Analysis of single-cell RNA-seq data (IV)

Hao Wu
Department of Biostatistics
and Bioinformatics
Rollins School of Public Health
Emory University

Ziyi Li
Department of Biostatistics
The University of Texas MD
Anderson Cancer Center

ENAR 2021 short course
March 2021

Course outline

- 8-9:15: Intro and data preprocessing.
- 9:15-9:45: Lab: preprocessing and visualization.
- 10-11:15: Normalization, batch effect, imputation, DE, simulator.
- 11:15-12: Lab: Normalization, batch effect, imputation, DE, simulator

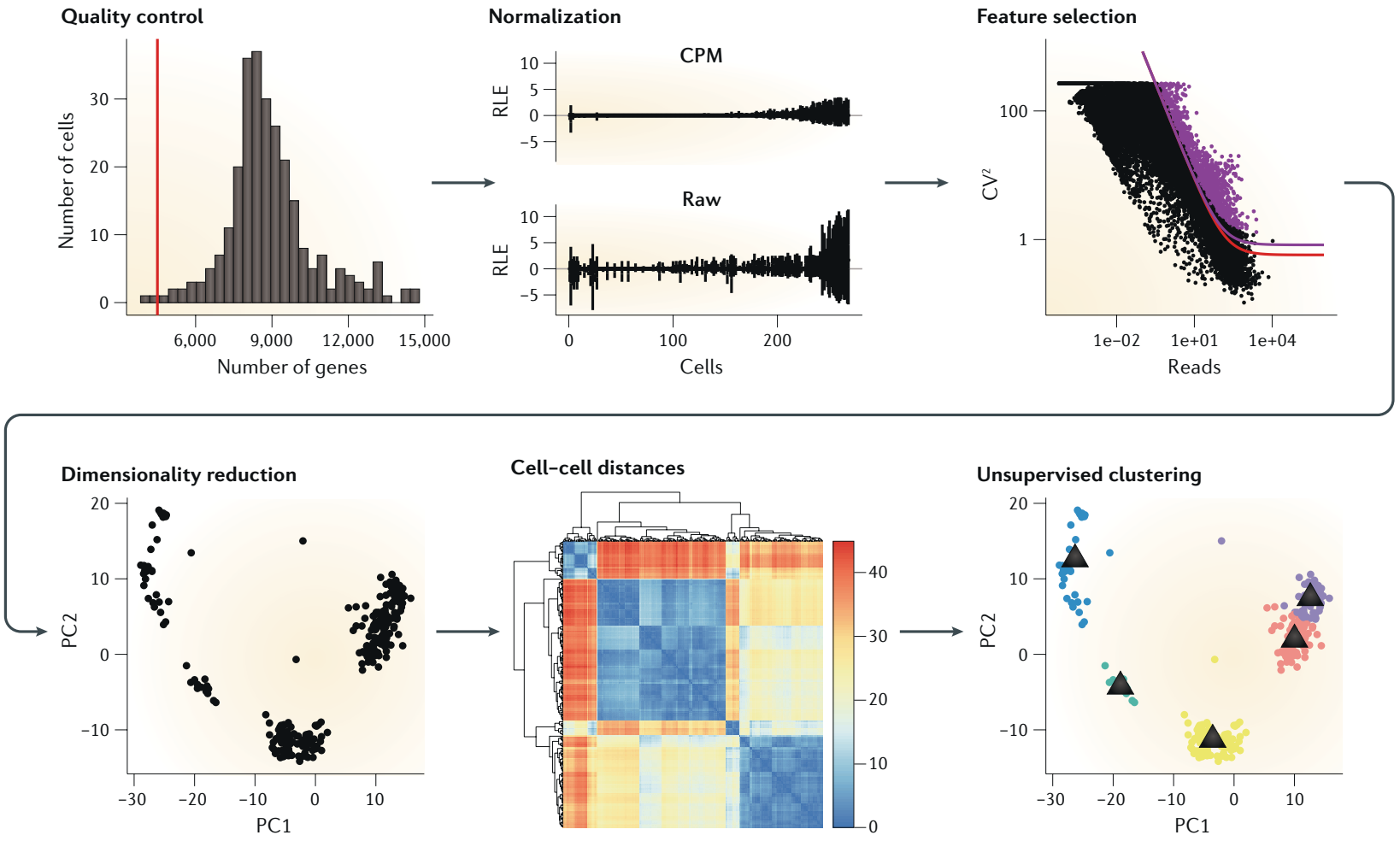
- 12-1: Lunch break

- 1-2: Clustering and pseudotime construction
- 2-2:30: Lab: Clustering and pseudotime construction
- **2:45–3:30: Supervised cell typing & related single cell data sources**
- 3:30-4: Lab: supervised cell typing.
- 4:15-5: scRNA-seq in cancer

Outline for this session

- **Background**
  - Motivation
  - Assumptions and challenges
- **Cell type annotation**
  - Existing methods
  - Performance comparisons and considerations
- **Obtain existing single cell datasets**
- **Data integration**

Example scRNA-seq analysis workflow



Kiselev et al. (2019) Nat. Rev. Genet.

Motivation

- Another paradigm to identify cell type.
- Cell clustering (unsupervised):
  - Cluster cells to multiple clusters (unsupervised). then assign cell type for each cluster. - laborious, lack of reproducibility
- Cell type assignment (supervised):
  - Directly assign each cell to a cell type.
  - Requires some training data (supervised) or marker gene info.
  - Potentially work better for data from multiple samples.
  - Can incorporate the hierarchy in cell types.
  - Cannot identify new cell types (restricted to the known cell types in the reference).

Cell type annotation

- Require the input of marker gene information
  - DigitalCellSorter (BMC bioinfo, 2019)
  - Garnett (Nature methods, 2019)
  - CellAssign (Nature methods, 2019)
  - SCINA (Genes, 2019)
  - scSorter (Genome Biology, 2021)
- Pre-train a classifier using scRNA-seq training data with generic machine learning methods: SVM, LDA, RF, kNN, RF
  - Scmap (Nature methods, 2018)
  - CHETAH (NAR, 2019)
  - CaSTLe (PloS One, 2018)
  - scPred (Genome Biology, 2019)

Cell type annotation (continue)

- Use either sc or bulk RNA-seq as reference
  - singleR (Nat Immunol, 2019)
- A comparison paper: Abdelaal et al. (2019, GB)
- Annotation performance is a trade-off between accuracy and un-assigned rate

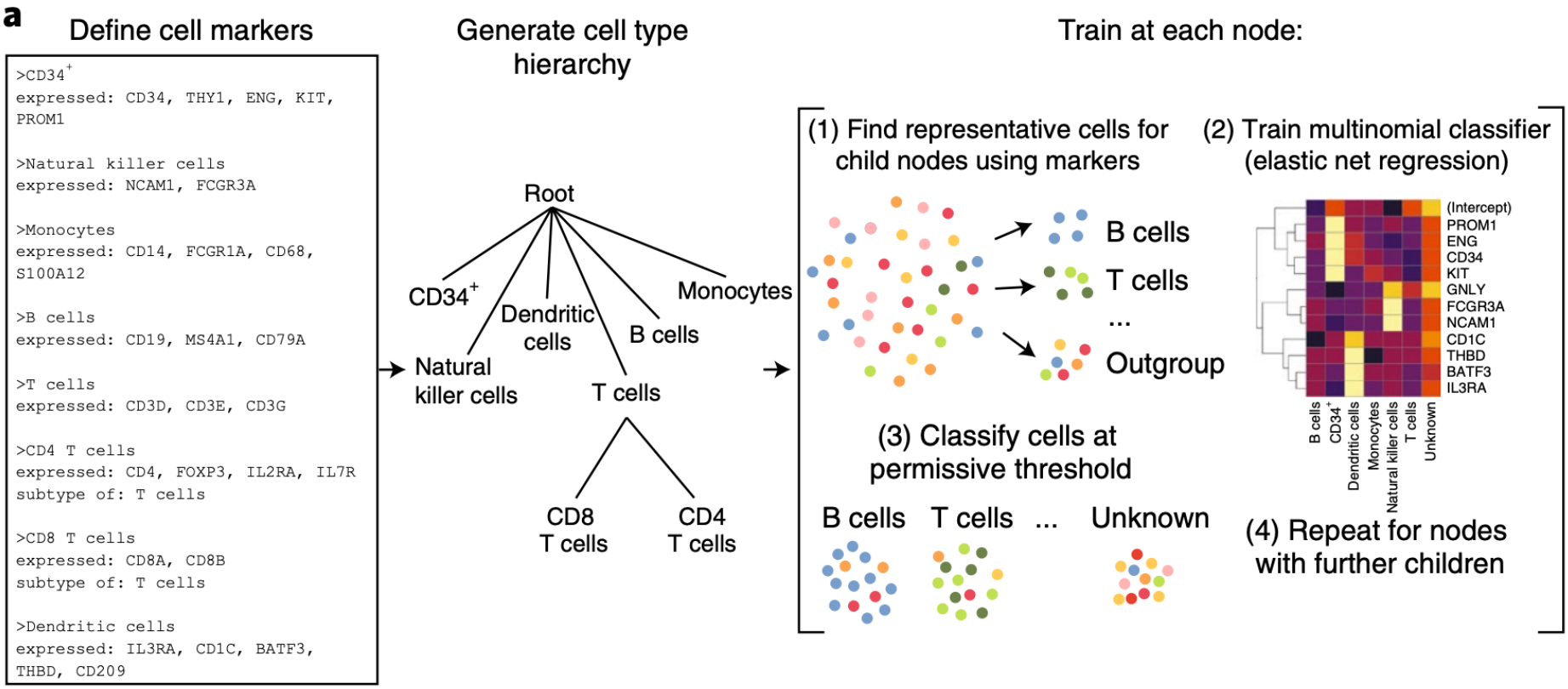
Name	Version	Language	Underlying classifier	Prior knowledge	Rejection option	Reference
Garnett	0.1.4	R	Generalized linear model	Yes	Yes	[14]
Moana	0.1.1	Python	SVM with linear kernel	Yes	No	[15]
DigitalCellSorter	GitHub version: e369a34	Python	Voting based on cell type markers	Yes	No	[16]
SCINA	1.1.0	R	Bimodal distribution fitting for marker genes	Yes	No	[17]
scVI	0.3.0	Python	Neural network	No	No	[18]
Cell-BLAST	0.1.2	Python	Cell-to-cell similarity	No	Yes	[19]
ACTINN	GitHub version: 563bcc1	Python	Neural network	No	No	[20]
LAmbDA	GitHub version: 3891d72	Python	Random forest	No	No	[21]
scmapcluster	1.5.1	R	Nearest median classifier	No	Yes	[22]
scmapcell	1.5.1	R	kNN	No	Yes	[22]
scPred	0.0.0.9000	R	SVM with radial kernel	No	Yes	[23]
CHETAH	0.99.5	R	Correlation to training set	No	Yes	[24]
CaSTLe	GitHub version: 258b278	R	Random forest	No	No	[25]
SingleR	0.2.2	R	Correlation to training set	No	No	[26]
scID	0.0.0.9000	R	LDA	No	Yes	[27]
singleCellNet	0.1.0	R	Random forest	No	No	[28]
LDA	0.19.2	Python	LDA	No	No	[29]
NMC	0.19.2	Python	NMC	No	No	[29]
RF	0.19.2	Python	RF (50 trees)	No	No	[29]
SVM	0.19.2	Python	SVM (linear kernel)	No	No	[29]
$SVM_{rejection}$	0.19.2	Python	SVM (linear kernel)	No	Yes	[29]
kNN	0.19.2	Python	kNN ($k = 9$)	No	No	[29]

Abdelaal et al. (2019, GB)

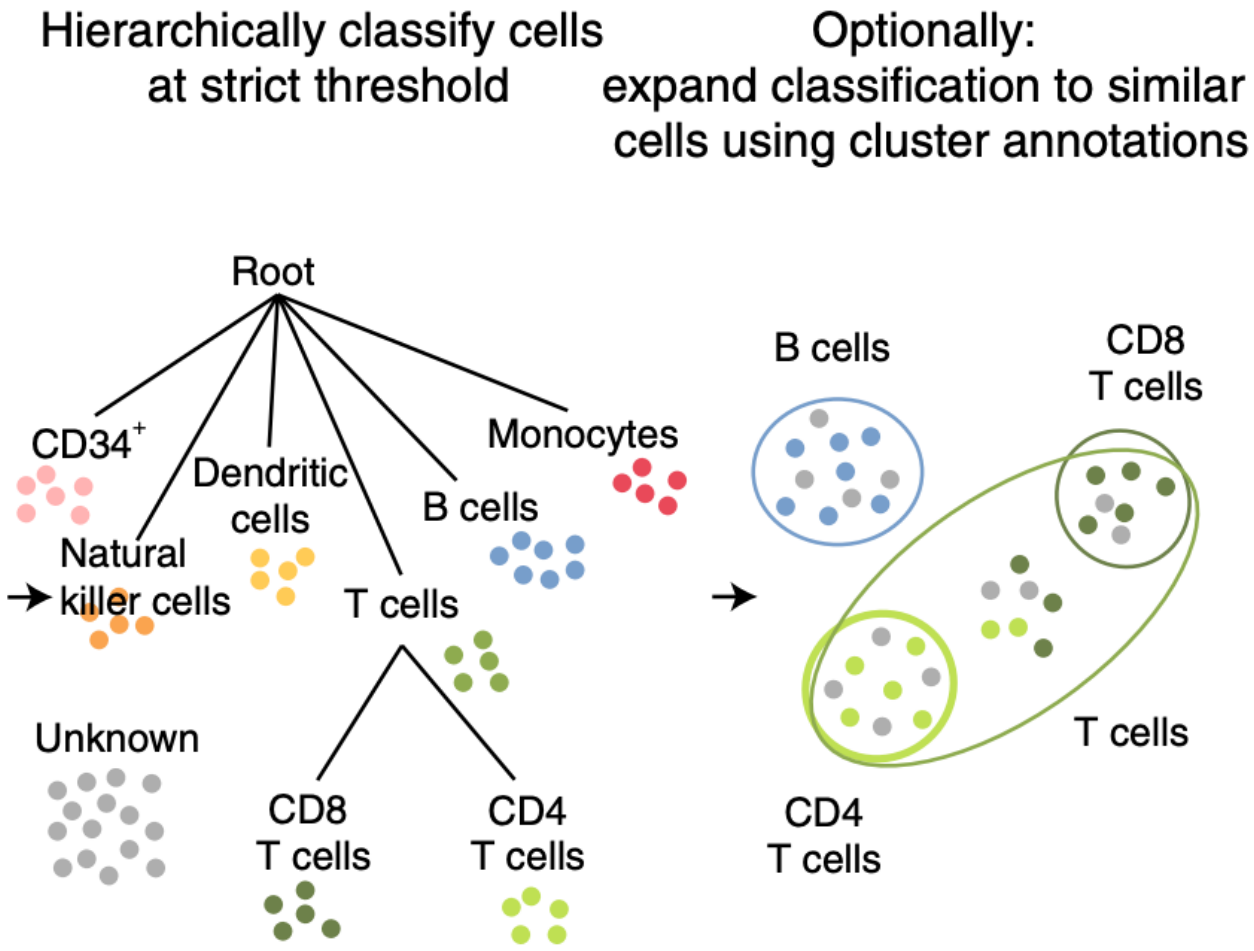
Cell type annotation

- Require the input of marker gene information
  - **Garnett** (Nature methods, 2019)
  - **scSorter** (Genome Biology, 2021)
- Pre-train a classifier using scRNA-seq training data with generic machine learning methods: SVM, LDA, RF, kNN, RF
  - **Scmap** (Nature methods, 2018)
  - **CHETAH** (NAR, 2019)
- Use either sc or bulk RNA-seq as reference
  - **singleR** (Nat Immunol, 2019)

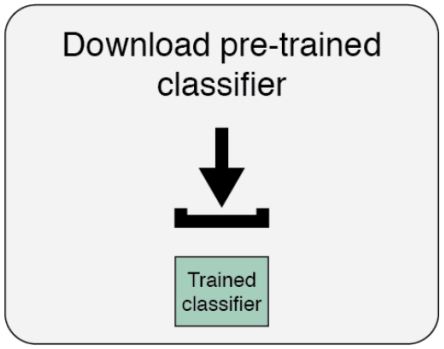
Garnett


a
Define cell markers
>CD34+
expressed: CD34, THY1, ENG, KIT, PROM1
>Natural killer cells
expressed: NCAM1, FCGR3A
>Monocytes
expressed: CD14, FCGR1A, CD68, S100A12
>B cells
expressed: CD19, MS4A1, CD79A
>T cells
expressed: CD3D, CD3E, CD3G
>CD4 T cells
expressed: CD4, FOXP3, IL2RA, IL7R
subtype of: T cells
>CD8 T cells
expressed: CD8A, CD8B
subtype of: T cells
>Dendritic cells
expressed: IL3RA, CD1C, BATF3, THBD, CD209
Generate cell type hierarchy
Root
CD34+
Dendritic cells
Monocytes
B cells
Natural killer cells
T cells
CD8 T cells
CD4 T cells
Train at each node:
(1) Find representative cells for child nodes using markers
B cells
T cells
...
Outgroup
(2) Train multinomial classifier (elastic net regression)
(Intercept)
PROM1
ENG
CD34
KIT
GNLY
FCGR3A
NCAM1
CD1C
THBD
BATF3
IL3RA
B cells
CD34+
Dendritic cells
Monocytes
Natural killer cells
T cells
Unknown
(3) Classify cells at permissive threshold
B cells
T cells
...
Unknown
(4) Repeat for nodes with further children

Garnett


Hierarchically classify cells at strict threshold
Optionally: expand classification to similar cells using cluster annotations
Root
CD34+
Dendritic cells
Monocytes
B cells
Natural killer cells
T cells
Unknown
CD8 T cells
CD4 T cells
B cells
CD8 T cells
T cells
CD4 T cells

Available pre-trained classifier for Garnett



Classifier	Marker file	Species	Tissue	Contributer	Training data source	Publication	Date posted
hsLung	hsLung_markers.txt	Human	Lung	Hannah Pliner	Lambrechts et. al.	Pliner et. al.	2019-10-17
hsPBMC	hsPBMC_markers.txt	Human	PBMC	Hannah Pliner	10x Genomics	Pliner et. al.	2019-10-17
mmLung	mmLung_markers.txt	Mouse	Lung	Hannah Pliner	Han et. al.	Pliner et. al.	2019-10-17
ceWhole	ceWhole_markers.txt	C. elegans	Whole	Hannah Pliner	Cao et. al.	Pliner et. al.	2019-10-17
mmBrain	mmBrain_markers.txt	Mouse	Brain and spinal cord	Hannah Pliner	Zeisel et. al.	Pliner et. al.	2019-10-17

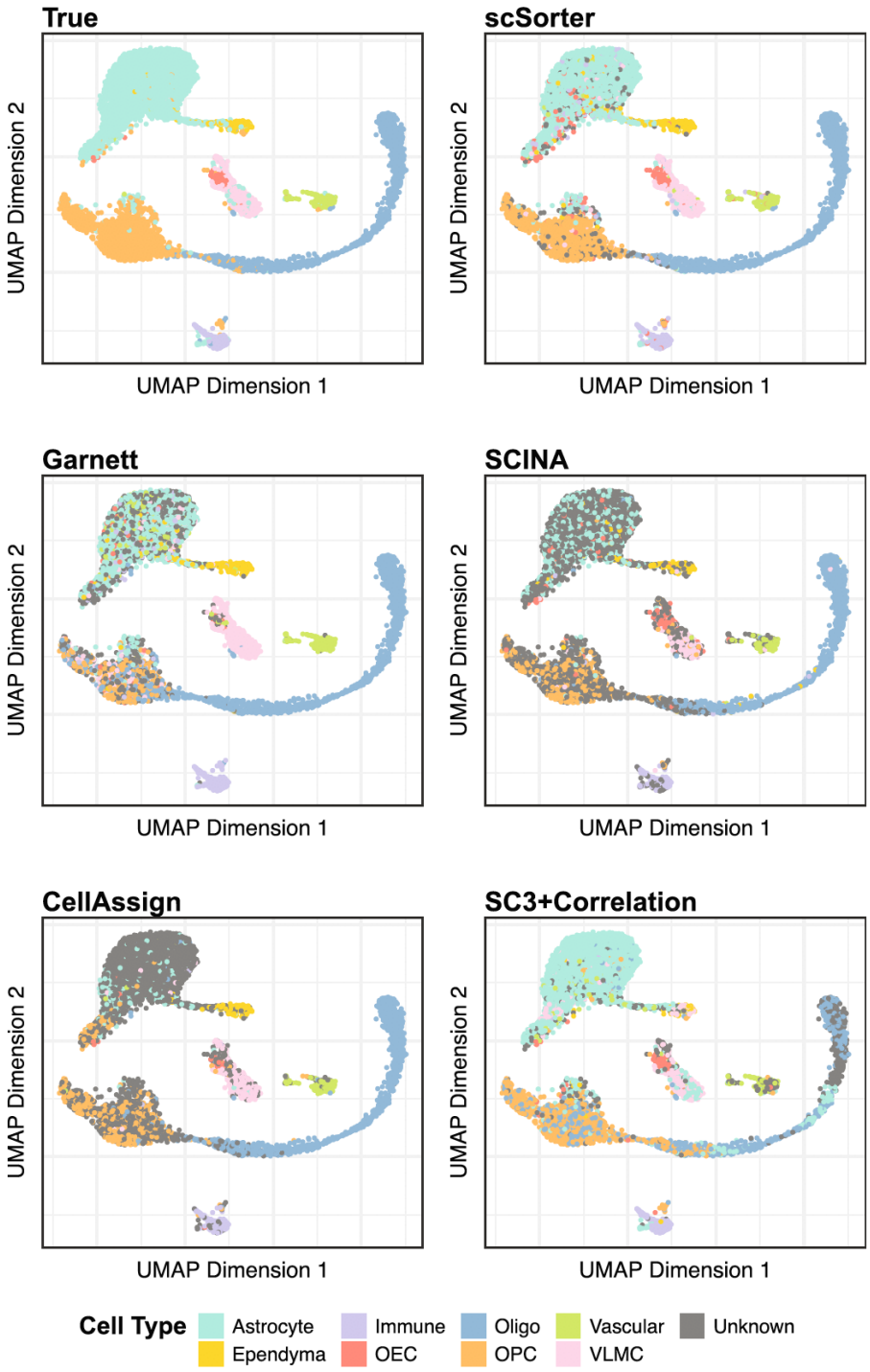
Example code for Garnett

```
marker_file_path <- system.file("extdata", "pbmc_test.txt", package =
"garnett")
pbmc_classifier <- train_cell_classifier(cds = pbmc_cds,
        marker_file = marker_file_path,
        db=org.Hs.eg.db,
        cds_gene_id_type = "SYMBOL",
        num_unknown = 50,
        marker_file_gene_id_type = "SYMBOL")
pbmc_cds <- newCellDataSet(as(mat, "dgCMatrix"),
        phenoData = pd,
        featureData = fd) # generate size factors for normalization
pbmc_cds <- estimateSizeFactors(pbmc_cds)
pbmc_cds <- classify_cells(pbmc_cds,
            pbmc_classifier,
            db = org.Hs.eg.db,
            cluster_extend = TRUE,
            cds_gene_id_type = "SYMBOL")
```

https://cole-trapnell-lab.github.io/garnett/docs/#2-classifying-your-cells

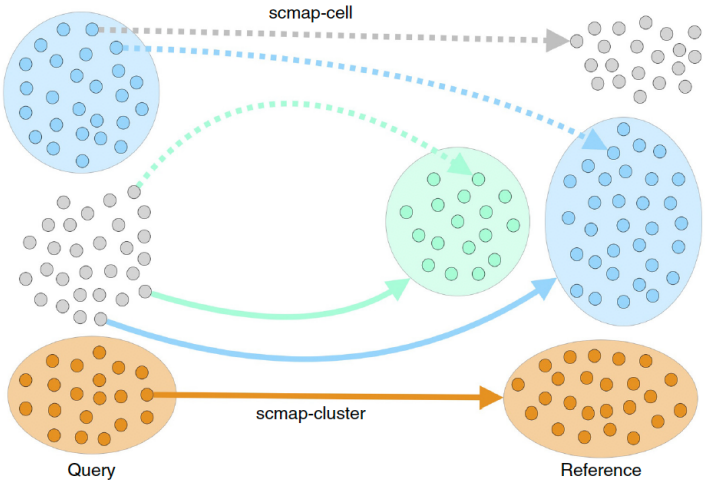
scSorter

- Given marker genes, their exact expression levels are not assumed known, and no reference dataset is used.
- Borrow information from non-marker genes

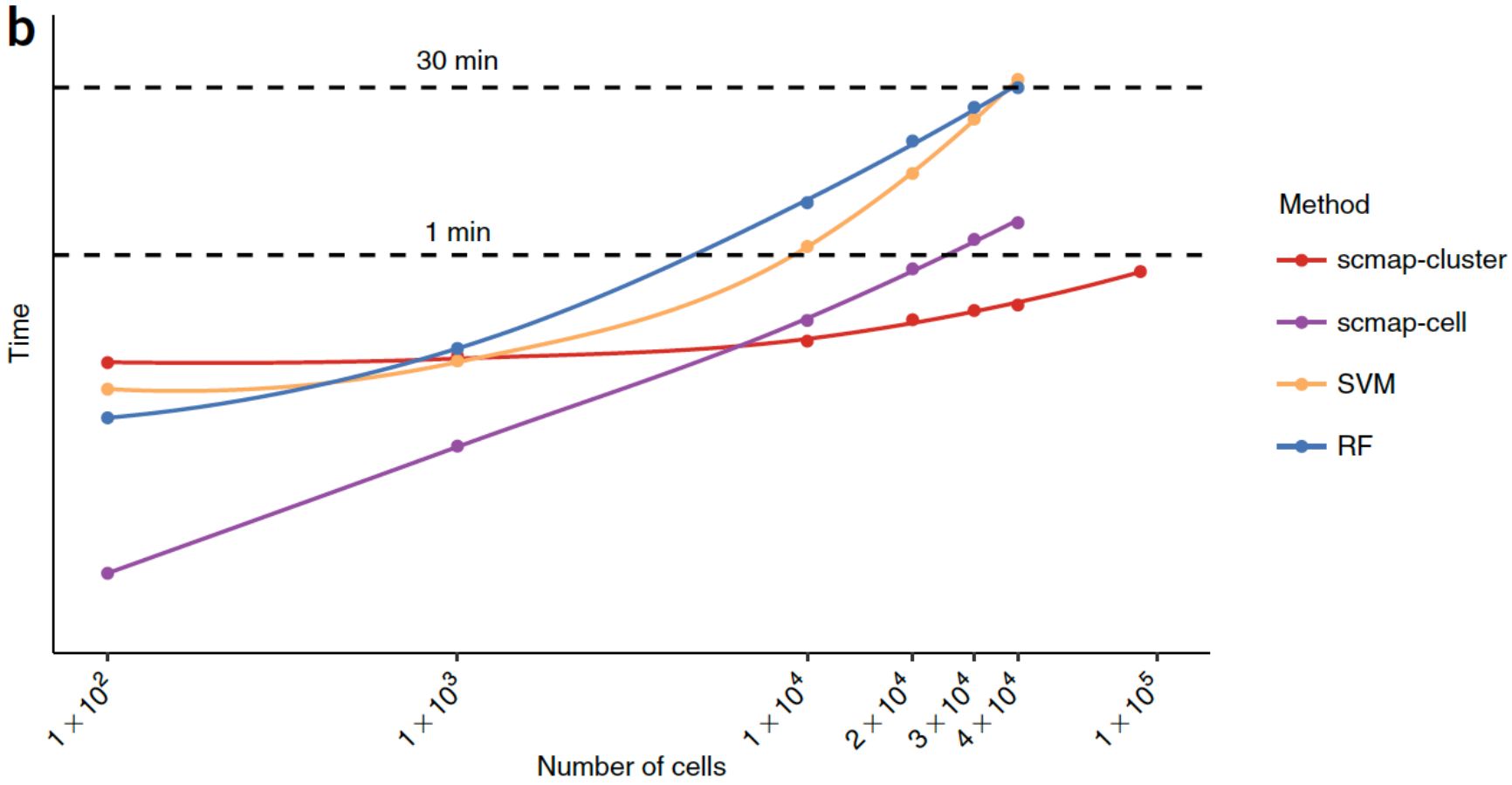


scmap

- Correlation-based cell label assignment
- Fast and accurate
- A correlation threshold to control the percentage of assigned cells, cells below the threshold are “unassigned”



scmap

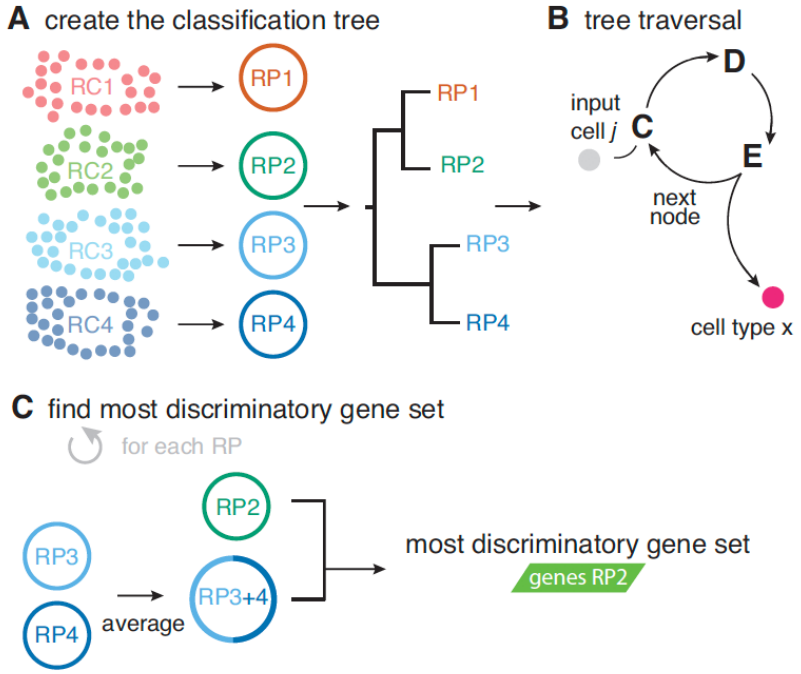

b
30 min
1 min
Time
Number of cells
1 + 10²
1 + 10³
1 + 10⁴
2 + 10⁴
3 + 10⁴
4 + 10⁴
1 + 10⁵
Method
scmap-cluster
scmap-cell
SVM
RF

Example code for scmap

```
sce <- SingleCellExperiment(assays =
    list(normcounts = as.matrix(trainmat)),
    colData = DataFrame(cell_type1 = trainlabel))
logcounts(sce) <- log2(normcounts(sce) + 1)
rowData(sce)$feature_symbol <- rownames(sce)
sce <- selectFeatures(sce, suppress_plot = TRUE)

sce_test <- SingleCellExperiment(assays =
    list(normcounts = as.matrix(testmat)),
    colData = DataFrame(cell_type1 = testlabel))
logcounts(sce_test) <- log2(normcounts(sce_test) + 1)
rowData(sce_test)$feature_symbol <- rownames(sce_test)

sce <- indexCluster(sce)
scmapCluster_results <- scmapCluster(projection = sce_test,
        index_list = list(metadata(sce)$scmap_cluster_index))
```

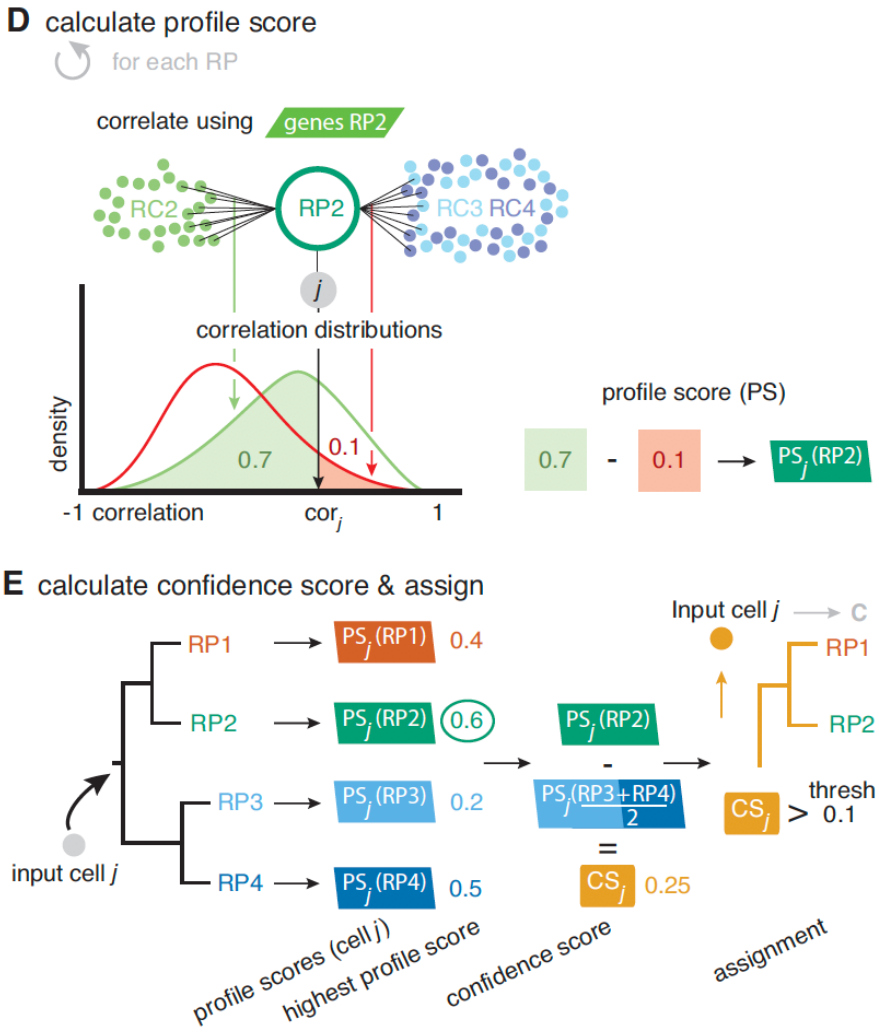
CHETAH

- First, a hierarchical classification tree is constructed from the reference scRNA-seq data
- Selecting the set of genes that best discriminates each reference cell type from all the cell types, collectively, in the opposite branch of the tree



CHETAH

- Calculate profiles score calculated from the position of input cell's correlation within these two reference cell distributions
- The confidence score is calculated as the difference of the highest profile score in chosen the branch and the average of profile scores in the other branch
- Cells do not meet confidence threshold will be labeled as ***unassigned*** if the evidence runs out at the top of the tree, or as ***intermediate*** if this happens within the classification tree



Example code for CHETAH

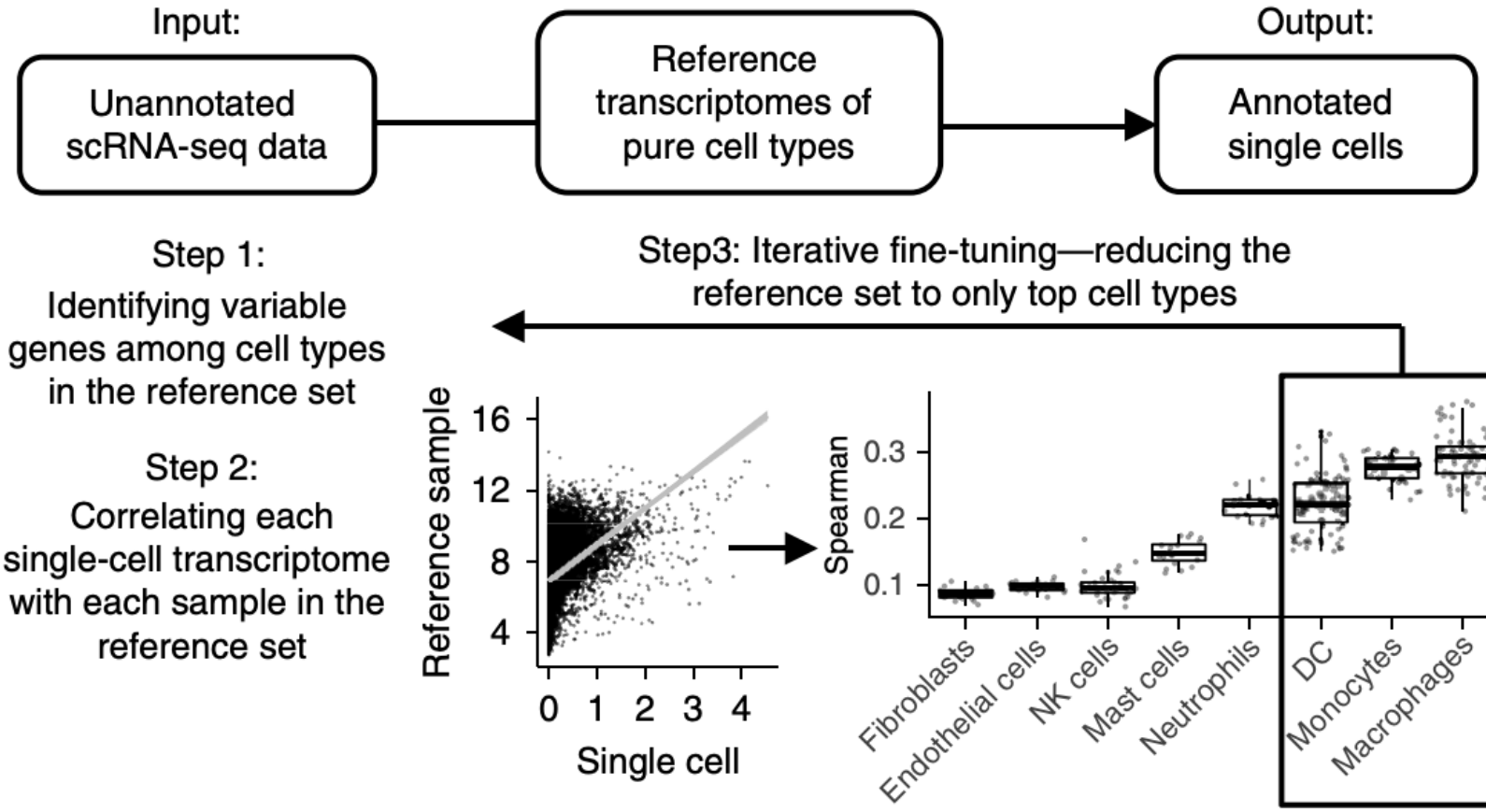
```
sce_train <- SingleCellExperiment(assays =
    list(counts = as.matrix(trainmat)),
        colData = DataFrame(celltypes=trainlabel))

sce_test <- SingleCellExperiment(assays =
    list(counts = as.matrix(testmat)),
         colData = DataFrame(celltypes = testlabel))

#run classifier
test <- CHETAHclassifier(input = sce_test,
                         ref_cells = sce_train)
test$celltype_CHETAH
```

SingleR

- Correlation based annotation
- Allow the use of bulk or scRNA-seq data as the reference
- Has a built-in reference from Human Primary Cell Atlas

SingleR

platforms all | rank 132 / 1974 | posts 8 / 1 / 3 / 1 | in Bioc 1.5 years
build warnings | updated < 1 month | dependencies 46



Example code for SingleR

```
# use pre-built reference data
library(celldex)
hpca.se <- HumanPrimaryCellAtlasData()
library(SingleR)
pred.hesc <- SingleR(test = hESCs, ref = hpca.se,
assay.type.test=1, labels = hpca.se$label.main)

# build reference data by ourselves
# SingleR() expects reference datasets to be normalized and log-
transformed.
library(scuttle)
sceM <- logNormCounts(sceM)
sceG <- sceG[,colSums(counts(sceG)) > 0] # Remove libraries with
no counts. sceG <- logNormCounts(sceG)
pred.grun <- SingleR(test=sceG, ref=sceM, labels=sceM$label,
de.method="wilcox")
```

Comparison of the methods

Abdelaal *et al. Genome Biology* (2019) 20:194
https://doi.org/10.1186/s13059-019-1795-z

Genome Biology

RESEARCH Open Access

A comparison of automatic cell identification methods for single-cell RNA sequencing data

Tamim Abdelaal[1,2†], Lieke Michielsen[1,2†], Davy Cats[3], Dylan Hoogduin[3], Hailiang Mei[3], Marcel J. T. Reinders[1,2] and Ahmed Mahfouz[1,2*]

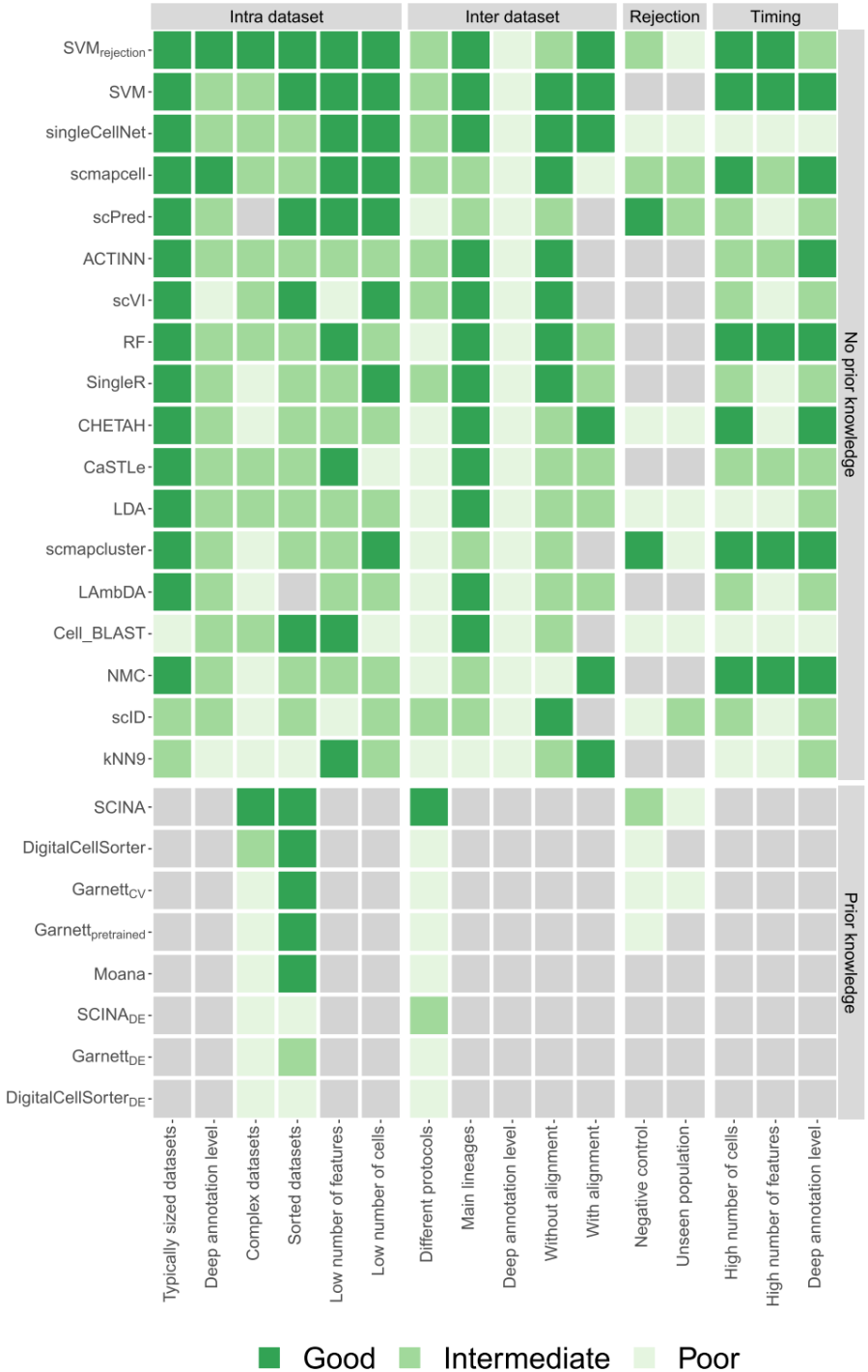

Intra dataset
Inter dataset
Rejection
Timing
SVM$_{rejection}$
SVM
singleCellNet
scmapcell
scPred
ACTINN
scVI
RF
SingleR
CHETAH
CaSTLe
LDA
scmapcluster
LAmbDA
Cell_BLAST
NMC
scID
kNN9
SCINA
DigitalCellSorter
Garnett$_{CV}$
Garnett$_{pretrained}$
Moana
SCINA$_{DE}$
Garnett$_{DE}$
DigitalCellSorter$_{DE}$
No prior knowledge
Prior knowledge
Typically sized datasets
Deep annotation level
Complex datasets
Sorted datasets
Low number of features
Low number of cells
Different protocols
Main lineages
Deep annotation level
Without alignment
With alignment
Negative control
Unseen population
High number of cells
High number of features
Deep annotation level
Good
Intermediate
Poor

Table 1 Automatic cell identification methods included in this study

Name	Version	Language	Underlying classifier	Prior knowledge	Rejection option	Reference
Garnett	0.1.4	R	Generalized linear model	Yes	Yes	[14]
Moana	0.1.1	Python	SVM with linear kernel	Yes	No	[15]
DigitalCellSorter	GitHub version: e369a34	Python	Voting based on cell type markers	Yes	No	[16]
SCINA	1.1.0	R	Bimodal distribution fitting for marker genes	Yes	No	[17]
scVI	0.3.0	Python	Neural network	No	No	[18]
Cell-BLAST	0.1.2	Python	Cell-to-cell similarity	No	Yes	[19]
ACTINN	GitHub version: 563bcc1	Python	Neural network	No	No	[20]
LAmbDA	GitHub version: 3891d72	Python	Random forest	No	No	[21]
scmapcluster	1.5.1	R	Nearest median classifier	No	Yes	[22]
scmapcell	1.5.1	R	kNN	No	Yes	[22]
scPred	0.0.0.9000	R	SVM with radial kernel	No	Yes	[23]
CHETAH	0.99.5	R	Correlation to training set	No	Yes	[24]
CaSTLe	GitHub version: 258b278	R	Random forest	No	No	[25]
SingleR	0.2.2	R	Correlation to training set	No	No	[26]
scID	0.0.0.9000	R	LDA	No	Yes	[27]
singleCellNet	0.1.0	R	Random forest	No	No	[28]
LDA	0.19.2	Python	LDA	No	No	[29]
NMC	0.19.2	Python	NMC	No	No	[29]
RF	0.19.2	Python	RF (50 trees)	No	No	[29]
SVM	0.19.2	Python	SVM (linear kernel)	No	No	[29]
$SVM_{rejection}$	0.19.2	Python	SVM (linear kernel)	No	Yes	[29]
kNN	0.19.2	Python	kNN ($k = 9$)	No	No	[29]

Obtain existing single cell datasets

- Information from the original papers:

Authors declare no competing interests. **Data and materials availability:** All data are available in the supplement; raw data are available through the Sequence Read Archive, accession number PRJNA434002. Analyzed data and visualization are available at https://autism.cells.ucsc.edu.

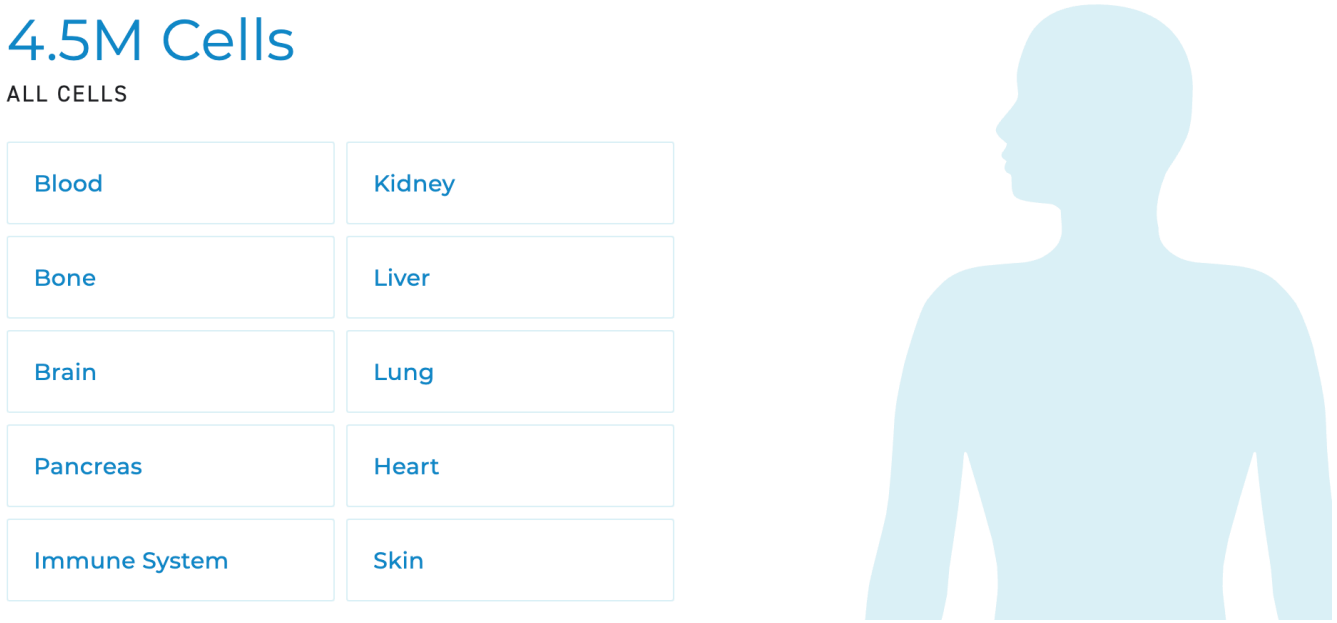
DATA AND SOFTWARE AVAILABILITY

The exome and RNA sequencing files are uploaded to European Genome-Phenome Archive (https://www.ega-archive.org/) and can be accessed using the accession number EGA: EGAS00001002606. Clinical data and all data used for this study are provided in the Supplementary tables. We have developed an interactive webtool (https://dlbcl.davelab.org) for survival analysis using clinical and genomic features.

Resulting fastq files for each sample were deposited in GEO (GSE116256).

Obtain existing single cell datasets

- Human cell atlas (https://data.humancellatlas.org/):



Obtain existing single cell datasets

- Human cell atlas (https://data.humancellatlas.org/):

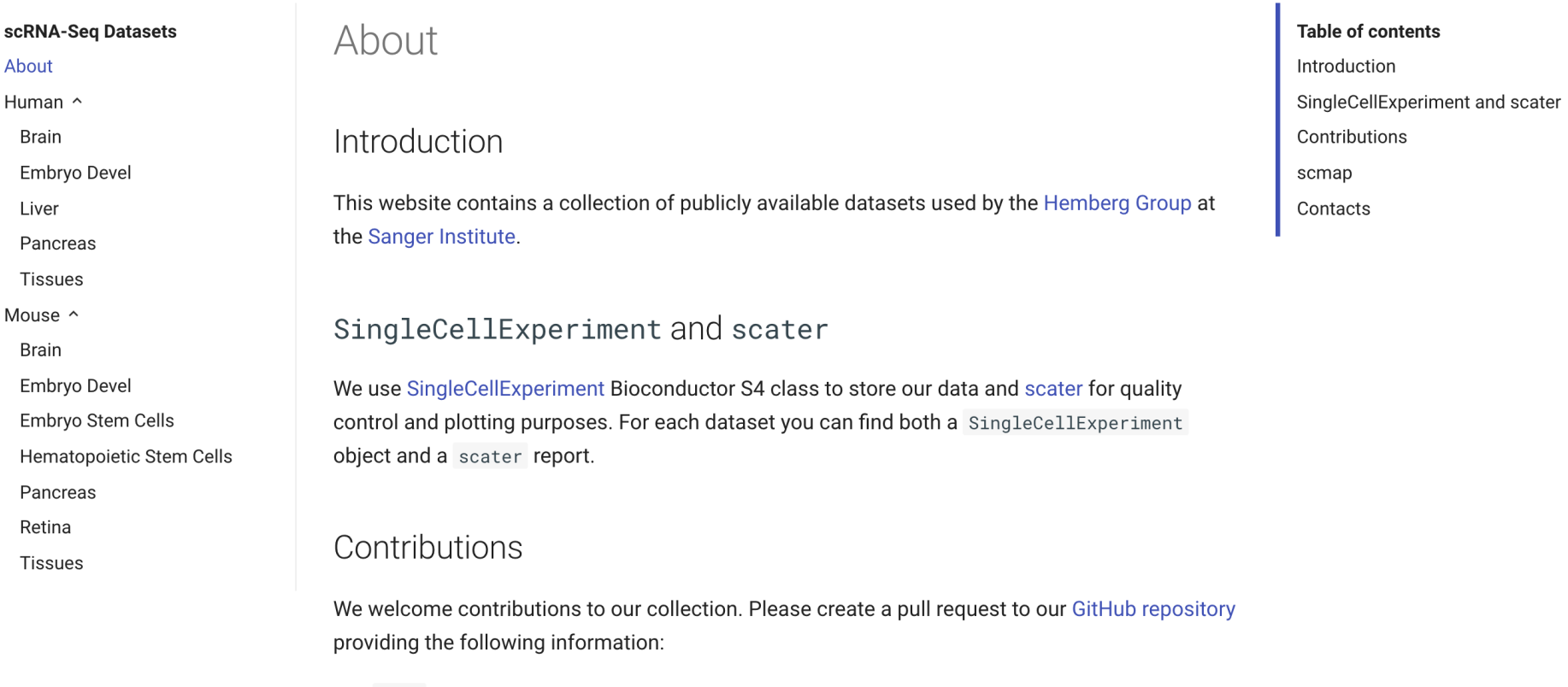
11 Donors 49 Specimens 1.4M Estimated Cells 88.9k Files 4.25 TB File Size

Projects Samples Files

Project Title	Project Downloads		Species	Sample Type	Organ / Model Organ	Selected Cell Type	Library Construction Method
(3)	Metadata	Matrices	(2)	(1)	(1) / (1)	(3)	(9)
1.3 Million Brain Cells from E18 Mice		–	Mus musculus	specimens	brain	neuron	10X v2 sequencing
Systematic comparative analysis of single cell RNA-sequencing methods		–	Homo sapiens, …	specimens	blood, brain	mononuclear c…	10X v2 sequencing, 10x v3 sequencing, CEL-seq2, DroNc-seq, Drop-seq, Seq-Well, Smart-seq2, inDrop, sci-RNA-seq
Tabula Muris: Transcriptomic characterization of 20 organs and tissues from Mus musculus at single cell resolution		–	Mus musculus	specimens	adipose tissue, …	Unspecified	Smart-seq2

Obtain existing single cell datasets

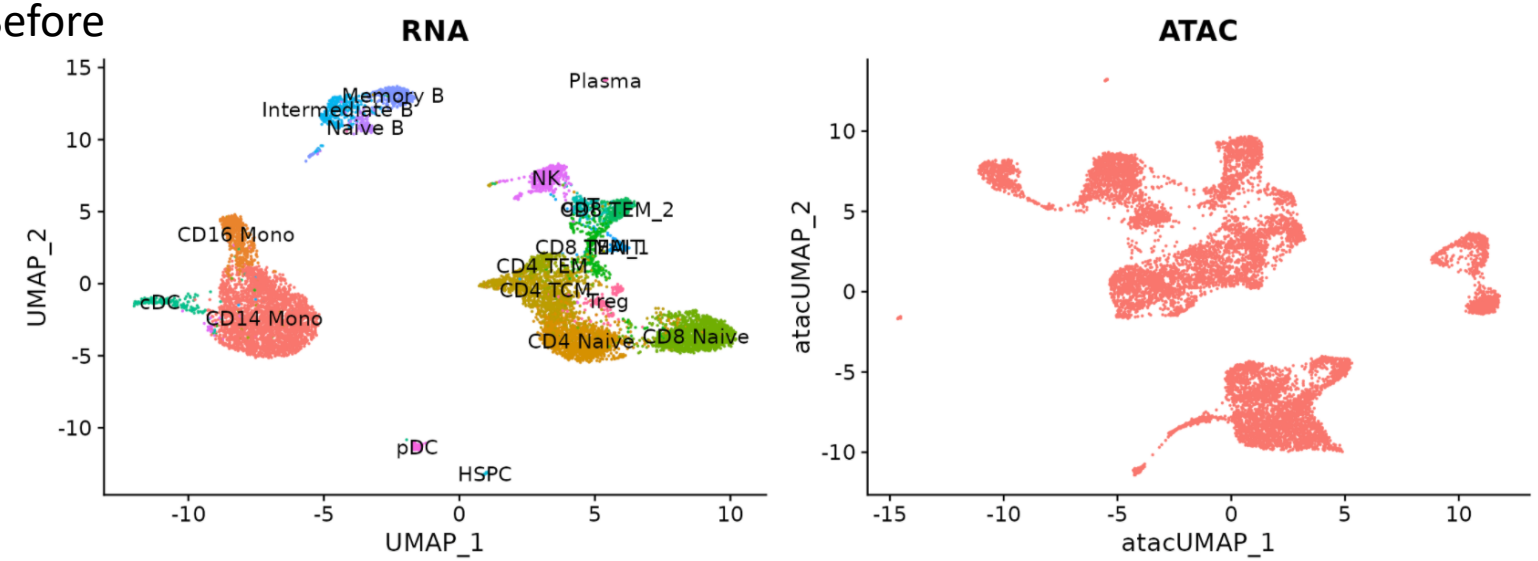
- Website (e.g. https://hemberg-lab.github.io/scRNA.seq.datasets/)



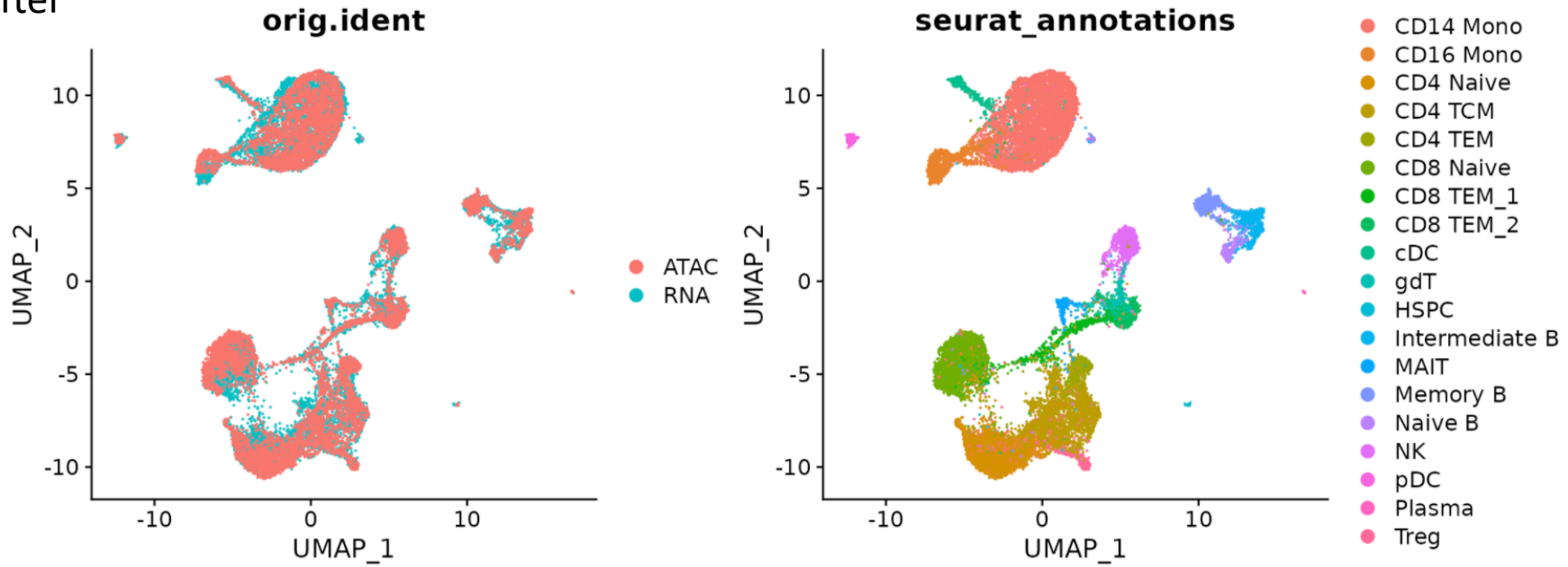
Data integration

- Integrate data from different platforms, conditions, species, etc.
- Similar to batch effect correction, but could be more broad
- Seurat V3: CCA (Cell, 2019)
- LIGER: Non-negative matrix factorization (Cell, 2019)
- Harmony: Shared embedding learning using a modified soft k-means (Nat Methods, 2019)
- scAlign: Shared embedding learning using revied autoencoder (Genome Biology, 2019)
- scMC: variance correction based on technical and biological variation (Genome Biology 2021)

Before

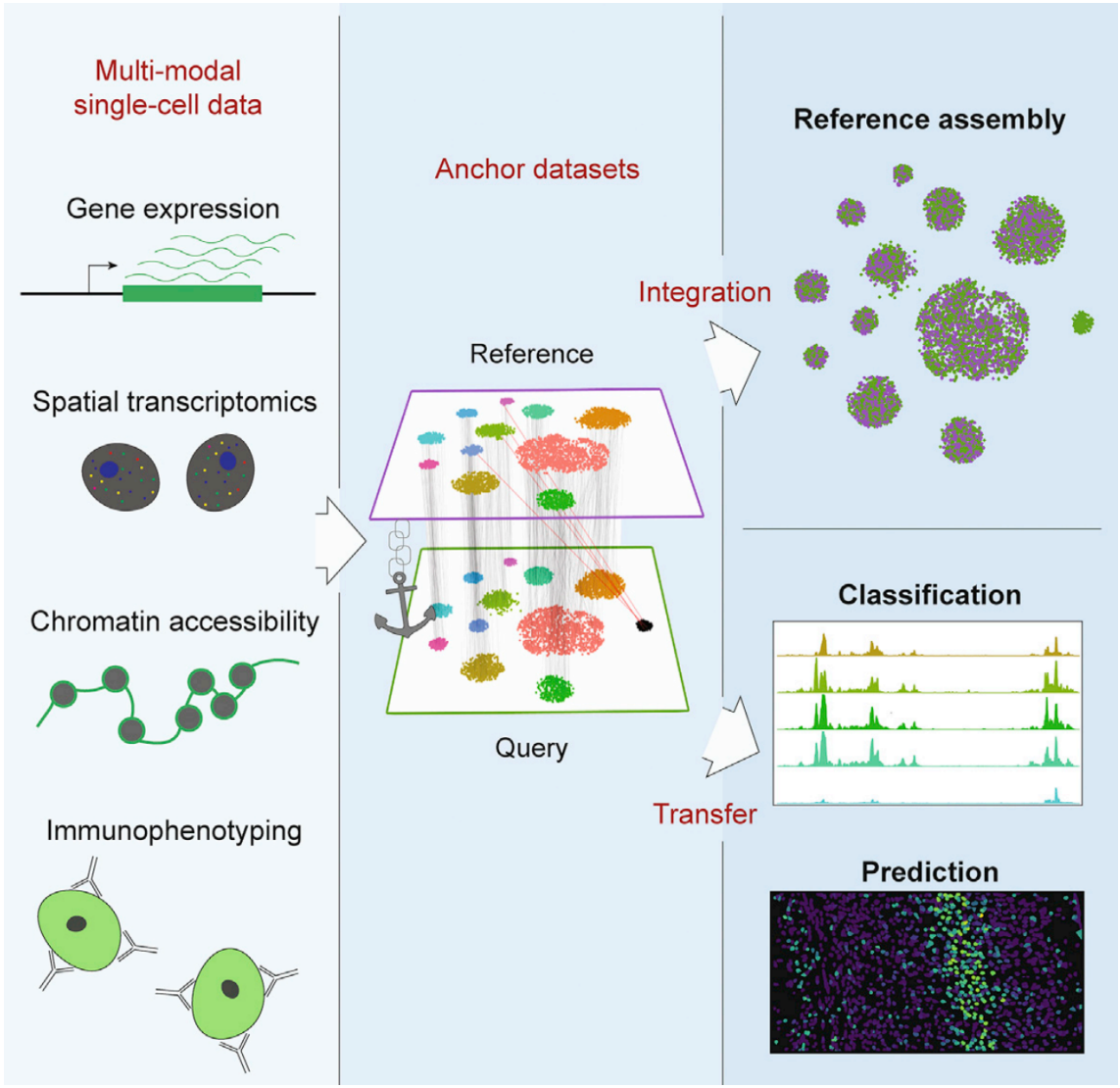


After

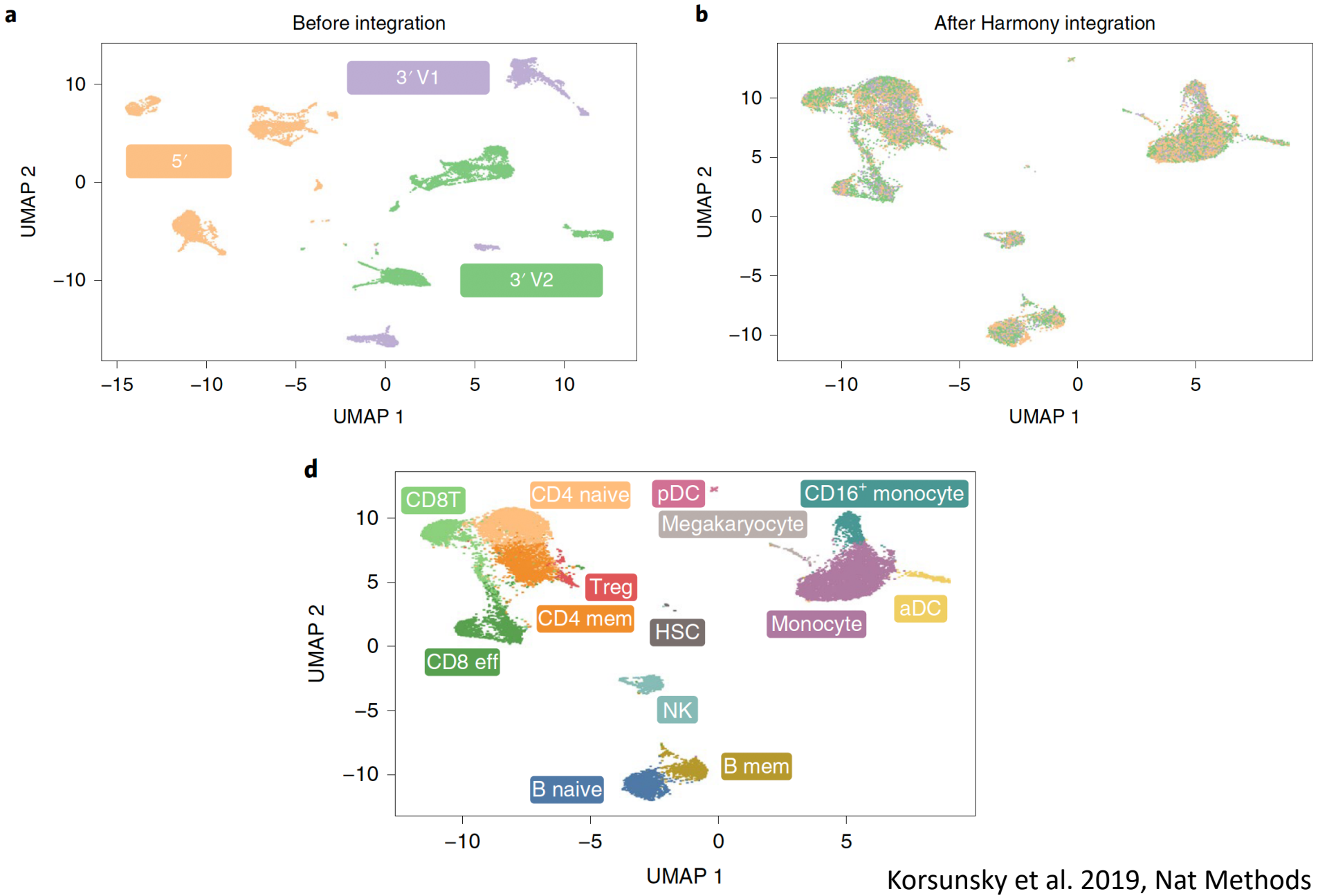


https://satijalab.org/seurat/articles/atacseq_integration_vignette.html

Seurat V3

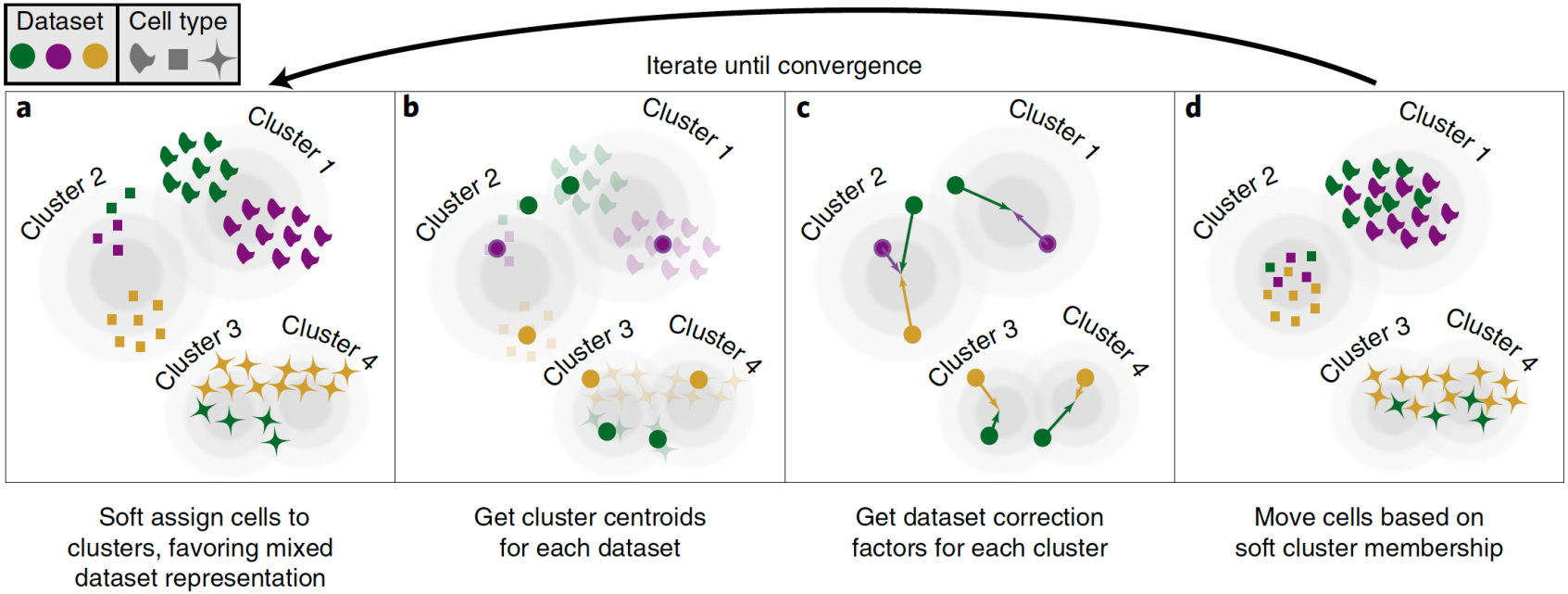


Stuart et al. 2019, Cell



Korsunsky et al. 2019, Nat Methods

Harmony


Dataset
Cell type
Iterate until convergence
a
Cluster 1
Cluster 2
Cluster 3
Cluster 4
b
Cluster 1
Cluster 2
Cluster 3
Cluster 4
c
Cluster 1
Cluster 2
Cluster 3
Cluster 4
d
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Soft assign cells to clusters, favoring mixed dataset representation
Get cluster centroids for each dataset
Get dataset correction factors for each cluster
Move cells based on soft cluster membership