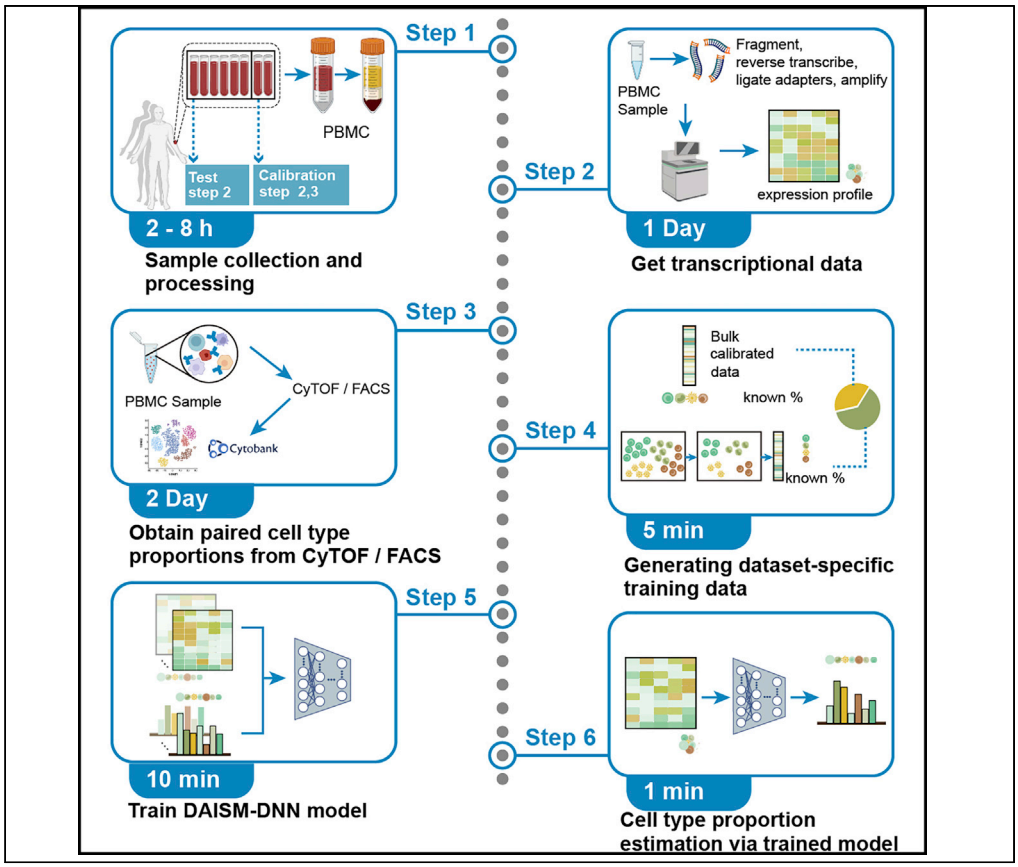
STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol to estimate cell type proportions from bulk RNA-seq using DAISM-DNN^{XMBD}



Computational protocols for cell type deconvolution from bulk RNA-seq data have been used to understand cellular heterogeneity in disease-related samples, but their performance can be impacted by batch effect among datasets. Here, we present a DAISM-DNN protocol to achieve robust cell type proportion estimation on the target dataset. We describe the preparation of calibrated samples from human blood samples. We then detail steps to train a dataset-specific deep neural network (DNN) model and cell type proportion estimation using the trained model.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Yating Lin, Shangze Wu, Xu Xiao, ..., Lei Zhang, Jiahuai Han, Rongshan Yu

zhanglei@xmu.edu.cn (L.Z.)
jhan@xmu.edu.cn (J.H.)
rsyu@xmu.edu.cn (R.Y.)

Highlights

A protocol for accurate cell type deconvolution with data-driven DNN-based approach

Obtain expression and cell proportions from calibrated samples

DAISM-DNN model training including parameter tuning and data formatting

Trained model can be applied to other biomedical experiments under the same conditions

Lin et al., STAR Protocols 3, 101587
September 16, 2022
https://doi.org/10.1016/j.xpro.2022.101587

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol to estimate cell type proportions from bulk RNA-seq using DAISM-DNN^{XMBD}

Yating Lin,[1] Shangze Wu,[1] Xu Xiao,[1,2] Jingbo Zhao,[3] Minshu Wang,[2,4] Haojun Li,[1] Kejia Wang,[4] Minwei Zhang,[5] Frank Zheng,[3] Wenxian Yang,[6] Lei Zhang,[7,9,*] Jiahuai Han,[8,*] and Rongshan Yu[1,2,6,10,*]

[1]School of Informatics, Xiamen University, Xiamen 361005, China

[2]National Institute for Data Science in Health and Medicine, Xiamen University, Xiamen 361005, China

[3]Amoy Diagnostics, Xiamen 361000, China

[4]School of Medicine, Xiamen University, Xiamen 361102, China

[5]Department of Critical Care Medicine, The First Affiliated Hospital of Xiamen University, Xiamen 361003, China

[6]Aginome Scientific, Xiamen 361005, China

[7]School of Life Science, Xiamen University, Xiamen 361102, China

[8]Research Unit of Cellular Stress of CAMS, Cancer Research Center of Xiamen University, School of Medicine, Xiamen University, Xiamen 361102, China

[9]Technical contact

[10]Lead contact

*Correspondence: zhanglei@xmu.edu.cn (L.Z.), jhan@xmu.edu.cn (J.H.), rsyu@xmu.edu.cn (R.Y.)
https://doi.org/10.1016/j.xpro.2022.101587

SUMMARY

Computational protocols for cell type deconvolution from bulk RNA-seq data have been used to understand cellular heterogeneity in disease-related samples, but their performance can be impacted by batch effect among datasets. Here, we present a DAISM-DNN protocol to achieve robust cell type proportion estimation on the target dataset. We describe the preparation of calibrated samples from human blood samples. We then detail steps to train a dataset-specific deep neural network (DNN) model and cell type proportion estimation using the trained model.

For complete details on the use and execution of this protocol, please refer to Lin et al. (2022).

BEFORE YOU BEGIN

Large-scale expression profiling of clinical samples has become feasible in routine clinical settings. However, these methods, such as high-throughput RNA-seq, only measure the average expression of genes from the heterogeneous samples in their entirety. As a result, they do not provide detailed information on the cellular compositions of the samples. To leverage the wealth of existing and clinically annotated bulk data, e.g., in The Cancer Genome Atlas (TCGA) and the International Cancer Genome Consortium (ICGC), there has been strong interest in computational "deconvolution" algorithms that can estimate individual cell type abundance from bulk tissue transcriptomic profiles.

Signature-based deconvolution methods derive an optimal dissection of original samples based on a set of pre-identified cell-type-specific expression patterns. However, the deconvolution accuracy would be plagued by technical bias when applied to samples profiled using different platforms from those of signature matrix (Vallania et al., 2018). Enrichment- (or marker-) based methods, which assign a per-cell-type score that can be used to compare the prevalence of a specific cell type across samples. However, the per-cell-type scores cannot be used for comparison across cell types as these values in general do not reflect the absolute abundances of different cells from a sample. Moreover, enrichment-based methods can sensitively distinguish between "coarse-grained" cell types (e.g., B

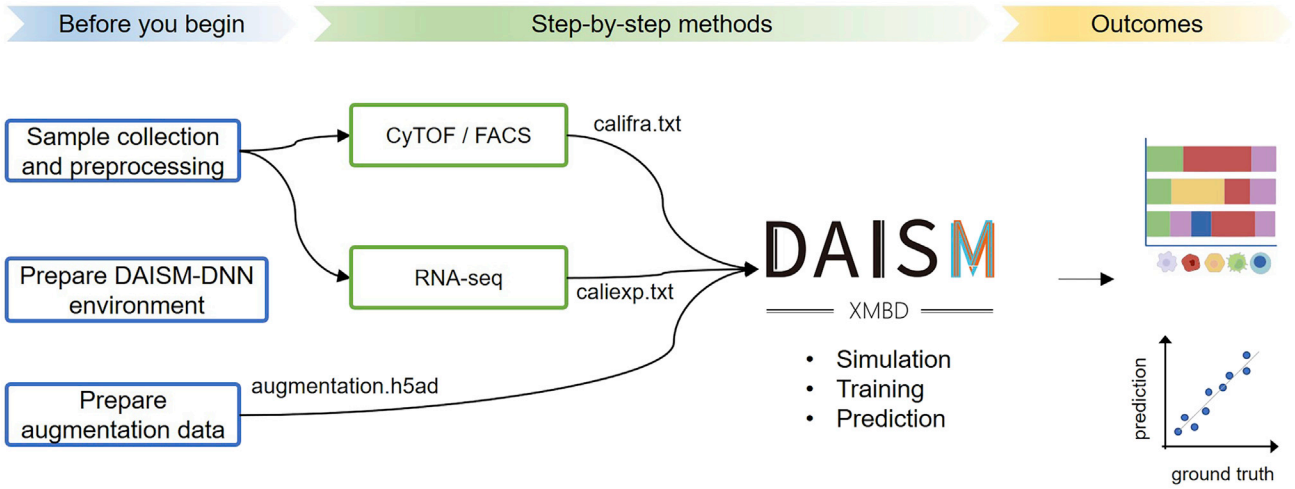


Figure 1. Overview of DAISM-DNN protocol

versus T cells), but often have low specificity in discriminating between "fine-grained" cell types (e.g., sub-populations of T cells) (Sturm et al., 2019).

Recently, the development of deep neural networks (DNNs) has provided the computational power to resolve complex biological problems using data-driven approaches with the vast trove of data available from the biomedical research community. However, it is still challenging for a DNN-based algorithm to deliver consistent performance under different experimental conditions due to statistical bias between training and actual data (Tegner and Gomez-Cabrero, 2022).

We propose the DAISM-DNN^{XMBD} (XMBD: Xiamen Big Data, a biomedicalopen software initiative in the National Institute for Data Science in Health and Medicine, Xiamen University, China) protocol for cell type deconvolution from bulk RNA-seq data. By training a dataset-specific deep neural network from a certain amount of calibration data drawn from the same experiment conditions as the target dataset, DAISM-DNN effectively medicates problems of source-target data mismatch problems in existing deconvolution methods, and produces highly accurate and reliable cell type abundance from bulk RNA-seq data. Note that the trained DAISM-DNN model can also be used across multiple biomedical experiments as long as these experiments are conducted under the same experimental conditions as those used in generating the calibration data.

To properly execute the pipelines of DAISM-DNN (Figure 1), in the before you begin section, we start from collecting peripheral blood mononuclear cells (PBMC) from human blood samples and preprocessing for the following expression profiling and cell proportion measurement. The protocol below provides information about preparing calibration samples for using DAISM-DNN on PBMCs samples. The same protocol can be extended for other sample types such as whole-blood samples or frozen tissue samples.

Institutional permissions

The studies described in this protocol received IRB approval by The First Affiliated Hospital of Xiamen University.

PBMC isolation and storage

Timing: 1 h, it takes longer if more than 6 samples are processed at the same time

Note: We recommend processing samples within 6 h from the collection.

1. Collect 10 mL peripheral blood of each sample in vacuum blood tubes.

   Note: Heparin or citrate is recommended as anticoagulant, EDTA anticoagulated PBMC samples may require additional washing steps.

2. Using aseptic technique, transfer anticoagulant-treated blood from each blood collection tube into 50 mL sterile conical tubes.
3. Dilute blood with an equal volume of PBS (room temperature (25°C), 1:1 dilution). Then mix the blood and medium by gently inverting the tube or pipetting up and down.
4. Gently layer the diluted blood onto 20 mL Ficoll-Paque Plus in a fresh 50 mL conical tube. Slowly dispense the first 5 mL of blood dropwise to avoid disturbing the separation medium and blood interface.
5. Centrifuge at 400 × *g* at room temperature for 20 min with the BRAKE OFF, to prevent disrupting the density gradient during deceleration.
6. Collect the PBMC layer at the interface between plasma and Ficoll-Paque Plus and transfer it into a new 15 mL conical tube (troubleshooting problem 1).
7. Wash the cell suspension with 15 mL of prewarmed serum-free RPMI 1640 medium and centrifuge at 300 × *g* at RT (25°C) for 5 min.
8. Discard the supernatant and flick the tube to detach the cell pellet. Repeat step 7.
9. Resuspend the cell pellet using 2 mL prewarmed serum-free RPMI 1640 medium.
10. Count cells and check the viability using 0.4% Trypan Blue Stain (Thermo Fisher Scientific) with an automated cell counter or manually using a Neubauer hemocytometer (or an alternative cell counter).
11. Centrifuge at 300 × *g* at RT for 5 min to pellet the cells.

⚠ CRITICAL: If fresh PBMCs are planned to be used immediately, please skip steps 12–21.

12. Discard the supernatant and re-suspend the pellet in cold freezing medium to obtain a cell density of 4–6 × 10^6 cells/mL.
13. Transfer the cell suspension into a 1 mL cryovial and store at −80°C. When longer storage (longer than 7 days) is planned, transfer the sample into −140°C or liquid nitrogen between 1 to 7 days after storage at −80°C.

   Note: It is recommended to use the samples within one year.

PBMC thawing

Timing: approximately 1 h, it takes longer if more than 6 samples are thawed at the same time

14. Warm complete RPMI 1640 medium with 10% FBS and 25 U/mL Benzonase for PBMCs in a 37°C water bath.

   Note: Benzonase treatment can reduce the viscosity and background by removing free DNA from lysed cells.

15. Remove PBMC samples from liquid nitrogen, and keep them in the Thermo-Flask™ (Thermo Fisher Scientific) with liquid nitrogen or on dry ice.
16. Thaw frozen vials in 37°C water bath for 2–3 min, and remove the tube once the PBMC sample just thawed.
17. Add 1 mL of warm complete RPMI 1640 medium to the cryovial in a dropwise manner. Then transfer cells to an appropriately labeled 15 mL centrifuge tube and gently pipet up and down to mix cells.

18. Centrifuge cells at 300 × *g* for 5 min at room temperature.
19. Remove supernatant and resuspend the cell pellet with 1 mL of warm complete RPMI 1640 medium.

Optional: Filter cells through a 70 μm cell strainer if you observe any clumps.

20. Add 9 mL warm complete RPMI 1640 medium to the tube and gently pipet up and down to mix cells.
21. Centrifuge at 300 × *g* for 5 min at room temperature.
22. Remove supernatant and resuspend the cell pellet with 1 mL of warm complete RPMI 1640 medium.
23. Count cells by adding 10 μL cells to 990 μL DPBS utilizing a hemocytometer (for example Novocyte Flow cytometer or an equivalent equipment).
24. For calibration samples, adjust the cell concentration to 1–3 × 10^6 cells/mL with warm complete RPMI 1640 medium for flow (or mass) cytometry or PBS for RNA purification. Transferred cells to new 5 mL polystyrene round-bottom tubes (flow or mass cytometry) or 1.5 mL tubes (RNA-Seq).

Note: Pipet the cell suspension and make sure it's well dispersed before transferring cells.

a. For flow (or mass) cytometry, place the 5 mL polystyrene round-bottom tubes in a 37°C CO_2 incubator for 15 min before staining.
b. For RNA purification, centrifuge at 300 × *g* for 5 min at room temperature.
   i. Discard the supernatant of the rest cell suspension.
   ii. The PBMC pellet can either be snap frozen in liquid nitrogen and stored at –80°C for later RNA purification or used directly in an RNA purification procedure using the RNeasy Mini Kit (1 × 10^6–1 × 10^7 cells).

Note: Make sure that supernatant is removed completely to avoid interference with the next steps of RNA isolation.

Note: We also recommend thawing no more than 12 PBMC samples at once to minimize the time thawed PBMCs are left at room temperature.

Prepare DAISM-DNN environment

Timing: 10 min

The DAISM-DNN package for Python is available through the following public repository: https://github.com/xmuyulab/DAISM-XMBD. The requirements of hardware are dataset dependent, the demo data provided with the DAISM-DNN source code on GitHub require a peak of 5.2 GB of random-access memory (RAM) during the DNN training process, so a computer with 8 GB of RAM should be able to run it smoothly.

25. DAISM-DNN is functional on all operating systems (Linux, Windows and Mac OSX) with Python 3. Python version ≥ 3.7 is recommended. Python can be downloaded from https://www.python.org/downloads/. Once the Python environment has been set up, DAISM-DNN can be easily installed via pip:

```
$ pip install daism
```

Optional: To better manage the development environment and avoid incompatible versions of dependencies required by different projects, we recommend using a virtual environment or

a docker container to run DAISM-DNN. The steps 26 and 27 under 'prepare DAISM-DNN environment' refer to preparing DAISM-DNN environment via conda virtual environment and docker container respectively. Only one of them needs to be implemented.

26. This step refers to creating a conda virtual environment and install daism package via pip or conda. A simplified strategy to run Python on any operating systems is to use Anaconda.
    a. Anaconda can be downloaded from https://www.anaconda.com/products/individual according to individual computer specifications.
    b. It is recommended to create a new conda environment:

```
$ conda create -n daism python=3.7
```

    c. Activate this environment:

```
$ conda activate daism
```

    d. Run the following command to install daism via pip or conda:
        i. For pip:

```
$ pip install daism
```

        ii. For conda:

```
$ conda install daism -c zoelin1130
```

Note: All package dependencies should be handled automatically when installing with pip or conda.

27. This step refers to building a docker container to run DAISM-DNN. We provide a docker image with DAISM-DNN installed: https://hub.docker.com/r/zoelin1130/daism.
    a. Please follow the instructions on https://docs.docker.com/engine/install/ to install Docker according to individual computer specifications. More details on using Docker can be found in https://docs.docker.com/get-started/.
    b. Once docker is installed, open the console and pull the docker image:

```
$ docker pull zoelin1130/daism:latest
```

    c. Create a container if GPUs are available to speed up computation:

```
$ docker run –gpus all -i -t –name run_daism -v example/:/workspace/example/ zoelin1130/
daism:latest /bin/bash
```

Optional: If GPU is not available on your device, create a container utilizing CPU:

```
$ docker run -i -t –name run_daism -v example/:/workspace/example/ zoelin1130/daism:latest
/bin/bash
```

Note: "run_daism" is your container name. It is strongly recommended to add -v parameter for implementing data and scripts mounting: mount the local volume "example" (from your machine) to "/workspace/example/ (to your container)" instead of directly copy them into the container.

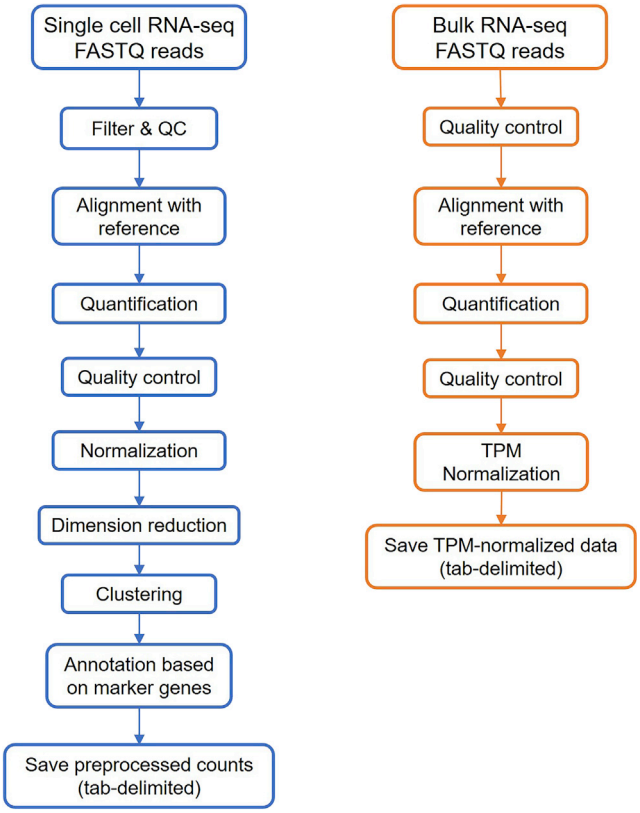


Figure 2. Schematic of preprocessing of single cell RNA-seq and bulk RNA-seq of purified cells

Prepare your augmentation transcriptional data

Timing: varied

To provide sufficient training data for training DAISM-DNN, additional training samples are generated by computing weighted combinations of expressions of calibration samples and simulated pseudo-bulk samples. In this study, we use single cell or purified bulk-RNA samples as augmentation data to create pseudo-bulk samples.

In all cases, the augmentation transcriptome output requires to be in the format of a matrix composed of columns associated to sample ID and rows to gene symbols. In our original study, we used a manually annotated dataset of single cell RNA-seq of PBMCs from 10× Genomics website, "8k PBMCs from a healthy donor" which is denoted as PBMC8k. We also used 1,533 RNA-seq samples of purified cells collected from various studies as augmentation data in case the calibration samples were profiled on high throughput sequencing. The cell type and GEO accession number of each RNA samples were listed on Data S1. These processed augmentation datasets are available at https://doi.org/10.17632/ysjwjvpnh3.1. Please note that steps 28 and 29 under 'prepare your augmentation transcriptional data' can be carried out independently as they refer to preprocessing of single cell RNA-seq data and bulk RNA-seq data of purified cells respectively (Figure 2).

Considering we have expression matrix and corresponding cell type annotation information of each sample, anndata (Wolf et al., 2018) was used to handle these annotated data matrices in memory and on disk. It can also be compatible with h5ad files. Depending on the h5ad format of augmentation in use, such a cell-type specific matrix can be obtained as following:

28. In case in which single cell RNA-seq data were used for augmentation.
    a. Align the raw single cell RNA-seq reads to the GRCh38 reference genome and quantify by Cell Ranger (Zheng et al., 2017).
    b. Use Seurat (Butler et al., 2018) to process the resulting expression matrix. The official Seurat tutorial is available at: https://satijalab.org/seurat/articles/pbmc3k_tutorial.html.
        i. First, exclude the cells with less than 500 genes or greater than 10% mitochondrial RNA content and genes expressed in less than five cells from analysis.
        ii. Then, exclude the cells with abnormally high gene counts which were considered as cell doublets from further analysis. The raw unique molecular identifier (UMI) counts were log-normalized, and the top 2,000 highly variable genes were called based on the average expression (between 0.0125 and 3) and average dispersion (>0.5).
        iii. Perform principal component analysis on the highly variable genes to further reduce the dimensionality of the data.
        iv. Finally, clusters were identified using the shared nearest neighbor-based clustering algorithm on the base of the first 20 principal components with an appropriate resolution.
        v. The identified clusters were annotated based on canonical marker expression patterns consistent with known immune cell types. The marker genes were obtained from the Cell Marker (Zhang et al., 2018) database for the target cell types in peripheral blood. Cell types were identified manually by checking if the respective marker genes were highly differentially expressed in each cluster.

    Note: The clusters without high expression on the selected marker genes or with high expression on the marker genes of other cell types were grouped into the "unknown" type.

    c. Save the preprocessed counts data as a tab-delimited file with gene symbols in row and sample names in column.
29. In case in which RNA-seq data of purified cells were used for augmentation (See "sequence data processing" for more details).
    a. Download the raw FASTQ reads from the NCBI website. The cell type and GEO accession number of each RNA samples were listed on Data S1.
    b. Use fastp (Chen et al., 2018) to perform quality control of FASTQ reads.
    c. Perform transcription and gene-level expression quantification using Salmon (Patro et al., 2017) with GRCh38 after quality control of FASTQ reads.

    Note: Users can choose different reference genome from GRCh38, but the version of the reference genome for calibration samples and augmentation data should preferably be consistent.

    d. Get transcripts per million (TPM) normalization results from salmon output. Merge the quantification results from different samples via R package tximport (Soneson et al., 2016).

    Note: If the samples are from different study, users should take the intersection of gene symbols of all expression profiles.

    e. Get reads count results from salmon output. Compute the total reads count for each sample. Remove the samples those total reads count is less than 1 million.
    f. Save TPM-normalized expression data as a tab-delimited file with gene symbols in row and sample names in column.
30. A comma-separated annotation table with two columns, sample names and annotated cell type, is also needed.
31. On the Shell Console, users can convert tab-delimited matrix files and annotation table to anndata format and save as a h5ad file via create_h5ad.py, which is available through the DAISM-DNN repository https://github.com/xmuyulab/DAISM-XMBD, as following:

```
$ python creat_h5ad.py -anno annotation_table.csv -exp expression.txt -outdir ./ -prefix
purified
```

Where -anno defines the comma-separated annotation table; -exp: the tab-delimited expression profile; -outdir: the folder where the output files are stored; -prefix: output filename prefix.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Biological samples		
Human peripheral blood sample (sex, age as required par study)	Any supplier	N/A
Chemicals, peptides, and recombinant proteins		
Ficoll-Paque Plus	GE Healthcare Life Sciences	Cat# GE17-1440-02
Phosphate-Buffered Saline w/o Ca & Mg (PBS)	Mediatech	Cat# 21040CV
RPMI 1640 (1×) Medium	Gibco	Cat# 11875093
0.4% Trypan Blue Solution	Thermo Fisher Scientific	Cat# T10282
Fetal Bovine Serum, heat inactivated, qualified (FBS)	Gibco	Cat# 10438026
Benzonase®	Thermo Fisher Scientific	Cat# C973K71
DPBS w/o Ca/Mg (DPBS)	Costar	Cat# 21-031-CM
Critical commercial assays		
RNeasy Mini Kit	QIAGEN	Cat# 74104
Deposited data		
Single cell RNA-seq (PBMC8k)	10× Genomics	https://support.10xgenomics.com/single-cell-gene-expression/datasets
augmentation data for DAISM	Lin et al., (2022); Mendeley Data	https://doi.org/10.17632/ysjwjvpnh3.1
Dataset SDY67 (both flow cytometry data and RNA-Seq data)	Zimmermann et al. (2016)	ImmPort (http://www.immport.org) with accession number SDY67
PBMC datasets	Monaco et al. (2019)	GEO (https://www.ncbi.nlm.nih.gov/geo/): GSE107990 and GSE59654
Software and algorithms		
Cytobank System	Cytobank	https://community.cytobank.org/
R software (v3.6.3)	R	https://cran.r-project.org
Cell Ranger (v2.1.0)	10× Genomics	https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/installation
Seurat (v3.1.1)	Butler et al. (2018)	https://satijalab.org/seurat
Salmon (v0.11.3)	Patro et al. (2017)	https://combine-lab.github.io/salmon
fastp (v0.20.0)	Chen et al. (2018)	https://github.com/OpenGene/fastp
tximport (v1.18.0)	Soneson et al. (2016)	https://bioconductor.org/packages/release/bioc/html/tximport.html
FlowSOM (v1.18.0)	Van Gassen et al. (2015)	https://bioconductor.org/packages/release/bioc/html/FlowSOM.html
Phenograph (v0.99.1)	Levine et al. (2015)	https://github.com/jacoblevine/PhenoGraph
DAISM-DNN^{XMBD}	Lin et al. (2022)	https://github.com/xmuyulab/DAISM-XMBD
Other		
70 μm Cell Strainer, Polypropylene Frame	Biologix	Cat# 15-1070
50 mL sterile conical tubes	Thermo Fisher Scientific	Cat# 339653
15 mL sterile conical tubes	Thermo Fisher Scientific	Cat# 339650
1 mL cryovial	Accumax	Cat# ACV01
15 mL Polypropylene Centrifuge Tubes	Corning Inc.	Cat# 430052
Falcon® 5 mL Round Bottom Polypropylene Tubes	Corning Inc.	Cat# 352063

(*Continued on next page*)

Continued		
REAGENT or RESOURCE	SOURCE	IDENTIFIER
Novocyte Flow cytometer	Agilent Technologies	Cat# 2010011AA
2100 Bioanalyzer	Agilent Technologies	Cat# G2939BA
Quantus™ Fluorometer	Promega	Cat# E6150
Neubauer hemocytometer	Thermo Fisher Scientific	Cat# 68052-14
Thermo-Flask™	Thermo Fisher Scientific	Cat# 2123-0010

MATERIALS AND EQUIPMENT

Complete RPMI 1640		
Reagent	Final concentration	Amount
RPMI Medium 1640 (1×)	N/A	500 mL
FBS	10%	50 mL
Benzonase® (25 kU/100 μL)	25 U/mL	55 μL
Total	**N/A**	**550 mL**
Store at 4°C for up to 2 weeks.		

STEP-BY-STEP METHOD DETAILS

Herein we describe Step-by-step methods for getting transcriptional data and corresponding cell type proportions data for the execution of DAISM-DNN. In this section, we provide detailed data processing steps and implementation of each module of DAISM-DNN.

Get your transcriptional data

Timing: 1 day

For both calibration samples and test samples, below is a general pipeline for preparation of RNA sequencing data for use with DAISM-DNN.

1. RNA is isolated using the RNeasy Mini Kit (QIAGEN) when using PBMC samples according to the manufacturer's instructions (please refer to https://www.qiagen.com/us/resources/resourcedetail?id=14e7cf6e-521a-4cf7-8cbc-bf9f6f a33e24&lang=en).
2. Measure the concentration of RNA isolated from PBMC samples using Quantus™ Fluorometer.

   Note: Calibrate the Quantus™ Fluorometer before measurement. If quantitating samples of higher concentration than the standard, dilute the sample to ensure it is within the linear range of the standard (troubleshooting problem 2).

3. Check RNA integrity using Agilent 2100 Bioanalyzer.

   Note: RNA integrity is assessed as an additional measure of the sample RNA quality. The Distribution Value 200 (DV200) is the percentage of RNA fragments longer than 200 nucleotides. The lower is the DV200 value, the more degraded is the RNA in the sample. We recommend using samples with DV200 higher than 30% for following steps.

4. Prepare 200 ng of total RNA in 5 μL of nuclease-free water for each sample.

   Pause point: RNA samples can be stored at −80°C for several months.

5. Generate and sequence cDNA libraries from isolated PBMCs using the Illumina NovaSeq 6000 System or an equivalent platform.

Process sequence data

Timing: 2 h

6. For processing RNA-seq data, quality assessment was carried out using fastp (Chen et al., 2018).
   a. Install fastp from the following sources: https://github.com/OpenGene/fastp.
   b. Once fastp installed, run the following command for paired end data:

```
> fastp -i in.R1.fq.gz -I in.R2.fq.gz -o out.R1.fq.gz -O out.R2.fq.gz
```

7. Perform sequence alignment to targeted genes from GRCh38 using Salmon (Patro et al., 2017) or an equivalent algorithm.
   a. Install Salmon according to the tutorial: https://combine-lab.github.io/salmon/getting_started/.
   b. In order to quantify transcript-level abundances, Salmon requires a target transcriptome. This transcriptome is given to Salmon in the form of a (possibly compressed) multi-FASTA file, with each entry providing the sequence of a transcript. Download the target transcript from Ensembl: http://asia.ensembl.org/info/data/ftp/index.html. For example, we'll be analyzing some human data, so we'll download and index the homo sapiens transcriptome.

```
>  curl  http://ftp.ensembl.org/pub/release-106/fasta/homo_sapiens/cdna/Homo_sapiens.GRCh38.cdna.all.fa.gz -o homo.fa.gz
```

   c. Extract a transcript-gene correspondence information file.

```
> zgrep ">" Homo_sapiens.GRCh38.cdna.all.fa.gz | sed 's/>//g' | sed 's/cdna.*gene_symbol://g' | sed 's/description.*//g' > gene_map.txt
```

   d. Next, we're going to build an index on our transcriptome.

```
> salmon index -t homo.fa.gz -i homo_index
```

   e. Quantifying the samples.

```
> salmon quant -i homo_index -l A \
     -1 out.R1.fq.gz \
     -2 out.R2.fq.gz \
     -g gene_map.txt \
     -p 8 –validateMappings -o salmon_out/${samp}
```

The -i argument tells salmon where to find the index. -l A tells salmon that it should automatically determine the library type of the sequencing reads (stranded vs. unstranded etc.). The -1 and -2 arguments tell salmon where to find the left and right reads for this sample (notice, salmon will accept gzipped FASTQ files directly). -g refers to the correspondence between transcripts and genes. If not specified, the output is quantification of gene expression. Otherwise, the output is the quantification of transcript expression. Finally, the -p 8 argument tells salmon to make use of 8 threads and the -o argument specifies the directory where salmon's quantification results should be written. You can read about salmon's many options in the documentation (https://salmon.readthedocs.io/en/latest/).

Note: After the salmon commands finish running, you should have a directory named "salmon_out", which will have a sub-directory for each sample. The main output file is called "quant.sf".

8. Get transcripts per million (TPM) normalization result from salmon output. Merge the quantification results from different samples via R package tximport.

```
> library(tximport)

> sampleList <- c("S1", "S2", "S3", "S4", "S5", "S6", "S7", "S8", "S9", "S10") #replace with your sample list

> fileList <- file.path("salmon_out/", sampleList, "quant.sf")

> names(fileList) <- sampleList

> txi <- tximport(fileList, type = "salmon", txOut = T)

> prefixOut <- "result/salmon"

> write.table(txi$abundance, file=paste(prefixOut, "TPM.txt", sep="."), sep= "\t",row.names = T, col.names = NA, quote = F)

> write.table(txi$counts, file=paste(prefixOut, "readscount.txt", sep="."), sep="\t", row.-names = T, col.names = NA, quote=F)
```

9. Compute the total reads count for each sample (salmon.readscount.txt). Remove the samples those total reads count is less than 1 million.
10. Save the TPM-normalization expression matrix with columns associated to sample ID and rows to genes specified using HUGO symbols as a tab-delimited file "caliexp.txt".

Obtain paired cell type proportions from flow or mass cytometry

Timing: 2 h

Every sample from calibration transcriptional dataset must have paired cell type proportions data for execution of DAISM-DNN.

11. Stains the sample with pre-selected antibodies that target surface and intracellular markers according to the manufacturer's instructions (For mass cytometry, please refer to https://www.fluidigm.com/download/7266. For flow cytometry, please refer to https://enquirebio.com/flow-cytometry).
12. Acquire data on flow or mass cytometry.
13. The acquired data can be normalized and analyzed with Cytobank (https://community.cytobank.org/) or other commonly used software for data cleaning, doublets, and dead cell removal.
14. The results were then exported as .fcs files for further analysis. Merge your FCS files from all samples using the cytofkit package (Chen et al., 2016) in R (v3.6.3). And scale the data with cytofAsinh-transformation.

```
>library(cytofkit)

>file_name <- list.files(raw_fcs_dir,pattern='.fcs$', full=TRUE)#replace 'raw_fcs_dir' with your directory

>combined_data <- cytof_exprsMerge(fcsFiles = file_name, transformMethod = "cytofAsinh", mergeMethod ="all")

>combined_data <- as.data.frame(combined_data)
```

15. Perform automated clustering with cytofkit using the FlowSOM algorithm (Van Gassen et al., 2015), or an equivalent algorithm, such as Phenograph (Levine et al., 2015).

```
>cell_clusters <- cytof_cluster(xdata = combined_data[,cluster_marker], method = "FlowSOM", FlowSOM_k = 40)

#if use phonograph alogorithm

>cell_clusters <- cytof_cluster(xdata = combined_data[,cluster_marker], method = " Rphenograph", Rphenograph_k = 1000)
```

Note: FlowSOM_k refers to number of clusters for meta clustering in FlowSOM. Rphenograph_k refers to integer number of nearest neighbours to pass to Rphenograph. "cluster_marker" refers to the subset antibodies users selected to identify specific populations in high-dimensional analysis.

16. After every single cell was assigned to a cluster, manually annotate each cluster based on its marker expression pattern compared with patterns of known immune cell types. Use heatmap to visualize mean values of normalized markers expression in each cluster (Figure 3).

```
>clustered_cells<-data.frame(combined_data[,cluster_marker], metacluster = cell_clusters)

#produce Marker-Cluster Heatmap

> heatmap_data <- clustered_cells %>%

  group_by_at(c('metacluster')) %>%

  summarise_if(is.numeric,mean,na.rm=TRUE)%>%

  data.frame()

>heatmap_data <- na.omit(heatmap_data)

>row.names(heatmap_data)<-as.character(heatmap_data[,'metacluster'])

>library(pheatmap)

>pheatmap(mat=heatmap_data[,cluster_marker],

     scale = "none",

     display_numbers =TRUE)
```

Note: The name of the cell type annotated to CyTOF data should be consistent with that of augmentation data (troubleshooting problem 3).

17. Calculate the cell type proportions of each sample according to manual annotation of each cell. The calibration fraction file should be in the format of a matrix composed of columns associated to sample ID and rows to cell type.

```
> cell_percentages <- cytof_clusterStat(data = clustered_cells, cluster = "metacluster", statMethod = "percentage")

> cell_percentages = as.data.frame(cell_percentages)

> cell_percentages = cell_percentages/100

# change the rownames of cell_percentages according to your annotation.
```

18. Save the calibration fraction file as a tab-delimited file.

```
> write.table(cell_percentages, file="califra.txt",sep="\t",quote=F)
```

⚠ CRITICAL: The same cell label should be used for a particular phenotype in both augmentation annotation table and calibration fraction file.

Generate dataset-specific training data via simulation modules of DAISM-DNN

Timing: 5 min

After expression profiles of test and calibration samples as well as corresponding cell type proportions are ready, users can start to run DAISM-DNN to achieve cell type proportion estimation. DAISM-DNN has two training set simulation modules. One is DAISM_simulation which uses DAISM strategy in generating mixtures (Figure 4). The other is Generic_simulation which generates training set only using purified cells.

19. Before generating the training set, the users can keep a number of hold-out samples from the input calibration dataset so that they can verify the performance of trained DAISM-DNN model on the input dataset. The following command can be used to create the hold-out data:

Note: In troubleshooting problem 4, we evaluated the effect of the size of training data generated from the same number of calibration samples on the deconvolution performance of DAISM-DNN.

```
$daism split -caliexp ./example/caliexp.txt -califra ./example/califra.txt -n 6 -seed 777
-outdir ./example/
```

The expression profile and corresponding cell type proportions file of calibration samples were saved to example folder. Where -caliexp defines the expression profile of calibration samples; -califra: the cell type fraction file of calibration samples; -n: the number of hold-out samples from calibration samples; -seed: random seed; -outdir: the folder where the output files are stored. This step will generate four files: the expression profiles and ground truth cell fraction files of hold-out samples (hold_out_exp.txt and hold_out_fra.txt) and the rest of calibration samples (rest_cali_exp.txt and rest_cali_fra.txt).

20. Generate dataset-specific training data populated from a certain amount of calibration samples using DAISM simulation module:

```
$daism DAISM_simulation -platform S -caliexp ./example/rest_cali_exp.txt -califra
./example/rest_cali_fra.txt -aug ./example/pbmc8k.h5ad -N 16000 -testexp ./example/tes-
texp.txt -outdir ./
```

Where -platform defines the platform of augmentation data (S refers to single cell RNA-seq while R refers to RNA-seq); -aug: purified samples expression h5ad file used as augmentation, here we use pbmc8k.h5ad for example; -N: number of simulation samples, the DAISM package set N to 16,000 as default; -outdir: the folder where the output files are stored. This step will generate two files: an DAISM-generated artificial RNA expression profiles based on calibration samples augmented with the augmentation data (DAISM_mixsam.txt) and their corresponding cell fractions (DAISM_mixfra.txt).

Optional: DAISM package also includes another simulation module which generates training dataset using only gene expressions of purified cells if the expression profile and corresponding cell type proportions of calibration samples are not available.

```
$daism Generic_simulation -platform S -aug ./example/pbmc8k.h5ad -N 16000 -testexp
./example/testexp.txt -outdir ./
```

Perform DAISM-DNN training

Timing: 10 min

A deep neural network can now be trained on the generated training data.

21. Use the DAISM-generated artificial profiles (DAISM_mixsam.txt) and corresponding artificial cell fractions (DAISM_mixfra.txt) to train the neural networks.

```
$daism training -trainexp ./output/DAISM_mixsam.txt -trainfra ./output/DAISM_mixfra.txt
-outdir ./ -ncuda 0 -p
```

Where -ncuda defines the serial number of GPU; -p: include this option to report the performance of trained model on training and validation set.

Note: The gene symbol and cell type list as well as the model will be saved in output folder specified.

Note: If the artificial mixtures generated using only gene expressions of purified cells, it is recommended to use "-sum2one" parameter in training and prediction process (troubleshooting problem 5).

Note: During the training process, DAISM-DNN randomly splits the training set and the validation set at a ratio of 8:2. DAISM-DNN will stop the training process when the validation error

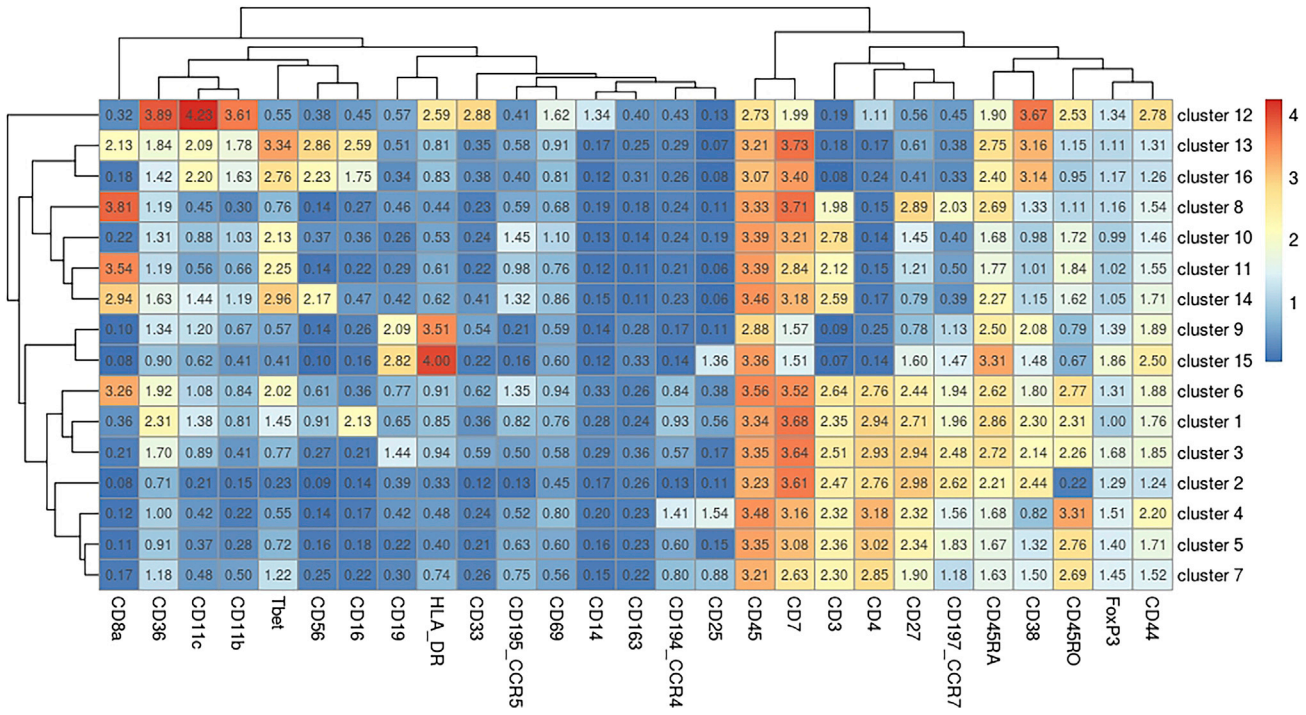


Figure 3. Heatmap showing mean values of normalized markers expression in each cluster of CyTOF data

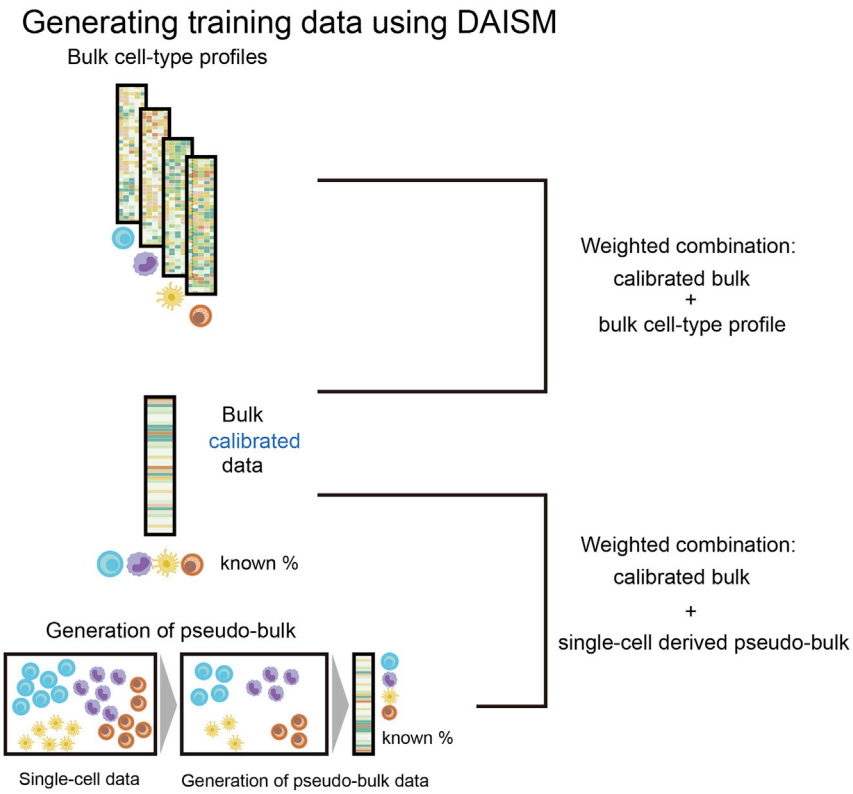


Figure 4. Schematic of generating pseudo-bulk training datasets using DAISM mixing strategy

did not decrease for 10 epochs, and select the model producing the best results on the validation set as the final model for prediction. A "-p" option is provided to report the performance of the final model on the training set and validation set in terms of Pearson correlation between predicted fraction and ground truth in console.

Validate hold-out samples on trained model

Timing: < 1 min

Users can validate the trained model on the hold-out samples (generated in step 19) to see if the desired performance can be achieved on test samples of interest.

22. Predict cell type proportions of hold-out samples via DAISM prediction module:

```
$daism prediction -testexp ./example/hold_out_exp.txt -model ./output/DAISM_model.pkl
-celltype ./output/DAISM_model_celltypes.txt -feature ./output/DAISM_model_feature.txt
-outdir ./
```

23. Run the following command to evaluate the performance on hold-out samples. The output "metrics.txt" file reports the Pearson correlation, spearman correlation, Lin's concordance correlation coefficient (CCC) and root-mean-square error (RMSE) between predicted fractions and ground truth.

```
$daism metrics -pred ./output/DAISM_result.txt -gt ./example/hold_out_fra.txt -outdir ./
```

Where -pred defines the prediction file; -gt defines the ground truth cell type proportions file.

Estimate cell type proportions via trained model

Timing: < 1 min

After validating the performance of the trained model on hold-out samples, users can perform cell type proportion estimation on the test samples of interest.

24. Predict cell type proportions of test data via DAISM prediction module:

```
$daism prediction -testexp ./example/testexp.txt -model ./output/DAISM_model.pkl -cell-
type ./output/DAISM_model_celltypes.txt -feature ./output/DAISM_model_feature.txt -out-
dir ./
```

Optional: DAISM package also provides one-stop DAISM-DNN module which integrates simulation, training and prediction in one command.

```
$daism DAISM -platform S -caliexp ./example/caliexp.txt -califra ./example/califra.txt -aug
./example/pbmc8k.h5ad -N 16000 -testexp ./example/testexp.txt -outdir ./ -ncuda 0 -write
```

Where -write defines whether to write simulation expression profile and corresponding cell type proportions file to disk. If users choose not to save simulation products, remove -write parameter.

EXPECTED OUTCOMES

DAISM-DNN provides a robust and accurate cell type proportion estimation from bulk RNA-seq data through training deep neural network on dataset-specific training set.

As outcome, DAISM-DNN deconvolute bulk RNA-seq to predict absolute proportions of cell types of interest. The result is a matrix composed of columns associated to sample ID and rows to cell types of interest.

We evaluated the performance of DNN models trained from DAISM-generated pseudo training data (DAISM-DNN) on the RNA-seq dataset SDY67 (Zimmermann et al., 2016). A total of 250 samples with ground truth proportions of five cell types (B cells, CD4 T cells, CD8 T cells, monocytes, and NK cells) from SDY67 were used for analysis in this paper. The performance of DAISM-DNN was measured from 30 permutation tests independently. For each permutation test, we used 50 randomly selected samples from SDY67 as testing data, and the remaining 200 samples were served as calibration data, which were augmented with the scRNA-seq data PBMC8k of the five cell types to create the training data respectively. DNN was trained on DAISM-generated training data (Figure 5). Pearson correlation, Lin's concordance correlation coefficient (CCC) and root-mean-square error (RMSE) between predicted fractions and ground truth were used to evaluate the performance of DAISM-DNN.

LIMITATIONS

Compared to other deconvolution algorithms, the DAISM-DNN model needs to be trained from the RNA-seq expression profiles and the corresponding ground truth cell fractions from a set of

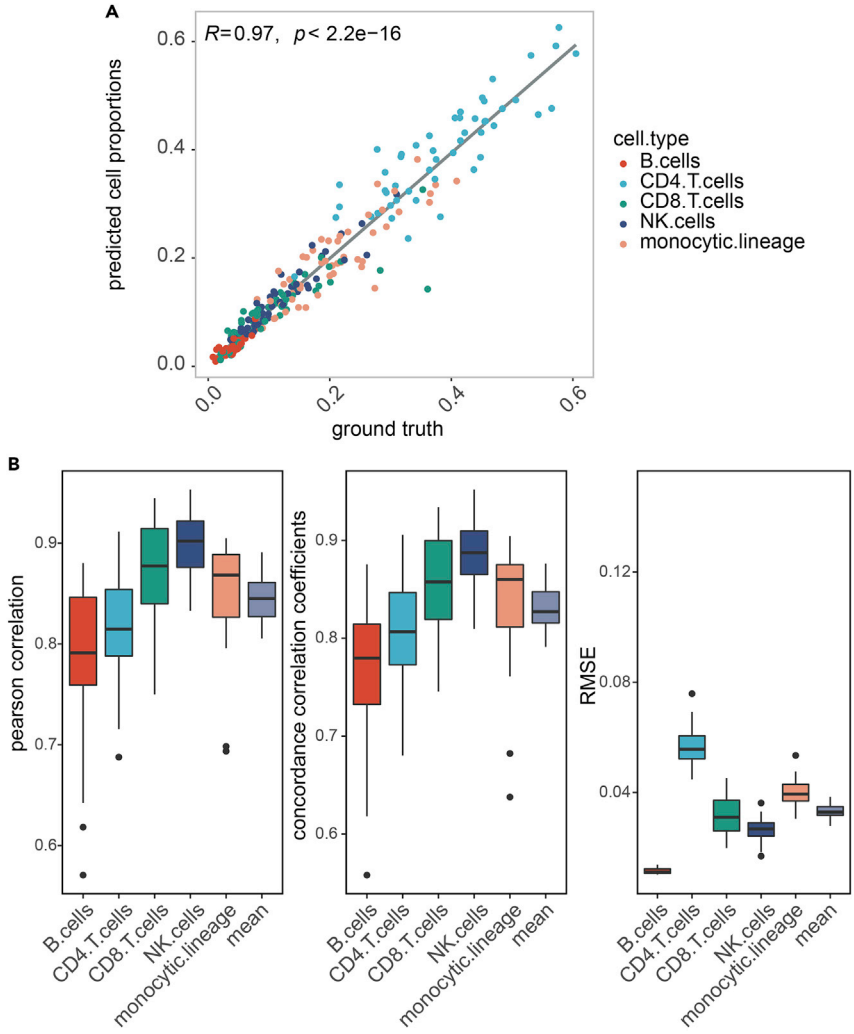


Figure 5. Performance evaluation of DAISM-DNN on RNA-seq dataset SDY67

(A) Scatterplots of ground truth fractions (x axis) and predicted cell fractions (y axis) for DAISM-DNN. Global Pearson correlation (r) is shown in scatter plots.

(B) The boxplots show the Pearson correlation, CCC and RMSE for each cell type in 30 permutation experiments.

calibration samples. Therefore, this protocol is more suitable for applications with rigorous quality control requirements, e.g., in a clinical setting where not only statistical correlation but also individual level prediction performance is expected. In addition, as the performance of DAISM-DNN may be affected by the number of calibration samples as well as the measurement errors of ground truth cell type fractions of calibration samples, it is highly recommended that the users train a high-quality DAISM-DNN model based on a sufficient number of calibration samples (e.g., 100–200), and then use the model on subsequent testing samples measured following the same standard operating procedure (SOP). Note that once a dataset-specific DAISM-DNN model is trained, it can be reused without the need for retraining as long as the expression profiles of testing samples are acquired using the same SOPs as those used in generating the model.

Another limitation of DAISM-DNN is that it requires RNA-seq expression profiles of purified cells or single cells that match the target cell types of interest for data augmentation. This could be substantially eased by leveraging public single-cell datasets with well-annotated cell types information.

TROUBLESHOOTING

Problem 1

Low yield and/or viability of PBMCs. (before you begin: step 6).

Potential solution

Ensure that Ficoll temperature is between 18°C to 20°C. Ficoll is less dense at higher temperatures. This may allow lymphocytes to enter the Ficoll layer instead of collecting at the interphase.

Avoid aspirating excessive Ficoll while collecting the PBMC layer. Exposure to Ficoll for long durations may be toxic to cells. Wash PBMCs with PBS two to three times.

Problem 2

Low or no fluorescence reading. (step-by-step method details: step 2).

Potential solution

Sample concentration was too low. Redilute sample and repeat measurement.

The wrong standard concentration was used to calibrate the Quantus™ Fluorometer. Check that the fluorescence is within the range of the standard curve used (e.g., if the RNA samples are of low concentration, then use the low concentration standard), and that the standard used the same as the sample.

Problem 3

The cell types of augmentation data may not match that of CyTOF data even called the same names. (step-by-step method details: step 16).

Potential solution

Users can use CITE-seq data or other data which provide single-cell transcriptome and surface proteins simultaneously for augmentation. First, perform clustering on normalized protein profiles of CyTOF and CITE-seq on surface markers in common, respectively. Then calculate Pearson correlation between each cluster of CyTOF and CITE-seq data based on the mean values of marker expressions. For each CyTOF cluster, we identified the best-matching cluster of CITE-seq according to the correlation of the two clusters. The details may refer to Lin et al. (2022).

Problem 4

How many in-silico samples are needed to be generated from calibration samples? (step-by-step method details: step 19).

Potential solution

To evaluate the effect of the size of training data generated from the same number of calibration samples on the deconvolution performance of DAISM-DNN, we tested DAISM-DNN with different training data sizes ranging from 640 up to 32,000 simulated samples on three PBMC datasets (GSE59654, GSE107990, SDY67), respectively. The performance of each model was measured from 30 permutation tests. In each permutation test, 50 randomly selected samples were held out as the test samples, and the remaining samples from the same dataset were used as calibration samples. The mean CCC performance across cell types improved as the training data size increased on all three datasets (Figure 6, reprinted with permission from Supplementary Figure S18 in Lin et al. (2022)). However, the increment started to level off when the training data size increased to 3,200 simulated samples for GSE59654 and GSE107990, and about 12,800 simulated samples for SDY67. Based on this result, we used 16,000 simulated samples as the default size of training data in our experiments.

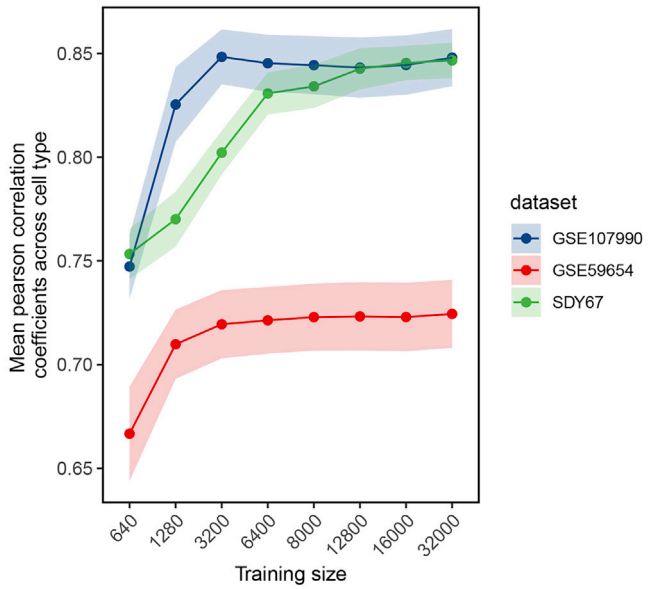


Figure 6. Assessment of the effect of training data size on mean Pearson correlation coefficients across cell types

Problem 5

The predicted proportions of cell types of interest do not meet "sum to one". (step-by-step method details: step 20).

Potential solution

Note that we don't use "sum to one" constraint in our objective equation of neural networks as in some cases the target cell types may not cover all cell types existing in the sample. Therefore, the sum of the fractions of target cell types may not necessarily equal to one. However, we provide a "-sum2one" option to add this constraint to DAISM-DNN models when it is necessary. This option can be used as follows:

```
#training

$daism training -trainexp ./output/Generic_mixsam.txt -trainfra ./output/Generic_mix-
fra.txt -outdir ./ -ncuda 0 -sum2one

#prediction

$daism prediction -testexp ./example/testexp.txt -model ./output/DAISM_model.pkl -cell-
type ./output/DAISM_model_celltypes.txt -feature ./output/DAISM_model_feature.txt -out-
dir ./ -sum2one
```

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Rongshan Yu (rsyu@xmu.edu.cn).

Materials availability

There are no newly generated materials associated with this protocol.

Data and code availability

The source code for DAISM-DNN is available at https://github.com/xmuyulab/DAISM-XMBD, https://zenodo.org/record/6606456. The dataset SDY67 provides both flow cytometry data and

RNA-Seq data and the data are available through ImmPort (http://www.immport.org) with accession number SDY67. The dataset GSE107990 and GSE59654 were downloaded from GEO (https://www.ncbi.nlm.nih.gov/geo/). Code for generating figures in this Protocol, including Troubleshooting, is available as an R script in Data S2. Augmentation data for DAISM are available at Mendeley data (https://doi.org/10.17632/ysjwjvpnh3.1).

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.xpro.2022.101587.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (81788101 to J.H., 82074508 to Y.H. and L.Z.).

AUTHOR CONTRIBUTIONS

Y.L., S.W., M.W., H.L., and X.X. generated figures and wrote original draft. W.Y., J.H., and R.Y. conceived the project and designed the methodology. L.Z., K.W., and J.Z. performed all experiments. F.Z. and M.Z. contributed to discussion and reviewed and edited the manuscript. All authors assisted in writing the manuscript.

DECLARATION OF INTERESTS

R.Y. and W.Y. are shareholders of Aginome Scientific. J.Z. and F.Z. are employees of Amoy Diagnostics.

REFERENCES

Butler, A., Hoffman, P., Smibert, P., Papalexi, E., and Satija, R. (2018). Integrating single-cell transcriptomic data across different conditions, technologies, and species. Nat. Biotechnol. *36*, 411–420. https://doi.org/10.1038/nbt.4096.

Chen, S., Zhou, Y., Chen, Y., and Gu, J. (2018). fastp: an ultra-fast all-in-one FASTQ preprocessor. Bioinformatics *34*, i884–i890. https://doi.org/10.1093/bioinformatics/bty560.

Chen, H., Lau, M.C., Wong, M.T., Newell, E.W., Poidinger, M., and Chen, J. (2016). Cytofkit: a bioconductor package for an integrated mass cytometry data analysis pipeline. PLoS Comput. Biol. *12*, e1005112. https://doi.org/10.1371/journal.pcbi.1005112.

Levine, J.H., Simonds, E.F., Bendall, S.C., Davis, K.L., Amir, E.a.D., Tadmor, M.D., Litvin, O., Fienberg, H.G., Jager, A., Zunder, E.R., et al. (2015). Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. Cell *162*, 184–197. https://doi.org/10.1016/j.cell.2015.05.047.

Lin, Y., Li, H., Xiao, X., Zhang, L., Wang, K., Zhao, J., Wang, M., Zheng, F., Zhang, M., Yang, W., et al. (2022). DAISM-DNNXMBD: highly accurate cell type proportion estimation with in silico data augmentation and deep neural networks. Patterns *3*, 100440. https://doi.org/10.1016/j.patter.2022.100440.

Monaco, G., Lee, B., Xu, W., Mustafah, S., Hwang, Y.Y., Carré, C., Burdin, N., Visan, L., Ceccarelli, M., Poidinger, M., et al. (2019). RNA-seq signatures normalized by mRNA abundance allow absolute deconvolution of human immune cell types. Cell Rep. *26*, 1627–1640.e7. https://doi.org/10.1016/j.celrep.2019.01.041.

Patro, R., Duggal, G., Love, M.I., Irizarry, R.A., and Kingsford, C. (2017). Salmon provides fast and bias-aware quantification of transcript expression. Nat. Methods *14*, 417–419. https://doi.org/10.1038/nmeth.4197.

Soneson, C., Love, M.I., and Robinson, M.D. (2016). Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. F1000Research *4*, 1521. https://doi.org/10.12688/f1000research.7563.2.

Sturm, G., Finotello, F., Petitprez, F., Zhang, J.D., Baumbach, J., Fridman, W.H., List, M., and Aneichyk, T. (2019). Comprehensive evaluation of transcriptome-based cell-type quantification methods for immuno-oncology. Bioinformatics *35*, i436–i445. https://doi.org/10.1093/bioinformatics/btz363.

Tegner, J., and Gomez-Cabrero, D. (2022). Data-driven bioinformatics to disentangle cells within a tissue microenvironment. Trends Cell Biol. *32*, 467–469. https://doi.org/10.1016/j.tcb.2022.03.009.

Van Gassen, S., Callebaut, B., Van Helden, M.J., Lambrecht, B.N., Demeester, P., Dhaene, T., and Saeys, Y. (2015). FlowSOM: using self-organizing maps for visualization and interpretation of cytometry data. Cytometry A. *87*, 636–645. https://doi.org/10.1002/cyto.a.22625.

Vallania, F., Tam, A., Lofgren, S., Schaffert, S., Azad, T.D., Bongen, E., Haynes, W., Alsup, M., Alonso, M., Davis, M., et al. (2018). Leveraging heterogeneity across multiple datasets increases cell-mixture deconvolution accuracy and reduces biological and technical biases. Nat. Commun. *9*, 4735. https://doi.org/10.1038/s41467-018-07242-6.

Wolf, F.A., Angerer, P., and Theis, F.J. (2018). SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. *19*, 15. https://doi.org/10.1186/s13059-017-1382-0.

Zhang, X., Lan, Y., Xu, J., Quan, F., Zhao, E., Deng, C., Luo, T., Xu, L., Liao, G., Yan, M., et al. (2018). CellMarker: a manually curated resource of cell markers in human and mouse. Nucleic Acids Res. *47*, D721–D728. https://doi.org/10.1093/nar/gky900.

Zheng, G.X.Y., Terry, J.M., Belgrader, P., Ryvkin, P., Bent, Z.W., Wilson, R., Ziraldo, S.B., Wheeler, T.D., McDermott, G.P., Zhu, J., et al. (2017). Massively parallel digital transcriptional profiling of single cells. Nat. Commun. *8*, 14049. https://doi.org/10.1038/ncomms14049.

Zimmermann, M.T., Oberg, A.L., Grill, D.E., Ovsyannikova, I.G., Haralambieva, I.H., Kennedy, R.B., and Poland, G.A. (2016). System-wide associations between DNA-methylation, gene expression, and humoral immune response to influenza vaccination. PLoS One *11*, e0152034. https://doi.org/10.1371/journal.pone.0152034.