Single Cell Analysis: “Under the Hood” Workshop II

April 10, 2019

Presenters (in order):

Jamie Diemer

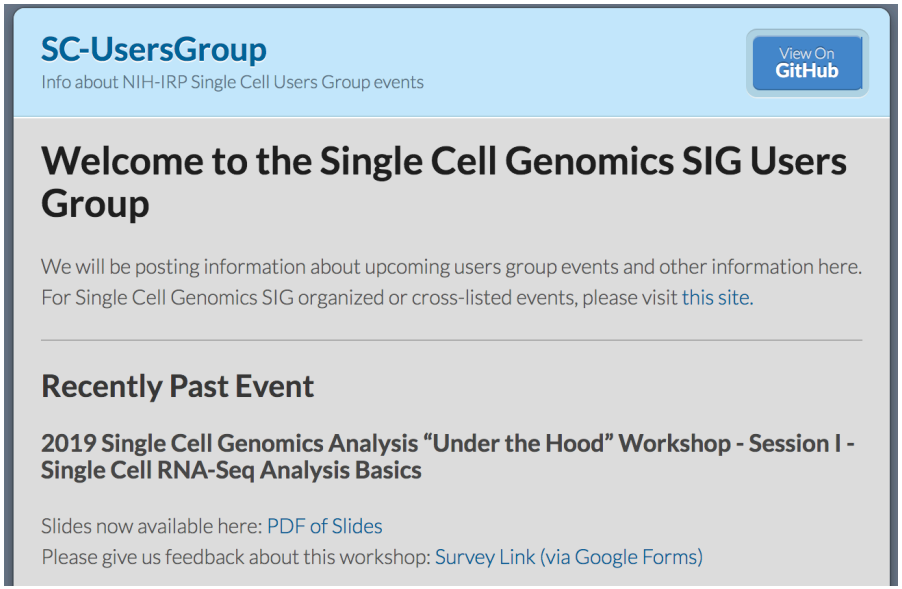
Assaf Magen

Abdalla Abdelmaksoud

Stefan Cordes

NIH Single Cell User's Group

Mike Kelly (NCI)
Jamie Diemer (NHGRI)
Erica Bresciani (NHGRI)
Chen Yao (NIAMS)
Stefan Cordes (NHLBI)
Lingling Miao (NIAMS)
Ben Voisin (NIAMS)
Supreet Agarwal (NCI)
Byunghyun Kang (NIAID)



https://nih-irp-singlecell.github.io/SC-UsersGroup/

- **Data presentations (2 x 20) and Discussion Sessions**
- **Join the Single Cell Genomics Listserv to receive our notifications (instructions on the Github page)**

Thanks for attending the first workshop!

- 241 registrants on EventBrite
- At or over capacity in a 160 seat conference room
- 121 registered on WebEx

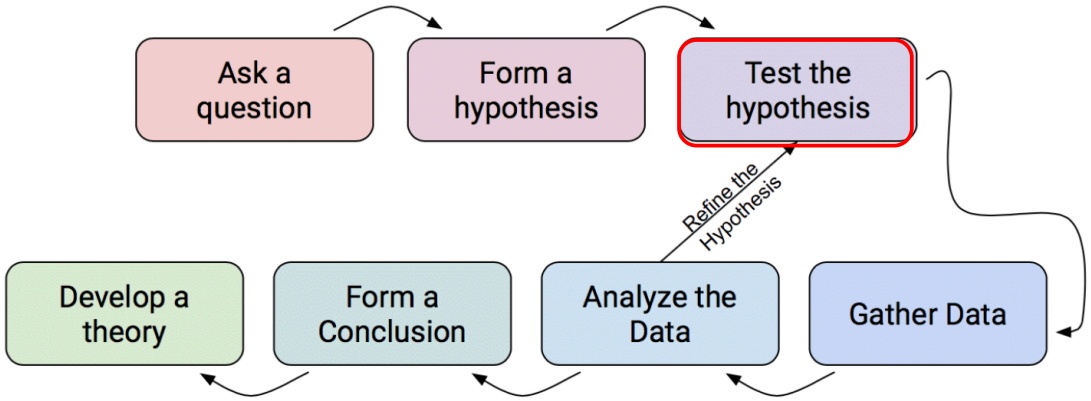
We use your feedback to guide future events

- User's Group meeting discussion topics
- Hands-on training needs of the community

Broad Outline

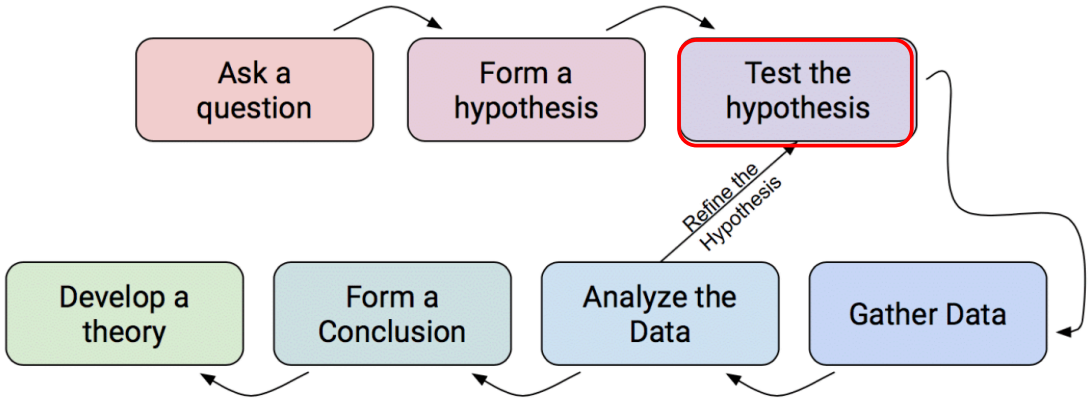
- Introduction and Objectives
- Experimental Design
- Analyzing multiple experimental conditions: methods for mitigating batch effects (technical variation)
- Integration of multiple datasets
- Introduction to trajectory analyses

Experimental Design



Single cell transcriptomic technologies are powerful tools to examine cellular heterogeneity

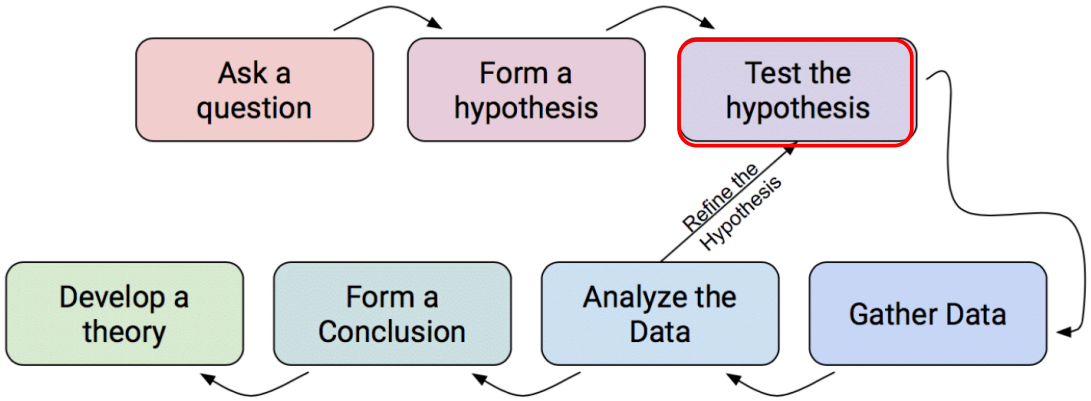
Experimental Design



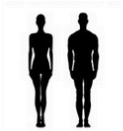
Single cell transcriptomic technologies are powerful tools to examine cellular heterogeneity

- Biological variation
- Technical variation

Experimental Design

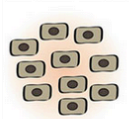


Today, part of our goal is to show you ways to address technical variation…but you can also minimize it up front with good experimental design

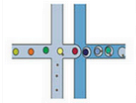


Tissue harvesting

Tissue dissociation



Cell enrichment (optional)



Single Cell RNA-Sequencing platform

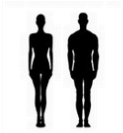
Library sequencing

Computational analysis

Pipeline for single cell experiments

- Considerations for experimental design at every step

Nguyen et al., 2018

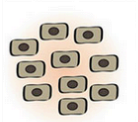


Tissue harvesting

- Biological replicates
- Technical replicates

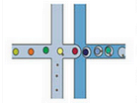
Tissue dissociation

- Enzymatic, manual
- Consistency



Cell enrichment (optional)

- FACS
- Dead cell removal



Single Cell RNA-Sequencing platform

- Droplet-based
- Plate-based
- Whole transcriptome vs 3’ counting

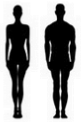
Library sequencing

- Depth of sequencing

Computational analysis

- Separation of biological and technical variation

Nguyen et al., 2018

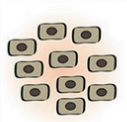


Tissue harvesting

- Biological replicates
- Technical replicates

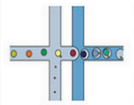
Tissue dissociation

- Enzymatic, manual
- Consistency



Cell enrichment (optional)

- FACS
- Dead cell removal

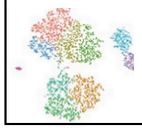


Single Cell RNA-Sequencing platform

- Droplet-based
- Plate-based
- Whole transcriptome vs 3’ counting

Library sequencing

- Depth of sequencing



Computational analysis

- Separation of biological and technical variation

Nguyen et al., 2018

Single Cell Experimental Design: A few references

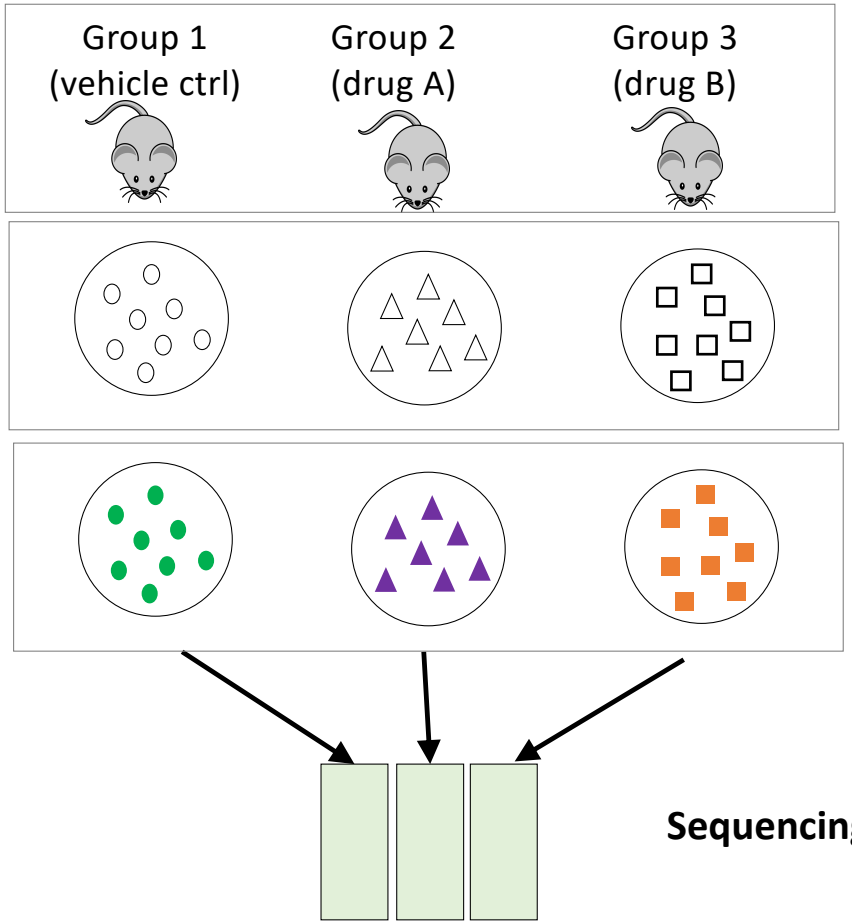
Nguyen et al. “Experimental Considerations for Single-Cell RNA Sequencing Approaches. Frontiers in Cell and Developmental Biology. 2018 (6:108).

Lafzi et al. “Tutorial: guidelines for the experimental design of single-cell RNA sequencing studies”. Nature Protocols. 2018: 13(2742-2757).

J Baran-Gale et al. “Experimental Design for Single-Cell RNA Sequencing”. Briefings in Functional Genomics 2018;**17**:233-239.

Hicks et al. “Missing Data and Technical Variability in Single-Cell RNA-Sequencing Experiments. Biostatistics 19(2018):562-578.

Confounded (unbalanced) Experimental Design

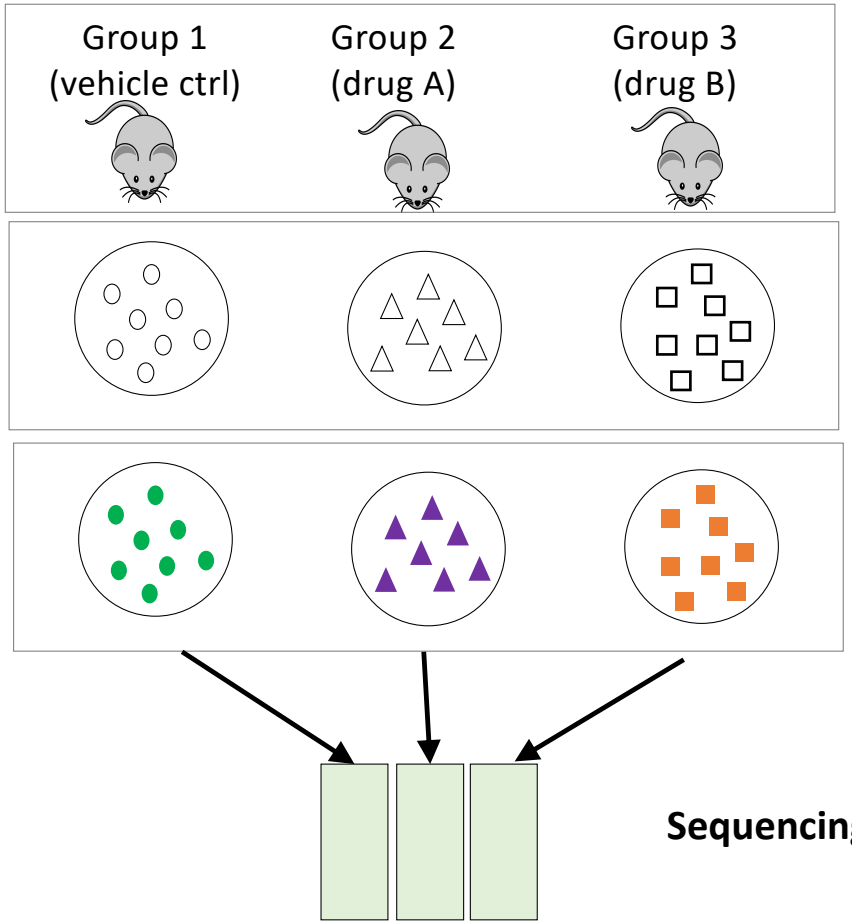


10X capture well (cDNA)

Prepared libraries (batches)

Sequencing lanes

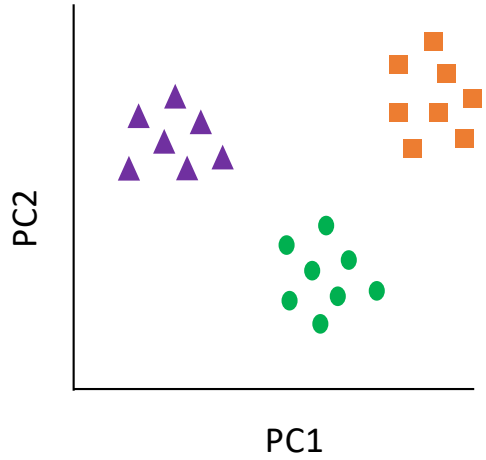
Confounded (unbalanced) Experimental Design



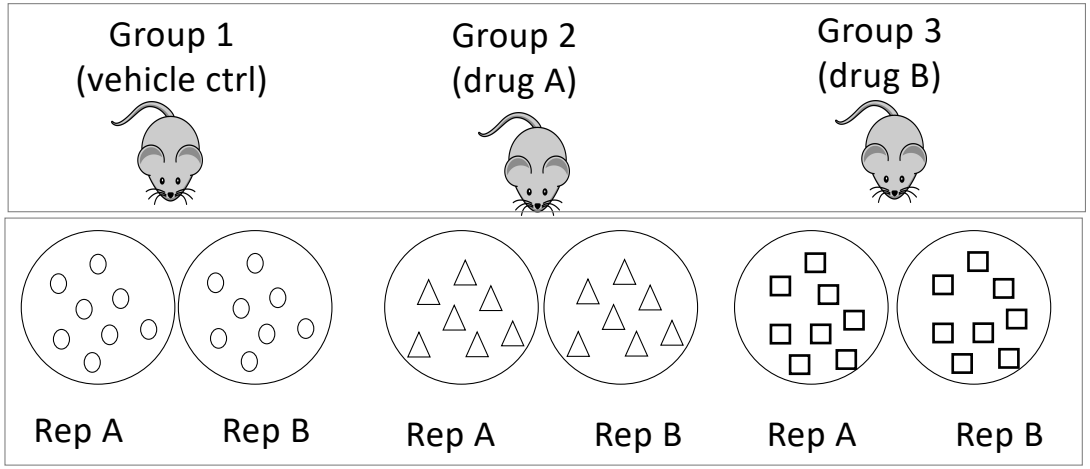
10X capture well (cDNA)

Prepared libraries (batches)

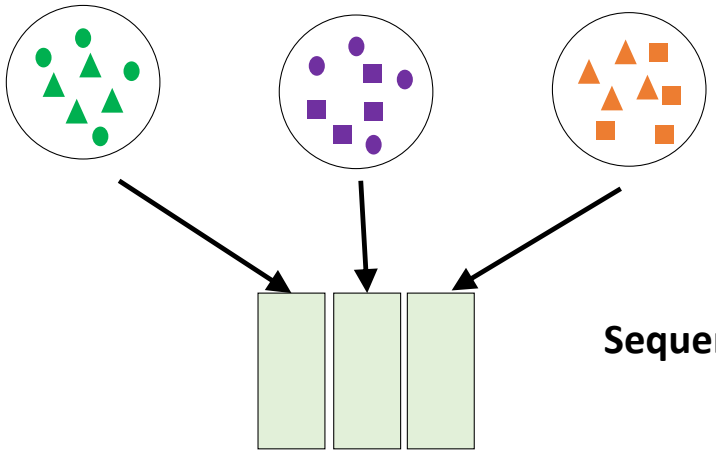
Sequencing lanes



Balanced Experimental Design



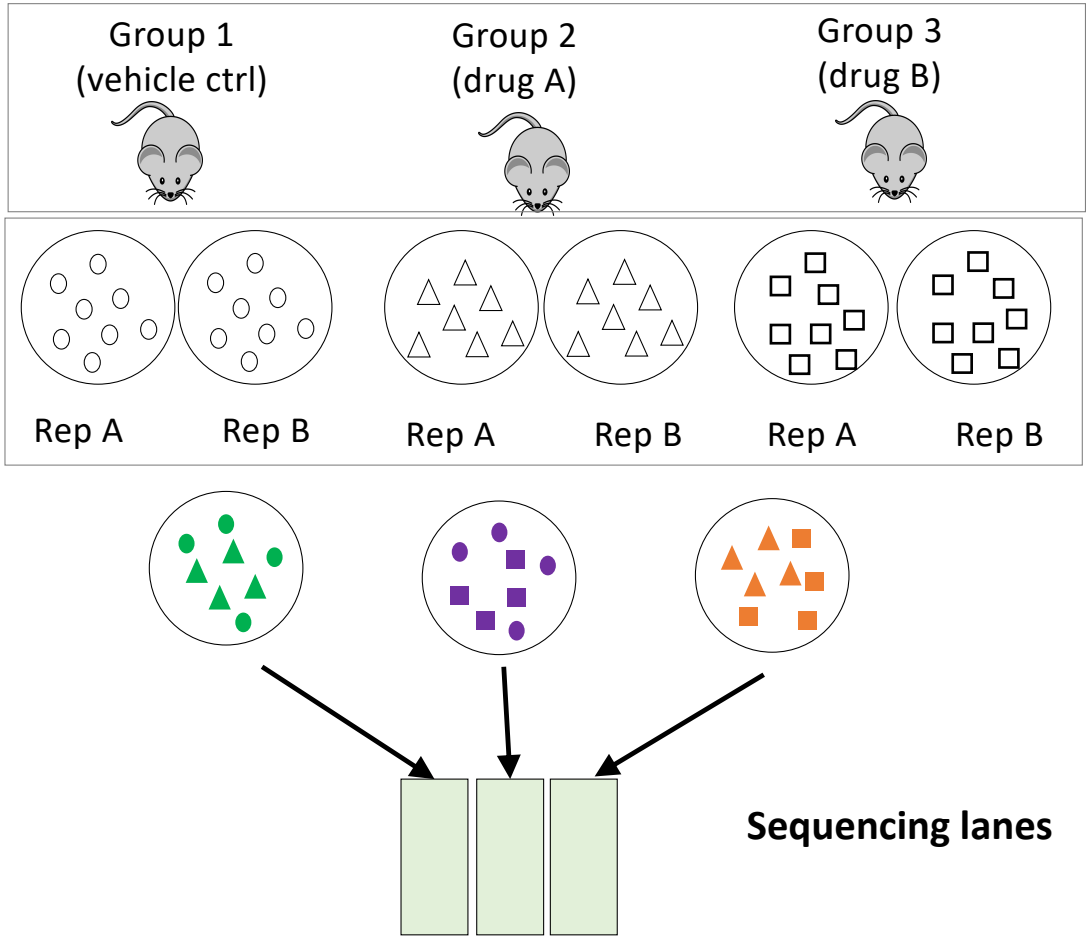
Technical replicates
10X capture wells
(cDNA)

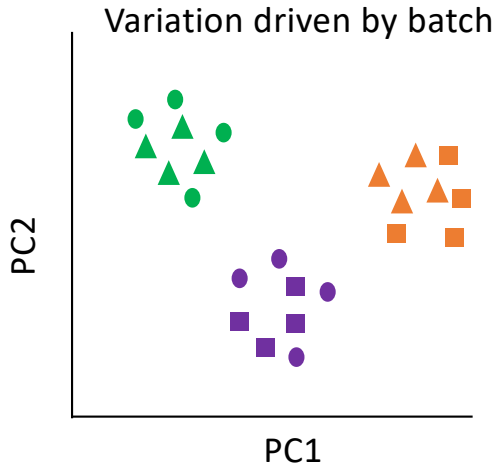


Multiplexed libraries

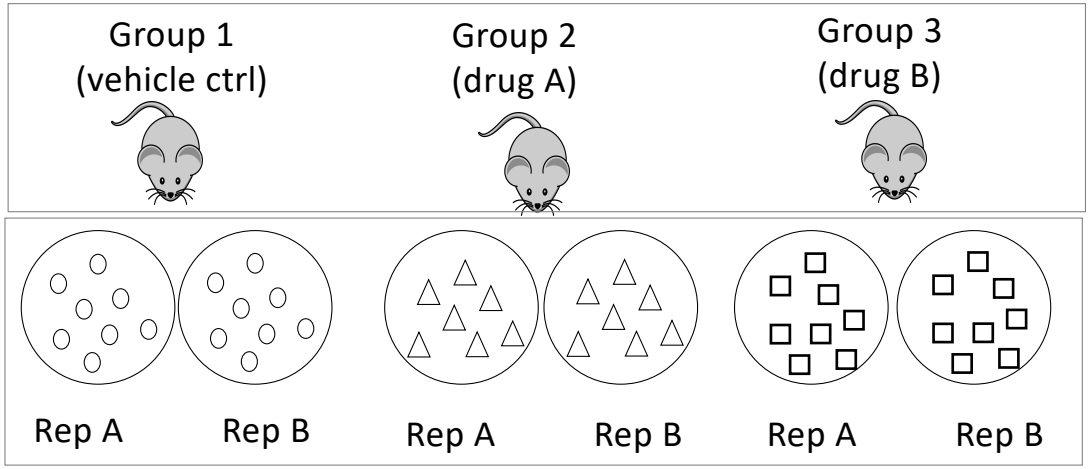
Sequencing lanes

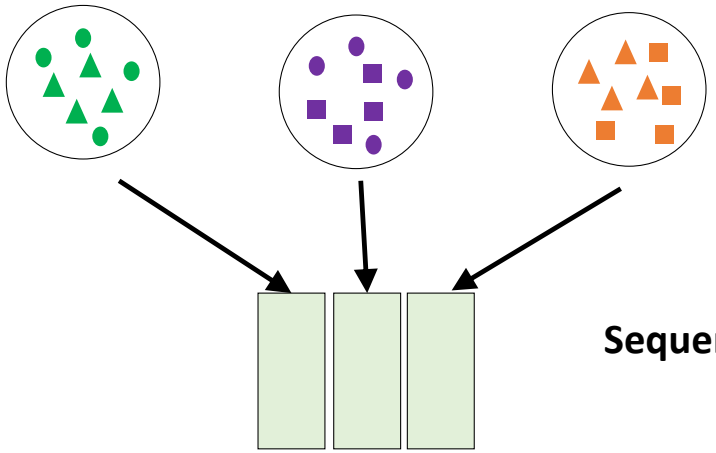
Balanced Experimental Design



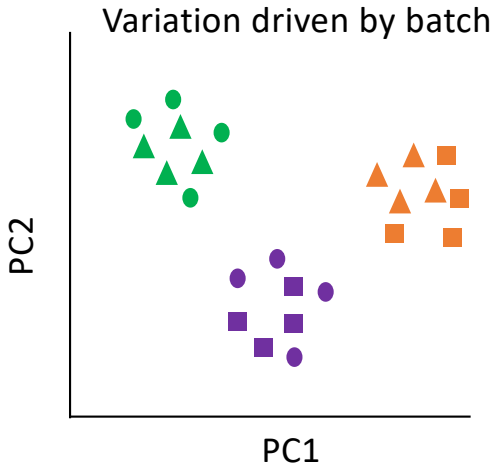


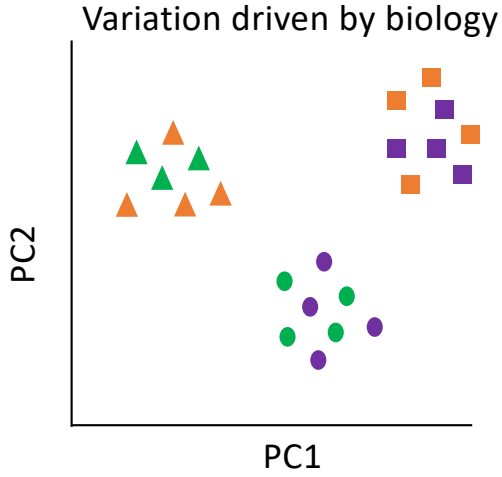
Balanced Experimental Design





Sequencing lanes





Batch effect

Technical differences when samples are processed and measured in different batches and do not correlate to any true biological variation

Beyond confounded experimental design…

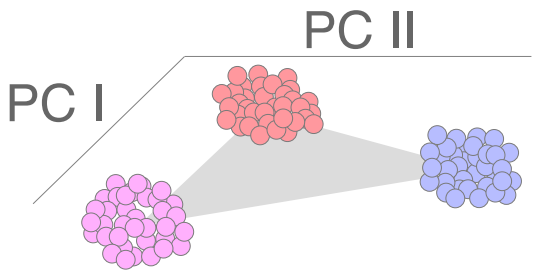
- Using different FACS antibodies from one patient to another…
- One sample took 2 hours to process, another took 10 minutes…
- Surgical technique…
- Comparing your data to someone else's (published datasets or collaborations)

Batch effects hamper meaningful interpretation of the data

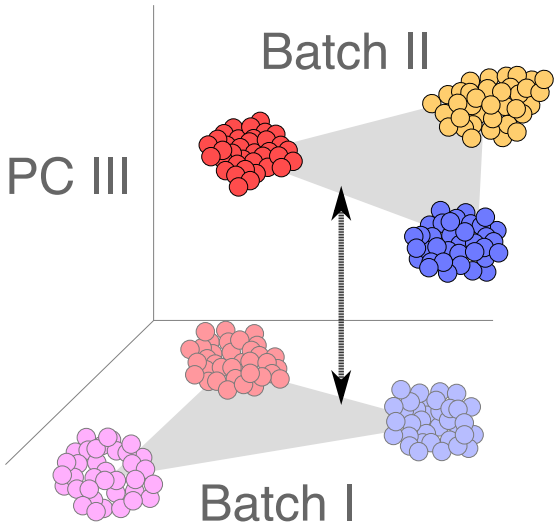
- Removing batch effects is important
- Integrating data from multiple sources
- Overview of available batch correction tools

Assaf Magen

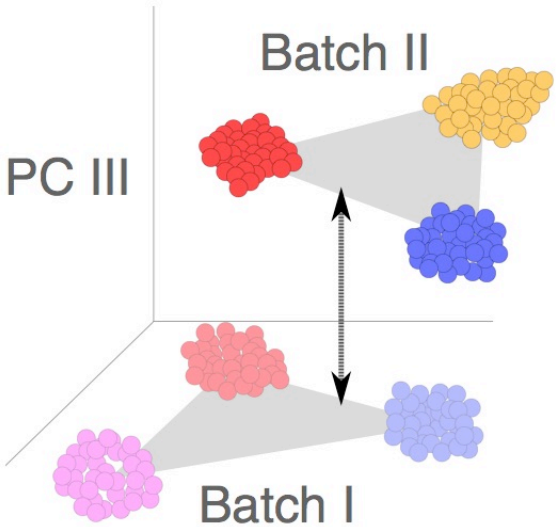
Batch effects hamper meaningful interpretation of the data

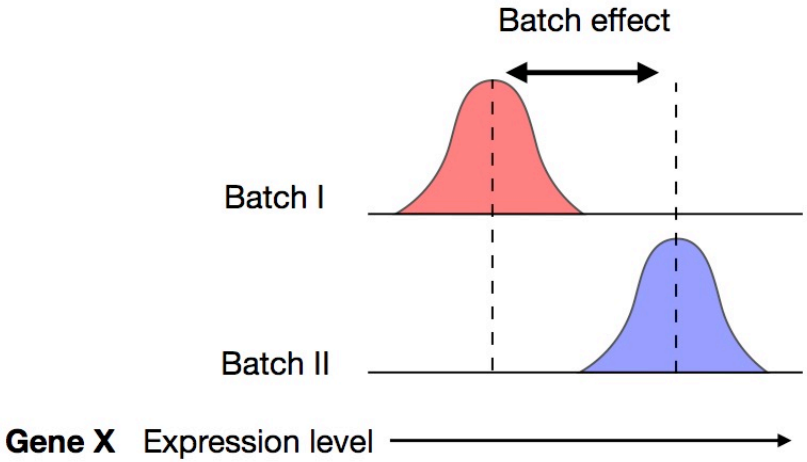


Batch effects hamper meaningful interpretation of the data

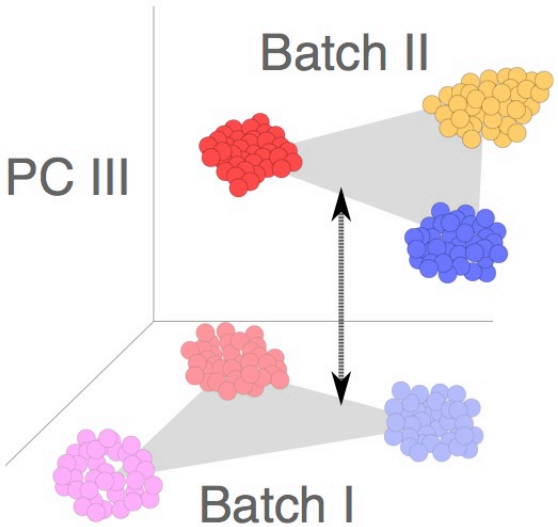


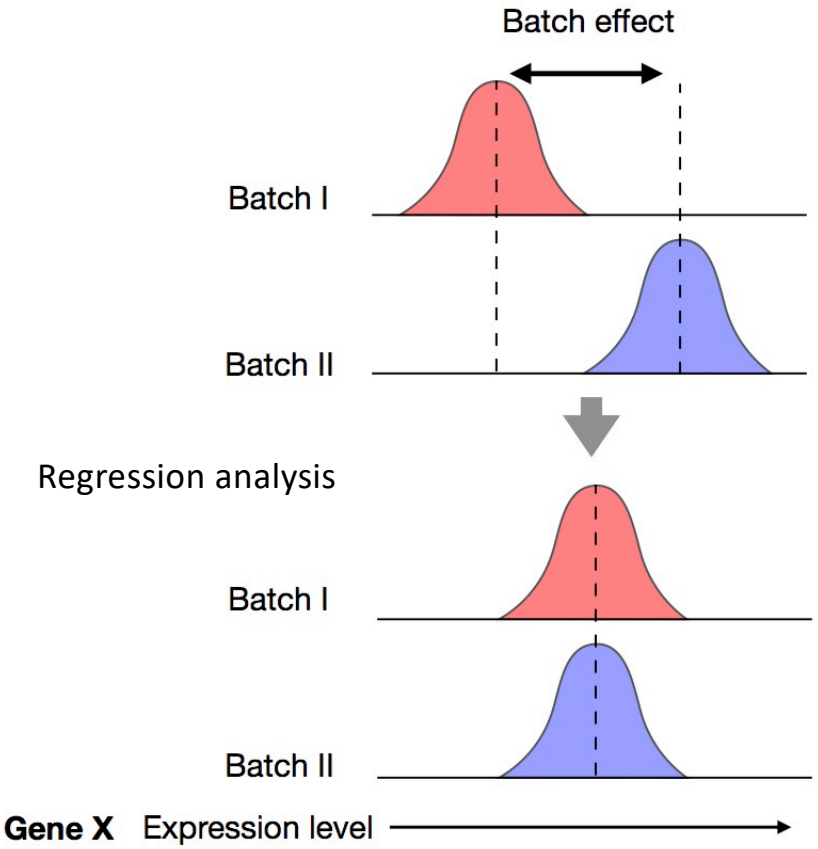
Conventional batch correction approaches





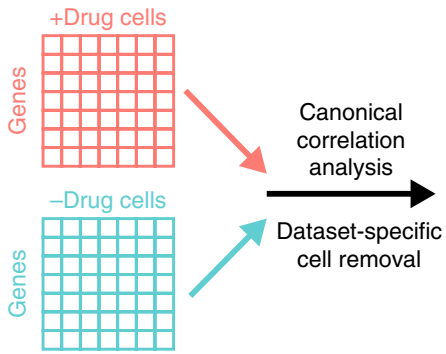
Conventional batch correction approaches





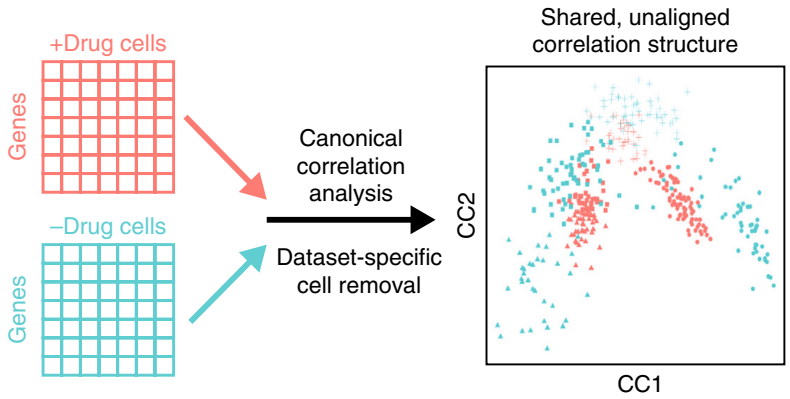
Adapted from Haghverdi et al., 2018

Mapping of cell types across experimental batches



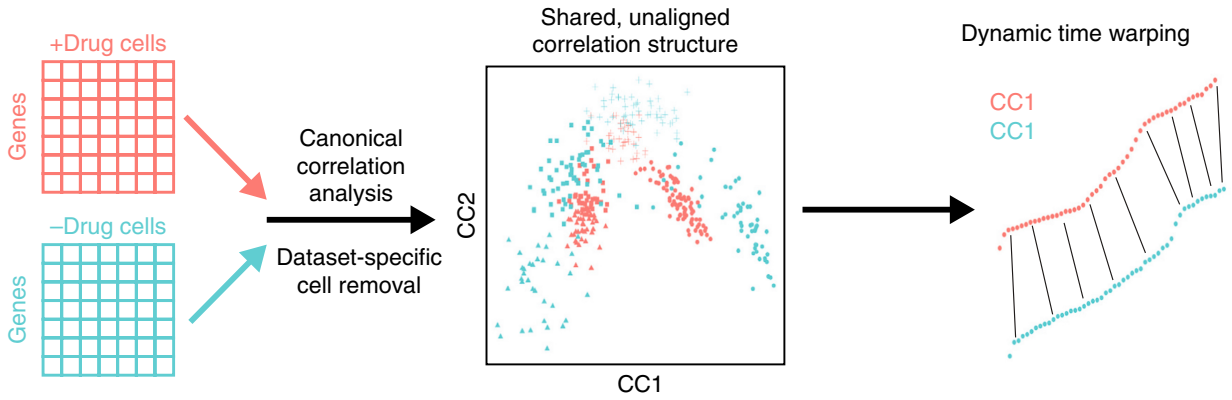
Butler et al., 2018

Mapping of cell types across experimental batches



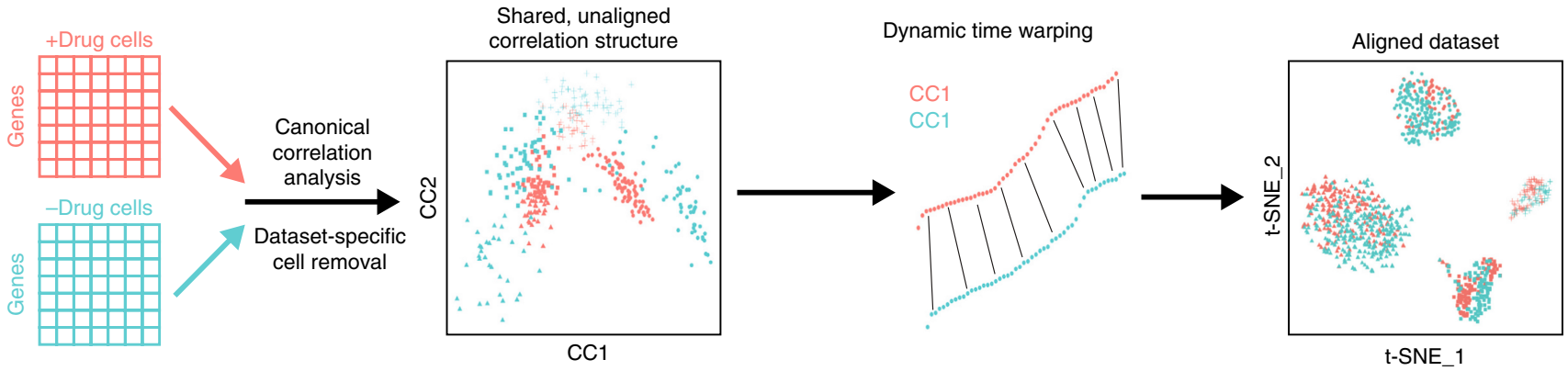
Butler et al., 2018

Mapping of cell types across experimental batches



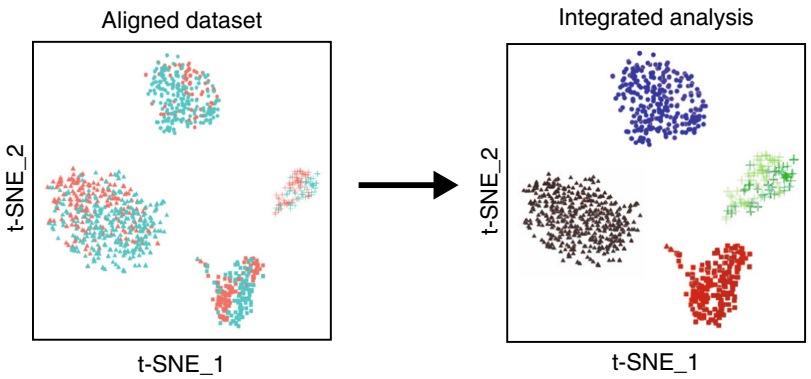
Butler et al., 2018

Mapping of cell types across experimental batches



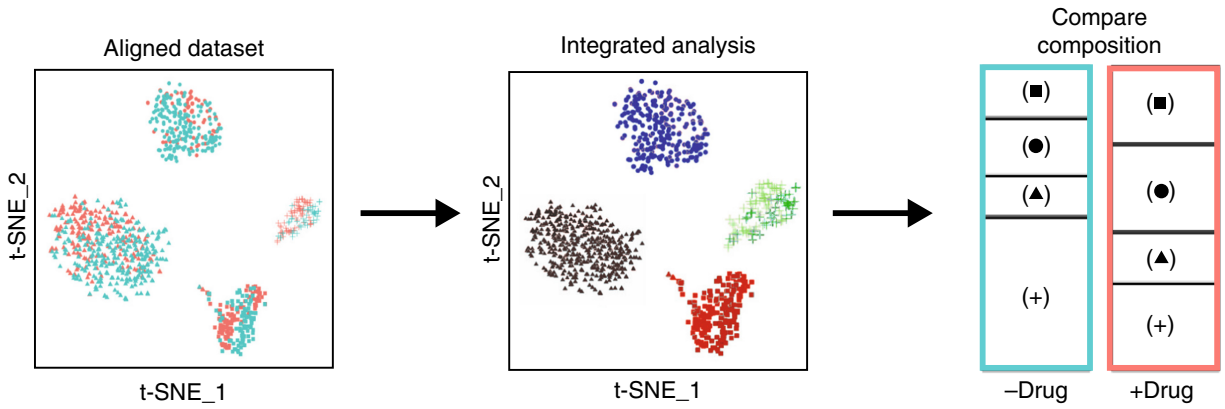
Butler et al., 2018

Comparative analysis of integrated data



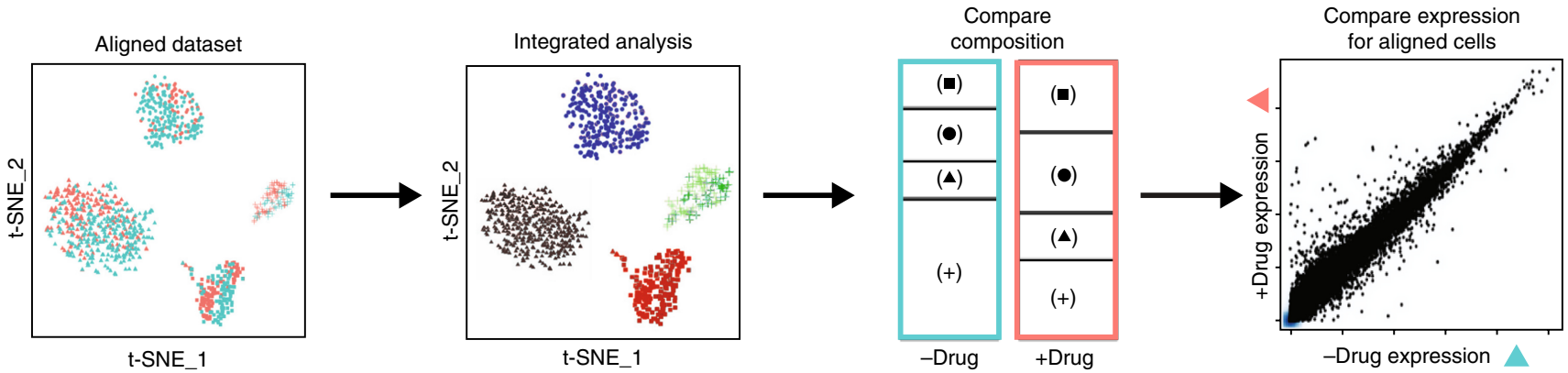
Butler et al., 2018

Comparative analysis of integrated data



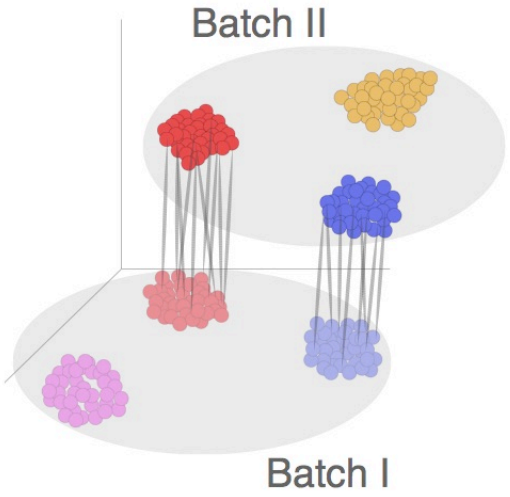
Butler et al., 2018

Comparative analysis of integrated data



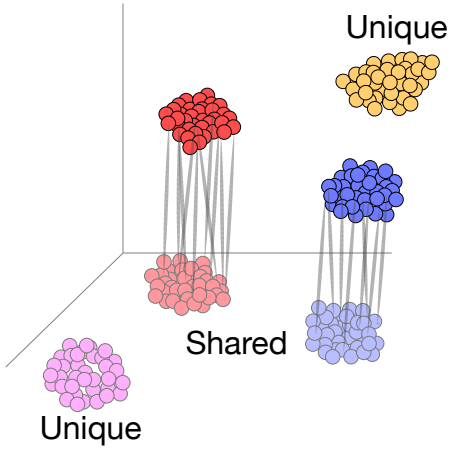
Butler et al., 2018

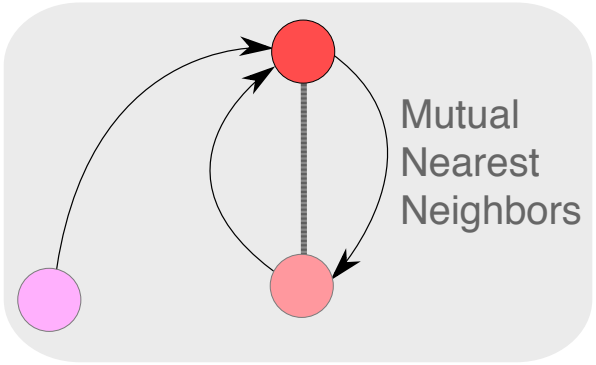
Anchor-based scRNAseq batch correction



Haghverdi et al., 2018

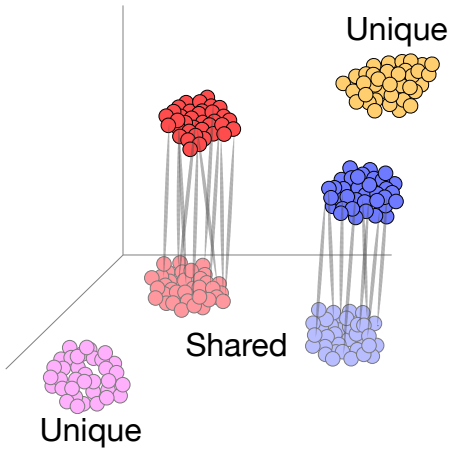
Anchor-based scRNAseq batch correction

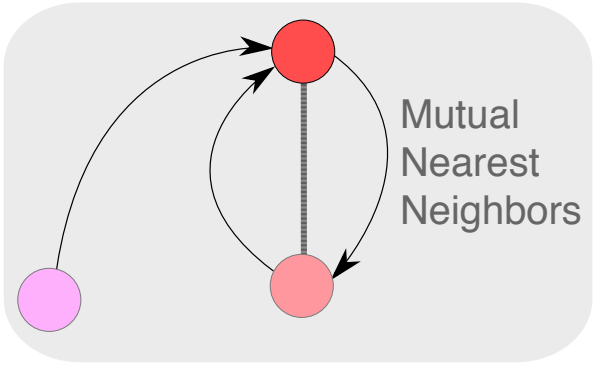


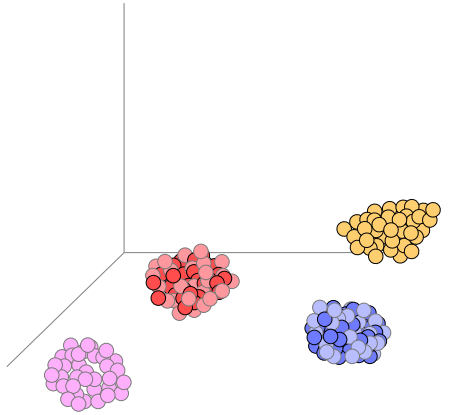


Haghverdi et al., 2018

Anchor-based scRNAseq batch correction

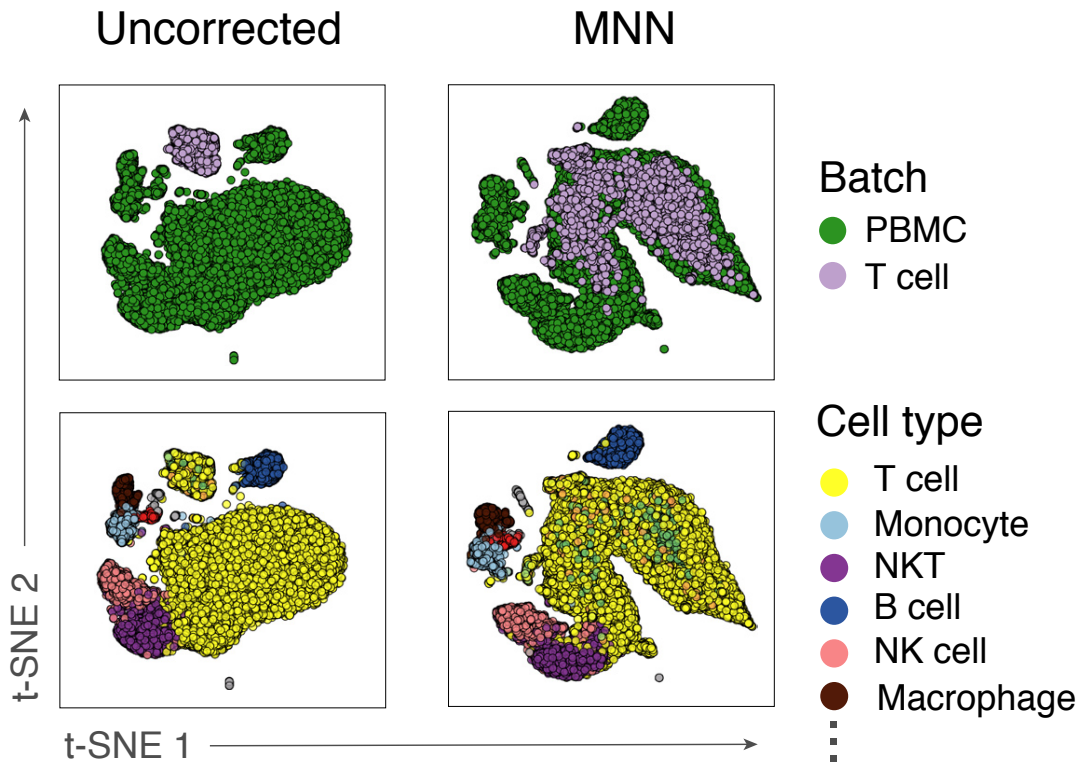






Haghverdi et al., 2018

MNN mapping of distinct lymphocyte data



Adapted from Haghverdi et al., 2018

Regression – Mark-Down

```
## Regress batch
```{r}
then <- Sys.time()

if(!file.exists(file.path(cache.data.dir, "regress_merged_murine_so.RData")) || regenerate) {
  # Assemble phenotypic data about the cells
  # ... Extract the metadata from the Seurat object
  pD <- merged_murine_so@meta.data

  # Assemble expression data
  eD <- GetAssayData(merged_murine_so)

  # Create the model and null model matrices
  mod <- model.matrix(~treatment, data = pD)

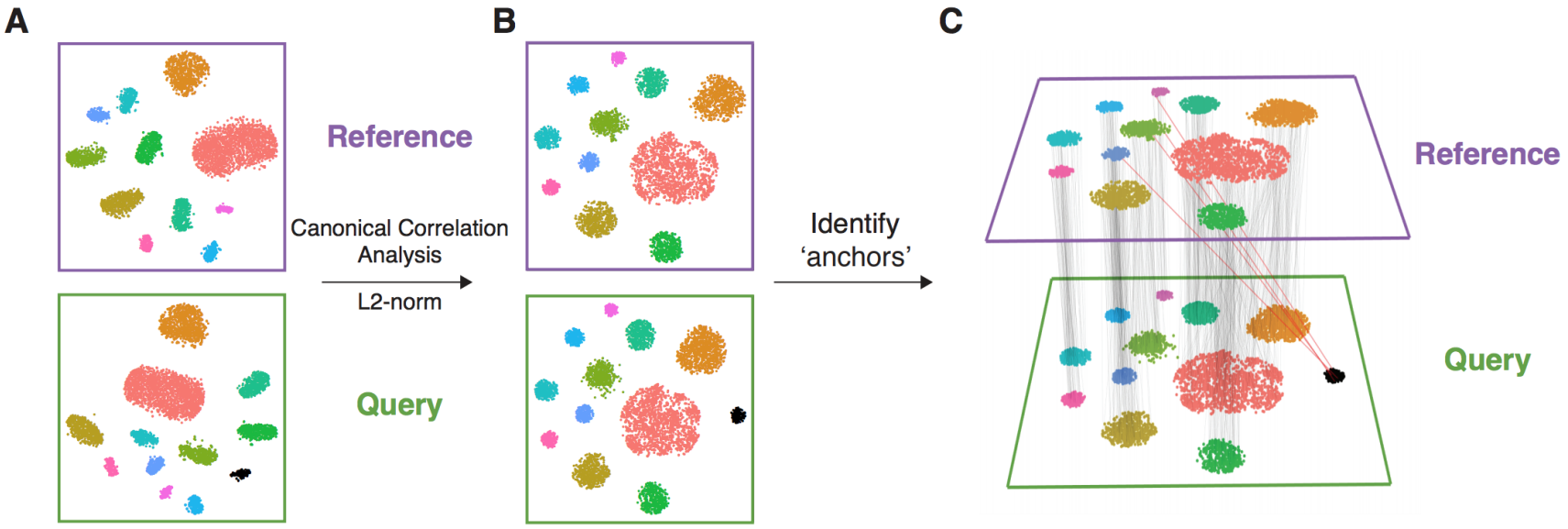
  # Filter genes to those with positive variance
  # genes.use <- names(which(apply(eD, 1, var) > 0))

  # Apply ComBat function to the data, using parametric empirical Bayesian adjustments
  combat_eD <- sva::ComBat(dat = as.matrix(eD),
                           batch = factor(pD$source),
                           mod = mod,
                           par.prior = TRUE,
                           prior.plots = TRUE)

  # Trim extreme values and prepare to resparsify
  max.out <- ceiling(max(eD) + 1)
  combat_eD[combat_eD < sparsify.threshold] <- 0
  combat_eD[combat_eD > max.out] <- max.out

  # Replace data with batch-corrected data
  regression_corrected_merged_murine_so <- SetAssayData(object = merged_murine_so,
                                                        slot = "counts",
                                                        new.data = as(round(combat_eD, digits = 4), "dgCMatrix"),
                                                        assay = "RNA")
  save(regression_corrected_merged_murine_so, file = file.path(cache.data.dir, "regress_merged_murine_so.RData"))
} else {
  load(file = file.path(cache.data.dir, "regress_merged_murine_so.RData"))
}

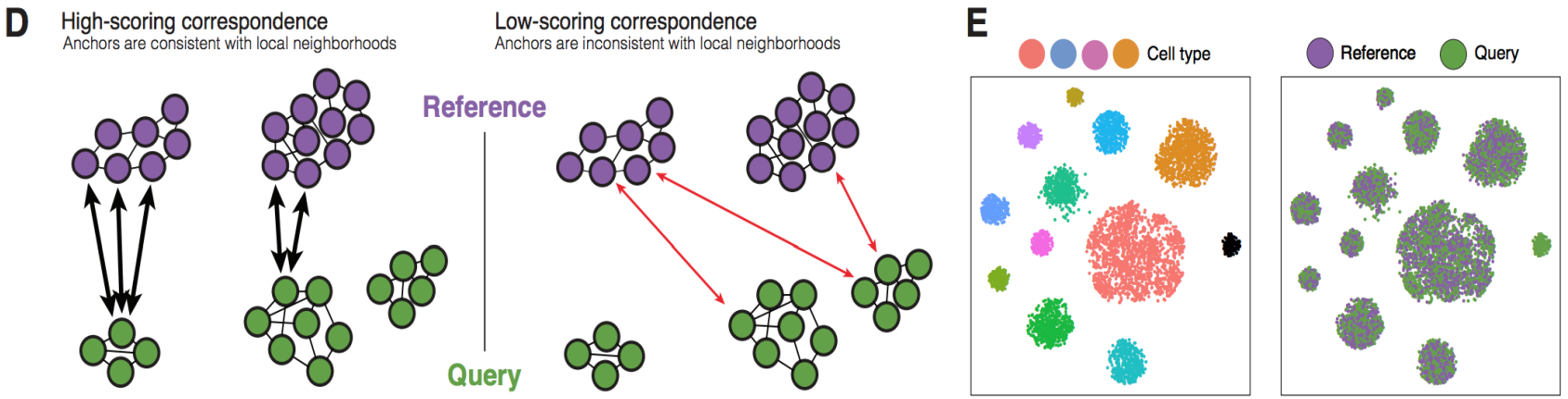
now <- Sys.time()
print(now - then)
```
```



Similar data sets with unique population "black"

Perform CCA followed by L2-normalization on the CC vectors

In CC shared space to identify MNN to serve as anchors (some incorrect)



Incorrect anchors tend to have low scores for neighborhood structure consistency

After filtering low scoring anchors

The Data

GSM2869485_ckitplus_b1	mouse	PB	ckitp
GSM2869486_ckitplus_b2	mouse	PB	ckitp
GSM2869487_ckitplus_b3	mouse	PB	ckitp
GSM2869488_Mouse_BM_1	mouse	BM	bm-derived
GSM2869489_Mouse_BM_2	mouse	BM	bm-derived
GSM2869490_Mouse_BM_3	mouse	BM	bm-derived
GSM2869491_Mouse_BM_4	mouse	BM	bm-derived
GSM2869492_Mouse_BM_5	mouse	BM	bm-derived

S. Lai et al. "Comparative transcriptomic analysis of hematopoietic system between human and mouse by Microwell-seq". *Cell Discovery* **4**(2018):34.

Pre-Processing

1. **Genes filter:** keep genes that have expression in at least 0.1 percent of total number of cells
2. **Barcode Filters**: Based off of distribution of data (median +/- (3-5) deviations)
    - High percentage of mitochondria
    - number of genes
    - low number of cell
3. **Normalization:** Using Seurat
4. **Annotations:** Annotations at cell level done using SingleR
    - **Human:** Blueprint Epigenomics data-set annotated to 28 cell types
    - **Mouse:** Immunological Genome Project (ImmGen) annotated to 20 main cell types

Hafemeister et al. "Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression." *bioRxiv* (2019

Aran, Dvir, et al. "Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage." *Nature immunology* 20.2 (2019)

Merge Data

```
# combinedObj.list is list of seurat objects
combinedObj.merged = merge(combinedObj.list[[1]],combinedObj.list[-(1)],
                           add.cell.ids = c(names(combinedObj.list[1]),names(combinedObj.list[-(1)])))
combinedObj.merged <-FindVariableFeatures(object = combinedObj.merged,
                                          selection.method = 'vst',
                                          mean.cutoff = c(0.0125, 3),
                                          dispersion.cutoff = c(0.5, Inf))
```

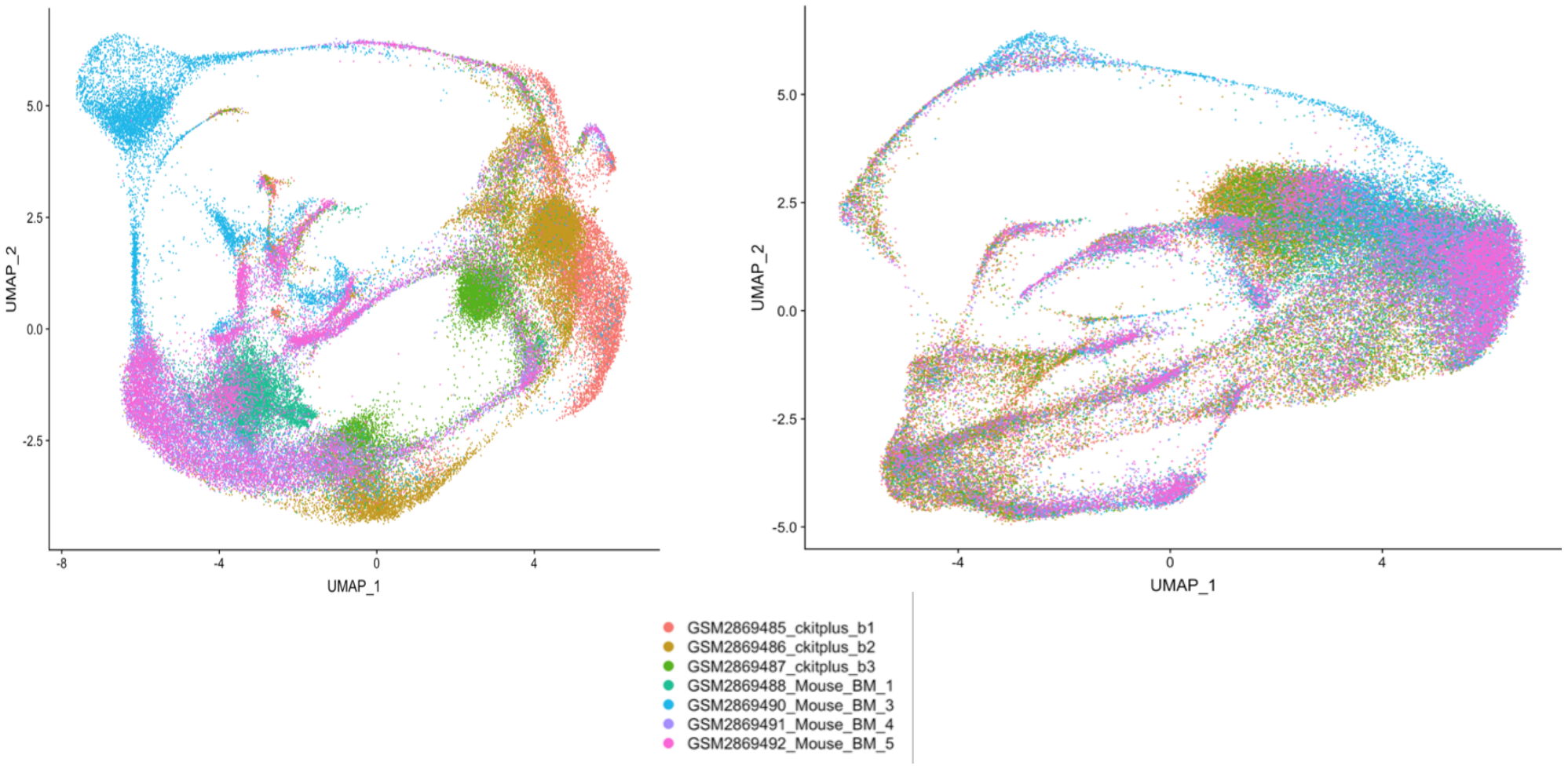
Integrate Data

```
# reference.list is list of seurat objects
for (i in 1:length(x = reference.list)) {
  reference.list[[i]] <- FindVariableFeatures(object = reference.list[[i]],
                                              selection.method = "vst", nfeatures = nAnchors, verbose = FALSE)
}

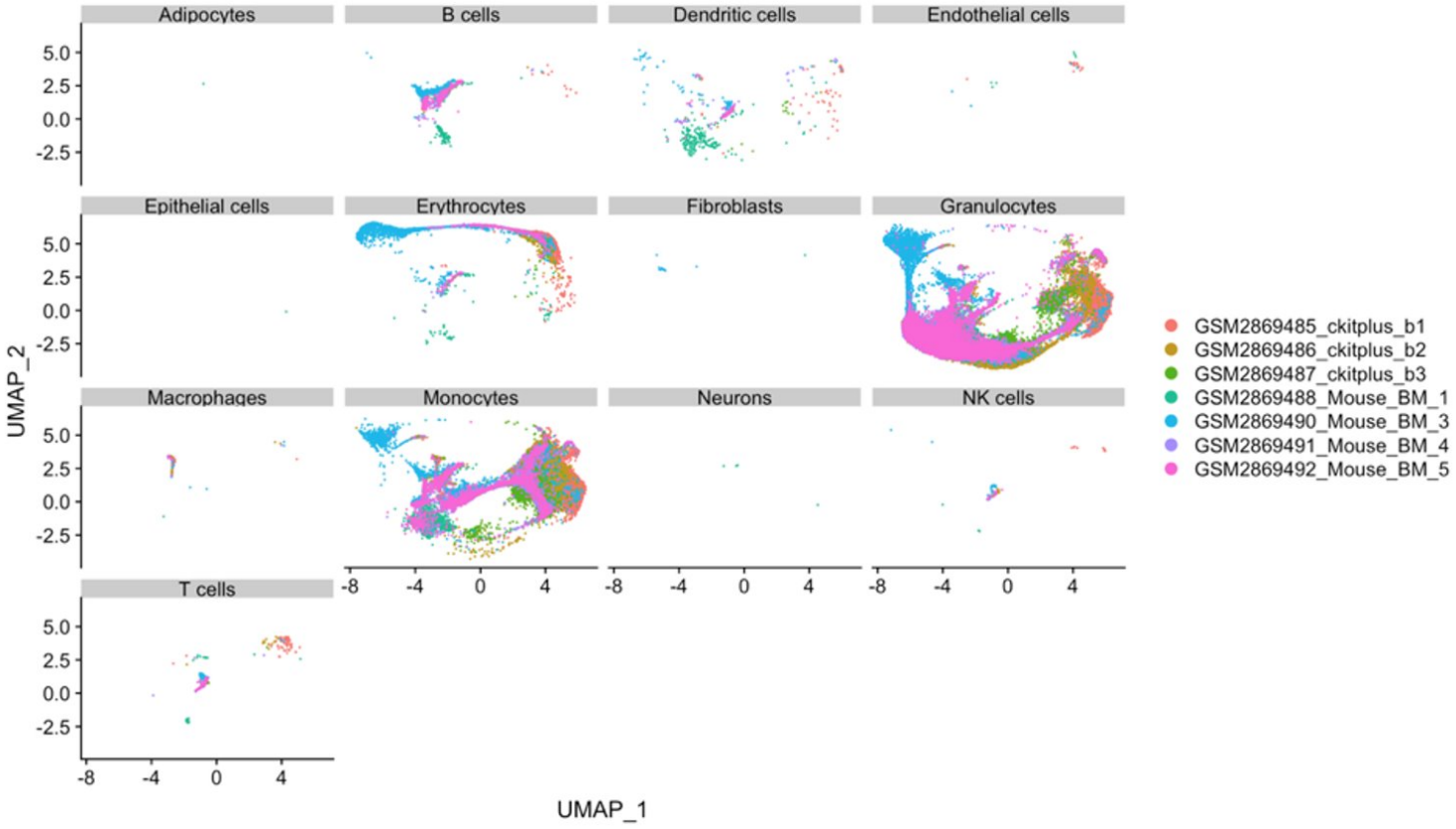

combinedObj.anchors <- FindIntegrationAnchors(object.list = reference.list,
                                              dims = 1:30,anchor.features = nAnchors)

combinedObj.integrated <- IntegrateData(anchorset = combinedObj.anchors, dims = 1:30)

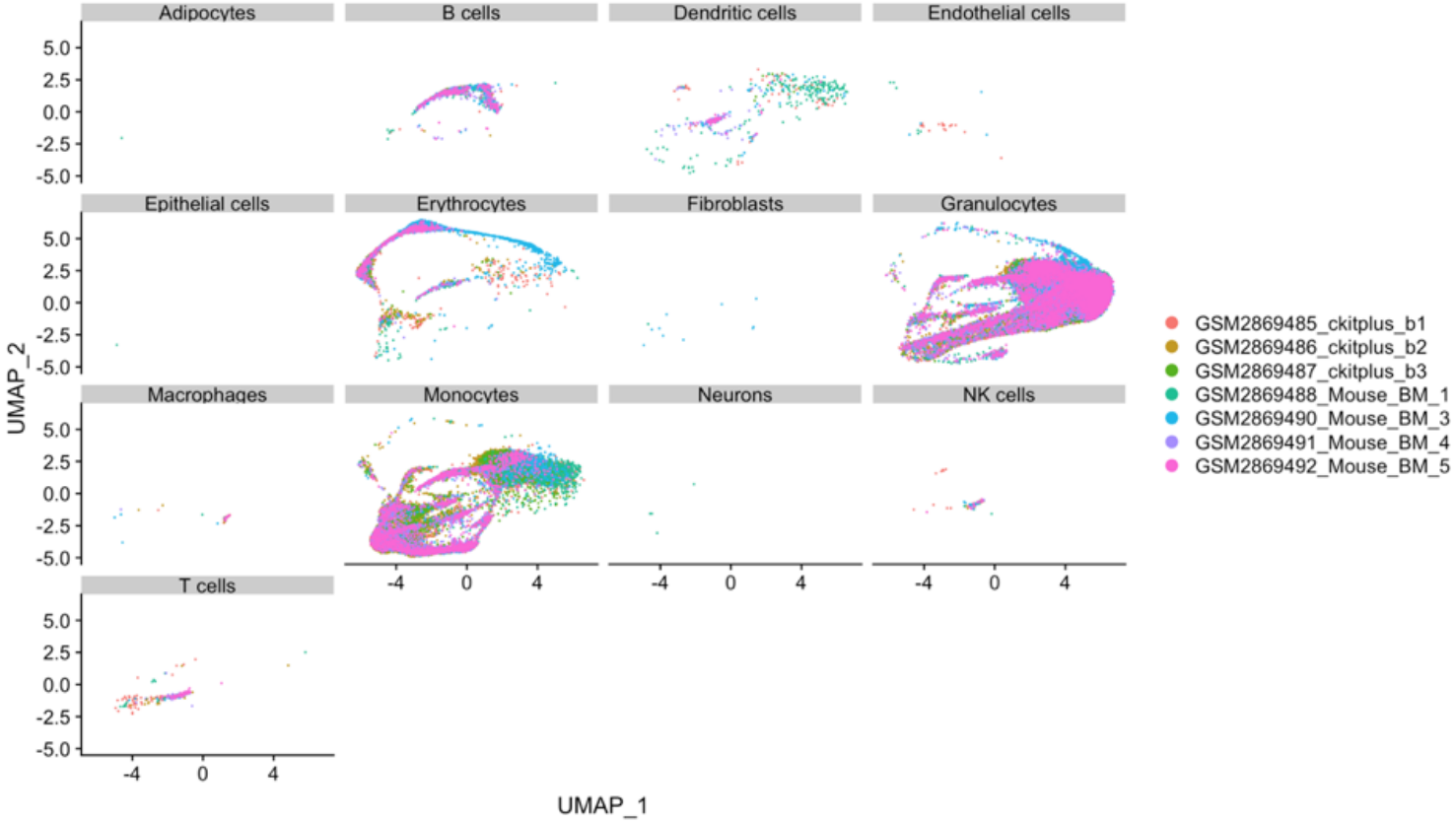
DefaultAssay(object = combinedObj.integrated) <- "integrated"
```


UMAP_2
5.0
2.5
0.0
-2.5
-8
-4
0
4
UMAP_1
UMAP_2
5.0
2.5
0.0
-2.5
-5.0
-4
0
4
UMAP_1
GSM2869485_ckitplus_b1
GSM2869486_ckitplus_b2
GSM2869487_ckitplus_b3
GSM2869488_Mouse_BM_1
GSM2869490_Mouse_BM_3
GSM2869491_Mouse_BM_4
GSM2869492_Mouse_BM_5

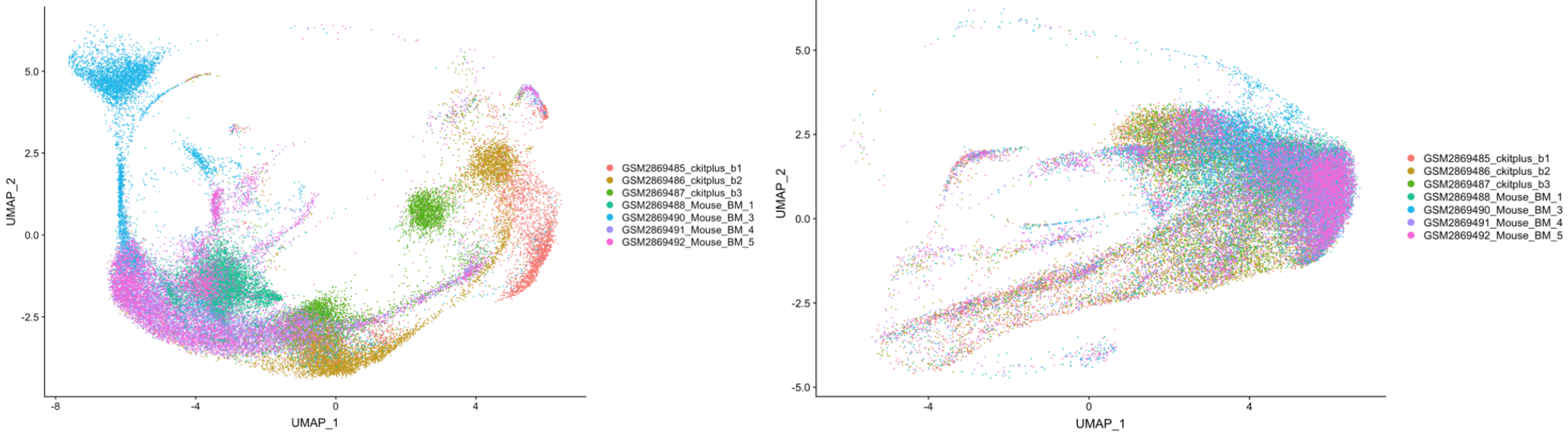
Merged


Adipocytes
B cells
Dendritic cells
Endothelial cells
Epithelial cells
Erythrocytes
Fibroblasts
Granulocytes
Macrophages
Monocytes
Neurons
NK cells
T cells
UMAP_2
UMAP_1
5.0
2.5
0.0
-2.5
-8
-4
0
4
GSM2869485_ckitplus_b1
GSM2869486_ckitplus_b2
GSM2869487_ckitplus_b3
GSM2869488_Mouse_BM_1
GSM2869490_Mouse_BM_3
GSM2869491_Mouse_BM_4
GSM2869492_Mouse_BM_5

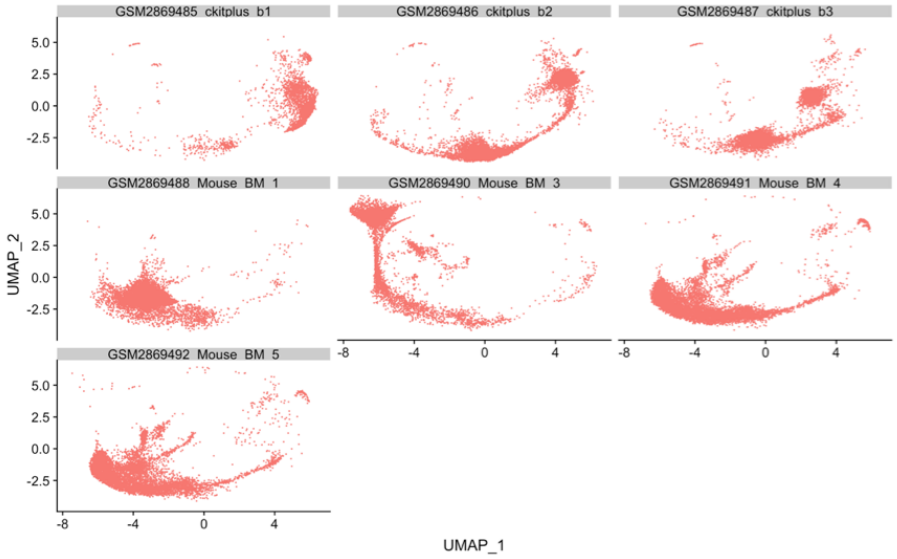
Integrated

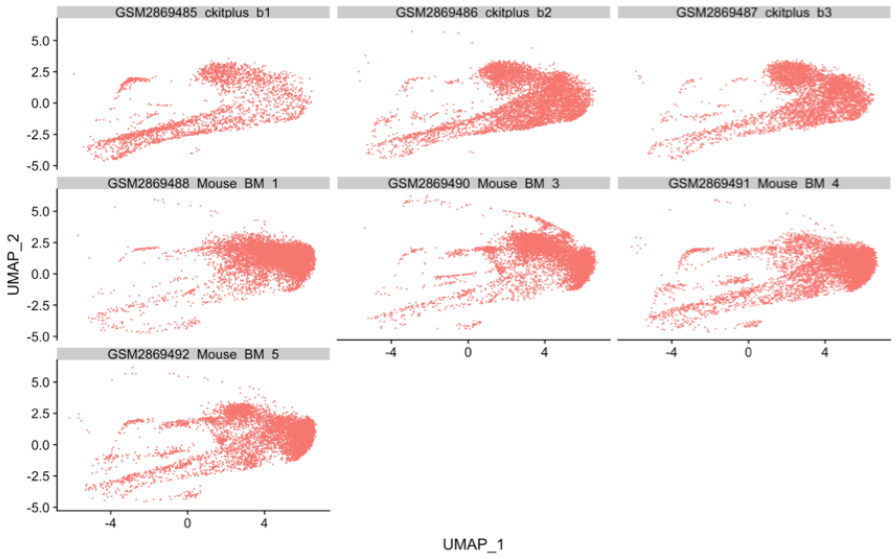

Adipocytes
B cells
Dendritic cells
Endothelial cells
Epithelial cells
Erythrocytes
Fibroblasts
Granulocytes
Macrophages
Monocytes
Neurons
NK cells
T cells
UMAP_2
UMAP_1
5.0
2.5
0.0
-2.5
-5.0
-4
0
4
GSM2869485_ckitplus_b1
GSM2869486_ckitplus_b2
GSM2869487_ckitplus_b3
GSM2869488_Mouse_BM_1
GSM2869490_Mouse_BM_3
GSM2869491_Mouse_BM_4
GSM2869492_Mouse_BM_5

Granulocytes

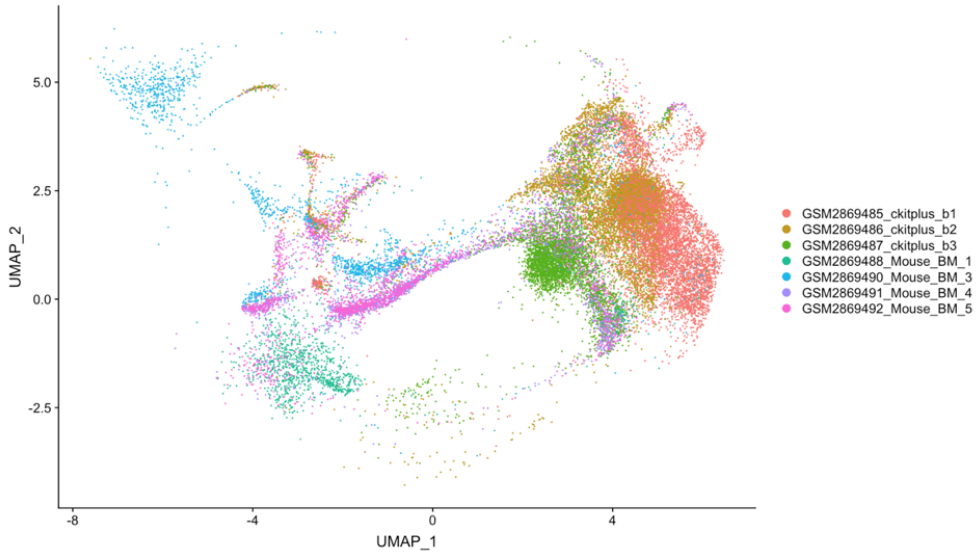


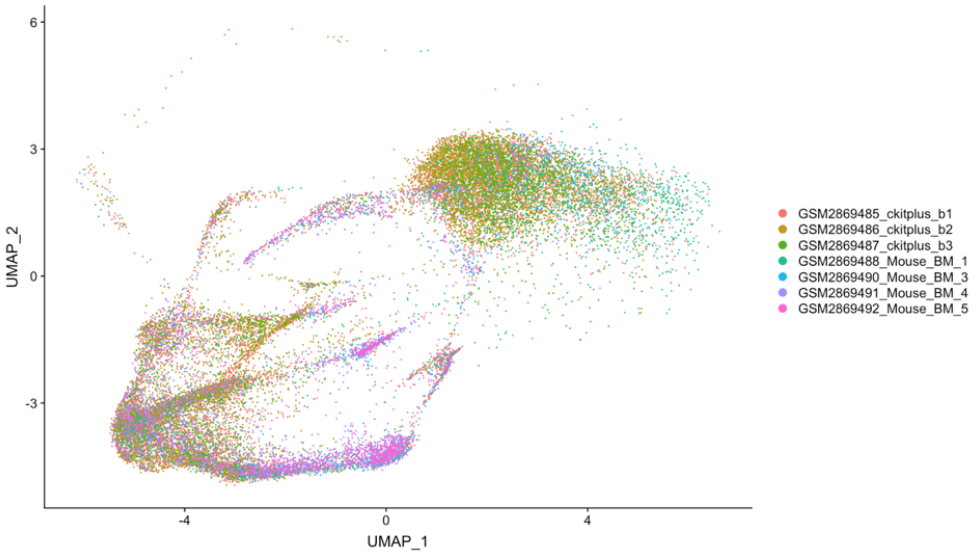
Granulocytes



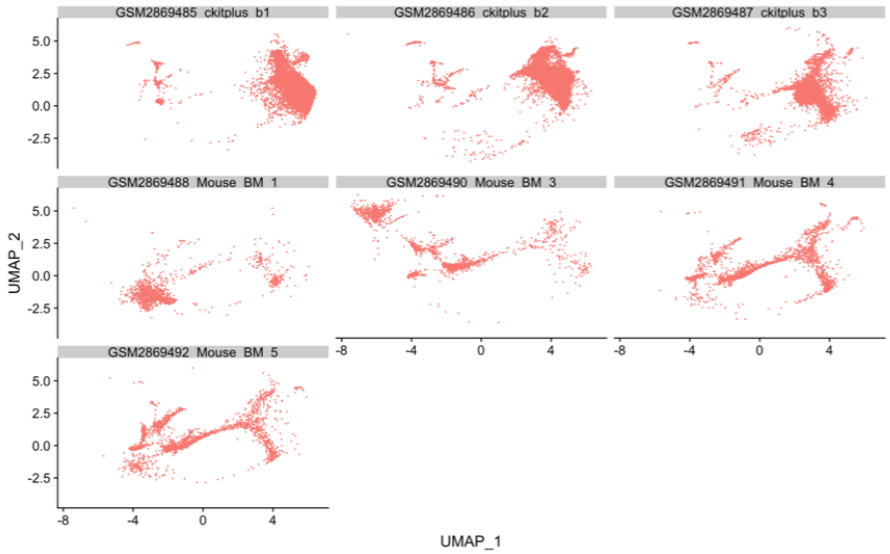


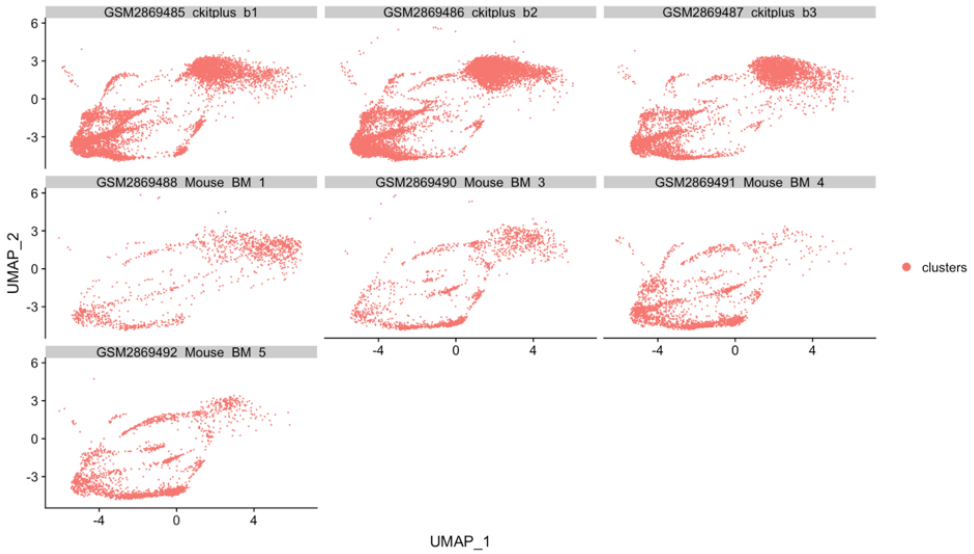
Monocytes


UMAP_2
5.0
2.5
0.0
-2.5
-8
-4
0
4
UMAP_1
GSM2869485_ckitplus_b1
GSM2869486_ckitplus_b2
GSM2869487_ckitplus_b3
GSM2869488_Mouse_BM_1
GSM2869490_Mouse_BM_3
GSM2869491_Mouse_BM_4
GSM2869492_Mouse_BM_5


UMAP_2
6
3
0
-3
-4
0
4
UMAP_1
GSM2869485_ckitplus_b1
GSM2869486_ckitplus_b2
GSM2869487_ckitplus_b3
GSM2869488_Mouse_BM_1
GSM2869490_Mouse_BM_3
GSM2869491_Mouse_BM_4
GSM2869492_Mouse_BM_5

Monocytes


GSM2869485_ckitplus_b1
GSM2869486_ckitplus_b2
GSM2869487_ckitplus_b3
GSM2869488_Mouse_BM_1
GSM2869490_Mouse_BM_3
GSM2869491_Mouse_BM_4
GSM2869492_Mouse_BM_5
UMAP_2
UMAP_1


GSM2869485_ckitplus_b1
GSM2869486_ckitplus_b2
GSM2869487_ckitplus_b3
GSM2869488_Mouse_BM_1
GSM2869490_Mouse_BM_3
GSM2869491_Mouse_BM_4
GSM2869492_Mouse_BM_5
UMAP_2
UMAP_1
clusters

Additional Dataset Integration Tools

LIGER: Welch et al. "Integrative inference of brain cell similarities and differences from single-cell genomics." *BioRxiv*(2018)

Conos: Barkas et al. "Wiring together large single-cell RNA-seq sample collections." *BioRxiv* (2018)

Harmony: Korsunsky et al. "Fast, sensitive, and flexible integration of single cell data with Harmony." *BioRxiv* (2018)

robustSingleCell: Robust clustering analysis and dataset integration via similarity analysis, including between mouse to human data. robustSingleCell @ GitHub or CRAN

Final Thoughts

- Can remove artifacts or biology
- Differential Expression
- When not to use
- Multispecies

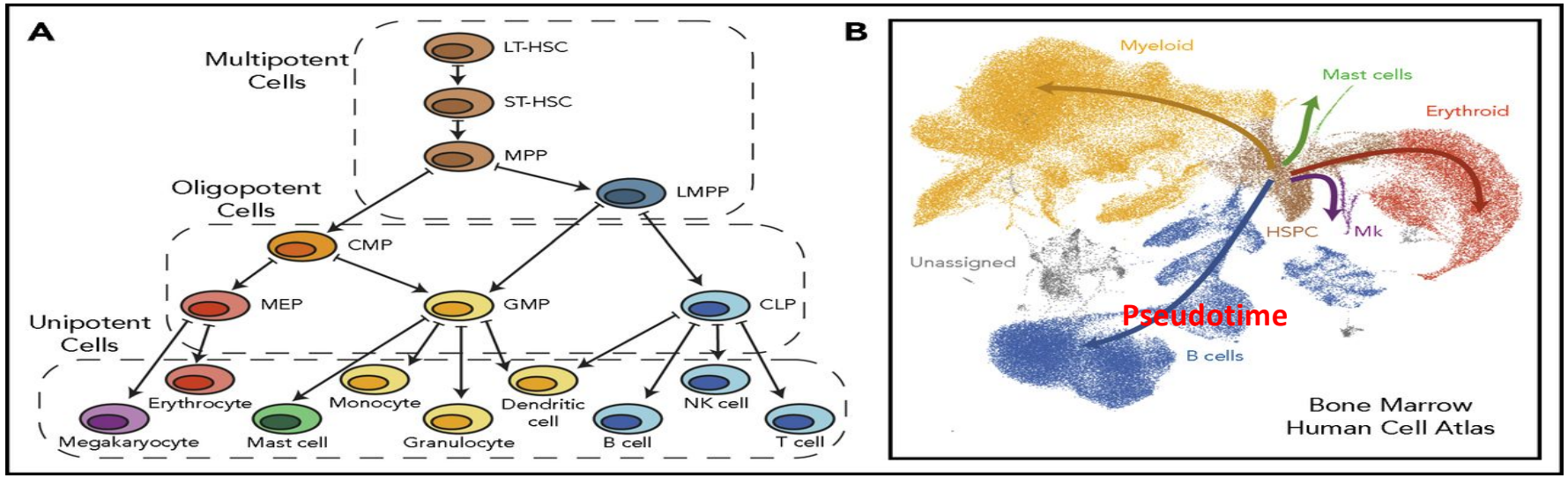
Trajectory Inference

Single Cell Users Group

Fate Mapping vs Trajectory Inference

- Cells proliferate and differentiate to generate and maintain multicellular organisms
- The gold standard for inferring relationships between progenitors and descendants is **fate mapping**
- Single cell genomics offers a ***complementary*** approach by measuring the transcriptional state of a population of cells in all states of differentiation
- Although temporal information is missing, much effort has recently been invested in inferring cellular trajectories from this translational landscape.

Single-Cell Genomics and Trajectory Inference



S. Wacham et al. “New Insights into hematopoietic differentiation landscapes from single-cell RNA sequencing” Blood 133(2019):1415-1426.

Assumptions in Trajectory Inference

- From the outset trajectory inference requires assumptions
  - Differentiation is assumed to be a continuous process
  - Cells differentiate asynchronously and are captured at multiple points along their differentiation routes
- Fundamental limitations on trajectory inference
  - Recently there have been concerns that these assumptions alone are insufficient to infer *unique* trajectories
  - Argue necessary to add additional assumptions (*implicitly made in current models*)
    - Trajectories are ***Markovian*** (determined by current cell-state, not history)
    - ***No oscillatory*** trajectories (can't discern cyclic behavior in a snapshot)

C.Weinreb et al. "Fundamental limits on dynamic inference from single-cell snapshots" PNAS 115(2018):E2467-E2476.

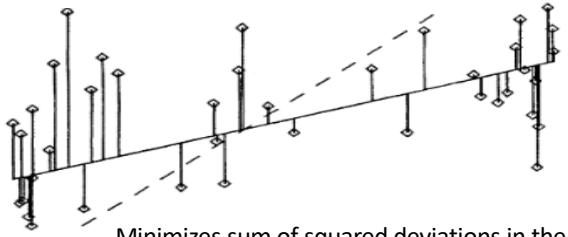
Implementations of TI

- More than 70 different implementations
  - It is ***one of the largest*** categories of single cell analysis tools
  - Recently 45 bench-marked on 110 real and 229 synthetic datasets
- Typical features
  - Start with mapping to simpler representation
    - Dimensional reduction
    - Clustering
    - Graph representation
  - Often construct ***principal curve/graph*** within the simpler representation
  - Order cells along the principal curve

W. Saelens et al. “A comparison of single-cell trajectory inference methods” Nature Biotechnology. April 2019

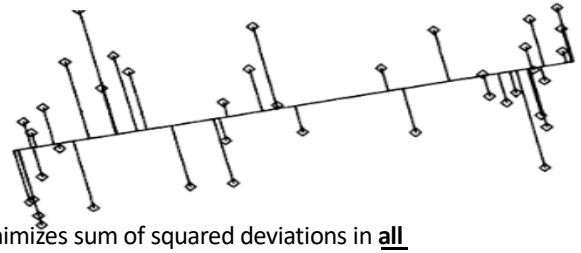
Principal Curves

Linear Regression



Minimizes sum of squared deviations in the response variable

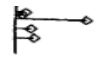
PCA line



Minimizes sum of squared deviations in **all** variables

All these are methods of characterizing distributions via a line that passes "midway" through the distribution

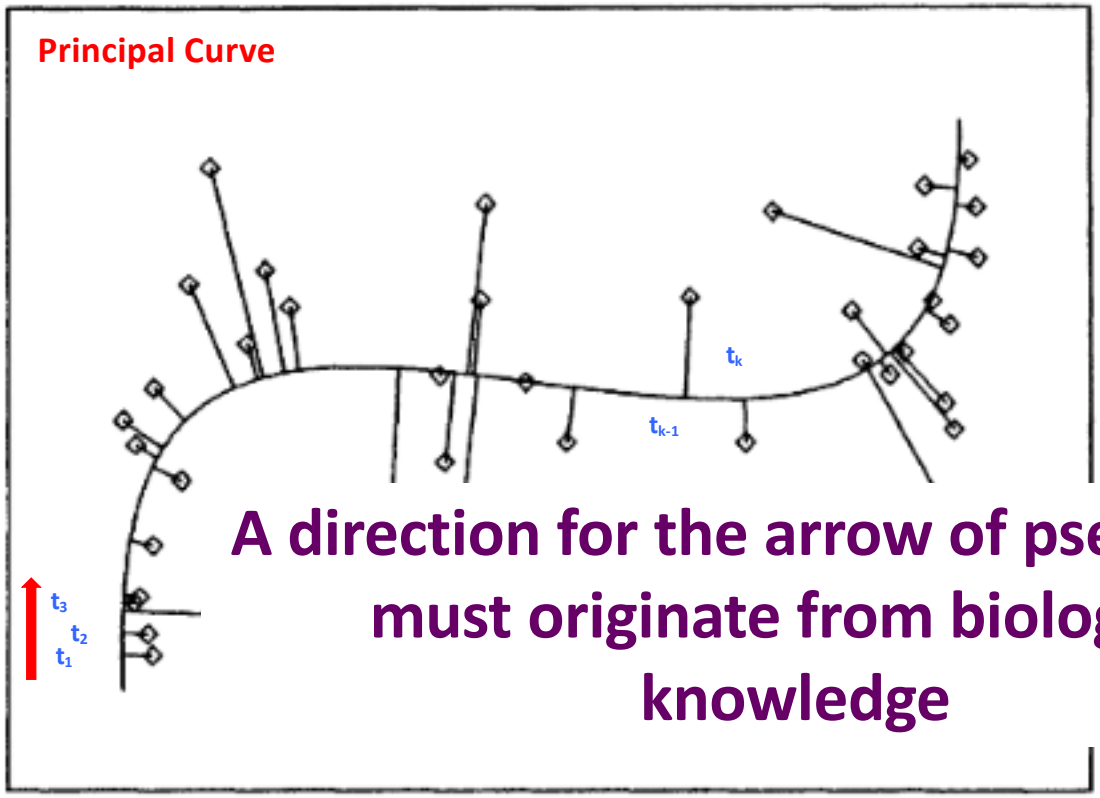
Minimizes sum of squared deviations in the response variable, s.t. smoothness constraints



Minimizes sum of squared deviations in **all** variables, s.t. smoothness constraints

T. Hastie and W. Stuetzle. "Principal Curves". J. Am. Stat. Assoc. 84(1989):502-516.

Principal Curves impose ordering



A direction for the arrow of pseudotime must originate from biological knowledge

T. Hastie and W. Stuetzle. “Principal Curves”. J. Am. Stat. Assoc. 84(1989):502-516.

Differences between Methods

- How the data is mapped to a simpler representation
- Whether/How the Principal Curve is computed
- Whether trajectories have fixed or inferred topology
- Whether the trajectories can cope with only linear, tree or general graph trajectories (discontinuous graphs, cycles, etc.)
- What kind of initial information needs to be supplied:
  - Earliest cell
  - Important genes
  - Cell type classification
- Performance and Robustness

Monocle

- The prototypical Trajectory Inference Tool
- Computes principal graph via **reverse graph embedding**
  - Finds mapping between the high dimensional gene expression space and much lower dimensional space
  - Simultaneously learns the principal graph structure of the lower dimensional space

Monocle – Usage

- Implemented in R
- Data encapsulated in CellDataSet objects
- Version 3 under active development
- During upgrades of Seurat and monocle, methods to convert Seurat objects into CellDataSets have become out of date
- We supply very a basic conversion method amongst our convenience functions – Use with caution!

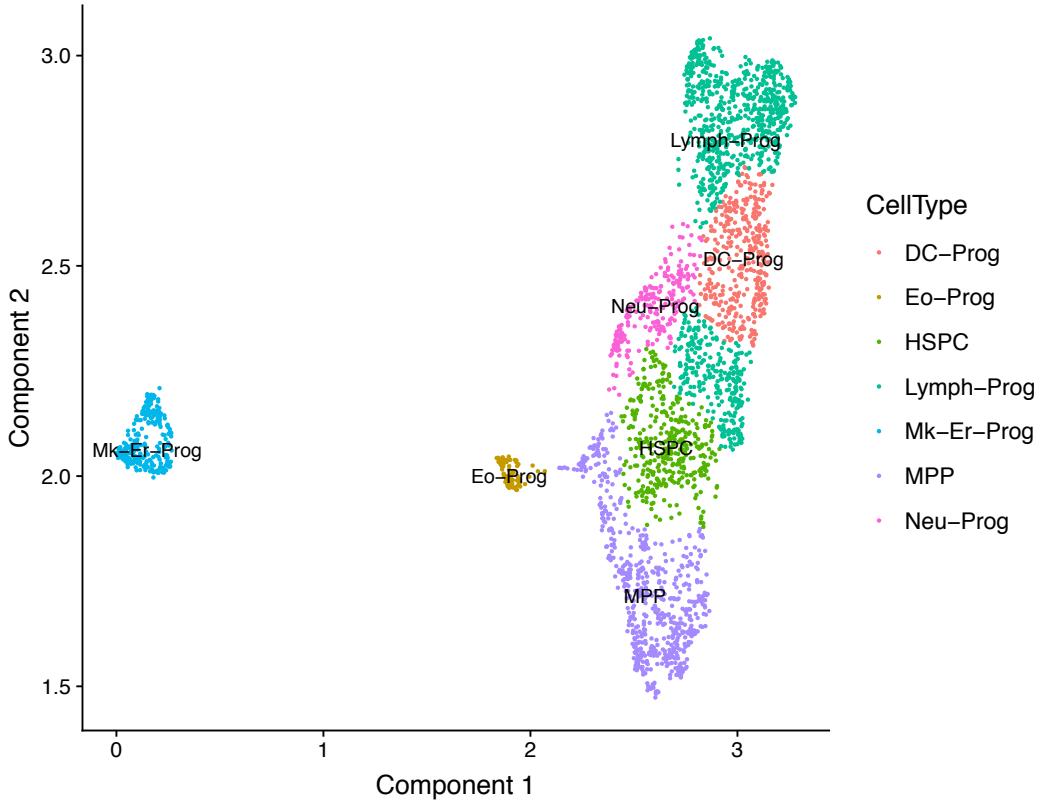
Monocle – Dimensional Reduction

```
# Read in data and preprocessing
...

# Clustering and annotation via Garnett
...

# (Non-linearly) Reduce dimensionality
cds <- reduceDimension(cds,
                       reduction_method = 'UMAP’)
```

Clustering and Annotation



Monocle – Graph Learning

```
# Partition cells into supergroups (distinct
# trajectories)
cds <- partitionCells(cds)

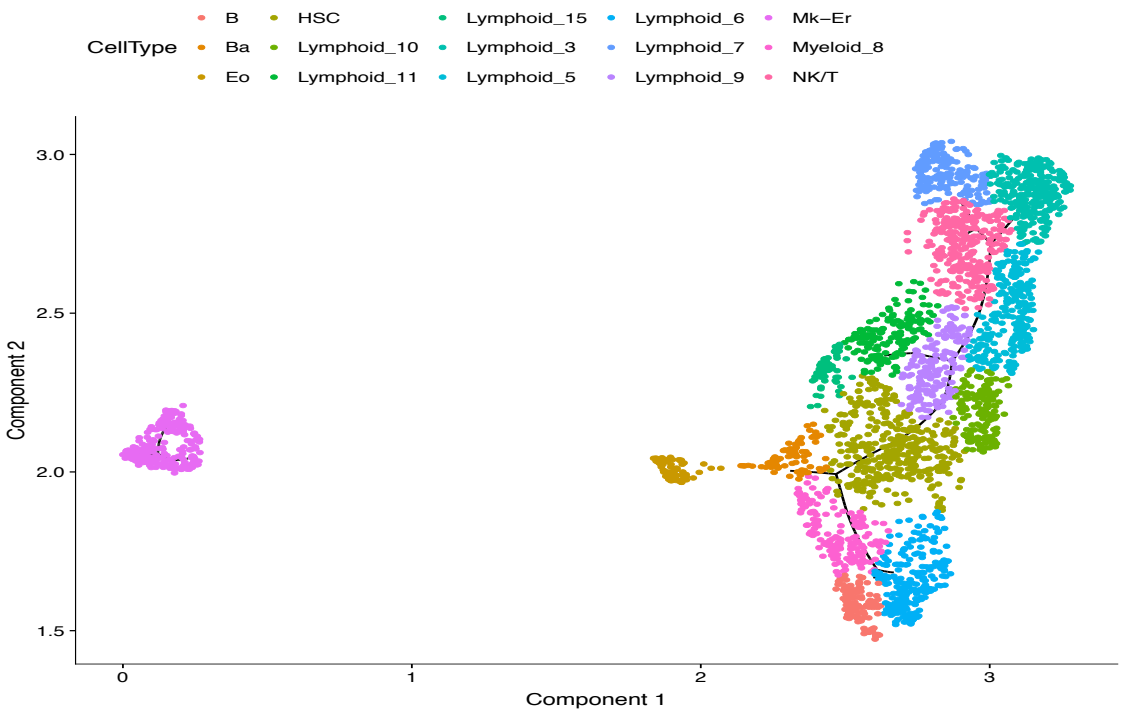
# Learn Principal Graph (there are two
# methods for RGE – 'SimplePPT' and
# 'DDRTree')
cds <- learnGraph(    cds,
                      RGE_method = 'SimplePPT')
```

Plot Cell Trajectories

```
# Learn Principal Graph
...

# Plot cell trajectory
Plot_cell_trajectory( cds,
                      color_by = “cell_type”)
```

Cell Trajectories



Monocle – Setting root state

```
# Use convenience function to find roots
hspcs_node <- estimate_root_node(
                    cds,
                    cell_phenotype = "cell_type",
                    root_type = "HSPC”)

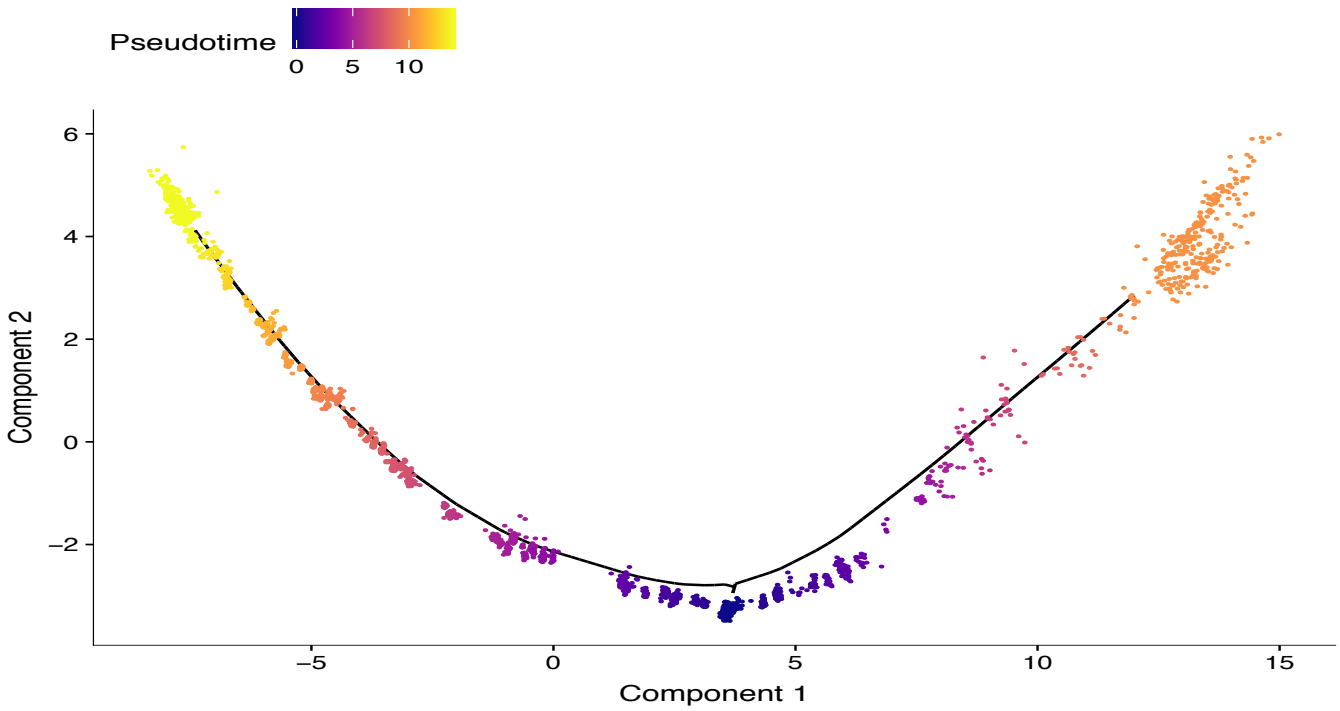
# Caution: Adjust root_type to most primitive
# cell type for your data
```

Monocle – Order Pseudotemporally

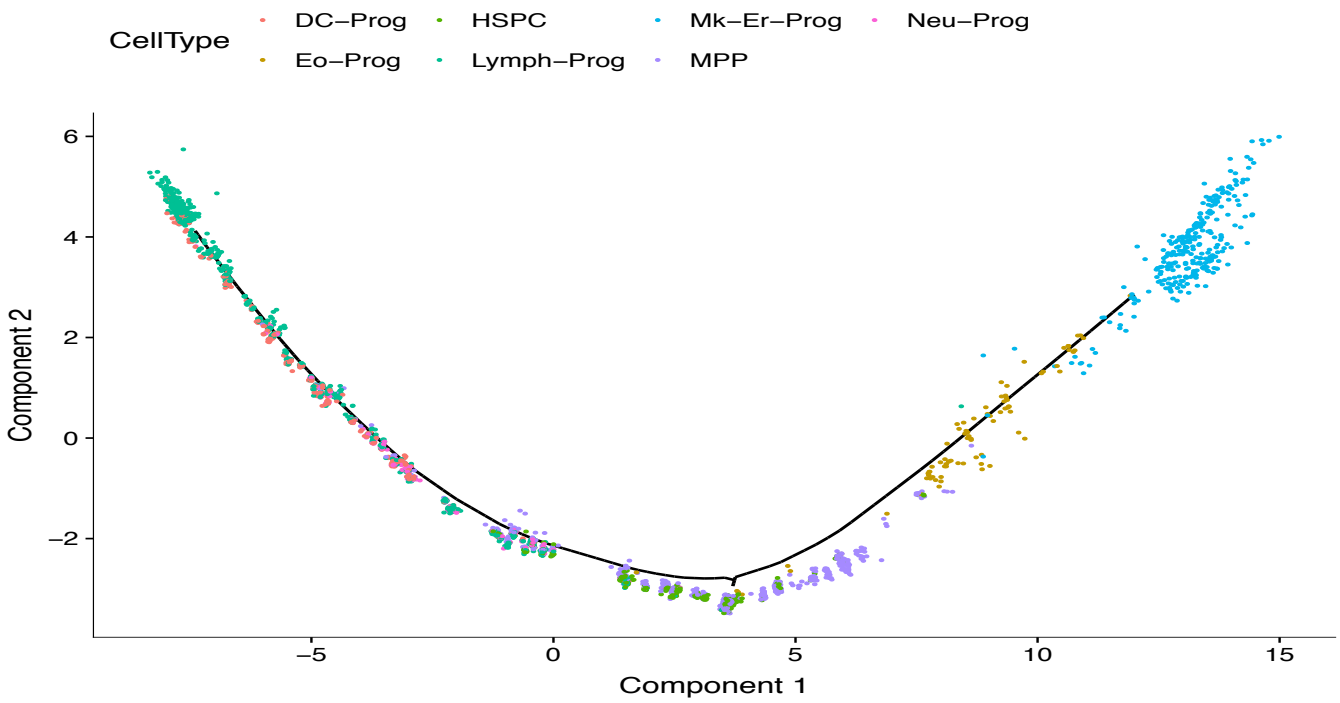
```
# Use principal grap to pseudotemporally
# order cells
cds <- orderCells(cds)

# Plot the trajectory
plot_cell_trajectory( cds,
                      color_by = "Pseudotime")
```

Pseudotemporal Ordering - PT



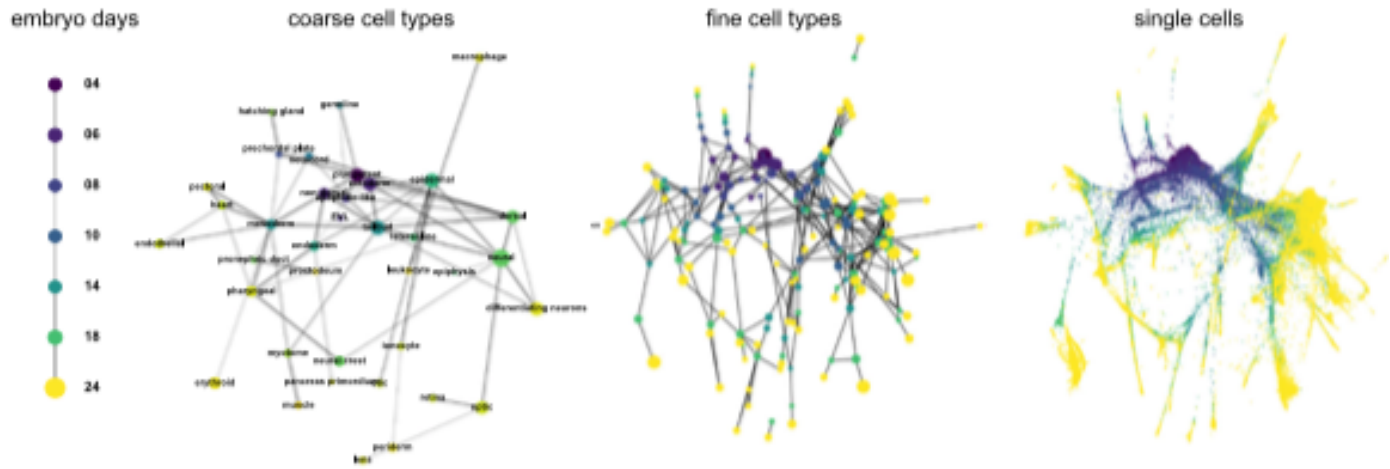
Pseudotemporal Ordering – Cell Type



Partition-based Graph Abstraction (PAGA)

- **Motivation**: There is tension between clustering and trajectory inference
  - Clustering confers discrete index to cells
  - Pseudotemporal ordering assumes the data lie on connected manifolds and labels cells with continuous labelsPartition-based graph abstraction
- Gives graph-like map of data manifold, based on estimating connectivity of manifold ***partitions*** (e.g. cell clusters)
- Preserves global topology of data, allowing analysis at different resolutions
- Unifies both the clustering and continuous change approaches

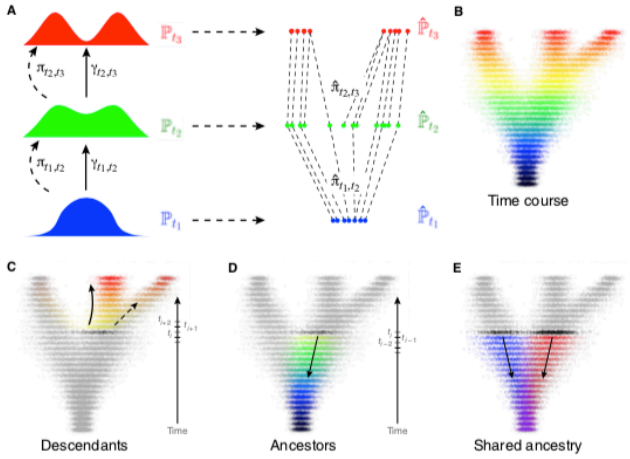
Partition-based graph abstraction (PAGA)



Implemented within scanpy in python

Waddington-OT

- **Approach**:
  - Inputs are dataset from synchronous developmental process at known time points
  - Rather than computing a Principle Graph, it tracks the locations of cells from one time slice to the next by a least action principle (Optimal Transport)

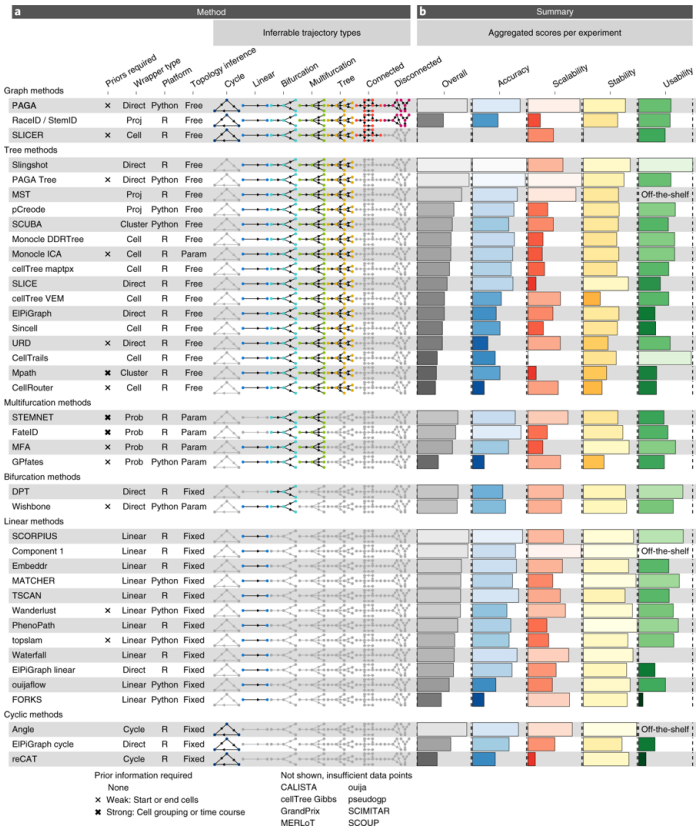


Comparisons

- 45 implementations were bench-marked on 110 real and 229 synthetic datasets
- Assessed for
  - Accuracy
    - Overall topology
    - Quality of assignment to branches
    - Cell positions
  - Scalability
  - Stability
  - Usability

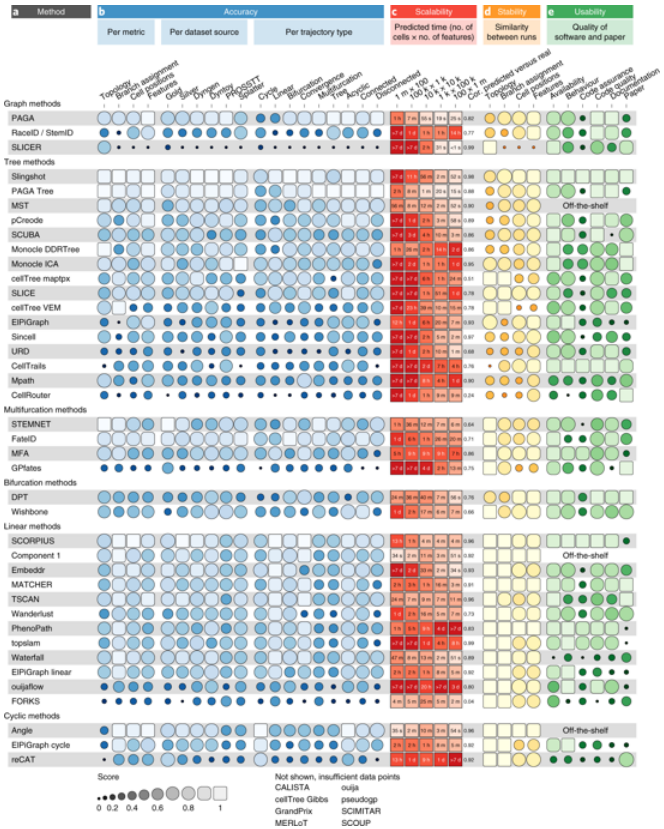
W. Saelens et al. “A comparison of single-cell trajectory inference methods” Nature Biotechnology. April 2019

Comparisons



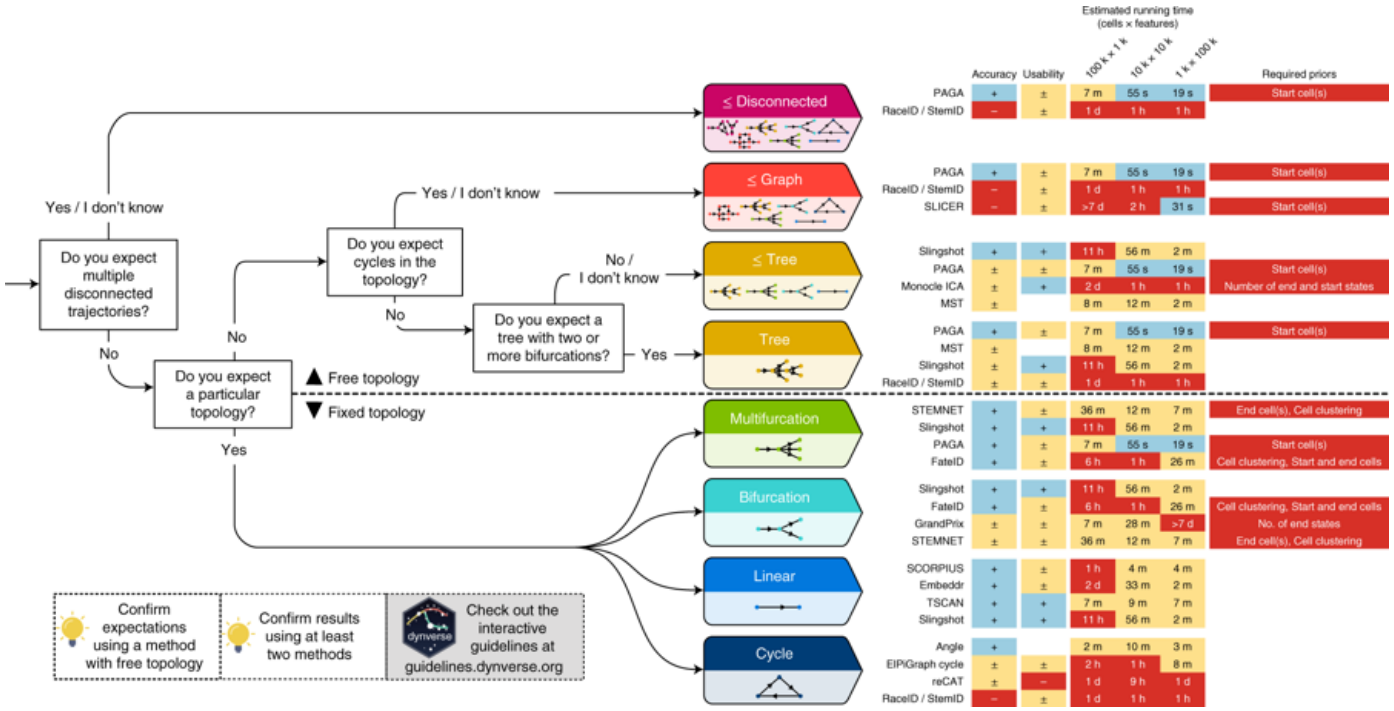
W. Saelens et al. “A comparison of single-cell trajectory inference methods” Nature Biotechnology. April 2019

Comparisons



W. Saelens et al. “A comparison of single-cell trajectory inference methods” Nature Biotechnology. April 2019

Comparison



W. Saelens et al. "A comparison of single-cell trajectory inference methods" Nature Biotechnology. April 2019

Conclusion

- Implementations of Trajectory Inference are abundant and their number continues to grow
- There are fundamental limitations on trajectory inference methods
- Definitive lineage tracing in conjunction with single cell [multi-]omics may help to improve trajectory inference

References

- Methods
  - C. Trapnell et al. “The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells”. *Nat Biotechnology* **32**(2014):381-386
  - X. Qiu et al. “Reversed graph embedding resolves complex single-cell trajectories”. *Nat Methods* **14**(2017):979.
  - F.A. Wolf et al. “PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells”. *Genome Biol* **20** (2019):59.
  - H. Chen et al. “STREAM: Single-cell Trajectories Reconstruction, Exploration And Mapping of omics data” bioRxiv. April 2018.
  - G. Schiebinger et al. “Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming”. *Cell* **176**(2019):928-943.
- Fundamental Limitations
  - C.Weinreb et al. “Fundamental limits on dynamic inference from single-cell snapshots”. *PNAS* **115**(2018):E2467-E2476.
- Comparisons
  - W. Saelens et al. “A comparison of single-cell trajectory inference methods”. *Nature Biotechnology*. April 2019