Statistics for genomics

Mayo-Illinois Computational Genomics Course

June 11, 2019

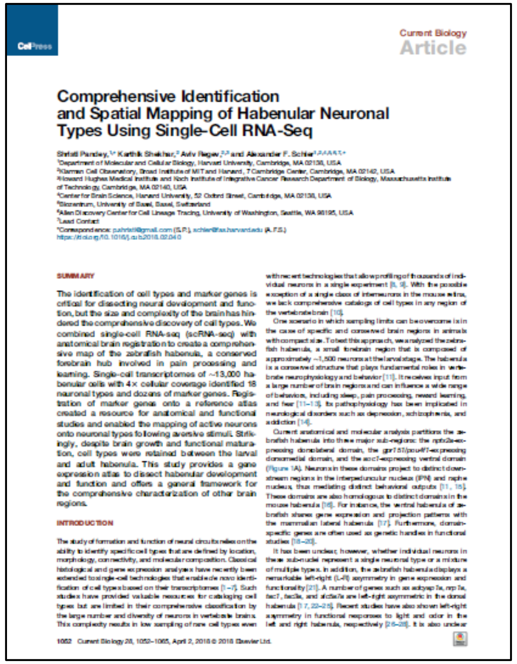
Dave Zhao
Department of Statistics
University of Illinois at Urbana-Champaign

Preparation

- `install.packages(c("Seurat", "glmnet", "ranger", "caret"))`
- Download sample GSM2818521 of GSE109158 from https://urlzs.com/7UNr6

Objective

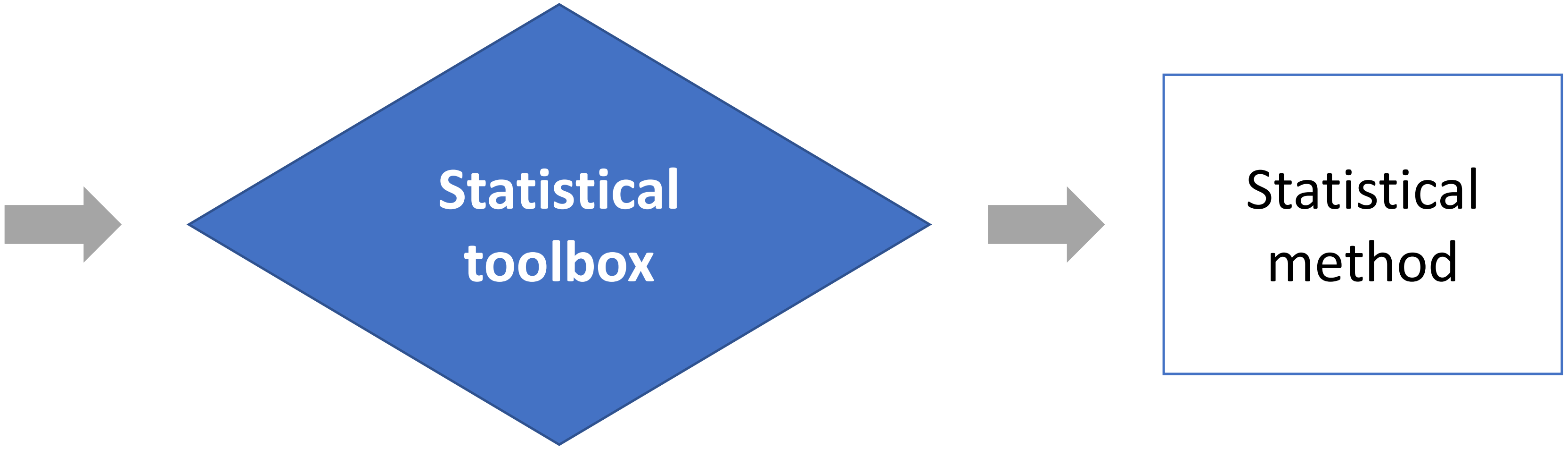
We will illustrate how to identify the appropriate statistical method for a genomics analysis


Current Biology
Article

Comprehensive Identification and Spatial Mapping of Habenular Neuronal Types Using Single-Cell RNA-Seq

Shristi Pandey, Karthik Shekhar, Aviv Regev, and Alexander F. Schier

SUMMARY



Classifying statistical tools

Data structure

Statistical task

	No dependent variables	Continuous outcome	Censored outcomes	Etc.
Visualize				
Identify latent factors				
Cluster observations				
Select features				
Etc.				

APPROPRIATE STATISTICAL METHODS

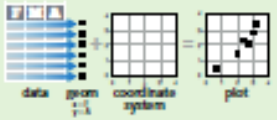
Examples from basic statistics

Data Visualization with ggplot2 Cheat Sheet

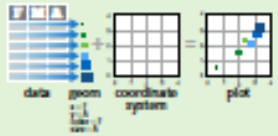
RStudio

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same few components: a **data** set, a set of **geoms**—visual marks that represent data points, and a **coordinate system**.



To display data values, map variables in the data set to aesthetic properties of the geom like **size**, **color**, and **x** and **y** locations.



Build a graph with **ggplot()** or **qplot()**

ggplot(data = mpg, **aes**(x = cty, y = hwy))
Begins a plot that you finish by adding layers to. No defaults, but provides more control than qplot().

data

```
ggplot(mpg, aes(hwy, cty)) +
 geom_point(aes(color = cyl)) +
 geom_smooth(method ="lm") +
 coord_cartesian() +
 scale_color_gradient() +
 theme_bw()
```

add layers, elements with +

layer = geom + default stat + layer specific mappings

additional elements

Add a new layer to a plot with a **geom_*()** or **stat_*()** function. Each provides a geom, a set of aesthetic mappings, and a default stat and position adjustment.

aesthetic mappings | data | geom

qplot(x = cty, y = hwy, color = cyl, data = mpg, geom = "point")
Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

last_plot()
Returns the last plot

ggsave("plot.png", width = 5, height = 5)
Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

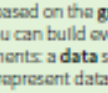
One Variable

Continuous

a <- ggplot(mpg, aes(hwy))

a + **geom_area**(**stat = "bin"**)
x, y, alpha, color, fill, linetype, size
b + geom_area(aes(y = ..density..), stat = "bin")

a + **geom_density**(kernel = "gaussian")
x, y, alpha, color, fill, linetype, size, weight
b + geom_density(aes(y = ..county..))



a + **geom_dotplot**()
x, y, alpha, color, fill

a + **geom_freqpoly**()
x, y, alpha, color, linetype, size
b + geom_freqpoly(aes(y = ..density..))

a + **geom_histogram**(binwidth = 5)
x, y, alpha, color, fill, linetype, size, weight
b + geom_histogram(aes(y = ..density..))

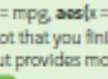
Discrete

b <- ggplot(mpg, aes(fl))

b + **geom_bar**()
x, alpha, color, fill, linetype, size, weight

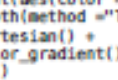
Graphical Primitives

map <- map_data("state")
c <- ggplot(map, aes(long, lat))

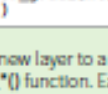


c + **geom_polygon**(aes(group = group))
x, y, alpha, color, fill, linetype, size

d <- ggplot(economics, aes(date, unemploy))

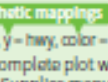


d + **geom_path**(lineend="butt", linejoin="round", linemitre=1)
x, y, alpha, color, linetype, size

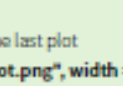


d + **geom_ribbon**(aes(ymin=unemploy - 900, ymax=unemploy + 900))
x, ymax, ymin, alpha, color, fill, linetype, size

e <- ggplot(seals, aes(x = long, y = lat))



e + **geom_segment**(aes(xend = long + delta_long, yend = lat + delta_lat))
x, xend, y, yend, alpha, color, linetype, size



e + **geom_rect**(aes(xmin = long, ymin = lat, xmax= long + delta_long, ymax = lat + delta_lat))
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

Two Variables

Continuous X, Continuous Y

f <- ggplot(mpg, aes(cty, hwy))

f + **geom_blank**()
(Useful for expanding limits)

f + **geom_jitter**()
x, y, alpha, color, fill, shape, size

f + **geom_point**()
x, y, alpha, color, fill, shape, size

f + **geom_quantile**()
x, y, alpha, color, linetype, size, weight

f + **geom_rug**(sides = "bl")
alpha, color, linetype, size

f + **geom_smooth**(method = lm)
x, y, alpha, color, fill, linetype, size, weight

f + **geom_text**(aes(label = cty))
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

Discrete X, Continuous Y

g <- ggplot(mpg, aes(class, hwy))

g + **geom_bar**(**stat = "identity"**)
x, y, alpha, color, fill, linetype, size, weight

g + **geom_boxplot**()
lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, shape, size, weight

g + **geom_dotplot**(binaxis = "y", stackdir = "center")
x, y, alpha, color, fill

g + **geom_violin**(scale = "area")
x, y, alpha, color, fill, linetype, size, weight

Discrete X, Discrete Y

h <- ggplot(diamonds, aes(cut, color))

h + **geom_jitter**()
x, y, alpha, color, fill, shape, size

Continuous Bivariate Distribution

i <- ggplot(movies, aes(year, rating))

i + **geom_bin2d**(binwidth = c(5, 0.5))
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight

i + **geom_density2d**()
x, y, alpha, colour, linetype, size

i + **geom_hex**()
x, y, alpha, colour, fill size

Continuous Function

j <- ggplot(economics, aes(date, unemploy))

j + **geom_area**()
x, y, alpha, color, fill, linetype, size

j + **geom_line**()
x, y, alpha, color, linetype, size

j + **geom_step**(direction = "hv")
x, y, alpha, color, linetype, size

Visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
k <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))

k + **geom_crossbar**(fatten = 2)
x, y, ymax, ymin, alpha, color, fill, linetype, size

k + **geom_errorbar**()
x, ymax, ymin, alpha, color, linetype, size, width (also **geom_errorbarh()**)

k + **geom_linerange**()
x, ymin, ymax, alpha, color, linetype, size

k + **geom_pointrange**()
x, y, ymin, ymax, alpha, color, fill, linetype, shape, size

Maps

data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))
map <- map_data("state")
l <- ggplot(data, aes(fill = murder))

l + **geom_map**(aes(map_id = state), map = map) + **expand_limits**(x = map$long, y = map$lat)
map_id, alpha, color, fill, linetype, size

Three Variables

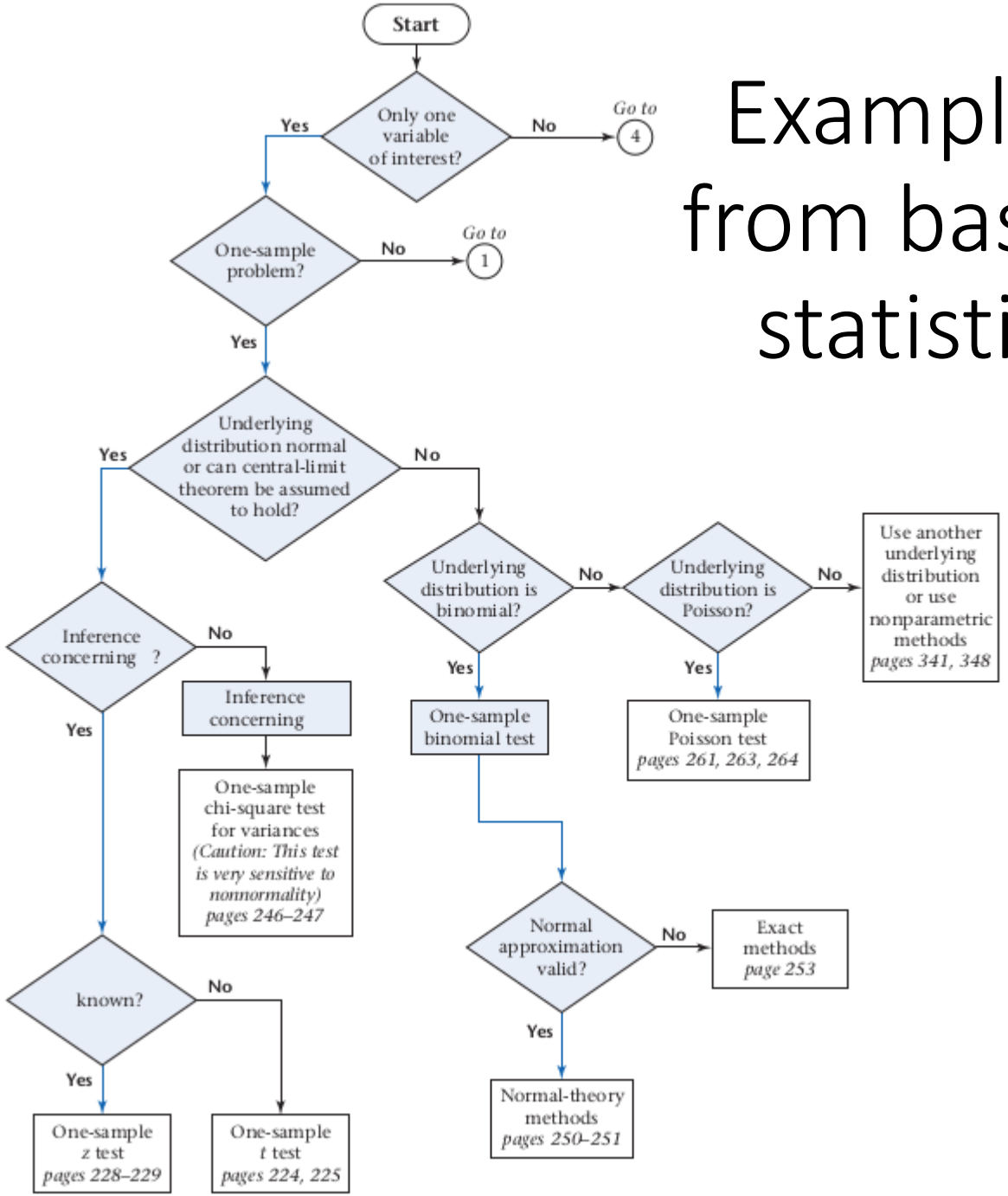
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))
m <- ggplot(seals, aes(long, lat))

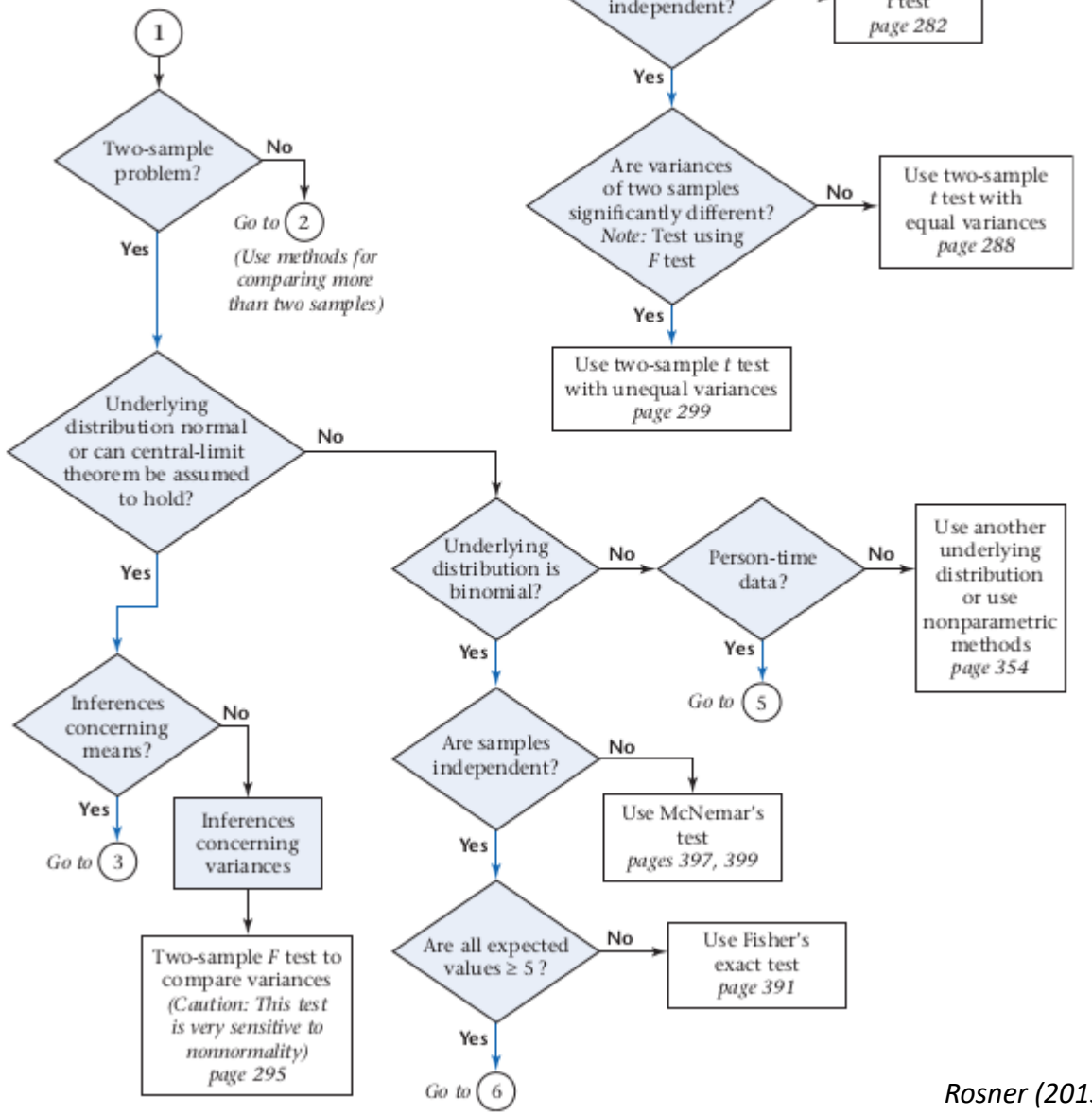
m + **geom_contour**(aes(z = z))
x, y, z, alpha, colour, linetype, size, weight

m + **geom_raster**(aes(fill = z), hjust=0.5, vjust=0.5, interpolate=FALSE)
x, y, alpha, fill (Fast)

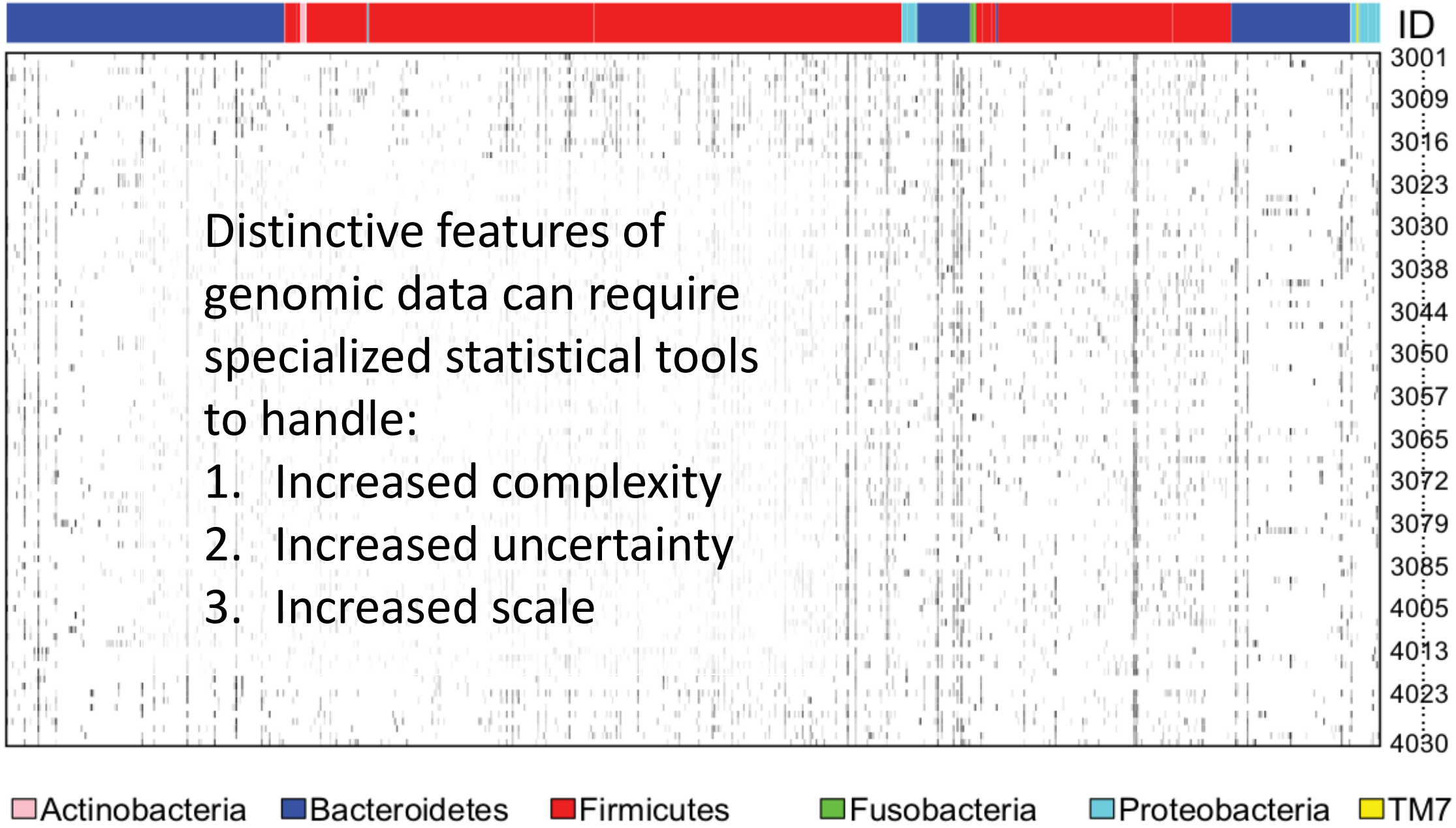
m + **geom_tile**(aes(fill = z))
x, y, alpha, color, fill, linetype, size (slow)

Examples from basic statistics

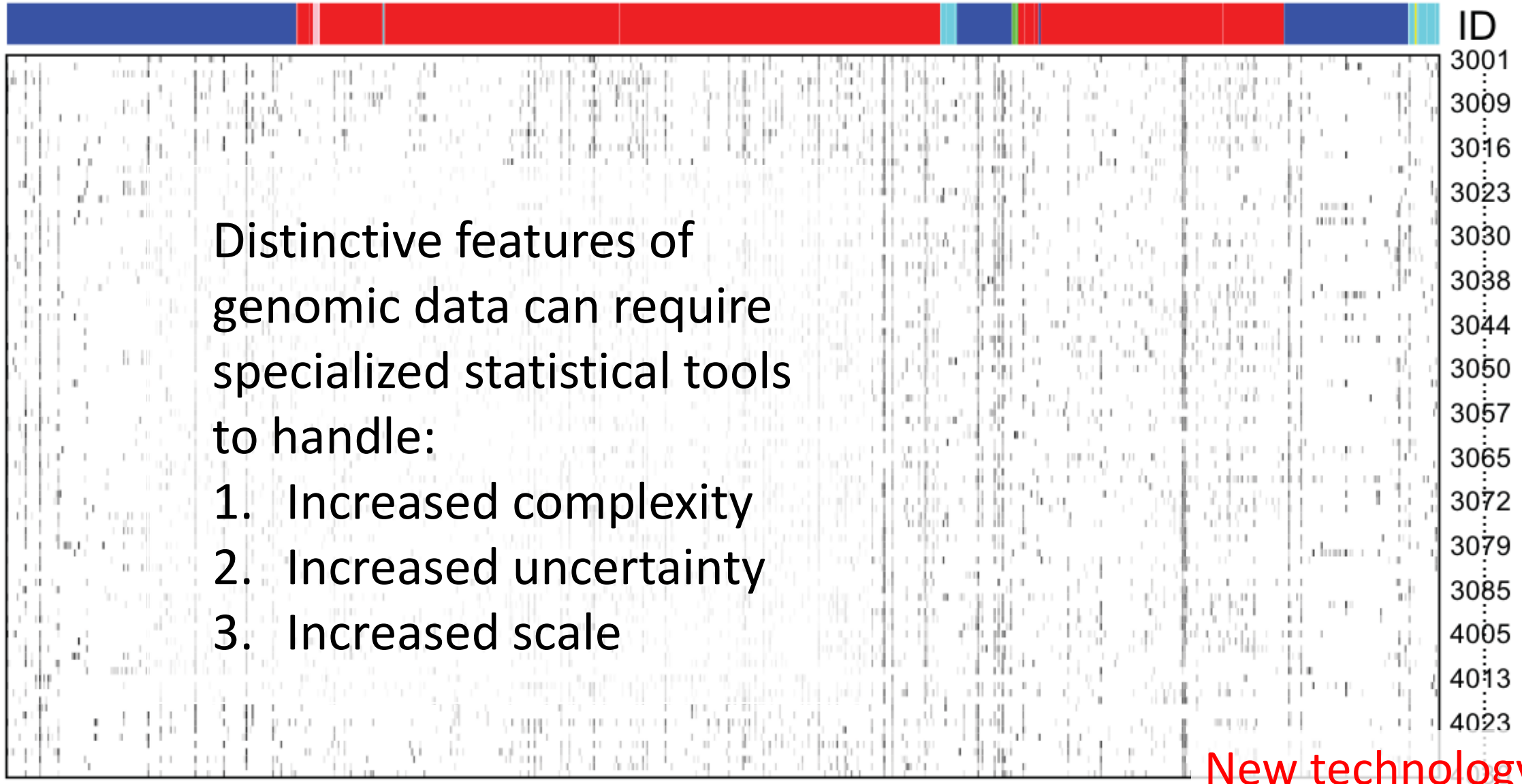




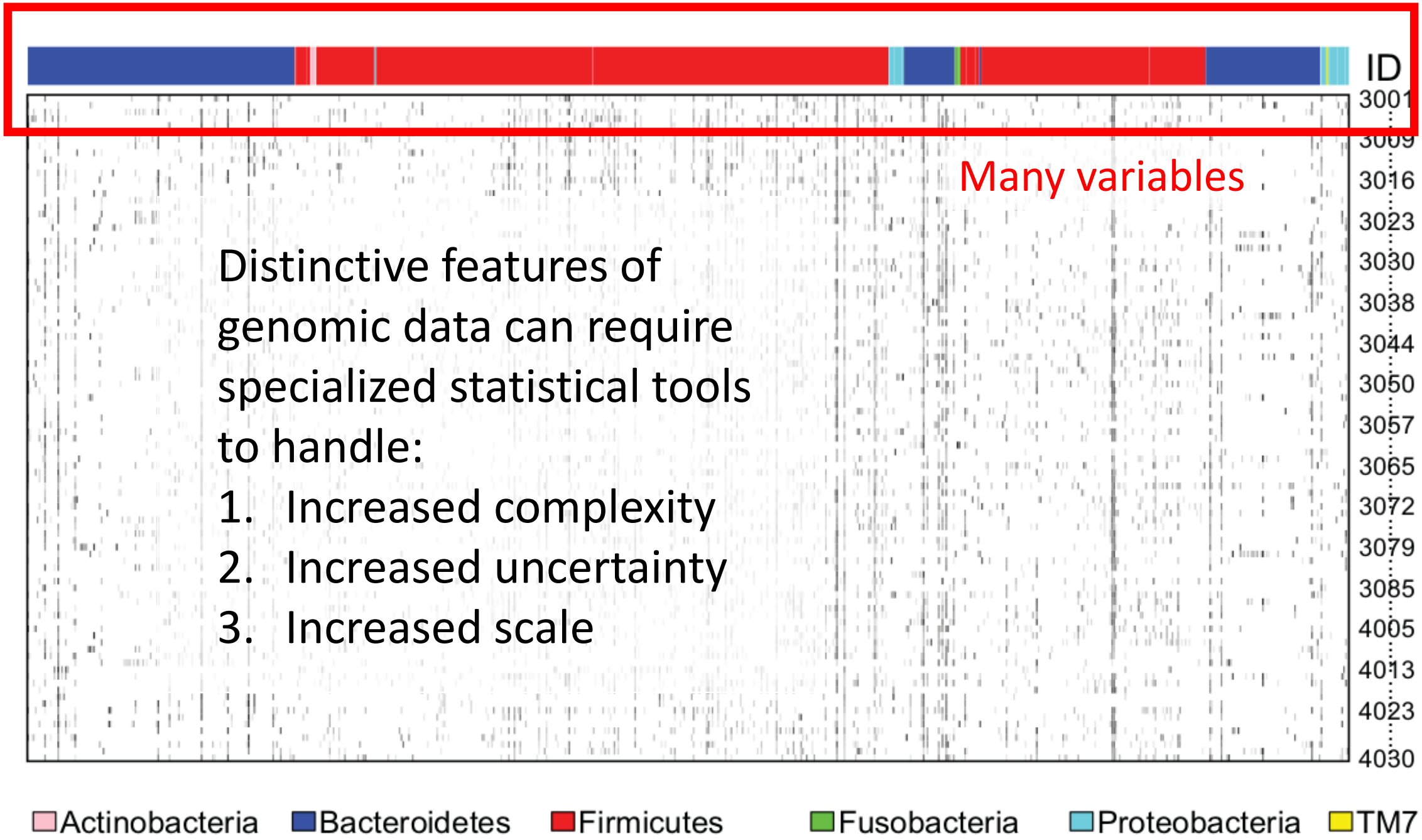
Rosner (2015)



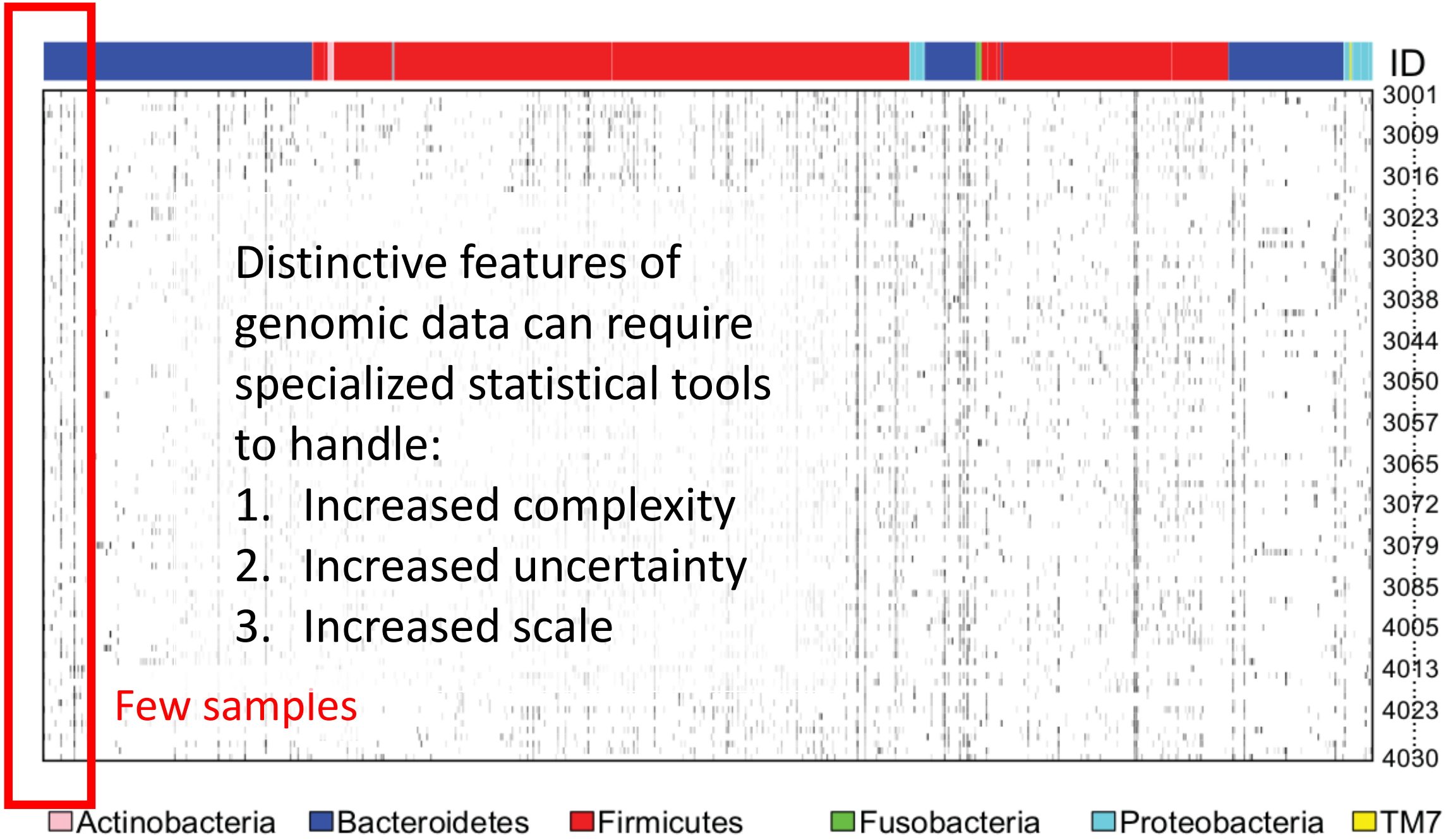
Wu, Chen et al. (2011)



Wu, Chen et al. (2011)



Wu, Chen et al. (2011)

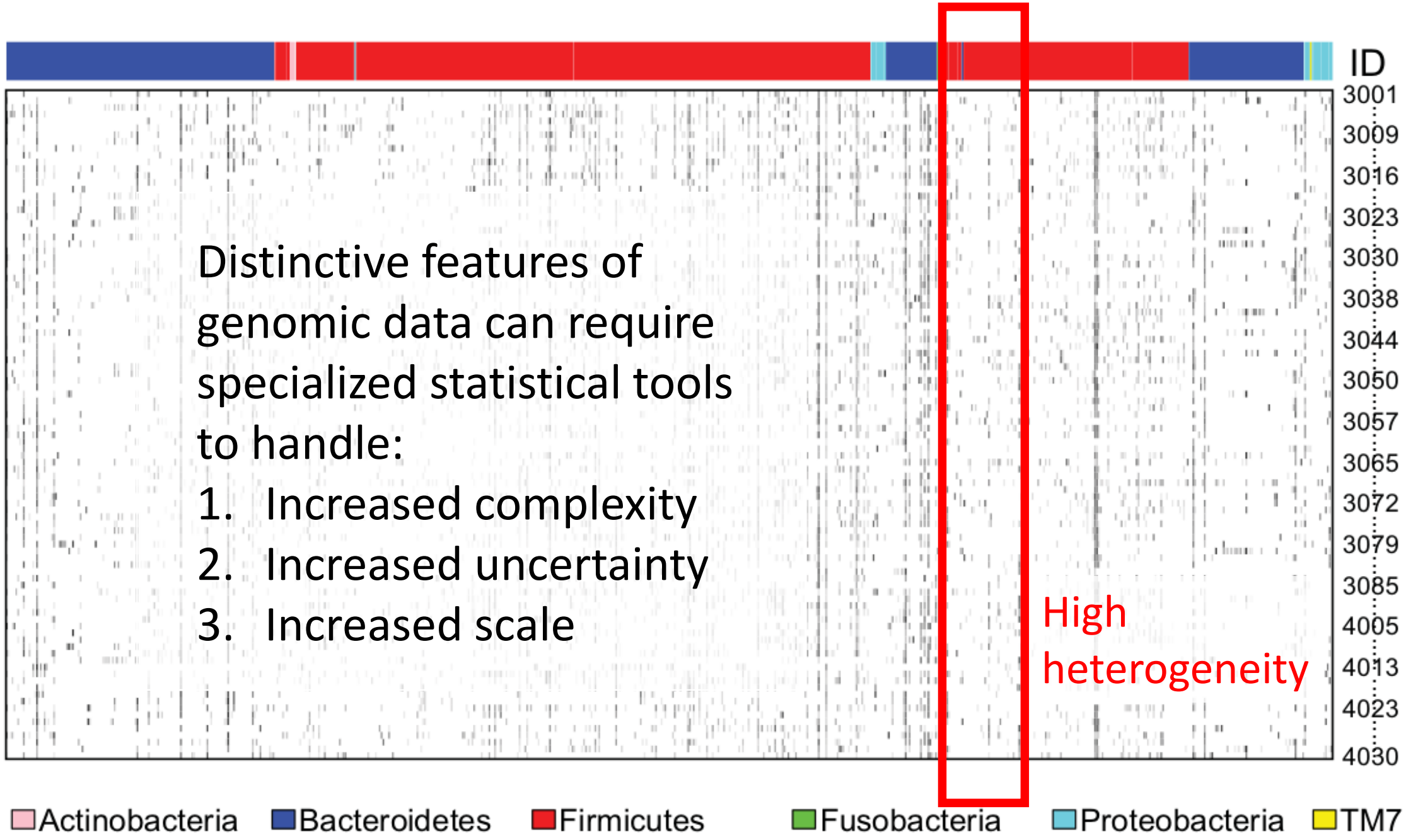


Distinctive features of genomic data can require specialized statistical tools to handle:

1. Increased complexity
2. Increased uncertainty
3. Increased scale

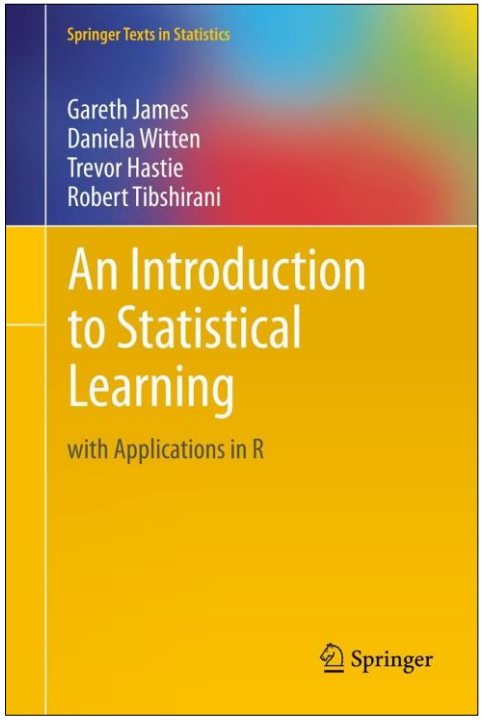
Few samples

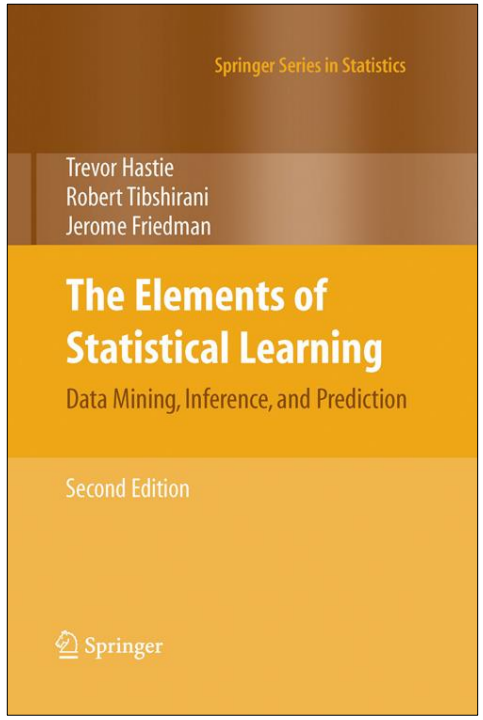
Wu, Chen et al. (2011)



Wu, Chen et al. (2011)

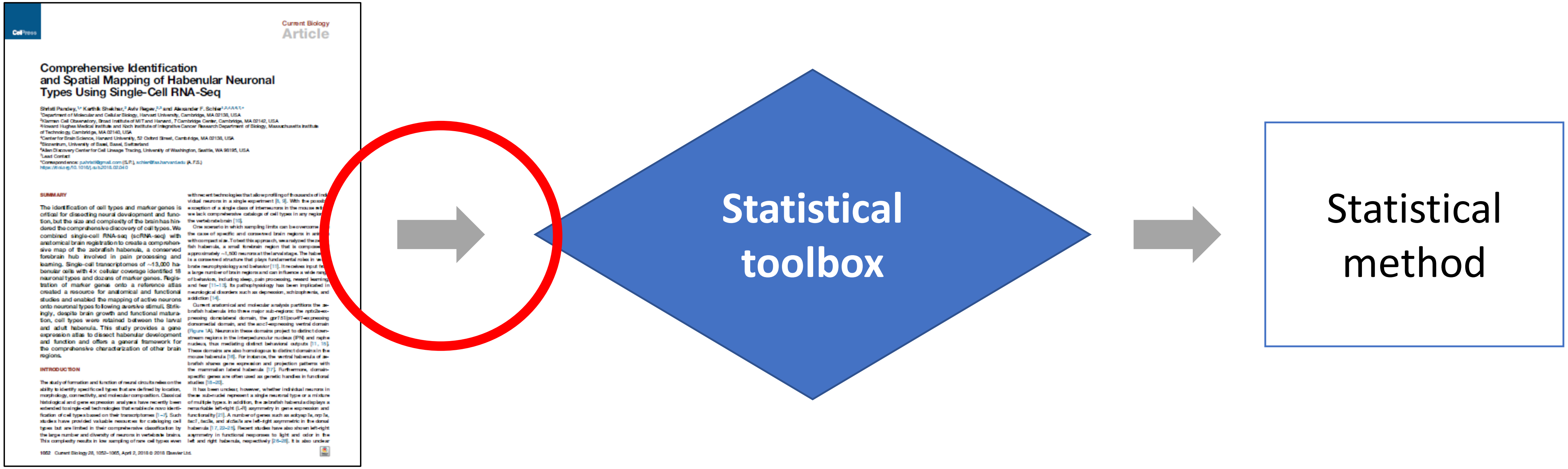
We will discuss some statistical tools that are frequently used in genomics



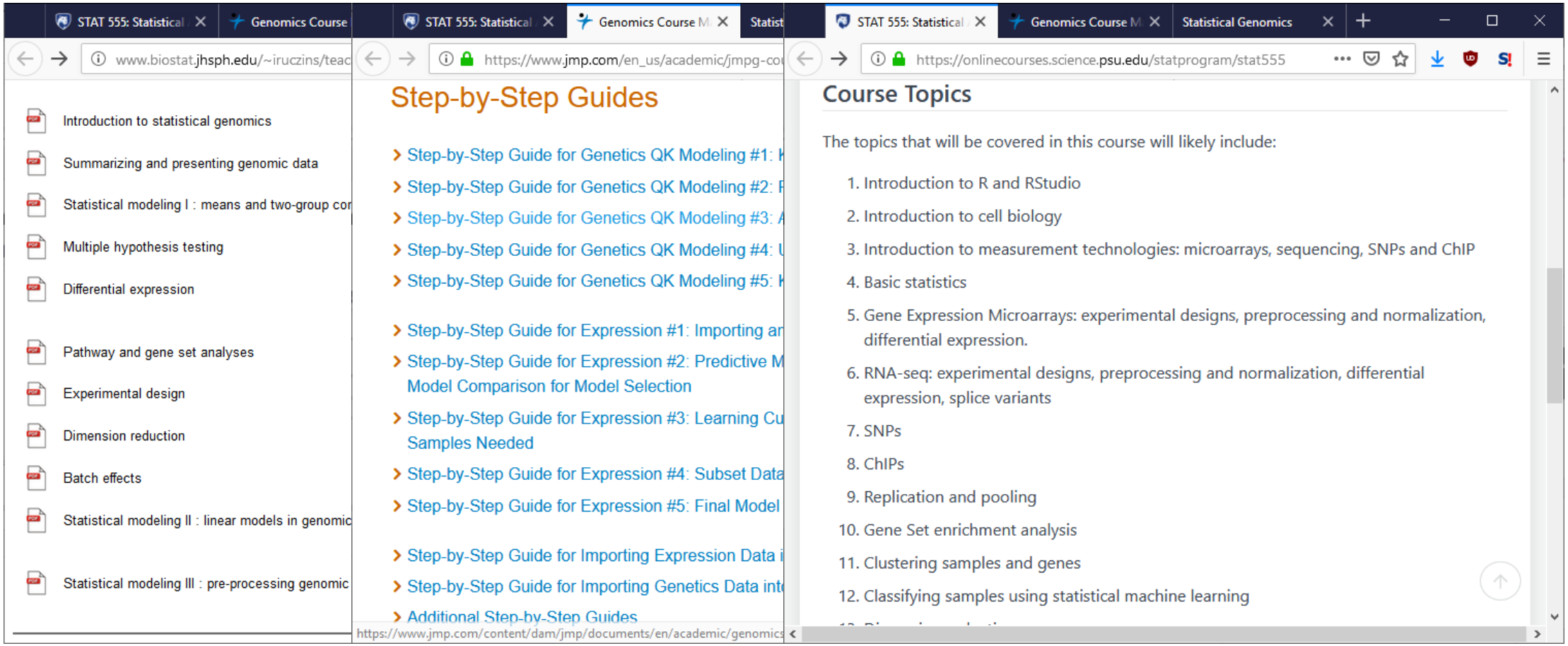


We will also discuss a general framework for organizing new tools.

Choosing the right tool for a given biological question requires creativity and experience

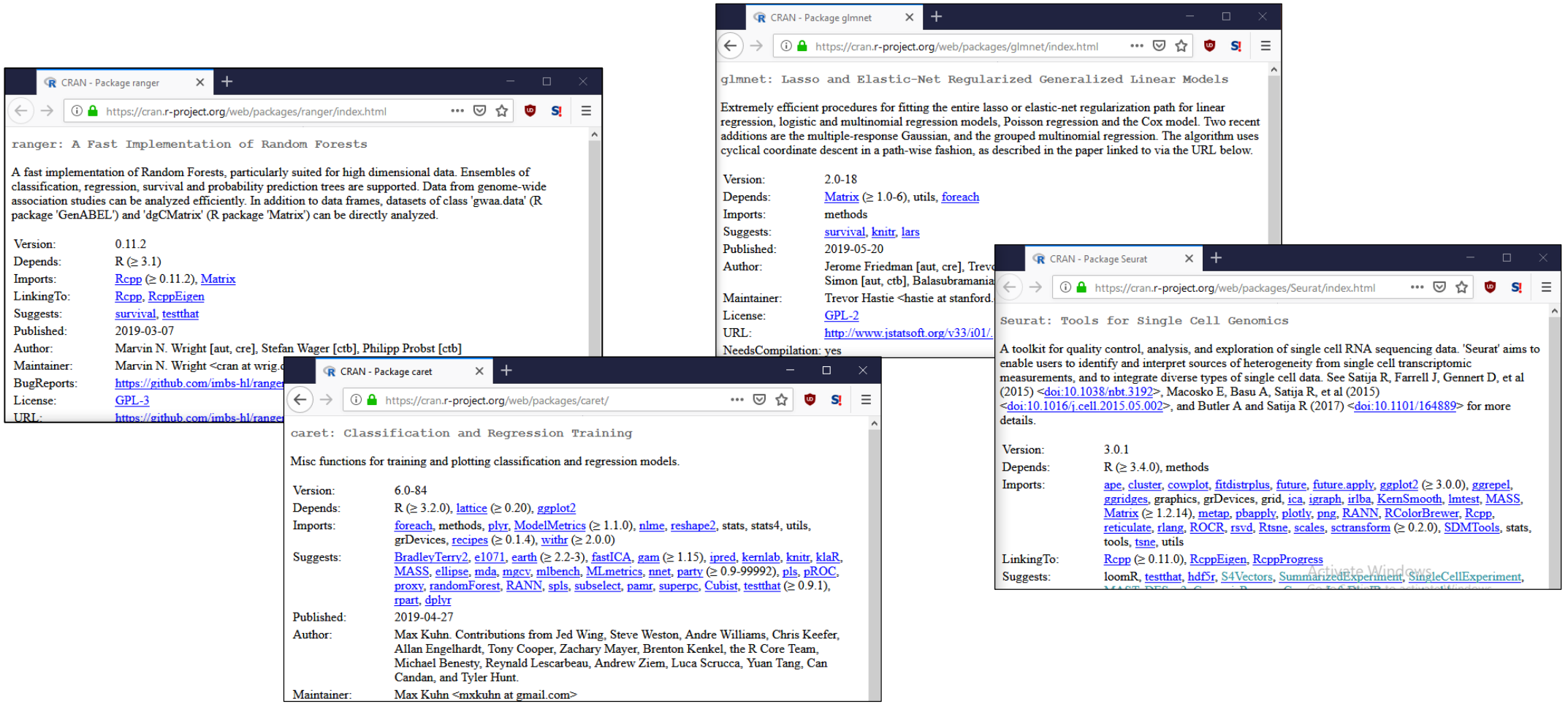


There are other frameworks that are organized by biological question rather than statistical tool

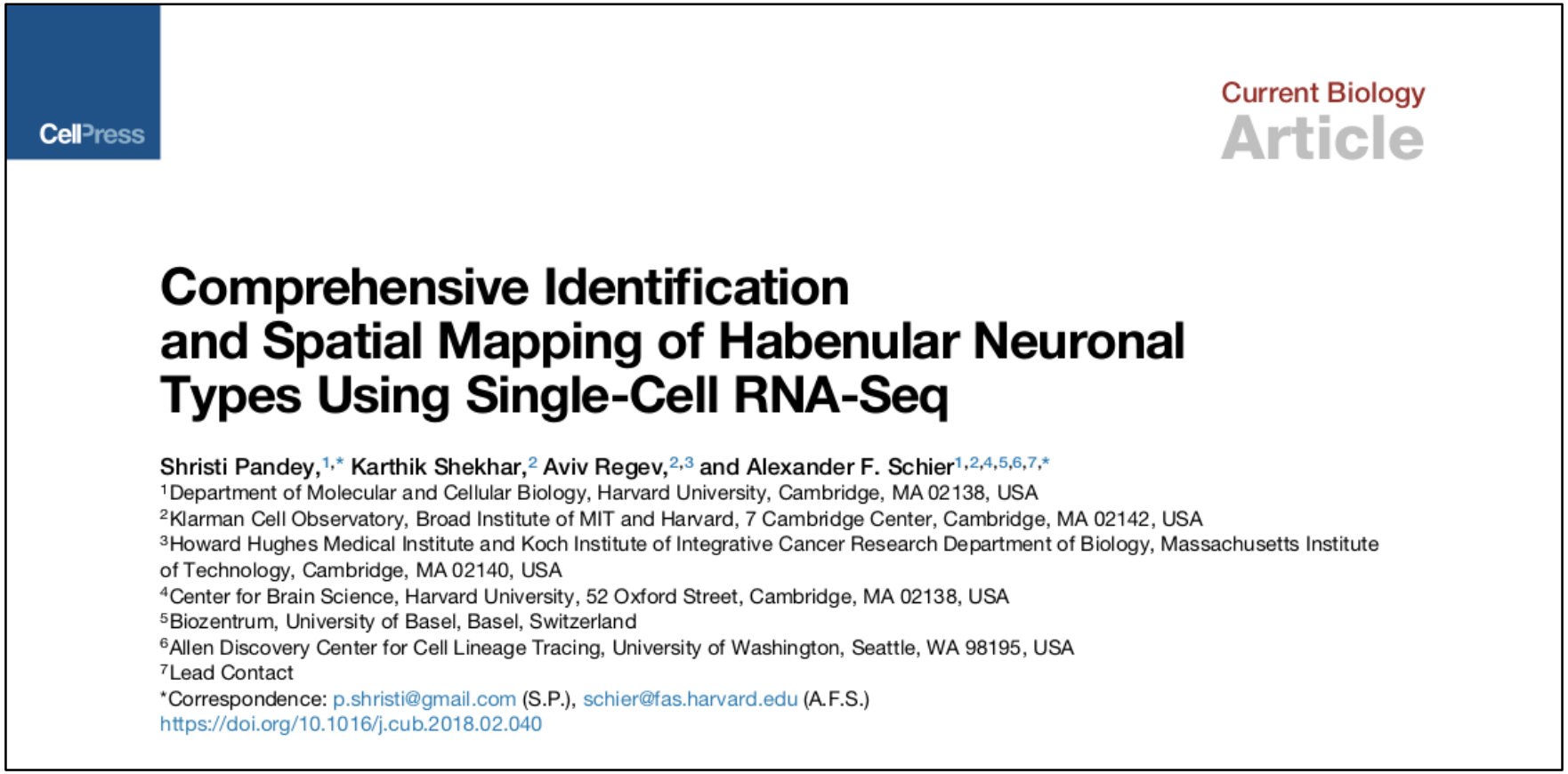


Example analysis using R

Use R packages to perform data analysis



To illustrate these tools, we will analyze single-cell RNA-seq data in R



CellPress

Current Biology
Article

Comprehensive Identification and Spatial Mapping of Habenular Neuronal Types Using Single-Cell RNA-Seq

Shristi Pandey,[1,*] Karthik Shekhar,[2] Aviv Regev,[2,3] and Alexander F. Schier[1,2,4,5,6,7,*]
[1]Department of Molecular and Cellular Biology, Harvard University, Cambridge, MA 02138, USA
[2]Klarman Cell Observatory, Broad Institute of MIT and Harvard, 7 Cambridge Center, Cambridge, MA 02142, USA
[3]Howard Hughes Medical Institute and Koch Institute of Integrative Cancer Research Department of Biology, Massachusetts Institute of Technology, Cambridge, MA 02140, USA
[4]Center for Brain Science, Harvard University, 52 Oxford Street, Cambridge, MA 02138, USA
[5]Biozentrum, University of Basel, Basel, Switzerland
[6]Allen Discovery Center for Cell Lineage Tracing, University of Washington, Seattle, WA 98195, USA
[7]Lead Contact
*Correspondence: p.shristi@gmail.com (S.P.), schier@fas.harvard.edu (A.F.S.)
https://doi.org/10.1016/j.cub.2018.02.040

Anatomy of a basic R command

```
pandey =
read.table("GSM2818521_larva_counts_matrix.txt")
```

- Case-sensitive
- Function(): performs pre-programmed calculations given inputs and options
- Variable: stores values and outputs of function, name cannot contain whitespace and cannot start with a special character

Basic preprocessing of single-cell RNA-seq data using Seurat

```
library(Seurat)
s_obj = CreateSeuratObject(counts = pandey,
min.cells = 3, min.features = 200)
s_obj = NormalizeData(s_obj)
s_obj = FindVariableFeatures(s_obj)
s_obj = ScaleData(s_obj)
```

Statistical methods for genomics

Where statistics appears in a standard genomic analysis workflow

1. Experimental design ← Not covered today
2. Quality control
3. Preprocessing
4. Normalization and batch correction

Uses statistics but is highly dependent on technology

5. Analysis ← The focus of today's discussion
6. Biological interpretation

Classifying statistical tools

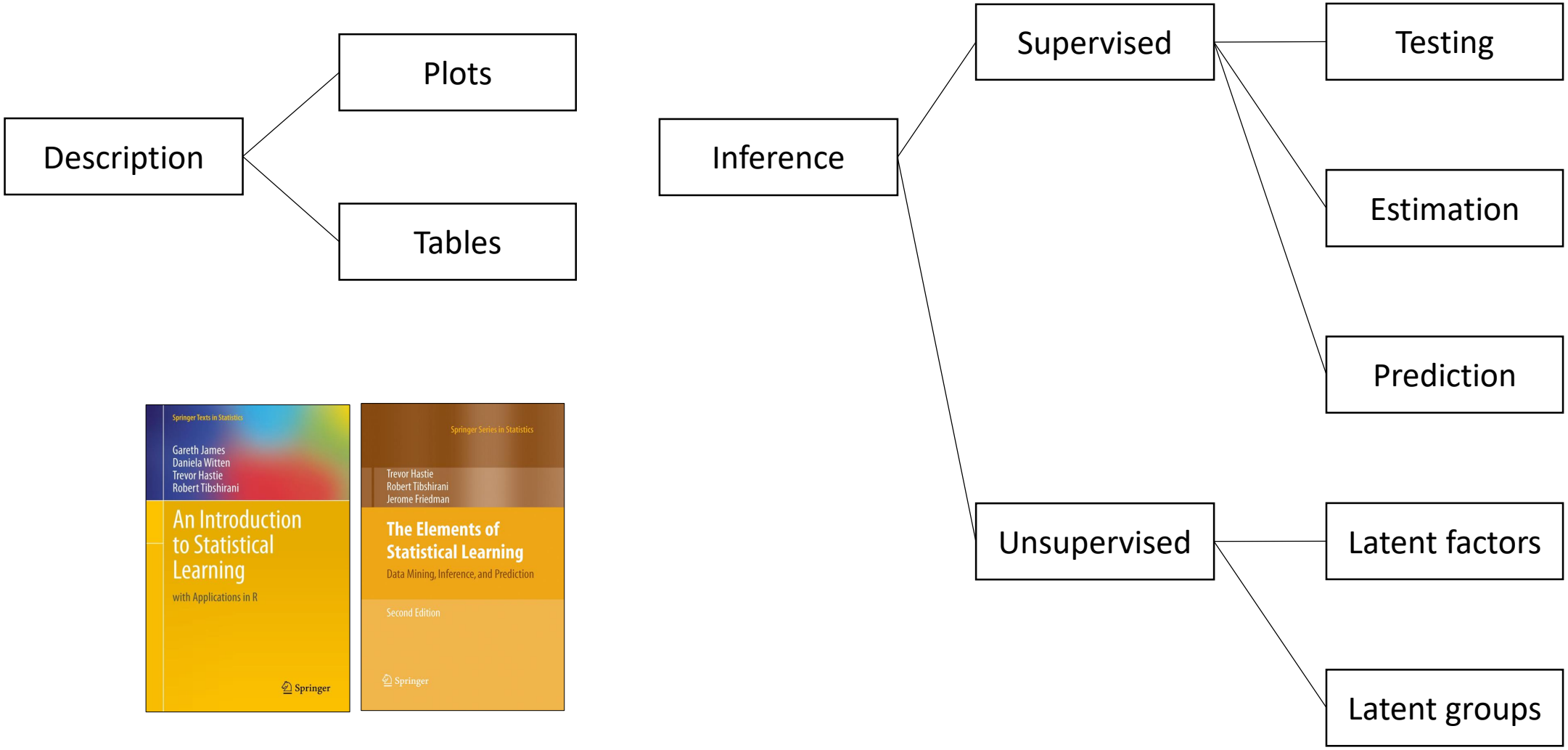
Data structure

Statistical task

	No dependent variables	Continuous outcome	Censored outcomes	Etc.
Visualize				
Identify latent factors				
Cluster observations				
Select features				
Etc.				

APPROPRIATE STATISTICAL METHODS

Classifying statistical tasks



Classifying data structures

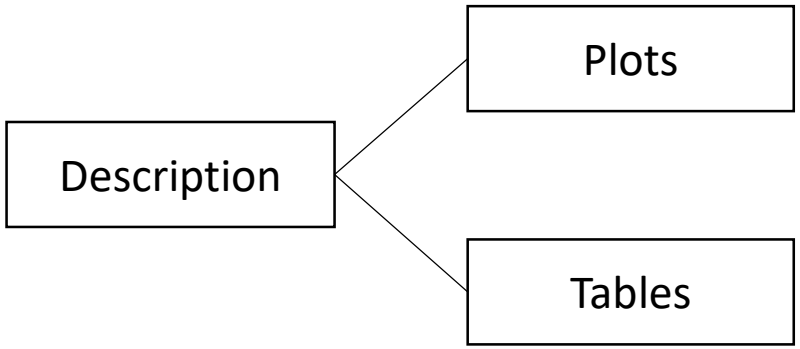
- Can vary widely, and classification is difficult
- Important factors in genomics:
    1. Data type
    2. Number of samples relative to number of variables

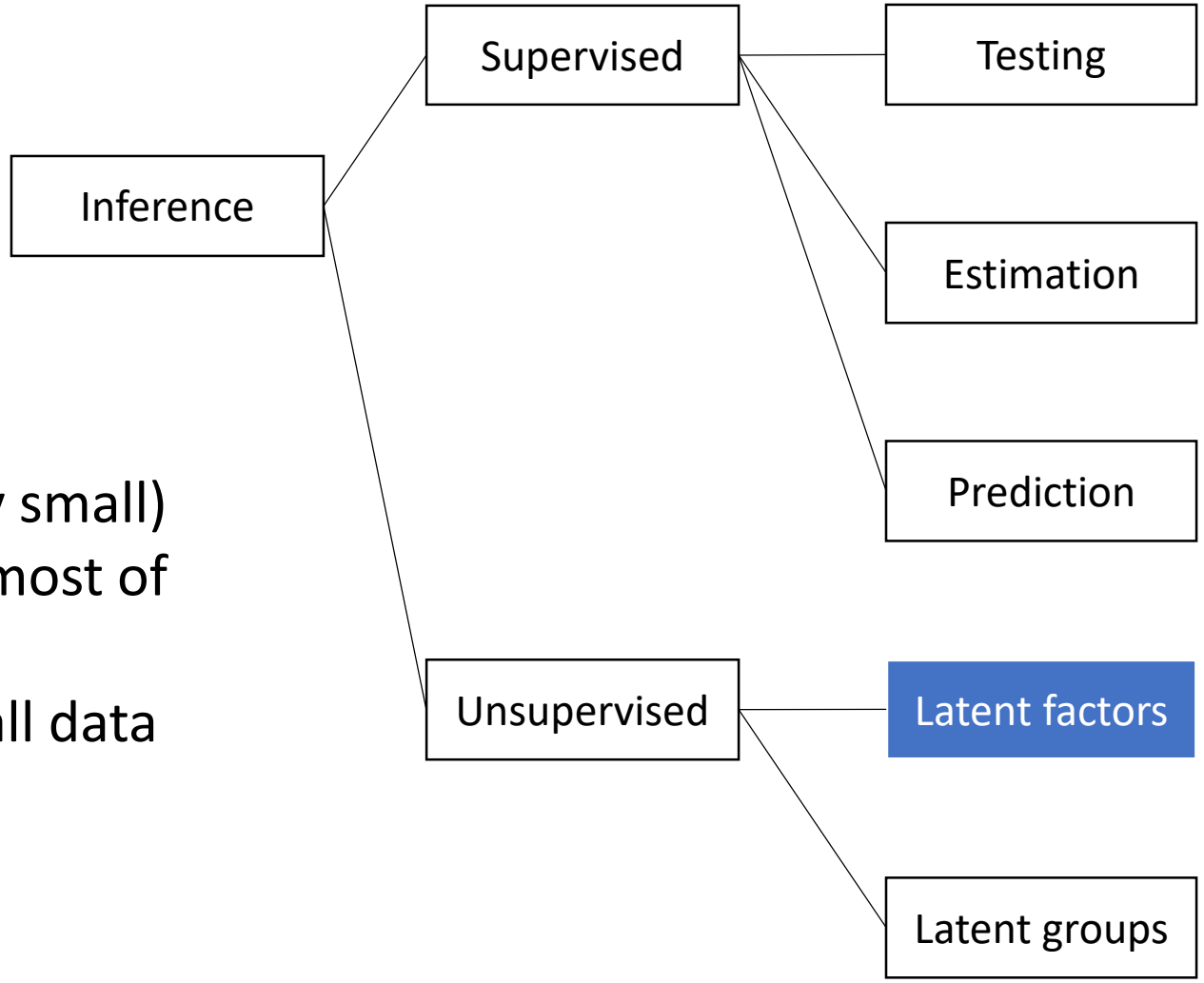
PCA

```
> dim(pandey)
[1] 24105  4365
> pandey[1:3, 1:2]
      larvalR2_AAACCTGAGACAGAGA.1 larvalR2_AAACCTGAGACTTTCG.1
SYN3                            0                           0
PTPRO                           1                           0
EPS8                            0                           0
```

Research question:
Can the gene expression information be summarized in fewer features?

PCA

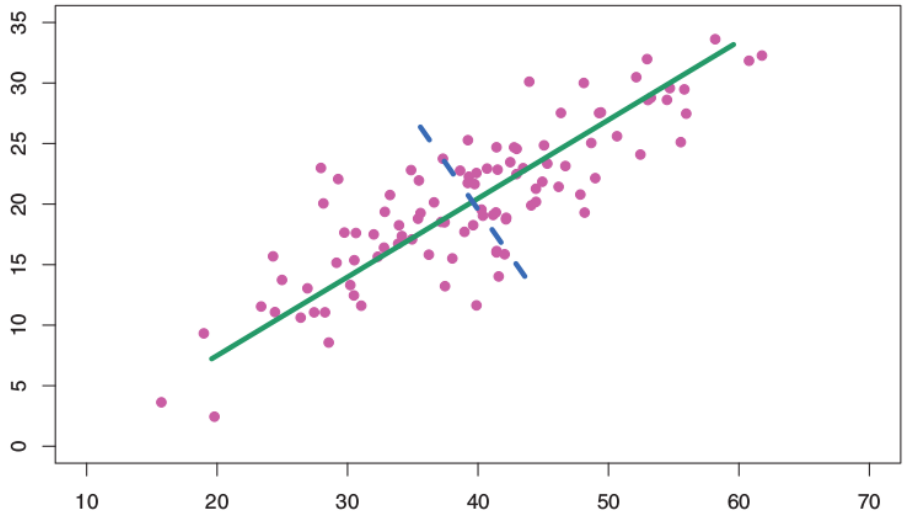


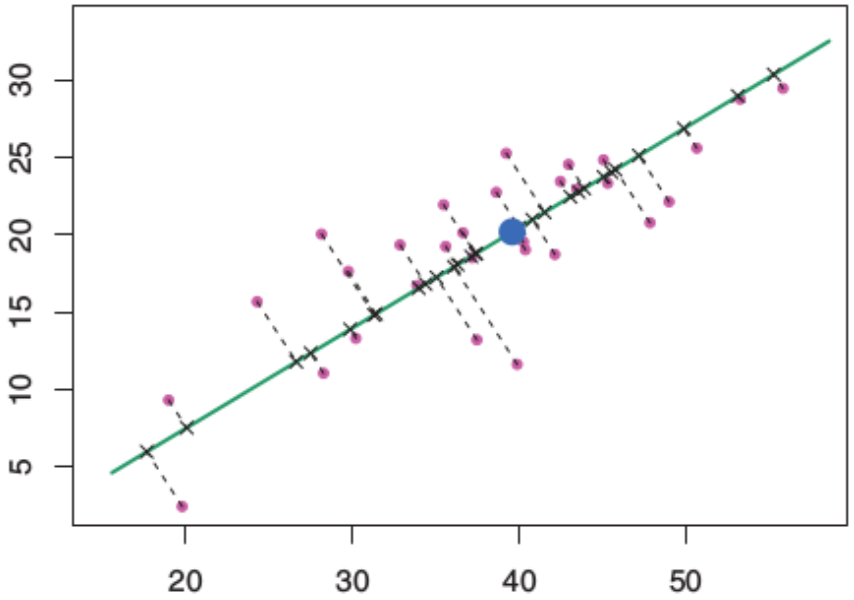


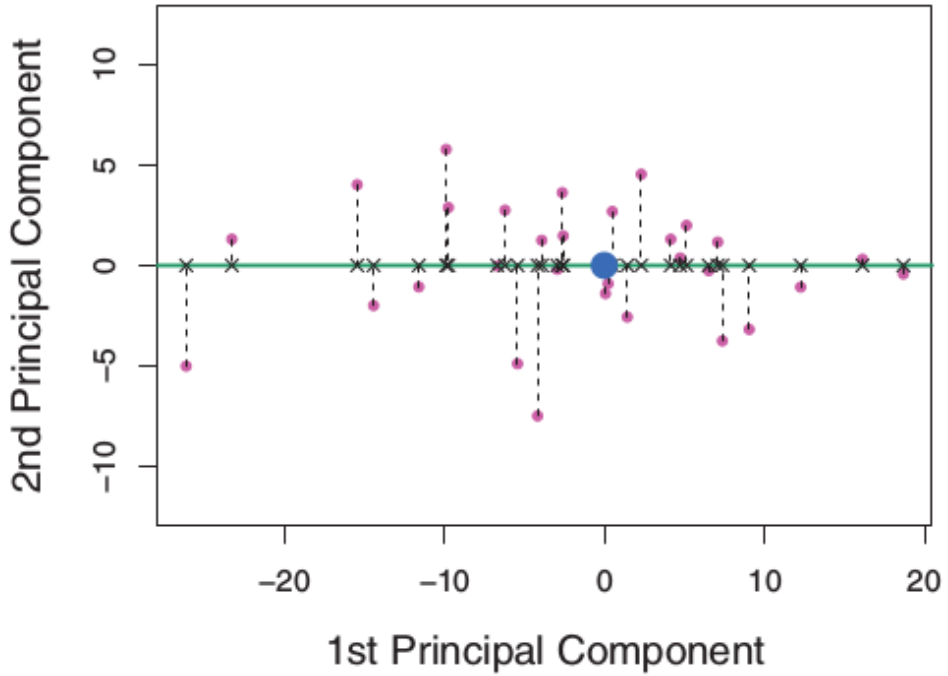
Statistical task: calculate a (usually small) set of latent factors that captures most of the information in the dataset

Data structure: can be applied to all data structures

PCA







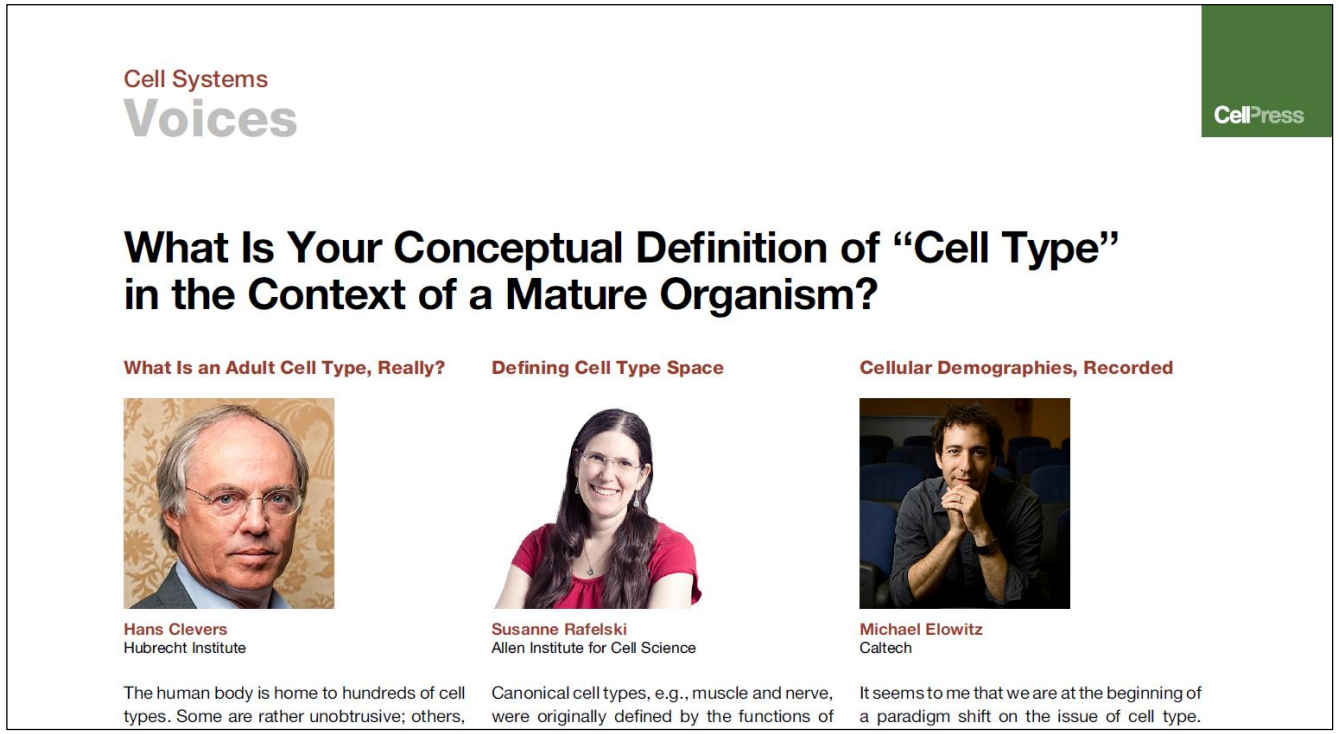
ISL Chapter 6

PCA using Seurat

```
s_obj = RunPCA(s_obj)
```

Graph clustering

Research question:
How many cell types exist in the larval zebrafish habenula?



Cell Systems
Voices

CellPress

What Is Your Conceptual Definition of "Cell Type" in the Context of a Mature Organism?

What Is an Adult Cell Type, Really?

Hans Clevers
Hubrecht Institute

The human body is home to hundreds of cell types. Some are rather unobtrusive; others,

Defining Cell Type Space

Susanne Rafelski
Allen Institute for Cell Science

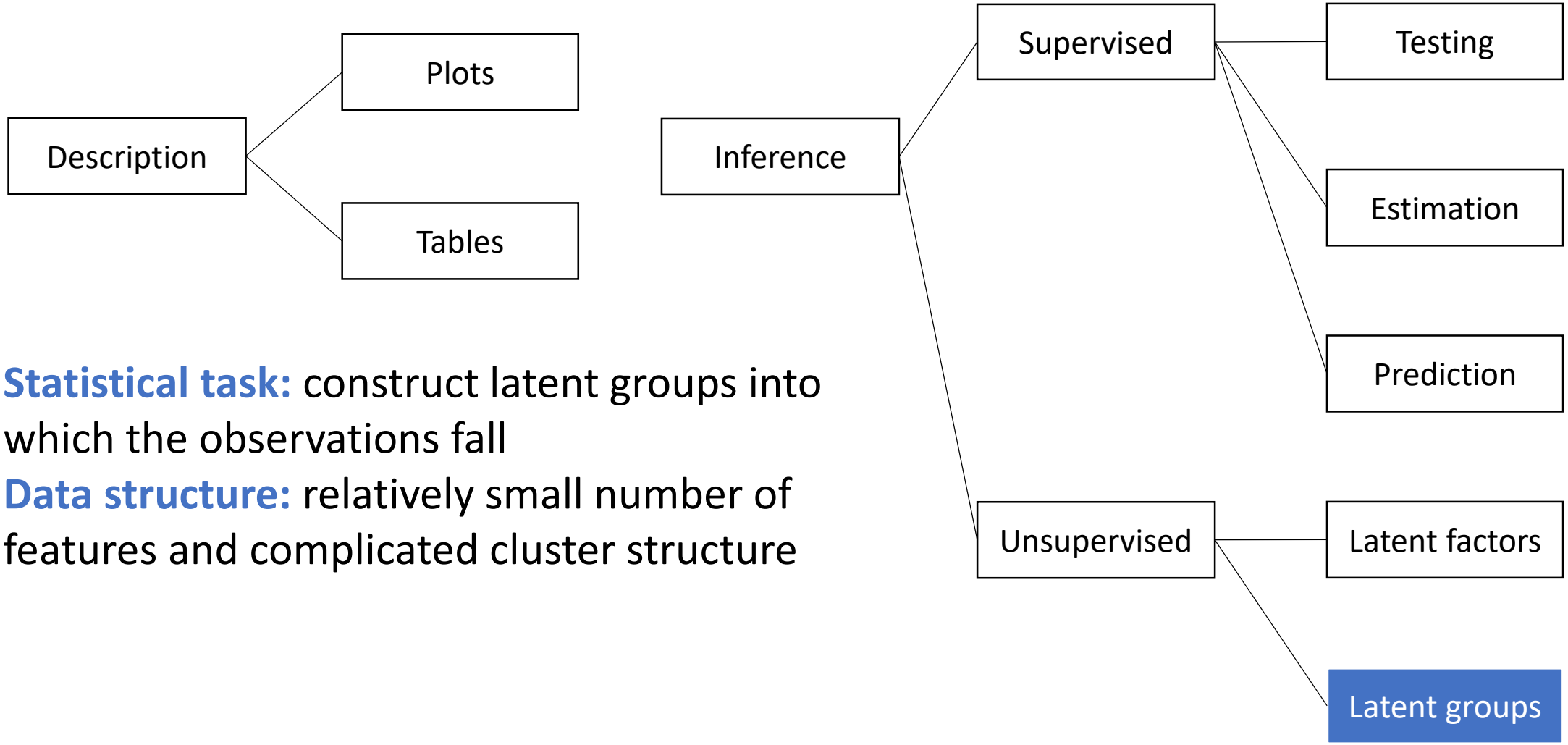
Canonical cell types, e.g., muscle and nerve, were originally defined by the functions of

Cellular Demographies, Recorded

Michael Elowitz
Caltech

It seems to me that we are at the beginning of a paradigm shift on the issue of cell type.

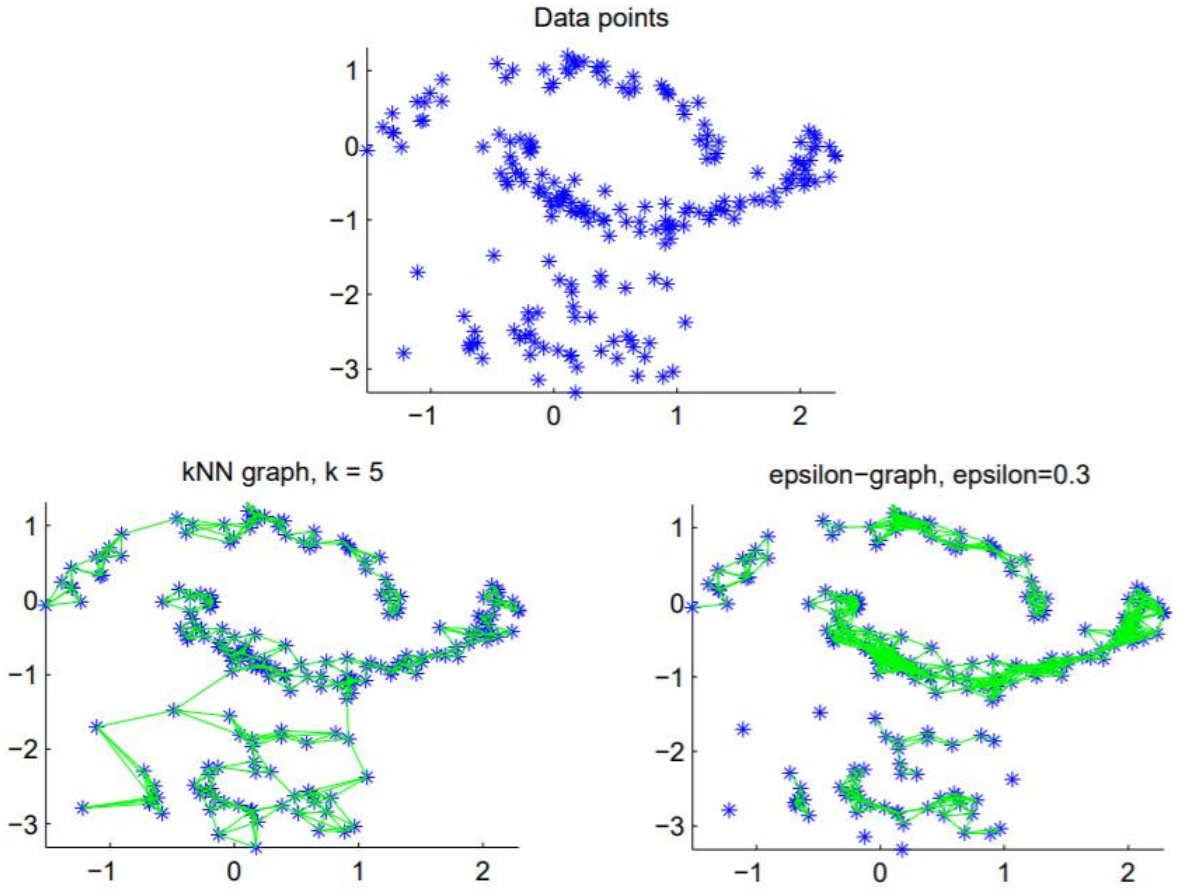
Graph clustering

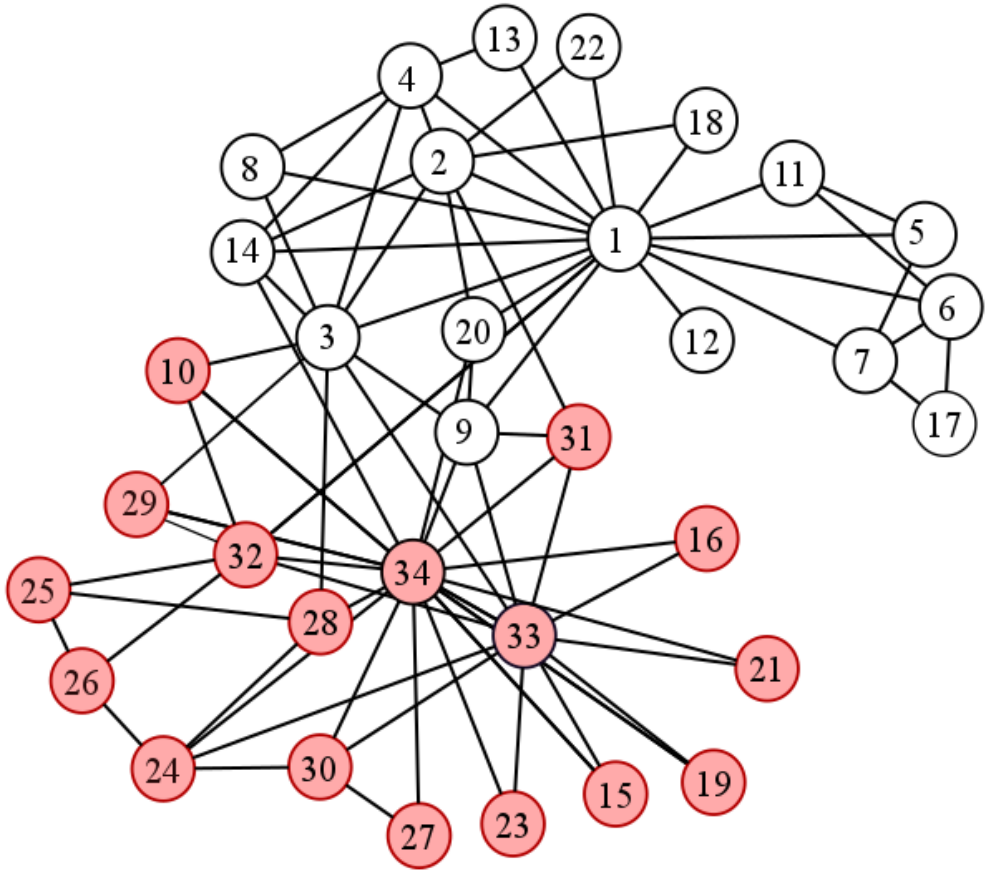


Statistical task: construct latent groups into which the observations fall

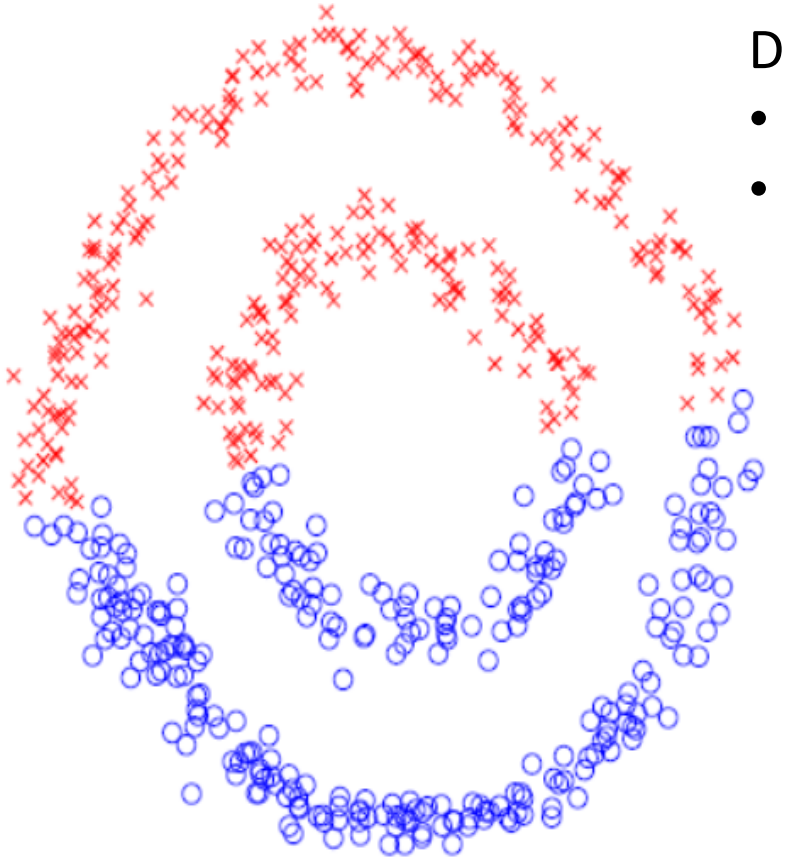
Data structure: relatively small number of features and complicated cluster structure

Graph clustering





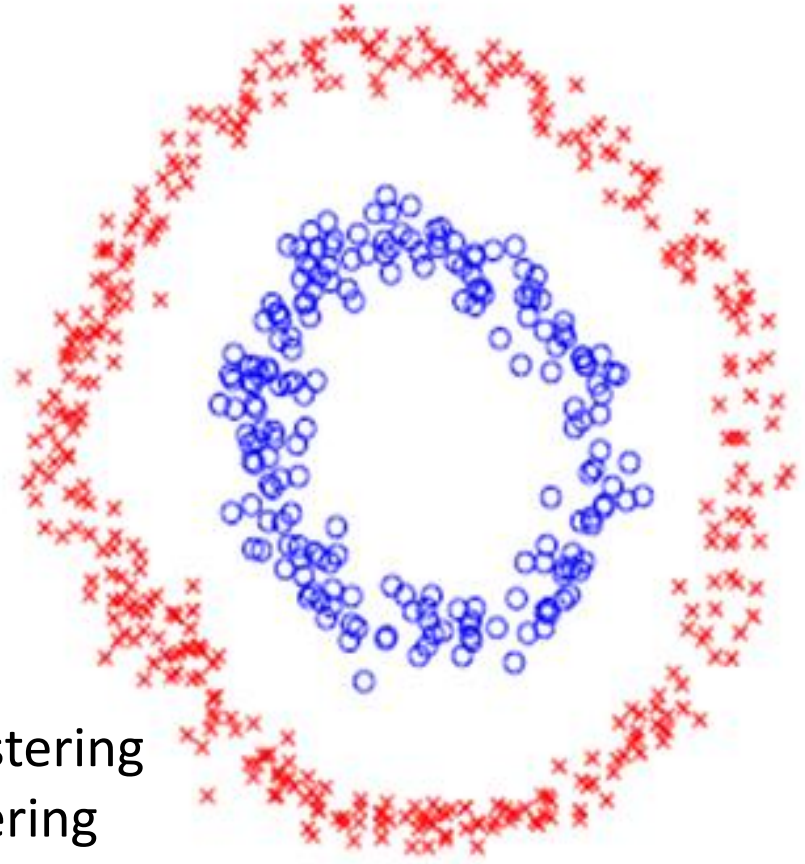
Comparison to other clustering methods



Distance

- Hierarchical clustering
- K-means clustering

vs.



Relationship

- Spectral clustering
- Graph clustering

http://webpages.uncc.edu/jfan/itcs4122.html

Graph clustering using Seurat

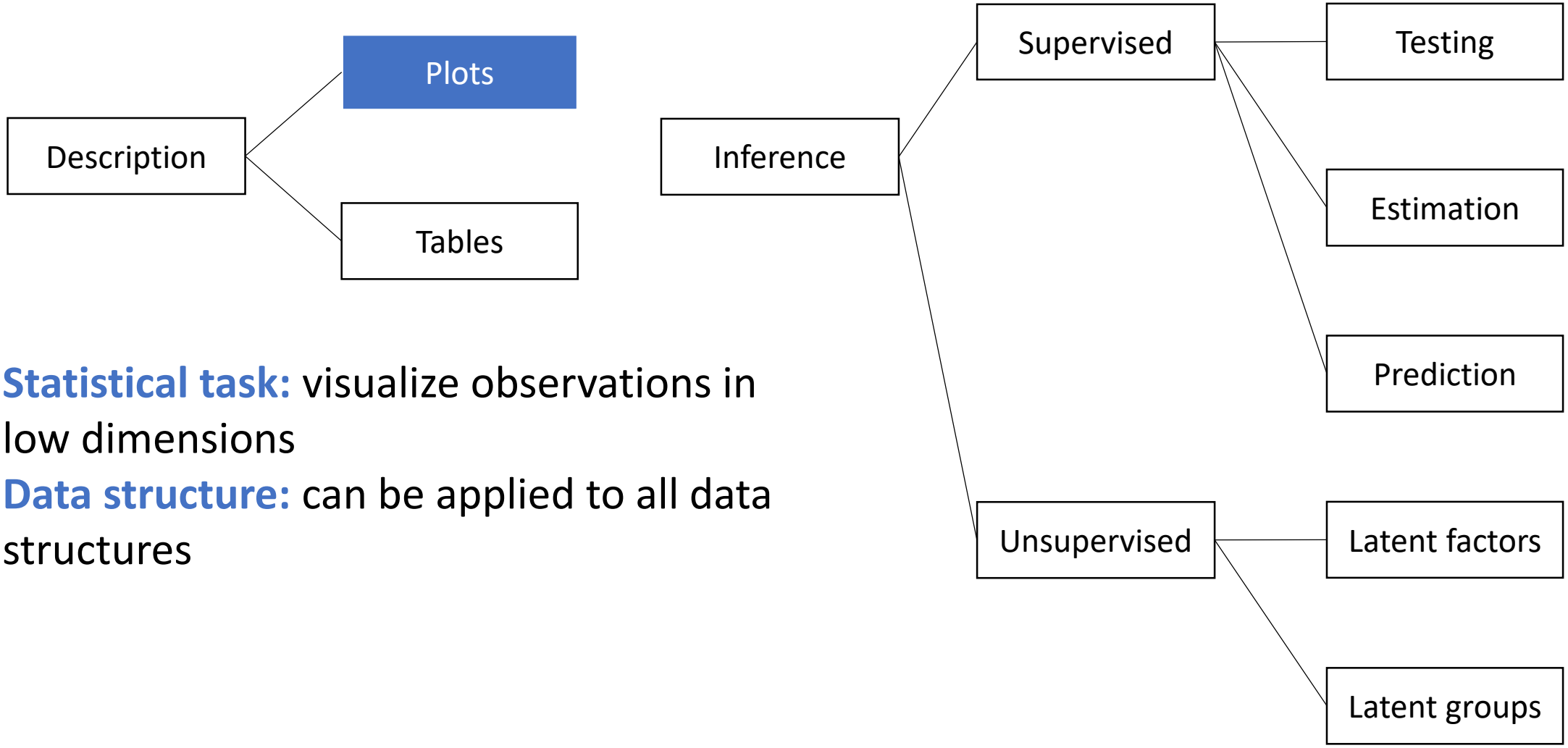
```
s_obj = FindNeighbors(s_obj, dims = 1:10)
s_obj = FindClusters(s_obj, dims = 1:10,
resolution = 0.1)
```

t-SNE plot

Research question:

How to visualize the different cell types?

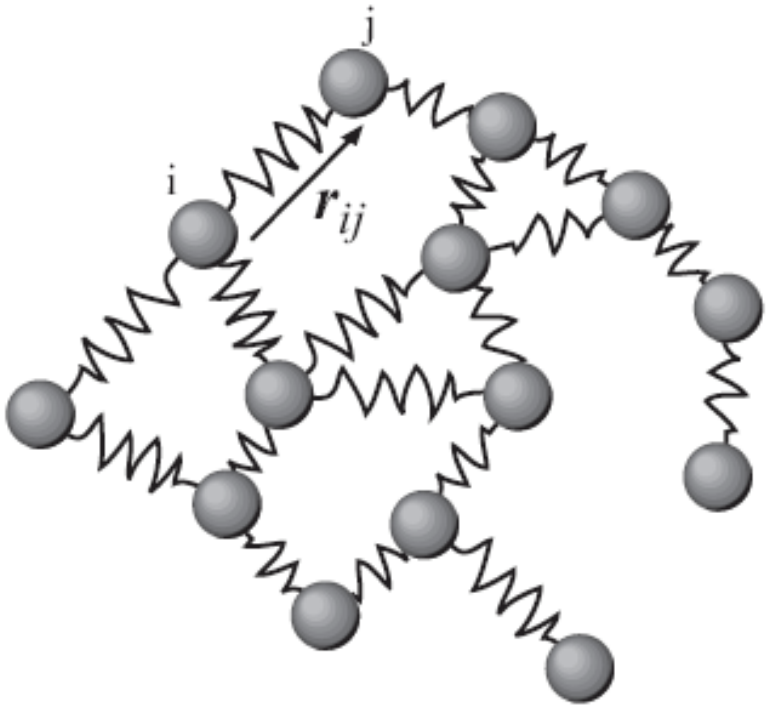
t-SNE plot

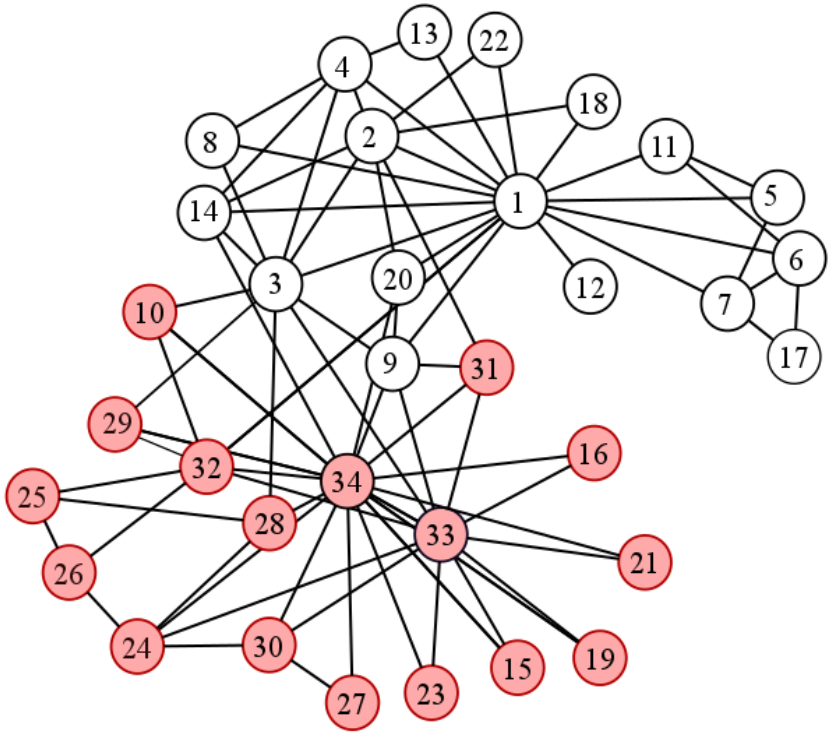


Statistical task: visualize observations in low dimensions

Data structure: can be applied to all data structures

t-SNE plot





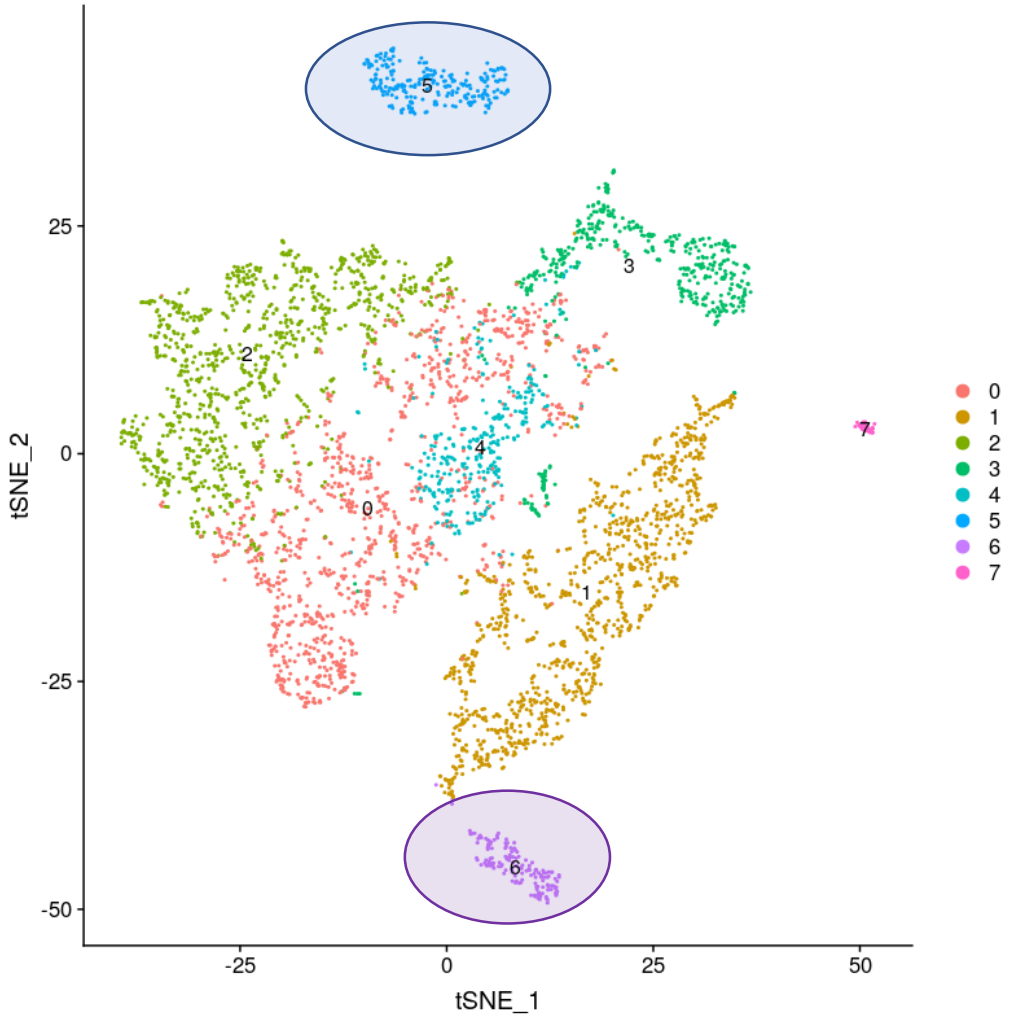
https://en.wikipedia.org/wiki/Spring_system

t-SNE plot

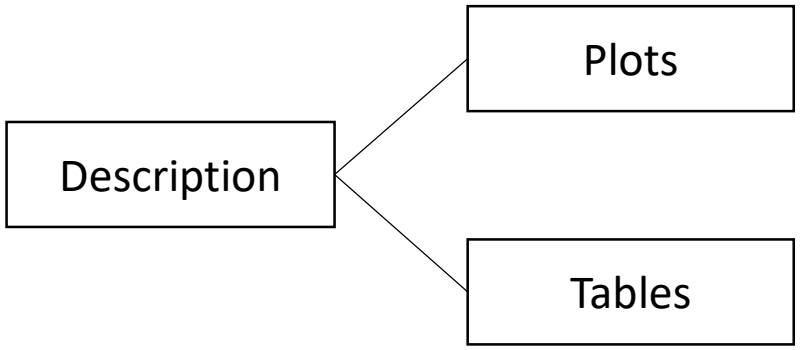
```
s_obj = RunTSNE(s_obj)
DimPlot(s_obj, reduction = "tsne", label = TRUE)
```

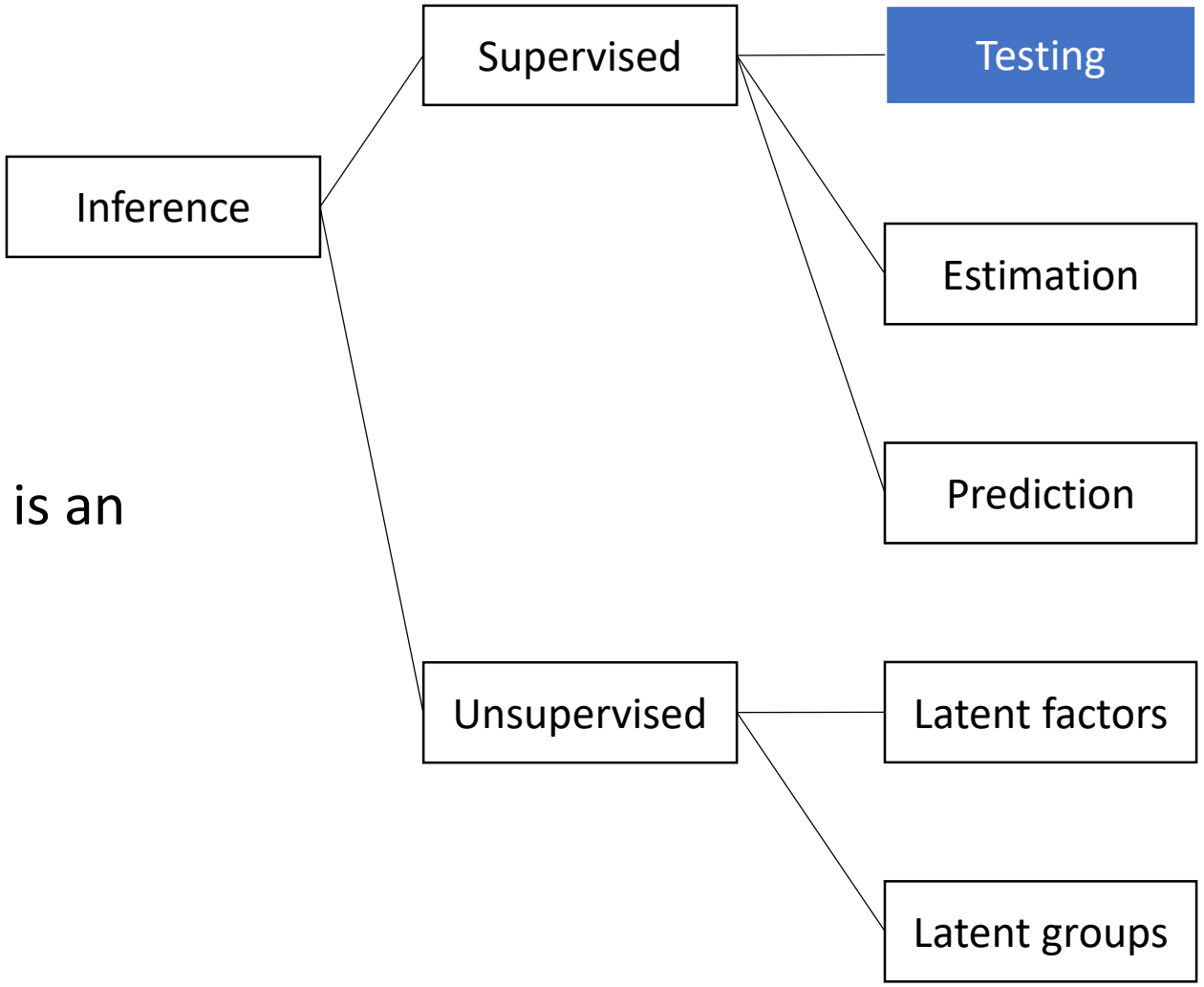
Wilcoxon test

Research question:
Does the expression of the gene PDYN differ between clusters 5 and 6?



Wilcoxon test





Statistical task: test whether there is an association between two variables

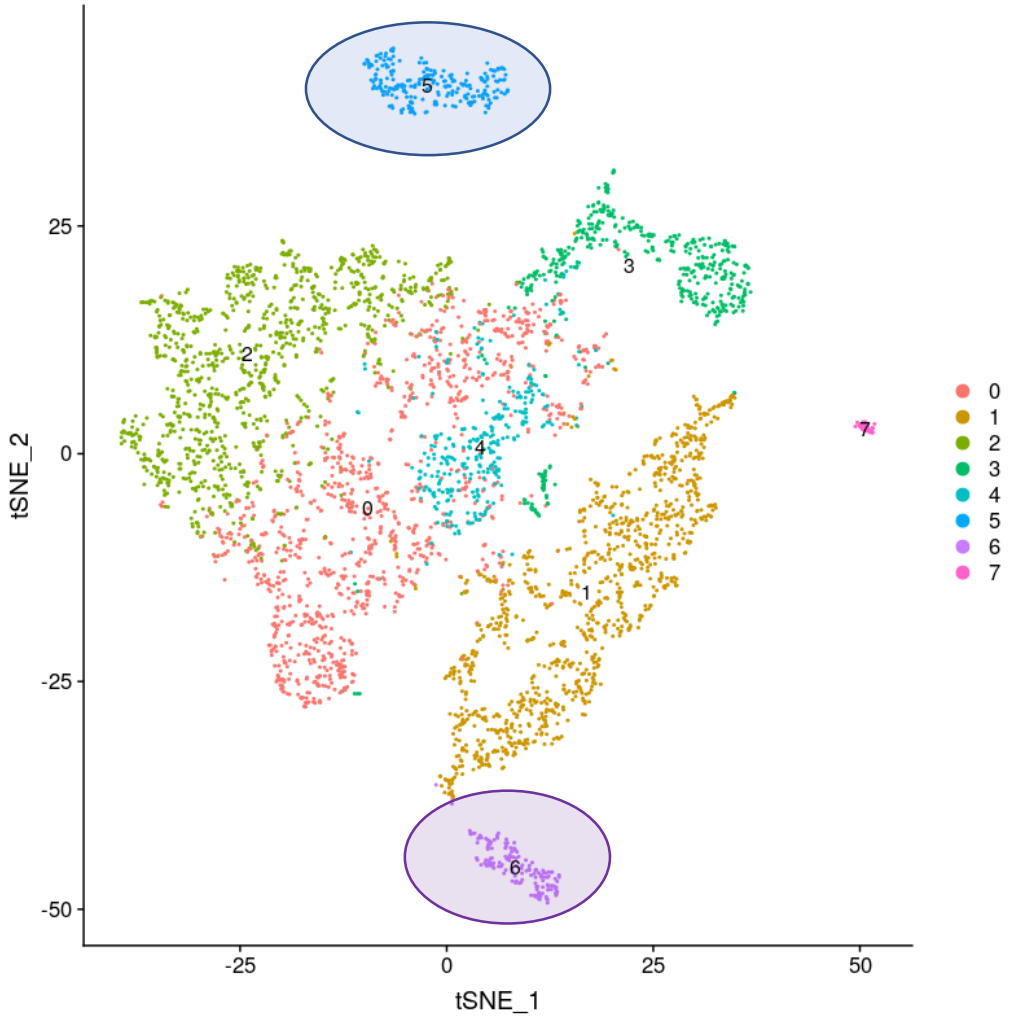
Data structure: one variable is dichotomous

Wilcoxon test using Seurat

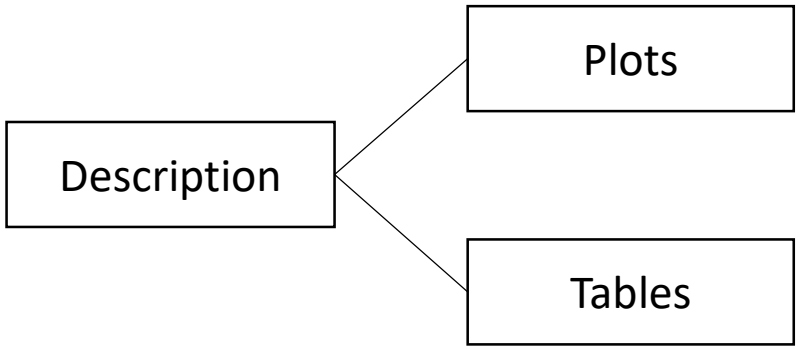
```
markers = FindMarkers(s_obj, ident.1 = 5,
ident.2 = 6)
markers["PDYN",]
```

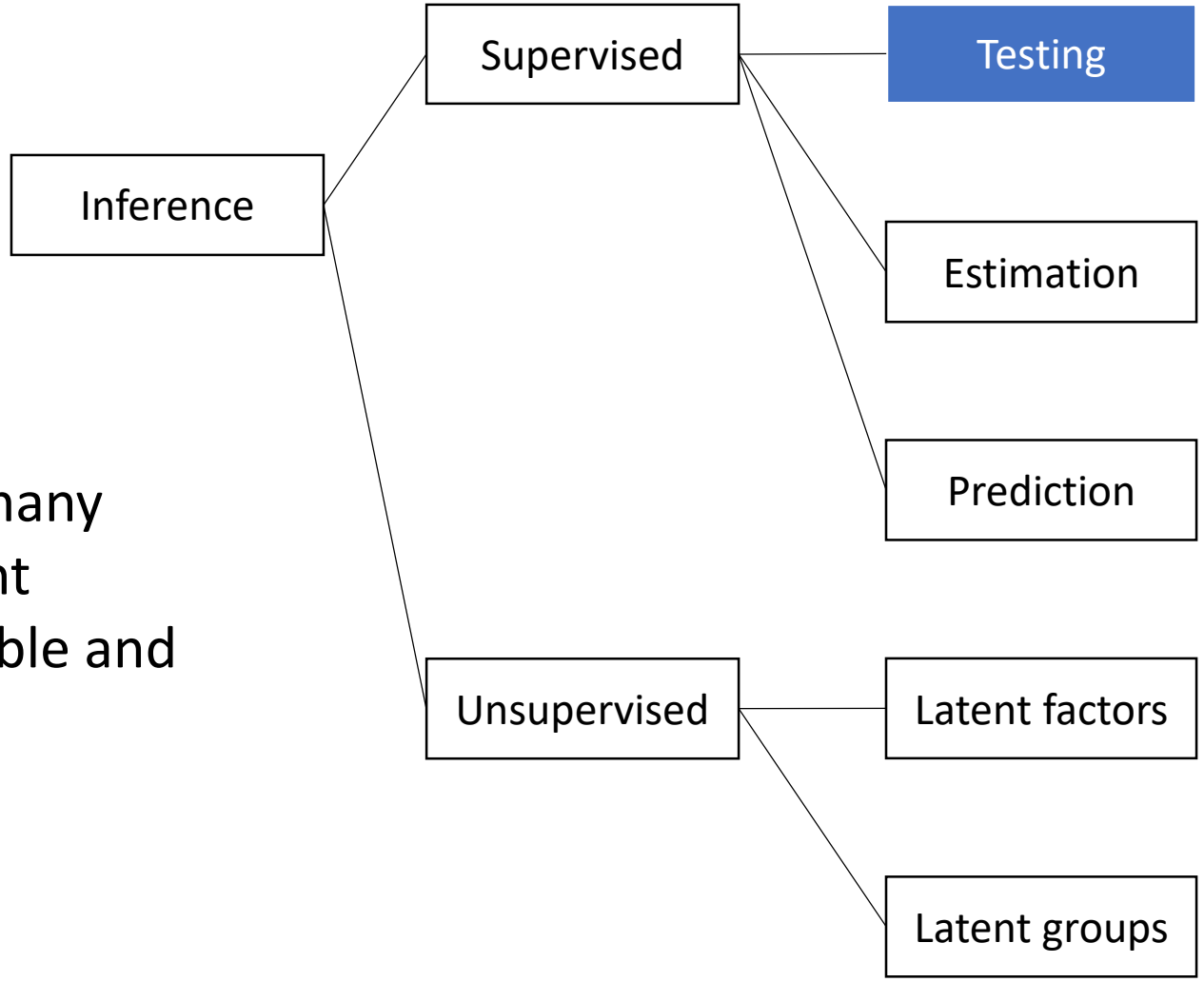
FDR control

Research question:
Which genes differ between clusters 5 and 6?



FDR control





Statistical task: identify which of many hypothesis tests are truly significant

Data structure: p-values are available and statistically independent

FDR control

- FDR = False Discovery Rate = expected value of

$$\frac{\#\ \text{false discoveries}}{\text{total}\ \#\ \text{of discoveries}}$$

- Statistical methods reject the largest number of hypothesis tests while maintaining FDR $\leq \alpha$, for some preset α

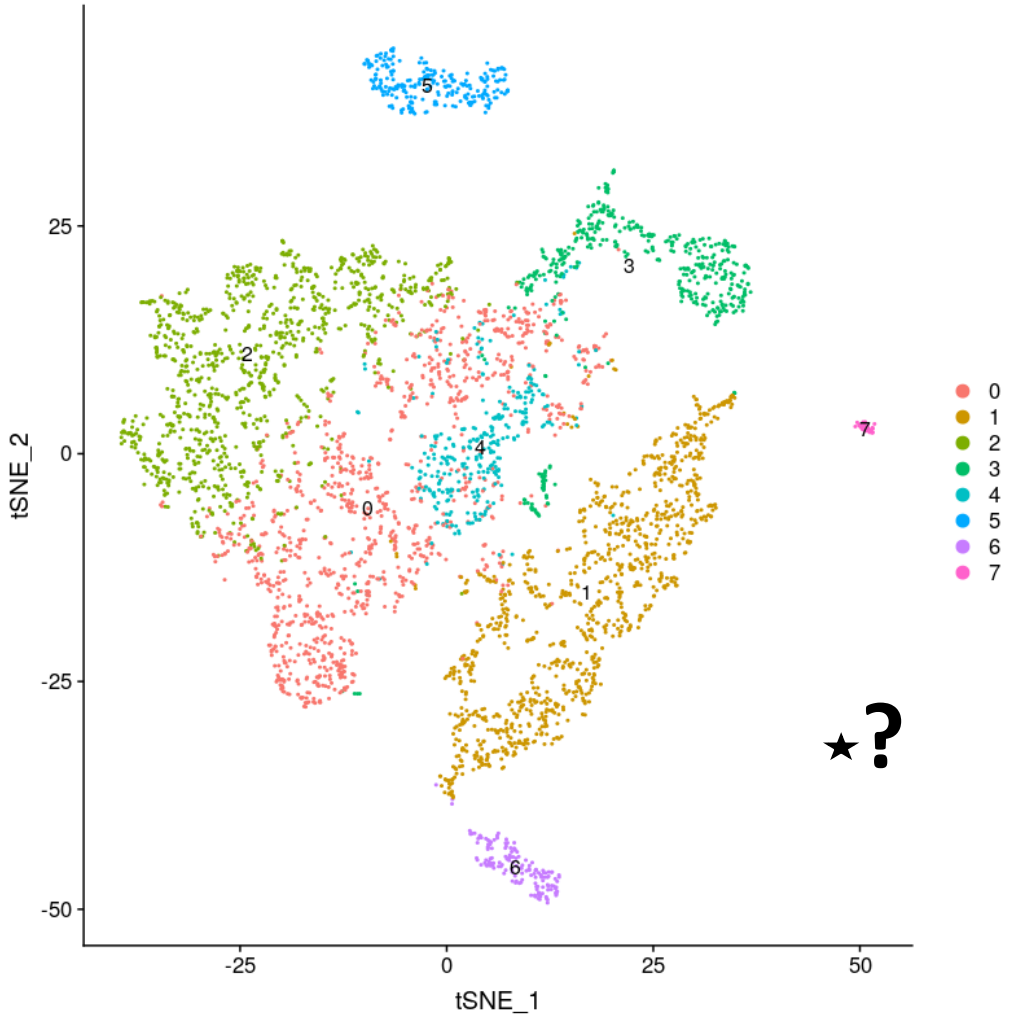
FDR control using Seurat

```
markers = FindMarkers(s_obj, ident.1 = 5,
ident.2 = 6)
head(markers)
sum(markers$p_val_adj <= 0.05)
```

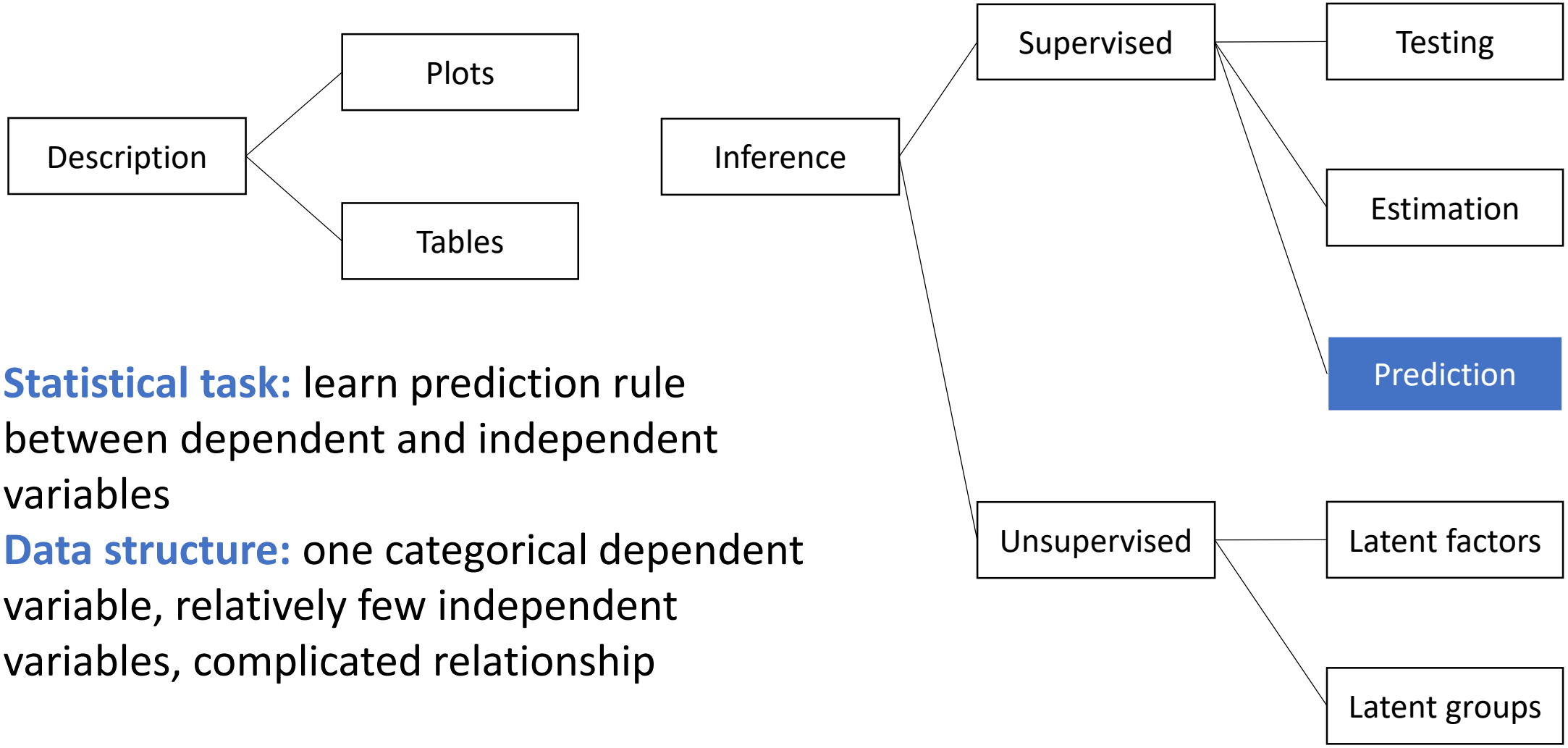
Random forest classification

Research question:

Given the principal components of the RNA-seq expression values of all genes from a new cell, how can we determine the cell's type?



Random forest classification

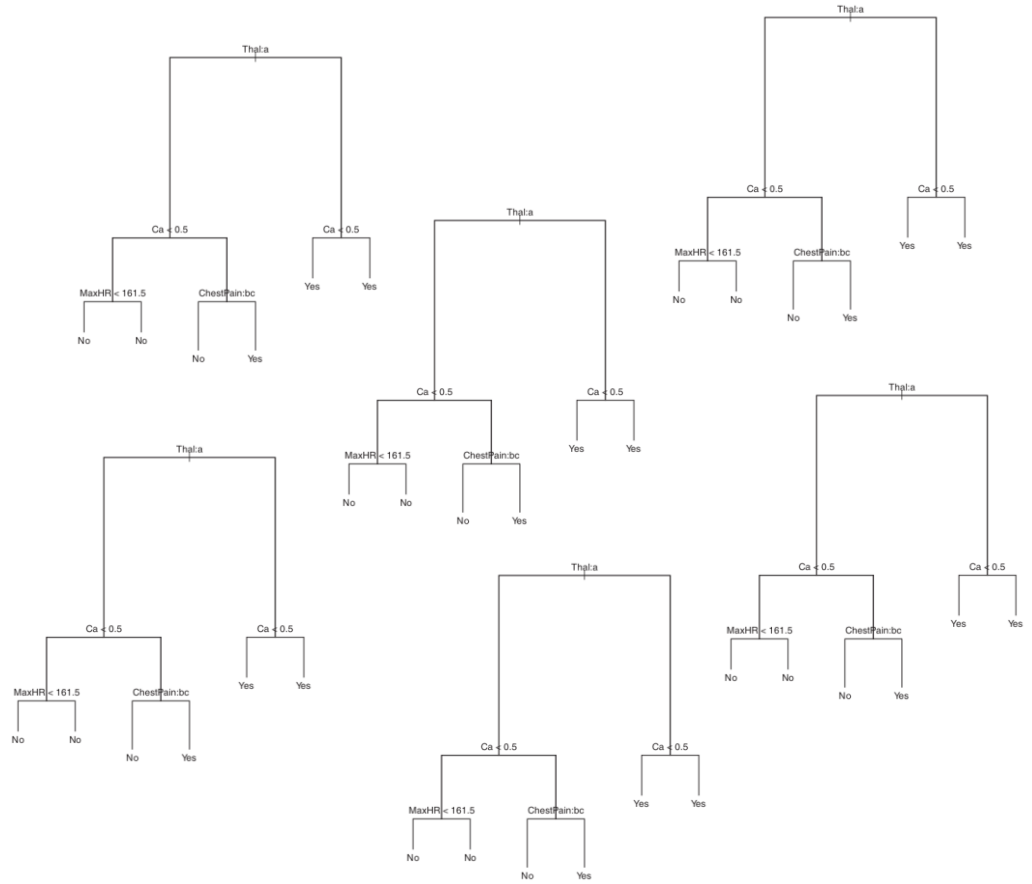


Statistical task: learn prediction rule between dependent and independent variables

Data structure: one categorical dependent variable, relatively few independent variables, complicated relationship

Random forest classification

Independent variables



Prediction

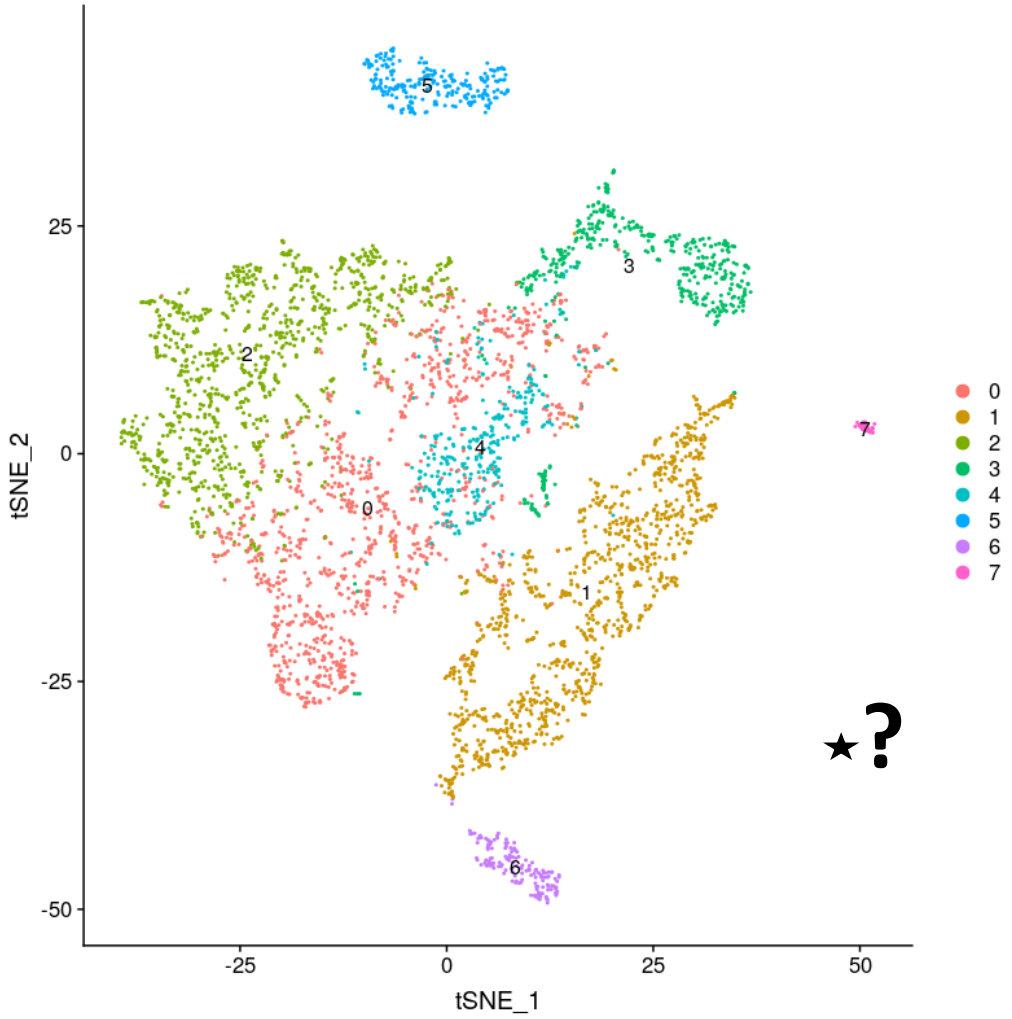
Random forest classification using caret and ranger

```
library(caret)
pcs = Embeddings(s_obj, reduction = "pca")[-(1:2), 1:10]
class = as.factor(Idents(s_obj))[-(1:2)]
dataset = data.frame(class, pcs)
rf_fit = train(class ~ .,
               data = dataset,
               method = "ranger",
               trControl = trainControl(method = "cv", number
= 3))
new_pcs = Embeddings(s_obj, reduction = "pca")[1:2, 1:10]
predict(rf_fit, new_pcs)
Idents(s_obj)[1:2]
```

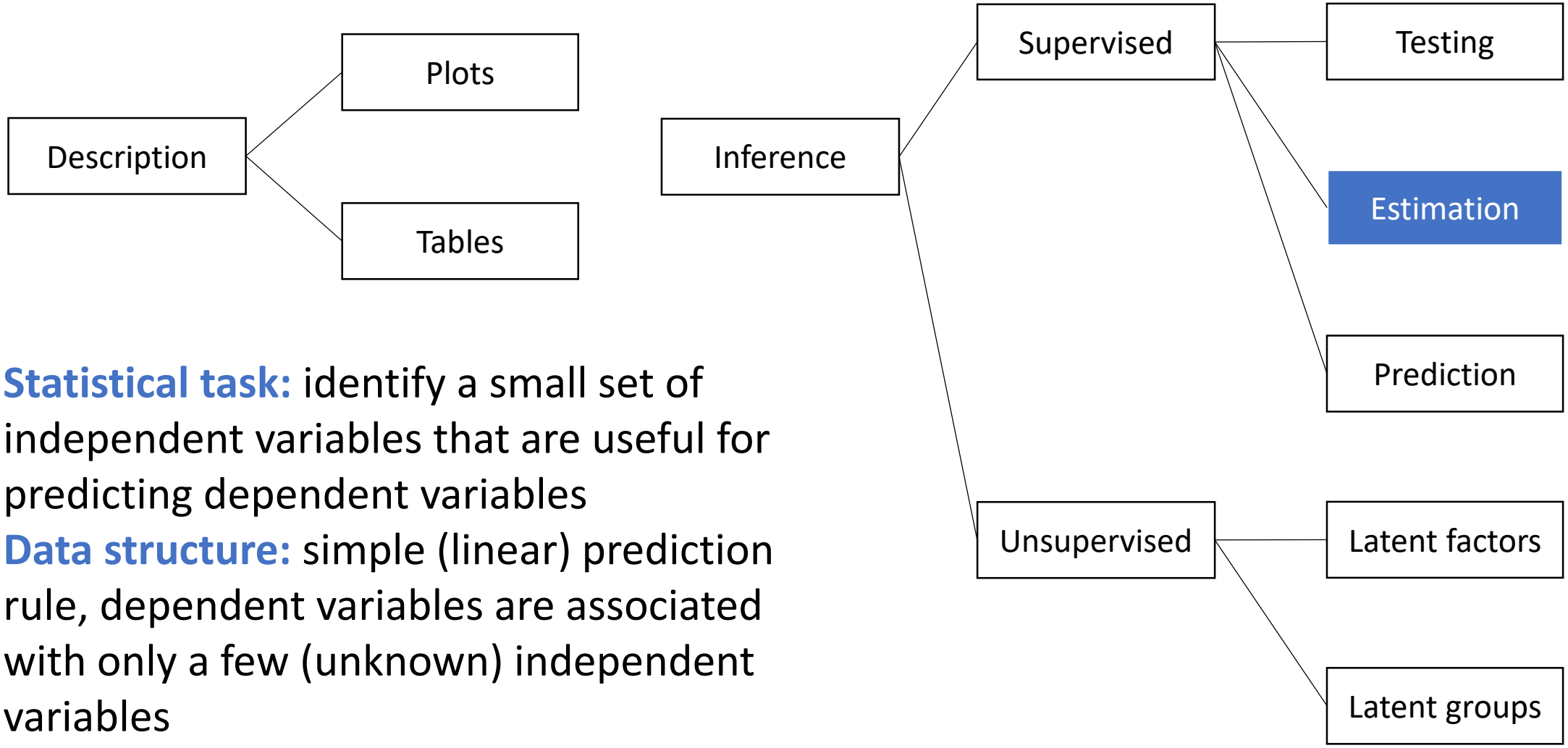
Lasso

Research question:

If we can only measure the expression of 10 genes in a new cell, which should we measure in order to most accurately predict the cell's type?



Lasso



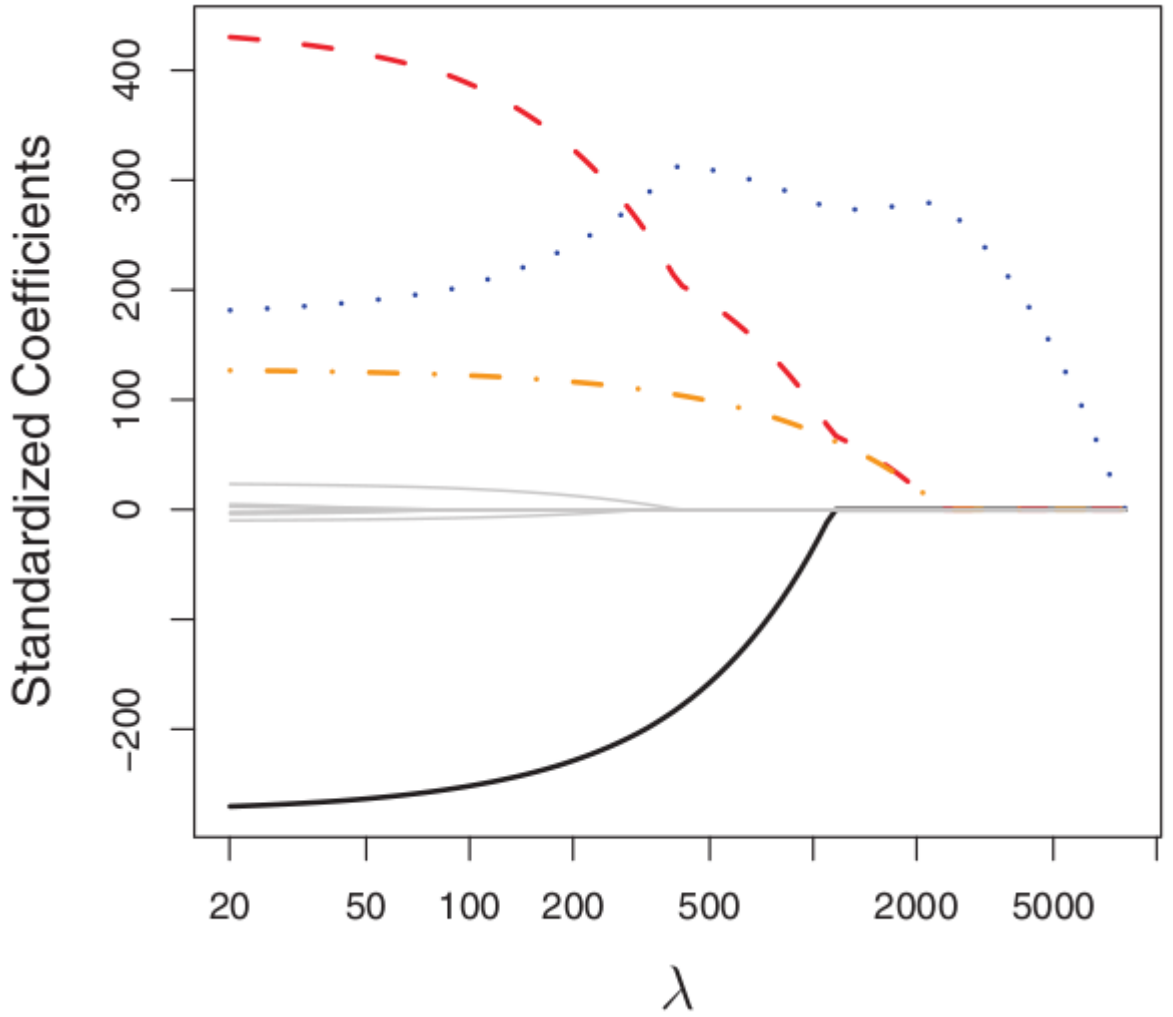
Statistical task: identify a small set of independent variables that are useful for predicting dependent variables

Data structure: simple (linear) prediction rule, dependent variables are associated with only a few (unknown) independent variables

Lasso

Estimates coefficients in the regression model

$$Y = \beta_0 + X_1\beta_1 + \cdots + X_p\beta_p$$

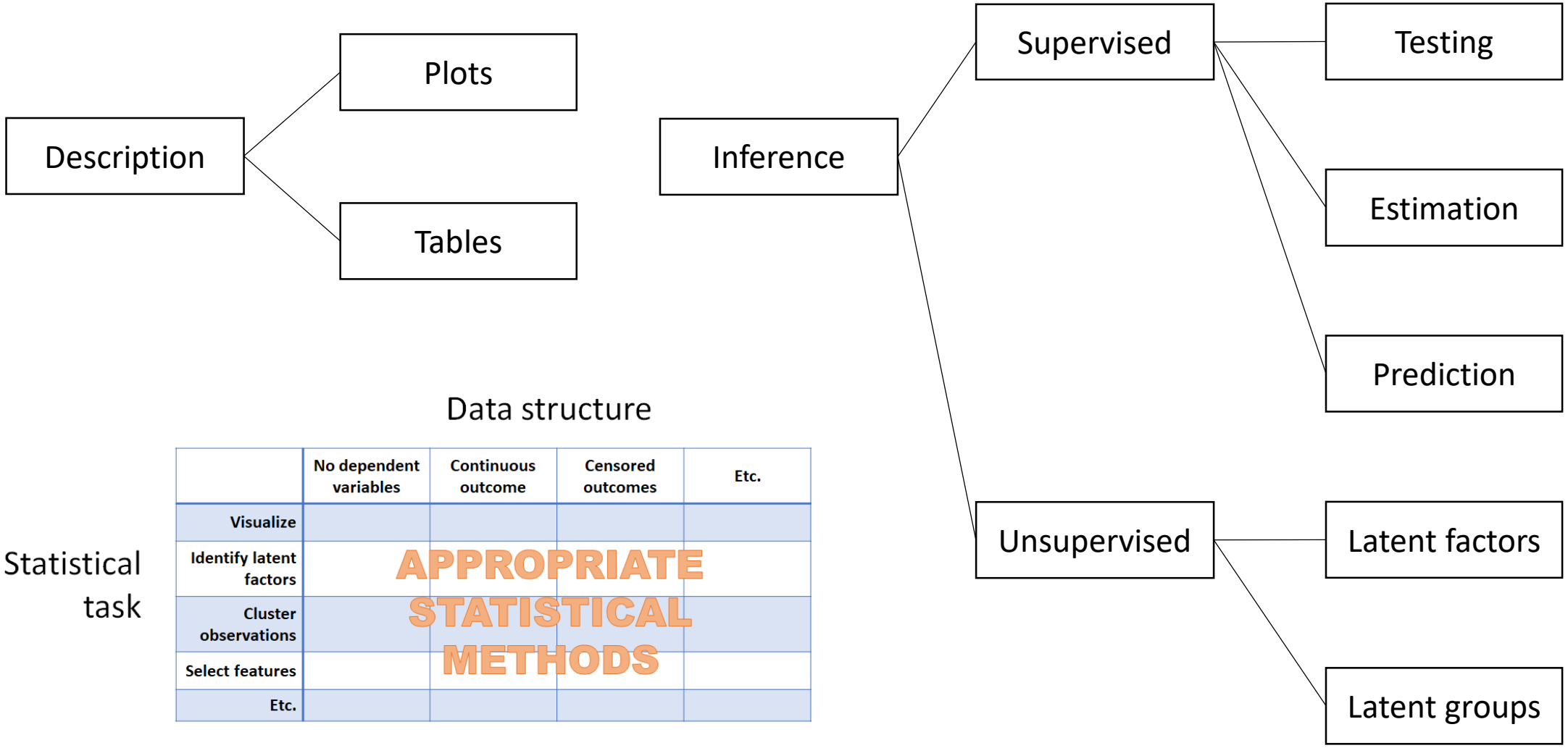


Lasso using glmnet

```
library(glmnet)
counts = t(GetAssayData(s_obj, slot = "counts"))[-(1:2),]
pcs = Embeddings(s_obj, reduction = "pca")[-(1:2), 1:10]
lasso = cv.glmnet(counts, pcs, family = "mgaussian", nfolds = 3)
lambda = min(lasso$lambda[lasso$nzero <= 10])
coefs = coef(lasso, s = lambda)
rownames(coefs$PC_1)[which(coefs$PC_1 != 0)]
new_counts = t(GetAssayData(s_obj, slot = "counts"))[1:2,]
new_pcs = predict(lasso, newx = new_counts, s = lambda)[,, 1]
predict(rf_fit, new_pcs)
Idents(s_obj)[1:2]
```

Conclusions

Statistical toolbox



Data structure

Statistical task

	No dependent variables	Continuous outcome	Censored outcomes	Etc.
Visualize				
Identify latent factors				
Cluster observations				
Select features				
Etc.				

APPROPRIATE STATISTICAL METHODS

Statistical tools

1. PCA
2. Graph clustering
3. t-SNE plot
4. Wilcoxon test
5. FDR control
6. Random forest classification
7. Lasso

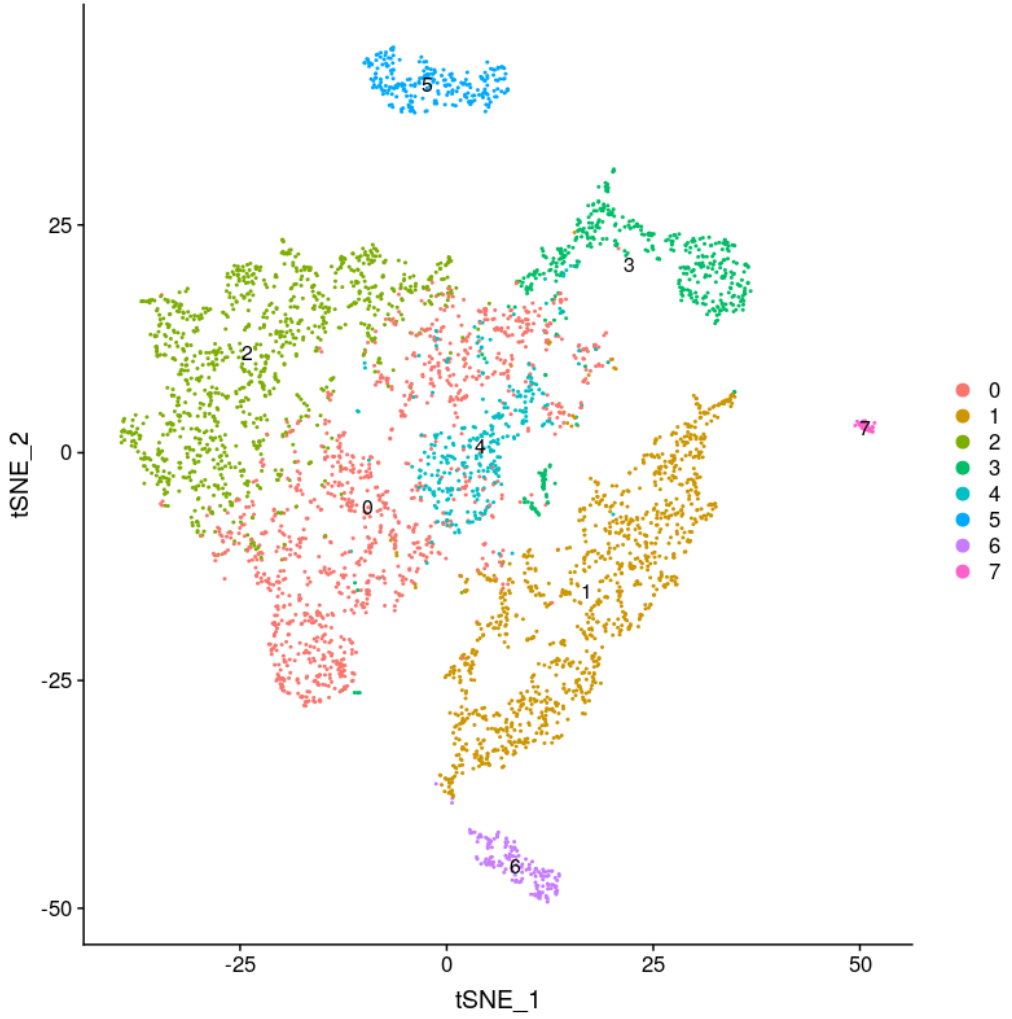
Can be applied to single-cell RNA-seq and beyond

CellPress

Current Biology
Article

Comprehensive Identification and Spatial Mapping of Habenular Neuronal Types Using Single-Cell RNA-Seq

Shristi Pandey,[1,*] Karthik Shekhar,[2] Aviv Regev,[2,3] and Alexander F. Schier[1,2,4,5,6,7,*]
[1]Department of Molecular and Cellular Biology, Harvard University, Cambridge, MA 02138, USA
[2]Klarman Cell Observatory, Broad Institute of MIT and Harvard, 7 Cambridge Center, Cambridge, MA 02142, USA
[3]Howard Hughes Medical Institute and Koch Institute of Integrative Cancer Research Department of Biology, Massachusetts Institute of Technology, Cambridge, MA 02140, USA
[4]Center for Brain Science, Harvard University, 52 Oxford Street, Cambridge, MA 02138, USA
[5]Biozentrum, University of Basel, Basel, Switzerland
[6]Allen Discovery Center for Cell Lineage Tracing, University of Washington, Seattle, WA 98195, USA
[7]Lead Contact
*Correspondence: p.shristi@gmail.com (S.P.), schier@fas.harvard.edu (A.F.S.)
https://doi.org/10.1016/j.cub.2018.02.040



To learn more

- Take systematic courses in basic statistics, statistical learning, and R/python
- Study recently published papers in your field of interest that use your technology of interest
- Consult tutorials, workshops, lab mates, and Google

Thank you

NIH Big Data to Knowledge (BD2K)
MAYO CLINIC
IGB
NSF
DMS-1613005