Single-Cell RNA-seq

March 1, 2023

Outline

- **Morning**
  - Quantification of single cell transcriptomes
  - Basic filtering and quality control in Seurat
  - Clustering and basic visualizations in Seurat
- **Afternoon**
  - Compositional analysis of clusters
  - Other statistical analysis
  - Pseudotime

Confirm you have these packages installed in your R studio:

Seurat
Matrix
dplyr
Fossil
ComplexHeatmap
Monocle

To check:

1) Open R studio
2) Run the commands:

```
library(Seurat)
library(Matrix)
library(dplyr)
library(fossil)
library(reshape2)
library(ComplexHeatmap)
library(monocle)
```

3) If no errors, then you're OK

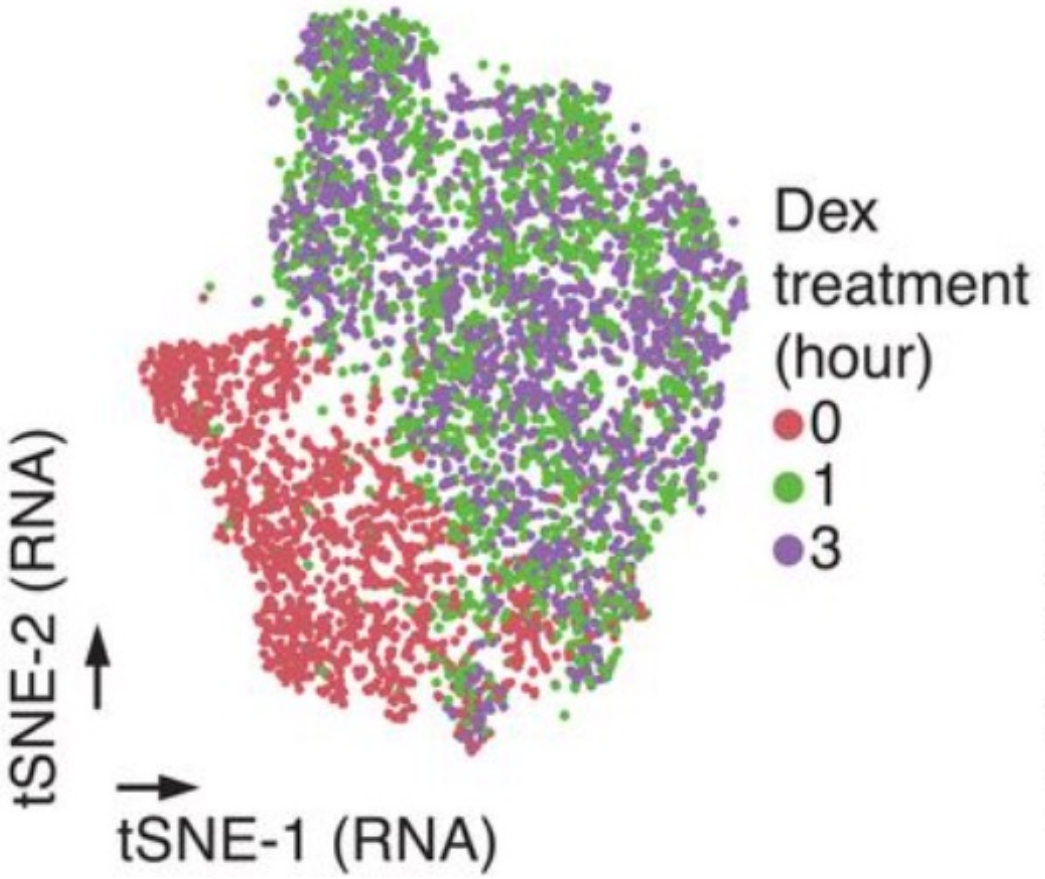
Purpose of scRNA-seq

- Gain deeper insights into the diversity of cell types within a sample
  - Identify new cell sub-populations
  - Trace out developmental trajectories
- Comparative analysis
  - Track changes in sub-populations across sample types
  - Determine marker genes for different sub-populations, or points along a trajectory
- Today: we will review typical data processing steps and analysis options at a high level. We will mostly use Seurat for our analysis for convenience, but there are many other options.

Conceptual difference from bulk RNA-seq

scRNA-seq data is *compositional*

- Bulk RNA-seq:
  - Expression is an average of all existing sub-populations
  - Differential analysis is effectively tracking changes in sub-population abundance
- scRNA-seq:
  - Allows us to directly track sub-populations across samples



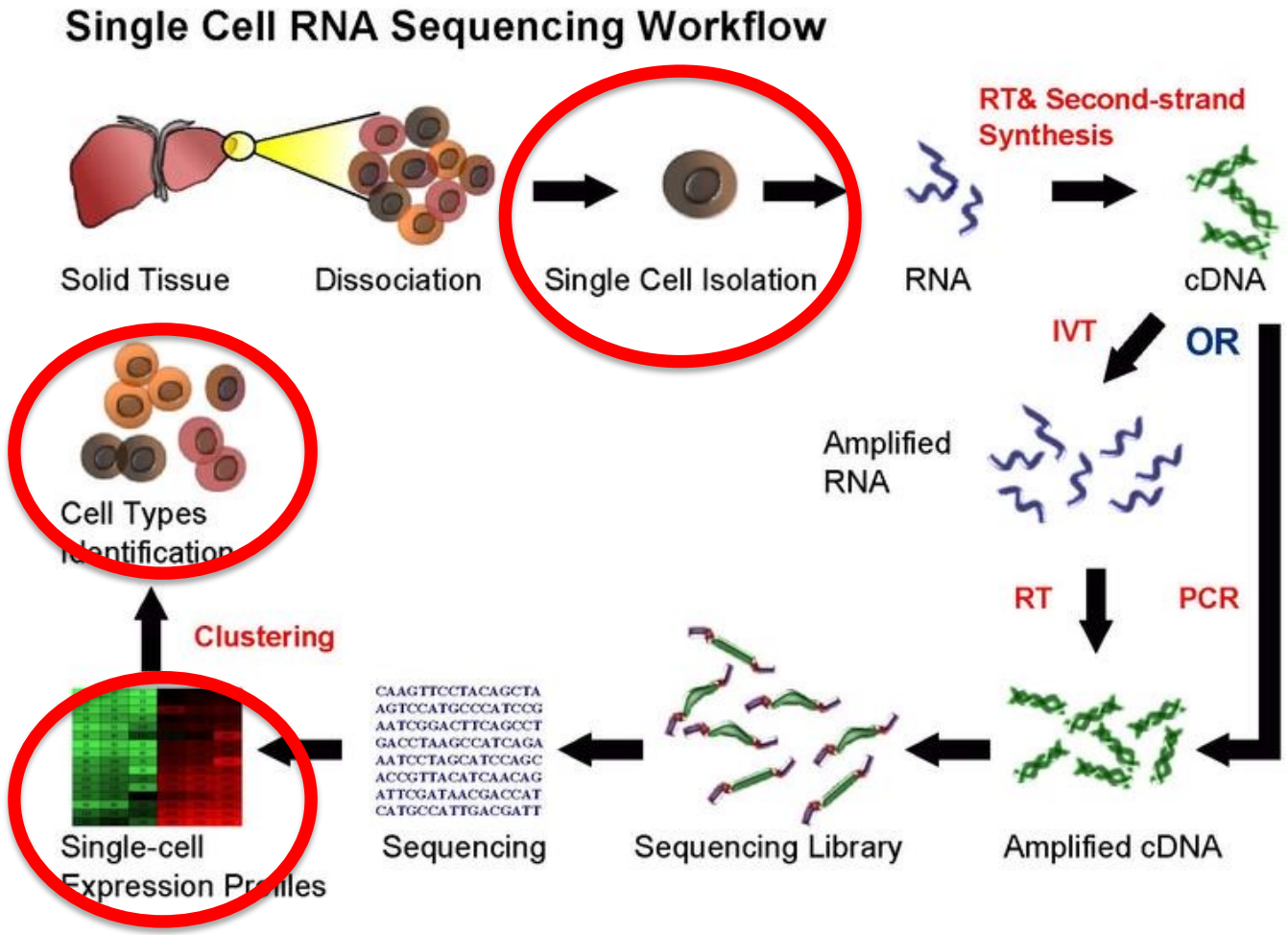
cribioinfo@uic.edu

Limitations

- Single-cell isolation is difficult
  - Tissue dissociation may be challenging or impossible
  - Cells change and die when not part of the tissue
  - Cell sorting (FACS) is rough on cells
  - High-throughput isolation strategies give some doublets, and the number of cells captured must be estimated
- Transcriptome profiling is shallow
  - We only recover a small fraction of the transcriptome
  - ~500-20,000 counts per cell, versus ~10M-40M counts in bulk RNA-seq
  - ~500-5,000 genes expressed, versus ~15,000-20,000 in bulk RNA-seq
- Correspondence to cell phenotype: mRNA ≠ Protein
  - Surface markers may not be expressed at the time of capture
  - Transcribed genes may not be translated
  - Annotation of cell types is a manual process

scRNA-seq workflow

- Many platforms are available for **isolation**
- **Quantification** steps are heavily platform-dependent
- Downstream analysis steps (like **clustering**) are more generalizable

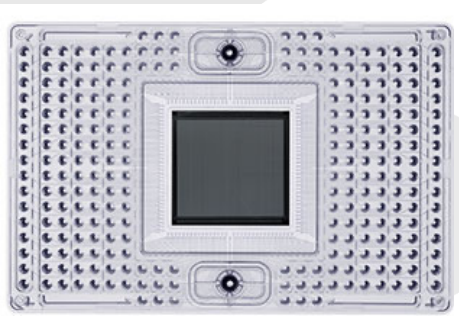


cribioinfo@uic.edu

Platforms: deposition-based (e.g., Fluidigm)

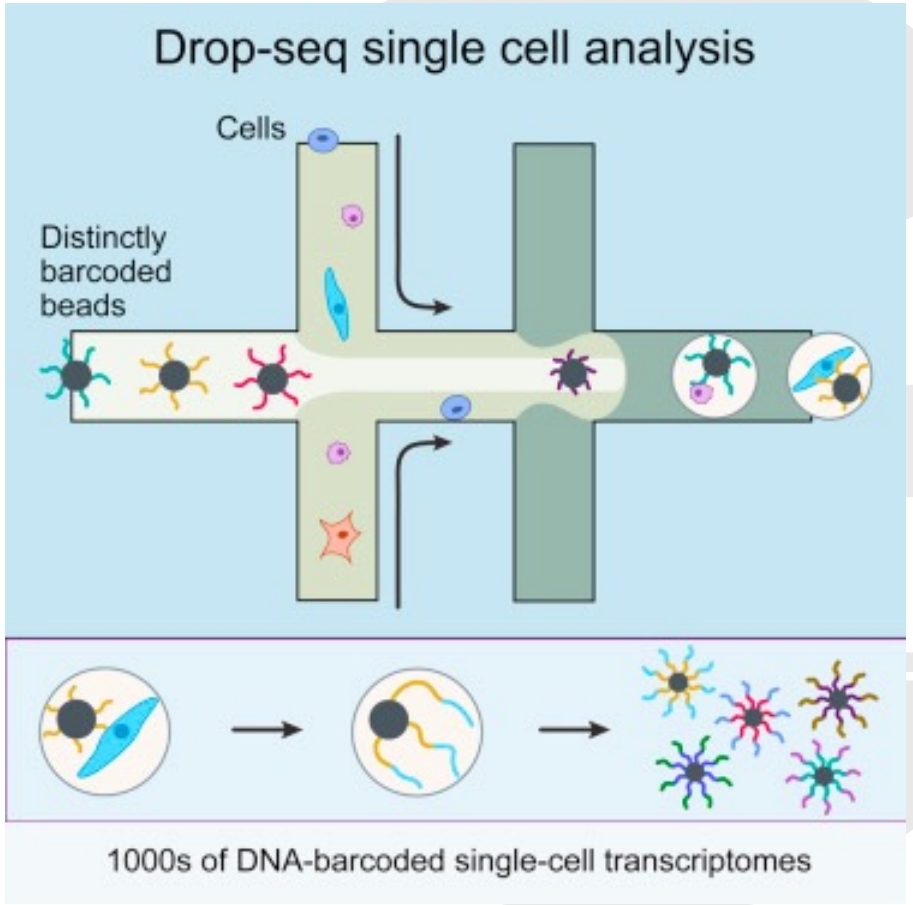
- Microfluidic or gravity-based deposition of single cells into micro/nanowells
  - Fluidigm – 96 or 800-well plates
  - Takara/Clontech uses 5184 nanowell chips
  - Others: BD Biosciences, Celsee
- Allows phenotyping of cells by microscopy, validation of true single cells
- Capture performance varies will cell morphology and size
- Generally lower throughput, higher cost per cell
- Can get higher sequencing depth per cell (~1-2M)
  - Fluidigm can also do qPCR assays instead of whole transcriptome

C1 chip



Platforms: droplet-based

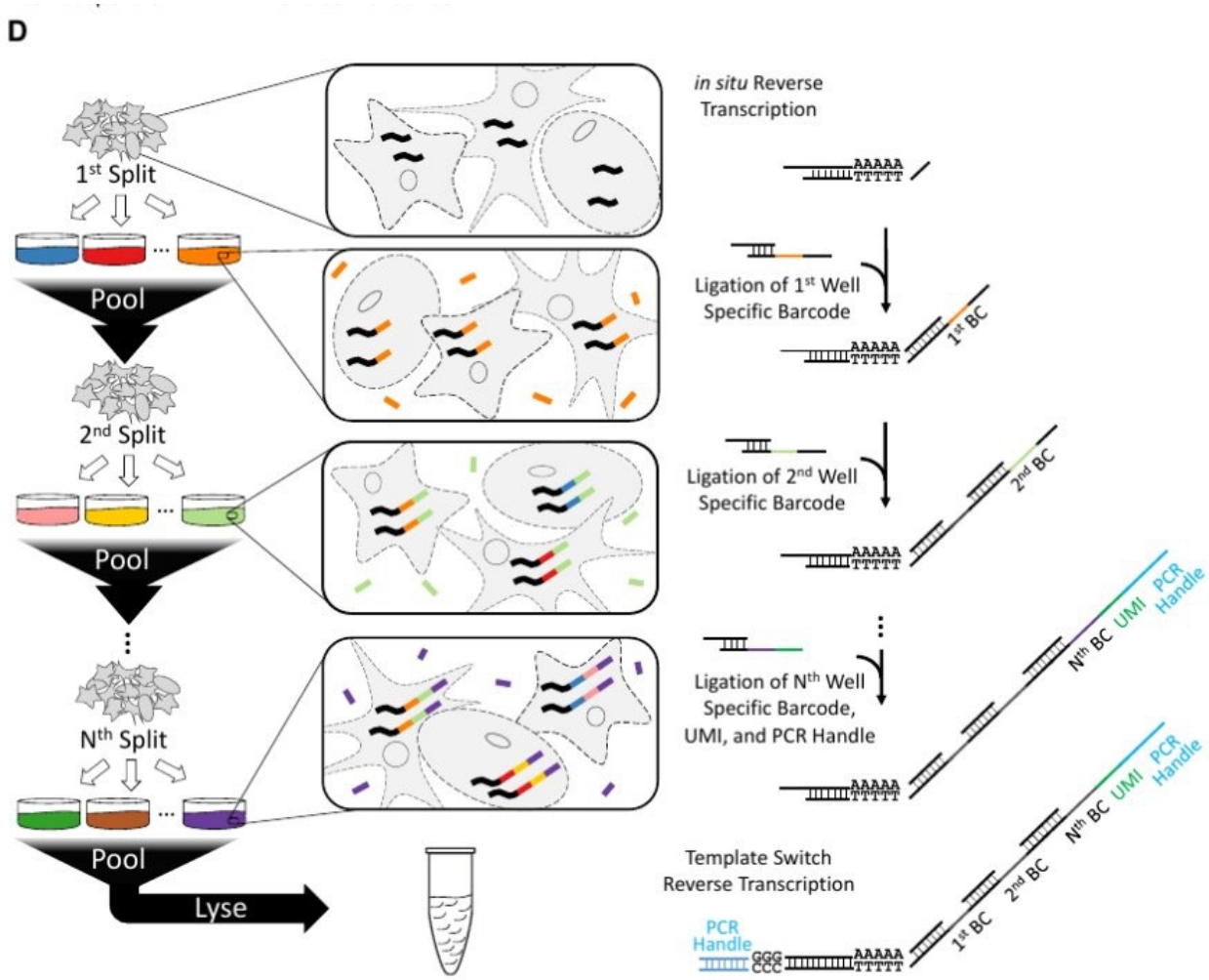
- Platforms:
  - Drop-seq (DYI), 10X Chromium, OneCellBio InDrop, BioRad
- Microfluidic capture of single cells in aqueous droplets with barcoded beads suspended in oil
- Capture 1000s-10,000s of cells
- No phenotyping, number of cells is an estimate
- Capture includes some doublet cells (estimates are ~1-5% of population). No explicit ability to detect doublets.
- Flexible to a wider range of morphologies and sizes
- Low cost per cell
- Lower counts per cell (~2-20k)



cribioinfo@uic.edu

SPLiT-seq

- Deposit cells into a set of pools, e.g., 96 well plate
  - Many cells per pool
- *In situ* reverse transcription, ligation of a pool-specific barcode
- Repeat 2-3 times with different random depositions
  - Combination of barcodes gives you probabilistic single-cell resolution
- Can yield upwards of millions of cells without special equipment, but requires cells to be fixed first



cribioinfo@uic.edu

Other platforms

- For genomics:
  - Menarini Silicon Biosystems
    - Imaging and precise isolation of single cells using non-uniform electric field, esp. for DNA-seq (e.g., CNVs)
  - Mission Bio (Tapestri)
    - Surface protein characterization and SNV/indel
  - Also Fluidigm

Add-ons to gene expression

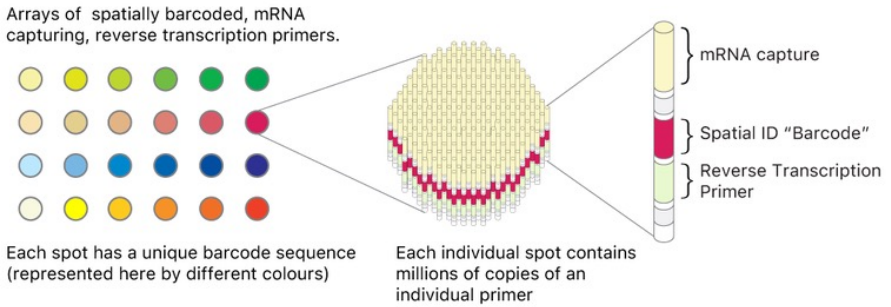
- VDJ profiling
  - E.g.: CDR sequences for T cells
  - Profiling immune repertoire on a single-cell level
- Phenotyping by antibody tags (“Cell Hashing”)
  - Use oligo-tagged antibodies specific to proteins of interest
  - Allows direct measurement of protein abundance per cell
    - Orthogonal phenotyping to transcriptome
  - Must validate antibodies first
  - Can also be used to multiplex cells from individual samples in a single capture
- Both involve a second library for sequencing, beyond standard 3’ RNA-seq
- Both available on 10X, possibly other systems
- Discuss options with Genomics Core and RIC during study planning

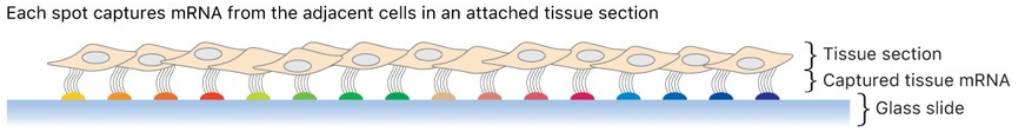
Single cell/nuclei ATAC-seq

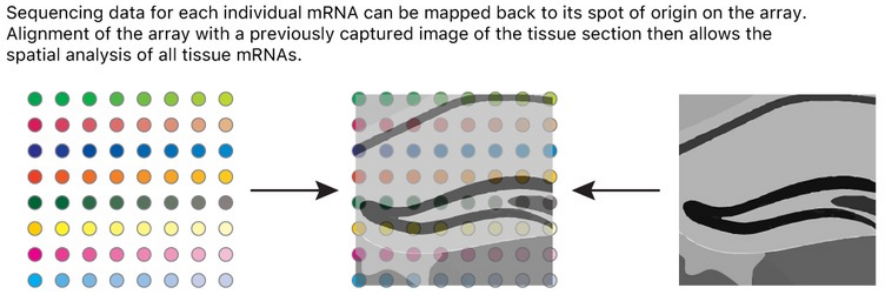
- Measure open chromatin in cells
  - Newer option from 10X allow simultaneous scATAC-seq and scRNA-seq from *same* cell
  - Can also run separately from scRNA-seq
- Quantification pipeline different from scRNA-seq
  - Need to find open chromatin peaks first
  - Quantification of reads against identified peaks
- Variability usually lower than scRNA-seq
  - Peak-to-peak variation: Lower dynamic range across peaks
  - Cell-to-cell variation: Fewer clusters for same experiment
- Broadly similar analysis steps
  - Some differences in specific methods (e.g., use LSI on binary present/absent matrix instead of PCA on scaled gene expression levels)
  - Will need to associate peaks to genes based on genomic coordinates

Alternative

- Spatial transcriptomics
  - Resolve gene expression in two dimensions against a tissue slide
  - May or may not be single-cell level
  - Compare to histology
  - Various platforms available



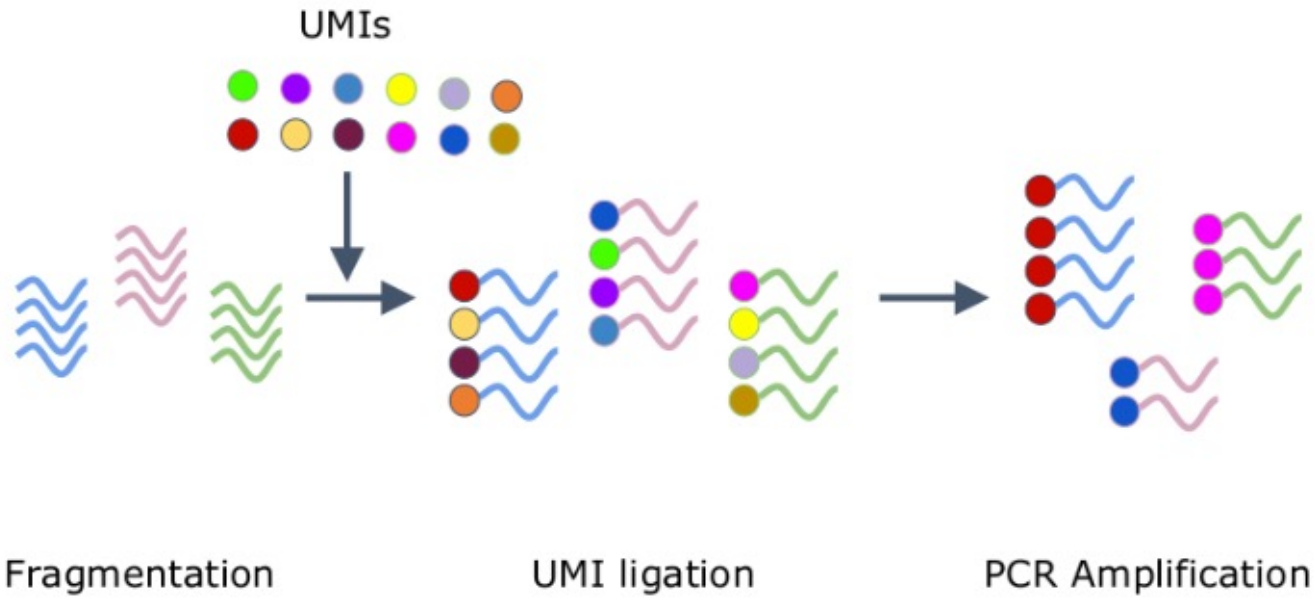




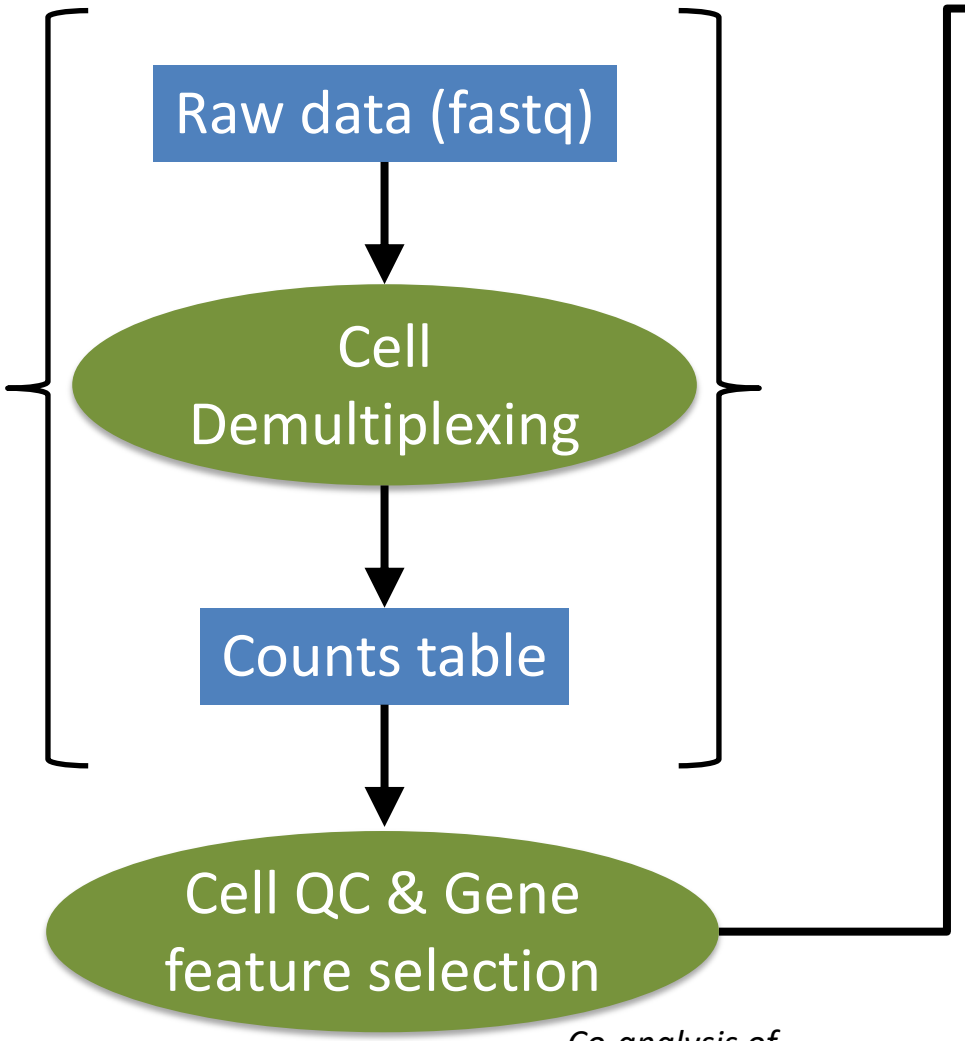
cribioinfo@uic.edu

Aside: Unique Molecular Identifiers (UMIs)

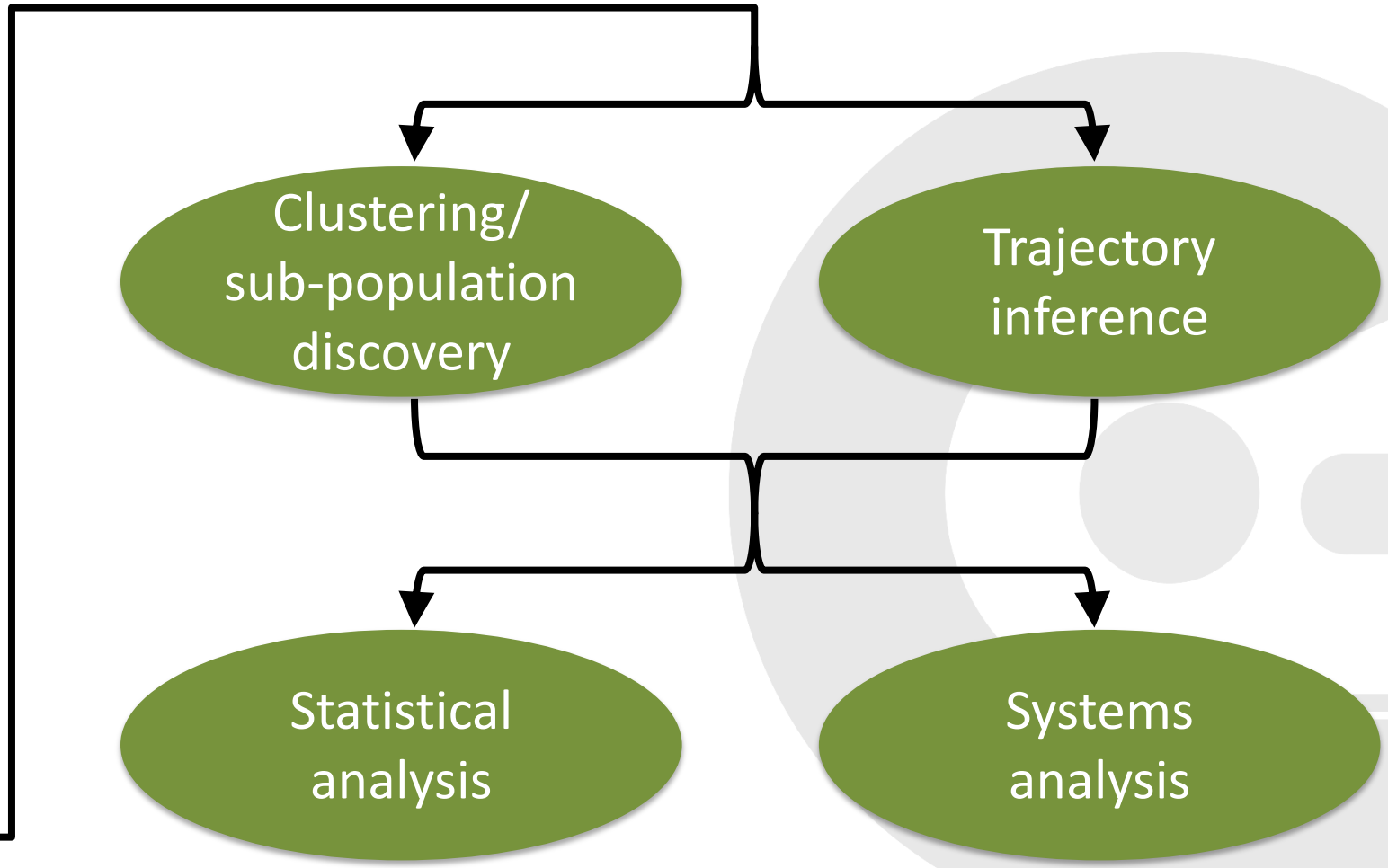
- scRNA libraries require a lot of PCR amplification
  - Don't want to count PCR duplicates in quantification
- UMIs are added during reverse transcription, before amplification
  - Random 6-12nt sequences (length varies by platform; always the same for a given technology)
- Identify PCR duplicates based on:
  - Same UMI on the same gene (3' RNA-seq)
  - Same UMI at the same position (complete transcripts)
- Quantification is ultimately a **UMI count** (especially for 3' chemistries)
  - *The total counts per gene is irrelevant*



scRNA-seq analysis outline

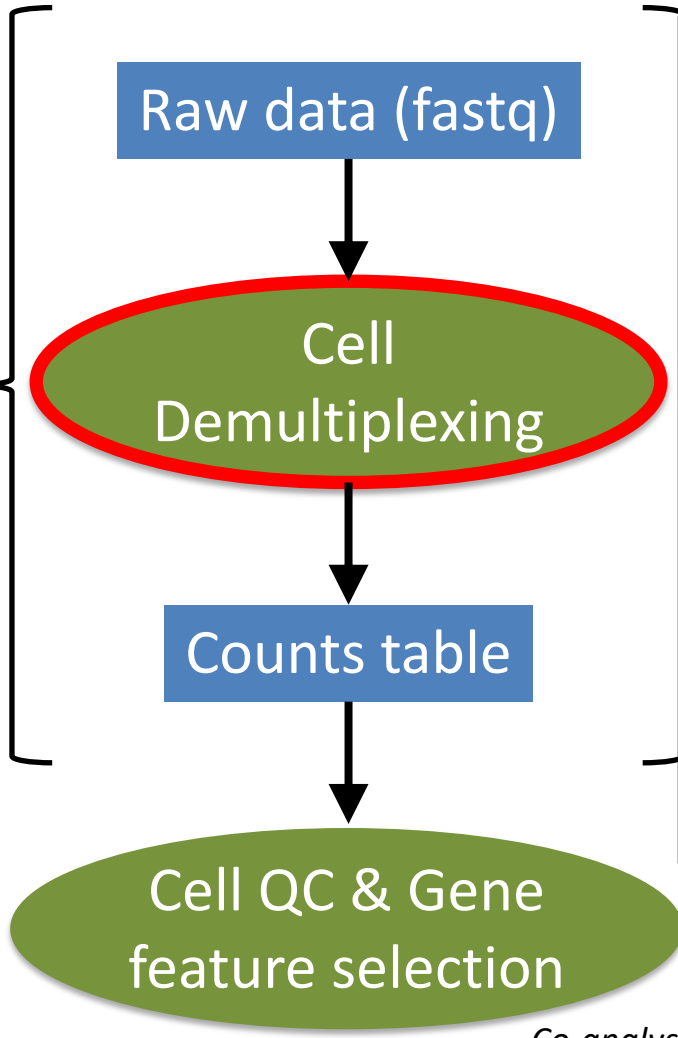


Co-analysis of multiple samples



cribioinfo@uic.edu

scRNA-seq analysis outline: demultiplexing



Demultiplexing is platform-specific

- 10X – CellRanger
- inDrop and DropSeq have different tools (RIC has developed our own for inDrop)

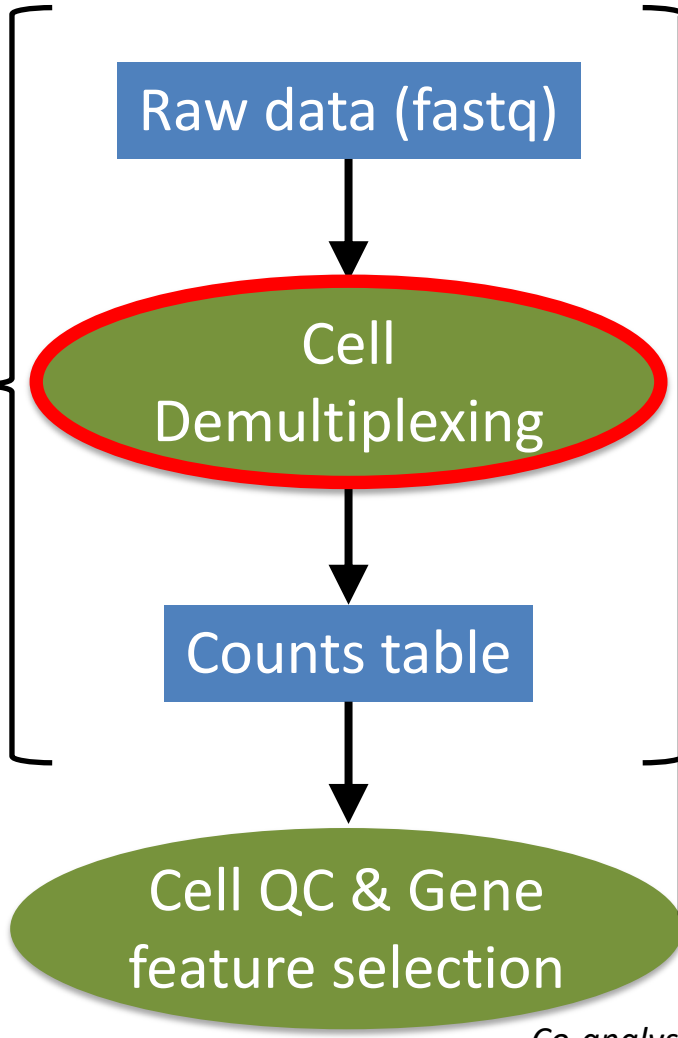
Computationally expensive

- We will not practice today – use already-demultiplexed data
- Ask the sequencing facility or your bioinformatics core facility (us!) for assistance if you cannot set up on your own

Outline of process is good to know for your own understanding

Co-analysis of multiple samples

scRNA-seq analysis outline: demultiplexing pt 1



Co-analysis of multiple samples

Raw data is paired-end

1. Identify cell barcodes from one end
2. Map cDNA sequence from other end

Example: inDrop v2 chemistry

R1: cDNA sequence (from 3')

`ATCTAGTAGGTCGTATATATGCGCGGAGGGCGCTAGCGTAGTCGACGTATAGGCGAGAA`

→ **Map to reference genome**

R2: Barcode + UMI

`BBBBBBBB[BBB]WWWWWWWWWWWWWWWWWWWWWWCCCCCCCCUUUUUUTTTTTTTTTT`

`Barcode1 Adapter Barcode2 UMI`

→ **Barcode1 + Barcode2 = cell identifier**

cribioinfo@uic.edu

scRNA-seq analysis outline: demultiplexing pt 2

Raw data (fastq)

Cell Demultiplexing

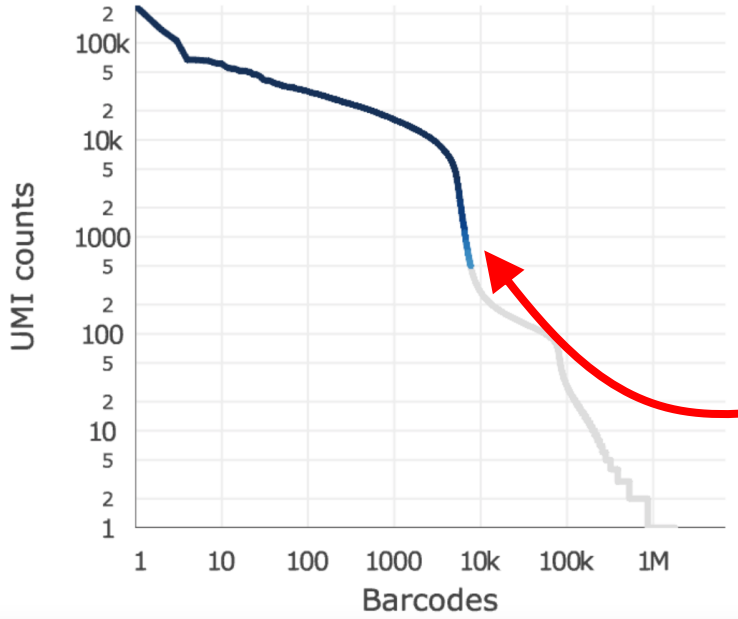
Counts table

Cell QC & Gene feature selection

Co-analysis of multiple samples

Estimate number of cells and generate counts table

1. Count unique UMIs mapping to each gene for each cell barcode
2. Plot cell barcodes in descending order by total UMI counts

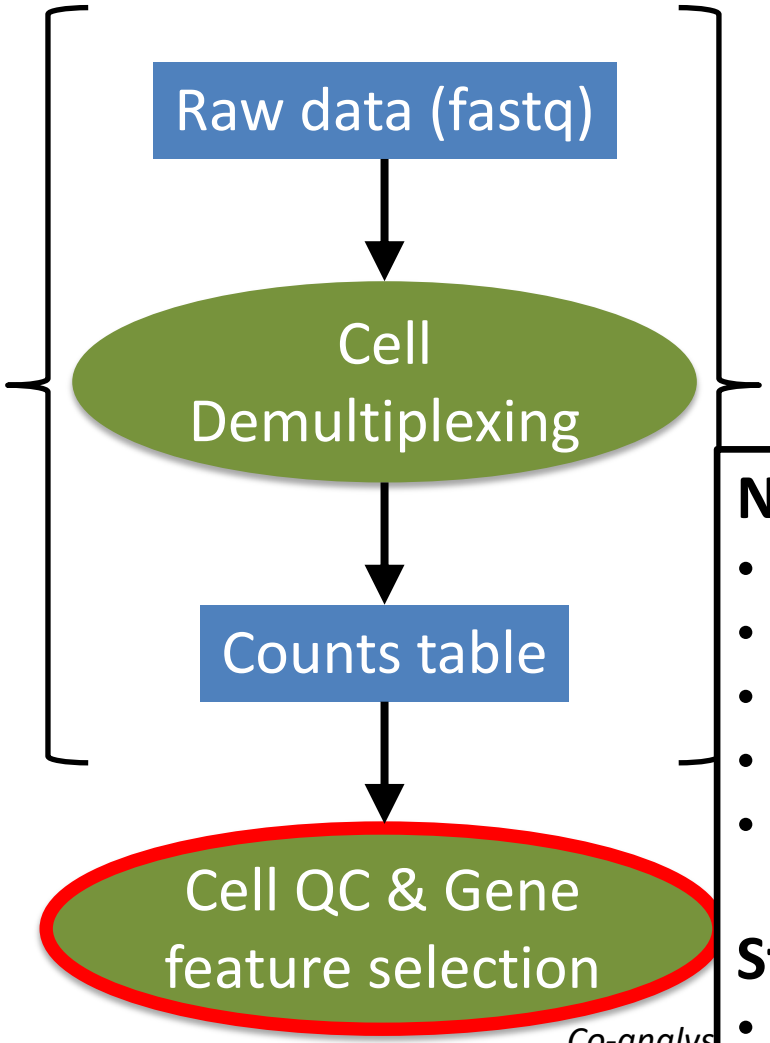


c

- Typically >100,000 cell barcodes identified
- Estimate count of “real” cells based sharp drop in the plot

cribioinfo@uic.edu

scRNA-seq analysis outline: Cell QC and Gene feature selection



Clustering/ sub-population discovery

Trajectory inference

Need to ensure data used downstream is robust

- Remove low-quality cells (dead or dying)
- Remove doublets?
- Normalization of gene expression
- Control for cell cycle?
- Feature selection for analysis

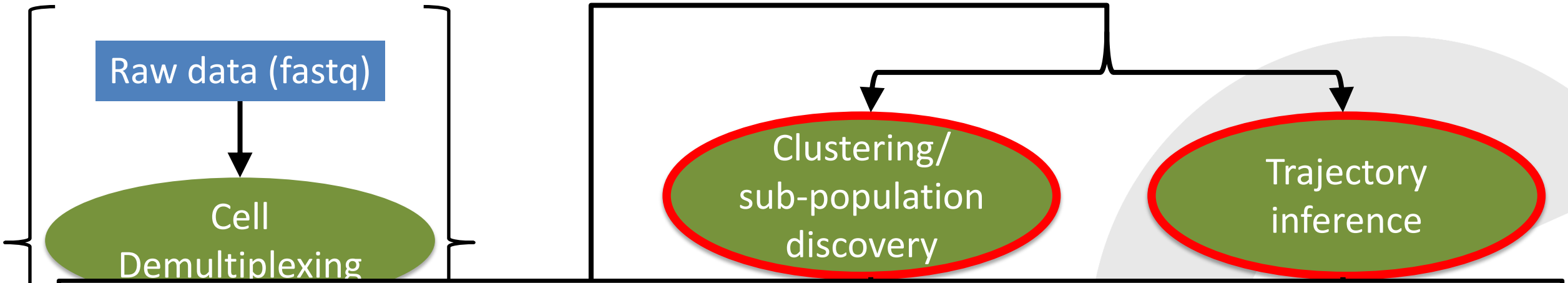
Steps are mostly the same regardless of platform

- Combine all samples to analyze together

Co-analys
multiple samples

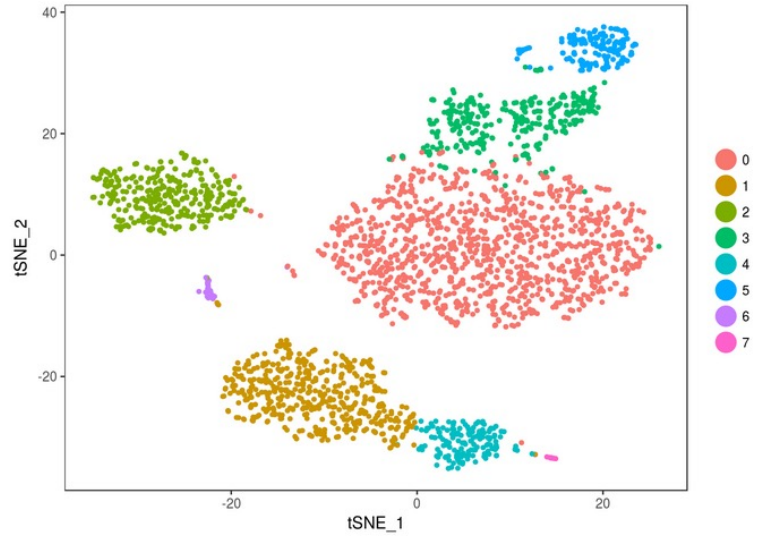
cribioinfo@uic.edu

scRNA-seq analysis outline: downstream analysis pt 1



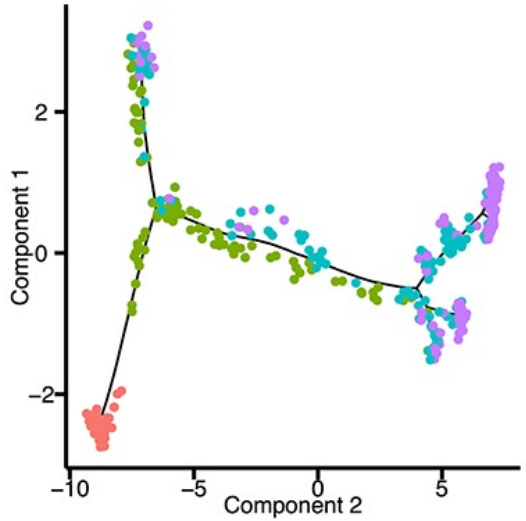
Clustering *(today – Seurat)*

- Analysis of cells into discrete groups
- Discovery & characterization of cell types

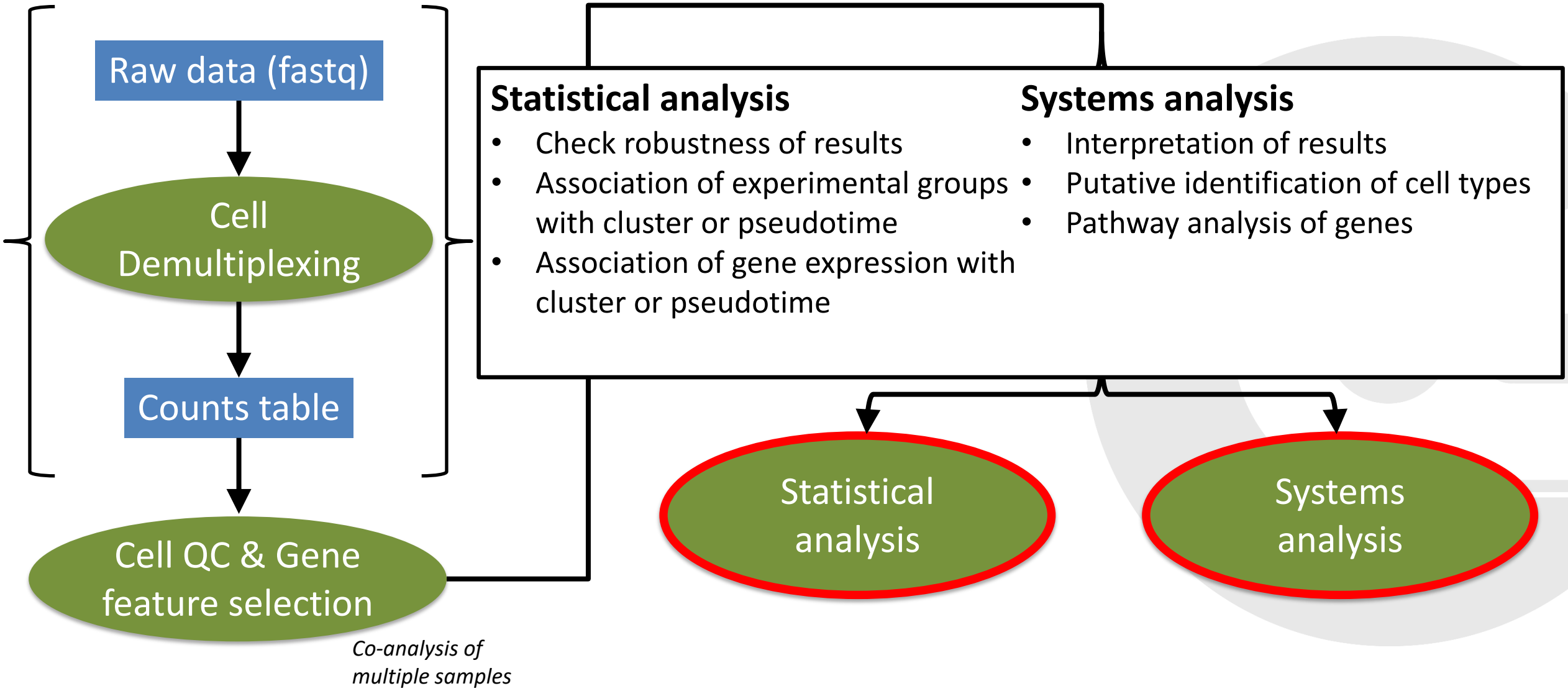


Trajectory inference *(lecture and take-home)*

- Pseudotime, RNA velocity, CytoTrace
- Inference of developmental processes



scRNA-seq analysis outline: downstream analysis pt 2



cribioinfo@uic.edu

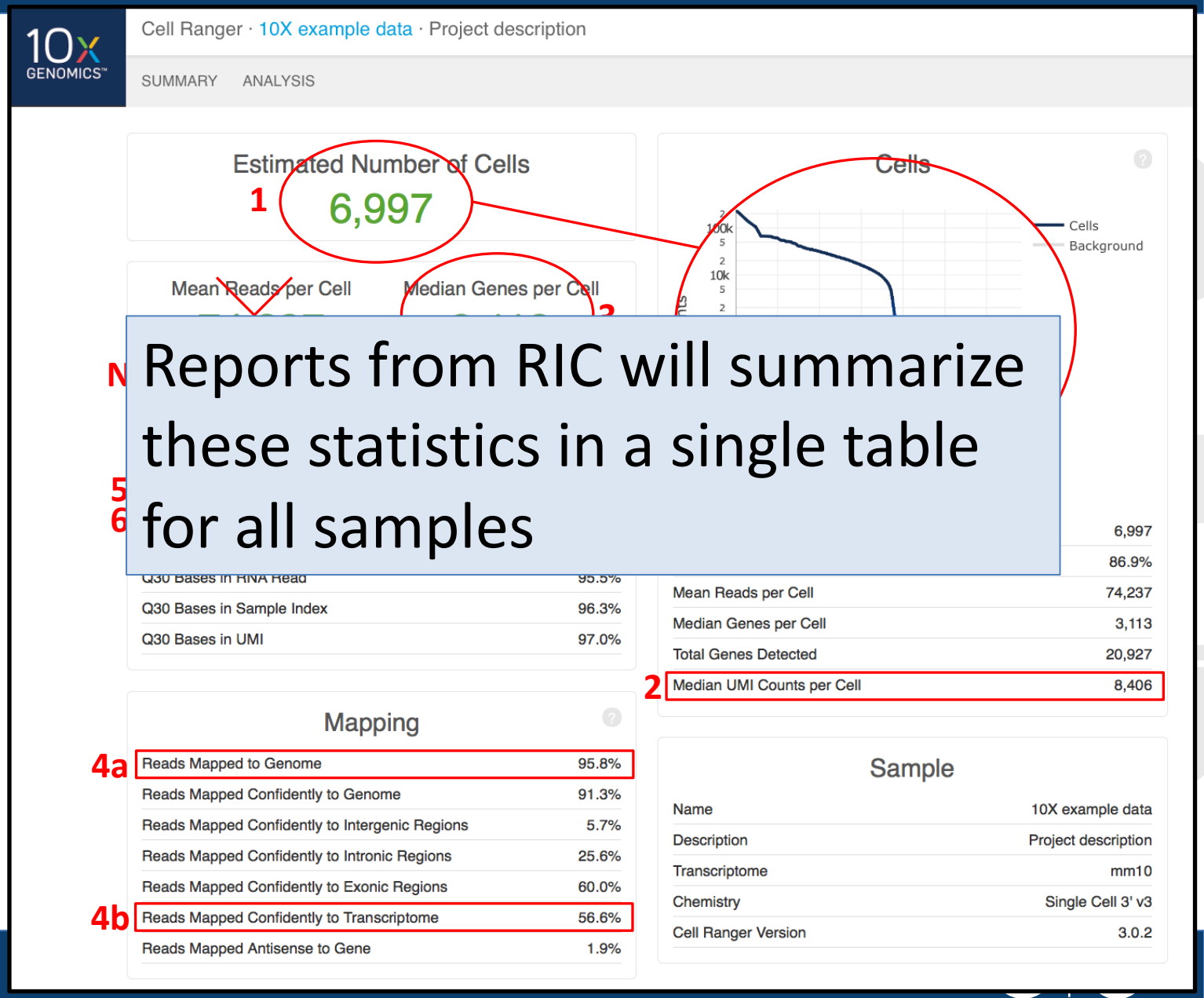
Exercise 1.1: CellRanger Report

`https://wd.cri.uic.edu/sc_rna/10X_example_report.html`

Details for the report are on the next slide…

Exercise 1.1: Demultiplexing QC checks (10X example)

1. Number of cells (estimate)
2. Median UMIs per cell *(not mean reads)*
3. Median genes per cell
4. Mapping %
   a. To genome (>90%)
   b. To transcriptome (~60%)
5. Barcode identification % (>95% for 10X)
6. Sequencing saturation/PCR duplication rate



cribioinfo@uic.edu

Counts table

- Two general formats:
  - Full matrix
  - Sparse matrix

Counts table: full matrix

- Two general formats:
  - Full matrix
  - Sparse matrix

- **Table with genes down the rows, cells across the columns**
- **More typical of DropSeq, inDrop pipelines**

Gene	cell1	cell2	cell3	cell4	cell5	cell6	cell7	cell8	cell9
ENSMUSG00000051951	215	297	259	116	175	261	267	169	238
ENSMUSG00000102851	1108	979	1199	1168	905	961	1286	4	6
ENSMUSG00000103147	1221	875	950	648	775	736	1096	15	16
ENSMUSG00000102331	20	9	21	11	14	9	9	8	28
ENSMUSG00000102343	1177	876	916	617	746	756	1081	15	12
ENSMUSG00000025900	1089	766	840	583	565	828	972	19	15
ENSMUSG00000102948	651	616	594	536	525	432	372	274	388
ENSMUSG00000025902	617	484	493	521	376	324	395	394	337
ENSMUSG00000104238	885	727	973	893	688	792	1026	2	6

File size: ~25-250M unzipped, ~2-20MB zipped.

Counts table: sparse matrix

- Two general formats:
  - Full matrix
  - Sparse matrix

- **Sparse format lists gene/cell/counts trios; 0s are omitted**
- **Typical of 10X data – CellRanger uses "MatrixMarket" format**
- **More efficient for sparse data (lots of genes are expressed only in a few cells)**

1. List of gene index, cell index, count
(matrix.mtx.gz)

```
%%MatrixMarket matrix coordinate integer general
%metadata_json: {"format_version": 2,
"software_version": "3.0.0"}
31053 1301 4220492
30976 1 73
30974 1 1
30973 1 56
30972 1 2
30971 1 11
30970 1 116
30969 1 123
```

Total genes, cells, and counts in entire data set

Gene 30973, cell 1, 56 counts

File size: ~10-150M zipped.

2. List of gene IDs
(features.tsv.gz)

```
ENSMUSG00000051951 Xkr4     Gene Expression
ENSMUSG00000089699 Gm1992   Gene Expression
ENSMUSG00000102343 Gm37381  Gene Expression
ENSMUSG00000025900 Rp1      Gene Expression
ENSMUSG00000025902 Sox17    Gene Expression
ENSMUSG00000104328 Gm37323  Gene Expression
ENSMUSG00000033845 Mrpl15   Gene Expression
ENSMUSG00000025903 Lypla1   Gene Expression
ENSMUSG00000104217 Gm37988  Gene Expression
ENSMUSG00000033813 Tcea1    Gene Expression
```

3. List of cell IDs
(barcodes.tsv.gz)

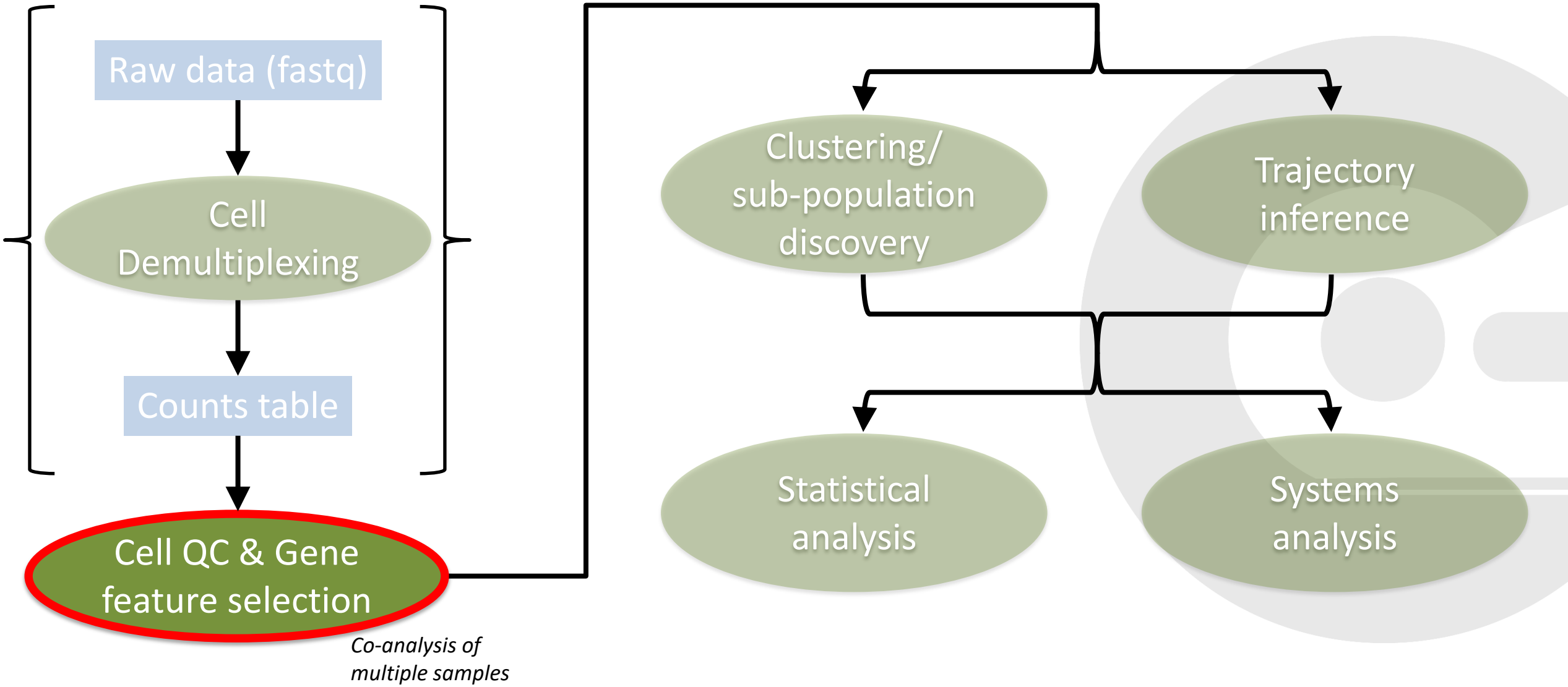
```
AAACGAATCAAAGCCT-1
AAACGCTGTAATGTGA-1
AAACGCTGTCCTGGGT-1
AAAGAACCAGGACATG-1
AAAGGTACACACGGTC-1
AAAGTCCAGTCACTAC-1
AAAGTCCGTGACTGTT-1
AAAGTCCTCCAGCCTT-1
AAAGTGAGTTCCTAAG-1
AAAGTGATCAGTGGGA-1
```

Exercise 1.2: Read data into Seurat

Practice with

- InDrop data (full matrix)
- 10X data (sparse matrix)
  - For 10X, need to download zip file and unpack first

Cell QC and Gene feature selection



cribioinfo@uic.edu

Analysis of multiple samples

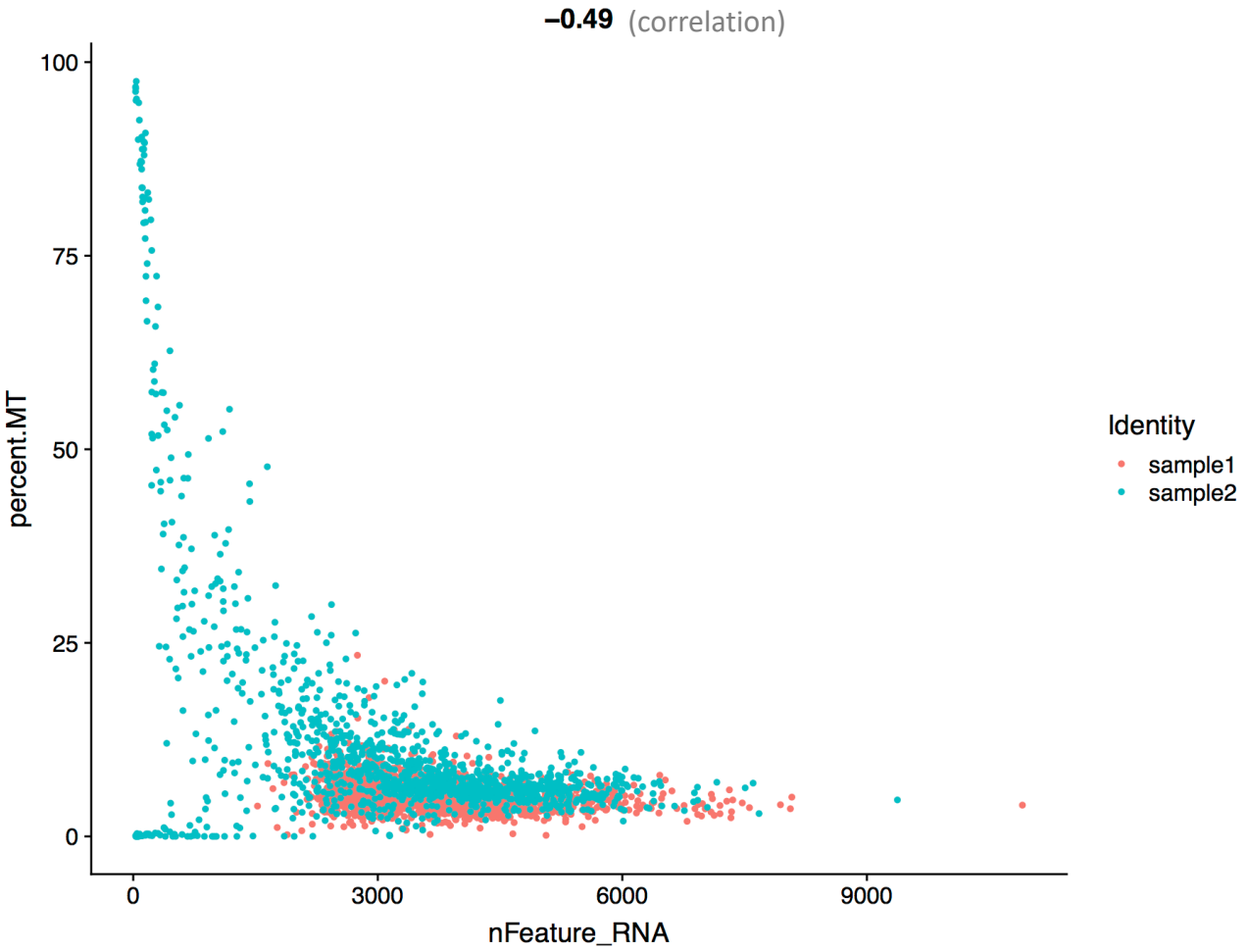
- Read in data sets one at a time to Seurat
- Use `merge()` function to combine together into a single object
- Then do downstream steps (QC, feature selection, clustering, etc.)
- We can differentiate samples later using:
  - `$orig.ident` in Seurat object: stores the "project" name
  - Adding prefixes to cell IDs in the `merge` call
- Cells from each sample can reinforce our power/resolution during clustering
- Easy to make apples-to-apples comparisons between cell clusters across samples

Quality control steps for single-cell RNA-seq

- Ensure data used for clustering/downstream analysis is high-quality
- Cell QC: remove bad cells
  - Low counts or low number of genes
  - High mitochondrial counts
- Gene feature selection
  - Look for more variable (informative genes)
- Normalization
  - Account for differences in sequencing depth, expression level
  - Data smoothing with PCA

Cell QC steps

- Remove cells with very low expression or low numbers of genes
  - Likely to be less reliable transcriptome measurements, more likely to be dead/dying cells
- Remove cells with high fraction of mitochondrial expression
  - Mitochondria spill open in dying cells
- Remove really highly expressed cells???
  - Possible doublets
- Check diagnostic plots to determine thresholds



cribioinfo@uic.edu

Exercise 1.3: Start QC steps for single-cell practice data set

- Read in two 10X data sets
  - Need to download zip file and unpack first
- Check cell QC statistics and do cell filtering

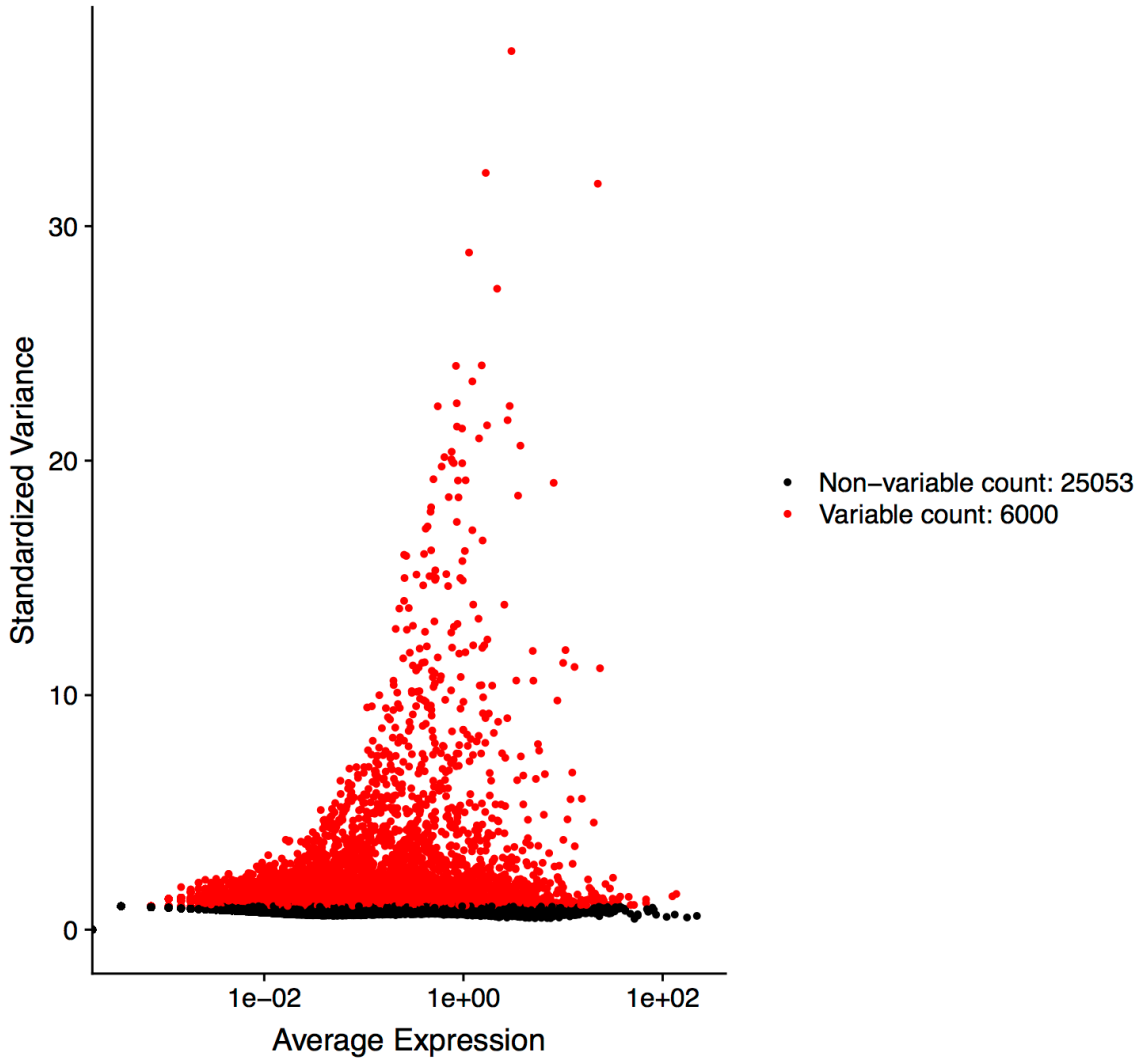
BREAK

Gene feature selection steps: normalization

- Normalize for differences in sequencing depth
  - LogNormalize: log-CPM, but per 10,000 instead of per millions (default)
  - CLR: centered log ratio transformation (similar to TMM)
  - RC: relative counts: linear CPM
- Z-score?
  - Positive: highlight *changes* in expression rather than levels, high expressed genes may not be the most meaningful
  - Negative: may amplify noise in low-expressed genes
- Account for cell cycle?
  - Have to know which cell cycle genes, and if their expression is independent of the biological effects of interest
- Difficult to know the optimal strategy
  - Normalization assumes there's some quantity that should be the same across cells, and we normalize to this
    - But we don't know that this is true
    - Different cells probably have different absolute expression levels

Gene feature selection steps: gene selection

- Select highly variable genes for clustering analysis
  - Control for expression level
  - 3 methods available within Seurat
- **This is not the only way**
  - Do we need to filter at all?
  - Could also look at % of cells expressing each gene
  - If interested in rare sub-populations, we may want to focus on genes expressed very infrequently

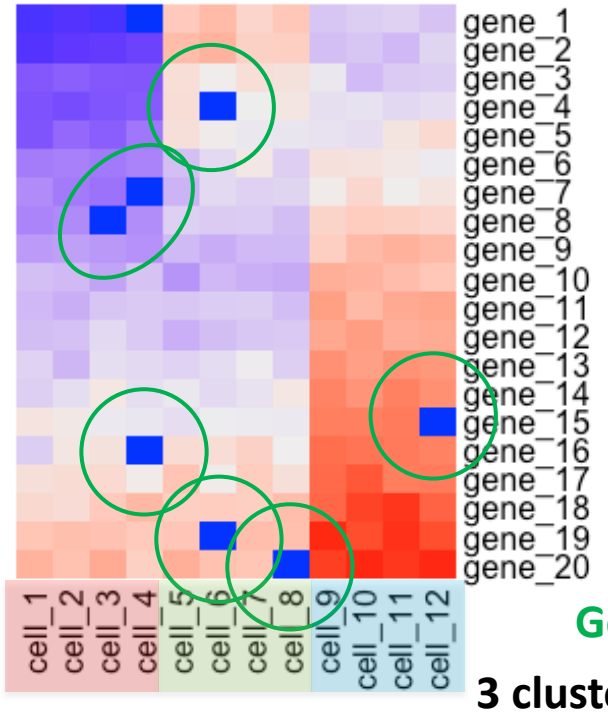


cribioinfo@uic.edu

Truncating noise with PCA

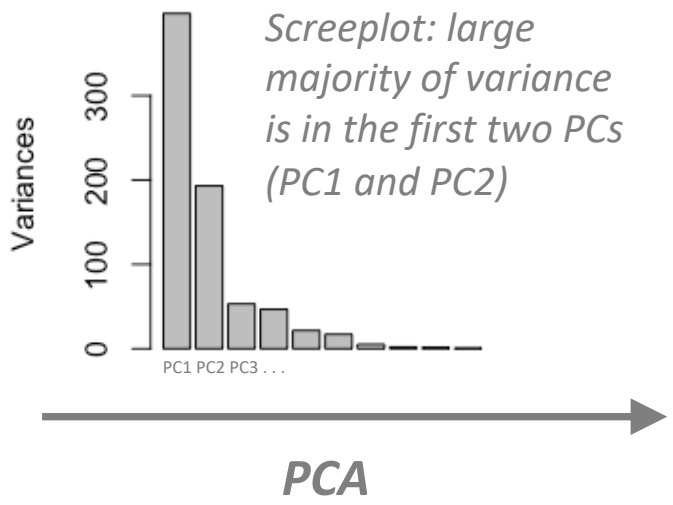
- One way to deal with sparsity in RNA-seq data:
  - Run PCA, take top principal components (PCs), cluster on these features
  - Noise will be captured & removed in bottom PCs
  - Retained PCs are a weighted combination of all genes, but (ideally) without the “noise”

Original data



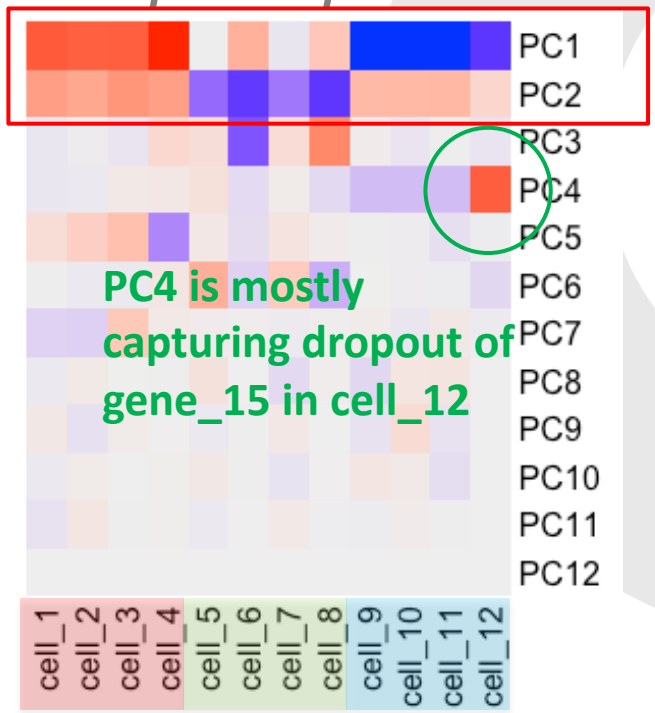
Gene dropouts

3 clusters of cells



Screeplot: large majority of variance is in the first two PCs (PC1 and PC2)

Principal components



PC4 is mostly capturing dropout of gene_15 in cell_12

PC1 and PC2 show associations with cell clusters, and are relatively free of noise: just use these for clustering.

Other PCs capture variability that we will disregard.

cribioinfo@uic.edu

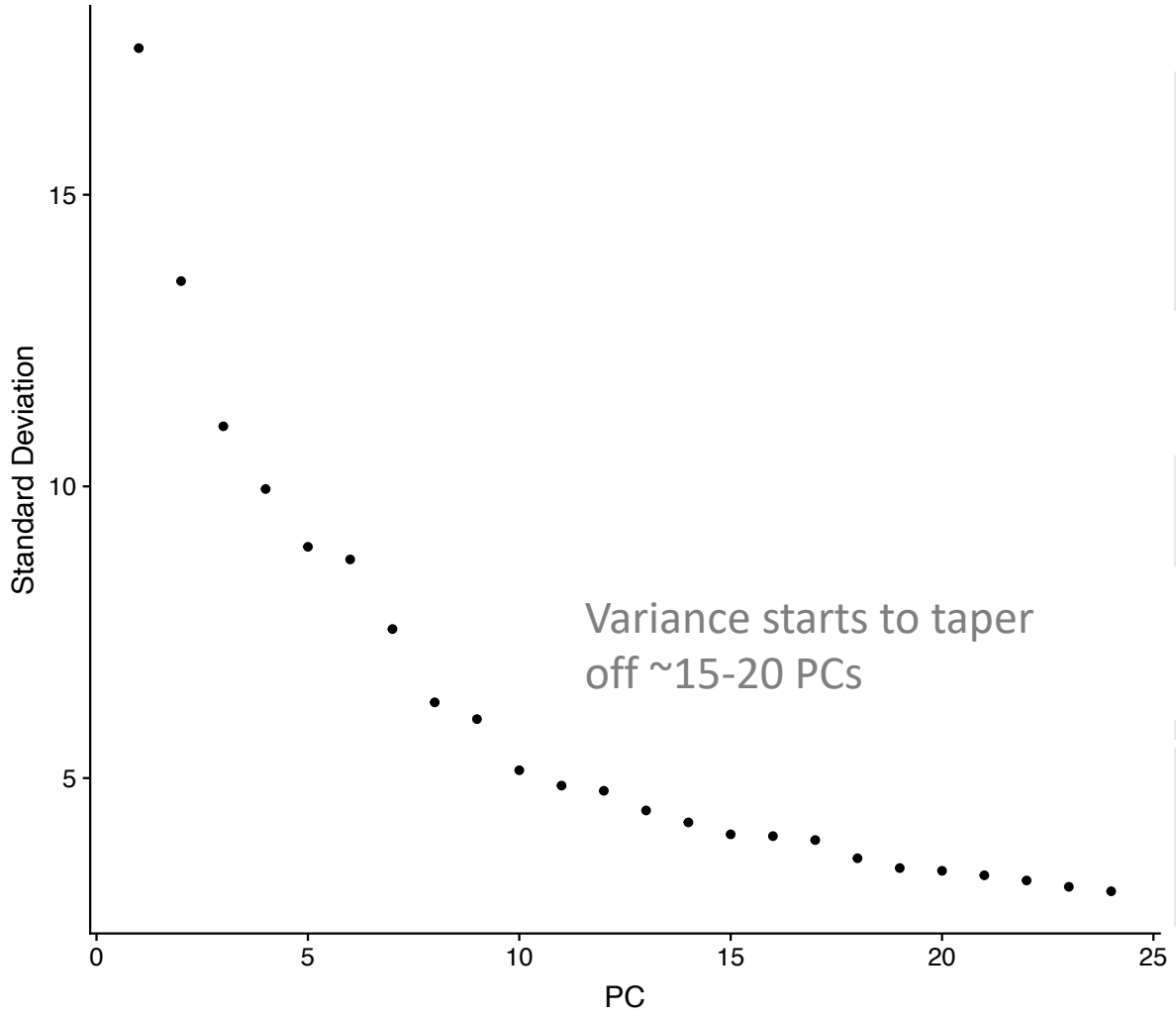
Things to think about with PCA

- Upstream QC choices will change PCs
  - Exclusion of “bad” cells
  - Normalization/scaling
  - Use all genes, or just the variable ones?
- Several options available for informing on right the number of PCs, but interpretation is *qualitative*
  - Screeplot/ElbowPlot
  - PC heatmap
  - JackStraw
- If unsure, better to aim for more PCs than fewer

cribioinfo@uic.edu

Things to think about with PCA

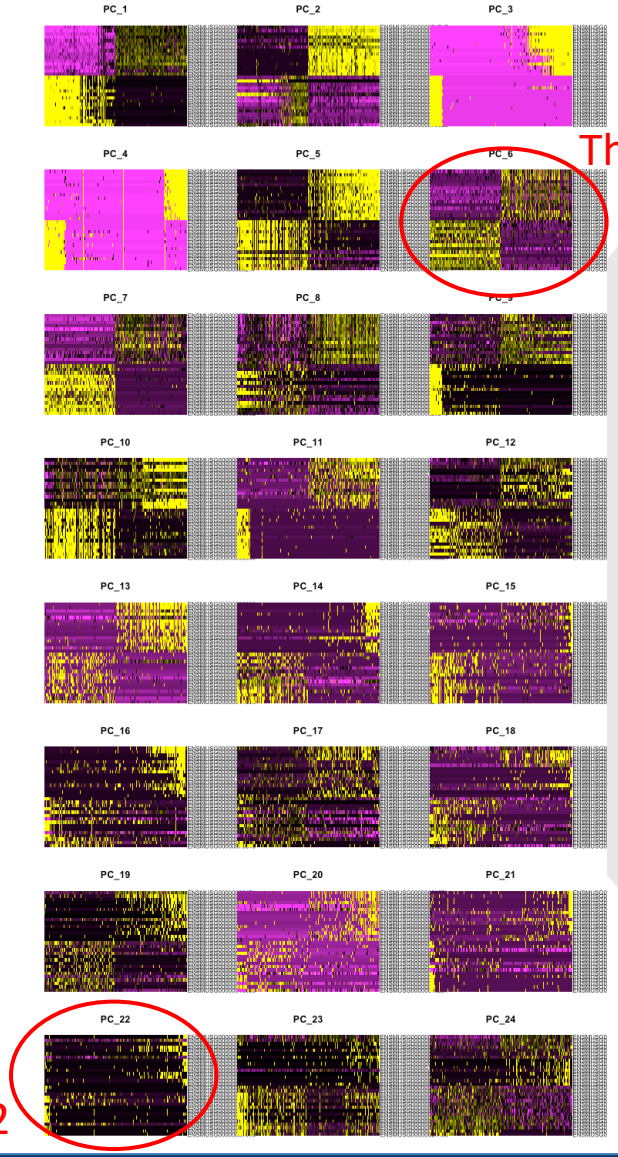
- Upstream QC choices will change PCs
  - Exclusion of “bad” cells
  - Normalization/scaling
  - Use all genes, or just the variable ones?
- Several options available for informing on right the number of PCs, but interpretation is *qualitative*
  - Screeplot/ElbowPlot
  - PC heatmap
  - JackStraw
- If unsure, better to aim for more PCs than fewer



cribioinfo@uic.edu

Things to think about with PCA

- Upstream QC choices will change PCs
  - Exclusion of “bad” cells
  - Normalization/scaling
  - Use all genes, or just the variable ones?
- Several options available for informing on right the number of PCs, but interpretation is *qualitative*
  - Screeplot/ElbowPlot
  - PC heatmap
  - JackStraw
- If unsure, better to aim for more PCs than fewer



There are differences at PC6

Plot PCs for top genes and top cells in a heatmap.

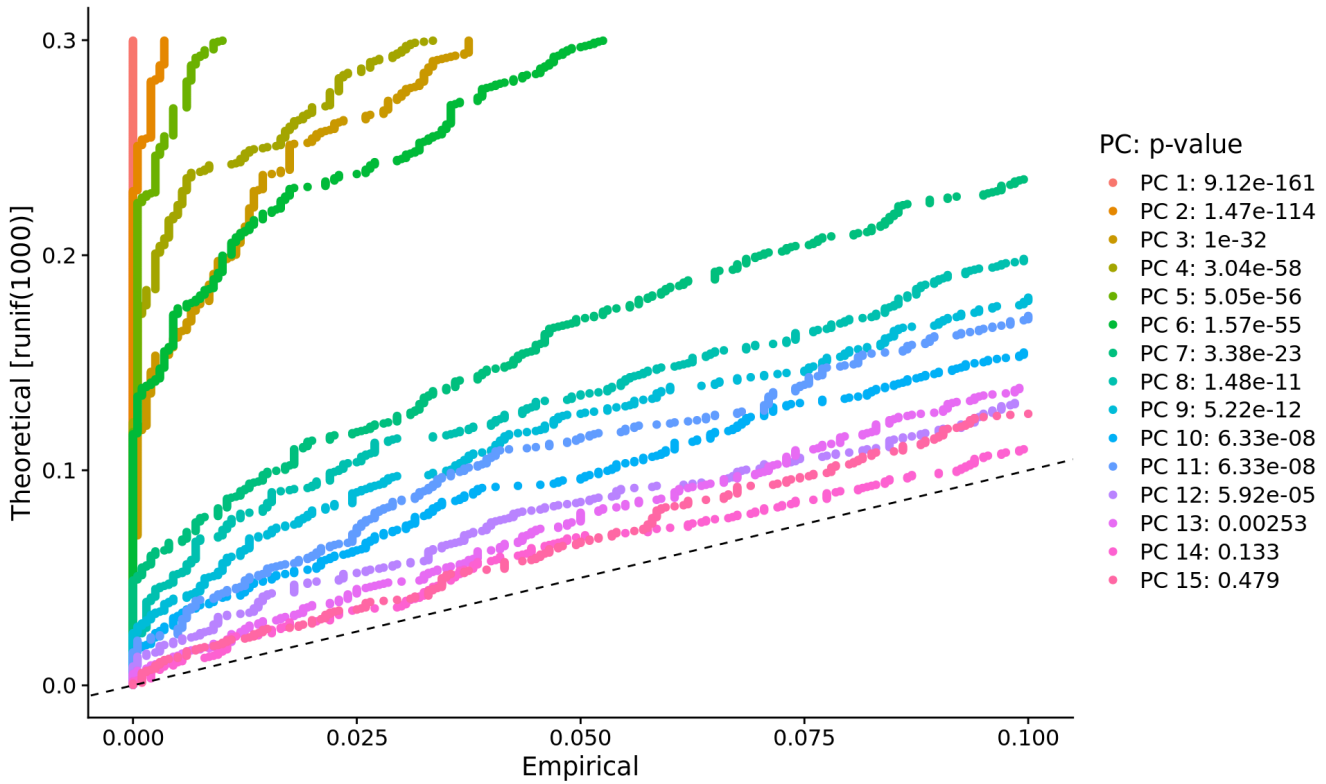
Look qualitatively for “signal” based on block pattern in top/bottom/left/right quadrants.

Pattern is weak at PC22

cribioinfo@uic.edu

Things to think about with PCA

- Upstream QC choices will change PCs
  - Exclusion of “bad” cells
  - Normalization/scaling
  - Use all genes, or just the variable ones?
- Several options available for informing on right the number of PCs, but interpretation is *qualitative*
  - Screeplot/ElbowPlot
  - PC heatmap
  - JackStraw
- If unsure, better to aim for more PCs than fewer



Compute P-values for PCs based on permutation test

- Computationally intensive (we’ll skip for today)
- PC may be “significant” even if only mildly informative – still requires a judgement call

Exercise 1.4: Gene feature selection steps

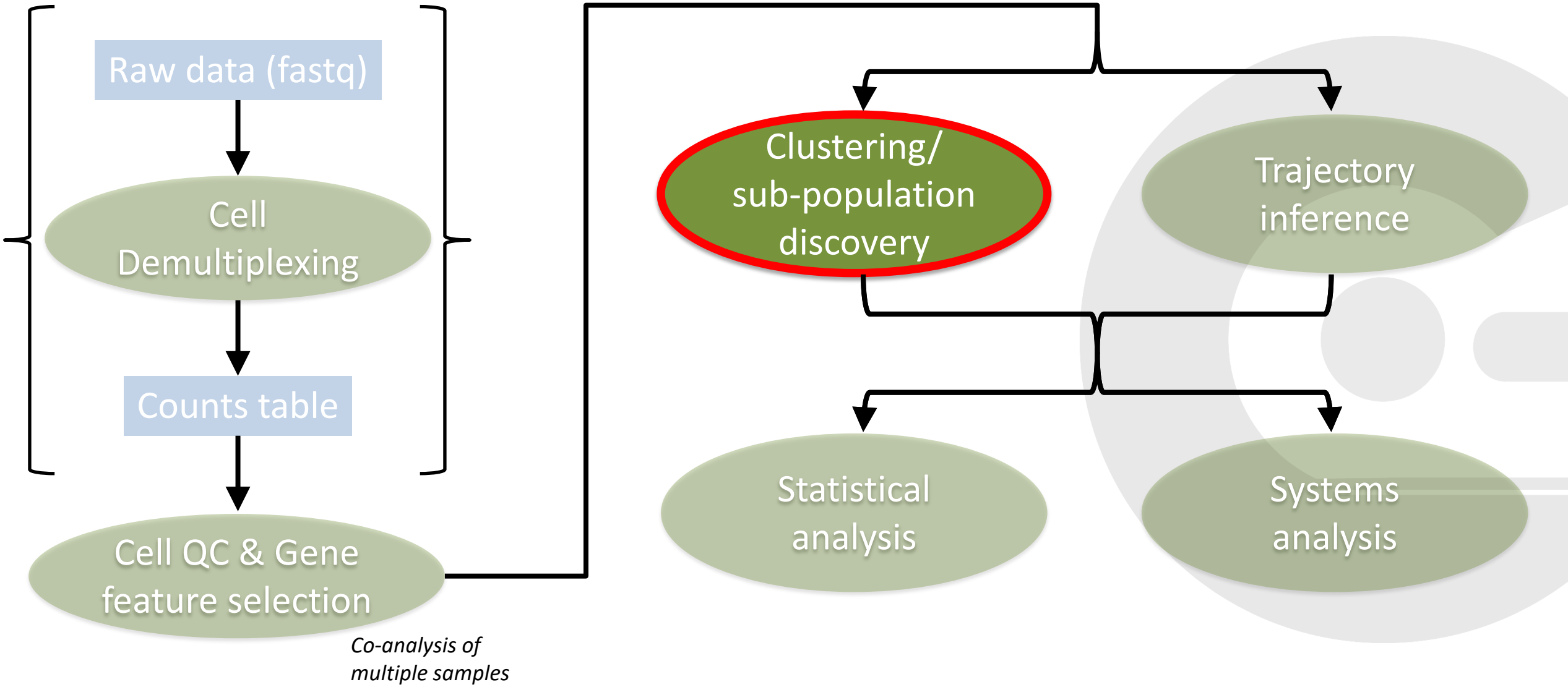
- Gene normalization
- Find variable genes
- Z-score
- PCA

cribioinfo@uic.edu

Additional notes on feature selection

- In general, we would recommend plotting and testing the top 50 or 100 PCs.
  - The number of PCs with significant levels of signal *depends* on the diversity of the cells in your sample
  - The ElbowPlot (aka screeplot), DimHeatmap, and JackStraw plots are all useful diagnostics to assess the amount of information contained in each PC.
- The interpretation of all of these is still somewhat subjective.
- ***You*** need to make the final determination of how many PCs dimensions to keep.
  - If in doubt, better to err on the higher side (more PCs) than the lower side.

Clustering



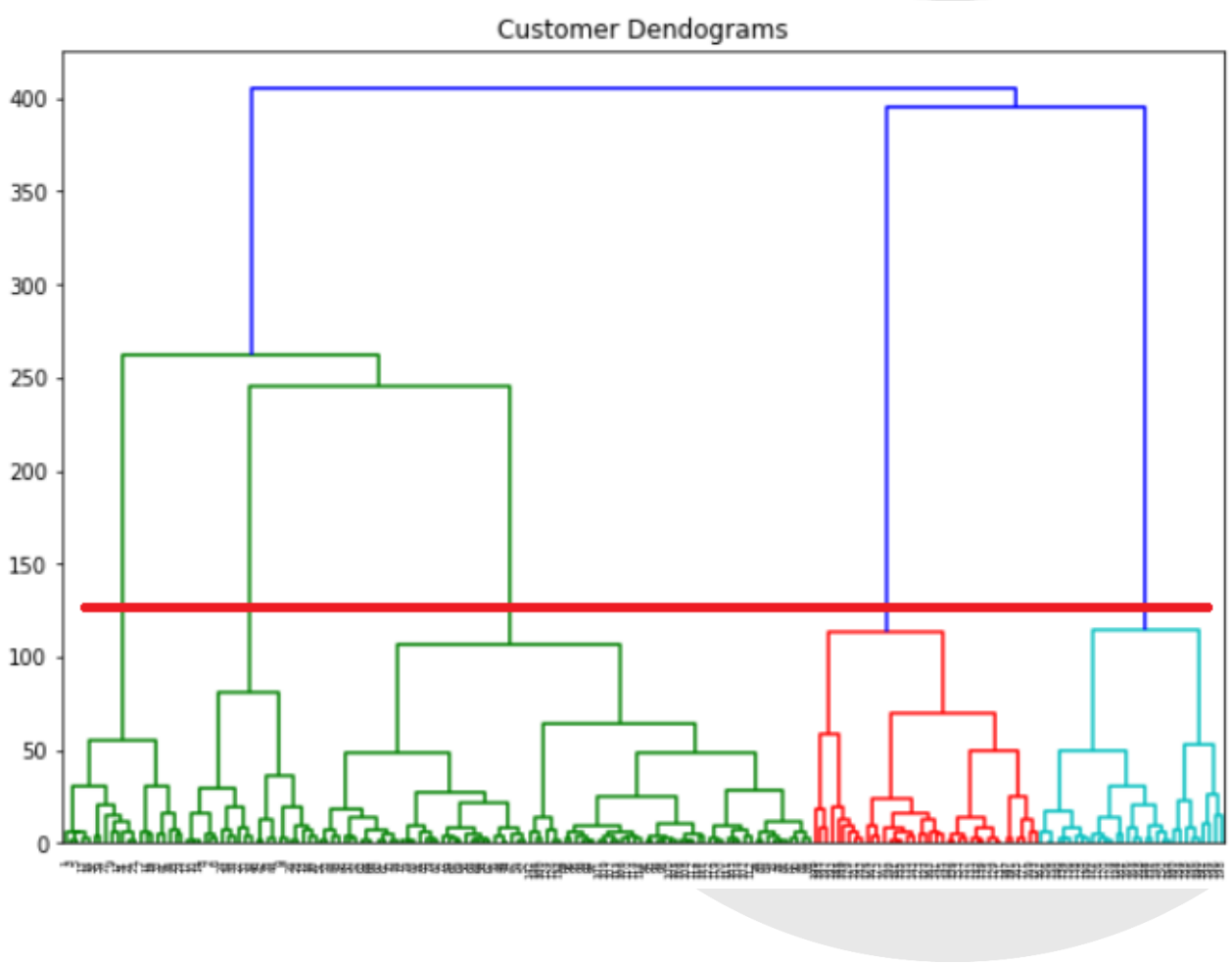
cribioinfo@uic.edu

Sub-population discovery (Clustering)

- Want to identify sets of cells that are relatively homogenous
- Common types of algorithms:
  - Hierarchical clustering – infer dendrogram from samples
  - K-means clustering – group cells based on similarity to centroids
  - Density/graph-based – find separations between groups of cells based on a minimum similarity metric
- **All methods will give different results**
- **All methods will be affected by upstream cell QC and gene feature selection choices**
- **All methods will be affected by parameters that you have to choose**
  - There may be default values, but these may not be optimal
  - These often include a distance metric, and a random number seed, plus other algorithm-specific choices

Hierarchical clustering

- Identify a dendrogram based on cell-cell similarities
  - Different algorithms for building the dendrogram
- Pros:
  - Gives a lot of detail about hierarchical relationships
  - Dendrograms are pretty
- Cons:
  - Requires user's selection of a "cut point" to identify discrete clusters
  - Scales poorly with large number of cells (quadratic scaling): becomes unusable with more than ~10,000 cells



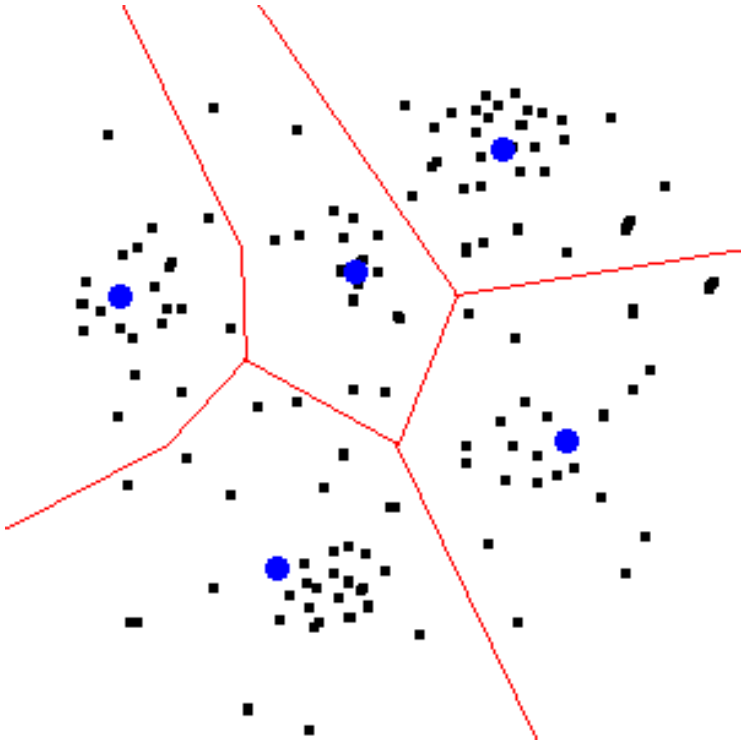
cribioinfo@uic.edu

K-means clustering

- Associates cells based on distance to a centroid
  - Centroids chosen randomly from the data
  - Updated until converged
- Pros:
  - Scales very well with large data sets
  - Tends to give roughly even-sized clusters
- Cons:
  - Requires *a priori* choice of the number of clusters K
  - May be sensitive to random initialization (initial centroid choice)
  - Tends to give roughly even-sized clusters (maybe you don't want that)

We can address these issues with adaptations to the algorithm

Centroids and cluster assignments

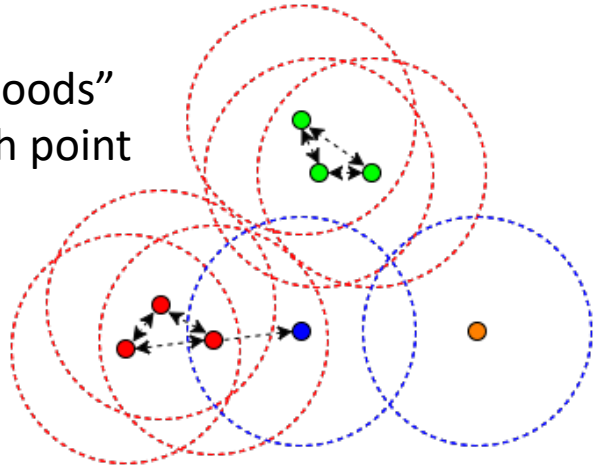


cribioinfo@uic.edu

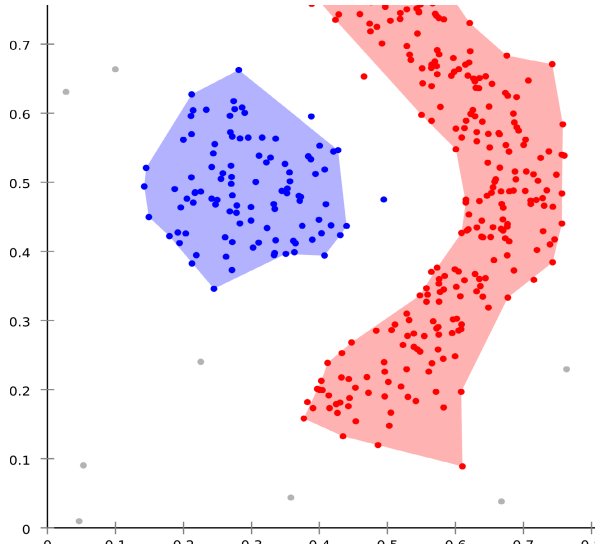
Density/graph-based clustering

- Methods: DBSCAN/OPTICS, **Louvain clustering (in Seurat)**, many others…
- Group cells based on a minimum distance
  - Look for other cells within the “neighborhood” of each cell
  - Cluster cells together until you reach a separation
- Pros:
  - Don’t need to set number of clusters
  - Can find clusters with very different sizes and shapes
- Cons:
  - Results are sensitive to the “neighborhood” distance
  - “Curse of dimensionality”: neighborhoods get very sparse with high dimensional data
    - Feature reduction with PCA is important
  - May scale poorly, depending on implementation (Seurat’s implementation scales fine)
  - Some cells may be “unclustered” (which may be OK)

“Neighborhoods” around each point



Clusters of varied shapes separated by boundaries



cribioinfo@uic.edu

Visualization of clusters: tSNE or UMAP

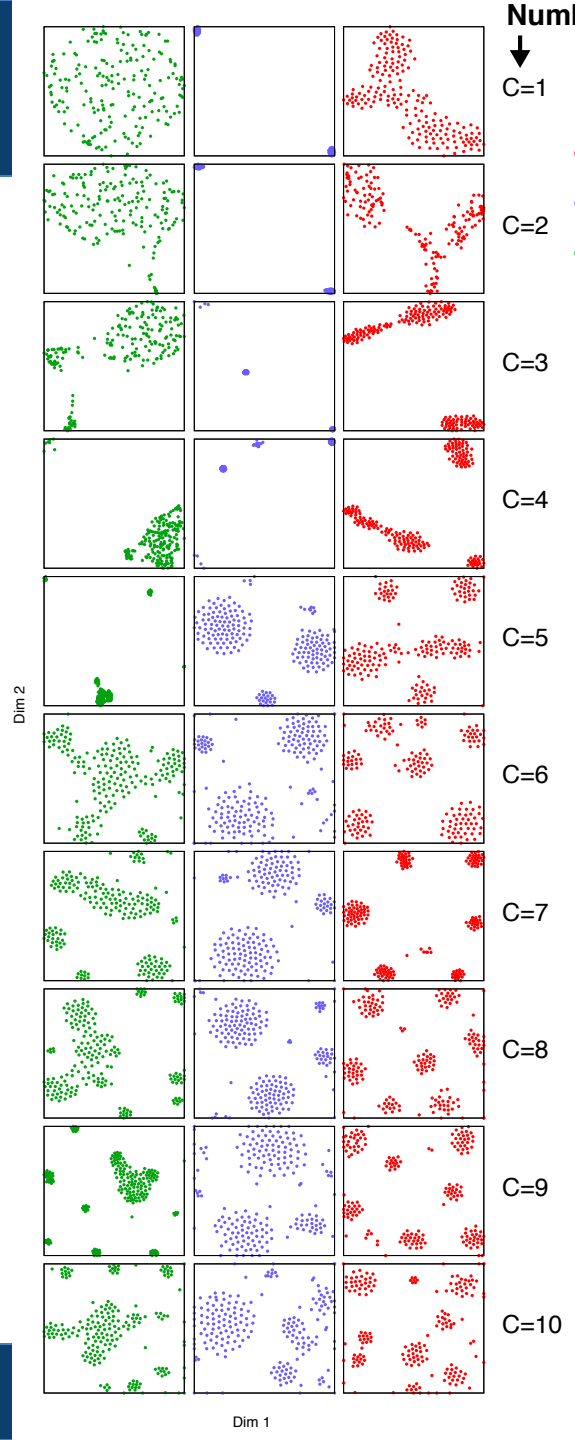
- tSNE: Computes a probability distribution of pairs of points: high probability for similar objects, low for dissimilar objects
  - Computes this probability in the original space, and in the reduced space, minimizes the difference
  - Result is dimensionality reduction – does not actually do any clustering/grouping of values
- Best used as a visualization option, NOT as an input to a clustering algorithm
  - Two dimensions may not be enough to capture all variability
  - Non-linear reduction may skew data
- Important parameter: **perplexity**
  - Similar to neighborhood distance in density/graph-based clustering

Sample data set in 2 dimensions with 3 clusters of points – tSNE run with different perplexity

https://distill.pub/2016/misread-tsne/

Always be skeptical!

- Understand how parameter choices affect outcomes
- Data visualizations can be misleading measures of clustering "quality" (robustness)
- Example: SIMLR
  - Computes distances using many different metrics, "learns" a similarity based on combination of metrics
  - Generates tSNE plot based on learned similarity metric
  - Requires choice of # clusters (C) first
  - BUT: the tSNE plot almost always looks "good" for *any* number of clusters



- Real scRNA-seq data set
- Simulated data set with 2 clusters
- Simulated data set with 1 cluster (i.e., random noise)

Analysis with SIMLR

SIMLR: Wang *et al.*, Nature Methods, 14, 414, 2017

Exercise 1.5: Clustering

- Run clustering on our data
- Visualizations
- Start differential gene expression across clusters
- Save our Seurat object as an R data file so that we can load it up again

cribioinfo@uic.edu

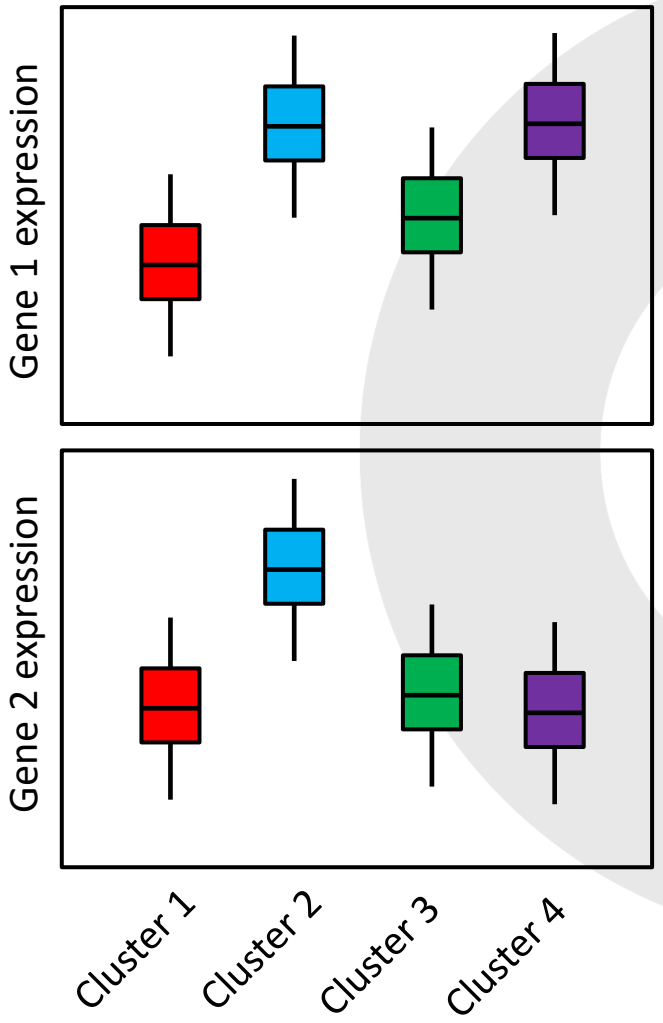
Differential expression across clusters

- Determine which genes are differentially expressed across clusters
- Several types of statistics available
  - Seurat has a number of options, but there are others also
  - Important to think about *how* the test works
- **These tests are NOT an independent confirmation of clustering quality**
  - Even clustering on random data will give you significantly differentially expressed genes: **p-values are inflated**
  - Use results to determine which genes are specific to different clusters

Comparison of what you get from differential statistics methods

- Non-parametric differential stats
  - Kruskal-Wallis (K-W): differences in median expression across multiple groups
  - Also Wilcox test for pair-wise comparisons (compare cluster 1 vs all other cells)
- ROC/AUC test (Receiver Operating Characteristic/Area Under the Curve)
  - Trains basic classifier.
  - Computes AUC value from trained classifier.
  - See if genes are always higher expressed in a cluster compared to all other clusters
    - No P-value, but gives a score between 0 and 1:
      - 0 = gene always lower in the cluster
      - 0.5 = random (no difference)
      - 1 = gene always higher in the cluster
- It can be interesting to run 2 different statistics and compare them

K-W and AUC tests differ in how they treat combinatorial differential expression



Combinatorial differential:
High in clusters 2 and 4
Significant for K-W
Not high value for AUC

Unique differential:
Only high in cluster 2
Significant for K-W
High value for AUC for cluster 2

cribioinfo@uic.edu

Other differential statistics

- In Seurat:
  - **“wilcox” (Wilcox rank-sum)**
  - **“roc” (ROC/AUC comparison)**
  - “bimod” (likelihood ratio test)
  - “t” (t-test)
  - “negbinom” (generalized linear model with negative binomial)
  - “poisson” (generalized linear model with poisson)
  - “LR” (logistic regression)
  - “MAST” (bioinformatics method called “MAST”)
  - “DESeq2” (use DESeq2 package)
- Not in Seurat:
  - Multi-group tests (like Kruskal-Wallis)
  - Fisher’s Exact test: test for frequency of presence/absence of expression

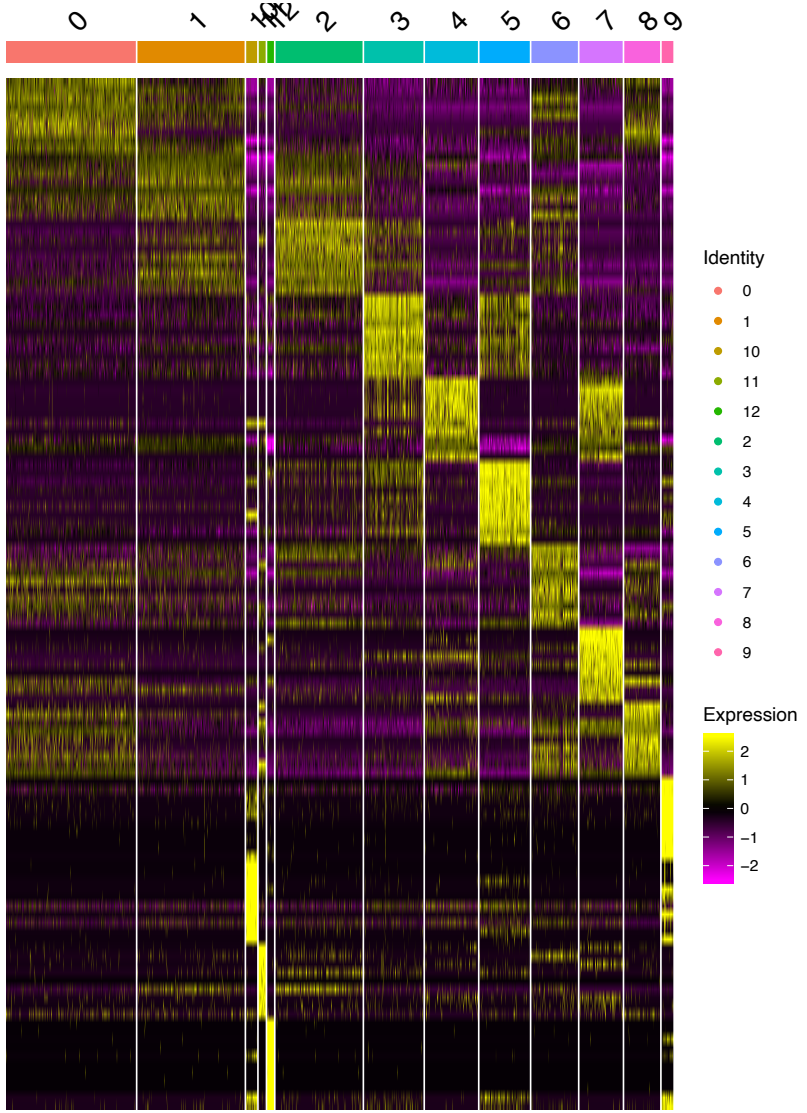
Takeaway: non-parametric tests (Wilcox or ROC) are a good place to start

- It’s not always clear if the assumptions in a parametric test are appropriate in scRNA-seq data
- You have plenty of power with lots of cells
- Keep in mind p-values will be inflated:
  - Clustering has generated strong separation between groups by design

What to do with differential statistics?

- Determine which features separate clusters
- Compare cell-specific gene sets across clusters
- Upload lists to pathway analysis software (IPA, DAVID)
- Visualize genes in heatmaps, tSNE plots, dotplots, etc.

We will cover in first exercise after lunch



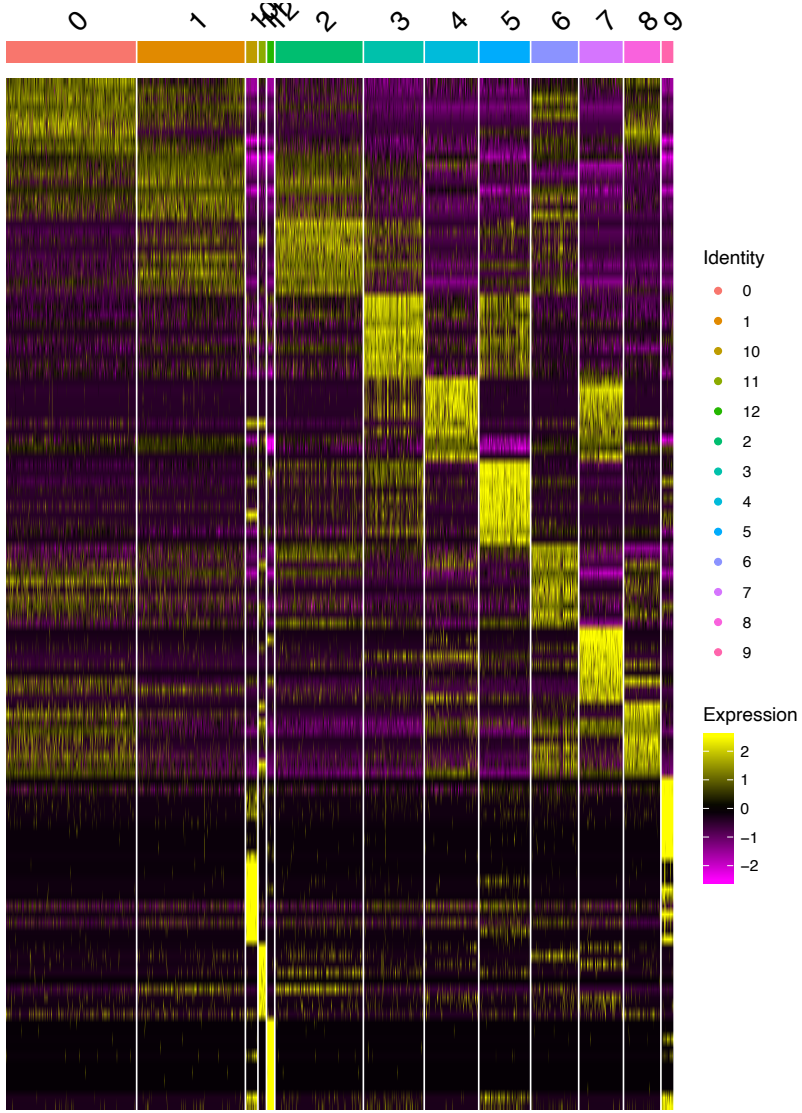
cribioinfo@uic.edu

LUNCH

Further analyses of Seurat results

What to do with differential statistics?

- Determine which features separate clusters
- Compare cell-specific gene sets across clusters
- Upload lists to pathway analysis software (IPA, DAVID)
- Visualize genes in heatmaps, tSNE plots, dotplots, etc.



cribioinfo@uic.edu

Cluster abundance across samples

- Count number of cells from each cluster in each sample
  - Use `table()` in R to get counts (next exercise)
  - Divide by row sums to get percent

Counts

cluster	sample1	sample2
0	404	231
1	539	64
2	39	511
3	193	193
4	49	194
5	169	64
6	130	11

Percent

cluster	sample1	sample2
0	0.265265923	0.182176656
1	0.353906763	0.050473186
2	0.025607354	0.402996845
3	0.126723572	0.152208202
4	0.032173342	0.152996845
5	0.110965200	0.050473186
6	0.085357846	0.008675079

- **For robust statistics, run differential analysis in edgeR on counts table**
  - You must have biological replicates
  - Result is differential *abundance*, rather than expression

Exercise 2.1: More visualizations and exploration

- Look at cluster abundance
- Data visualization

Moving beyond Seurat

- Seurat is a nice package with a lot of functionality
- But: there may be some analyses we want to do outside of Seurat
- Procedural things:
  - Extracting information from the Seurat object (saving gene lists, cluster identities, gene expression table, etc.)
  - Converting between gene IDs and gene symbols
  - Preparing to do pathway analysis
- Other statistical/informatics analysis:
  - Abundance of clusters across samples (analysis in edgeR)
  - Comparing clustering results to other algorithms, testing the robustness of a clustering result

The Seurat object

- Data stored under slots (@) and attributes ($ or [[ ]])
  - `sc_subset@slot`
  - `sc_subset$attribute`
- Most values are nested
  - `sc_subset@slot1$attr1@slot2`
  - Example: tSNE coordinates
- We can investigate these in R Studio
- Ultimately, you will get to a data frame, matrix, or vector

`sc_subset@reductions$tsne@cell.embeddings`

	tSNE_1	tSNE_2
s1_AAACCCAAGTATGGCG	12.95543	13.771535
s1_AAACCCATCGAATGCT	-17.48028	1.917677
s1_AAACGAAAGACCAACG	-15.37493	17.857038
s1_AAACGAAAGTGATAAC	-27.04702	-15.718186
s1_AAACGAACAAGCGAAC	-17.18280	17.521326
s1_AAACGAACAGAGGTTG	21.20406	13.253725

Some important pieces

- Metadata slot (data frame):
  - `sc_subset@meta.data`
  - Columns of interest include original ident (`$orig.ident`), clusters at a given resolution (`$RNA_snn_res.0.5`), and "assigned" clusters (`$seurat_clusters`)
  - You can add more columns if you want to (e.g., add cell type labels) by modifying this data frame
- Assays slot:
  - Normalized, log-scaled expression (sparse matrix):
    - `sc_subset@assays$RNA@data`
  - Normalized, z-scored expression (regular matrix):
    - `sc_subset@assays$RNA@scale.data`
  - Variable genes:
    - `sc_subset@assays$RNA@var.features`
- Reductions slot:
  - Principal components
    - `sc_subset@reductions$pca@cell.embeddings`
  - tSNE coordinates:
    - `sc_subset@reductions$tsne@cell.embeddings`

Exercise 2.2: Investigate Seurat Object in R

- Find a few important pieces of information in our Seurat object
- Intersect gene IDs with gene symbols
- Practice exporting a differential expression gene list (e.g., for pathway analysis)

Custom Analysis 1: Clustering robustness

cribioinfo@uic.edu

Review our steps up through clustering

1. Read in data
2. Cell filtering
3. Gene normalization/scaling
4. Feature selection
5. Clustering

cribioinfo@uic.edu

Review our steps up through clustering

1. Read in data
2. Cell filtering
3. Gene normalization/scaling
4. Feature selection
5. Clustering

Parameters chosen:

*changes to any of these **will** affect clustering results*

Min counts per cell (2000), max counts per cell (60,000), min genes per cell (1000), max MT per cell (10%)

Normalization algorithm (log-scale CPM), number of variable genes (6000), algorithm for variable genes (vst), scaling algorithm (z-score)

Dimensionality reduction algorithm (PCA), number of dimensions (20)

Choice of algorithm (Louvain), choice of distance metric (Euclidean), resolution (0.5)

What is our clustering result?

- Assignment of cells to a cluster – partitioning the cells into subsets
  - Cluster name is arbitrary
- Goal is to compare different clustering results to each other

Result 1

Cell	Cluster
Cell1	1
Cell2	1
Cell3	2
Cell4	2
Cell5	2
Cell6	2
Cell7	2
Cell8	3
Cell9	3
Cell10	3

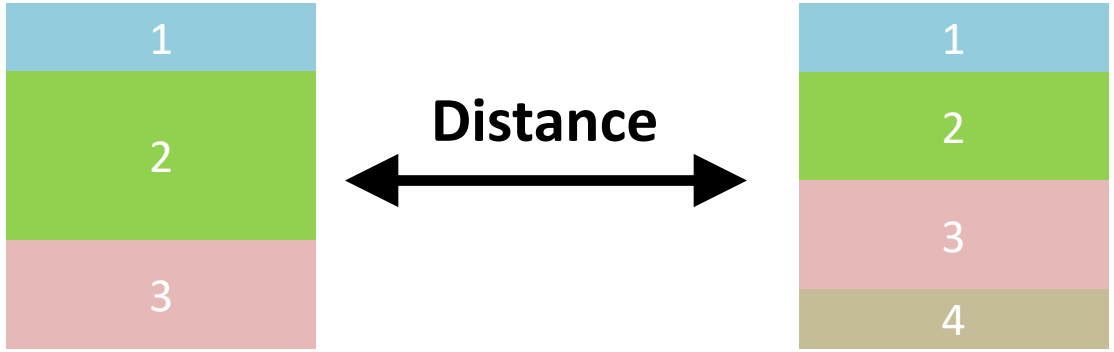
Result 2

Cell	Cluster
Cell1	1
Cell2	1
Cell3	2
Cell4	2
Cell5	2
Cell6	3
Cell7	3
Cell8	3
Cell9	4
Cell10	4

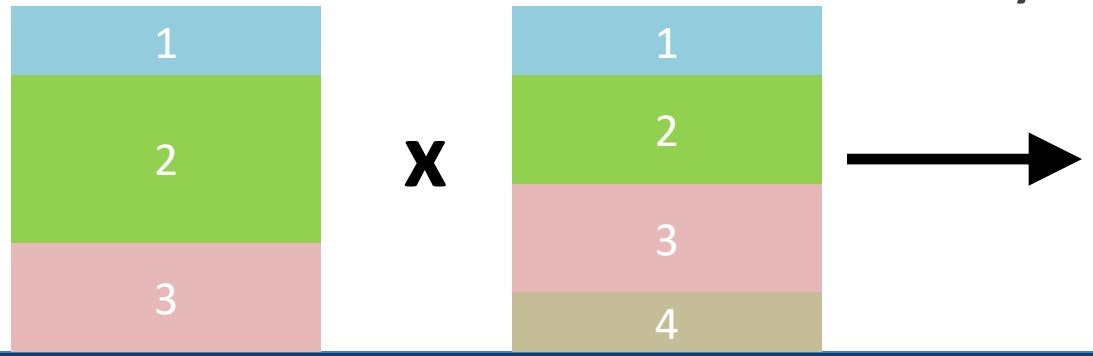
How to assess clustering robustness?

- Vary some parameters, rerun the other steps
  - Change resolution, number of PCs, number of variable features, etc.

1. Measure difference in clustering results



2. Compare cell sets in each cluster directly



Result 1 cluster 1 vs Result 2 cluster 1
Result 1 cluster 1 vs Result 2 cluster 2
Result 1 cluster 1 vs Result 2 cluster 3
. . .

cribioinfo@uic.edu

Metric 1: Overall distance between clustering results: **Rand index**

- Number of pairs of cells that are co-clustered **consistently**, divided by total number of pairs of cells
  - **Consistent**: in the same cluster in both results, or in different clusters in both results
- Formally:

$$\frac{(\#\ same) + (\#\ different)}{total\ pairs}$$

  - Same: pair is co-clustered in both results
  - Different: pair are in different clusters in both results
- Value is between 0 (completely different) and 1 (identical)

cribioinfo@uic.edu

Worked example

Result 1

Cell	Cluster
Cell1	1
Cell2	1
Cell3	2
Cell4	2
Cell5	2
Cell6	2
Cell7	2
Cell8	3
Cell9	3
Cell10	3

Result 2

Cell	Cluster
Cell1	1
Cell2	1
Cell3	2
Cell4	2
Cell5	2
Cell6	3
Cell7	3
Cell8	3
Cell9	4
Cell10	4

$$\frac{(\#\ same) + (\#\ different)}{total\ pairs}$$

Same = 6: 1&2; 3&4; 3&5; 4&5; 6&7; 9&10

Different = 29: 1&3…10; 2&3…10; 3&8…10; 4&8…10; 5&8…10; 6&9…10; 7&9…10

Total pairs = 45 (10 choose 2)

Rand index: 35/45 = 0.78

cribioinfo@uic.edu

Adjusted Rand Index

- Alternative option: adjusted rand index
  - You expect to see *some* same/different pairs by chance alone
  - We can adjust for that by subtracting the expected number of pairs from the calculation
  - Useful if the number of clusters, or sizes of the clusters, changes a lot between comparisons
  - May see negative values if the number of overlaps is less than expected by chance

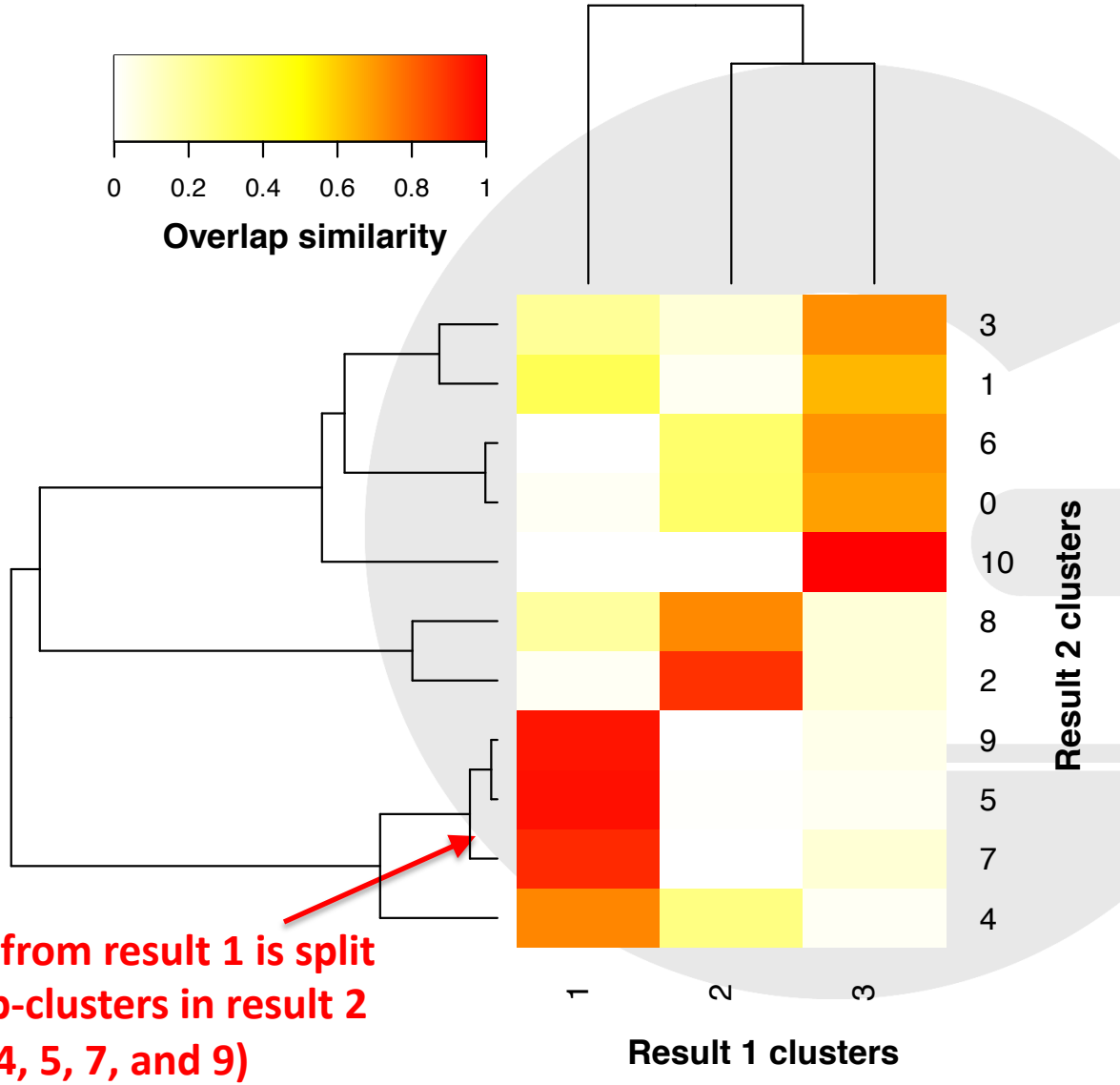
Computing the Rand Index

- Package in R: fossil
  - Palaeoecological and Palaeogeographical Analysis Tools
  - `rand.index()`
  - `adj.rand.index()`
- Value is between 0 (completely different) and 1 (identical)
  - *Similarity* metric
  - Do `1 - rand.index()` to get *distance*

cribioinfo@uic.edu

Metric 2. Compare cell sets in each cluster

- Determine correspondence between clusters in result 1 and result 2
  - Does one cluster get split into multiple sub-parts?
  - Are cells mixed and matched across clusters?
- Goal is a similarity matrix between individual clusters



cribioinfo@uic.edu

Overlap index or coefficient

- Intersection between cluster lists divided by the smaller of the two lists
- Formally:

$$\frac{intersection(cluster\ A, cluster\ B)}{min\ \_size(cluster\ A, cluster\ B)}$$

- Value is between 0 (completely different) and 1 (identical)
  - Similarity metric
- In R: we'll write our own function!
  - Set up a loop over all clusters from both results
  - Save it as a function for future convenient use

cribioinfo@uic.edu

Worked example

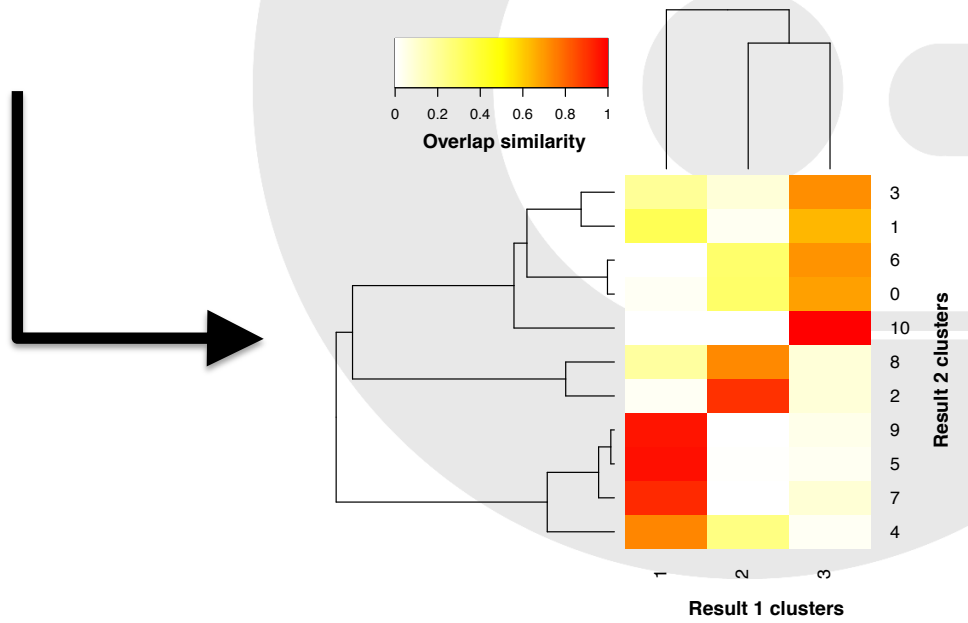
Result 1

Cell	Cluster
Cell1	1
Cell2	1
Cell3	2
Cell4	2
Cell5	2
Cell6	2
Cell7	2
Cell8	3
Cell9	3
Cell10	3

Result 2

Cell	Cluster
Cell1	1
Cell2	1
Cell3	2
Cell4	2
Cell5	2
Cell6	3
Cell7	3
Cell8	3
Cell9	4
Cell10	4

Result 1	Result 2	Overlap Index
Cluster1	Cluster1	2/2 = 1
Cluster1	Cluster2	0/2 = 0
Cluster2	Cluster2	3/3 = 1
Cluster2	Cluster3	2/3 = 0.67
etc...		



cribioinfo@uic.edu

How to think about our options

- **Rand index** is useful to get a high-level overview between a large number of different clustering results
  - Example: you've generated 20 different clustering results with different parameter settings.
    - Which ones are most similar?
    - What settings are the results most sensitive to?
- **Overlap index** is useful to get a more detailed view on how two results compare
  - Which cluster from result 1 corresponds to cluster(s) from result 2?
  - Is one cluster getting split up? Are the clustering results very different?

Exercise 2.3: Cluster comparisons

- Run clustering a couple different times in Seurat with different resolutions
- Compute Rand Indices between results
- Compute Overlap Indices between clusters from two results
  - Plot as heatmap

Custom Analysis 2: Putative cell type identification

cribioinfo@uic.edu

How do I define a cell type?

- **Phenotype/cell marker**
  - Expression of CD69 – T cell
- How specific do I want to be?
  - T cell vs {T reg vs T helper vs Naïve T vs T responder vs T mem etc.}
  - CellMarker database (http://biocc.hrbmu.edu.cn/CellMarker/index.jsp)
- Advantages:
  - Easy to understand assignment
- Difficulties:
  - Subjective: different people may have different ideas about what defines a cell type
  - Specificity: many markers may be specific for multiple cell types
  - Relies on protein expression, but we're measuring gene expression
    - Use cluster averages to deal with sparsity
    - Consider using cell hashing option in libraries

How do I define a cell type?

- **Transcriptome similarity**
  - Correlate expression per cell or per cluster with “gold standard” transcriptome for a cell type
- Source of “gold standard”?
  - One option: Human Primary Cell Atlas (HPCA)
  - Other scRNA-seq or bulk RNA-seq data sets
- Advantages:
  - More robust than looking at a handful of genes
  - Fairer apples-to-apples comparison of transcriptomes
- Difficulties:
  - Confirming purity and accuracy of cell type labels in “gold standard”
  - Availability of reference data sets for your species and tissue
  - Level of detail for sub-types (e.g., T reg vs T helper)

Data to use

- Use log-scaled normalized expression (`sc_subset@assays$RNA@data`)
  - We DO care about overall abundance
    - @scale.data is z-scored: good for comparing across clusters, but less useful when cell type depends on higher expression for key genes
  - We don't know if many clusters will share the same cell type
    - Differentially expressed genes may omit genes that are consistently expressed (not differential), but still informative with respect to cell type
- Averaging expression per cluster may be a useful first step
  - Averages out sparsity/variability from single cells
  - Closer to what a bulk RNA-seq profile would look like
  - Computationally faster to analyze clusters than individual cells

Exercises today

- We will do our analysis without any extra packages
  - Look at marker plots
  - Compute correlations of cell types vs transcriptome data sets
- Many R packages are available also
  - SingleR
  - SCSA
- Statistics are not very complicated – main hurdle is finding the right references for comparison (markers or transcriptome datasets)

Exercise 2.4: Putative cell type

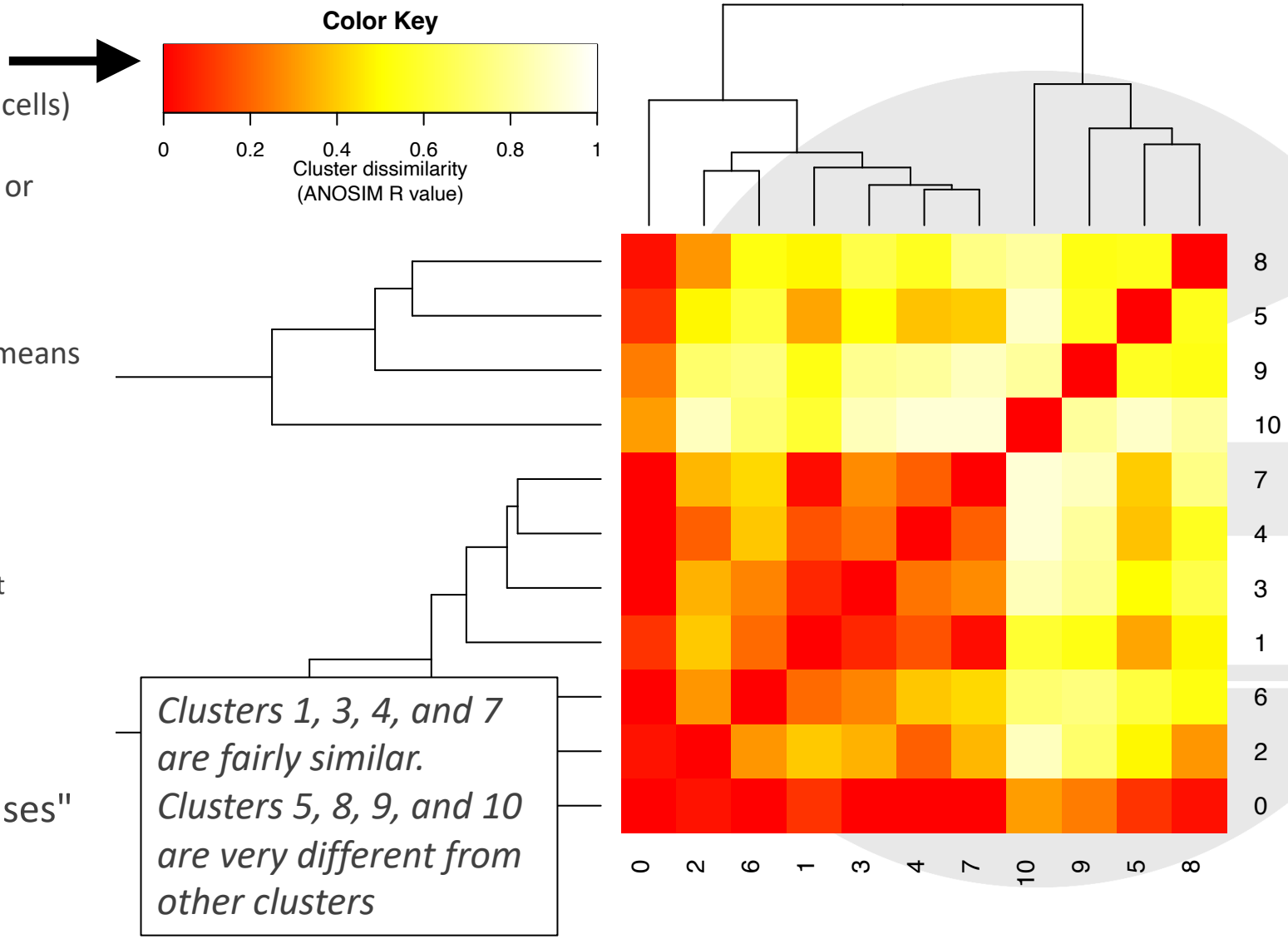
- Look at expression of marker genes across clusters
- Correlate cluster expression levels with cell types from HPCA (Take Home?)

Other things to consider

- There are a lot of directions you can go, depending on what you want to learn and measure
- Having the analysis in R gives you the ability to pull in a lot of other functionality
  - Need to be able to manipulate the R objects
- Being familiar with statistical methods in other areas can give you other ideas to test
- Pathway analysis: export cluster lists to databases like IPA, GO

Example: diversity statistics (from metagenomics)

- Dissimilarity analysis (Beta diversity)
  - Measure of overall dissimilarity between entities (cells) based on overall composition (gene expression)
  - Which sub-populations (clusters) are more similar or different?
  - Quantify with ANOSIM R statistic (degree of overlap/separation between groups)
  - Alternatively compute distances between cluster means ("signatures")
- Alpha diversity
  - Measure of diversity with an entity (cell)
  - How widely distributed is expression in a cell?
    - Are a lot of genes active and evenly expressed, or is it focused on a narrow subset?
    - Does this change across sub-populations?
  - Shannon entropy
- `vegan` package in R
- Example exercises included in "Extra Exercises" section



Clusters 1, 3, 4, and 7 are fairly similar. Clusters 5, 8, 9, and 10 are very different from other clusters

cribioinfo@uic.edu

Systems analysis

- Identify genes of interest, e.g.:
  - ROC > 0.75 for each cluster
  - log2FC > 1 & adj p-value < 0.01 for each cluster
  - Etc…
- Intersect with external gene lists:
  - Pathways
    - Biological functions of cell types
  - Upstream regulators
    - TFs that may be controlling cell differentiation
- *No exercise today, but this fits directly into our pathway analysis workshop day*

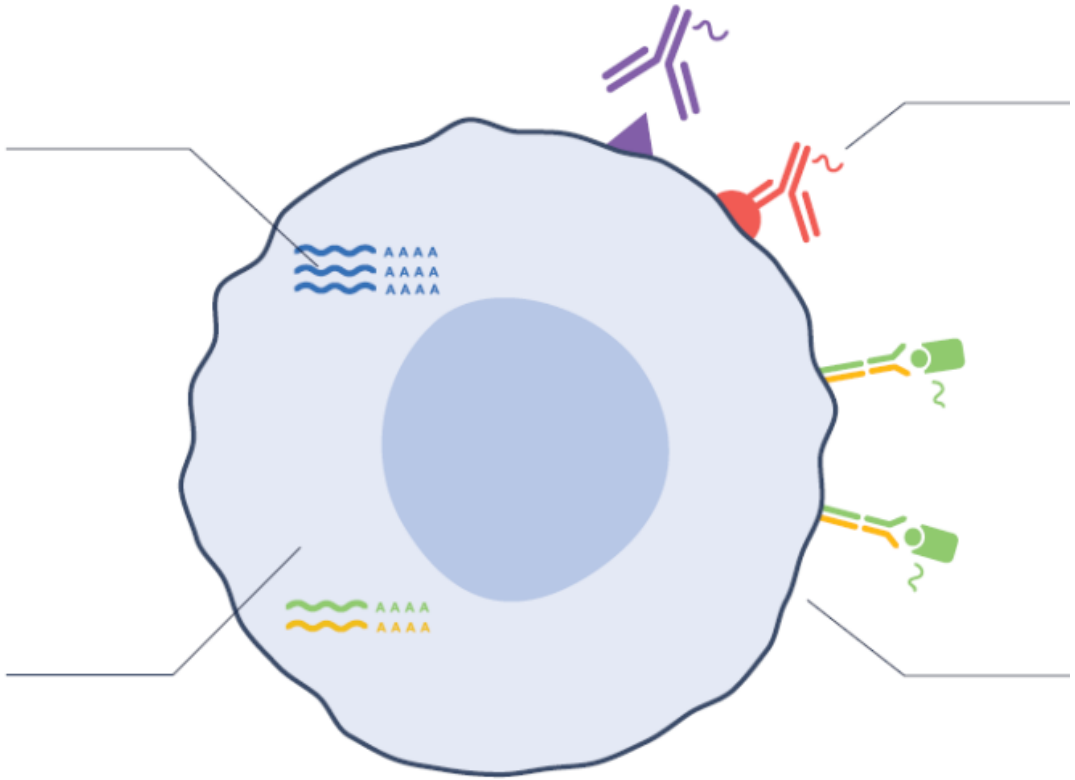
BREAK

Please complete our workshop survey

http://go.uic.edu/RICWorkshopSurvey

Supplemental single-cell libraries

Gene Expression (GEX)
"Standard" single-cell library
Provides information on basic gene expression in each cell.



Feature Barcoding (FBC)
DNA tagged antibodies can provide information on expressed, cell surface proteins.

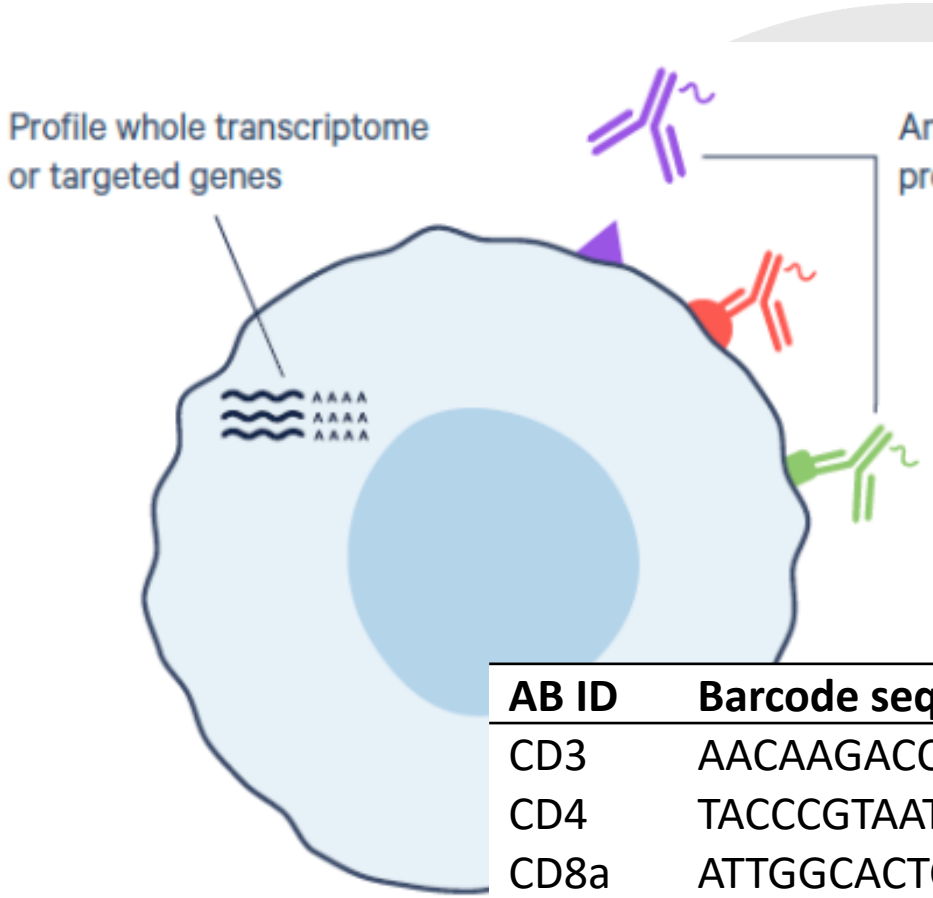
Immune Profiling (VDJ)
Details of clonotype details of expressed immune receptors, e.g. TCR genes.

Feature Barcoding (FBC)
DNA tagged antigens can provide information on antigen specificity of captured T and B cell.

cribioinfo@uic.edu

Feature Barcoding (FBC) libraries

- Cells are incubated with antibodies with an attached DNA fragment (antibody-DNA tag)
  - Each antibody as a specific sequence in the DNA tag (barcode)
- Normal single-cell capture is performed.
- During library creation, DNA tags are amplified and have cell barcode added like mRNAs from the cell.
- Antibodies “cocktails” can be used to identify cell surface proteins on each cell
- Can also use FBC to “cell-hash” and multiplex multiple biological samples into a single capture
  - Use same antibody with different DNA tags for each sample.

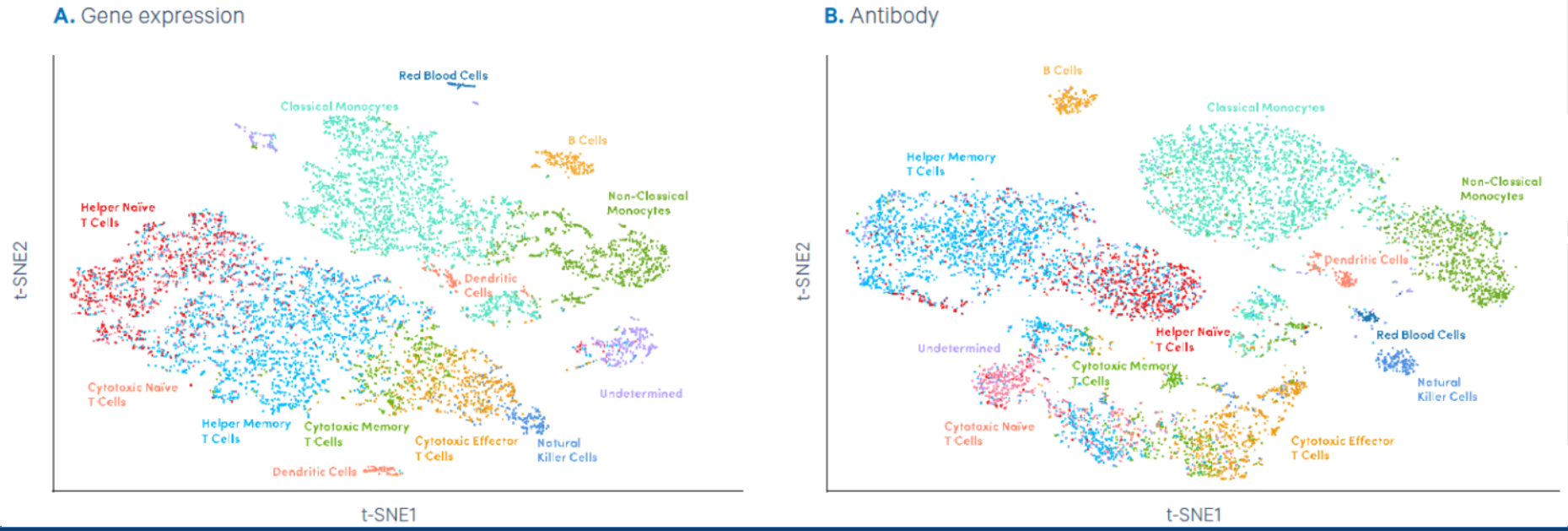


AB ID	Barcode sequence
CD3	AACAAGACCCTTGAG
CD4	TACCCGTAATAGCGT
CD8a	ATTGGCACTCAGATG
CD14	GAAAGTCAAAGCACT
CD15	ACGAATCAATCTGTG

cribioinfo@uic.edu

Analyzing FBC data

- Can process like typical scRNA-seq project and then use FBC data as additional data to visualize or explore clusters/cells detected in GEX data.
- Can combine GEX and FBC data for clustering (use CellRanger for clustering).



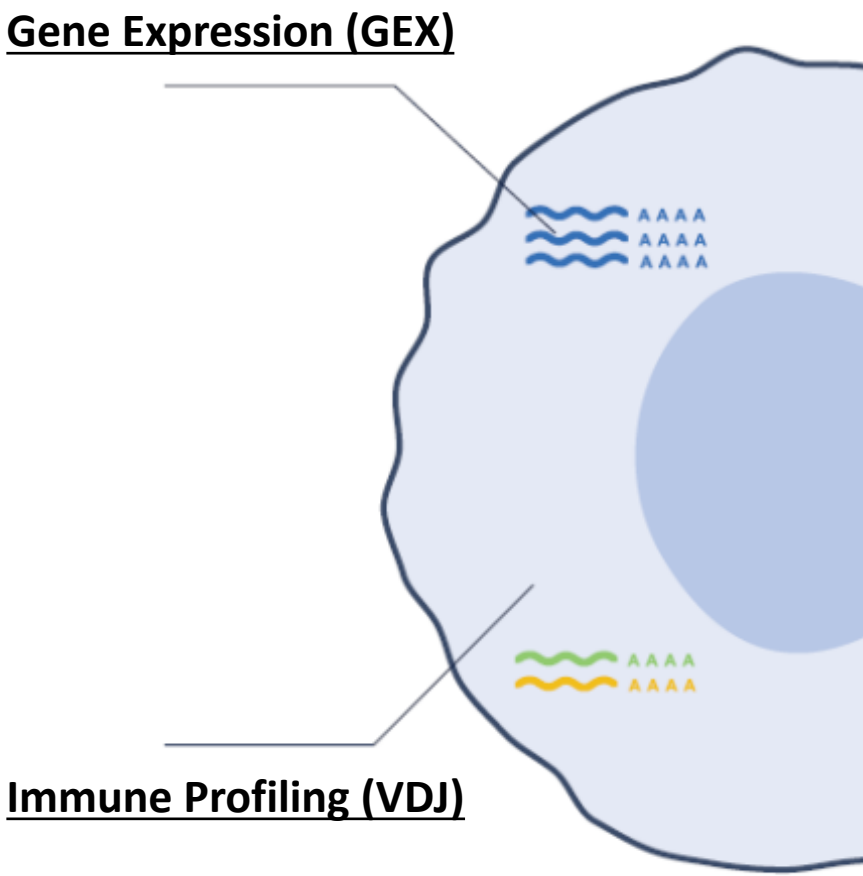
cribioinfo@uic.edu

Exercise 2.5: Incorporating Feature Barcoding data

- Load FBC data into Seurat.
- Generate basic visualizations and stats from antibody data.

Immune Profiling, a.k.a. V(D)J profiling

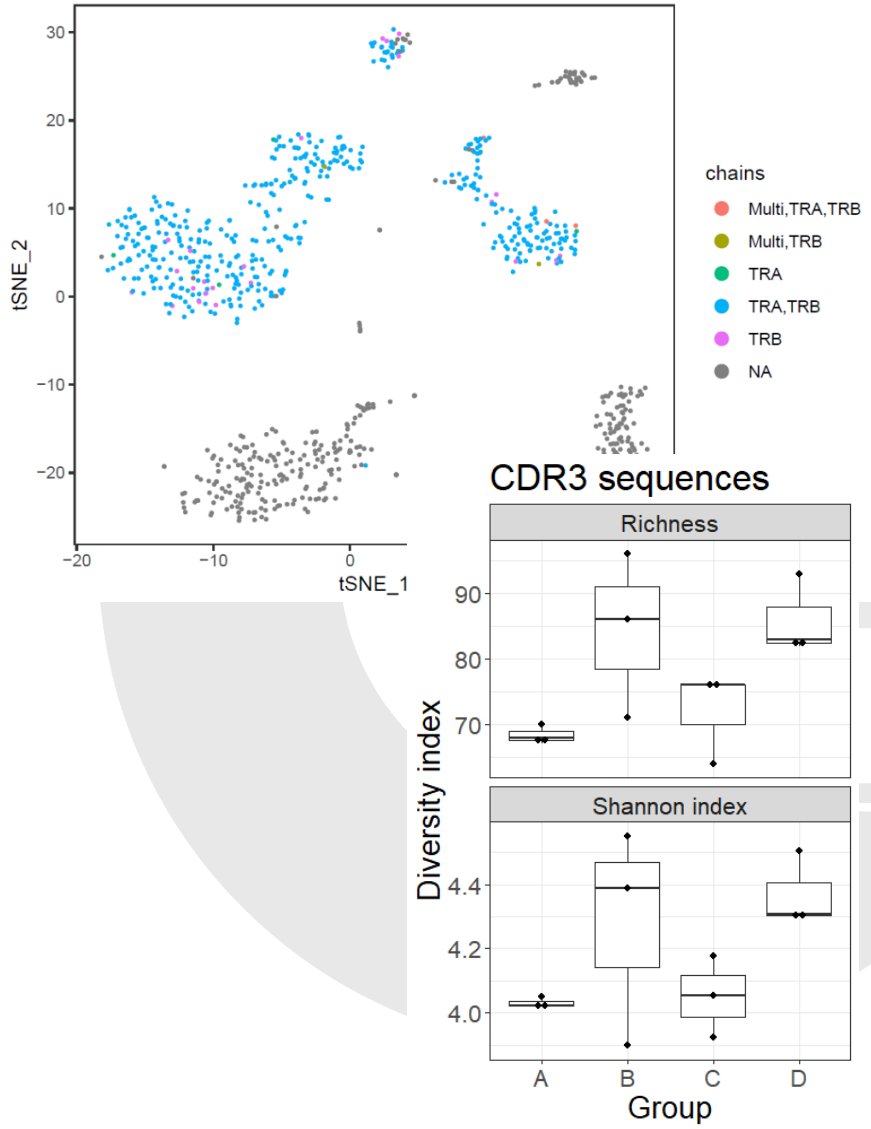
- Obtain full-length sequences of expressed TCR/BCR genes in immune cells.
- Akin to FBC, Immune profiling will happen in same capture thus will have same cell barcode.
- Sequence data are assembled to produce full-length TCR and BCR genes
  - Additional annotation of chain (TRA vs. TRB), V, D, J, and C regions
  - Reports CDR3 (nucleotide and amino acid) sequences for each contig



cribioinfo@uic.edu

Analyzing Immune Profiling Data

- Can visualize TRA/TRB presence on tSNE plot
  - Tabulate percent of cells in each cluster with TCR data.
- Compute diversity of V(D)J or CDR3 in samples or clusters.
  - Does diversity of TCR sequences increase/decrease with experimental groups.
  - May need to subsample/rarefy to account for samples with more T-cells.
- Look for “public” clones (CDR3 sequences detected in more than one sample)
- Other analyses… (What is your hypothesis? What are you looking for in the data?)
- ***Note***: CDR3 data can be very sparse (most sequences will be observed in only one cell) Could use phylogenetic or sequence similarity methods to probe.



cribioinfo@uic.edu

Exercise 2.6: Incorporating Immune Profiling, a.k.a. V(D)J, data

- Load V(D)J data into R.
- Generate basic visualizations and stats from V(D)J data.

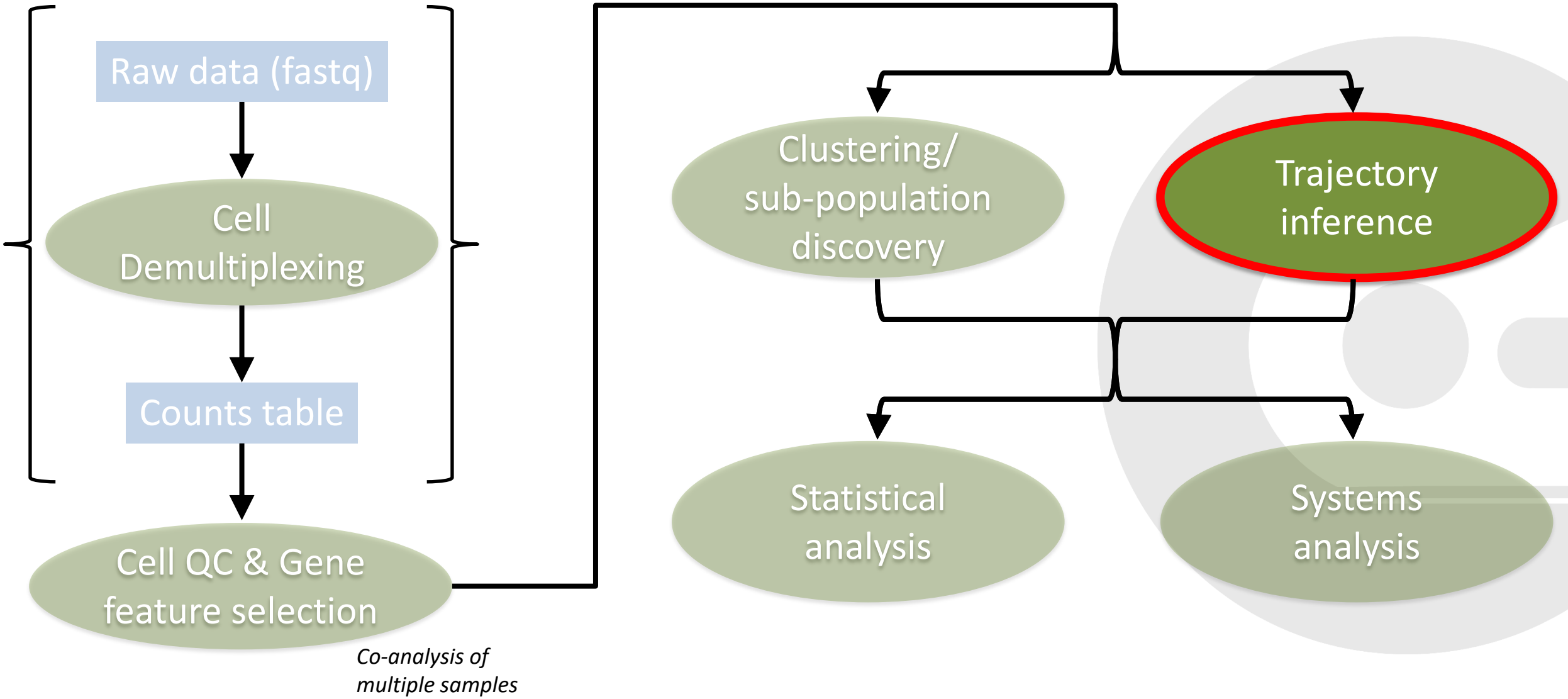
Additional items in "Extra (Take Home) Exercises"

- *Loading V(D)J data into a Seurat object*
- *Diversity analysis of V(D)J data.*

Trajectory inference

Pseudotime, RNA Velocity, and CytoTrace

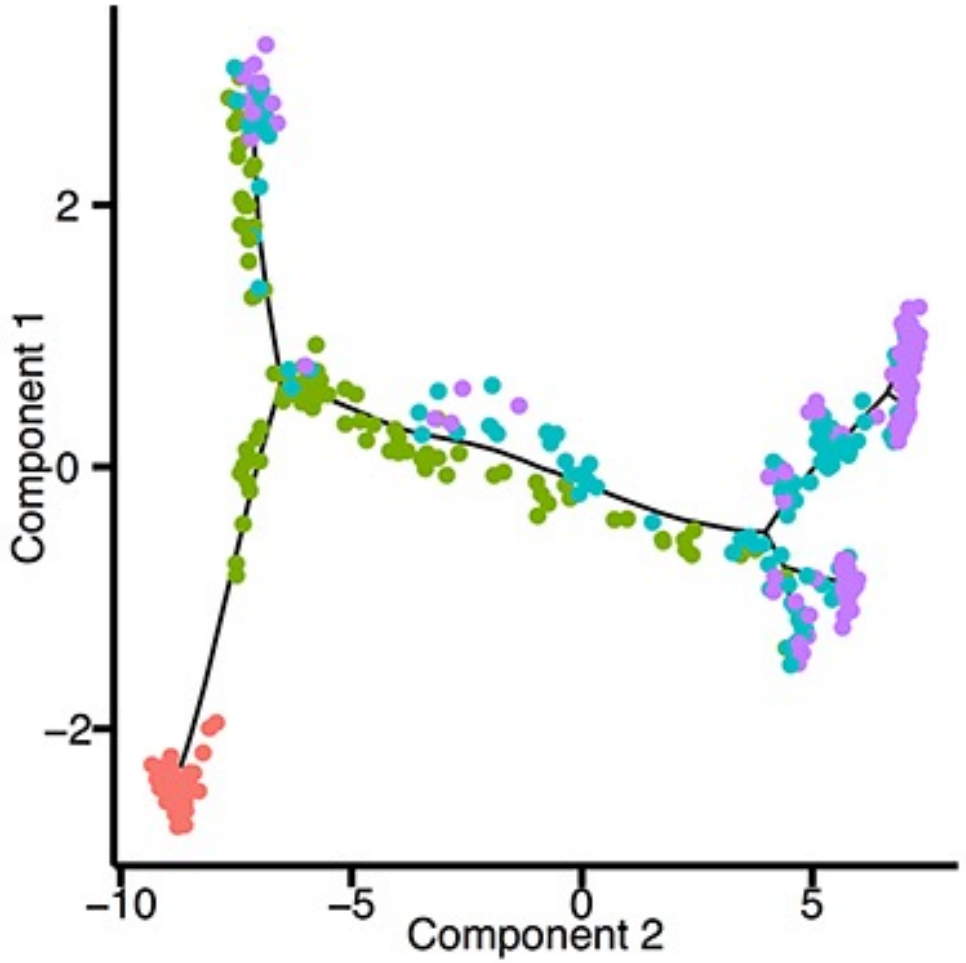
Trajectory inference: pseudotime and RNA velocity



cribioinfo@uic.edu

Motivation

- Individual cells represent a continuum of states, e.g.
  - Response to a drug
  - Developmental stage
- Complementary idea to the clustering approach
  - Clustering = group cells into discrete types
  - Trajectory = place cells along a continuous path
- Desired inference:
  - **Path:** a connection (may be branched or looping) between cells representing some process
  - **Ordering:** an indication of which direction the cells are going
- **We often pursue trajectory analysis *after* clustering**
  - May make the most sense on subsets of cells, such as those that are part of the sample developmental type (e.g., all neuronal cells)



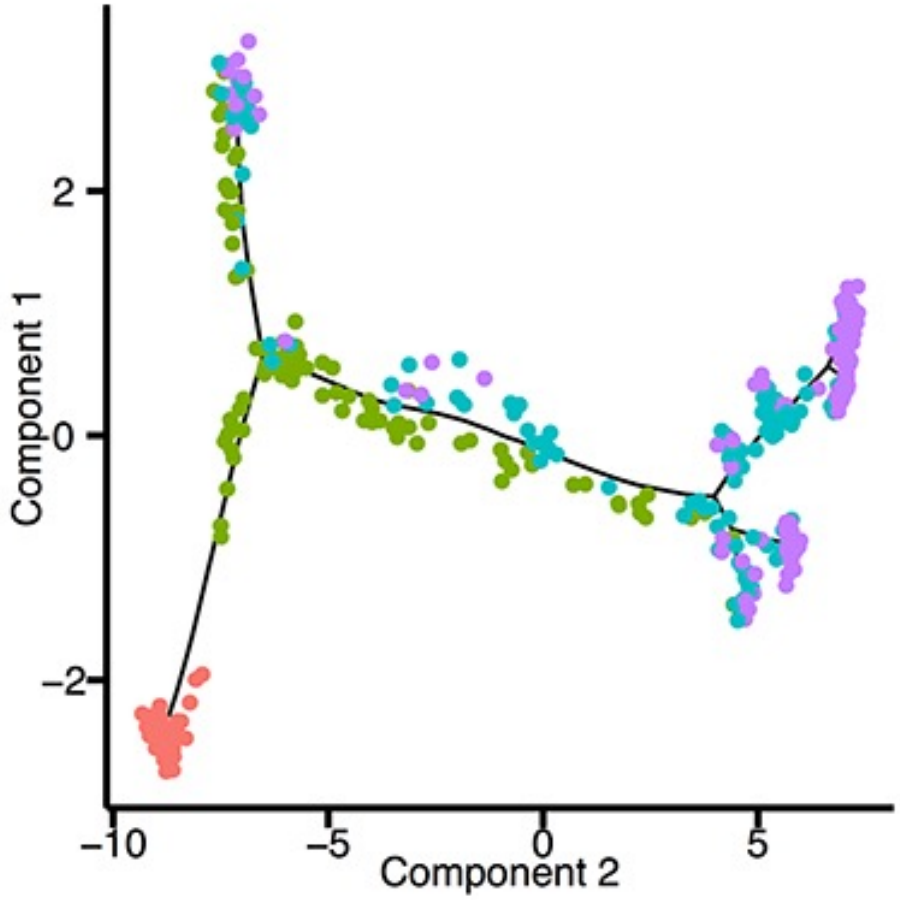
cribioinfo@uic.edu

Overview of methods

Method	Path	Ordering	Caveats/notes
Pseudotime	Yes	Kind of	• Uses same data as clustering, results will be correlated • Results be sensitive to nonlinear ordination • “Time” variable doesn’t distinguish between branches, direction is set arbitrarily
RNA velocity	Yes, qualitative	Yes	• Independent of clustering data • Requires extensive re-processing of data • May be inaccurate for 10X or other 3’ biased data
CytoTRACE	No	Yes	• Independent of clustering data • Model is simple and easy to run • Validation is in from developmental processes. May not apply to, e.g., a drug response.

Pseudotime

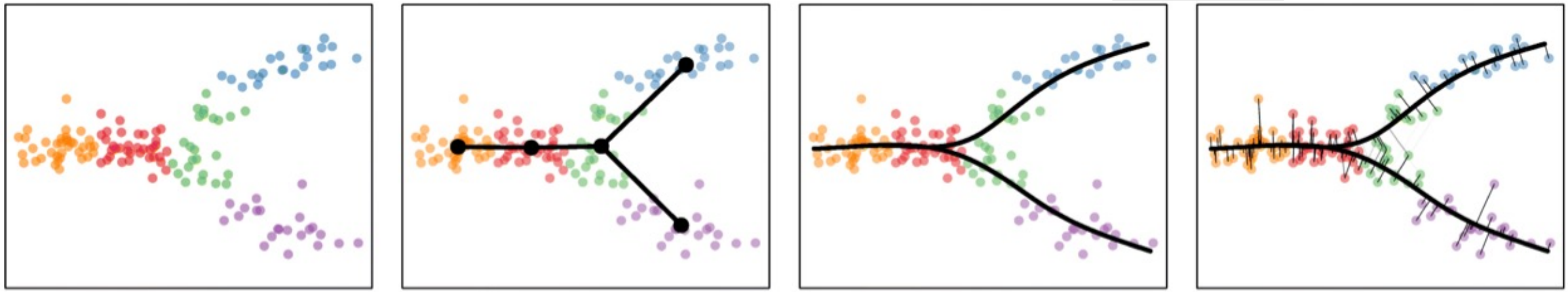
- Infer a path that connects cells
  - Based on transcriptome similarity
  - Often relies on non-linear dimensionality reduction
- Results from pseudotime:
  - "Time" for each cell
    - Order is arbitrary, but quantifies progress along a path
    - May have branched trajectories
- *Take-home exercises: Monocle2*



cribioinfo@uic.edu

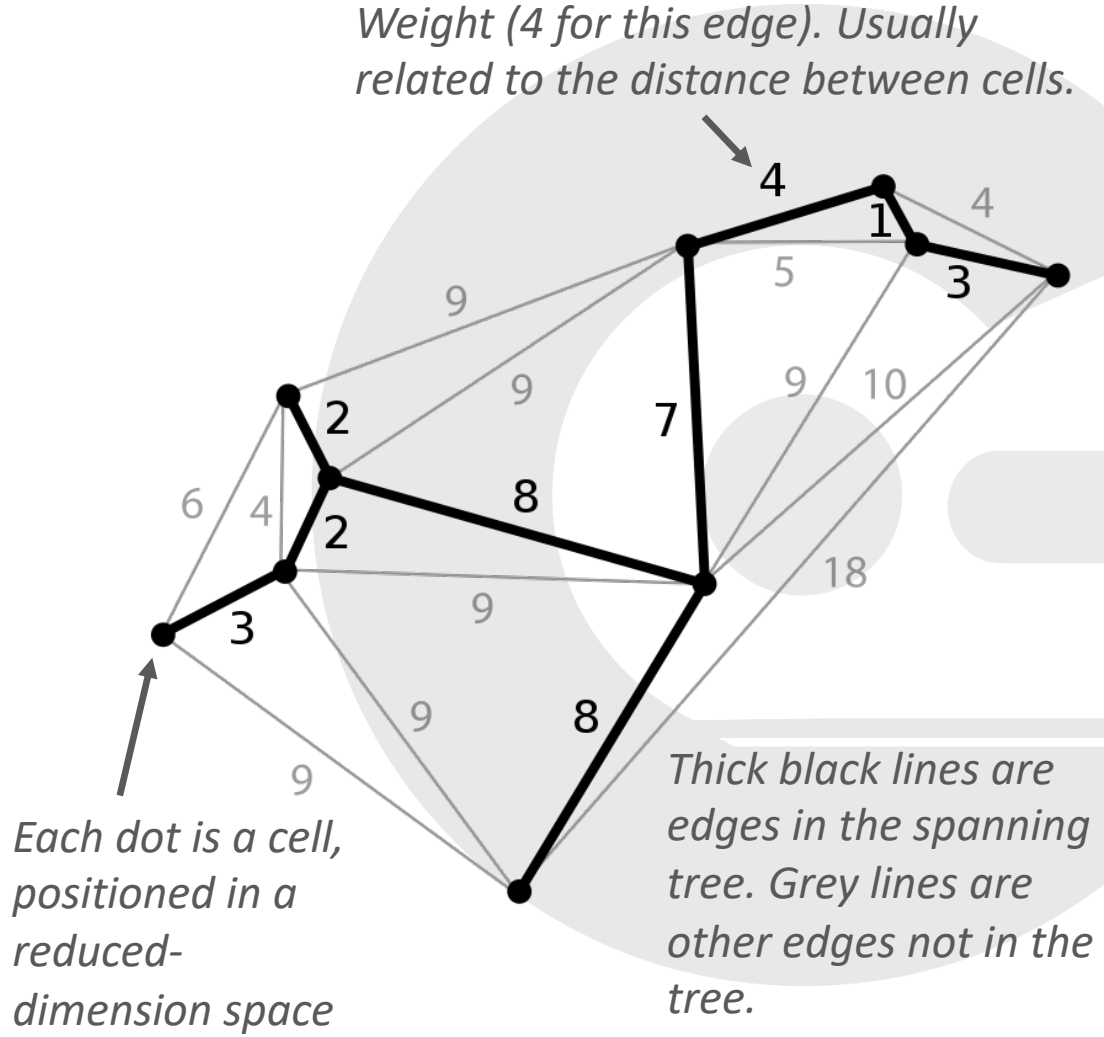
General analysis strategy

- Cell QC, feature selection (as before)
- Dimensionality reduction (linear or non-linear)
- Trajectory inference through reduced dimensions
  - Many algorithms used by different methods
  - Most Common is **Minimum Spanning Tree**
  - Some use prior information, others are *de novo*



Minimum spanning tree

- Cells are represented as points in reduced dimension space
  - E.g., 2 dimensions from tSNE or UMAP, or DDRTree (Monocle)
- Need all dots connected to each other (edges), without cycles (**spanning tree**)
- Minimize total weight of edges (**minimum**)
- Pseudotime comes from the longest connected path
  - No inherent direction
  - May not be unique
- *Updated version in Monocle3 includes the possibility of cycles*



cribioinfo@uic.edu

Practical considerations

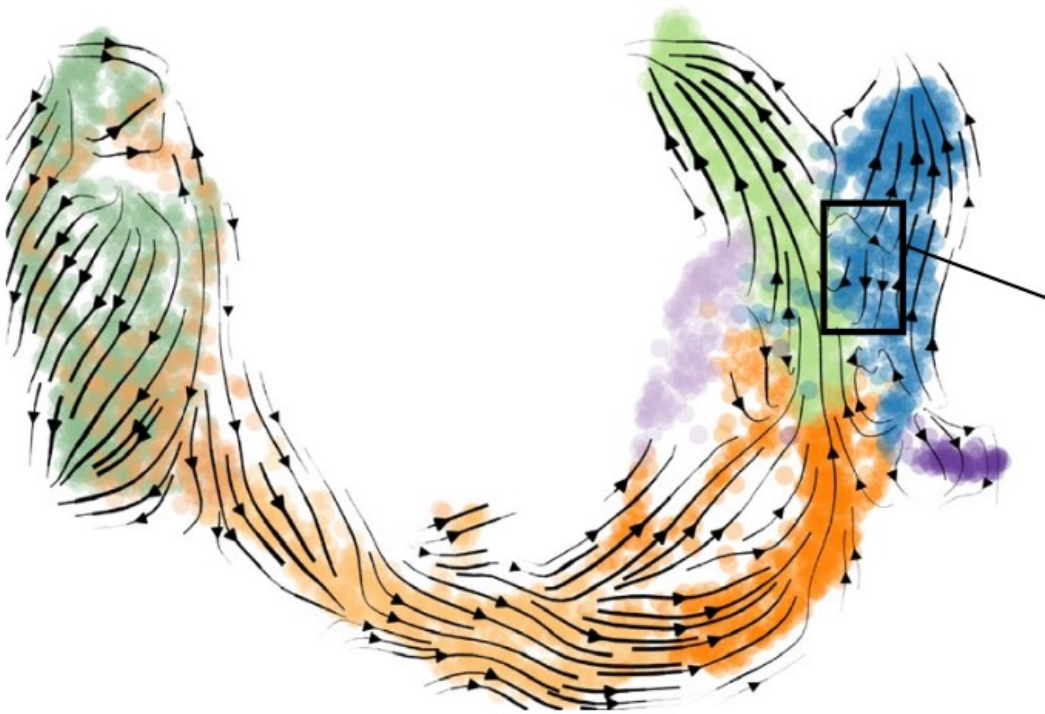
- Requires pre-processing of data like what we did for clustering
  - Same issues of robustness apply
- Often relies more heavily on non-linear dimensionality reduction
  - Adds an extra set of parameters/choices
- Optimal trajectory may not be unique for a given data set
- No inference of *directionality*, just connectivity
  - You would infer directionality as part of your interpretation, e.g., “cells on one side are transcriptionally consistent with a precursor state, so I infer that this is the starting point”

R packages

- Monocle
  - Original package: Monocle (2014)
  - **Updated package: Monocle2 (2017)**
    - **Example take-home exercise**
  - Newest package: Monocle3 (2019)
    - Adds features to detect cycles, multiple trajectories
    - However, very difficult to install… (lots of dependencies)
- TSCAN (2016)
- Waterfall (2015)
- Wanderlust/Wishbone (2014/2016)
- *Many others (>50)*

RNA velocity

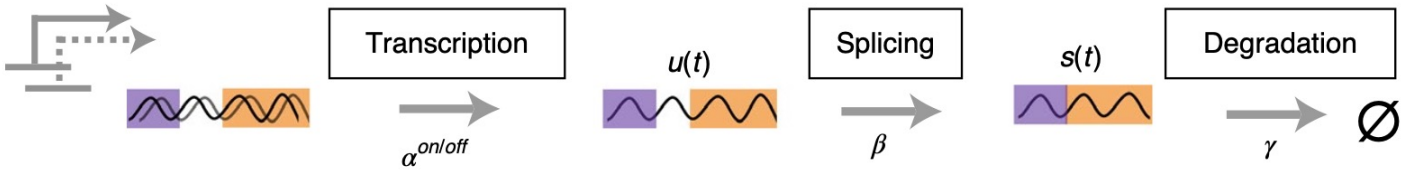
- RNA velocity: change in mRNA abundance
- Use estimates of gene splicing dynamics to infer trajectory of a cell
  - Estimate spliced vs unspliced transcripts: **need to run new quantification from BAM file**
  - Compute RNA velocity per cell
- Visualize/propagate trajectories
  - Estimate cell-to-cell conversion probabilities
  - Infer *directional* trajectories throughout the population
  - Project velocities into non-linear embeddings



Figures from Bergen *et al.*, *Nature Biotech., 38, 1408-1414, 2020* unless otherwise noted

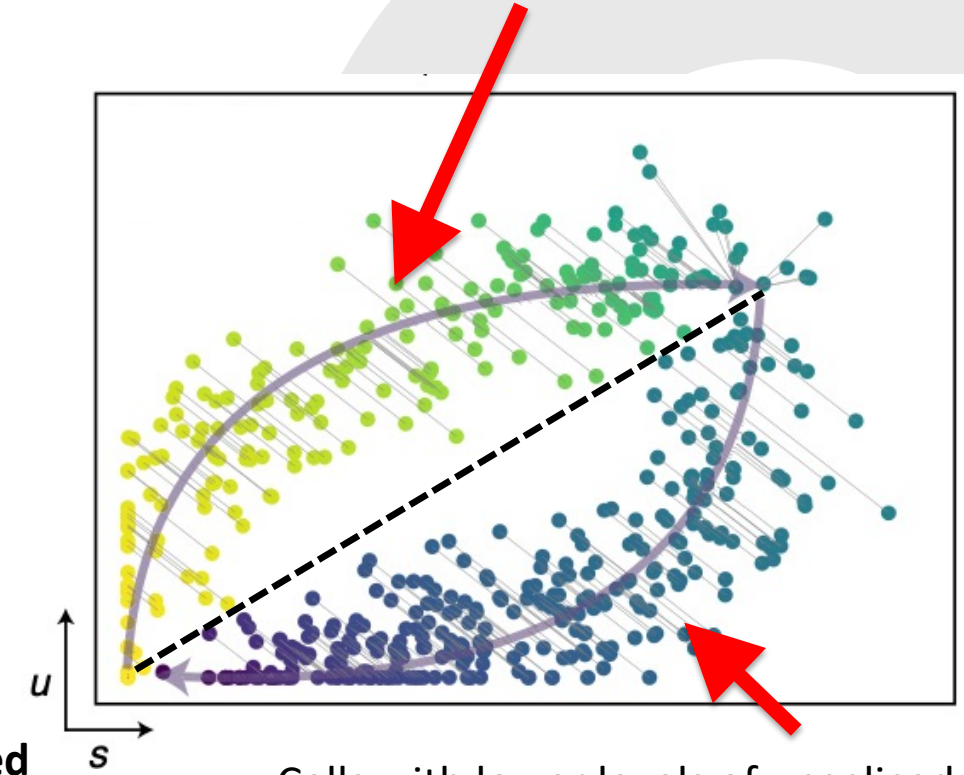
RNA velocity: Splicing differences

- Measure relative levels of exonic to intronic reads for each gene to estimate unspliced (U) vs spliced (S)
  - Compute from read alignments in BAM file



- Compare these ratios across cells
  - Higher unspliced reads = upregulated = positive velocity

Cells with higher levels of unspliced reads for this gene: positive velocity



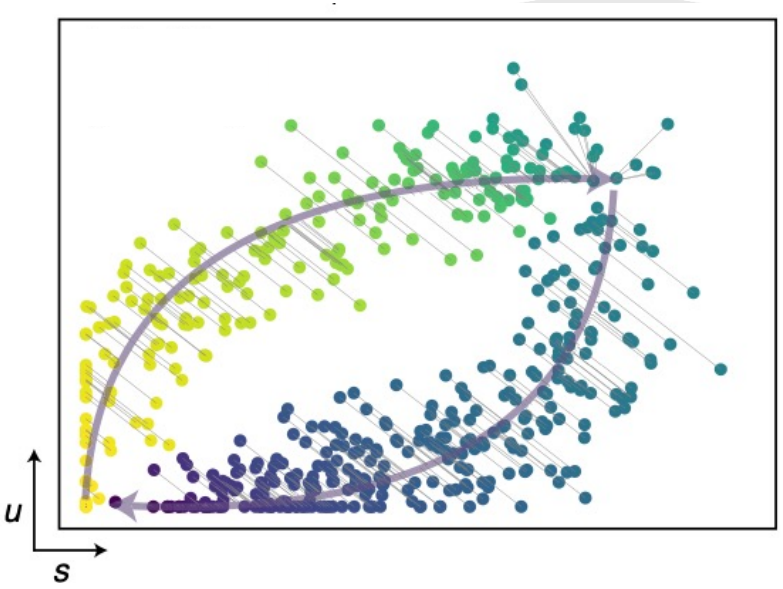
U = unspliced
S = spliced

Cells with lower levels of unspliced reads for this gene: negative velocity

cribioinfo@uic.edu

Velocity calculations

- Velocity for one gene: calculation of U/S ratio, kinetic models
  - Result: one-dimensional value for that gene; can be positive or negative based on activation or repression
- Velocity for one cell:
  - N-dimensional vector of velocities for all genes being considered
  - E.g., 6000 dimensions, if analysis is based on 6000 top genes
- This gives a *direction* for the cell (in 6000 dimensional space)



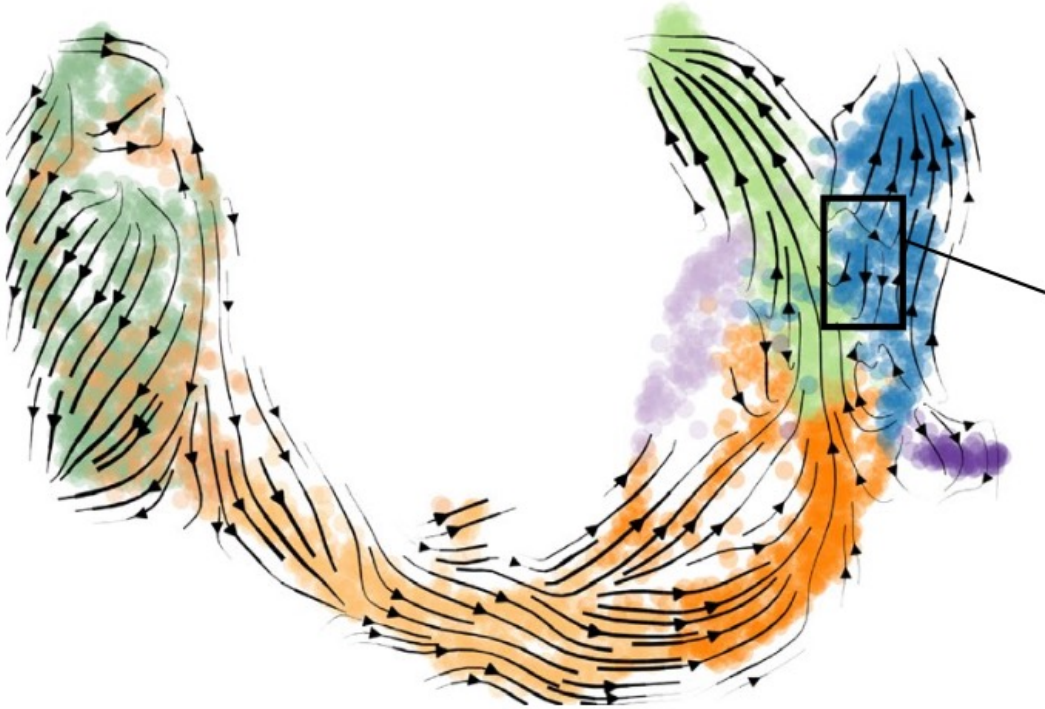
Velocity per cell for THIS gene, + 5999 more…

cribioinfo@uic.edu

Cell trajectories from velocities

- The *trajectory* of a cell depends on its *direction* AND its *position*
  - *Position* is based on the expression level of the genes (also 6000 dimensions)
  - Measure *transition probability* of cell A to cell B by comparing the difference in positions (cell A – cell B) to the velocities of cell A
- Visualization in 2D embedded space (tSNE or UMAP)
  - Non-linear projections require approximations of velocity vectors
  - Try to match projections of velocities based on computed transition probabilities and velocities, and embedded positions. This is another estimation step.

Visualization of cells AND velocities in 2D space

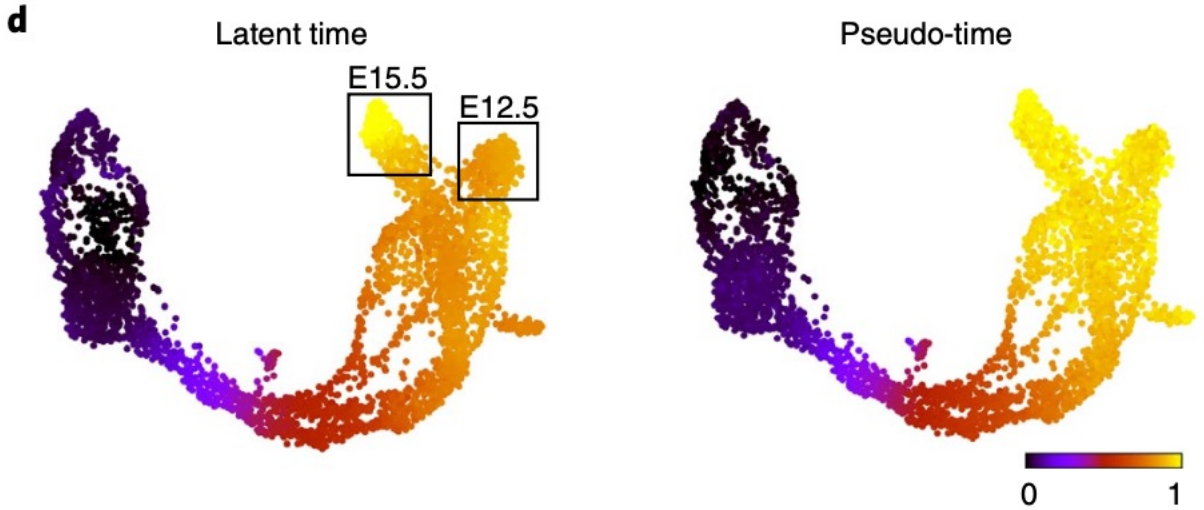


cribioinfo@uic.edu

Software packages for RNA velocity

- Velocyto (http://velocyto.org/velocyto.py/) (Manno *et al.*, Nature, 2018)
  - Python (maybe R? hasn't been updated recently)
  - Original RNA velocity paper
- scVelo (https://scvelo.readthedocs.io) (Bergen *et al.*, Nature Biotech, 2020)
  - Python, built by scanpy developers (similar to Seurat, but in python)
  - Includes more advanced modeling parameters
    - Dynamical modeling/latent time
    - Pseudotime analysis option also

Latent time vs pseudotime from scVelo



cribioinfo@uic.edu

Basic analysis steps: part 1

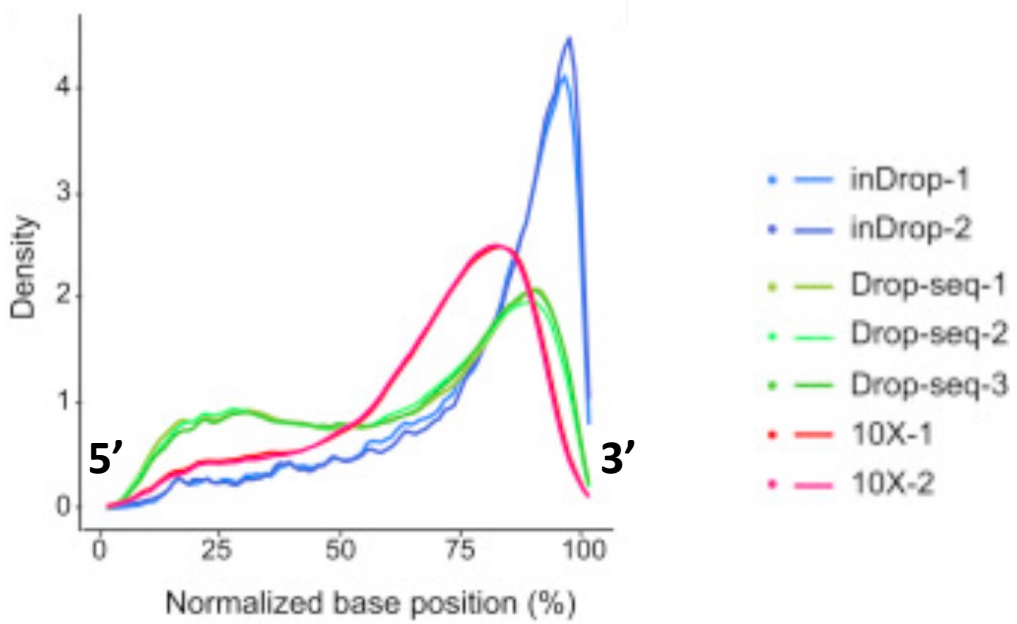
1. Quantify spliced/unspliced with velocyto
   - `veloctyo run10x [cellranger folder] [gtf file]`
   - (Use same for velocyto and scVelo)
   - Command line; takes awhile: run on compute node (~6 hrs, 20GB RAM for 7k cells)
   - Produces a *loom* file: specific HDF5 structured data file for genomics data
     - Loom specifications: https://linnarssonlab.org/loompy/format/index.html
     - Velocyto documentation: http://velocyto.org/velocyto.py/

Basic analysis steps: part 2

2. Run processing steps in velocyto or scVelo (in python):
   - Read in loom file(s)
   - Import cells of interest and tSNE or UMAP reduction from Seurat
   - Normalization, feature selection, PCA
   - Velocity calculation
   - Transition probabilities
   - Visualizations
   - Other steps per package workflow…

Caveats with RNA velocity

- Most scRNA-seq are 3’ captures, and biased towards 3’ end: splicing information may be incomplete

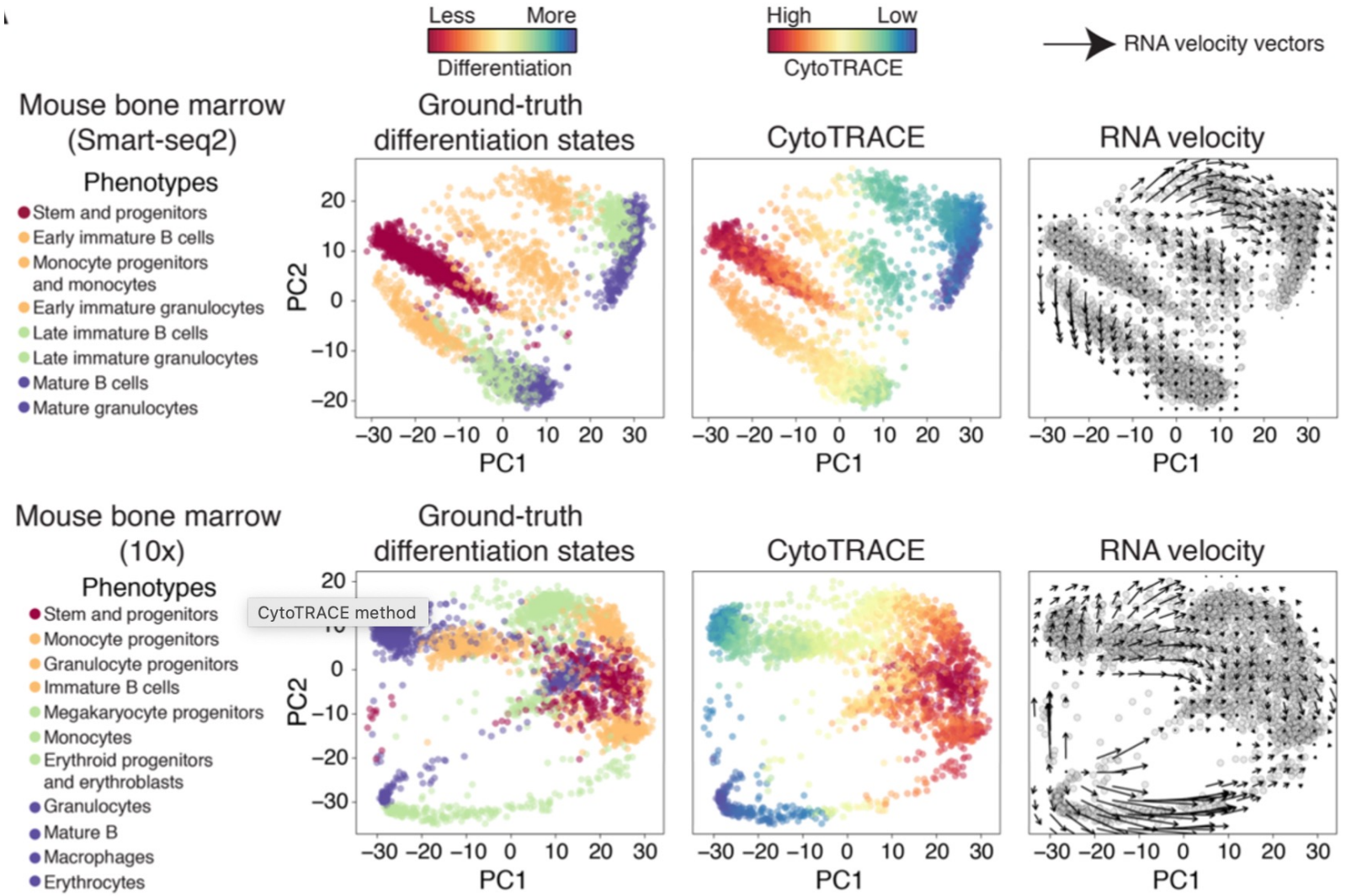


Zhang *et al.*, **Comparative Analysis of Droplet-Based Ultra-High-Throughput Single-Cell RNA-Seq Systems**, Molecular Cell, 73, P130-142, 2019.

- Need to select genes with enough data for reasonable estimates
  - Analysis steps include similar selection of top genes, PCA reduction and top PC selection, nearest neighbors, and other parameters as we did in clustering analysis
  - Many model-specific assumptions (time invariance, gene invariance, steady state, etc.)
  - Results may be sensitive to these choices

Caveats with RNA velocity: counterexample

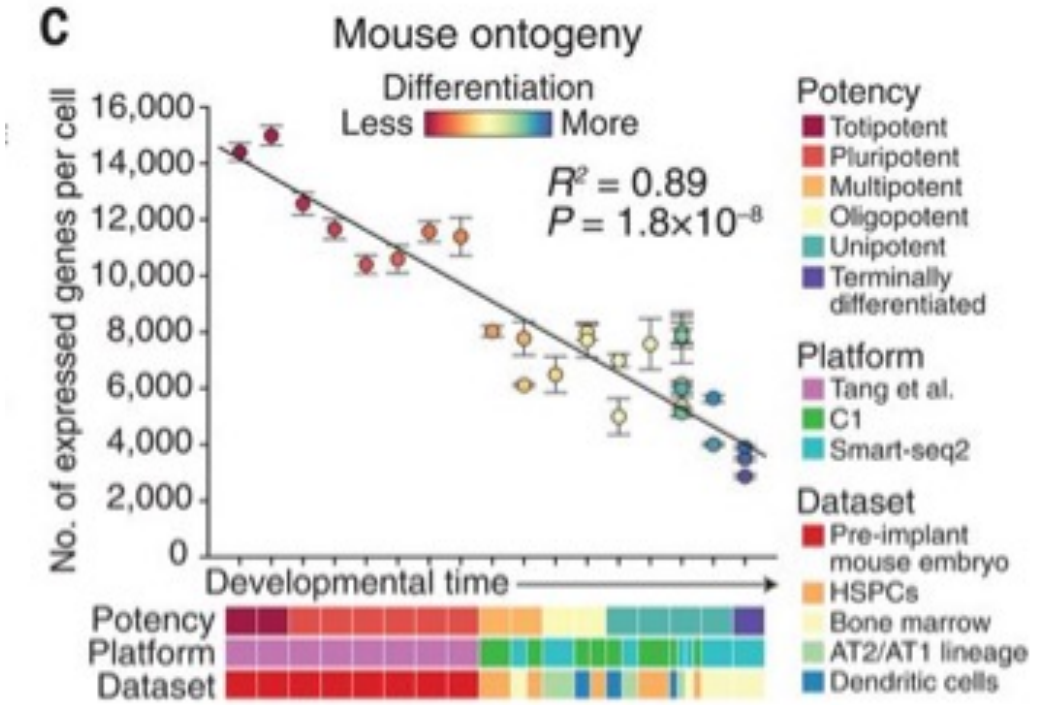
- RNA velocity inference is *backwards* in 10X data, correct in Smart-seq2 data (full transcript)
  - Analysis from CytoTrace paper supplement
  - Data sets in lymphopoiesis

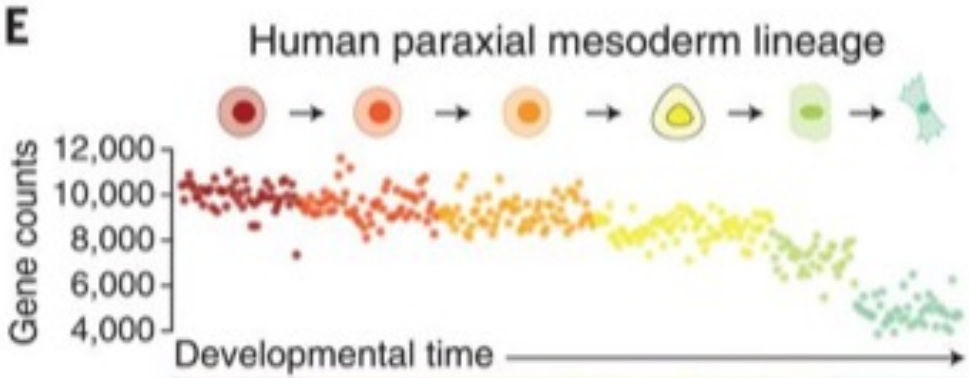


cribioinfo@uic.edu

CytoTRACE

- Observation that number of “detectably expressed genes” per cell correlates to status along lineage
  - Later lineage: fewer genes expressed
- Method: quantify diversity of gene expression per cell, use that to quantify a “time”
  - Result is a “time” value per cell
  - Time is ordered: lower values = earlier developmental stages
- Available online and as R package.
  - Online version (https://cytotrace.stanford.edu) limited to data less than 2.5 GB in size and less than 15k cells.
  - Some features of R package require Python installation
- Example exercises in handout.

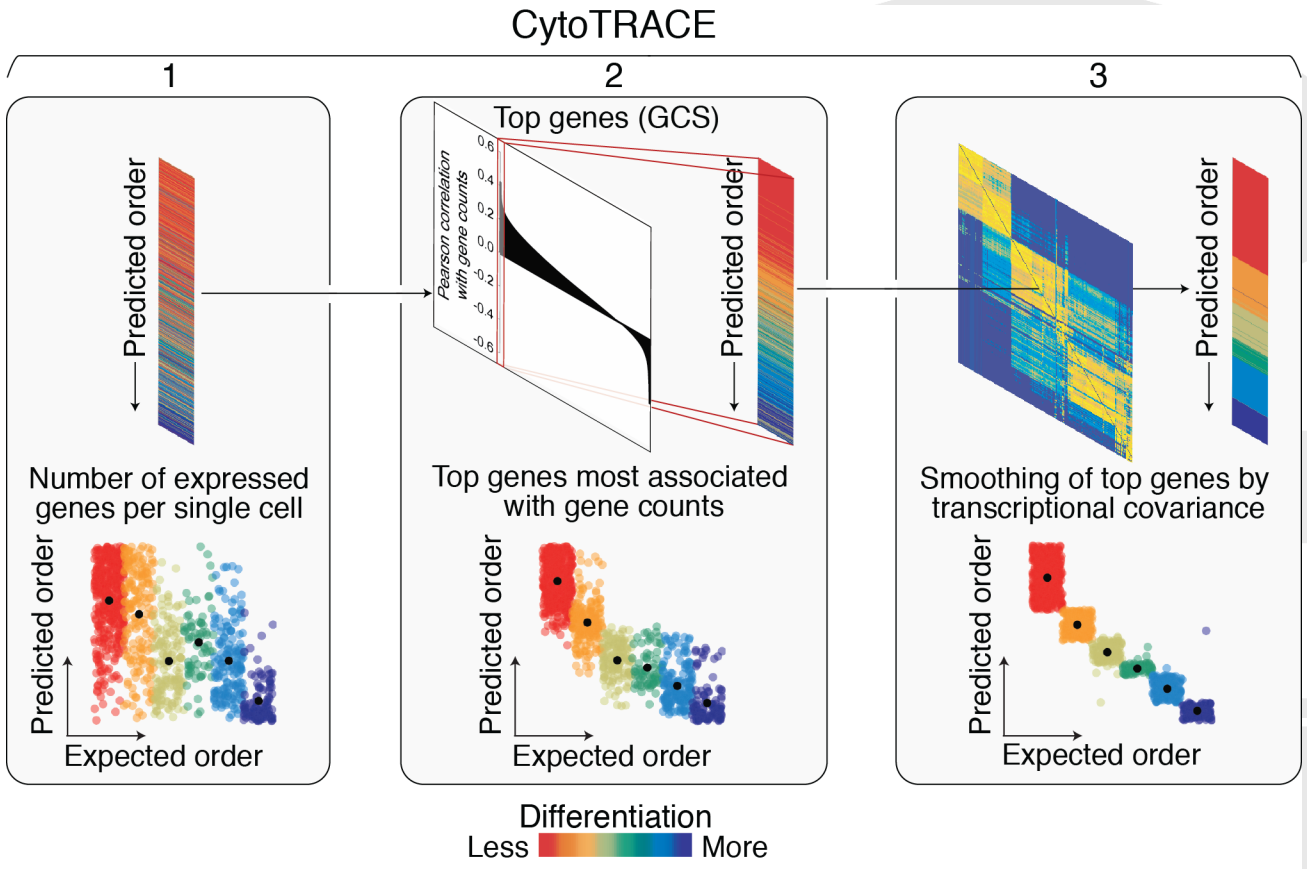




cribioinfo@uic.edu

CytoTRACE

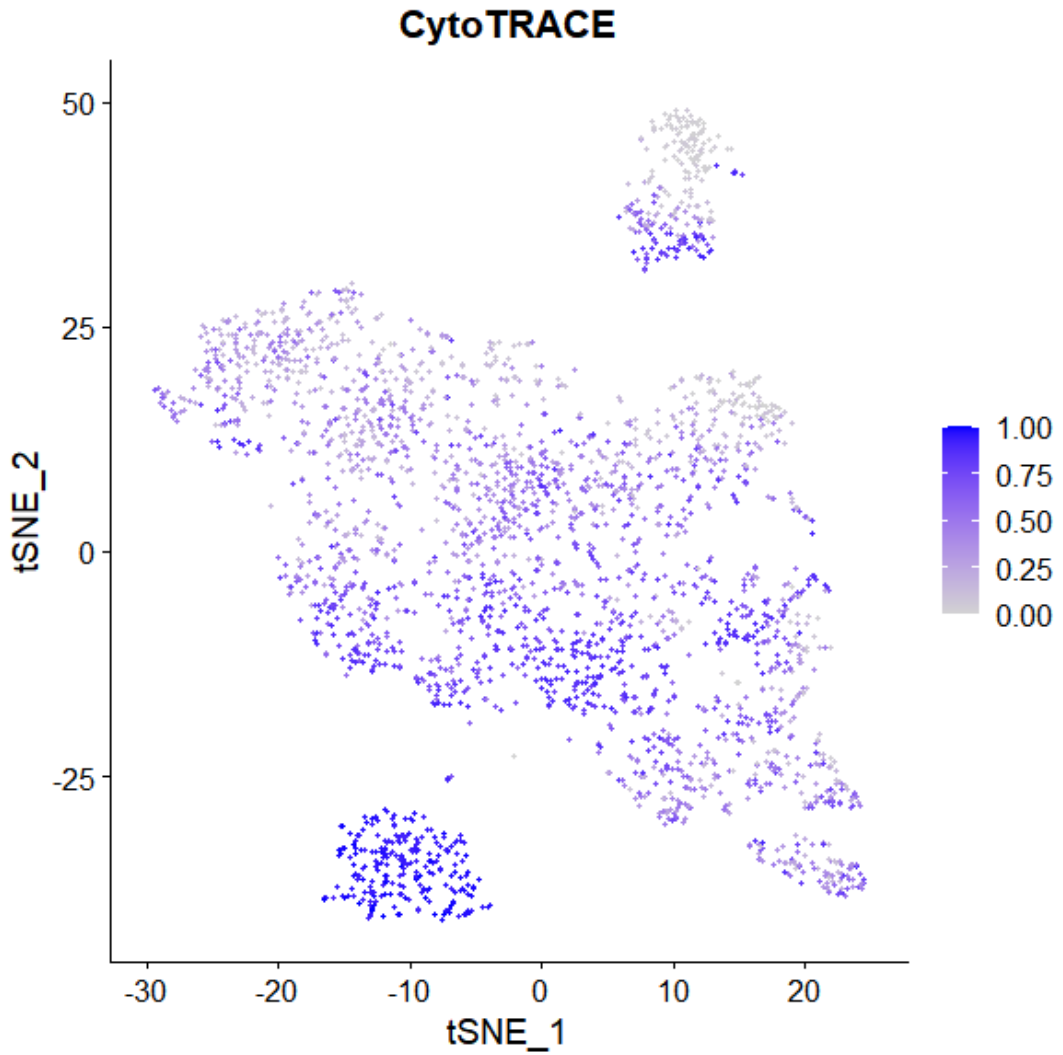
1. Compute number of expressed genes per cell
2. Find genes in which the normalized expression correlation with gene counts per cell. This is gene count signature (GCS)
3. Refine the GCS and then rank cells based refined, smoothed GCS with 0=more differentiated to 1=less differentiated



cribioinfo@uic.edu

Results from CytoTrace

- Numerical “time” variable for each cell
  - Scaled from 0 (earliest) to 1 (latest): *relative* time point based on collection of cells analyzed
  - Easy to run: one command in R, takes a few minutes for typical data set
- No *path* information
  - Can pair with pseudotime analysis for that inference



cribioinfo@uic.edu

Overview of methods again

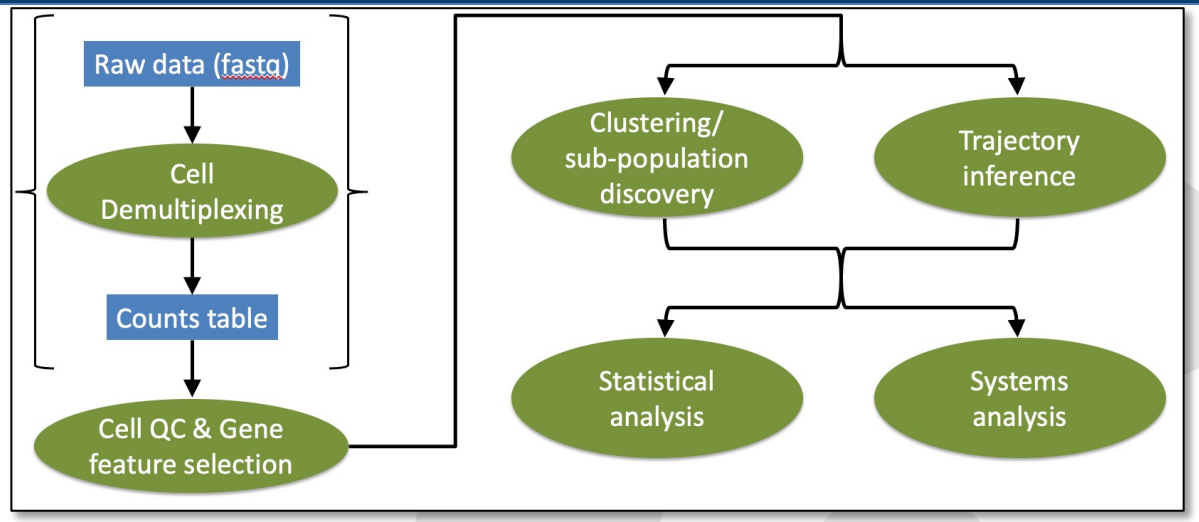
Method	Path	Ordering	Caveats/notes
Pseudotime	Yes	Kind of	• Uses same data as clustering, results will be correlated • Results be sensitive to nonlinear ordination • "Time" variable doesn't distinguish between branches, direction is set arbitrarily
RNA velocity	Yes, qualitative	Yes	• Independent of clustering data • Requires extensive re-processing of data • May be inaccurate for 10X or other 3' biased data
CytoTRACE	No	Yes	• Independent of clustering data • Model is simple and easy to run • Validation is in from developmental processes. May not apply to, e.g., a drug response.

Trajectory analysis methods are very much under active development, and there is no well-established "best" (or even "good") method.

Other steps you may want to do

- Statistical analysis – comparisons of trajectories or pseudotime with:
  - Clusters: where are clusters relative to inferred trajectory
  - Genes: which genes contribute most to trajectory, or have biggest changes at early/later points in trajectory
  - Other trajectory results (robustness)
- Pathway analysis: get pathways associated with top genes

Summary/review



- Most single-cell studies will follow the same general strategy
- **Methods we have outlined are one plausible approach, but not the only one, or necessarily the best one**
  - "Best" may be impossible to quantify
- Packages like Seurat make it easy to manage complex data objects
- Important to understand what you're doing at each step
  - What each command is doing
  - Parameter choices, other algorithms to consider or compare against
  - Robustness of results
- Be comfortable working with objects in R
  - Save RDS objects
  - Extract data frames/vectors from objects for additional analysis
- RNA velocity: same as above, but in python

THANK YOU!

Please complete our workshop survey TODAY:

http://go.uic.edu/RICWorkshopSurvey