Briefings in Bioinformatics, 2022, **23(2)**, 1–13

https://doi.org/10.1093/bib/bbab573

Problem Solving Protocol

A universal approach for integrating super large-scale single-cell transcriptomes by exploring gene rankings

Hongru Shen[†], Xilin Shen[†], Mengyao Feng[†], Dan Wu, Chao Zhang, Yichen Yang, Meng Yang, Jiani Hu, Jilei Liu, Wei Wang, Yang Li, Qiang Zhang, Jilong Yang, Kexin Chen and Xiangchun Li

Corresponding authors: Xiangchun Li, Tianjin Cancer Institute, Tianjin Medical University Cancer Institute and Hospital, Huanhu Xi Road, Tiyuan Bei, Hexi District, Tianjin 300060, China. Tel.: +86 022-23372231; E-mail: lixiangchun2014@foxmail.com; Kexin Chen, Department of Epidemiology and Biostatistics, Tianjin Medical University Cancer Institute and Hospital, Huanhu Xi Road, Tiyuan Bei, Hexi District, Tianjin 300060, China. Tel.: +86 022-22372231; E-mail: chenkexin@tmu.edu.cn

[†]These authors contributed equally to this work.

Abstract

Advancement in single-cell RNA sequencing leads to exponential accumulation of single-cell expression data. However, there is still lack of tools that could integrate these unlimited accumulations of single-cell expression data. Here, we presented a universal approach *iSEEEK* for integrating super large-scale single-cell expression via exploring expression rankings of top-expressing genes. We developed *iSEEEK* with 11.9 million single cells. We demonstrated the efficiency of *iSEEEK* with canonical single-cell downstream tasks on five heterogenous datasets encompassing human and mouse samples. *iSEEEK* achieved good clustering performance benchmarked against well-annotated cell labels. In addition, *iSEEEK* could transfer its knowledge learned from large-scale expression data on new dataset that was not involved in its development. *iSEEEK* enables identification of gene–gene interaction networks that are characteristic of specific cell types. Our study presents a simple and yet effective method to integrate super large-scale single-cell transcriptomes and would facilitate translational single-cell research from bench to bedside.

Keywords: deep learning, gene ranking, single-cell transcriptomes

Introduction

Large volume of single-cell transcriptomes is accumulating rapidly. Technical improvements in single-cell RNA sequencing (scRNA-seq) [1] lead to rapid drop in sequencing cost and allows for millions of cells to be sequenced. This was exemplified by the establishment of international collaborative projects on single-cell such as Human Cell Atlas (HCA) [2], COVID-19 Atlas [3], Single Cell Expression Atlas [4], Tabula Muris Atlas [5] and Mouse Cell Atlas [6], which aim at depicting reference map of single-cell signatures. Consequently, integration of these super large-scale data is a challenge and crucial in the era of single-cell data science [7].

Traditional single-cell transcriptome analysis methods such as Seurat [8, 9] and Scanpy [10] are to learn feature representation of gene expression profiles via dimensional reduction on expression profiles of high variable genes (HVGs). However, the deep learning methods such as single-cell variational inference (scVI) [11] and meta-learning approach for identifying and annotating cell types (MARS) [12], in essence analogous to traditional methods, are to perform dimensionality reduction on gene expression of single cells specifically in a nonlinear manner. However, there remain several challenges for single-cell analysis. For instance, there are high discrepancies in the selection of HVGs among different

Xiangchun Li is a Professor at Tianjin Medical University Cancer Institute and Hospital. He has extensive experience in deciphering human cancer genomes via bioinformatics and deep learning.

Kexin Chen is a Professor at Tianjin Medical University Cancer Institute and Hospital. She is an expert on epidemiology and experienced in data mining.

Jilong Yang is a Professor at Tianjin Medical University Cancer Institute and Hospital. He is an expert on sarcoma and molecular biology.

Hongru Shen is a Ph.D. student at Tianjin Medical University. She is currently working on dissection of large-scale single-cell transcriptomic data by deep learning algorithm.

Xilin Shen is a Ph.D. student at Tianjin Medical University. Shen is investigating tumor microenvironment of brain tumor.

Mengyao Feng is a master student at Tianjin Medical University. He is currently working on identifying immunotherapy biomarkers from single-cell.

Dan Wu is a Ph.D. student at Tianjin Medical University Cancer Institute and Hospital.

Chao Zhang is a Ph.D. student at Tianjin Medical University Cancer Institute and Hospital. He is currently working on elucidating single-cell transcriptome of sarcoma.

Yichen Yang is a research assistant at Tianjin Medical University Cancer Institute and Hospital. He has done abundant studies on bioinformatics.

Meng Yang is a research assistant at Tianjin Medical University Cancer Institute and Hospital. She is an expert on tumor genomics and biostatistics.

Jiani Hu is a master student at Tianjin Medical University. She is working on the dissecting tumor genomics.

Jilei Liu is a master student at Tianjin Medical University. He is familiar with biostatistics and currently working on genomic analysis.

Wei Wang is a research assistant at Tianjin Medical University Cancer Institute and Hospital. He is an expert on molecular mechanisms of efficacy of immunotherapy.

Yang Li is a scientist at Tianjin Medical University Cancer Institute and Hospital. She has done abundant studies on transcriptome and gut microbiome.

Qiang Zhang is doctor at Tianjin Medical University Cancer Institute ang Hospital. He is experienced in characterizing medical imaging features in relation to thyroid cancer.

Received: October 11, 2021. **Revised:** December 12, 2021. **Accepted:** December 14, 2021

© The Author(s) 2022. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oup.com

Downloaded from https://academic.oup.com/bib/article/23/2/bbab573/6511497 by Tianjin Medical University user on 14 October 2023

methods [13] and the batch effect further complicates HVG selection [14]. Noise and batch effect are unavoidable as sequencing samples were often compiled from multiple experiments, handling by different personnel, sequenced with different instruments and protocols [15, 16]. The batch effect masks the biological variations and entails batch correction. However, overcorrection is often inevitable [17].

Herein, we introduced *iSEEEK*, a universal approach for integrating super large-scale single-cell transcriptomes via exploring the rankings of top-expressing genes. We hypothesize that the expression information of a single cell is manifested by the rankings of its top-expressing genes. Therefore, we formulated feature representation of single-cell transcriptomes as natural language processing (NLP) task in that the sentence of each single-cell was constructed by concatenation of gene symbols of top-expressing genes ordered by their expression levels. Tremendous progress and enormous achievement were obtained in NLP task. The emergence of Generative Pre-trained Transformer (GPT) [18], Bidirectional Encoder Representations from Transformer (BERT) [19] and Enhanced Language Representation with Informative Entities (ERINE) [20] algorithms revolutionized deep learning in the domain of natural language understanding such as document classification, question answering and semantic similarity assessment. The essence of these algorithms is devoted to modeling associations among tokens and sentences as pretraining task. We developed *iSEEEK* to model the rankings of top-expressing genes on a dataset of 11.9 million single cells. Subsequently, we applied the pretrained *iSEEEK* in downstream tasks such as delineation of cell clusters on three heterogeneous datasets such as peripheral blood mononuclear cells (PBMCs) [9], HCA [21] and expression profiles of 20 organs from Tabula Mursi [5]. We also tested the transferability of *iSEEEK* on a new dataset that was not involved in its development. In addition, we demonstrated the applicability of *iSEEEK* to extract gene–gene interaction networks that are specific for CD4/8+ T cells obtained from fluorescence-activated cell sorting (FACS). The performance of iSEEEK is not expected to outperform current approaches. The purpose is to provide an alternative approach for handling super large-scale transcriptomic data of single cells from a different perspective. *iSEEEK* would facilitate the integration of super large-scale single-cell transcriptomes and translational single-cell research from bench to bedside.

Results

iSEEEK: integration of single-cell expression via exploring expression ranKings of top-expressing genes

iSEEEK was trained with masked language model task to model the expression rankings of the top-expressing genes. *iSEEEK* was trained with 11.9 million single cells collected from public databases covering a variety of cell types from different human tissues under different conditions and mouse tissues (Supplementary Table 1). *iSEEEK* takes as input a sequence of gene symbols ranked by their expression levels (see Methods). The model learns the information of the ranking of the *n* top-expressing genes in a decreased order per cell. In this study, we examined *iSEEEK* with the rankings of the top 126 expressing genes. *iSEEEK* was trained as a masked language modeling task [19, 22]. In this study, the masked language model task randomly masks some of the genes in the input and predicts the vocabulary indexes of masked genes based on their bidirectional contexts. The vocabulary consists of 20 706 protein-encoding genes. *iSEEEK* benefits from multi-head self-attention mechanism and bidirectional encoder representation. The aggregation of feature representations from multi-head attentions improved efficiency and precision. We applied the same data sampling strategy during training as proposed by Devlin *et al*. [19]: the training data generator randomly chooses 15% of the gene positions for prediction. If the i^{th} gene is chosen, we replace it with (1) the [MASK] token 80% of the time, (2) a random gene 10% of the time, (3) the original unchanged gene 10% of the time. *iSEEEK* was trained by cross-entropy loss by comparing its predictions to the original genes (Figure 1**A**). *iSEEEK* consists of 8 transformer layers each with 576 hidden units and 8 attention heads. Detailed parameters of *iSEEEK* are listed in Supplementary Table 2. The developed *iSEEEK* is able to learn the representations of expression-based gene rankings. The latent features extracted from the pretrained *iSEEEK* model can be used as input for downstream task including delineation of cell clusters, identification of marker genes and exploration of cell developmental trajectory (Figure 1**B**).

Clustering performance of *iSEEEK*

We evaluated the clustering performance of *iSEEEK* on three heterogeneous datasets that encompassed bone marrow dataset from HCA Census of Immune Cells [21] ($n = 282\ 558$), PBMCs [9] ($n = 43\ 073$) and Tabula Mursi dataset [5] ($n = 54\ 865$ cells). The HCA bone marrow dataset consisted of 18 cell types with different proportions. The PBMC dataset consisted of CD4+ T cell, CD8+ T cell, natural killer (NK) cells, FCGR3A+ and CD14+ monocytes. The Tabula Mursi dataset included single cells of 20 organs from *Mus musculus*.

iSEEEK was able to reveal distinct cell clusters underlying the composition of each dataset. On the HCA bone marrow dataset, the cell subsets were well separated and the megakaryocytes with low proportion (0.32%) were captured by *iSEEEK* (Figure 2**A**). On the PBMC dataset, *iSEEEK* revealed 23 cell clusters involving eight immune cell subgroups (Figure 2**B**). The cytotoxic lymphocyte cells were gathered together but divided into CD4+ T cell, CD8+ T cell and NK cell subgroup, and monocytes with different markers (FCGR3A+ or CD14+) are also well mapped in particular. On the Tabula

Downloaded from https://academic.oup.com/bib/article/23/2/bbab573/6511497 by Tianjin Medical University user on 14 October 2023

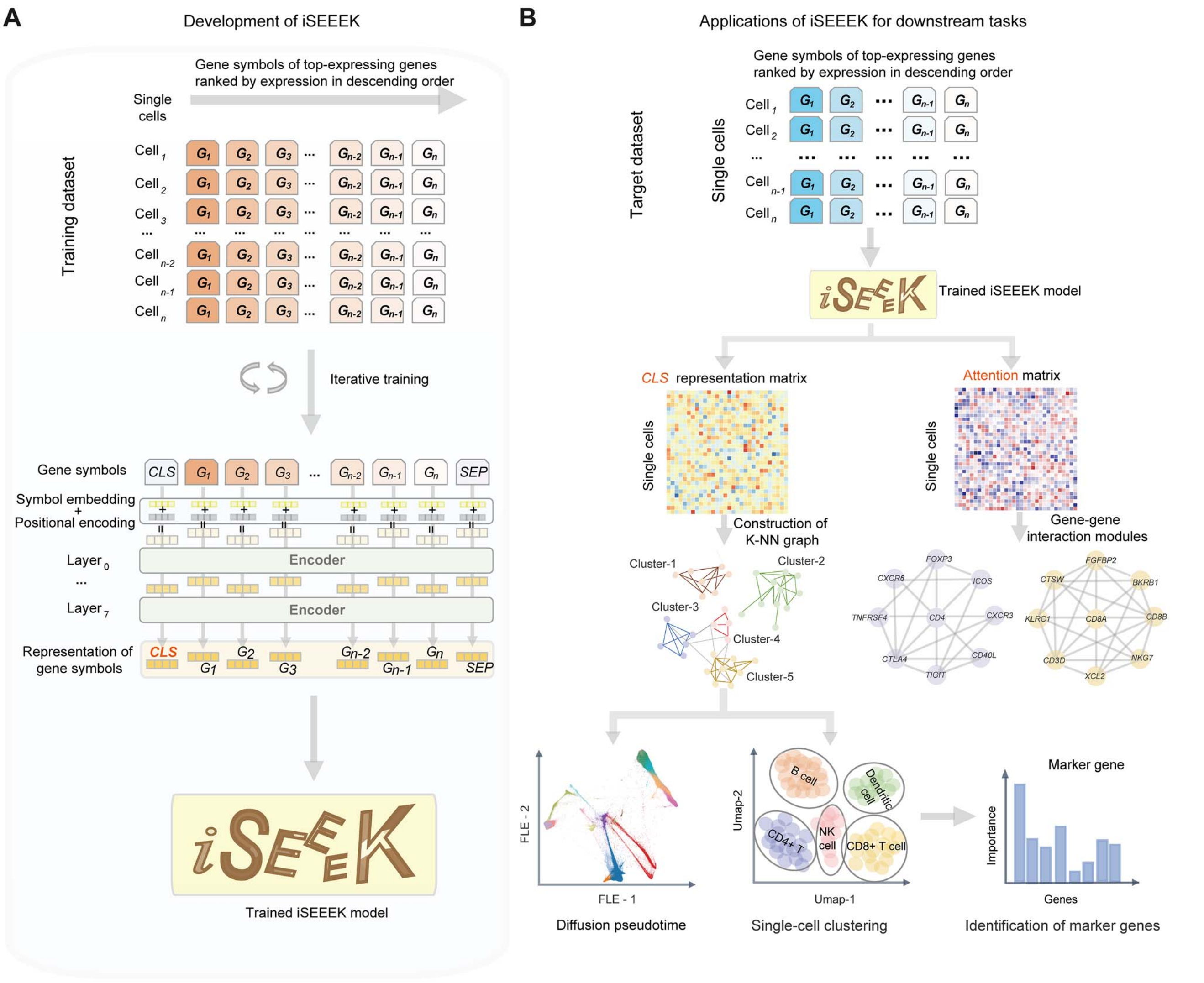

Figure 1. A flowchart depicting the development and downstream applications of *iSEEEK*. (**A**) Development of *iSEEEK* based on the genes symbols of top-expressing genes ranked by expression in descending order for large-scale single cells. (**B**) Downstream application of *iSEEEK* includes delineation of single-cell clustering, pseudotime inference of cell trajectory, identification of marker genes and exploration of cluster-specific gene–gene interaction modules.

Mursi dataset from *M. musculus* composed of 20 mouse organs, *iSEEEK* was able to identify 55 distinct cell types that are well matched with the identity and lineage of organs (Figure 2**C** and Supplementary Figure 1). In the qualitative measurement of cell clustering obtained from *iSEEEK* against putative cell labels, we found that *iSEEEK* achieved an adjusted rand index (ARI) of 0.61 for HCA bone marrow dataset, 0.34 for PBMC dataset and 0.72 for Tabula Mursi dataset. The clustering performance achieved by *iSEEEK* was comparable to those achieved by Scanpy, SC3 and Seurat (Supplementary Figure 2). The Uniform Manifold Approximation and Projection (UMAP) plots of Scanpy, SC3 and Seurat across these three datasets are provided in Supplementary Figure 3.

In addition, we found that *iSEEEK* can work effectively on new dataset that was not involved in the development of *iSEEEK*. As an example, we examined *iSEEEK* on a new dataset obtained from previous study that consisted of 68 579 PBMCs from a healthy donor [23]. *iSEEEK* achieved an ARI of 0.29, which was comparable to Scanpy (Supplementary Figure 4), and the UMAP visualization of the new dataset is shown in Figure 2**D**. Subsequently, we finetuned *iSEEEK* model on this new dataset (Figure 2**E**). We observed that the finetuned *iSEEEK* model achieved an ARI of 0.33 (Figure 2**F**). We found that finetuning *iSEEEK* for one epoch is sufficient (Supplementary Figure 5). The UMAP visualization plots of finetuning *iSEEEK* with different epochs are provided in Supplementary Figure 5. *iSEEEK* exhibited robust performance for different number of top-expressing genes being used (Supplementary Figure 6). *iSEEEK* achieved higher clustering performance over principal component analysis of gene rankings (Supplementary Figure 7). In addition, we showed that *iSEEEK* achieved a comparable acceptance rate of kBET as compared with batch correction methods such as ComBat [24], MNN [25] and BBKNN [26], Seurat [8, 9], Harmony [27], Scanorama [28], Pegasus [29], scVI [11], scArches [30], iMAP [31] and

Downloaded from https://academic.oup.com/bib/article/23/2/bbab573/6511497 by Tianjin Medical University user on 14 October 2023

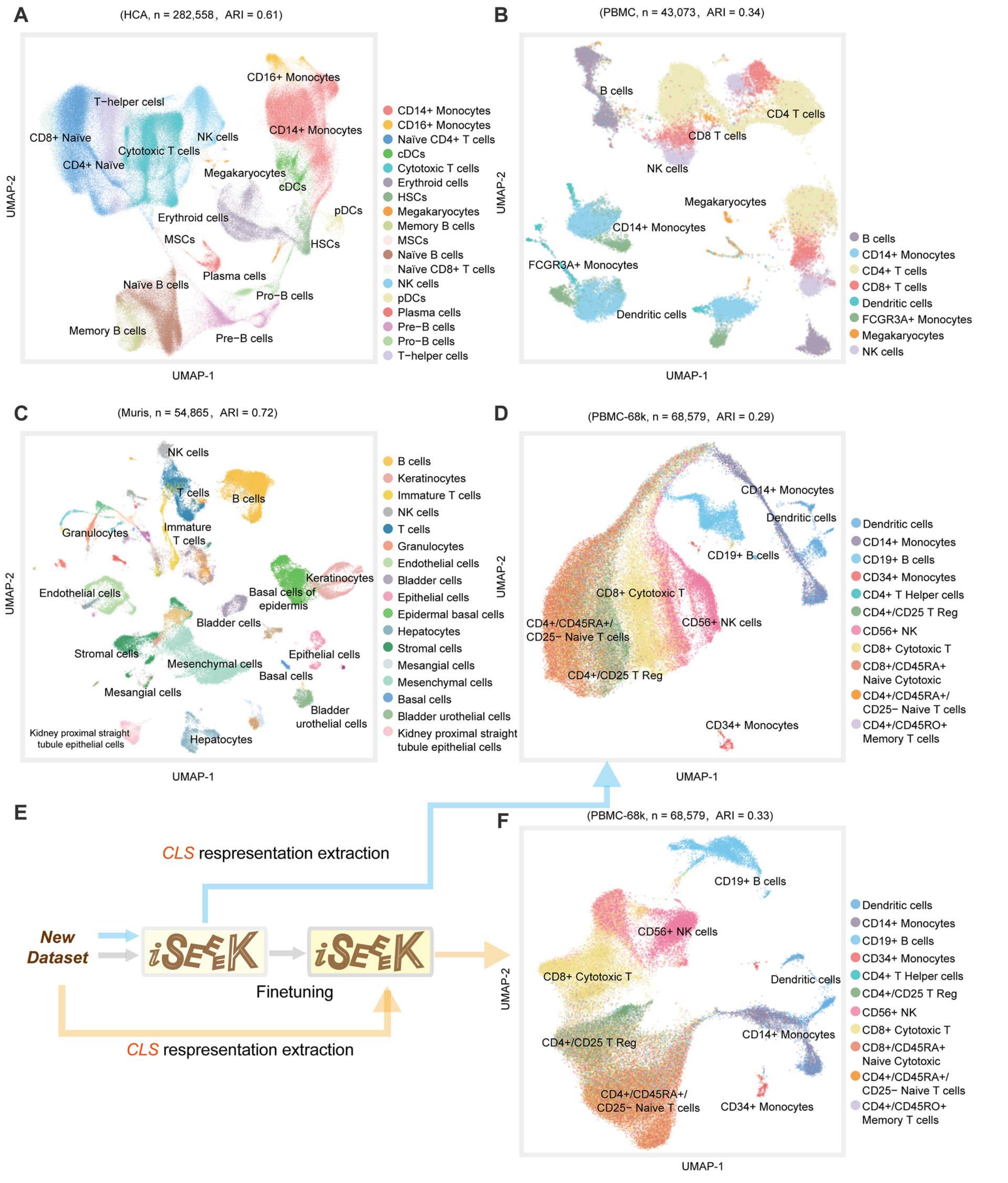

Figure 2. The clustering performances of *iSEEEK*. UMAP visualization of feature representations learned by *iSEEEK* on the (**A**) HCA dataset, (**B**) PBMC dataset, (**C**) Tabula Mursi dataset and (**D**) PBMC-68 k dataset that was not involved in the development of *iSEEEK*. (**E**) Fine-tuning *iSEEEK* with new dataset PBMC-68 k. (**F**) UMAP visualization of feature representations of PBMC-68 k dataset with features extracted from *iSEEEK* being fine-tuned on the PBMC-68 k dataset.

DESC [32] measured on the HCA bone marrow dataset (Supplementary Figure 8). *iSEEEK* is scalable especially on large-scale dataset as it works on mini batches of data sequentially to avoid loading all data into memory. The memory usage and runtime of *iSEEEK* along with Scanpy [10], scSCope [33] and SHARP [34] are provided in Supplementary Table 3.

iSEEEK reserves the development trajectory of B cells on HCA dataset

We used the feature representation learned by *iSEEEK* to construct pseudotemporal trajectories of bone marrow cells on HCA bone marrow dataset (see Methods). We identified a developmental trajectory rooted at stem cells toward multiple cell types with distinguishable interme-

Downloaded from https://academic.oup.com/bib/article/23/2/bbab573/6511497 by Tianjin Medical University user on 14 October 2023

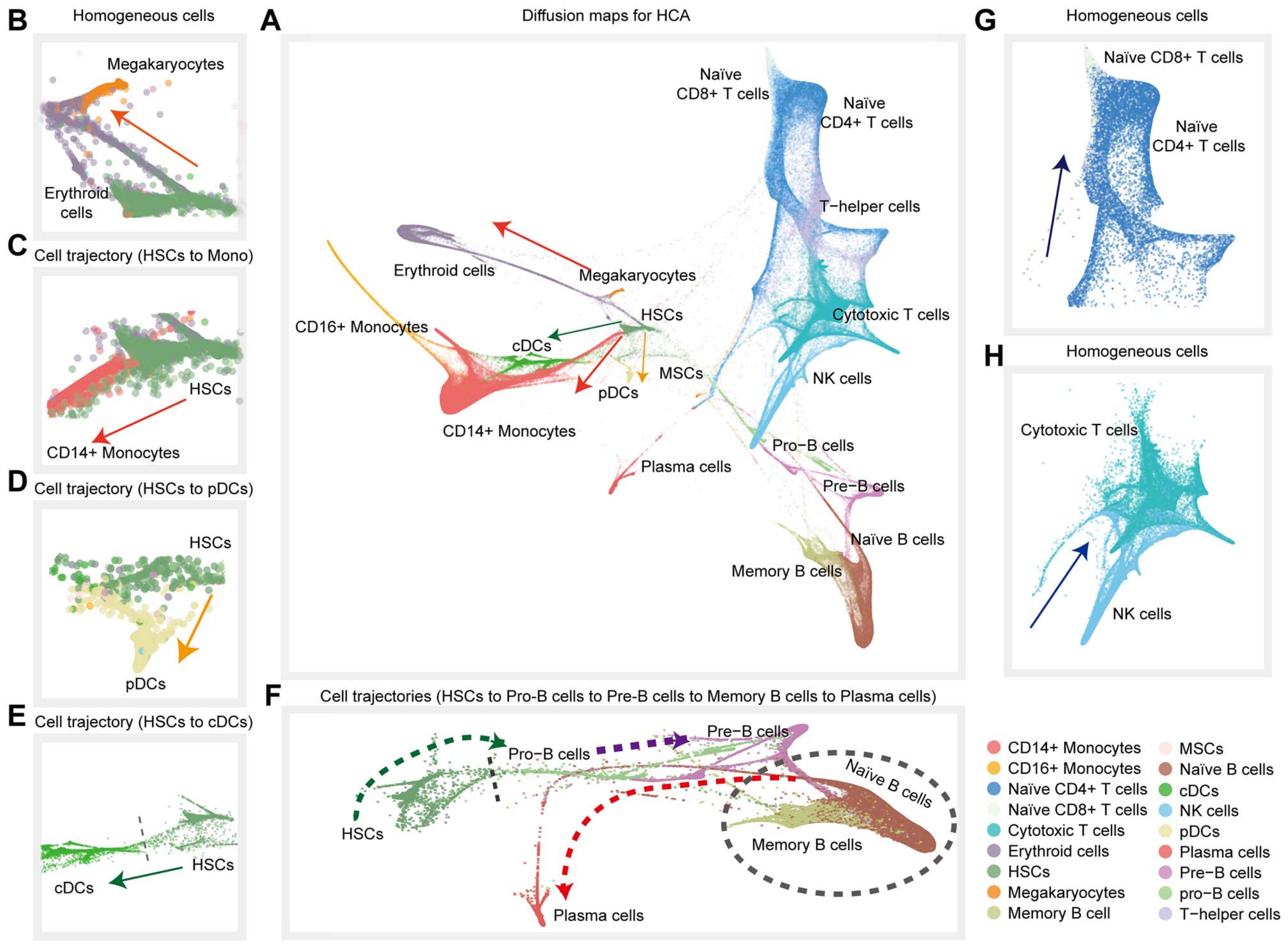

Figure 3. Diffusion pseudotime analysis of bone marrow cells in HCA dataset. (**A**) The panorama diffusion map of HCA dataset with the cell types colored. (**B**) Bifurcation of megakaryocytes and erythroid cells. Bifurcation of CD14+ monocytes (**C**), pDCs (**D**) and cDCs (**E**) from HSCs. (**F**) The developmental trajectory of B cells from HSCs, toward Pro-B cells, pre-B cells, matured naïve B cells, memory B cells and plasma cells. The arrows represent the directionality of the cell developmental trajectory. (**G**) Bifurcation of naïve CD4+ T cells and naïve CD8+ T cells, similarly, (**H**) cytotoxic T cells and NK cells.

diate stages (Figure 3**A**). We identified a developmental trajectory of B cells (Figure 3), with an initial wave of B cell progenitors (Pro-B cells) derived from hematopoietic stem cells (HSCs), then followed by precursors of B cells (pre-B cells), matured naïve B cells (Figure 3**F**), and finally bifurcated into memory B cells and plasma cells [35]. Meanwhile, we also observed differentiation of HSCs into multiple types of immune cells including plasmacytoid dendritic cells (pDCs), conventional dendritic cells (cDCs) and CD14+ monocytes (Figure 3**C**–**E**). In addition, the baicalia type of cell trajectories was observed for megakaryocytes and erythroid cells [36] (Figure 3**B**), naïve CD4+ T cells and naïve CD8+ T cells (Figure 3**G**), cytotoxic T cells and NK cells (Figure 3**H**), suggesting that they were originated from the same progenitor cells [37]. The developmental trajectory derived from *iSEEEK* is consistent with trajectory inferred from Scanpy (Supplementary Figure 9).

iSEEEK enables discovery of marker genes and gene interaction modules

We added and trained a classifier at the end of *iSEEEK* for identification of marker genes on the dataset of FACS-sorted CD4/8+ T cells (see Methods). An apparent separation of CD4+ and CD8+ T cells was observed on the UMAP visualization plot (Figure 4**A**). The identified marker genes for CD4+ T cells include *CD4*, *TXNIP* and *CD2* (Figure 4**B**). CD8+ T cells were featured by cytotoxic markers such as *CD8A*, *CD8B*, *KLRK1* and *NKG7* (Figure 4**B**).

We respectively obtained gene interaction networks that are characteristic of CD4+ and CD8+ T cells through analyzing the attention matrices of *iSEEEK* for the dataset of FACS-sorted CD4/8+ T cells (see Methods, Figure 4**C** and **D**). A CD4+ T cell specific gene interaction module (Figure 4**E**) derived from Figure 4**C** was featured by genes that involved in the development and function of CD4+ T cells (i.e. *CXCR6, FOXP3*, *ICOS*, *CCR7* and *SELL*) [38] and immune suppression (i.e. *PDCD1*, *TIGIT, BATF* and *TNF* receptor family) [39–41] (Figure 4**E**). These interactions are overrepresented in the STRING gene–gene interaction database (16/244 interactions; hypergeometric test, $P = 5.0e-4$). Among these interactions, *CD2/PTPRC* interaction is involved in the activation of T cell receptor [42]. *FOXP3/TNFRSF18* interaction is critical for T cell differentiation [43]. The CD8+ T cell specific module

Downloaded from https://academic.oup.com/bib/article/23/2/bbab573/6511497 by Tianjin Medical University user on 14 October 2023

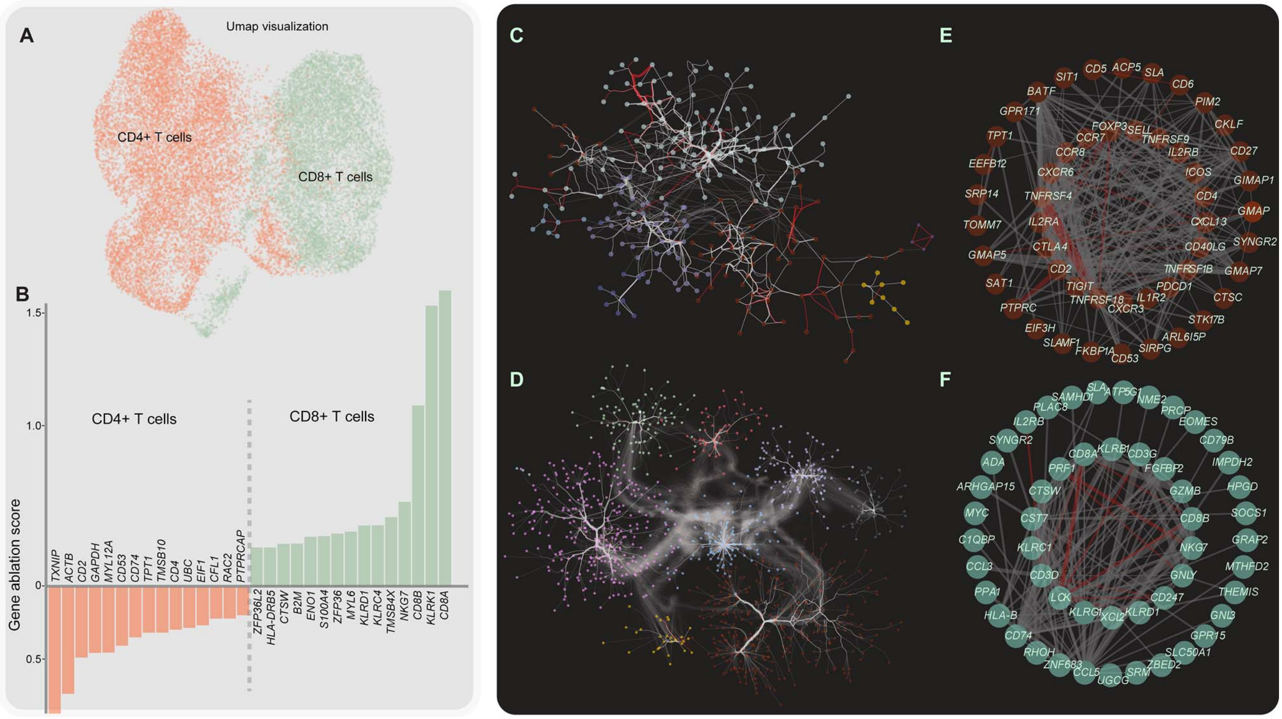

Figure 4. Marker genes and exemplified gene–gene interaction networks deciphered from FACS-sorted CD4/8+ T cells dataset. (**A**) UMAP visualization CD4+ and CD8+ T cells. (**B**) Barplot representation of marker genes for CD4+ and CD8+ T cells. (**C** and **D**) The gene–gene interaction networks for CD4+ and CD8+ T cells, respectively. (**E** and **D**) The gene interaction modules characteristic of CD4+ and CD8+ T cells, respectively. The red edge indicates it is represented in STRING gene–gene interaction database. The thickness of the edge is proportional to attention weights among interacted genes.

(Figure 4**F**) is characterized by interactions among cytotoxic genes including *GNLY*, *NKG7*, *PRF1*, *LCK* and *KLRD1* [44]. In addition, the CD8+ T cell recruitment gene *CCL5* [45] exhibited strong interaction with markers of CD8+ T cells including *CD8A*, *CD8B* and *GZMB*. Gene interactions from the CD8+ T cell-specific module are enriched in STRING database (12/144 interactions; hypergeometric test, $P = 1.3e\text{-}3$).

Discussion

In this study, we presented a universal approach *iSEEEK* for integrating super large-scale single-cell transcriptomes by exploring the rankings of top-expressing genes. *iSEEEK* was developed on 11.9 million single-cell transcriptomes covering a wide variety of cell types from *Homo sapiens* and *M. musculus*. The notable features of *iSEEEK* are that it only relies on gene rankings but not on actual expression levels; thus, its sensitivity to batch effect should be decreasing. This feature makes *iSEEEK* a good candidate for integrating super large-scale amount of single-cell expression data. The performance of *iSEEEK* is expected to improve as more and more data are involved in its development.

This study demonstrated that pretraining on the rankings of top-expressing genes from super large-scale scRNA-seq data is effective. The efficiency of cell cluster delineation on the extracted latent features of the pretrained *iSEEEK* was demonstrated on three heterogeneous datasets encompassing different cell types, sequencing with different protocol and deriving from different species. Across these three datasets, *iSEEEK* achieved comparable ARI metric as compared with Scanpy. In addition, *iSEEEK* also worked efficiently on new dataset that was not involved in its development. Finetuning *iSEEEK* for one epoch appears sufficient to improve its clustering performance.

iSEEEK enables to maximize the value of big data from single-cell transcriptomes in simple and yet effective way. *iSEEEK* can make use of single-cell transcriptomes from different species, for example, *H. sapiens* and *M. musculus* in our study. *iSEEEK* circumvents the tremendous challenge of batch correction in single-cell integration by modeling gene expression rankings rather than actual expression levels. As *iSEEEK* is not relying on actual expression levels but rather on the ranking of top-expressing genes, its sensitivity to batch effect is decreasing, which was verified in this study (Supplementary Figure 8). Batch correction methods such as ComBat [24], MNN [25] and BBKNN [26] require explicit knowledge of the batch information. However, the batch information is not always available and often neglected by researchers; therefore, traditional methods are not appropriate for data integration of multiple datasets without batch information. In addition, traditional methods [8, 9] are memory hungry as they require to load all data into memory, hampering their ability to process super large-scale dataset

Downloaded from https://academic.oup.com/bib/article/23/2/bbab573/6511497 by Tianjin Medical University user on 14 October 2023

(Supplementary Table 3). In contrast, *iSEEEK* was trained in a stochastic manner that only a small batch of samples are processed at each time step. Thus, memory consumption of *iSEEEK* is much lower than traditional methods and it can benefit from acceleration brought by graphical processing unit.

iSEEEK is quite different from that of other traditional methods as they require selection of hypervariable genes (HVGs), batch correction and data normalization [46, 47], whereas *iSEEEK* uses the ranking of top-expressing genes and does not require selection of HVGs. Batch correction methods are sensitive to data volume and the number of batches, and the robustness of the batch correction is difficult to assess in large-scale dataset [24–26]. Meanwhile, the consistency and reproducibility of the HVGs are also difficult to control by different HVG selection methods [13]. *iSEEEK* takes as input the rankings of top-expressing genes, which may be less informative intuitively as compared with the use of expression levels of HVGs as traditional methods. However, *iSEEEK* was able to precisely identify cell types of small proportions such as FCGR3A+ and CD14+ monocytes in the PBMC dataset (Figure 2**B**), suggesting that the rankings of top-expressing genes are sufficient for delineation of cell types with small proportions.

We demonstrated that feature representation of the rankings of top-expressing genes learned by *iSEEEK* preserved the chronological order of cell development trajectories. We verified the continuous and identifiable cell trajectory from Pro-B cells derived from HSCs toward plasma cells [35] on HCA bone marrow dataset (Figure 3**F**).

As a preliminary endeavor, we demonstrated that by analyzing *iSEEEK* for the input of CD4/8+ T cells, we were able to identify gene interaction modules manifested the features of CD4/8+ T cells. Functional related tend to have strong interactions. The attention mechanism in *iSEEEK* makes it possible to learn interaction among different genes. As the attention mechanism enables modeling gene interaction by taking into account the influence of other genes, it has the potential to learn complex gene–gene interaction networks and may shed new lights on gene regulation circuits.

In this study, we formulate single-cell transcriptome integration as a language modeling task. Recent advances in NLP will benefit single-cell integration. The paradigm of pretraining-then-finetuning is a de facto procedure in NLP as this paradigm is robust to overfitting and has the advantage of making use of super large-scale data and reducing the need of big data on downstream tasks [48].

iSEEEK is built upon the paradigm of NLP tasks where a sentence is represented as the indices of tokens within a discrete vocabulary. This entails the input for *iSEEEK* must be a list of indices defined by a vocabulary of gene symbols, which consists of 20 701 protein coding genes. Therefore, we simplified the expression profile of each cell as the indices of the top-expressing genes in the vocabulary. In the pretraining stage, *iSEEEK* is trained to predict the discrete indices for the masked gene symbols in the vocabulary. Although this simplification could make *iSEEEK* insensitive to batch effect, it discards information that are not captured by actual gene expression, which is one of the limitations of our study. The other limitation of *iSEEEK* is that it is not able to deal with batch correction as the ranking of top-expressing genes was obtained without batch correction. In the current setting, the latent features learned by *iSEEEK* can be used as input for the other batch correction methods if correction of batch effect is desired.

Herein, we provided a universal, scalable, transferable, effective and easy-to-use approach for integration of super large-scale single-cell transcriptomes. *iSEEEK* can be finetuned on a specific dataset to tackle specific downstream tasks. We expected that *iSEEEK* may be helpful for researchers to elucidate the heterogeneous and dynamic biological processes underlying human diseases with the accumulation of single-cell transcriptomes.

Conclusions

In the study, we presented a universal approach for integrating super large scale for single-cell transcriptomes by modeling feature representation of the rankings of top-expressing genes as a masked language modeling task. We are in the process of developing a web server running *iSEEEK* that would be freely available to the research community. Our work represented a new paradigm in the integration of super large-scale single-cell transcriptomes and may be helpful for the elucidation of the dynamic and heterogeneity of single cells.

Methods

Dataset and preprocessing

We collected expression matrices of 11.9 million single cells from previous studies. Detailed information for these studies is provided in Supplementary Table 1. We discarded mitochondrial genes, ribosomal genes and non-protein coding genes. Subsequently, we concatenated the 126 top-expressing genes as a sentence for each single cell. Eventually, we obtained a text file of 11.9 million sentences. The five datasets used in downstream task of *iSEEEK* are described below:

HCA—Bone marrow data of 282 588 cells from 64 healthy donors in HCA project subjected to 10× sequencing protocol [21]. There are 18 cells types annotated by HCA including erythrocytes, mesenchymal stem cells, HSC and diverse immune cells.

PBMCs—This dataset was downloaded from Gene Expression Omnibus repository [9] (GSE96583). It consists of 43 095 single cells obtained from five individuals (3 systemic lupus erythematosus and 2 control) subjected to 10× sequencing. All cells were grouped into eight categories: B cells, CD4+ T cells, CD8+ T cells, dendritic

Downloaded from https://academic.oup.com/bib/article/23/2/bbab573/6511497 by Tianjin Medical University user on 14 October 2023

cells, megakaryocytes, FCGR3A+ monocytes, CD14+ monocytes and natural killer cells.

Tabula Mursi—A dataset of 100 000 single-cell from Mouse Cell Atlas [5] across 20 different organs subjected to 10× and Smart-seq2 sequencing protocols. About 54 865 cells were sorted by FACS, and therefore, we used these 54 865 cells for evaluation.

PBMC-68 k—This PBMC-68 k dataset included 68 579 PBMCs obtained from a healthy donor (http://support.10xgenomics.com/single-cell/datasets).

FACS-sorted CD4/8+ T cells—This dataset includes 12 670 CD4+ and 9012 CD8+ T cells that were sorted by FACS from tumor patients diagnosed with liver cancer, colorectal cancer and lung cancer [49–51]. They were subjected to smart-seq sequencing.

The *iSEEEK* model

iSEEEK consists of an embedding layer and 8 encoder layers each with 576 hidden units and 8 attention heads.

Embedding layer

The embedding layer takes the embeddings of a sequence of 128 gene symbol tokens and their position embeddings as input. An input representation of gene symbol token can be represented as $\left[CLS, G_1, G_2, \ldots, G_n, SEP\right]$. *CLS* is the classification token and *SEP* is the sentence separation token. G_i is the gene symbol of the ith gene. The gene symbols are first converted into indexes in the gene symbol dictionary. The gene symbol dictionary consists of protein-encoding genes.

Encoder layer

The encoder layer is a transformer that is the core component of *iSEEEK*. It consists of a multi-head self-attention and a feed-forward network interconnected with layer normalization layer. Residual connection is added to improve information flow. The multi-head self-attention enables the model to capture contextual information. The self-attention head is formulated as:

$$\boldsymbol{Attention}\,(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \boldsymbol{softmax}\left(\frac{\mathrm{Q}\mathrm{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\mathrm{V}$$

The self-attention head takes *Q*, *K* and *V* as inputs and applies softmax transformation. *Q*, *K* and *V* are projected from the input. The scaling factor $\sqrt{d_k}$ is used to mitigate the extreme small gradient [52].

Input representations

We constructed a dictionary with protein-encoding genes. For each cell, we prepared a sequence of 128 tokens, where tokens are gene symbols and/or special tokens are [CLS], [SEP] and [PAD]. We filtered out genes with extremely low expression (i.e. an expression level of 1 or 0) and ranked them according to their expression levels. We padded [PAD] token to the input sequence if the number of genes is <126. The first token is always [CLS] and the last token is always [SEP].

Model pretraining

iSEEEK take a sequence gene symbols with a maximum length of 126 as input. We applied the same data sampling strategy during training as BERT [19]: the training data generator randomly chooses 15% of the gene positions for prediction. If the ith gene is chosen, we replace it with (1) the [MASK] token 80% of the time, (2) a random gene 10% of the time, (3) the origin al unchanged gene 10% of the time. *iSEEEK* was trained by cross-entropy loss by comparing its predictions to the original genes. We used an optimizer of Adam with learning rate of 1e-4, $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate warmup over the first 10 000 steps, linear decay of the learning rate, a batch size of 64, the epochs of 48. The *PyTorch* (version 1.7.1) and *transformers* (version 4.6.0) packages were used to develop *iSEEEK*.

Identification of marker genes

We added a classifier to the end of the pretrained *iSEEEK* and trained on the FACS-sorted CD4/8+ T cells. The parameters of the pretrained *iSEEEK* were frozen and parameter updating was applied for the linear classifier. We trained this classifier with a learning rate of 0.001 and batch size of 16 with Adam optimizer for 30 epochs. We quantitatively measure the impact of a specific gene as the difference between the logit values for the original gene sequence and gene sequence with that gene replaced with [UNK] token. Specifically, for an input gene sequence of $S = [G_1,\ G_2,\ \ldots,\ G_n]$, we obtained $S^* = [G_1,\ UNK,\ \ldots,\ G_n]$ by replacing G_2 with *UNK*. Let *L* and L^* denote the logit values obtained from the classifier, the influence of G_2 on the decision made by this classifier is defined as:

$$\Delta = L - L^*$$

For a specific cell type, we rank the influence of genes by the average value of Δ and those ranked on the top are considered to be marker genes.

Diffusion pseudotime analysis

The affinity matrix of cells $W_{n\times n}$ was constructed from representation features of the *CLS* token, which is performed using community detection algorithms [53], and the hierarchical navigable small world (HNSW) algorithm [54] is applied to find the top-k nearest neighbors. A scaled Gaussian kernel is used to define the distance between cell-*x* and cell-*y* as:

$$K\,(x, y) = \left(\frac{2\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2}\right)^{\frac{1}{2}} \exp\left(-\frac{\|x - y\|^2}{\sigma_x^2 + \sigma_y^2}\right)$$

x and y are representation features of the *CLS* token for cell-*x* and cell-y, respectively. $\sigma_x$is the local kernel width of x, calculated as the median value of *x* and its top-k

Downloaded from https://academic.oup.com/bib/article/23/2/bbab573/6511497 by Tianjin Medical University user on 14 October 2023

nearest cells. The affinity matrix is defined as:

$$W(x, y) = \left\{ \begin{array}{c} k'(x, y), y \in n(x)/x \in n(x) \\ 0, \textbf{otherwise} \end{array} \right.$$

where $k'\left(x, y\right)$ is defined as:

$$k'(x, y) = \frac{K(x, y)}{q(x)q(y)}$$

The Markov chain transition matrix P and the symmetric transition matrix Q are then calculated based on the affinity matrix as follows:

$$D = \textbf{diag}\left(\sum_{y} W(x, y)\right),$$

$$P = D^{-1}W, Q = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$$

The symmetrical matrix Q can be decomposed as UAU^{T}. Let $\Psi = D^{-\frac{1}{2}}U$. A family with parameter timescale of t for approximated diffusion maps $\{\Psi_t\}_{t\in\mathbb{N}\cup\{\infty\}}$is defined as:

$$\Psi_t(x_i) = \begin{pmatrix} \lambda_1^t \Psi_1(i) \\ \lambda_2^t \Psi_2(i) \\ \vdots \\ \lambda_{n-1}^t \Psi_{n-1}(i) \end{pmatrix}$$

The approximated diffusion pseudotime (DPT) maps $\left\{\Psi_t{}'\right\}_{t\in\mathbb{N}\cup\{\infty\}}$ are constructed based on the aforementioned diffusion maps as:

$$\Psi_t{}'(x_i) = \sum_{t'=1}^{t} \Psi_{t'}(x_i) = \begin{pmatrix} \lambda_1 \frac{1-\lambda_1^t}{1-\lambda_1} \Psi_1(i) \\ \lambda_2 \frac{1-\lambda_2^t}{1-\lambda_2} \Psi_2(i) \\ \vdots \\ \lambda_{n-1} \frac{1-\lambda_{n-1}^t}{1-\lambda_{n-1}} \Psi_{n-1}(i) \end{pmatrix}$$

The diffusion maps and diffusion pseudotime maps are performed using package *Pegasus* [29] (v1.4.3) with K set to 30. The cell trajectory was visualized with force-directed layout embedding algorithm [55]. We set δ and nδ as its the default parameter: $\delta = 2.0$ and n$\delta = 5000$.

Construction of gene interaction network

We constructed the cell-type-specific gene interactions, respectively, for CD4+ and CD8+ T cells based on the FACS-sorted CD4/8+ T cell dataset [23]. For each input sequence consisted of n genes, we can extract an attention matrix a of n columns and n rows corresponding to each attention head. Attention weight $a_{i,j}$ denotes the attention of gene i to gene j. Gene attention matrix of a specific cell type was constructed from the attention matrix a for each cell from that cell type. Specifically, we define an indicator function $f(i, j, \theta)$ that returns 1 if the attention weight between gene i and j $a_{i,j} > \theta$, and 0 otherwise. The attention matrix a specific cell type (C_a) was constructed as follows:

$$C\alpha(f) = \sum_{x\in X} \sum_{i=1}^{|x|} \sum_{j=1}^{|x|} f(i, j, \theta) \times \alpha i, j$$

θ is a threshold to filter out low attentions and a value of 0.05 was used in this study.

Given that attentions between gene i and j are not identical to j and i, therefore the attention matrix a specific cell type was further refined as:

$$G(i, j) = C\alpha(f)i, j + C\alpha(f)j, i$$

We retained the top 10% interactions in $G(i, j)$ in subsequent analysis. Network construction was carried out with Python package *networkx* (version 2.5). Functional modules of networks were detected through Louvain community detection algorithm [56] based on package *python-community* (version 0.15). Overrepresentation of detected modules in STRING gene–gene interaction database [57] was evaluated with hypergeometric test. A $P < .05$ was considered statistically significant. The gene interaction networks were visualized using *Cytoscape* (version 3.8.2) [58].

Single-cell clustering and evaluation

We extracted the represented features of each single-cell with the pretrained *iSEEEK*. The extracted features were used as input to the K-Nearest Neighbors (KNN) algorithm to construct KNN graphs for subsequent single-cell community detection by Leiden [59] algorithm. We applied single-cell clustering pipeline implemented in Scanpy (v1.6.0) to perform single-cell clustering on KNN graph. The uniform manifold approximation and projection [60] (UMAP) is used for visualizing clustering result.

For comparison, we also performed single-cell clustering using Scanpy (v1.6.0), Seurat (v3.1.5) [8, 9] and SC3 (v1.15.1) [61] as the benchmarking tools. The conventional single-cell analysis is based on the gene expression. We first filtered out cells and the criteria: the number of expression genes <200 or mitochondrial counts >30%. The highly variable genes (HVGs) were selected with default parameters (i.e. *max_mean* = 3 and *min_mean* = 0.0125). We used the *FindIntegrationAnchors* and *IntegrateData* from Seurat to construct the gene expression matrix. We used the 50 principal components to construct the KNN graph and subsequently applied Leiden community detection algorithm to delineate cluster with default parameter (i.e. resolution = 1) for Seurat and Scanpy (v1.6.0). For SC3, we used the approach of combining unsupervised and supervised methods [61] on the HVG expression matrix. We also

Downloaded from https://academic.oup.com/bib/article/23/2/bbab573/6511497 by Tianjin Medical University user on 14 October 2023

analyzed the data by using the gene rankings as input for Scanpy. For Seurat and SC3, we randomly selected 10 samples from the HCA bone marrow dataset for comparison due to memory overflow on our server.

We also compared the computationally time and cost-memory of *iSEEEK* and other methods, including Scanpy (v1.6.0) [10], scSCope (v0.1.5) [33] and SHARP (v1.1.0) [34]. For the sake of fairness, the analysis step of scScope and SHARP gathered the HVG-expression matrix from Scanpy. The runtime of data preprocessing ranged from data loading to construction of HVG-expression matrix and those of feature extraction ranged from input matrix to features matrix. The peak memory usage for each method is reported.

We used *ARI*, normalized mutual information (*NMI*), *V-measure* and *Silhouette coefficient* to measure clustering performance. The clustering metrics were calculated with sklearn (v0.21.2) python package.

ARI—The *ARI* metric is calculated on the contingency table summarizing the truth labels and clustering. In the contingency table, rows and columns represent truth and clustering labels, respectively. *ARI* is defined as:

$$ARI = \frac{\Sigma_{ij}\binom{n_{ij}}{2} - \left[\Sigma_i\binom{a_i}{2}\Sigma_j\binom{a_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\Sigma_i\binom{a_i}{2}+\Sigma_j\binom{a_j}{2}\right]-\left[\Sigma_i\binom{a_i}{2}\Sigma_j\binom{a_j}{2}\right]/\binom{n}{2}}$$

[[DmEquation12]]where n_{ij} denoted the numbers of cell in common between clustering labels and truth labels, a_i the sum of i^{th} row and a_j the sum of j^{th} column of the contingency table.

NMI—The *NMI* metric is used to evaluate the similarity between the clustering labels and actual labels. Assuming that the clustering labels and actual labels of *N* cells are *U* and *V*, the entropy is defined as:

$$H(U) = -\sum_{i=1}^{|U|} P(i)\log\left(P(i)\right)$$

where $p(i) =| U_i | /N$ is the probability that a cell picked at random from *U* falls into U_i. In a similar manner, $H(V)$ is defined as:

$$H(V) = -\sum_{j=1}^{|V|} P'(j)\log\left(P'(j)\right)$$

where $p'(j) =| V_j | /N$. The mutual information between *U* and *V* is calculated as:

$$MI\left(U,V\right) = \sum_{i=1}^{|U|}\sum_{j=1}^{|V|} P\left(i,j\right)\log\left(\frac{P\left(i,j\right)}{P(i)P'(j)}\right)$$

where $p\left(i,j\right) =| U_i \cap V_j | /N$ is the probability that a cell picked at random falls into classes U_i and V_j. The NMI is thus defined as:

$$NMI\left(U,V\right) = \frac{MI\left(U,V\right)}{\sqrt{H(U)H(V)}}$$

V-measure—The *V-measure* metric is calculated as the harmonic average value of homogeneity and compactness. The homogeneity is defined as:

$$h = 1 - \frac{H\left(C|K\right)}{H(C)}$$

$H\left(C|K\right)$ is the conditional entropy of category division under given cluster division conditions, which is calculated as:

$$H\left(C|K\right) = -\sum_{C=1}^{|C|}\sum_{K=1}^{|K|}\frac{n_{c,k}}{n}\log\left(\frac{n_{c,k}}{n_k}\right)$$

$\mathbf{H}\left(\mathbf{C}\right)$ is the conditional entropy of category division, which is defined as:

$$H(C) = -\sum_{C=1}^{|C|}\frac{n_c}{n}\log\left(\frac{n_c}{n}\right)$$

where n is the number of instances.$n_c$and n_k are the number of instances in cluster c and k, respectively.$n_{c,k}$ is the number of instances in cluster c, which are classified into cluster k.

The compactness is defined as c:

$$c = 1 - \frac{H\left(K|C\right)}{H(C)}$$

V-measure is defined as:

$$v = \frac{2 \times h \times c}{h + c}$$

Silhouette Coefficient—The *Silhouette Coefficient* is calculated using the mean intracluster distance (a) and the mean nearest cluster distance (b) for each sample. The *Silhouette Coefficient* for sample (i) is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\left(a(i), b(i)\right)}$$

Batch correction and evaluation

We used the acceptance rate of kBET [62] as a measurement of batch effect. The acceptance rate measures whether cells from different batches are well-mixed in the local neighborhood of each cell. The acceptance rate obtained from *iSEEEK* was compared with the other batch correction methods on the HCA bone marrow dataset, and the methods included Combat (v1.8.0) [63], MNN (v1.8.0) [25], BBKNN (v1.7.1) [26], Seurat (v3.1.5) [8, 9], Harmony (v0.1.6) [27], Scanorama (v1.7.1) [28], Pegasus (v1.4.0) [29], scVI (v0.0.0) [11], scArches (1.7.0) [30], iMAP (1.0.0) [31], DESC (2.1.1) [32]. Seurat was run with default parameters. The other methods used HVG-expression matrix from Scanpy as input.

Downloaded from https://academic.oup.com/bib/article/23/2/bbab573/6511497 by Tianjin Medical University user on 14 October 2023

kBET acceptance rate. We assumed the dataset of single-cell with batches of m, and there are n_j cells in batch j. The batch mixing frequency denotes as $f = \left(f_1, \cdots, f_m\right)$, where $f_j = \frac{n_j}{N}$. The number of neighbors of cell-i belonging to batch j is n_{ji}^k. Its χ^2test statistic with degrees of (m-1) is calculated as:$k_i^k = \sum_{j=1}^m \frac{\left(n_{ji}^k - f_j \cdot k\right)^2}{f_j \cdot k}$. The P-value is calculated as: $p_i^k = 1 - F_{m-1}\left(k_i^k\right)$, where $F_{m-1}(x)$ represents the cumulated density function. The kBET acceptance rate is defined as the percentage of cells that accept the null hypothesis at significance level α as follows:

$$\boldsymbol{kBET - rate} = \frac{\sum_{i=1}^N I\left(p_i^k \geq \alpha\right)}{N} \times 100\%$$

$I(x)$ is the indicator function where $I(x) = 1$ if $x > 0$ otherwise $I(x) = 0$. We used Pegasus (v1.4.3) to calculate the kBET acceptance rate by setting K and α to 5 and 0.01, respectively.

Key Points

- *iSEEEK* can work effectively on new dataset that was not involved in its development, enabling transferability of knowledge learned from large-scale transcriptomes on new dataset.
- *iSEEEK* reserves the developmental trajectory of cells.
- *iSEEEK* is capable of identifying cell-type-specific markers.
- *iSEEEK* is able to identify gene–gene interaction networks.

Supplementary data

Supplementary data are available online at https://academic.oup.com/bib.

Code availability

The source code and the pretrained model of *iSEEEK* is publicly available at Github: https://github.com/lixiangchun/iSEEEK.

Author contributions

X.L. and K.C. designed and supervised the study; H.S, X.S., M.F. and X.L. performed data collection, analysis, and wrote the manuscript; H.S., X.S. and X.L. developed the model; C.Z., D.W., X.S. M.F., J.H., J.L., Y.Y., Y.L., M.Y. W.Z. and Q.Z. collected data; X.L., K.C., J.Y. and H.S. revised the manuscript.

Acknowledgements

We are grateful for researchers for their generosity to made their data publicly available.

Funding

National Natural Science Foundation of China (grant no. 31801117 to X.L., 31900471 to M.Y. and 82073287 to Q.Z.); Tianjin Municipal Health Commission Foundation (grant no. RC20027 to Y.L.) and IRT_14R40 from the Program for Changjiang Scholars and Innovative Research Team in University in China (K.C.).

References

1. Fan HC, Fu GK, Fodor SP. Expression profiling. Combinatorial labeling of single cells for gene expression cytometry. *Science* 2015;**347**(6222):1258367.
2. Regev A, Teichmann SA, Lander ES, *et al. The human cell atlas*, *Elife* 2017;**6**:e27041.
3. Ren X, Wen W, Fan X, *et al.* COVID-19 immune features revealed by a large-scale single-cell transcriptome atlas. *Cell* 2021;**184**(7):1895–1913.e19.
4. Papatheodorou I, Moreno P, Manning J, *et al.* Expression atlas update: from tissues to single cells. *Nucleic Acids Res* 2020;**48**(D1):D77–83.
5. Tabula Muris C, Overall c, Logistical c, Organ c, *et al.* Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature* 2018;**562**(7727):367–72.
6. Han X, Wang R, Zhou Y, *et al.* Mapping the mouse cell atlas by microwell-Seq. *Cell* 2018;**173**(5):1307.
7. Lahnemann D, Koster J, Szczurek E, *et al.* Eleven grand challenges in single-cell data science. *Genome Biol* 2020;**21**(1):31.
8. Butler A, Hoffman P, Smibert P, *et al.* Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat Biotechnol* 2018;**36**(5):411–20.
9. Kang HM, Subramaniam M, Targ S, *et al.* Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nat Biotechnol* 2018;**36**(1):89–94.
10. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol* 2018;**19**(1):15.
11. Lopez R, Regier J, Cole MB, *et al.* Deep generative modeling for single-cell transcriptomics. *Nat Methods* 2018;**15**(12):1053–8.
12. Brbic M, Zitnik M, Wang S, *et al.* MARS: discovering novel cell types across heterogeneous single-cell experiments. *Nat Methods* 2020;**17**(12):1200–6.
13. Yip SH, Sham PC, Wang J. Evaluation of tools for highly variable gene discovery from single-cell RNA-seq data. *Brief Bioinform* 2019;**20**(4):1583–9.
14. Finak G, McDavid A, Yajima M, *et al.* MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biol* 2015;**16**:278.
15. Tung PY, Blischak JD, Hsiao CJ, *et al.* Batch effects and the effective design of single-cell gene expression studies. *Sci Rep* 2017;**7**:39921.
16. Hicks SC, Townes FW, Teng M, *et al.* Missing data and technical variability in single-cell RNA-sequencing experiments. *Biostatistics* 2018;**19**(4):562–78.

Downloaded from https://academic.oup.com/bib/article/23/2/bbab573/6511497 by Tianjin Medical University user on 14 October 2023

17. Luecken M, Büttner M, Chaichoompu K, *et al.* Benchmarking atlas-level data integration in single-cell genomics. *bioRxiv* 2020; 2020.2005.2022.111161.
18. Radford A, Narasimhan K, Salimans T, *et al. Improving language understanding by generative pre-training*, 2018; preprint at https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageunderstandingpaper.pdf
19. Devlin J, Chang M-W, Lee K, *et al. BERT: pre-training of deep bidirectional transformers for language understanding*. arXiv:1810.04805. 2018.
20. Zhang Z, Han X, Liu Z, *et al.* ERNIE: enhanced language representation with informative entities. arXiv preprint arXiv:190507129. 2019.
21. Regev A, Teichmann S, Rozenblatt-Rosen O, *et al.* The human cell atlas white paper. *arXiv preprint arXiv:181005192* 2018.
22. Taylor WL. "Cloze procedure": a new tool for measuring readability. *J Quarter* 1953;**30**(4):415–33.
23. Zheng GX, Terry JM, Belgrader P, *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat Commun* 2017;**8**:14049.
24. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 2007;**8**(1):118–27.
25. Haghverdi L, Lun AT, Morgan MD, *et al.* Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat Biotechnol* 2018;**36**(5):421–7.
26. Polański K, Young MD, Miao Z, *et al.* BBKNN: fast batch alignment of single cell transcriptomes. *Bioinformatics* 2020;**36**(3):964–5.
27. Korsunsky I, Millard N, Fan J, *et al.* Fast, sensitive and accurate integration of single-cell data with harmony. *Nat Methods* 2019;**16**(12):1289–96.
28. Hie B, Bryson B, Berger B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. *Nat Biotechnol* 2019;**37**(6):685–91.
29. Li B, Gould J, Yang Y, *et al.* Cumulus provides cloud-based data analysis for large-scale single-cell and single-nucleus RNA-seq. *Nat Methods* 2020;**17**(8):793–8.
30. Lotfollahi M, Naghipourfar M, Luecken MD, *et al.* Mapping single-cell data to reference atlases by transfer learning. *Nat Biotechnol* 2021;1–10.
31. Wang D, Hou S, Zhang L, *et al.* iMAP: integration of multiple single-cell datasets by adversarial paired transfer networks. *Genome Biol* 2021;**22**(1):63.
32. Li X, Wang K, Lyu Y, *et al.* Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis. *Nat Commun* 2020;**11**(1):2338.
33. Deng Y, Bao F, Dai Q, *et al.* Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning. *Nat Methods* 2019;**16**(4):311–4.
34. Wan S, Kim J, Won KJ. SHARP: hyperfast and accurate processing of single-cell RNA-seq data via ensemble random projection. *Genome Res* 2020;**30**(2):205–13.
35. LeBien TW, Tedder TF. B lymphocytes: how they develop and function. *Blood* 2008;**112**(5):1570–80.
36. Klimchenko O, Mori M, DiStefano A, *et al.* A common bipotent progenitor generates the erythroid and megakaryocyte lineages in embryonic stem cell–derived primitive hematopoiesis. *Blood* 2009;**114**(8):1506–17.
37. Trinchieri G. Biology of natural killer cells. *Adv Immunol* 1989;**47**: 187–376.
38. Luckheeram RV, Zhou R, Verma AD, *et al.* CD4(+)T cells: differentiation and functions. *Clin Dev Immunol* 2012;**2012**:925135.
39. Harjunpaa H, Blake SJ, Ahern E, *et al.* Deficiency of host CD96 and PD-1 or TIGIT enhances tumor immunity without significantly compromising immune homeostasis. *Onco Targets Ther* 2018;**7**(7):e1445949.
40. Watts TH. TNF/TNFR family members in costimulation of T cell responses. *Annu Rev Immunol* 2005;**23**:23–68.
41. Murphy TL, Tussiwand R, Murphy KM. Specificity through cooperation: BATF-IRF interactions control immune-regulatory networks. *Nat Rev Immunol* 2013;**13**(7):499–509.
42. Koretzky GA, Picus J, Schultz T, *et al.* Tyrosine phosphatase CD45 is required for T-cell antigen receptor and CD2-mediated activation of a protein tyrosine kinase and interleukin 2 production. *Proc Natl Acad Sci U S A* 1991;**88**(6):2037–41.
43. Ono M, Yaguchi H, Ohkura N, *et al.* Foxp3 controls regulatory T-cell function by interacting with AML1/Runx1. *Nature* 2007;**446**(7136):685–9.
44. Yang HQ, Wang YS, Zhai K, *et al.* Single-cell TCR sequencing reveals the dynamics of T cell repertoire profiling during pneumocystis infection. *Front Microbiol* 2021;**12**:637500.
45. Chang LY, Lin YC, Mahalingam J, *et al.* Tumor-derived chemokine CCL5 enhances TGF-beta-mediated killing of CD8(+) T cells in colon cancer by T-regulatory cells. *Cancer Res* 2012;**72**(5): 1092–102.
46. Dillies M-A, Rau A, Aubert J, *et al.* A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Brief Bioinform* 2013;**14**(6):671–83.
47. Pachter L. *Models for transcript quantification from RNA-Seq*. arXiv preprint arXiv:11043889, 2011.
48. Rasmy L, Xiang Y, Xie Z, Tao C, Zhi D. Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ Digit Med*, 2021;**4**(1): 86.
49. Guo X, Zhang Y, Zheng L, *et al.* Global characterization of T cells in non-small-cell lung cancer by single-cell sequencing. *Nat Med* 2018;**24**(7):978–85.
50. Zheng C, Zheng L, Yoo JK, *et al.* Landscape of infiltrating T cells in liver cancer revealed by single-cell sequencing. *Cell* 2017;**169**(7):1342–1356.e16.
51. Zhang L, Yu X, Zheng L, *et al.* Lineage tracking reveals dynamic relationships of T cells in colorectal cancer. *Nature* 2018;**564**(7735):268–72.
52. Vaswani A, Shazeer N, Parmar N, *et al.* Attention is all you need. *arXiv preprint arXiv:170603762*, 2017.
53. Reichardt J, Bornholdt S. Statistical mechanics of community detection. *Physical review E* 2006;**74**(1):016110.
54. Malkov YA, Yashunin DA. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans Pattern Anal Mach Intell* 2018;**42**(4):824–36.
55. Schiebinger G, Shu J, Tabaka M, *et al.* Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell* 2019;**176**(4):928–943.e922.
56. Blondel VD, Guillaume J-L, Lambiotte R, *et al.* Fast unfolding of communities in large networks. *J Stat Mech* 2008;**2008**(10):10008.
57. Szklarczyk D, Gable AL, Lyon D, *et al.* STRING v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Res* 2019;**47**(D1):D607–13.
58. Shannon P, Markiel A, Ozier O, *et al.* Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res* 2003;**13**(11):2498–504.
59. Traag VA, Waltman L, van Eck NJ. From Louvain to Leiden: guaranteeing well-connected communities. *Sci Rep* 2019;**9**(1): 5233.

Downloaded from https://academic.oup.com/bib/article/23/2/bbab573/6511497 by Tianjin Medical University user on 14 October 2023

60. Becht E, McInnes L, Healy J, *et al.* Dimensionality reduction for visualizing single-cell data using UMAP. *Nat Biotechnol* 2019;**37**(1):38–44.
61. Kiselev VY, Kirschner K, Schaub MT, *et al.* SC3: consensus clustering of single-cell RNA-seq data. *Nat Methods* 2017;**14**(5): 483–6.
62. Buttner M, Miao Z, Wolf FA, *et al.* A test metric for assessing single-cell RNA-seq batch correction. *Nat Methods* 2019;**16**(1): 43–9.
63. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 2007;**8**(1):118–27.

Downloaded from https://academic.oup.com/bib/article/23/2/bbab573/6511497 by Tianjin Medical University user on 14 October 2023