplementary Fig. 8**) differentially expressed genes based on the expression data used in this paper using fano factor (FF) (**Supplementary Note**). Since we have trained CNNC using the KEGG database, we removed KEGG genes from the test set. Next we performed hierarchical clustering[42] using CNNC and PC based on all sc and bulk data (**Fig. 5b**, and **5c**). For comparison, we selected the top 8 clusters for the resulting hierarchical clustering tree for all inputs (see also **Supplementary Fig. 8**). Next, for each input we calculated the significant GO terms (q-values < 0.05)[43] and plotted the results in **Fig. 5d**. As can be seen, using CNNC as the input led to the identification of more significant GO terms for the same set of genes indicating that the clustering obtained using this input is more aligned with current biological knowledge.

Discussion and conclusion

Gene co-expression analysis has been a widely successful method for the analysis of gene expression data starting over two decades ago with the introduction of microarrays. Several methods have been suggested for this task and several other methods use co-expression analysis as a component in a more elaborate modeling framework.

While co-expression analysis performs well in some cases, it suffers from a number of drawbacks that often led to overfitting (false positives) or missing key relationships (false negatives). The former can be attributed to the unsupervised nature of most co-expression methods making it hard to 'train' them on a labeled dataset. The latter often resulted from the nature of the data used for co-expression analysis (bulk or population of cells data) which led to masking of relationships that existed in single cells. Moreover, while certain more sophisticated methods attempted to utilize gene expression to infer causality (for example, Bayesian network based methods [44]) these were only able to detect directed interactions, were based on very specific probabilistic modeling assumptions, and did not directly provide a confidence score for the resulting edges.

To address these issues we presented CNNC, a general framework for co-expression analysis which is based on convolutional NN (CNN). The key idea here is to convert the input data into a co-expression histogram which is very suitable for CNNs. Unlike most prior methods our method is supervised which allows the CNN to zoom in on subtle differences between positive and negative pairs. Supervision also helps fine tune the scoring function based on the different application. For example, different features may be important for analyzing TF-gene interactions when compared to inferring proteins in the same pathway. In addition to the supervised approach the fact that the network can utilize the large volumes of scRNA-Seq data allows it to better overcome masking issues reducing false negative.

Analysis of several different interaction prediction tasks indicates that CNNC can improve upon prior, unsupervised methods. It can also be naturally extended to

integrate complementary data including epigenetic and sequence information. Finally, CNNC is easy to use either with general data or with condition specific data. For the former, users can download the data and implementation from the supporting website (**Supplementary Fig. 9**), provide a list of labels (positive and negative pairs for their system of interest) and retrieve the scores for all possible gene pairs. These in turn can be used for any downstream application including clustering, network analysis etc.

In addition to comparing CNNC to prior methods we have also used it to evaluate the advantages conferred by scRNA-Seq data. Models trained with scRNA-Seq data outperformed those trained with bulk data for all systems we looked at. This supports prior findings[45, 46] and addressed a key criticism of co-expression analysis – that many interactions are observed or missed due to aggregation effects from the collection of cells rather than because they truly represent specific molecular events. While the scRNA-Seq data we used contained two orders of magnitude more samples, the total number of cells profiled is smaller (each bulk experiment often profiles at least three orders of magnitude more cells than a single scRNA-Seq profile[47]). In addition, scRNA-Seq coverage is often two orders of magnitude less than bulk experiments so that total number of reads in the two datasets is not very different. Even when comparing with the same number of profiles for bulk and sc we find that CNNC performs better when using scRNA-Seq data. This result seems to indicate that despite the much greater noise associated with scRNA-Seq, such data can provide more accurate models for the same overall costs, coverage and sample size.

CNNC is implemented in Python and both data and an open source version of the software are available from the supporting website. We hope that this method would become a useful component in future co-expression studies.

Online methods

Dataset sources and pre-process pipelines

We used mouse scRNA-Seq dataset collected by Alavi et al[23]. The dataset consists of uniformly processed 43,261 expression profiles from over 500 different scRNA-Seq studies. For each profile, expression values are available for the same set of 20,463 genes. Mouse bulk RNA-Seq dataset were downloaded from Mouse Encode project [24]. That data included 249 samples and we only utilized genes that are present in the scRNA-Seq dataset leading to the same number of genes for both datasets. mESC Dnase data was also downloaded from Mouse Encode project[24] (ENCFF096WRW.bed). Mouse TF motif information is from TRANSFAC database[48]. PWM values were calculated by Python package 'Biopython'[49].

For the DNase and PWM analysis we followed prior papers and defined the transcription start site (TSS) region as 10KB upstream to 1KB downstream from the TSS for each gene[27, 28]. For each TF and gene pair, using Biopython package we calculated the score between the TF motif sequence and both the '+/-' sequences at all possible positions along the TSS region of the gene, and then selected the maximum one as the final PWM score. The maximum Dnase peak signal in the TSS region was calculated as the scalar Dnase value for each gene.

Labeled data:

mESC ChIP-seq peak region data was downloaded from GTRD database, and we used peaks with threshold p value < 10^{-300}. If one TF X has at least one ChIP-seq peak signal in or partially in the TSS region of gene Y, as defined above, we say that X regulates Y.

KEGG and Reactome pathway data were downloaded by the R package 'graphite'[50]. KEGG contains 290 pathways and Reactome contains 1581 pathways. For both, we only select directed edges with either activation or inhabitation edge types and filter out cyclic gene pairs where genes regulate each other mutually (to allow for a unique label for each pair). In total, we have 3,057 proteins with outgoing directed edges in

KEGG and the total number of directed edges is 33,127. For Reacotome the corresponding numbers are 2,519 and 33,641.

Constructing the input histogram

For any gene pair *a* and *b*, we first log transformed their expression, and then uniformly divided the expression range of each gene to 32 bins. Next we created the 32X32 histogram by assigning each sample to an entry in the matrix and counting the number of samples for each entry. Due to the very low expression levels and even more so to dropouts in scRNA data, the value in zero-zero position is always very large and often dominates the entire matrix. To overcome this, we added pseudocounts to all entries. We combined bulk and scRNA-Seq NEPDFs by concatenating them as a 32X64 matrix to achieve better performance.

CNN for RPKM data

We followed VGGnet[51] to build our convolutional neural networks (CNN) model (**Supplementary Fig. 1**). The CNN consists of stacked layers of x 3X3 convolutional filters (equation (1)) (x is a power of 2, ranging from 32 to 64 to 128) and interleaved layers of 2X2 maxpooling (equation (2)). We used the constructed input data as input to CNN. Each convolution layer computes the following function:

$$\text{Convolution } (X)_{i,j}^{k} = \sum_{m=1}^{3} \sum_{n=1}^{3} W_{i,j}^{k} X_{i+m,j+n} \quad (1)$$

Where *X* is the input from the previous layer, (*i,j*) is output position, *k* is convolutional filter index and *W* is the filter matrix of size 3X3. In other words, each convolutional layer computes a weighted average of the prior layer values where the weights are determined based on training. The maxpooling layer computes the following function:

$$\text{maxpooling } (X)_{i,j}^{k} = \max\left(\left\{X_{i,j}^{k}, X_{i+1,j}^{k}, X_{i,j+1}^{k}, X_{i+1,j+1}^{k}\right\}\right) \quad (2)$$

Where *X* is input, (*i,j*) is output position and *k* is the convolutional filter index. In other words, the layer selects one of the values of the previous layer to move forward.

Overall structure

The overall structure of the CNN is presented in **Supplementary Fig. 1**. The input layer of the CNN is either 32X32 or 32X64 as discussed above. In addition, the CNN contains 10 intermediate layers and a single one or three-dimension output layer. The ten layers include both convolutional and maxpooling layers, and the exact dimensions of each layer are shown in **Supplementary Fig. 1**. Following ref 52 [52] we used rectified linear activation function (ReLU) as the activation function (equation (3)) across the whole network, except the final classification layers where 'sigmoid' function (equation (4)) was used for two categories classification and 'softmax' function (equation (5)) for multiple categories classification. These functions are defined below.

$$\mathrm{ReLU}\ (x) = \begin{cases} x & if\ x \geq 0 \\ 0 & if\ x < 0 \end{cases} \quad (3)$$

$$\mathrm{Sigmoid}_{\theta}\ (x) = {}^{1}/_{1 + e^{\theta x}} \quad (4)$$

$$\mathrm{Softmax}_{\theta}\ (x) = \frac{1}{\sum_{j=1}^{k} e^{\theta_j x}} \begin{bmatrix} e^{\theta_1 x} \\ e^{\theta_2 x} \\ \dots \\ e^{\theta_k x} \end{bmatrix} \quad (5)$$

Training and testing strategy

We evaluated the CNN using cross validation. In these, training and test datasets are strictly separated to avoid information leakage. See **Supplementary Note, Supplementary Fig. 10** and **Supplementary Table 1** for details. For the three labels (causality analysis) we did the following: for each gene, we generated (*a, b*) (label1) and (*b, a*)'s (label2) NEPDF matrices. For the 0 label we generated a (*a*, *N*) NEPDF matrices for GTRD where *N* was a random gene and *a* was the TF. 0 labels for KEGG and Reactome were generated from random (*M, N*) gene pairs. After training, we used *p*1(*a, b*) + *p*2(*a, b*) as the probability that *a* interacts *b*, *p*2(*a, b*) – *p*2(*b, a*) as the pseudo probability that *b* regulates *a*.

434 **Integrating expression, sequence and DNase data**

435 To integrate Dnase and PWM data with the processed RNA-Seq data, we first
436 computed the max value for a PWM scan and DNase accessibility for each promotor
437 region. We next generated a two-value vector from this data for each pair and
438 embedded it to a 512D vector using one fully connected layer containing 512 nodes.
439 Next these are concatenated with the expression processed data to form a 1024D
440 vector which serves as input to a fully connected 512-node plus 128-node layer neural
441 network classifier. See **Supplementary Fig. 1** for details. Early stopping strategy by
442 monitoring validation loss function is used to avoid overfitting.

443

444 **Selection of edges for the IL-17 pathway analysis**

445 We performed leave-one-pathway-out validation to evaluate CNNC' performance for
446 predicting edges for individual pathways. We selected a relatively small pathway
447 ('IL-17' from KEGG) to improve our ability to present it visually. We discuss more
448 general results for KEGG as well (**Fig. 4**). For this analysis we only selected directed
449 edges with either activation or inhabitation types and filtered out cyclic gene pairs
450 where genes regulate each other mutually to purify the edge types. In total, we had 6
451 nodes and 4 directed edges for the IL-17 pathway. Next, we trained CNNC with the
452 entire KEGG dataset excluding any interactions for the six 'IL-17' pathway proteins.

453

454 **Hierarchical clustering and GO term enrichment analysis**

455 We performed hierarchical clustering followed by GO term enrichment analysis to
456 evaluate CNNC' performance in downstream analysis. We selected the top 2,000 (or
457 1,000 (**Supplementary Fig. 8**)) genes with highest Fano factor (**Supplementary**
458 **Note**) We obtained the similarity matrices for the filtered gene list based on CNNC, sc
459 PC, bulk PC and sc&bulk PC respectively. We cut the tree at 8 clusters for all inputs.
460 Next, we performed GO term enrichment analysis using fisher's exact test and
461 counted the significant GO terms for each of the cluster result. Significance of
462 difference between different inputs was computed using one-side

Wilcoxon-Mann-Whitney test for the q-values of the four strategies (**Fig. 5d**).

Acknowledgements

Work partially supported by NIH grant 1R01GM122096, US National Science Foundation (DBI-1356505) to ZBJ. and a James S. McDonnell Foundation Scholars Award in Studying Complex Systems to Z.B.-J.

Author contributions

Y.Y. and Z.B conceived the method. Y.Y. implemented CNNC and the support website. Y.Y. and Z.B designed the experiments. Y.Y. and Z.B wrote the manuscript.

Competing interests

None

References

1. De Smet, R. & Marchal, K. Advantages and limitations of current network inference methods. *Nat Rev Microbiol* **8**, 717-729 (2010).
2. Stuart, J.M., Segal, E., Koller, D. & Kim, S.K. A gene-coexpression network for global discovery of conserved genetic modules. *Science* **302**, 249-255 (2003).
3. van Dam, S., Vosa, U., van der Graaf, A., Franke, L. & de Magalhaes, J.P. Gene co-expression analysis for functional classification and gene-disease predictions. *Brief Bioinform* (2017).
4. Butte, A.J. & Kohane, I.S. Unsupervised knowledge discovery in medical databases using relevance networks. *Proc AMIA Symp*, 711-715 (1999).
5. Eisen, M.B., Spellman, P.T., Brown, P.O. & Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A* **95**, 14863-14868 (1998).
6. Newman, M.E. Spectral methods for community detection and graph partitioning. *Phys Rev E Stat Nonlin Soft Matter Phys* **88**, 042822 (2013).
7. Langfelder, P. & Horvath, S. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics* **9**, 559 (2008).
8. Bartlett, T.E., Muller, S. & Diaz, A. Single-cell Co-expression Subnetwork Analysis. *Sci Rep* **7**, 15066 (2017).
9. Chan, T.E., Stumpf, M.P.H. & Babtie, A.C. Gene Regulatory Network Inference from Single-Cell Data Using Multivariate Information Measures. *Cell Syst* **5**, 251-267 e253

(2017).

10. Huynh-Thu, V.A., Irrthum, A., Wehenkel, L. & Geurts, P. Inferring regulatory networks from expression data using tree-based methods. *PLoS One* **5** (2010).

11. Faith, J.J. et al. Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biol* **5**, e8 (2007).

12. Lee, H.K., Hsu, A.K., Sajdak, J., Qin, J. & Pavlidis, P. Coexpression analysis of human genes across many microarray data sets. *Genome Res* **14**, 1085-1094 (2004).

13. Song, L., Langfelder, P. & Horvath, S. Comparison of co-expression measures: mutual information, correlation, and model based indices. *BMC Bioinformatics* **13**, 328 (2012).

14. Daub, C.O., Steuer, R., Selbig, J. & Kloska, S. Estimating mutual information using B-spline functions--an improved similarity measure for analysing gene expression data. *BMC Bioinformatics* **5**, 118 (2004).

15. Basso, K. et al. Reverse engineering of regulatory networks in human B cells. *Nat Genet* **37**, 382-390 (2005).

16. Kotlyar, M., Fuhrman, S., Ableson, A. & Somogyi, R. Spearman correlation identifies statistically significant gene expression clusters in spinal cord development and injury. *Neurochem Res* **27**, 1133-1140 (2002).

17. Qian, J., Dolled-Filhart, M., Lin, J., Yu, H. & Gerstein, M. Beyond synexpression relationships: local clustering of time-shifted and inverted gene expression profiles identifies new, biologically relevant interactions. *J Mol Biol* **314**, 1053-1066 (2001).

18. Reverter, A. & Chan, E.K. Combining partial correlation and an information theory

approach to the reversed engineering of gene co-expression networks. *Bioinformatics* **24**, 2491-2497 (2008).

19. Kumari, S. et al. Evaluation of gene association methods for coexpression network construction and biological knowledge discovery. *PLoS One* **7**, e50411 (2012).

20. Wang, Y.X., Waterman, M.S. & Huang, H. Gene coexpression measures in large heterogeneous samples using count statistics. *Proc Natl Acad Sci U S A* **111**, 16371-16376 (2014).

21. Freytag, S., Gagnon-Bartsch, J., Speed, T.P. & Bahlo, M. Systematic noise degrades gene co-expression signals but can be corrected. *BMC Bioinformatics* **16**, 309 (2015).

22. Crow, M., Paul, A., Ballouz, S., Huang, Z.J. & Gillis, J. Exploiting single-cell expression to characterize co-expression replicability. *Genome Biol* **17**, 101 (2016).

23. Amir Alavi, M.R., Aiyappa Parvangada, Zhilin Huang, Ziv Bar-Joseph scQuery: a web server for comparative analysis of single-cell RNA-seq data. (2018).

24. Yue, F. et al. A comparative encyclopedia of DNA elements in the mouse genome. *Nature* **515**, 355-364 (2014).

25. Johnson, D.S., Mortazavi, A., Myers, R.M. & Wold, B. Genome-wide mapping of in vivo protein-DNA interactions. *Science* **316**, 1497-1502 (2007).

26. Yevshin, I., Sharipov, R., Valeev, T., Kel, A. & Kolpakov, F. GTRD: a database of transcription factor binding sites identified by ChIP-seq experiments. *Nucleic Acids Res* **45**, D61-D67 (2017).

27. Schulz, M.H. et al. Reconstructing dynamic microRNA-regulated interaction networks. *Proc Natl Acad Sci U S A* **110**, 15686-15691 (2013).

559 28. Schulz, M.H. et al. DREM 2.0: Improved reconstruction of dynamic regulatory
560 networks from time-series expression data. *BMC Syst Biol* **6**, 104 (2012).
561 29. Wang, J. et al. Single-Cell Co-expression Analysis Reveals Distinct Functional
562 Modules, Co-regulation Mechanisms and Clinical Outcomes. *PLoS Comput Biol* **12**,
563 e1004892 (2016).
564 30. Khan, A. et al. JASPAR 2018: update of the open-access database of transcription
565 factor binding profiles and its web framework. *Nucleic Acids Res* **46**, D260-D266
566 (2018).
567 31. Gitter, A., Carmi, M., Barkai, N. & Bar-Joseph, Z. Linking the signaling cascades and
568 dynamic regulatory networks controlling stress responses. *Genome Res* **23**, 365-376
569 (2013).
570 32. van Noort, V., Snel, B. & Huynen, M.A. Predicting gene function by conserved
571 co-expression. *Trends Genet* **19**, 238-242 (2003).
572 33. Parikshak, N.N. et al. Integrative functional genomic analyses implicate specific
573 molecular pathways and circuits in autism. *Cell* **155**, 1008-1021 (2013).
574 34. Willsey, A.J. et al. Coexpression networks implicate human midfetal deep cortical
575 projection neurons in the pathogenesis of autism. *Cell* **155**, 997-1007 (2013).
576 35. Young, J.H. & Marcotte, E.M. Predictability of Genetic Interactions from Functional
577 Gene Modules. *G3 (Bethesda)* **7**, 617-624 (2017).
578 36. Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y. & Morishima, K. KEGG: new
579 perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res* **45**,
580 D353-D361 (2017).

581 37. Fabregat, A. et al. The Reactome Pathway Knowledgebase. *Nucleic Acids Res* **46**,
582 D649-D655 (2018).
583 38. Angelini, C. & Costa, V. Understanding gene regulatory mechanisms by integrating
584 ChIP-seq and RNA-seq data: statistical solutions to biological problems. *Front Cell*
585 *Dev Biol* **2**, 51 (2014).
586 39. D'Haeseleer, P., Liang, S. & Somogyi, R. Genetic network inference: from
587 co-expression clustering to reverse engineering. *Bioinformatics* **16**, 707-726 (2000).
588 40. Chen, F.E., Huang, D.B., Chen, Y.Q. & Ghosh, G. Crystal structure of p50/p65
589 heterodimer of transcription factor NF-kappaB bound to DNA. *Nature* **391**, 410-413
590 (1998).
591 41. Kramer, I. Signal Transduction. (Elsevier, 2015).
592 42. Bar-Joseph, Z., Gifford, D.K. & Jaakkola, T.S. Fast optimal leaf ordering for
593 hierarchical clustering. *Bioinformatics* **17 Suppl 1**, S22-29 (2001).
594 43. Ashburner, M. et al. Gene ontology: tool for the unification of biology. The Gene
595 Ontology Consortium. *Nat Genet* **25**, 25-29 (2000).
596 44. Spirtes, P. & Zhang, K. Causal discovery and inference: concepts and recent
597 methodological advances. *Appl Inform (Berl)* **3**, 3 (2016).
598 45. Wagner, A., Regev, A. & Yosef, N. Revealing the vectors of cellular identity with
599 single-cell genomics. *Nat Biotechnol* **34**, 1145-1160 (2016).
600 46. Wang, Y. & Navin, N.E. Advances and applications of single-cell sequencing
601 technologies. *Mol Cell* **58**, 598-609 (2015).
602 47. Ozsolak, F. & Milos, P.M. RNA sequencing: advances, challenges and opportunities.

Nat Rev Genet **12**, 87-98 (2011).

48. Wingender, E., Dietze, P., Karas, H. & Knuppel, R. TRANSFAC: a database on transcription factors and their DNA binding sites. *Nucleic Acids Res* **24**, 238-241 (1996).

49. Cock, P.J. et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**, 1422-1423 (2009).

50. Sales, G., Calura, E., Cavalieri, D. & Romualdi, C. graphite - a Bioconductor package to convert pathway topology to gene network. *BMC Bioinformatics* **13**, 20 (2012).

51. Karen Simonyan, A.Z. Very Deep Convolutional Networks for Large-Scale Image Recognition. (2014).

52. Xavier Glorot, A.B., Yoshua Bengio Deep Sparse Rectifier Neural Networks. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, PMLR*, 315-323 (2011).

625

626

627

628

629

630

631

632

633

634

635

636

637 **Figure legends**

638

639 **Figure 1 The CNNC architecture**

640 For each gene pair, expression levels from bulk and sc RNA-Seq are transformed into
641 two 32X32 normalized empirical probability function (NEPDF) matrices, and the two
642 are concatenated to form a combined 32X64 NEPDF matrix (left). The combined
643 NEPDF serves as an input to a convolutional neural network (CNN). The intermediate
644 layer of the CNN can be further concatenated with input vectors representing
645 Dnase-seq and PWM data (top). The output layer contains three probability nodes
646 where *p0* is the probability that genes *a* and *b* are not interacting, *p1* encodes the
647 case that gene *a* regulates gene *b*, and *p2* is the probability that gene *b* regulates
648 gene *a*.

649

Figure 2 GTRD TF-target prediction

(**a**) ROCs for Pearson Correlation (PC) based on scRNA-Seq. Light gray lines represent the performance for each TF. Black line represents the median ROC, and light green region represents the 25~75 quantile. (**b**) PC for bulk RNA-Seq. (**c**) PC for combined bulk and scRNA-Seq. (**d**) ROCs for Mutual Information (MI) when using scRNA-Seq. (**e**) MI when using bulk RNA-Seq. (**f**) MI using the combined bulk and scRNA-Seq. (**g**) ROCs for CNNC using scRNA-Seq. (**h**) CNNC using bulk RNA-Seq. (**i**) CNNC for combined data. Inset – Top ranking CNNC pairs are much more likely to be correct pairs when compared to other methods. (**j**) Comparison of TF-target predictions with additional data. Columns 1-3 show median AUROC of PC, MI, and CNNC using scRNA-Seq, bulk and the combine data, respectively. 4th column shows prediction of TF-gene interactions using only PWM or Dnase. 5th column shows performance when integrating expression, sequence and DNase data.

Figure 3 Predicting undirected pathway edges

(**a**) Overall ROCs for CNNC performance on KEGG pathway undirected edge prediction with bulk and scRNA-Seq. (**b**) The Area Under the Receiver Operating Characteristic curve (AUROC) histogram for (**a**). (**c**) The overall ROCs for performance of CNNC on Reactome pathway undirected edge prediction with bulk and scRNA-Seq. (**d**) The AUROC histogram for (**c**).

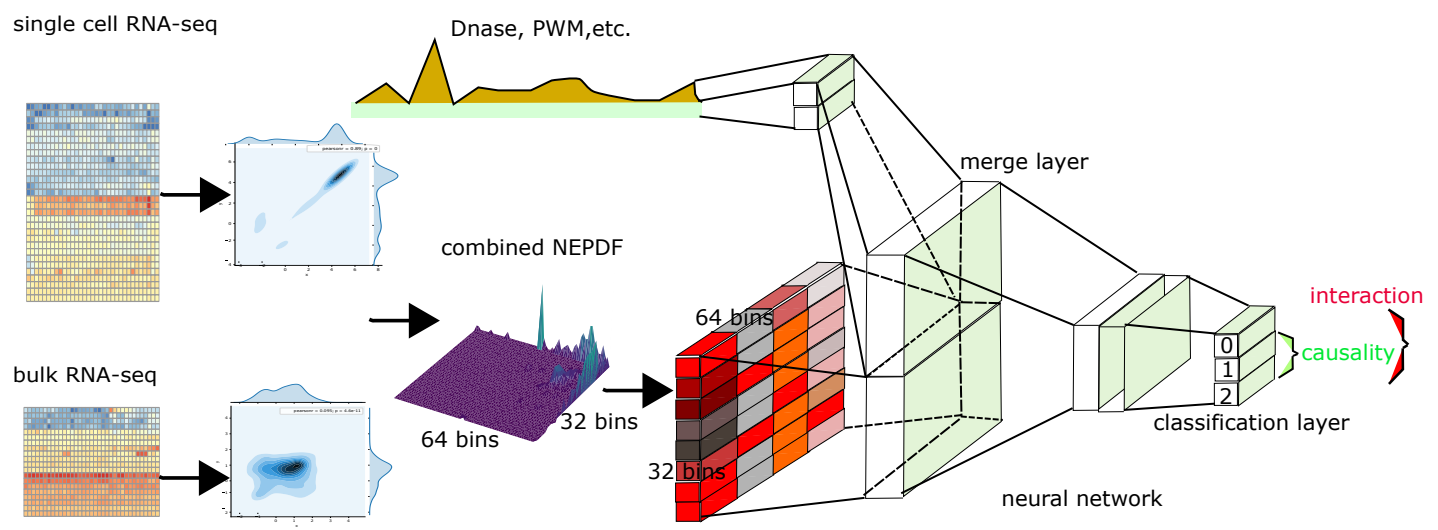
Figure 4 Directed (causal) edge prediction

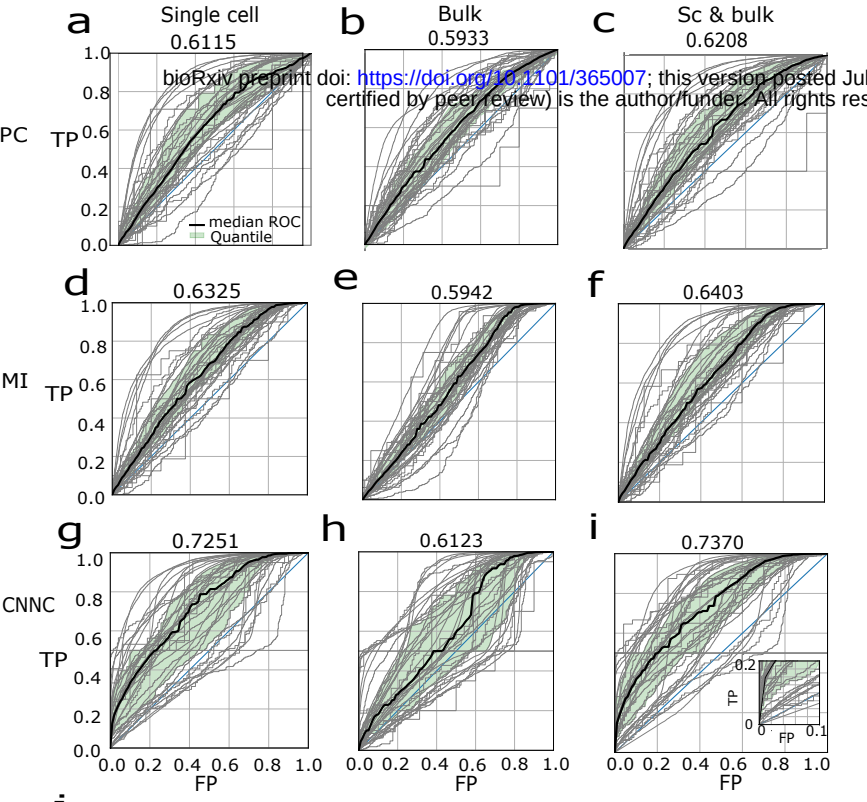
(**a**) Overall ROCs for performance of CNNC on GTRD directed prediction with bulk and scRNA-Seq. (**b**) The AUROC histogram for (**a**). (**c**) Overall ROCs for performance of CNNC on KEGG pathway directed edge prediction with bulk and scRNA-Seq. (**d**) The AUROC histogram for (**c**). (**e**) Overall ROCs for performance of CNNC on Reactome pathway directed edge prediction with bulk and scRNA-Seq. (**f**) The

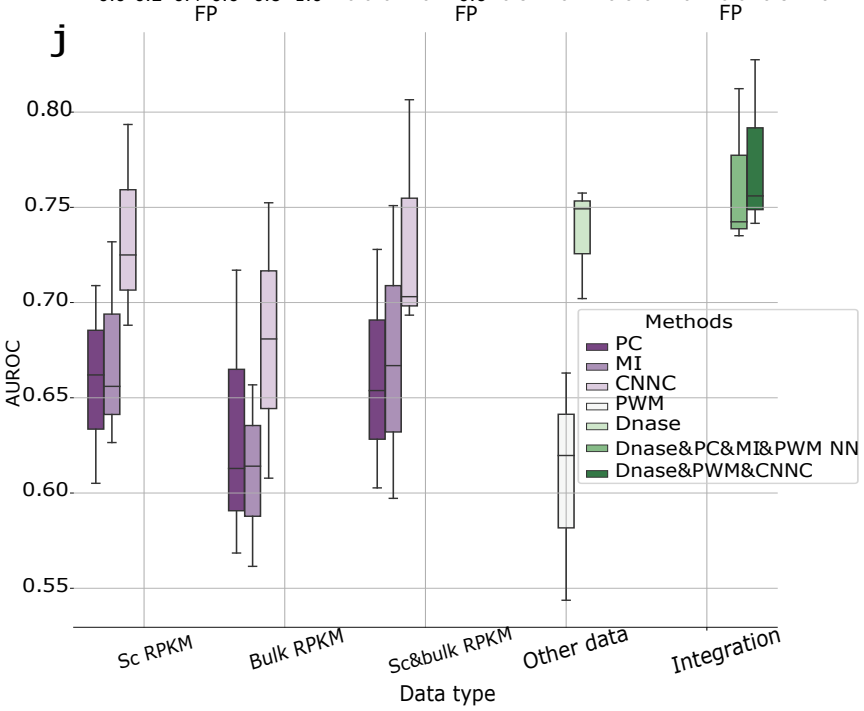
AUROC histogram for (**e**). (**g**) A typical NEPDF sample from a KEGG interaction that is correctly predicted as label 1. (**h**) A typical NEPDF sample that is correctly predicted as label 2.

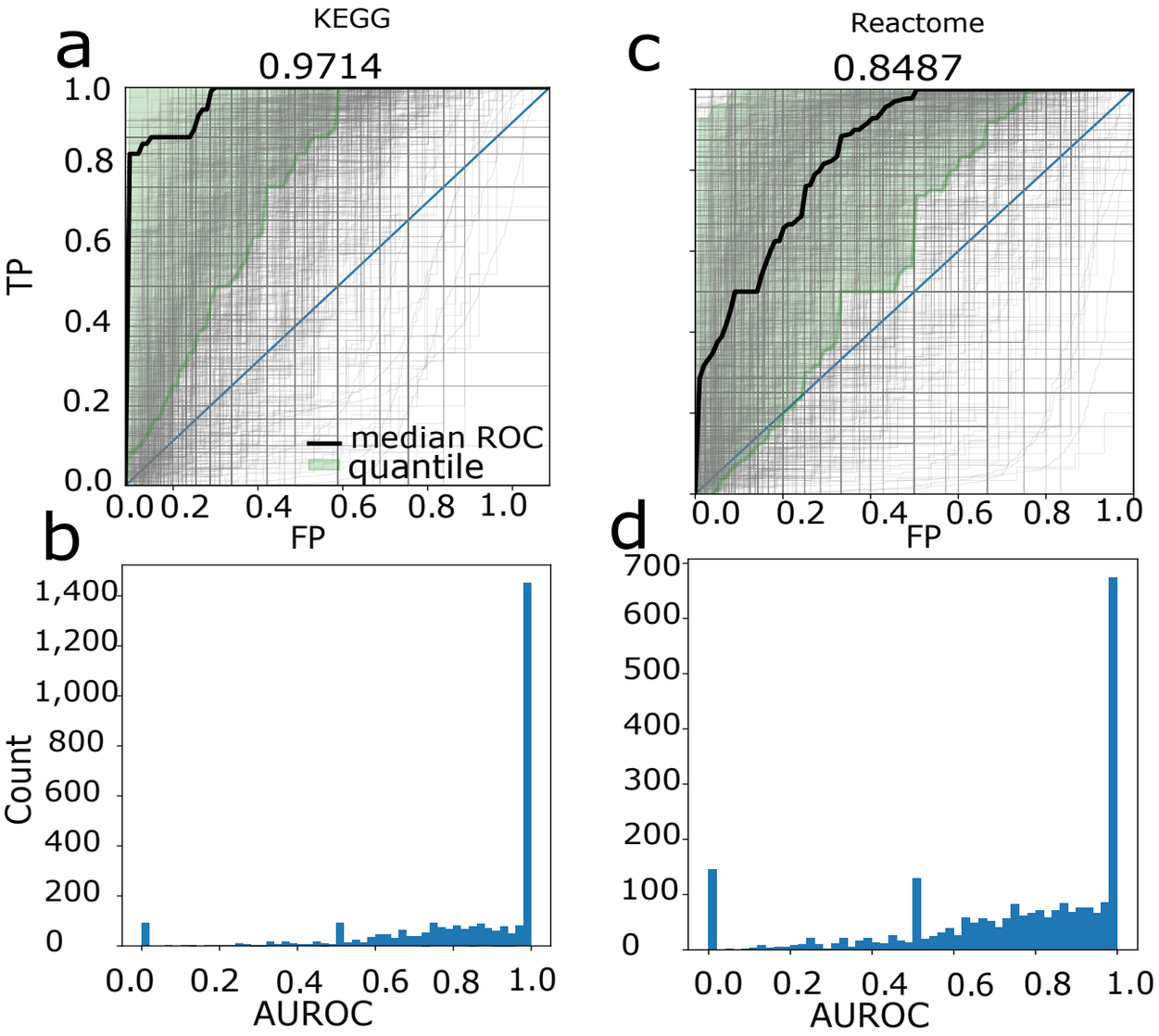
Figure 5 Downstream applications using CNNC

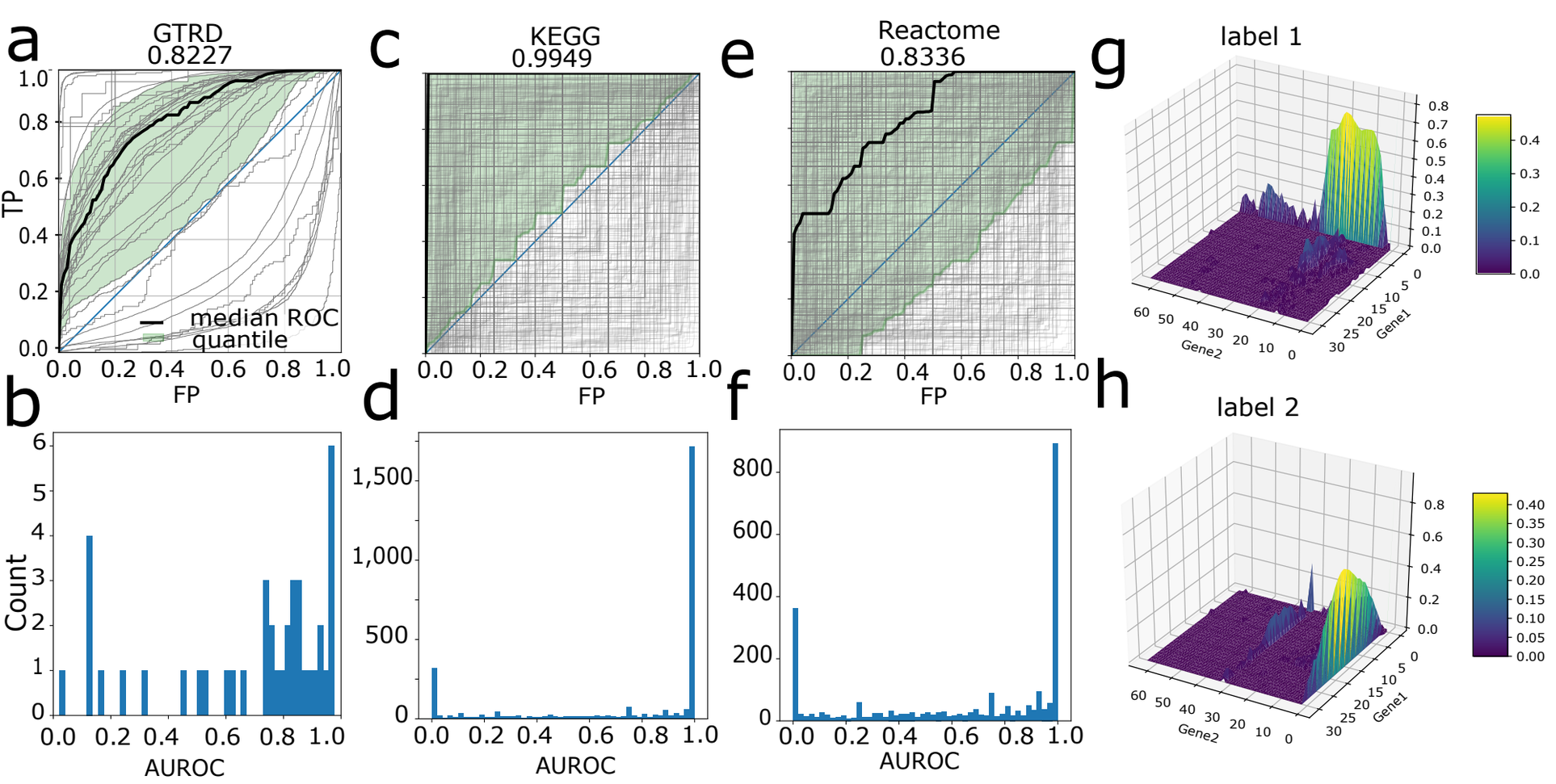
CNNC can be used as a component in downstream analysis algorithms including for pathway analysis and clustering. (**a**) Top: Directed edges annotated in KEGG for the IL-17 pathway Bottom: predicted directed edges for the pathway. (**b**) Hierarchical clustering of top 2,000 DE genes based on CNNC similarity matrix score. The number under the horizontal line represents the distance between the two groups, and the black horizontal line shows the resulting eight clusters groups. (**c**) Hierarchical clustering based using PC as the input. (**d**) GO term analysis of the clustering results from (**Figs. 5b, 5c, Supplementary Figs. 8a** and **8b**).


single cell RNA-seq
Dnase, PWM,etc.
merge layer
combined NEPDF
bulk RNA-seq
64 bins
32 bins
64 bins
32 bins
interaction
causality
0
1
2
classification layer
neural network






a
KEGG
0.9714
TP
FP
median ROC
quantile
b
Count
AUROC
c
Reactome
0.8487
d
AUROC


a
GTRD
0.8227
TP
FP
median ROC
quantile
c
KEGG
0.9949
FP
e
Reactome
0.8336
FP
g
label 1
Gene1
Gene2
b
Count
AUROC
d
AUROC
f
AUROC
h
label 2
Gene1
Gene2