Predicting cell population-specific gene expression from genomic sequence

Lieke Michielsen[1,2,3], Marcel J.T. Reinders[1,2,3], Ahmed Mahfouz[1,2,3,*]

[1] Department of Human Genetics, Leiden University Medical Center, Einthovenweg 20, 2333ZC, Leiden, The Netherlands
[2] Leiden Computational Biology Center, Leiden University Medical Center, Einthovenweg 20, 2333ZC, Leiden, The Netherlands
[3] Delft Bioinformatics Lab, Delft University of Technology, Van Mourik Broekmanweg 6, 2628XE, Delft, The Netherlands

* Correspondence to: a.mahfouz@lumc.nl

Abstract

Most regulatory elements, especially enhancer sequences, are cell population-specific. One could even argue that a distinct set of regulatory elements is what defines a cell population. However, discovering which non-coding regions of the DNA are essential in which context, and as a consequence which genes are expressed, is a difficult task. Some computational models tackle this problem by predicting gene expression directly from the genomic sequence. These models are currently limited to predicting bulk measurements and can mainly make tissue-specific predictions. Here, we present the first model that leverages single-cell RNA-sequencing data to predict gene expression at an unprecedented resolution. We show that cell population-specific models outperform tissue-specific models especially when the expression profile of a cell population and the corresponding tissue are dissimilar. Further, we show that our model can prioritize GWAS variants and learn motifs of transcription factor binding sites. We envision that our model can be useful for delineating cell population-specific regulatory elements.

Introduction

In multicellular organisms, every cell has the same DNA apart from somatic mutations. Yet its function and the related proteins and genes expressed vary enormously. This is among others caused by transcriptional and epigenetic regulation. Proteins that are binding the DNA sequence around the transcription start site (TSS) control whether a gene is transcribed in a cell [1,2]. Which transcription factors, and thus which DNA motifs, are essential differ per cell population [1–4]. As such, mutations in regulatory regions might affect specific tissues or cell populations differently. Improving our understanding of these regulatory mechanisms will help us relate genomic functions to a phenotype.

For example, while promoter sequences are identical across the four major human brain cell populations (neurons, oligodendrocytes, astrocytes, and microglia), almost all enhancer sequences, the regions in the DNA where a transcription factor binds, are cell population-specific [3]. These population-specific regulatory elements are discovered by combining single-cell measurements of different data types, including chromatin accessibility, ChIP-seq, and DNA methylation. Bakken et al., for instance, identified differentially methylated and differentially accessible regions across neuronal cell populations in the human brain [5]. These two types of regions have been identified using two types of measurements, however, they almost show no overlap. This emphasizes the complexity of transcriptional regulation and the need for more measurements to fully resolve these mechanisms at the cell population-specific level.

An alternative approach would be to train a computational model that directly predicts gene expression from the genomic sequence around the TSS. This way, we can learn which regulatory sequences are important for transcriptional regulation in different contexts. Several computational methods have been developed for this task [6–12]. These methods have in common that they one-hot encode the DNA sequence and input this to either a convolutional neural network (CNN) or transformer. ExPecto, Xpresso, and ExpResNet predict expression measurements from bulk RNA-sequencing, while Basset, Basenji, BPNet, and the Enformer model predict regulatory signals, such as cap analysis gene expression (CAGE) reads or TF binding from CHIP-nexus.

A promising application of these models is to prioritize variants that have been identified using genome-wide association studies (GWAS) [6,13]. Using GWAS many potential disease-associating variants have been identified [14–16]. Within each locus, however, it is often challenging to pinpoint which variant is causal and which gene is affected by the variant.

These current methods, however, are all designed for predicting expression from bulk gene expression data. This means that they are either tissue-specific or could be applied to FACS-sorted cells [13]. Since transcriptional regulation is even more context-specific, the resolution of current methods is not sufficient, especially for heterogeneous tissues where single-cell RNA-sequencing has revealed hundreds of cell populations [5,17,18]. To increase the resolution, the models would ideally be trained on single-cell RNA-sequencing (scRNA-seq) data.

Here, we present a deep learning model that uses a CNN to learn cell population-specific expression in scRNA-seq data from genomic sequences. Since single-cell and bulk data have different characteristics and distributions, we explored whether this type of model is suitable for single-cell data. We show that (i) cell population-specific models outperform tissue-specific models on several tissues from the Tabula Muris, (ii) increasing the resolution improves the predictions for human brain cell populations, and (iii) *in-silico* mutagenesis of the input sequence can be used to prioritize GWAS variants.

Results

Predicting cell population-specific gene expression using scEP

Here, we present scEP (single-cell Expression Prediction), a multitask convolutional neural network (CNN) to predict cell population-specific gene expression using genomic sequences only (Figure 1A, S1). We developed scEP by adapting the Xpresso model [9], which was originally designed for bulk data, to single-cell data. Similar to Xpresso, we use two types of input to the model: (1) the DNA sequence around the transcription start site (TSS) (7kb upstream - 3.5kb downstream) to model transcription and (2) five half-life time features (5' UTR length, 3' UTR length, ORF length, intron length, and exon junction density) to model mRNA degradation. We input the one-hot encoded DNA sequence into a CNN. The output of the CNN is concatenated with the half-life time features and fed to a fully connected network (see **Methods**). Since our model is a multitask CNN, the desired output of the fully connected network is the gene expression for every cell population. To obtain one expression value per cell population, we aggregate the single-cell expression into pseudobulk measurements (see **Methods**).

Since single-cell and bulk data have different characteristics, we tested whether scEP performs equally well on single-cell and bulk data. We used scRNA-seq data from five different tissues (limb muscle, spleen, gland, marrow, and lung) from the Tabula Muris [13] (Table S1). Here, we used the FACS-sorted cells that were sequenced using the Smart-seq2 protocol. Using the annotations defined by the authors, we aggregate the values per cell population and per tissue into pseudobulk values. For four tissues (limb muscle, spleen, marrow, and lung), there are also bulk RNA-sequencing datasets available (Table S2). We compared the pseudobulk to the bulk expression per tissue and noticed that these are indeed correlated (r_{muscle} = 0.76, r_{spleen} = 0.80, r_{marrow} = 0.68, r_{lung} = 0.76) (Figure S2).

Next, we trained three different models: 1) a tissue-specific model on the bulk values ($scEP_{t,b}$), 2) a tissue-specific model on the pseudobulk values ($scEP_{t,pb}$), 3) a cell population-specific model on the pseudobulk values ($scEP_{cp,pb}$) (Figure 1B). The cell population-specific model is, in contrast to the tissue-specific models, a multitask model that predicts the expression of all cell populations in a tissue simultaneously. We evaluated the performance of the models by calculating the Pearson correlation between the true and predicted expression values. In general, the tissue-specific models trained on pseudobulk reach higher performance than the models trained on bulk (Figure 1C-D). Even though the bulk and pseudobulk values are correlated, the pseudobulk distributions are bimodal compared to the normally distributed bulk data (Figure S2-3). This turns the problem more into a classification problem (is a gene low or high expressed), which might be easier to learn. On average, predicting cell population-specific expression is more difficult than predicting tissue-specific expression: $scEP_{cp,pb}$ performs slightly worse than $scEP_{t,pb}$ (median correlation of 0.72 vs 0.76), but still better than $scEP_{t,b}$ (0.67).

The Tabula Muris scRNA-seq datasets were generated using two different protocols: 10X Genomics, a droplet-based method, and FACS-sorted Smart-seq2, a plate-based method. When comparing $scEP_{t,pb}$ and $scEP_{cp,pb}$ trained on the two different protocols, e.g. lung-droplet vs. lung-FACS, we conclude that they perform equally well (Figure 1DE, S4-5). Depending on the tissue and cell population, one performs slightly higher than the other, but there is no significant difference. This is as expected since the pseudobulk values of both protocols are highly correlated (Pearson correlation > 0.9) (Figure S6). Hence, the protocol used to create the single-cell dataset thus does not influence the results.

For $scEP_{cp,pb}$, we tested how the two types of input features, DNA sequence and half-life time, influence the performance. We tested different lengths of the input sequence and whether one of the two features was enough to predict expression (Figure S7). A range of different sequence lengths results in the same performance (3.5-3.5, 7-3.5, and 10-5kb upstream-downstream). A longer sequence gives more information, but also adds more noise. Since the model also becomes more complex, more parameters have to be learned and it takes more time and memory to train the model. Therefore, we decided to use 7kb upstream and 3.5kb downstream for further experiments. We also observed that adding the half-life time features results in higher performance, suggesting that these features are not easily captured from DNA sequences alone.

For the cell population-specific models, the performance varies considerably across different populations (Figure 1E). Comparing the populations in the lung dataset, for instance, the performance of the endothelial cells is very high compared to leukocytes (Figure 1F, S8). In general, the performance of scEP increases when more genes and cells are measured in a population (Fig 1F, S9). The leukocyte population is small (35 cells) and fewer genes are non-zero compared to other cell populations in the lung (8,678 out of 20,467 vs. 12,715 on average). The ciliated cell population, on the other hand, is also small (25 cells), but this model reaches a higher performance. In this cell population, however, more genes were non-zero (11,717) compared to the leukocyte population. Hence, to train the model, we need a good representation of the cell population that includes enough expressed genes.

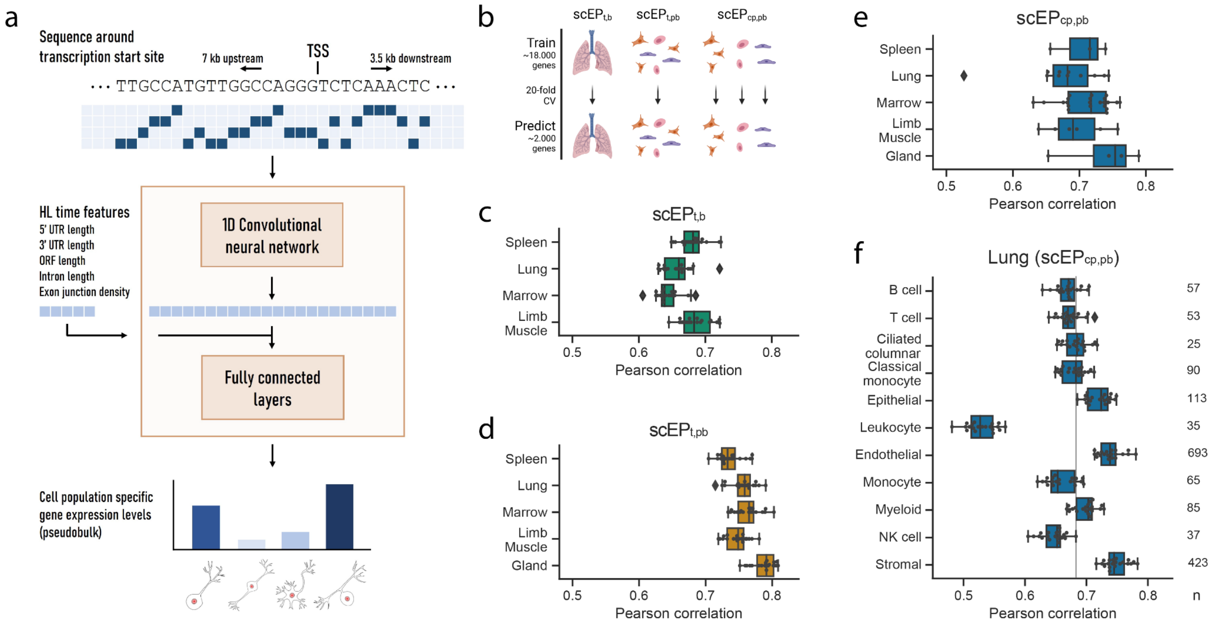

Figure 1. Schematic overview of scEP and performance on TM datasets. a) We one-hot encode the DNA sequence around the transcription start site (TSS) and input this to a one-dimensional convolutional neural network (CNN). The output of the CNN is flattened and concatenated with the five half-life time features. The fully connected layers output the cell population's specific gene expression levels (Figure S1, see **Methods**). **b)** Schematic overview of the experiment. **c-d)** Performance of scEP$_{t,b}$ and scEP$_{t,pb}$ respectively. Every dot is the performance (Pearson correlation) across one fold of the 20-fold CV. **e)** Performance of scEP$_{cp,pb}$ summarized per tissue. Every dot represents the model's performance on a cell population in that tissue (median Pearson correlation across the 20 folds). **f)** Performance of scEP$_{cp,pb}$ on the lung. The grey line indicates the median performance across all cell populations. Every dot is the performance across one fold of the 20-fold CV.

Cell population-specific models outperform tissue-specific models

Now that we know that all models are well-trained, we predicted cell population-specific expression using the three different models to see whether increasing the resolution of the models increases the performance (Figure 2A). Since scEP$_{t,b}$ and scEP$_{t,pb}$ were trained using tissue-specific expression values, these models predict the same value for every cell population. On all datasets, scEP$_{cp,pb}$ outperforms the tissue-specific models, which shows the benefit of training the models on a higher resolution (Figure 2B, S10A). Especially in more heterogeneous tissues, where the gene expression of cell populations is weakly correlated to the corresponding tissue, we see a large improvement (Figure 2C, S10B). For the lung-FACS dataset, for instance, the performance increases the most for immune cell populations ($\Delta_{cp,t}$ for B cells: 0.11, NK cells: 0.10, T cells: 0.09 (see **Methods**)) and the least for lung-specific populations (stromal cells: 0.01, endothelial cells: 0.02, epithelial cells: 0.04). For the B cells in the lung, the tissue-specific model predicts a too high value for 3,387 genes (absolute error > 4), while all these genes are not expressed in the B cells in the lung (Figure 2D). The B cell-model predicts a lower value for these genes (Figure 2E). Almost all these genes, however, are expressed in the lung dataset, so the lung-model learned this correctly (Figure 2F).

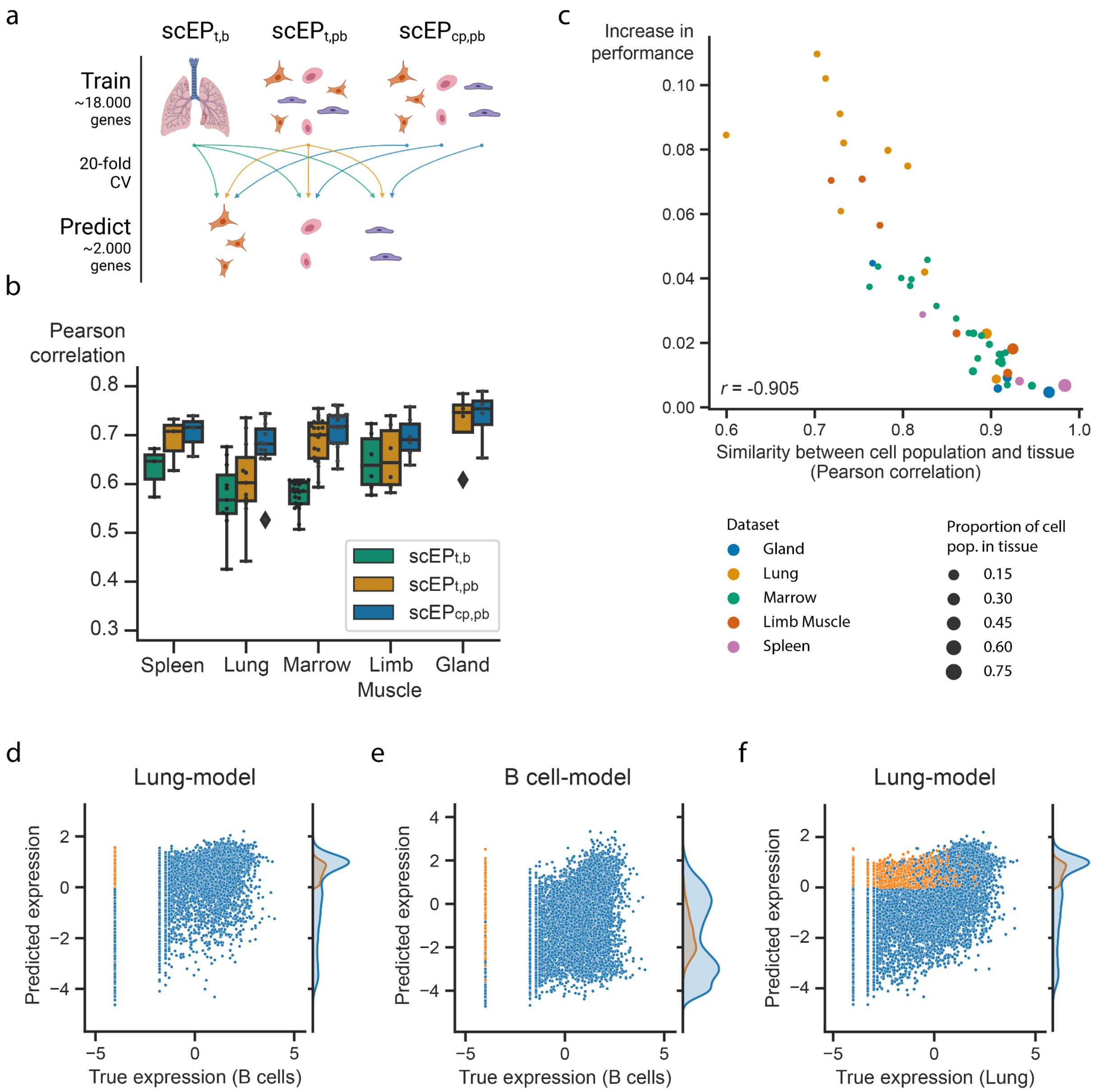

Figure 2. Comparison of the three scEP models for making cell population-specific predictions. a) Schematic overview of the experiment. **b)** Boxplot showing the models' performance on the cell population-specific task. Every point in the boxplot is the performance of a model on one cell population in that tissue (median Pearson correlation across the 20 folds). **c)** Similarity between a cell population and corresponding tissue (Pearson correlation between the true pseudobulk expression values) vs. the increase in performance (median Pearson correlation of $\text{scEP}_{\text{cp,pb}}$ - $\text{scEP}_{\text{t,pb}}$). **d-f)** Comparing the predictions made by the lung-model and the B cell-model. Genes where the lung-model predicts a too high value are plotted in orange. **d-e)** True expression of the B cells vs. predicted expression by the **d)** lung-model and **e)** B cell-model. **f)** True expression of the lung cells vs. predicted expression of the lung model.

Some of the Tabula Muris datasets contain similar cell populations. For instance, B cells, macrophages, and T cells are measured in four, three, and three tissues, respectively. We hypothesized that if our models are cell population-specific, they should accurately predict the expression of a cell population in one tissue with a model trained on the same cell population but from another tissue. A cell's tissue will of course slightly change the expression for (some) genes, but we will ignore these differences for now. Therefore, we predicted the expression for common cell populations using three different types of models: 1) $scEP_{cp,pb}$ trained on the same cell population, but from a different tissue, 2) $scEP_{cp,pb}$ trained on a different cell population, but from the same tissue, 3) $scEP_{t,pb}$ trained on the same tissue (Figure 3A). Again, the cell population-specific models outperform the tissue-specific models, even though they are predicting either a different dataset or a different cell population than they were trained on (Figure 3B, S11-12). This indicates that if you want to train a model for a cell population from a specific tissue where no single-cell data is available, you can better use a model trained on a similar cell population from a different tissue than relying on a tissue-specific model. Whether the model trained on a different cell population and the same dataset or vice versa performs better, differs per dataset and cell population. When predicting the expression of B cells in the limb muscle, the models trained on B cells in the marrow and lung even outperform the model trained on B cells in the limb muscle itself (Figure 3C). The models trained on different cell populations within the limb muscle perform variably when predicting B cells (Figure 3D). The models trained on immune populations, e.g. T cells or macrophages, perform similarly, but the muscle-specific populations perform worse. This difference between the B cell and the endothelial, mesenchymal stem cell, and skeletal muscle satellite cell models might seem small but is significant across the 20 folds (p-value = 9.5e-07 for all three populations, one-sided Wilcoxon rank sum test).

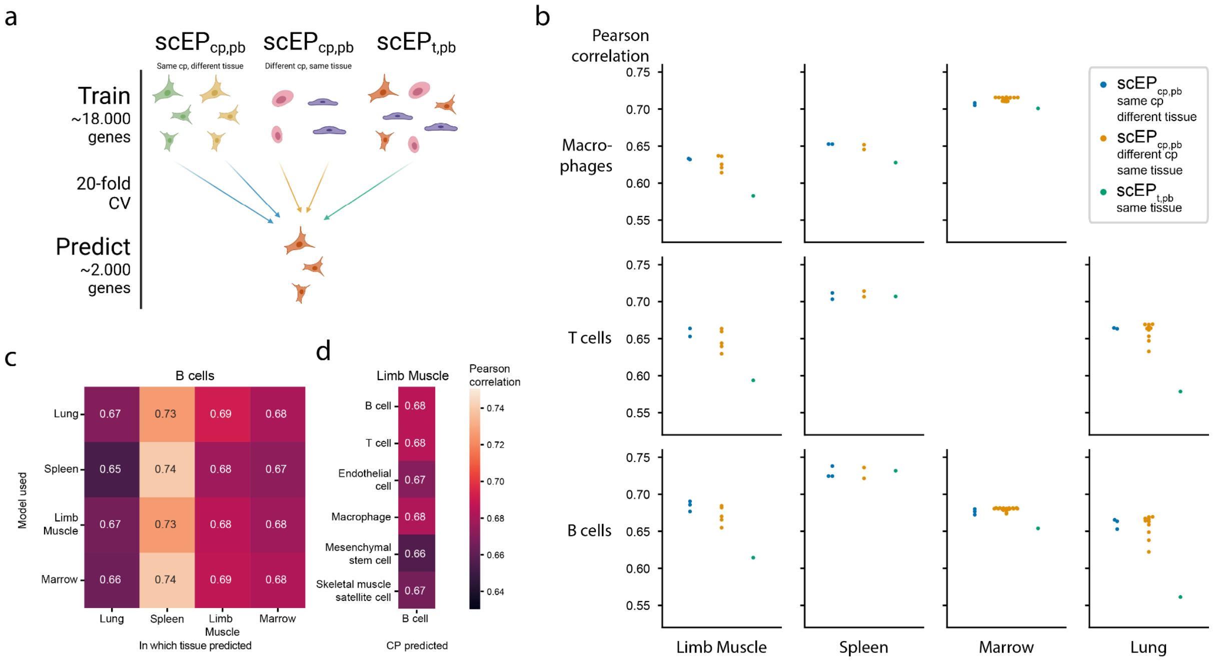

Figure 3. Predictions of scEP are cell population-specific. a) Schematic overview of the experiment. **b)** Performance (Pearson correlation) of three different types of models on different cell populations (rows) in different tissues (columns). Every dot is the median correlation of one model across the 20 folds. Since there are no T cells and macrophages defined in respectively the Marrow and Lung dataset, these boxes are missing. **c)** Pearson correlation of different models when predicting the expression of B cells in different tissues. The rows indicate on which tissue $scEP_{cp,pb}$ is trained, and the columns indicate for which tissue the expression of the B cells is predicted. **d)** Pearson correlation of different $scEP_{cp,pb}$ when predicting the expression of B cells in the limb muscle. Again the rows indicate which model is used.

scEP learns expression patterns across human brain cell populations

Next, we applied scEP to a human brain dataset of the motor cortex [5]. This dataset is annotated at different resolutions including a class (GABAergic, glutamatergic, and non-neuronal) and subclass (20 subclasses) level. Again, we trained models of different resolutions: a tissue-, class-, and subclass-specific model (scEP$_t$, scEP$_c$, and scEP$_{sc}$ respectively). We used the trained models to predict the subclass-specific expression values (Figure 4A). Since scEP$_t$ was trained on the tissue-specific pseudobulk expression, it predicts the same expression for all subclasses. The class-specific model, on the contrary, is a multitask model. Here, we use the predictions of the parent class of each subclass (e.g. the non-neuronal predictions for astrocytes) (Figure S13). Similar to the Tabula Muris, we can conclude that increasing the resolution increases the performance: scEP$_c$ outperforms scEP$_t$, and scEP$_{sc}$ outperforms scEP$_c$ (Figure 4B). For some subclasses, e.g. L2/3 IT, the performance barely improves when comparing scEP$_c$ and scEP$_{sc}$. Similar to the Tabula Muris, the true expression values of the subclass and corresponding class of such cases are strongly correlated (Figure S14).

Another advantage of increasing the resolution is that we can test whether scEP$_{sc}$ learns the correct pattern for a gene across the subclasses. For every gene, we calculate the Pearson correlation between the true and prediction expression across the subclasses. If the expression of a gene shows some variance across the subclasses, scEP$_{sc}$ learns the pattern correctly (Figure 4C). Genes that are variable across the subclasses are most interesting to study. For these genes, it is most important that we predict the pattern correctly. An example is *CACNA1I*, a gene coding for a subtype of voltage-gated calcium channel which has been associated with schizophrenia [15,19–22]. Here scEP$_{sc}$ correctly learns that the expression in neuronal populations is higher than in non-neuronal (r = 0.91) (Figure 4D).

In-silico mutagenesis reveals the most interesting GWAS variants

Since scEP can predict expression from the DNA sequence, we expect that it can also predict how the expression changes when the sequence mutates. Therefore, we applied *in-silico* mutagenesis (ISM) to the sequence of *CACNA1I* and evaluated the predicted change in gene expression. When comparing all possible mutations for the Sst Chodl subclass, scEP$_{sc}$ predicts that mutations in the region around the TSS affect expression the most (Figure 4E). We did not input this location into the model, so the model learned correctly that this is the most important region for transcriptional regulation. No other regions were found that affect the expression that strongly.

Besides visualizing the mutation pattern for one subclass, we can also visualize how ISM affects two subclasses differently. Here, we compared Sst Chodl and L2/3 IT (Figure S15). The Sst Chodl subclass is more sensitive to mutations than the L2/3 IT class for *CACNA1I*, which might be explained by the fact that *CACNA1I* is also higher expressed in Sst Chodl cells.

ISM can, for instance, be used to prioritize variants of interest for diseases. As an example, we focused on *CACNA1I*, which is linked to 18 Schizophrenia-associated variants according to the NHGRI-EBI Catalog [23]. Two of these variants, rs7288455 and rs5757730, fall within our input region (7kb upstream and 3.5kb downstream of the *CACNA1I* TSS). Mutating the reference A allele with the C or G variant at the position of rs7288455 increases the predicted expression for all cell populations (Figure 4F). The disease-associated variant, the A allele, is expected to decrease the expression [15,23], which is in line with our predictions. In general, the increase in expression is not related to the class a subclass belongs to. Our model suggests that the expression of *CACNA1I* increases the most in Sst Chodl cells. For the Sst Chodl subclass, this is a big increase compared to all other induced mutations (top 0.5% mutations with the strongest effect) (Figure S16). For the other variant, rs5757730, which lies in an intronic region, we see no difference in expression (Figure S17).

Supporting our predictions, rs7288455, but not rs5757730, overlaps with an ENCODE candidate *cis*-regulatory element. These results show that scEP can be used to prioritize GWAS hits.

In total, there are 3,971 GWAS variants associated with Schizophrenia [23]. Here, we focused on the genes that have two or more variants in the input region (20 genes, 49 variants) (Table S3). For these variants, we predicted the effect of all possible substitutions to prioritize the variants similarly to *CACNA1I* (Figure S18). For most genes, scEP predicts a profound effect for only one of the variants. Considering *HLA-B* rs2507989, for instance, substituting ‘A’ with ‘C’ decreases the expression, while none of the mutations at rs139099016 and rs1131275 are predicted to affect the expression. For some genes, however, all variants seem to barely affect the expression.

Next, we checked if we can interpret the model predictions by characterizing the genomic sequences identified by scEP to have a strong effect on gene expression. For *MROH-6* rs10866912, two substitutions are predicted to create an opposite effect. Substituting the reference ‘T’ with a ‘C’ is predicted to decrease the expression while mutating with a ‘G’ is predicted to increase the expression (Figure 4G). This variant is part of a binding site for the transcription factor *INSM1*, a transcriptional repressor [24] (Figure 4H). Substituting the ‘T’ with a ‘C’, the sequence of the reference genome becomes more similar to the consensus motif, while substituting with a ‘G’ makes the two sequences more dissimilar. This supports the predictions from scEP.

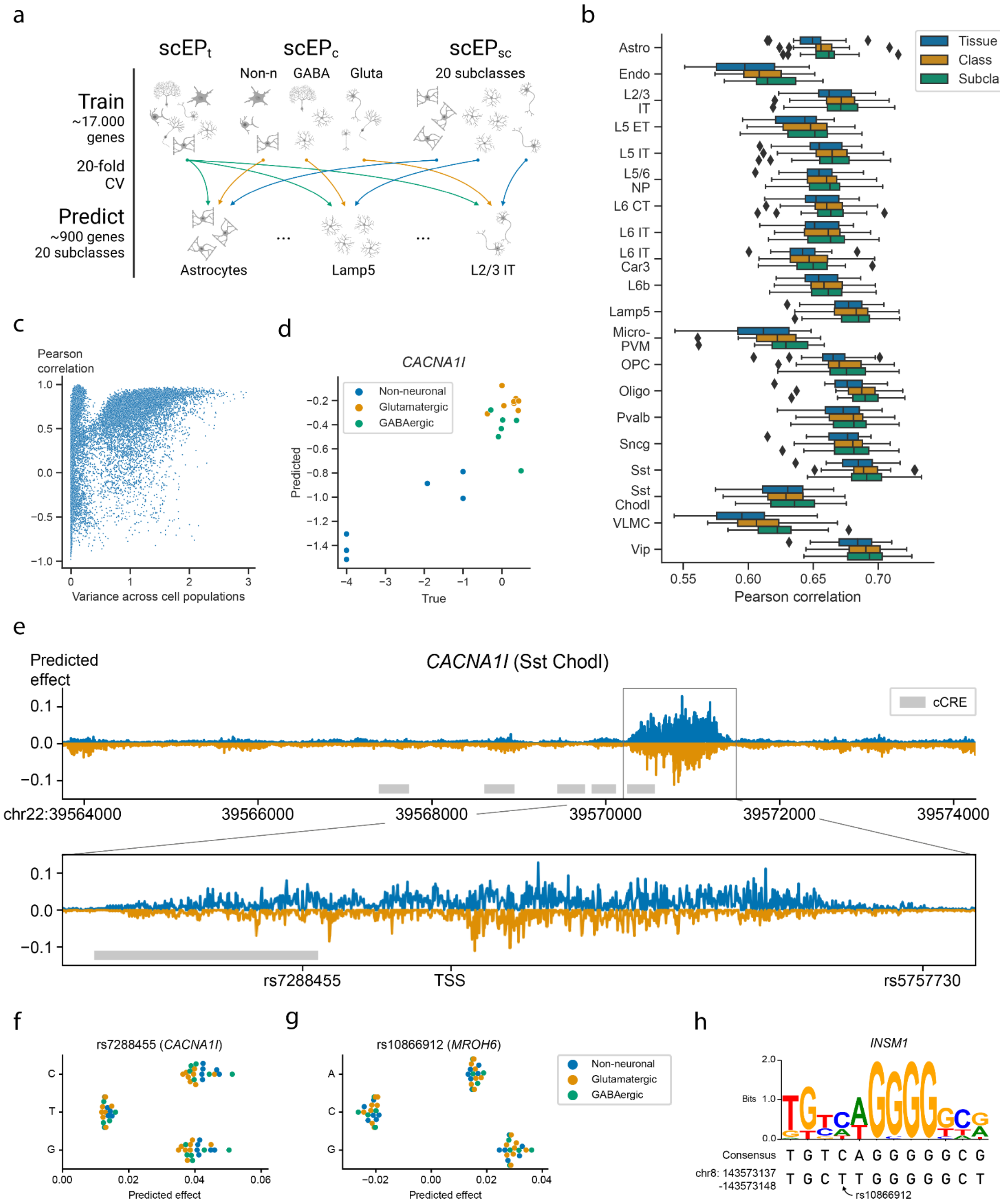

Figure 4. Performance of scEP on the human motor cortex. a) Schematic overview of the experiment. We train a tissue-, class-, and subclass-specific model (scEP$_t$, scEP$_c$, scEP$_{sc}$ respectively) to predict the subclass-specific expression levels. **b)** Boxplots showing the Pearson correlation between the true and predicted values. Every point in the boxplot is the performance on a fold (n=20). **c)** Scatterplot showing the relation between the variance of a gene across the pseudobulk values of the subclasses and the Pearson correlation between the true and predicted values across the subclasses. Every dot is a gene. **d)** True and predicted expression for *CACNA1I.* Every dot is the expression in a subclass. Dots are colored according to their class. **e)** Mutation profile for *CACNA1I* for the Sst Chodl subclass. For every position, we calculated the difference in expression for all three possible substitutions and visualized the substitution with the highest absolute predicted effect. Mutations that are predicted to increase or decrease the expression are plotted in blue and orange respectively. The grey rectangles indicate the position of candidate *cis*-Regulatory Elements (cCREs) derived from ENCODE data [25]. **f-g)** Predicted effect of the three substitutions for **f)** rs7288455 on *CACNA1I* expression, and **g)** rs10866912 on *MROH6* expression. Every dot is one subclass and the dots are colored according to the class. **h)** Sequence logo and the consensus sequence for the *INSM1* transcription factor motif together with the sequence of the reference genome.

Discussion

We presented scEP, a model to predict cell population-specific gene expression using the genomic sequence only. To our knowledge, this is the first model that leverages single-cell data for this task, which allows us to predict gene expression at an unprecedented resolution. We showed that scEP outperforms tissue-specific bulk and pseudobulk models especially when the expression profile of a cell population is dissimilar to that of the corresponding tissue. This emphasizes the importance of using single-cell data for heterogeneous tissues such as the brain.

We showed that it is possible to prioritize GWAS variants using scEP. Considering the expression of *CACNA1I*, we noticed that one variant, which also overlaps with an ENCODE *cis*-regulatory element, is predicted to have a big effect, while the other has a negligible effect. It could be that the latter affects splicing given that it is an intronic variant. Alternatively, this variant could have been identified during a GWAS study because it is in a linkage disequilibrium block with other (associating) variants, or the variant could affect a completely different gene. GWAS variants are usually linked to the closest gene, but this is not always the correct gene. For instance, the variant rs1421085 was always thought to affect *FTO*, but due to long-range interactions, it affects *IRX3,* a gene a half megabase further [26–28].

Using our model, it is difficult to test whether a variant affects another far away gene since we have a limited input region. Hence, we could only test two variants related to Schizophrenia for *CACNA1I,* out of the 18 variants associated with *CACNA1I [23]*. Ideally, we would increase the length of the input sequence. Using CNNs, however, it is not easy to learn long-range interactions. The Enformer model, which uses a 200kb sequence as input, tackles this problem by combining transformers and CNNs [11]. The Enformer model predicts reads instead of expression values, so we cannot directly extend it or use it for single-cell data. An alternative approach might be to use their well-trained model to get an embedding for every input sequence and to use this embedding to make cell population-specific expression predictions.

All scEP models reach a Pearson correlation of approximately 0.7 regardless of the cell population or tissue trained on. This somewhat low performance raises the question of whether the predictions of these models are trustworthy enough. On the one hand, looking at individual scatterplots (Figure 2D-F, S8), we see for many genes still a relatively high absolute error. Furthermore, it is doubtful how cell population-specific the models are since most cell populations are affected similarly during ISM. On the other hand, we have shown that the model learned the importance of the region around the TSS, the transcription factor binding motif for *INSM1*, and the pattern of most genes correctly among the different subclasses in the motor cortex.

Two future enhancements we envision to improve the performance of our model are concerning the half-life time features and the output of the model. Currently, we extract five features from the mRNA sequence to approximate the half-life time. Recently, a new model, Saluki, was developed that could predict mRNA degradation rates directly based on the sequence of the gene [29]. Replacing these features with the output of Saluki or combining the models directly might improve the predictions. Furthermore, the current output of scEP is the pseudobulk expression for every cell population. By averaging over all cells from that population, however, the variance within a population is lost. Ideally, we might want to predict a distribution for each gene for each population instead of just one aggregated value.

In summary, we provide scEP, a model to predict cell population-specific gene expression by leveraging the resolution of single-cell data. Since this is the first model that uses single-cell data this opens the way for many new developments in this area because have shown now that it is possible.

We envision that our method will be useful for discovering cell population-specific regulatory elements and prioritizing GWAS variants.

Methods

Architecture of scEP

scEP is a one-dimensional convolutional neural network (CNN) adapted from the Xpresso model [9] (Figure 1A, S1). The input to the CNN is four channels with the one-hot encoded sequence around the transcription start site (TSS) (7kb upstream and 3.5kb downstream). Every channel represents one of the four nucleotides (A, C, T, G). For some positions, the exact nucleotide is not known (e.g. any nucleic acid (N) or a purine nucleotide (R)). The exact coding scheme for such positions is shown in Table S4. The CNN consists of two convolutional layers. The output of the convolutional layers is flattened and concatenated with the half-life time features. This is inputted to a fully connected (FC) layer. For the multitask model (used for the cell population-specific predictions), we have only one FC layer. For the other models, we use two FC layers. The output of the FC layers is the aggregated expression per tissue or cell population. Compared to Xpresso, we designed scEP as a multitask model so that it can predict the expression of multiple cell populations simultaneously. Furthermore, we decreased the number of half-life time features that we input into the model from eight to five. The three features we removed (5’ UTR, ORF, and 3’ UTR GC content) correlated less with half-life time, so we removed them to make the model less complex [30,31].

Training scEP

We split the genes into a train, validation, and test dataset and do 20-fold cross-validation. These sets are the same across all experiments (i.e. one train, validation, and test set for mouse genes and one for human genes) such that the results of different models can be compared. We update the weights of scEP using the Adam optimizer based on the loss on the training set. The initial learning rate is set to 0.0005 and if the loss on the validation set is not improved from 5 epochs, the learning rate is reduced by a factor of 10. We train the model for 40 epochs and evaluate the performance using the mean squared error. The model with the lowest loss on the validation set is used for evaluation on the test dataset. Since there is always some stochasticity when training a CNN, we always train 5 models and average the predictions. We used the following software packages for training the model: Pytorch (version 1.9.0) [32], CUDA (version 11.1), cuDNN (version 8.0.5.39), and python (version 3.6.8).

Datasets

Tabula Muris

The single-cell Tabula Muris data [33] for the five different tissues (gland, spleen, lung, limb muscle, and bone marrow) and two different protocols (10X and FACS-sorted Smart-seq2) were downloaded from: https://figshare.com/projects/Tabula_Muris_Transcriptomic_characterization_of_20_organs_and_tissues_from_Mus_musculus_at_single_cell_resolution/27733. To extract input features, we downloaded the reference genome (MM10-PLUS) that was used during the alignment from: https://s3.console.aws.amazon.com/s3/object/czb-tabula-muris-senis?region=us-west-2&prefix=reference-genome/MM10-PLUS.tgz.

The four bulk datasets (spleen, lung, limb muscle, and bone marrow) from the Tabula Muris were downloaded from https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE132040. For the input features, we used the same reference genome as for the single-cell data.

Human motor cortex data

The human motor cortex data from the Allen institute [5] was downloaded from the Cytosplore Comparison Viewer. We downloaded the reference genome (version GRCh38.p2) and corresponding GTF file with information about the location of transcription start sites of the genes here: (https://www.gencodegenes.org/human/release_22.html)

Aggregated expression values

For the single-cell datasets, we used the annotations defined by the authors to aggregate the expression values per tissue or per cell population using $\log_{10}(mean(x))\ (without\ pseudocount)$ into pseudobulk values. The advantage of not adding a pseudocount to the normalization is that the distribution looks more like a normal distribution, which makes it easier to train the model (Figure S19). A limitation, however, is that we could not calculate the exact value for genes that were not expressed in any of the cells. For these genes, we replaced the pseudobulk values with -4, since this extrapolated well (Figure S19). For the bulk data, we aggregated over the samples instead of the cells. Here, we set the genes that are not expressed in any of the cells to -3. We standardize the expression values before running the model such that the average expression in each cell population or tissue is zero and the standard deviation is one. Before analyzing the results and comparing the predictions across cell populations, we undo the z-score normalization, but keep the log-normalization.

Input features

Sequence around the transcription start site

Before extracting the sequences around the transcription start site, we remove genes that are transgenes, ERCC spike-ins, genes without a coding region, and genes on the Y chromosome. This resulted in 20,467 mouse genes and 18,138 human genes. Some genes had multiple transcripts. We downloaded a list with canonical transcripts for each gene from biomart and we used the transcript and transcription start site belonging to the canonical transcript. If the canonical transcript was not defined, we used the transcript that had the longest coding region. After having defined the transcription start site for each gene, we used seqkit [34] to extract sequences from the FASTA file containing the reference genome.

Half-life time features

For every gene we extracted five half-life time features: 5' UTR length, 3' UTR length, ORF length, intron length, and exon junction density ($\frac{\#exons}{length\ ORF} * 1000$). All features are log-normalized using $log_{10}(x + 0.1)$.

Evaluating the predictions

We did 20-fold cross-validation and trained all models five times since there is some randomness when training a deep learning model. For every gene in the test dataset, we averaged the predictions of the five models. We evaluated the performance for every cell population by calculating the Pearson correlation between the true and predicted expression of the genes in the test set. o evaluate the increase in performance between the tissue-specific and cell population-specific model on the Tabula Muris datasets, we calculate: $\Delta_{cp,t} = median\ Pearson\ correlation\ (scEP_{cp,pb}) - median\ Pearson\ correlation\ (scEP_{t,pb})$. On the motor cortex dataset, we also evaluated the performance of each gene by calculating the Pearson correlation between the true and predicted expression per cell population.

In-silico mutagenesis

For *CACNA1I*, we mutated all positions *in-silico*, which means we tested all possible substitutions at every position. We undid the z-score normalization and calculated the difference in expression between the original prediction and the mutated prediction. The models used during these experiments were the models where *CACNA1I* itself was originally in the test set. For every position, we only plotted one predicted difference in expression in Figure 4E. This is the substitution that was predicted to have the biggest absolute effect. We downloaded the locations of the candidate *cis*-regulatory elements using screen registry v3 (release date 2021) [25]. When plotting the difference between two cell populations, we ignored the positions where one is positive and the other predicts a negative effect. This rarely happened and if it was the case, the predicted effect was very small.

Code and data availability

The pseudobulk expression values, trained models, and predictions are available on Zenodo: https://doi.org/10.5281/zenodo.7044908.
The code to reproduce the figures, train your own models, look at the effect of variants, and do *in-silico* mutagenesis is on GitHub: https://github.com/lcmmichielsen/scEP.

Acknowledgments

We would like to thank Dr. Stavros Makrodimitris for his insightful discussion and his example of PyTorch code for convolutional neural networks.
Figures 1B, 2A, 3A, 4A, and S12 were created with BioRender.com.

Author contributions

L.M., M.J.T.R., and A.M. conceived the study and designed the experiments. L.M. performed all the experiments and wrote the paper. L.M., M.J.T.R., and A.M. reviewed and approved the paper.

Funding

This research was supported by an NWO Gravitation project: BRAINSCAPES: A Roadmap from Neurogenetics to Neurobiology (NWO: 024.004.012).

References

1. Lambert SA, Jolma A, Campitelli LF, Das PK, Yin Y, Albu M, et al. The Human Transcription Factors. Cell. 2018;172:650–65.

2. Vaquerizas JM, Kummerfeld SK, Teichmann SA, Luscombe NM. A census of human transcription factors: function, expression and evolution. Nat Rev Genet. 2009;10:252–63.

3. Nott A, Holtman IR, Coufal NG, Schlachetzki JCM, Yu M, Hu R, et al. Brain cell type-specific enhancer-promoter interactome maps and disease-risk association. Science. 2019;366:1134–9.

4. Janssens J, Aibar S, Taskiran II, Ismail JN, Gomez AE, Aughey G, et al. Decoding gene regulation in the fly brain. Nature. 2022;601:630–6.

5. Bakken TE, Jorstad NL, Hu Q, Lake BB, Tian W, Kalmbach BE, et al. Comparative cellular analysis of motor cortex in human, marmoset and mouse. Nature. Nature Publishing Group; 2021;598:111–9.

6. Kelley DR, Snoek J, Rinn JL. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Res. 2016;26:990–9.

7. Kelley DR, Reshef YA, Bileschi M, Belanger D, McLean CY, Snoek J. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. Genome Res. 2018;28:739–50.

8. Zhou J, Theesfeld CL, Yao K, Chen KM, Wong AK, Troyanskaya OG. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. Nat Genet. Nature Publishing Group; 2018;50:1171–9.

9. Agarwal V, Shendure J. Predicting mRNA Abundance Directly from Genomic Sequence Using Deep Convolutional Neural Networks. Cell Rep. Elsevier B.V.; 2020;31:107663.

10. Zhang Y, Zhou X, Cai X. Predicting Gene Expression from DNA Sequence using Residual Neural Network. bioRxiv. Cold Spring Harbor Laboratory; 2020;2020.06.21.163956.

11. Avsec Ž, Agarwal V, Visentin D, Ledsam JR, Grabska-Barwinska A, Taylor KR, et al. Effective gene expression prediction from sequence by integrating long-range interactions. Nat Methods. 2021;18:1196–203.

12. Avsec Ž, Weilert M, Shrikumar A, Krueger S, Alexandari A, Dalal K, et al. Base-resolution models of transcription-factor binding reveal soft motif syntax. Nat Genet. Nature Publishing Group; 2021;53:354–66.

13. Wesolowska-Andersen A, Zhuo Yu G, Nylander V, Abaitua F, Thurner M, Torres JM, et al. Deep learning models predict regulatory variants in pancreatic islets and refine type 2 diabetes association signals. Elife [Internet]. 2020;9. Available from: http://dx.doi.org/10.7554/eLife.51503

14. Wightman DP, Jansen IE, Savage JE, Shadrin AA, Bahrami S, Holland D, et al. A genome-wide association study with 1,126,563 individuals identifies new risk loci for Alzheimer’s disease. Nat Genet. 2021;53:1276–82.

15. Yao X, Glessner JT, Li J, Qi X, Hou X, Zhu C, et al. Integrative analysis of genome-wide association studies identifies novel loci associated with neuropsychiatric disorders. Transl Psychiatry. 2021;11:69.

16. Schizophrenia Working Group of the Psychiatric Genomics Consortium. Biological insights from 108 schizophrenia-associated genetic loci. Nature. 2014;511:421–7.

17. Tasic B, Menon V, Nguyen TN, Kim TK, Jarsky T, Yao Z, et al. Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. Nat Neurosci. NIH Public Access; 2016;19:335–46.

18. Tasic B, Yao Z, Graybuck LT, Smith KA, Nguyen TN, Bertagnolli D, et al. Shared and distinct transcriptomic cell types across neocortical areas. Nature. Nature Publishing Group; 2018;563:72–8.

19. Ikeda M, Takahashi A, Kamatani Y, Momozawa Y, Saito T, Kondo K, et al. Genome-Wide Association Study Detected Novel Susceptibility Genes for Schizophrenia and Shared Trans-Populations/Diseases Genetic Effect. Schizophr Bull. 2019;45:824–34.

20. Li Z, Chen J, Yu H, He L, Xu Y, Zhang D, et al. Genome-wide association analysis identifies 30 new susceptibility loci for schizophrenia. Nat Genet. 2017;49:1576–83.

21. Lam M, Chen C-Y, Li Z, Martin AR, Bryois J, Ma X, et al. Comparative genetic architectures of schizophrenia in East Asian and European populations. Nat Genet. 2019;51:1670–8.

22. Pardiñas AF, Holmans P, Pocklington AJ, Escott-Price V, Ripke S, Carrera N, et al. Common schizophrenia alleles are enriched in mutation-intolerant genes and in regions under strong background selection. Nat Genet. 2018;50:381–9.

23. Buniello A, MacArthur JAL, Cerezo M, Harris LW, Hayhurst J, Malangone C, et al. The NHGRI-EBI GWAS Catalog of published genome-wide association studies, targeted arrays and summary statistics 2019. Nucleic Acids Res. 2019;47:D1005–12.

24. Castro-Mondragon JA, Riudavets-Puig R, Rauluseviciute I, Lemma RB, Turchi L, Blanc-Mathieu R, et al. JASPAR 2022: the 9th release of the open-access database of transcription factor binding profiles. Nucleic Acids Res. 2022;50:D165–73.

25. ENCODE Project Consortium, Moore JE, Purcaro MJ, Pratt HE, Epstein CB, Shoresh N, et al. Expanded encyclopaedias of DNA elements in the human and mouse genomes. Nature. 2020;583:699–710.

26. Smemo S, Tena JJ, Kim K-H, Gamazon ER, Sakabe NJ, Gómez-Marín C, et al. Obesity-associated variants within FTO form long-range functional connections with IRX3. Nature. 2014;507:371–5.

27. Claussnitzer M, Dankel SN, Kim K-H, Quon G, Meuleman W, Haugen C, et al. FTO Obesity Variant Circuitry and Adipocyte Browning in Humans. N Engl J Med. 2015;373:895–907.

28. Broekema RV, Bakker OB, Jonkers IH. A practical view of fine-mapping and gene prioritization in the post-genome-wide association era. Open Biol. 2020;10:190221.

29. Agarwal V, Kelley D. The genetic and biochemical determinants of mRNA degradation rates in mammals [Internet]. bioRxiv. 2022 [cited 2022 Mar 29]. p. 2022.03.18.484474. Available from: https://www.biorxiv.org/content/10.1101/2022.03.18.484474v1

30. Sharova LV, Sharov AA, Nedorezov T, Piao Y, Shaik N, Ko MSH. Database for mRNA Half-Life of 19 977 Genes Obtained by DNA Microarray Analysis of Pluripotent and Differentiating Mouse Embryonic Stem Cells. DNA Res. Oxford Academic; 2009;16:45–58.

31. Spies N, Burge CB, Bartel DP. 3′ UTR-isoform choice has limited influence on the stability and translational efficiency of most mRNAs in mouse fibroblasts. Genome Res. Cold Spring Harbor Laboratory Press; 2013;23:2078–90.

32. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach H, Larochelle H, Beygelzimer A, d\textquotesingle Alché-Buc F, Fox E, Garnett R, editors. Advances in Neural Information Processing Systems 32. Curran Associates, Inc.; 2019. p. 8024–35.

33. Schaum N, Karkanias J, Neff NF, May AP, Quake SR, Wyss-Coray T, et al. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. Nature. Nature Publishing Group; 2018;562:367–72.

34. Shen W, Le S, Li Y, Hu F. SeqKit: A Cross-Platform and Ultrafast Toolkit for FASTA/Q File Manipulation. PLoS One. 2016;11:e0163962.