Bioinformatics, 38(5), 2022, 1393–1402
https://doi.org/10.1093/bioinformatics/btab834
Advance Access Publication Date: 10 December 2021
Original Paper

Data and text mining

CellMeSH: probabilistic cell-type identification using indexed literature

Shunfu Mao [1,†], Yue Zhang [2,†], Georg Seelig[1,2,*] and Sreeram Kannan[1,*]

[1]Electrical and Computer Engineering Department, University of Washington, Seattle, WA 98195, USA and [2]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA 98195, USA

*To whom correspondence should be addressed.

[†]The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Associate Editor: Inanc Birol

Received on August 9, 2020; revised on November 21, 2021; editorial decision on November 27, 2021; accepted on December 6, 2021

Abstract

Motivation: Single-cell RNA sequencing (scRNA-seq) is widely used for analyzing gene expression in multi-cellular systems and provides unprecedented access to cellular heterogeneity. scRNA-seq experiments aim to identify and quantify all cell types present in a sample. Measured single-cell transcriptomes are grouped by similarity and the resulting clusters are mapped to cell types based on cluster-specific gene expression patterns. While the process of generating clusters has become largely automated, annotation remains a laborious *ad hoc* effort that requires expert biological knowledge.

Results: Here, we introduce CellMeSH—a new automated approach to identifying cell types for clusters based on prior literature. CellMeSH combines a database of gene–cell-type associations with a probabilistic method for database querying. The database is constructed by automatically linking gene and cell-type information from millions of publications using existing indexed literature resources. Compared to manually constructed databases, CellMeSH is more comprehensive and is easily updated with new data. The probabilistic query method enables reliable information retrieval even though the gene–cell-type associations extracted from the literature are noisy. CellMeSH is also able to optionally utilize prior knowledge about tissues or cells for further annotation improvement. CellMeSH achieves top-one and top-three accuracies on a number of mouse and human datasets that are consistently better than existing approaches.

Availability and implementation: Web server at https://uncurl.cs.washington.edu/db_query and API at https://github.com/shunfumao/cellmesh.

Contact: gseelig@uw.edu or ksreeram@uw.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Single-cell RNA sequencing (scRNA-seq) provides an unprecedented resolution in understanding cellular heterogeneity at the single-cell level, and offers novel biological insights into multi-cellular organisms (Baron *et al.*, 2016; Cao *et al.*, 2017; Consortium, 2018; Grün and van Oudenaarden, 2015; Han *et al.*, 2018; Jaitin *et al.*, 2014; Klein *et al.*, 2015; Rosenberg *et al.*, 2018; Satija *et al.*, 2015; Setty *et al.*, 2016; Shin *et al.*, 2015; Tang *et al.*, 2009; Tasic *et al.*, 2016; Trapnell *et al.*, 2014; Usoskin *et al.*, 2014; Welch *et al.*, 2016; Zeisel *et al.*, 2015; Zheng *et al.*, 2017). A key step required in order to enable the aforementioned applications is cell-type identification that annotates cells with biologically meaningful cell types. However, cell-type annotation remains primarily a manual process, and automatic cell-type identification is an important open problem (Lähnemann *et al.*, 2020).

One line of automatic cell-type identification methods (Chen *et al.*, 2013; Fisher, 1935; Franzén *et al.*, 2019; Hänzelmann *et al.*, 2013; Stachelscheid *et al.*, 2013; Zhang *et al.*, 2018b) (Supplementary File 3 Table 1) annotates *clusters of cells* obtained via a standard scRNA-seq workflow. However, the range of cell types that can be annotated as well as the accuracy of these methods remains insufficient. A typical scRNA-seq analysis workflow (Butler *et al.*, 2018; Orr Ashenberg and Institute, 2019; Wolf *et al.*, 2018; Zhang *et al.*, 2020) begins with the preparation of gene–cell expression matrix [a gene–cell expression matrix is obtained from the raw-reads after a sequence of steps such as read quality control (Andrews, 2010), alignment (Dobin *et al.*, 2012; Kim *et al.*, 2013) and quantification (Bray *et al.*, 2016; Li and Dewey, 2011)]. This matrix is used as a starting point on which clustering (Kiselev *et al.*, 2017; Lin *et al.*, 2017; Mukherjee *et al.*, 2018; Sun *et al.*, 2019;

Zhang *et al.*, 2018a), dimension reduction (Pierson and Yau, 2015; Wang *et al.*, 2017), and differential expression analysis (Soneson and Robinson, 2018) are further applied giving rise to a set of genes that are expressed specific to a cluster (which we refer to as cluster differentially expressed genes or DEGs). To annotate clusters with cell types, existing methods (Chen *et al.*, 2013; Fisher, 1935; Franzén *et al.*, 2019; Hänzelmann *et al.*, 2013; Stachelscheid *et al.*, 2013; Zhang *et al.*, 2018b) use the cluster DEGs to query databases that connect genes to cell types. The databases are collected either from a few specific studies (Chen *et al.*, 2013; Stachelscheid *et al.*, 2013), from manual literature surveys (Fisher, 1935; Hänzelmann *et al.*, 2013; Zhang *et al.*, 2018b) or from scRNA-seq experiments that have their clustered cells pre-annotated according to cell-type markers manually compiled from literature (Franzén *et al.*, 2019). The database query mechanisms can return a list of unsorted cell types (Franzén *et al.*, 2019; Stachelscheid *et al.*, 2013; Zhang *et al.*, 2018b) or a list of cell types sorted by their statistical significance with the query genes, essentially based on Fisher's exact test (Chen *et al.*, 2013; Fisher, 1935) or a Kolmogorov–Smirnov test (Hänzelmann *et al.*, 2013). The common issue for these cell-type identification methods is that their databases are not comprehensive; more critically it is also laborious to update and expand them.

Another line of recent work (Alavi *et al.*, 2018; Aran *et al.*, 2019; Butler *et al.*, 2018; Hou *et al.*, 2019; Kiselev *et al.*, 2018; Ma and Pellegrini, 2019; Pliner *et al.*, 2019; Tan and Cahan, 2019; Zhang *et al.*, 2019) (Supplementary File 3 Table 1) predicts cell types for single cells (rather than clusters) using the gene–cell expression matrix directly. However, these methods require either existing annotated gene expression profiles (Alavi *et al.*, 2018; Aran *et al.*, 2019; Butler *et al.*, 2018; Hou *et al.*, 2019; Kiselev *et al.*, 2018; Ma and Pellegrini, 2019; Tan and Cahan, 2019) or hand-curated cell-type marker-gene files (Pliner *et al.*, 2019; Zhang *et al.*, 2019) as prior knowledge. The majority of these methods follow a machine learning approach, by first training a model on prior knowledge, and then utilizing the trained model either to classify the input gene expression vector to a reference cell type (Ma and Pellegrini, 2019; Pliner *et al.*, 2019; Tan and Cahan, 2019), or to project the input gene expression vector to an embedding vector and match to the reference cell type that has the most similar embedding (Alavi *et al.*, 2018). Some of these methods follow a more statistical approach (without training), by annotating the input gene expression vector with the reference cell type that has the highest correlation (Aran *et al.*, 2019; Butler *et al.*, 2018; Hou *et al.*, 2019; Kiselev *et al.*, 2018) or maximum a posteriori estimation score (Zhang *et al.*, 2019) for the input. None of these methods are able to annotate cell types that have not been seen in prior experiments (Alavi *et al.*, 2018; Aran *et al.*, 2019; Butler *et al.*, 2018; Hou *et al.*, 2019; Kiselev *et al.*, 2018; Ma and Pellegrini, 2019; Tan and Cahan, 2019) or absent from the marker file which typically contains only a small number of known cell types (Pliner *et al.*, 2019; Zhang *et al.*, 2019).

The main goal of this article is to address the shortcomings of existing cell-type identification methods (Supplementary File 3 Table 1) by exploiting indexed literature resources such as MEDLINE (2015) and Gene2pubmed (Maglott *et al.*, 2007). We particularly focus on cell-type annotation at the cluster level (Chen *et al.*, 2013; Fisher, 1935; Franzén *et al.*, 2019; Hänzelmann *et al.*, 2013; Stachelscheid *et al.*, 2013; Zhang *et al.*, 2018b). MEDLINE contains Medical Subject Headings (MeSH) (MeSH, 2019), a set of hierarchically organized biological terms, including cell types, for a large class of biomedical publications, while Gene2pubmed is a database of *NCBI* genes (Maglott *et al.*, 2007) associated with these same publications. A natural approach is then to build a database that connects genes with MeSH cell types. Since genes and cell types are indexed for a large class of publications, the database forms a rich resource in associating genes with cell types. Furthermore, since the underlying resources (MEDLINE and Gene2pubmed) expand as new papers come up, the extracted database can also be automatically updated. However, connecting these genes and MeSH cell types simply based on the number of papers where they co-occur results in spurious gene–cell relationships, and biases due to the widely varying number of publications mentioning a gene or cell-type. Existing query methods (Fisher, 1935; Hänzelmann *et al.*, 2013) may not work well for such a noisy database, because they all implicitly assume that the database is noiseless and has only true gene–cell associations. Therefore, utilizing these literature resources necessitates the design of novel query methods. In addition, it is possible that there is prior knowledge about the cell types for the query data. For example, we might know the tissue that the cell type originates from, or have a set of potential cell types. Existing systems (Alavi *et al.*, 2018; Chen *et al.*, 2013) have not explored such information. Enabling the query methods to utilize prior knowledge will help reduce the search space for better performance.

Here, we propose CellMeSH (cell-type annotation with MeSH terms), a new method to annotate clustered single-cell data, comprising two key parts: a database of gene–cell-type mappings, and a novel query method. Its accompanying web server and open-source application programming interface (API) are able to take an input of a set of genes (such as the DEGs of a cluster of cells) and optional prior knowledge of possible cell types, and output a list of candidate cell types sorted by their relevance to the genes (Fig. 1). Unlike many of the methods that assign cell types to cells using gene cell expressions directly, CellMeSH neither needs a separate training dataset, nor requires a manually curated set of marker genes. CellMeSH is designed for cluster-level cell-type annotation given the DEGs for the cluster. Its open-source API is expected to be a component plugable into existing cluster-level cell-type annotation procedures (Satija *et al.*, 2015; Zhang *et al.*, 2020) to make the end-to-end cell-type annotation experience more smooth by combining existing procedures (e.g. clustering, differential gene expression analysis) and CellMeSH's unique ability of predicting cell types automatically based on given DEGs as markers.

There are three key innovations in CellMeSH. First, CellMeSH builds its database in a scalable way, by automatically linking genes indexed in Gene2pubmed and MeSH cell types indexed in MEDLINE from millions of publications. Such large-scale gene–cell linking makes the database more comprehensive and easier to expand when new literature comes online. Second, to address the challenges of publication bias and potentially error-prone gene–cell associations in building the database, we develop a novel probabilistic database query method using maximum likelihood estimation. Third, if the query data are known to include certain types of cells, CellMeSH is able to use that knowledge to constrain the search and improve the annotation results.

Through a variety of experiments on human and mouse scRNA-seq datasets, we demonstrate that CellMeSH has richer information in its database linking genes and cell types, a robust query method, and an overall better annotation performance than existing cluster-level annotation methods. We also conducted experiments to compare CellMeSH to cell-level annotation methods, and CellMeSH shows a comparable annotation performance without using annotated reference data.

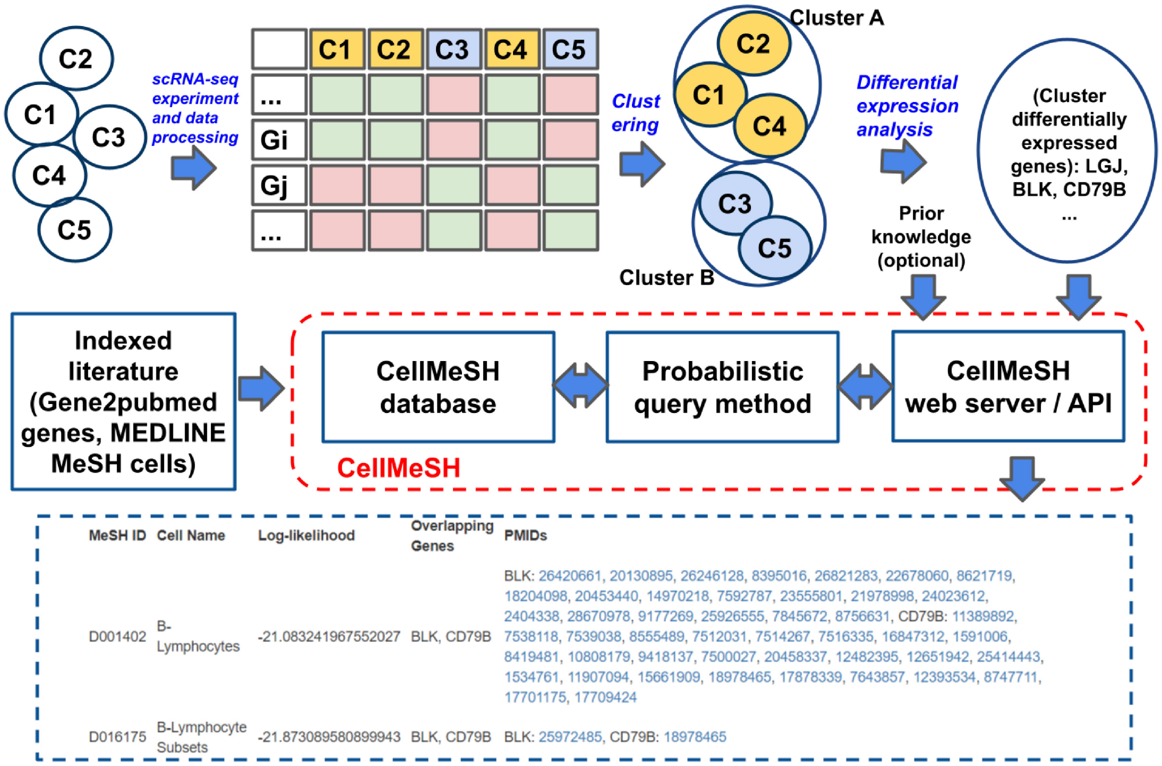
Below, we first go through the key parts of CellMeSH including the database, query method, and optional usage of prior knowledge. We next demonstrate the superior annotation performance of CellMeSH for human and mouse scRNA-seq datasets. We then describe the CellMeSH web server and its open-source API. Finally, we discuss future directions for CellMeSH.

2 Materials and methods

2.1 CellMeSH database

The CellMeSH database is a collection of tables and each table has genes (species-specific, e.g. mouse, human, *Caenorhabditis elegans*, etc.) as rows and cell types (species-independent) as columns. Every entry in the table contains a list of publications, each of which is indexed with the gene and the cell type.

To construct the CellMeSH database, we first filter MEDLINE for references containing MeSH cell types (Fig. 2). MEDLINE (2015) is a bibliographic database containing around 30 million


C2
C1
C3
C4
C5
scRNA-seq experiment and data processing
C1
C2
C3
C4
C5
...
Gi
Gj
...
Clust ering
Cluster A
C2
C1
C4
C3
C5
Cluster B
Differential expression analysis
(Cluster differentially expressed genes): LGJ, BLK, CD79B ...
Prior knowledge (optional)
Indexed literature (Gene2pubmed genes, MEDLINE MeSH cells)
CellMeSH database
Probabilistic query method
CellMeSH web server / API
CellMeSH

MeSH ID	Cell Name	Log-likelihood	Overlapping Genes	PMIDs
D001402	B-Lymphocytes	-21.083241967552027	BLK, CD79B	BLK: 26420661, 20130895, 26246128, 8395016, 26821283, 22678060, 8621719, 18204098, 20453440, 14970218, 7592787, 23555801, 21978998, 24023612, 2404338, 28670978, 9177269, 25926555, 7845672, 8756631, CD79B: 11389892, 7538118, 7539038, 8555489, 7512031, 7514267, 7516335, 16847312, 1591006, 8419481, 10808179, 9418137, 7500027, 20458337, 12482395, 12651942, 25414443, 1534761, 11907094, 15661909, 18978465, 17878339, 7643857, 12393534, 8747711, 17701175, 17709424
D016175	B-Lymphocyte Subsets	-21.873089580899943	BLK, CD79B	BLK: 25972485, CD79B: 18978465

Fig. 1. CellMeSH overview. CellMeSH enables automated annotation of cell types, which is usually the last step of scRNA-seq analysis. CellMeSH can be accessed through a web server and API. The input to CellMeSH are the DEGs of clustered cells and optional prior knowledge of cell types, the outputs are ranked candidate cell types for each cluster, and for each candidate cell the log-likelihood score (Equation (2)), the overlapping genes and the related literature resources. CellMeSH relies on a database created by linking the Gene2pubmed genes and MEDLINE MeSH cell types. The database is queried by a probabilistic method based on maximum likelihood estimation

references to biomedical and life science journal articles, including to most articles in PubMed. Each MEDLINE reference associates a set of MeSH (MESH, 2019) terms with each article. MeSH includes 570 terms related to cell types (nested under the MeSH category ‘Cell’ with tree number A11). Cell types are not species-specific, as the MeSH ontology does not distinguish cell types with respect to species (Though cell types are species-independent, when predicting cell types given species-specific genes, cell types that do not exist in the target species are likely to have low scores and are unlikely to be selected.). Filtering MEDLINE for MeSH cell types results in a reduced dataset of 3.8M articles.

Then, for each target species (human and mouse), we further filter MEDLINE to keep only the references containing the species-specific genes from Gene2pubmed, a database that links standardized NCBI genes (Maglott *et al.*, 2007) with PubMed articles. Gene2pubmed currently references 20 164 human and 27 322 mouse genes. Species-specific filtering results in a reduced dataset of around 300 000 articles for human and around 209 000 articles for mouse. Each article is therefore associated with a set of *NCBI* genes, as well as with a set of MeSH cell types.

We next construct two distinct tables, one for each species—human and mouse. A gene and a cell type are considered to co-occur if there is at least one article that is associated with the cell type in MEDLINE and with the gene in Gene2pubmed. For example, in the article with PubMed ID p = 1591006, we have indexed gene $g =$ ‘CD79B’ (from Gene2pubmed) and indexed MeSH cell types $c1 =$ ‘B-Lymphocytes’ and $c2 =$ ‘Hematopoietic Stem Cells’ (from MEDLINE), meaning that g co-occurs with both $c1$ and $c2$ in p. We construct per table where each gene is a row and each cell type is a column and the entry denotes a list of articles in which the gene co-occurs with the cell type.

The CellMeSH database statistics are as follows. For human, 3.8% of all possible (20 164 × 570) gene–cell pairs have non-zero counts, and around 300 000 PubMed articles each contain at least one pair. For mouse, 2.4% (27 322 × 570) gene–cell pairs have non-zero counts, and around 209 000 PubMed articles each containing at least one pair. The CellMeSH database (in SQL or Excel format) can be downloaded from our Github page (Mao and Zhang, 2021b).

2.2 Probabilistic query method

There are two major issues with using a literature-derived database. The first issue is publication bias. Some genes or cell types are studied much more than others and, consequently, there are more publications and thus more associations containing those genes or cell types. The second issue is noise in the gene–cell-type mapping. The CellMeSH database is inherently noisy, as it links genes and cell types at an article level, and the simple fact of an article mentioning a cell type and a gene together does not imply that the gene serves as a marker for the cell type. This leads to potentially spurious associations between genes and cell types.

First, we highlight how to address the issue of publication bias by applying TF-IDF (Term Frequency-Inverse Document Frequency) (Wikipedia, 2020c) which is a re-weighting method commonly used in Natural Language Processing (NLP) (Manning, 2008; Rajaraman, 2011), and by applying column normalization. Specifically let $w_C(g)$ denote the weight, which is the number of co-occurrences of gene g in the cell type C. Using TF-IDF transformation, the new weight is given by $w_C(g) \leftarrow w_C(g) \times \log \frac{N_C}{K_g}$ where N_C is the total number of available cell types in the database, and K_g is the total number of cell types with non-zero weights for gene g, i.e. $K_g = \sum_C 1_{C:w_C(g)>0}$. TF-IDF addresses the publication bias of genes since the transformation results in lower weights for common genes (since, for these genes, K_g is larger). After TF-IDF transformation, the weight is further adjusted by column normalization: $w_C(g) \leftarrow \frac{w_C(g)}{\sum_{g' \in C} w_C(g')}$. Column normalization addresses the publication bias of cell types since the transformation reduces the weight of genes occurring in common cell types.

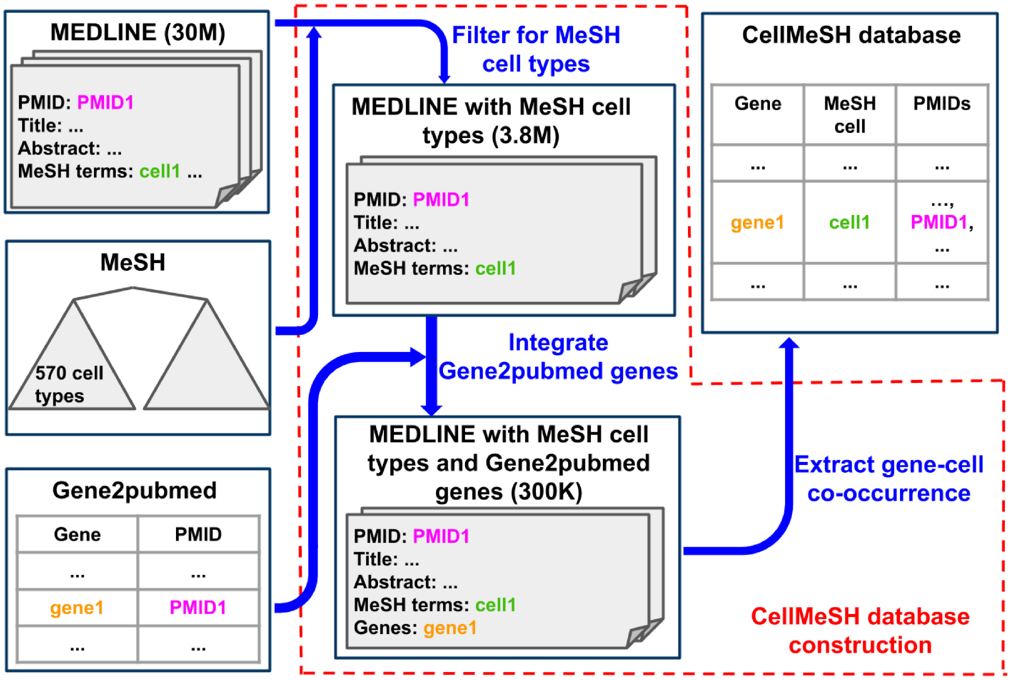


Fig. 2. CellMeSH database construction. Here, we illustrate the construction of the CellMeSH database for the human species. We start with the 30 million MEDLINE references, and keep the ones containing MeSH cell types (there are 3.8 million such references). We further filter away the references not having human genes in Gene2pubmed, after which 300 thousand MEDLINE references remain. Each remaining MEDLINE reference p contains several MeSH cell types $\{c\}$ and several genes $\{g\}$, and we append p to each (g, c) pair in the final CellMeSH database

We then query the weight-adjusted database using a probabilistic method, which is designed to address the issue of noise from spurious gene–cell associations. Our query method takes input of $w_C(g)$ which is the adjusted weight of gene g in cell type C. The method also takes input of a query Q which is a list of genes. The method outputs the database cell types sorted by their significance to the query.

Our probabilistic query method assumes the following generative model for the observed query data (based on which the inference is performed): (i) a cell type is first chosen (with a uniform prior probability). (ii) Associated with the cell type is a probability distribution on the genes given by $p(g|C)$. A natural model for $p(g|C)$ is to take it to be proportional to the weight $w_C(g)$. (iii) However, the previous model ensures that only genes with non-zero weight for the cell type will be present in the query—this need not be the case in our noisy dataset. To model this noise, we assume that with probability $1 - \alpha$ the gene is sampled randomly from the list of all genes, and with probability α it is sampled from the cell-type specific gene distribution (in experiments, α is fixed as 0.5).

We also denote the total number of genes as N_g and the total number of genes with non-zero weight in cell type C as K_C, i.e. $K_C = \sum_g 1_{g:w_C(g)>0}$. Thus, the probability of picking a gene from a cell type can be written as follows:

$$P(g|C) = \begin{cases} \alpha \cdot w_C(g) & \text{if } g \in Q \cap C \\ (1-\alpha)\dfrac{1}{N_g - K_C} & \text{if } g \in Q \cap \overline{C} \end{cases} \qquad (1)$$

We denote by $P(Q|C)$, the probability that the list of query genes is obtained from a particular cell type C. We utilize $P(g|C)$ as the probability that we see gene g in the query given the cell type is C. Assuming that each gene is sampled independently, we have $P(Q|C) = \prod_{g \in C} P(g|C)$.

To predict the cell type for a set of query genes (e.g. DEGs, denoted as Q) associated with a cluster, we look at a list of candidate cells (e.g. MeSH cell types) and for each candidate cell C we calculate its log-likelihood score $L(Q|C)$ as follows:

$$L(Q|C) = \log P(Q|C) = \sum_g \log P(g|C) \qquad (2)$$

Candidate cells are then sorted based on their log-likelihood scores. We can therefore utilize maximum likelihood estimation to predict the top-1 cell type $\hat{C}^*$ that maximizes our chance of seeing the query:

$$\hat{C}^* = \text{argmax}_C \log L(Q|C) \qquad (3)$$

2.3 Using prior knowledge

In practice, it is possible or even likely that some prior knowledge, for example about cell types expected in a given tissue of origin, is available for a set of queries. Such information is typically utilized in cell-level annotation methods. For example, Seurat (Butler *et al.*, 2018) needs an existing annotated gene expression profile that contains cell types that are similar to the query cells for label transfer, and Garnett (Pliner *et al.*, 2019) utilizes a manually curated cell-type marker-gene file which includes only a small number of cell types that are expected to cover the queries. Such information is not typically used in cluster-level annotation methods (Alavi *et al.*, 2018; Chen *et al.*, 2013), but CellMeSH is able to leverage optional prior knowledge to refine the search space for candidate cell types. Specifically, the probabilistic query method can be restricted to a subset of the CellMeSH database consisting of MeSH cell types that belong to one or multiple subtrees within the MeSH hierarchy. For instance, when annotating the PBMC dataset (Zheng *et al.*, 2017), we can limit search to cell types under 'Blood Cells' in the MeSH ontology (Supplementary File 3 Section 10).

3 Results

3.1 Cell-type annotation performance

We quantified the cell-type identification performance of CellMeSH for four scRNA-seq datasets with known cell types:

two Tabula Muris (TM) datasets (Consortium, 2018), the Mouse Cell Atlas (MCA) dataset (Han *et al.*, 2018) and the human Peripheral Blood Mononuclear Cells (PBMCs) dataset (Zheng *et al.*, 2017).

In our evaluation (Supplementary File 3 Section 2), we used clusters and reference annotations obtained in the original papers. For each cluster, we extracted the top $n = 50$ DEGs by 1-versus-rest gene expression ratio. These genes are assumed to be marker genes of the reference cell type and are used as a query input for CellMeSH. We then queried CellMeSH with marker genes for each cluster and visualized results using heatmaps that show how well the top-three retrieved candidate MeSH cell types agree with the reference cell type. To validate our results, we manually curated mappings between the reference cell types (called in original papers) and their correct MeSH cell types (Supplementary File 2).

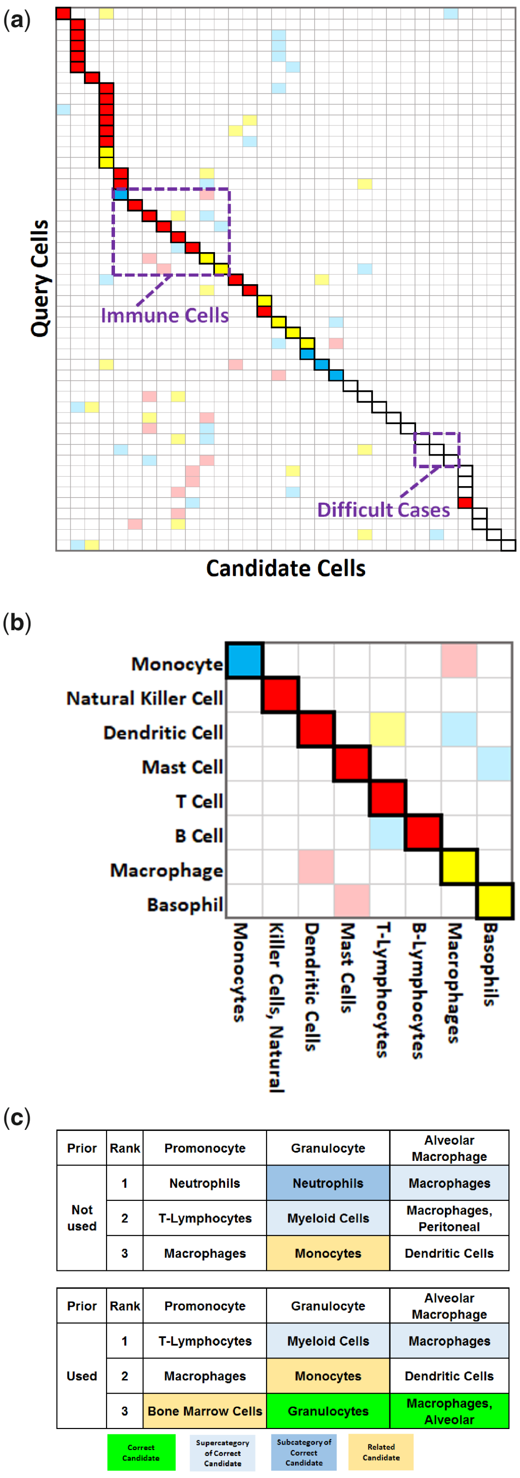
3.1.1 TM datasets

This dataset contains cells that were captured from 20 different tissues in 3-month-old mice and that were clustered into 99 annotated cell types. The dataset has two subsets, with cells captured by using a microfluidic-droplet method (denoted as the TM-Droplet dataset, containing 55 656 cells) or by fluorescence activated cell sorting (FACS) (denoted as the TM-FACS dataset, containing 44 949 cells). Here we focus on the TM-Droplet dataset (Fig. 3) but the results for TM-FACS are similar (Supplementary File 3 Fig. 4). Annotation results for the entire TM-Droplet dataset are summarized in the heatmap shown in Figure 3a (see Supplementary File 3 Fig. 3 for detailed cell-type names). The diagonal bordered boxes, indicating the expected annotations, are mostly filled with red, yellow or blue colors used to highlight the top three retrieved cell types, which clearly demonstrates the effective annotation ability of CellMeSH. To see this more clearly, in Figure 3b, we focus on the annotation heatmap for only the immune cell types. For that subset, the correct result is contained within top three candidates for all queries.

The bordered boxes forming a vertical stack in Figure 3a are the result several true cell types being mapped to the same MeSH cell term due to the limited resolution of the MeSH cell types. For example, both Luminal Epithelial Cell of Mammary Gland and Kidney Collecting Duct Epithelial Cell, etc. are mapped to 'Epithelial Cells'.

Bordered boxes without color in Figure 3a imply that the correct candidate may not exist due to a limit in the coverage of the MeSH cell types, or, more likely, that it is not within the top three retrieved results due to the noise in CellMeSH database. Still, even then, the CellMeSH query results provide useful insights into the true cell types, as illustrated in Figure 3c (prior not used). For instance, the query Promonocyte does not have an exact same MeSH term; the closest term we could manually match is 'Monocyte-Macrophage Precursor Cells' (see Supplementary File 2). For the query Granulocyte, the correct MeSH term 'Granulocytes' is rank-5 (data not shown) in the retrieved results. However, the top two results 'Neutrophils' and 'Myeloid Cells' are, respectively, the subcategory and supercategory of 'Granulocytes' in the MeSH tree. Similarly, for the query Alveolar Macrophage, the correct MeSH candidate 'Macrophages, Alveolar' actually is rank-5 (data not shown). However, the top rank result 'Macrophages' is also close as it is a supercategory of 'Macrophages, Alveolar'. Additional annotation results for uncolored bordered boxes in Figure 3a are shown in Supplementary File 4. Finally, we have conducted a quantitative analysis for the whole dataset showing that many top results are in fact closely related to the correct cell type in the MeSH tree (see Supplementary File 3 Table 2).

By leveraging prior knowledge, in particular restricting the search to cell types contained in the TM dataset, the CellMeSH annotation is further improved, as illustrated in Figure 3c (prior used). Specifically, the third result for the query Promonocyte now is 'Bone Marrow Cells'. Promonocytes are cells arising from a Monoblast [in Bone Marrow (Wikipedia, 2020a)] and developing into a Monocyte (Wikipedia, 2020b) and thus seem relevant to the query. Moreover, for the queries Granulocyte and Alveolar



Prior	Rank	Promonocyte	Granulocyte	Alveolar Macrophage
Not used	1	Neutrophils	Neutrophils	Macrophages
	2	T-Lymphocytes	Myeloid Cells	Macrophages, Peritoneal
	3	Macrophages	Monocytes	Dendritic Cells

Prior	Rank	Promonocyte	Granulocyte	Alveolar Macrophage
Used	1	T-Lymphocytes	Myeloid Cells	Macrophages
	2	Macrophages	Monocytes	Dendritic Cells
	3	Bone Marrow Cells	Granulocytes	Macrophages, Alveolar

🟩 Correct Candidate
Supercategory of Correct Candidate
Subcategory of Correct Candidate
Related Candidate

Fig. 3. Annotation results for Tabula Muris droplet dataset. (**a**) Annotation heatmap for queries of all cells. The *y*-axis represents the query cells $\{r\}$ and *x*-axis represents the candidate cells $\{c\}$. A border box for entry $(x = c, y = r)$ indicates *c* is the correct candidate for query *r*. The red, yellow or blue color indicates *c* has rank 1, rank 2 or rank 3 among retrieved results for *r*; the colors are shown lighter if *c* is not the correct candidate. (**b**) Annotation heatmap for queries of immune cells. The correct candidate cells are within the top three retrieved results for all queries. (**c**) The first table shows the top three results (prior not used) for several query cells corresponding to the uncolored border boxes in the heatmap. For instance, for the query Granulocyte, the correct candidate 'Granulocytes' is ranked fifth (data not shown) among the retrieved results. The top 2 results 'Neutrophils' and 'Myeloid Cells' are accurate to a certain extent because they are the subcategory and supercategory of 'Granulocytes' in the MeSH tree. In the second table, by using prior knowledge we obtain better results for these queries

Macrophage, the correct MeSH cell types are now contained within top three results.

3.1.2 MCA and PBMC datasets

These datasets show similar annotation performance to the TM datasets (see Supplementary File 3 Fig. 5 and Fig. 7, respectively). Again, bordered boxes on the diagonal are mostly filled with red, yellow or blue colors, indicating the top three retrieved cell types match the reference labels.

3.2 Comparison with cluster-level annotation methods

We compared CellMeSH to several existing cluster-level methods (Alavi *et al.*, 2018; Chen *et al.*, 2013; Fisher, 1935; Franzén *et al.*, 2019; Hänzelmann *et al.*, 2013; Zhang *et al.*, 2018b) (Supplementary File 3 Table 1) using top-k ($k = 1$, 3) accuracy (Supplementary File 3 Section 2.3) for the three mouse datasets (TM-Droplet, TM-FACS and MCA). We use the mouse datasets instead of the human PBMC dataset because they contain many more queries (51, 76 and 193 queries [Some queries (e.g. Cell in Cell Cycle) and are excluded as there are no matching candidates in any databases (e.g. CellMeSH, CellMarker, PanglaoDB, etc.)], respectively) than the PBMC dataset (only 10 queries), and therefore can show more reliable quantitative trends.

We queried CellMeSH with the previously extracted marker genes for each cluster, and calculated the top-k accuracy (percentage of queries for which the correct cell type is in the top-k results) for each dataset. We similarly queried existing methods and obtained top-k results for each dataset. The query results of existing methods could derive from a different ontology and therefore contain different cell-type names from the MeSH terms. In order to calculate the accuracy of these methods, we thus manually created mappings between the given query cell types and the candidate cell types from other ontologies, as summarized in Supplementary File 2.

3.2.1 Overall top-k accuracy gain

In Figure 4a and b, we first compare CellMeSH with two other web servers: Enrichr (Chen *et al.*, 2013) and scQuery (Alavi *et al.*, 2018) as their input and output formats are most similar to CellMeSH (Supplementary File 3 Table 1). We were unable to compare to the web servers of PanglaoDB (Franzén *et al.*, 2019), CellMarker (Zhang *et al.*, 2018b), and CellFinder (Stachelscheid *et al.*, 2013) in an automatic way. Instead, we downloaded the existing CellMarker (Zhang *et al.*, 2018b) database and queried it using the hypergeometric (or Fisher's exact test) (Fisher, 1935).

CellMeSH provides the most accurate results for all three mouse datasets. Specifically, in the TM-Droplet dataset, CellMeSH achieved top-1 accuracy of 58.8%, meaning that in 58.8% of queries, the first retrieved candidate cell type is correct. The top-1 accuracy is 15.7% higher than that of the second-best method, Enrichr. This is to be expected because the Enrichr cell types come from the Mouse Gene Atlas (MGA) database (Su *et al.*, 2004), which contains only 96 cell types. Besides, some of the MGA cell types (such as 'Heart', 'Kidney' and 'Stomach' etc.) actually refer to organs.

We find that CellMeSH has higher coverage and resolution than the other methods including Enrichr. For example, for query Classical Monocyte, while CellMeSH returns 'Monocytes' as the first candidate, there is no monocyte term covered in MGA and Enrichr returns its first result as 'Macrophage Bone Marrow 6 hr LPS'. For query Duct Epithelial Cell, while CellMeSH returns 'Epithelial Cells' as the first result, Enrichr returns the organ terms 'Bladder', 'Liver' and 'Stomach' as its top three results (see Supplementary File 5 for details). The top-3 accuracy of CellMeSH further increases to 88.2% (this implies that 88.2% of queries get at least one of the top three results correct), which is 31.3% higher than that of Enrichr.

CellMeSH also consistently outperforms other methods on the other two datasets. Its top-1 (or top-3) accuracy is 3.9% (or 11.9%) higher in the TM-FACS dataset, and 6.4% (or 22.7%) higher in the MCA dataset, than the second-best method, Enrichr.

3.2.2 Top-k accuracy gain from probabilistic method

Both the CellMeSH database and the probabilistic query method contribute to the overall top-k accuracy gains of CellMeSH. To isolate the contribution of the probabilistic query method to the overall CellMeSH performance, we compared it to the more established hypergeometric test (Fisher, 1935) and gene set variation analysis (GSVA) (Hänzelmann *et al.*, 2013) that are suggested in Diaz-Mejia *et al.* (2019), by querying the same CellMeSH database for the three mouse datasets. The details for hypergeometric test and GSVA are in Supplementary File 3 Section 2.4 and 2.5 respectively.

As shown in Figure 4c and d, the probabilistic method performs uniformly better than other methods. Compared to the best performance of GSVA and the hypergeometric test, the probabilistic query method has a top-1 accuracy gain of 13.7%, 6.6% and 7.3% in the TM-Droplet, TM-FACS and MCA datasets, respectively. The numbers for top-3 accuracy gain are 19.6%, 3.9% and 8.8%.

3.2.3 Top-k accuracy gain from CellMeSH database

To isolate the contribution of the CellMeSH database, here, we compare the performance of alternative databases obtained as follows.

We prepared gene–cell co-occurrence matrices by aggregating the cell-type marker-genes files from PanglaoDB (Franzén *et al.*, 2019) and CellMarker (Zhang *et al.*, 2018b), both of which are manually compiled from the literature. The resulting mouse gene–cell co-occurrence matrix of CellMarker has 7208 genes and 313 cell terms, where the matrix value for a particular gene–cell pair represents the number of records (i.e. publications) where they co-occur. The resulting human gene–cell co-occurrence matrix of CellMarker has 8973 genes and 364 cell terms. These matrices are mostly sparse, with $< 1\%$ positive counts, and maximum counts below 10. The resulting PanglaoDB database, a binary gene–cell matrix, has 4679 genes (for both mouse and human) and 178 cell types. There are 8230 (1%) non-zero entries. Details for CellMarker and PanglaoDB databases can be found in Supplementary File 3 Section 2.6 and 2.7 respectively.

We then compared the CellMeSH database to these two databases. We queried the CellMeSH and CellMarker databases using the probabilistic query method, since these databases are count-valued matrices, which can be handled effectively by the probabilistic query method, as illustrated in Figure 4c and d and in Supplementary File 3 Figure 8. For PanglaoDB, the query method is the hypergeometric test, since the database is essentially a binary matrix.

As Figure 4e and f illustrates, using the CellMeSH database achieves a higher accuracy than using the PanglaoDB and CellMarker databases. Compared to the best performance out of PanglaoDB and CellMarker databases, the CellMeSH database has top-1 accuracy gain of 21.6%, 3.9% and 0.6% in the TM-Droplet, TM-FACS and MCA datasets, respectively. The numbers for top-3 accuracy gain increase to 21.6%, 10.5% and 5.7%.

3.2.4 Impact of prior knowledge

Our evaluations in Figure 4a–f for CellMeSH do not use prior knowledge and the search space covers all of the 570 MeSH cell types. In contrast, many existing methods, especially the ones that predict cell types for single cells, leverage prior knowledge so that the reference gene expression profiles (Alavi *et al.*, 2018; Aran *et al.*, 2019; Butler *et al.*, 2018; Hou *et al.*, 2019; Kiselev *et al.*, 2018; Ma and Pellegrini, 2019; Tan and Cahan, 2019) or manually curated cell-type marker-gene files (Pliner *et al.*, 2019; Zhang *et al.*, 2019) contain a more constrained and related set of candidate cell types. CellMeSH can similarly use such prior knowledge to reduce the search space for better annotation accuracy. For example, we selected the correct MeSH cell types for the TM

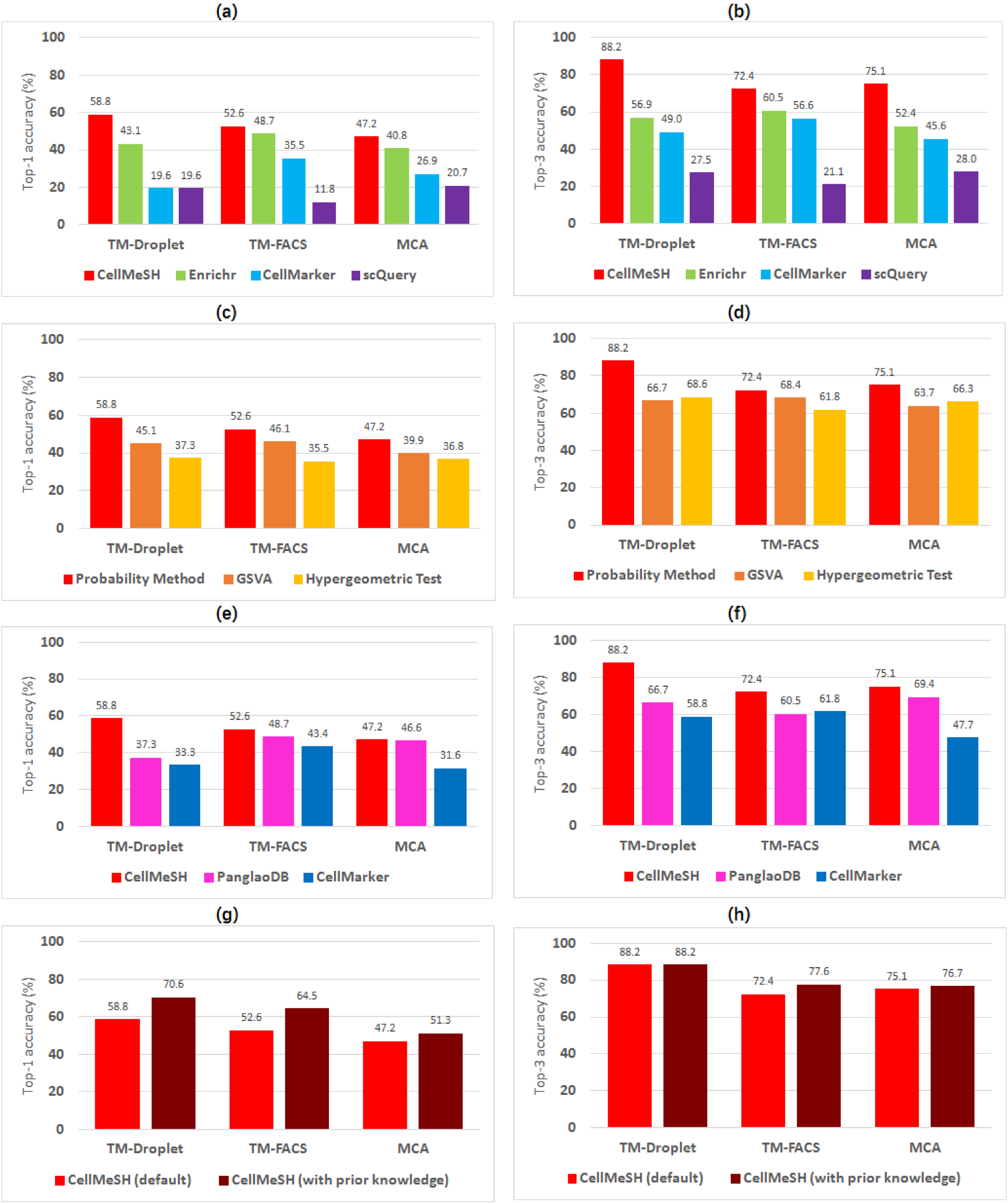


Fig. 4. Comparison of CellMeSH to other methods. Each bar plot has *y*-axis as the top-k ($k = 1$ or 3) accuracy (%) and is grouped by different mouse datasets, and for each group, we show the top-k accuracy of different methods. Top-k accuracy refers to the percentage of queries where one of the candidate cells among the top k retrieved cells is accurate. (**a**, **b**) Comparison of CellMeSH to other systems. The CellMarker database is queried using the hypergeometric test. (**c**, **d**) Comparison of the probabilistic query method to other query methods. We use the CellMeSH database with all query methods. (**e**, **f**) Comparison of the CellMeSH database to other databases. CellMeSH and CellMarker are both non-binary gene–cell matrices and therefore we use probabilistic method to query, whereas PanglaoDB is a binary gene–cell matrix and we use hypergeometric test to query. (**g**, **h**) Comparison of the default version of CellMeSH that uses a search space of all 570 MeSH cell types to CellMeSH with prior knowledge. Correct MeSH cell types and MeSH tree descendants from Tabula Muris dataset are used as prior information for TM-Droplet and TM-FACS queries, while correct MeSH cell types and MeSH tree descendants from MCA dataset are used as prior for MCA queries

datasets and their descendant cell types in MeSH ontology to be the prior knowledge for the TM-Droplet and TM-FACS queries. Similarly, we selected the correct MeSH cell types for the MCA dataset and their descendant cell types as the prior knowledge for the MCA queries. As shown in Figure 4g and h, utilizing prior knowledge improves Top-1 accuracies by 11.8%, 11.9% and 4.1% for TM-Droplet, TM-FACS and MCA, respectively. The gain is smaller for MCA because it has the largest number of queries, so that the search space is not reduced significantly (see Supplementary File 3 Section 10 for more detail).

3.2.5 Impact of gene number

Our evaluations so far have been using the top $n = 50$ DEGs as the marker genes for each query cell type, as the performance tends to peak around $n = 50$ for most of the methods (see Supplementary File 3 Figs. 9 to 11); A smaller number of genes may not provide sufficient information, and a larger number of genes may increase noise, both of which could result in degraded annotation performance. If we select the optimal number of genes for each method (e.g. different settings of the database and the query method), the CellMeSH database together with the probabilistic query method still consistently outperforms all other methods for all datasets (see Supplementary File 5).

3.2.6 Impact of cell population

CellMeSH suggests cell-type annotations for a cluster based on its DEGs. DEGs are obtained by comparing gene expression for cells in the cluster of interest to gene expression of cells in the remaining clusters. In principle, DEGs associated with a cluster can thus change depending on what other cells are present in a dataset. To test whether such context changes result in different cell-type assignments, we compared CellMeSH predictions for the TM-Droplet immune cells in the context of all other cells with the predictions when only the immune cells are present. The predictions are mostly consistent between the different contexts (see Supplementary File 3 Section 13). We hypothesize that if the DEGs are true markers, they should show up in most cases since our analysis looks at a large number (e.g. 50) of genes.

3.2.7 Impact of clustering resolution

The CellMeSH annotation can be affected by clustering resolution (i.e. whether a set of cells are considered as one cluster or multiple clusters). We have explored this issue in our developed web server UNCURL-App (Section 3.4) which enables end-to-end cluster-level cell type annotation that integrates CellMeSH to automatically assign cell types to clusters. We found that a human-in-the-loop reclustering (either to split a large cluster or to merge several small clusters) is necessary for further improved annotation [see UNCURL-App (Zhang *et al.*, 2020) for details].

3.3 Compare with cell-level annotation methods

Next, we compared CellMeSH with Seurat (label transfer) (Butler *et al.*, 2018) and SingleR (Aran *et al.*, 2019), two annotation methods that use existing annotated scRNA-seq (or bulk RNA-seq) datasets as a reference, and assign the cell types from existing datasets to query cells that have similar gene expressions (see Supplementary File 3 Table 1). For this comparison, we utilized the TM-Droplet as the query dataset, and the TM-FACS as the reference, since they share most cell types but are not identical, a common situation in practical applications. For CellMeSH, we grouped the TM-Droplet cells into clusters based on the reference annotation. For each cluster, CellMeSH assigned cell types by querying the database where only the TM-FACS MeSH labels were considered as priors. For Seurat and SingleR, we first individually label all of the TM-Droplet cells using the annotated gene–cell expression profile from TM-FACS. We then grouped the TM-Droplet cells into clusters based on their original annotation. The labels for a cluster are the labels for the cells in the cluster, sorted by number of assigned cells. For example, the top assigned label for a given cluster would be the label that occurred most frequently for the cells in that cluster. There are 43 queries that assigned cell types by all methods (Seurat, SingleR and CellMeSH). The results are summarized in Figure 5. All three methods have very similar top-1 accuracy. We note that although top-1 accuracy for Seurat and CellMeSH are the same, the queries with correct results are different. More importantly, even without prior knowledge of TM-FACS labels, default CellMeSH still reaches top-1 accuracy of 58.8%. An initial CellMeSH-based annotation without prior knowledge could thus in principle be used to select a dataset that is similar to the query dataset and that can then be used for improving the annotation. Please see Supplementary File 3 Section 11 for details.

3.4 CellMeSH web server and API

CellMeSH has a stand-alone web server (Zhang and Mao, 2021), which is able to take in a list of marker genes and an optional choice of known MeSH cells as prior knowledge, and returns a ranked list of predicted MeSH cell types, together with the supporting genes and PubMed articles for further reference (Fig. 1). The web server also provides options to use other databases (e.g. CellMarker) and query methods (e.g. GSVA, hypergeometric test).

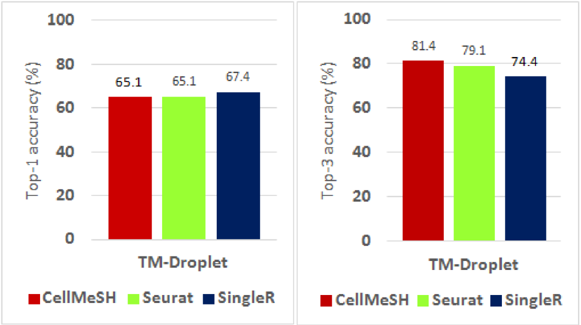


Fig. 5. Comparison of CellMeSH to cell-level annotation methods. To compare CellMeSH with Seurat and SingleR, we use the TM-Droplet dataset as the query. CellMeSH uses the TM-FACS labels as prior knowledge, and Seurat and SingleR use the TM-FACS annotated gene–cell expressions as the reference dataset. When comparing annotation results at cluster-level, all methods show comparable performance

We have open-sourced the CellMeSH database and the probabilistic query method (Mao and Zhang, 2021a) as a Python API to assist the community efforts on automating cell-type identification. We have utilized CellMeSH API to annotate the cluster-based Seurat approach (Satija *et al.*, 2015) for the PBMC dataset (see Supplementary File 3 Section 12). Specifically, here Seurat first takes input of a gene–cell expression matrix and outputs cell clusters. CellMeSH then annotates each cluster based on its DEGs produced by Seurat. This demonstrates CellMeSH's potential to complement existing tools and avoid manual cell assignment. We have also integrated CellMeSH API into our developed web server UNCURL-App (Zhang *et al.*, 2020) for interactive scRNA-seq data analysis, which combines data preprocessing, dimensionality reduction, clustering, differential expression analysis, cell-type annotation and interactive re-clustering into an online graphical user interface.

4 Discussion

We have developed CellMeSH, a method with accompanying web server and API to identify cell types directly from the literature, in order to make the scRNA-seq analysis more convenient.

CellMeSH is similar to gene set-enrichment methods (e.g. Enrichr, GSVA, Hypergeometric test) in terms of input and output formats, but several aspects make CellMeSH different from them. First, existing enrichment tools either focus on only query method (GSVA, Hypergeometric test) or prepare database in a limited way (Enrichr) by collecting the gene–ontology (e.g. diseases, tissues, cell types, etc.) relations manually from a small set of publications, which is difficult to scale. Second, existing enrichment tools assume the gene–ontology relations have no noise (e.g. no spurious gene–cell relation), whereas the CellMeSH query method is designed to handle spurious gene–cell associations, which is necessary as a database scales automatically and introduces noise (Fig. 4c and d). Finally, existing enrichment tools are not optimized for the scRNA-seq annotation problem to the same extent as CellMeSH. For instance, CellMeSH adopts a parameter of top 50 genes for its prediction, and it optionally takes prior knowledge (Fig. 1, Section 2.3) as input and explores the MeSH ontology hierarchy to reduce the search space in order for better prediction. CellMeSH has also been experimented on various scRNA-Seq datasets to show it is able to provide better annotations than using existing enrichment methods directly (Fig. 4a–d).

Experiments on both human and mouse scRNA-seq datasets demonstrate CellMeSH's superior cell-type identification performance. Nevertheless, there are still several limitations with CellMeSH.

In particular, the cell-type annotations provided by MeSH terms are somewhat coarse, and might not be enough to represent a comprehensive listing of all fine-grained cell types and subtypes present in model organisms such as human or mouse. Specifically, it is possible that the ground truth cell type (a subtype of CD4+ T cell) for a cluster does not exist in the MeSH ontology, and consequently

CellMeSH will not be able to predict that exact cell type but to provide a similar cell type assignment. This is an information limit issue that becomes a common problem for various tools with different prediction methods. For example, we have experimented and verified that given similar prior knowledge, CellMeSH and label transfer Seurat show similar annotation performance (see Fig. 5). To resolve this issue, it is helpful to resort to fine-grained gene–cell-type relationships from raw literature. Previously, the work of CellMarker approached this direction but it required manually efforts to survey the literature and pick up cell types and their marker genes. We believe CellMeSH is the first attempt to automate this, and have shown it actually contains richer information than CellMarker (Fig. 4e and f). As it is possible that CellMeSH still may not be able to provide an exact fine-grained cell-type, its predictions can be a useful guide for researchers to pick related scRNA-Seq experiment data, which may not appear in literature yet, for further analysis using cell-level annotation methods such as label transfer Seurat.

In addition, the CellMeSH database does not contain the information for a pair of gene and cell type regarding to whether they are associated in genomic or epigenomic or transcriptomic level, or whether the gene is upregulated or downregulated for the cell type. Such information is not available from the Gene2pubmed and MeSH indexings where CellMeSH database is built upon. The association only indicates whether a gene and a cell type are indexed together with Gene2pubmed and MeSH for publications. While such association contains spurious relations, such noisy signals should be small when we consider large number of publications. We also designed query algorithm to model the noises, and experiments show good annotation performance (Figs 3 and 4).

Moreover, for other species, even other model organisms, gene–cell information is limited due to a lack of indexed publications. However, the CellMeSH approach could in principle be generalized given a list of putative cell types.

To address these limitations, it is helpful to build a fully automated solution that picks up fine-grained cell types as well as gene–cell-type relationships directly from raw literature. This is an interesting direction and a challenging task, as such information are expressed in a rich way (e.g. T cell is also written as T-Lymphocytes), requires relationship classifications (e.g. genomic, epigenomic or transcriptomic level, up or down regulation) and scattered in different places (e.g. main text or supplementary tables), and the harvested data will be noisier than CellMeSH. More advanced techniques in natural language processing area can be a promising approach. Ideally, such an approach would enable the identification of new cell types and gene–cell-type relations in papers using unsupervised or semi-supervised named entity recognition and relation extraction techniques (Nadeau and Sekine, 2007; Yadav and Bethard, 2019).

There are also terms within the MeSH ontology that may be useful but are not under the 'Cell' heading, such as tissues, organs and diseases. Designing the query methods utilizing these information is another interesting future direction. For instance, we can refine our search scope if we know the tissue information of the query; or if such information is missing, we could provide them from an extended Cell/Tissue-MeSH database.

Acknowledgements

The authors thank Sumit Mukherjee for giving them many insightful suggestions and pointing them to many useful resources.

Author contributions

S.M., Y.Z., G.S. and S.K conceptualized the tool. S.M. and Y.Z. implemented the probabilistic method and database. Y.Z. implemented the webserver, S.M. and Y.Z. were involved in experiment and analysis. S.M., Y.Z., G.S. and S.K. wrote and edited the paper

Funding

This work was supported by National Institutes of Health awards R21CA246358 and R01HG009892 to G.S. This work was also supported by National Institutes of Health 1R01HG008164, National Science Foundation CCF 1651236 and NSF CIF 1703403 to S.K.

Conflict of Interest: G.S. is a co-founder and shareholder of Split Biosciences, a scRNA-seq company.

Data Availability

The data underlying this article are available at https://github.com/shunfumao/cellmesh.

References

Alavi,A. *et al.* (2018) scQuery: a web server for comparative analysis of single-cell RNA-seq data. Nature Communications, 9, 4768.

Andrews,S. (2010) Fastqc-a quality control tool for high throughput sequence data. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/ (15 May 2020, date last accessed).

Aran,D. *et al.* (2019) Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nat. Immunol.*, **20**, 163–172.

Baron,M. *et al.* (2016) A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Syst.*, **3**, 346–360.e4.

Bray,N.L. *et al.* (2016) Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.*, **34**, 525–527.

Butler,A. *et al.* (2018) Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. Biotechnol.*, **36**, 411–420.

Cao,J. *et al.* (2017) Comprehensive single-cell transcriptional profiling of a multicellular organism. *Science*, **357**, 661–667.

Chen,E.Y. *et al.* (2013) Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. *BMC Bioinformatics*, **14**, 128.

Consortium,T.T.M. (2018) Single-cell transcriptomics of 20 mouse organs creates a tabula muris. *Nature*, **562**, 367–372.

Diaz-Mejia,J.J. *et al.* (2019) Evaluation of methods to assign cell type labels to cell clusters from single-cell RNA-sequencing data. F1000Research, 8, 296. https://doi.org/10.12688/f1000research.18490.3.

Dobin,A. *et al.* (2012) Star: ultrafast universal rna-seq aligner. Bioinformatics. http://bioinformatics.oxfordjournals.org/content/early/2012/10/25/bioinformatics.bts635.abstract.

Dobin,A. *et al.* (2013) Star: ultrafast universal RNA-seq aligner. *Bioinformatics*, **29**, 15–21.

Fisher,R.A. (1935) The logic of inductive inference. *J. R. Stat. Soc.*, **98**, 39.

Franzén,O. *et al.* (2019) PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data. *Database*, **2019**, baz046.

Grün,D. and van Oudenaarden,A. (2015) Design and analysis of single-cell sequencing experiments. *Cell*, **163**, 799–810.

Han,X. *et al.* (2018) Mapping the mouse cell atlas by microwell-seq. *Cell*, **172**, 1091–1107.e17.

Hänzelmann,S. *et al.* (2013) GSVA: gene set variation analysis for microarray and RNA-seq data. *BMC Bioinformatics*, **14**, 7.

Hou,R. *et al.* (2019) scMatch: a single-cell gene expression profile annotation tool using reference datasets. *Bioinformatics*, **35**, 4688–4695.

Jaitin,D.A. *et al.* (2014) Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types. *Science*, **343**, 776–779.

Kim,D. *et al.* (2013) TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol.*, **14**, R36.

Kiselev,V.Y. *et al.* (2017) SC3: consensus clustering of single-cell RNA-seq data. *Nat. Methods*, **14**, 483–486.

Kiselev,V.Y. *et al.* (2018) scmap: projection of single-cell RNA-seq data across data sets. *Nat. Methods*, **15**, 359–362.

Klein,A.M. *et al.* (2015) Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, **161**, 1187–1201.

Lähnemann,D. *et al.* (2020) Eleven grand challenges in single-cell data science. *Genome Biol.*, **21**, 31.

Li,B. and Dewey,C.N. (2011) Rsem: accurate transcript quantification from rna-seq data with or without a reference genome. *BMC Bioinformatics*, **12**, 323.

Lin,P. *et al.* (2017) CIDR: ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol.*, **18**, 59.

Ma,F. and Pellegrini,M. (2019) ACTINN: automated identification of cell types in single cell RNA sequencing. *Bioinformatics*, **36**, 533–538.

Maglott,D. *et al.* (Jan. 2007) Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Res.*, **35**, D26–D31.

Manning,C. (2008) *Introduction to Information Retrieval*. Cambridge University Press, New York.

Mao,S. and Zhang,Y. (2021a) Cellmesh api. https://github.com/shunfumao/cellmesh (4 December 2021, date last accessed).

Mao,S. and Zhang,Y. (2021b) Cellmesh database download. https://github.com/shunfumao/cellmesh/tree/master/cellmesh/db_download (4 December 2021, date last accessed).

MEDLINE. (2015) MEDLINE Indexing Online Training Course. https://www.nlm.nih.gov/bsd/indexing/training/USE_010.html (09 April 2019, date last accessed).

MESH. (2019) Medical Subject Headings. https://www.nlm.nih.gov/mesh/meshhome.html (09 April 2019, date last accessed).

Mukherjee,S. *et al.* (2018) Scalable preprocessing for sparse scRNA-seq data exploiting prior knowledge. *Bioinformatics*, **34**, i124–i132.

Nadeau,D. and Sekine,S. (2007) A survey of named entity recognition and classification. *Int. J. Ling. Lang. Resour*., **30**, 3–26.

Orr Ashenberg,D.S. and Institute,K.G.B. (2019) Workshop for analysis of single-cell RNA-seq data. https://broadinstitute.github.io/2019_scWorkshop/ (15 May 2020, date last accessed).

Pierson,E. and Yau,C. (2015) ZIFA: dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol*., **16**, 241.

Pliner,H.A. *et al.* (2019) Supervised classification enables rapid annotation of cell atlases. Nature Methods, 16, 983–986. https://doi.org/10.1038/s41592-019-0535-3.

Rajaraman,A. (2011) *Mining of Massive Datasets*. Cambridge University Press, Cambridge.

Rosenberg,A.B. *et al.* (2018) Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science*, **360**, eaam8999-182.

Satija,R. *et al.* (2015) Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol*., **33**, 495–502.

Setty,M. *et al.* (2016) Wishbone identifies bifurcating developmental trajectories from single-cell data. *Nat. Biotechnol*., **34**, 637–645.

Shin,J. *et al.* (2015) Single-cell RNA-seq with waterfall reveals molecular cascades underlying adult neurogenesis. *Cell Stem Cell*, **17**, 360–372.

Soneson,C. and Robinson,M.D. (2018) Bias, robustness and scalability in single-cell differential expression analysis. *Nat. Methods*, **15**, 255–261.

Stachelscheid,H. *et al.* (2013) CellFinder: a cell data repository. *Nucleic Acids Res*., **42**, D950–D958.

Su,A.I. *et al.* (2004) A gene atlas of the mouse and human protein-encoding transcriptomes. *Proc. Natl. Acad. Sci. USA*, **101**, 6062–6067.

Sun,Z. *et al.* (2019) A Bayesian mixture model for clustering droplet-based single-cell transcriptomic data from population studies. *Nat. Commun*., **10**, 1649–2041.

Tan,Y. and Cahan,P. (2019) SingleCellNet: a computational tool to classify single cell RNA-seq data across platforms and across species. *Cell Syst*., **9**, 207–213.e2.

Tang,F. *et al.* (2009) mRNA-seq whole-transcriptome analysis of a single cell. *Nat. Methods*, **6**, 377–382.

Tasic,B. *et al.* (2016) Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nat. Neurosci*., **19**, 335–346.

Trapnell,C. *et al.* (2014) The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol*., **32**, 381–386.

Usoskin,D. *et al.* (2014) Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing. *Nat. Neurosci*., **18**, 145–153.

Wang,B. *et al.* (2017) Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nat. Methods*, **14**, 414–416.

Welch,J.D. *et al.* (2016) SLICER: inferring branched, nonlinear cellular trajectories from single cell RNA-seq data. *Genome Biol*., **17**, 106.

Wikipedia. (2020a) Monoblast–wikipedia. https://en.wikipedia.org/wiki/Monoblast (20 January 2020, date last accessed).

Wikipedia. (2020b) Promonocyte–wikipedia. https://en.wikipedia.org/wiki/Promonocyte (20 January 2020, date last accessed).

Wikipedia. (2020c) tf-idf–wikipedia, https://en.wikipedia.org/wiki/Tf–idf (21 February 2020, date last accessed).

Wolf,F.A. *et al.* (2018) SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol*., **19**, 15.

Yadav,V. and Bethard S. (2019) A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2145–2158.

Zeisel,A. *et al.* (2015) Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, **347**, 1138–1142.

Zhang,A.W. *et al.* (2019) Probabilistic cell-type assignment of single-cell RNA-seq for tumor microenvironment profiling. *Nat. Methods*, **16**, 1007–1015.

Zhang,J.M. *et al.* (2018a) An interpretable framework for clustering single-cell RNA-Seq datasets. *BMC Bioinformatics*, **19**, 93.

Zhang,X. *et al.* (2018b) CellMarker: a manually curated resource of cell markers in human and mouse. *Nucleic Acids Res*., **47**, D721–D728.

Zhang,Y. and Mao,S. (2021) Cellmesh web server. https://uncurl.cs.washington.edu/db_query (12 April 2021, date last accessed).

Zhang,Y. *et al.* (2020) UNCURL-app: interactive database-driven analysis of single cell RNA sequencing data, doi: 10.1101/2020.04.

Zheng,G.X.Y. *et al.* (2017) Massively parallel digital transcriptional profiling of single cells. *Nat. Commun*., **8**, 14049.