

# VisRseq: R-based visual analytics software for sequencing data

Hamid Younesy

*Graphics Usability and Visualization, Simon Fraser University, Canada*

Torsten Möller

*Visualization and Data Analysis Group, University of Vienna, Austria*

Matthew C. Lorincz

*Department of Medical Genetics, The University of British Columbia, Canada*

Mohammad M. Karimi

*Biomedical Research Centre, The University of British Columbia, Canada*

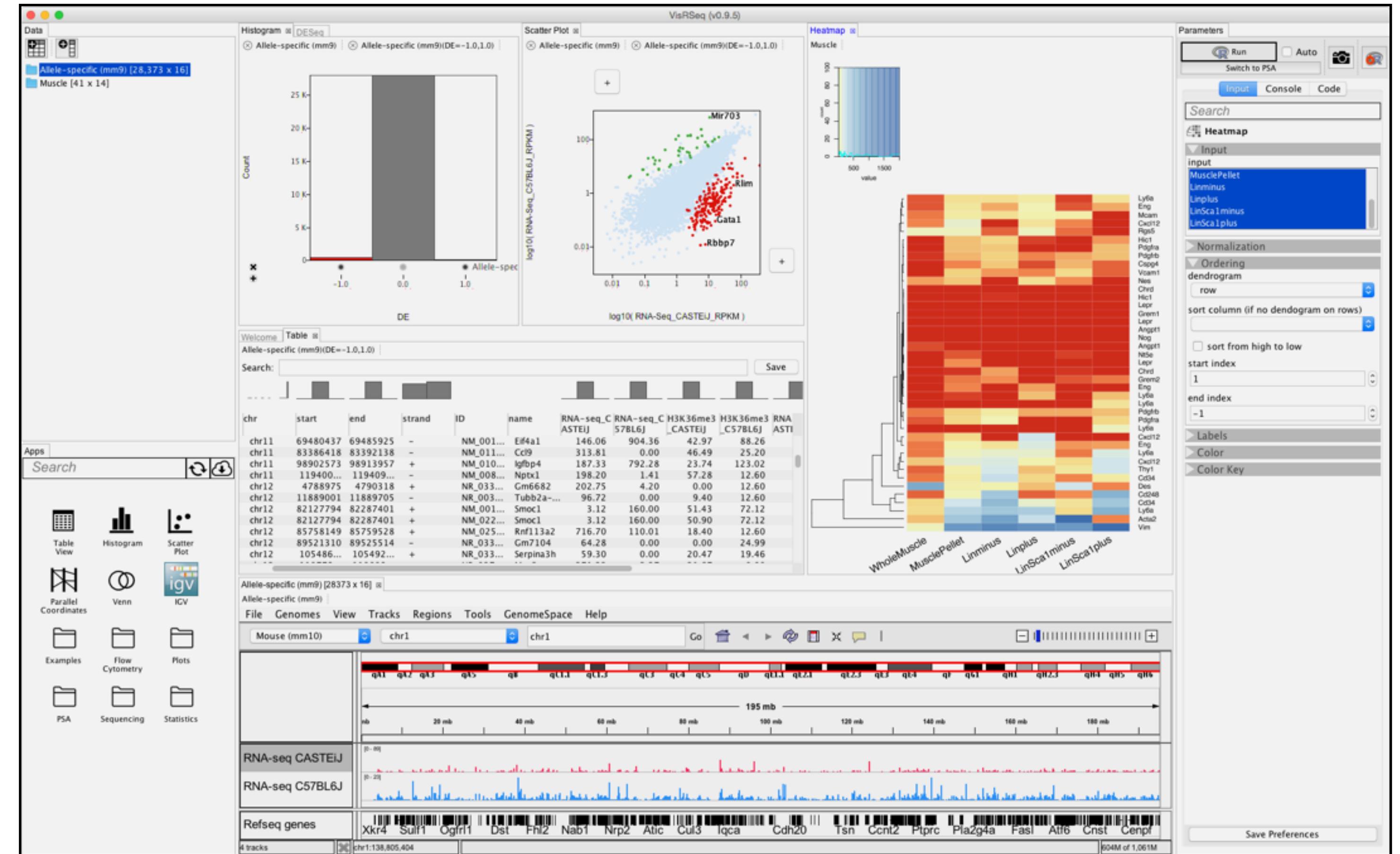
Steven J.M. Jones

*Genome Sciences Centre, BC Cancer Agency, Canada*



# Overview

- introduction
- motivation
- VisRseq
- case study
- future works



# motivation

# motivation

analysis software

# motivation

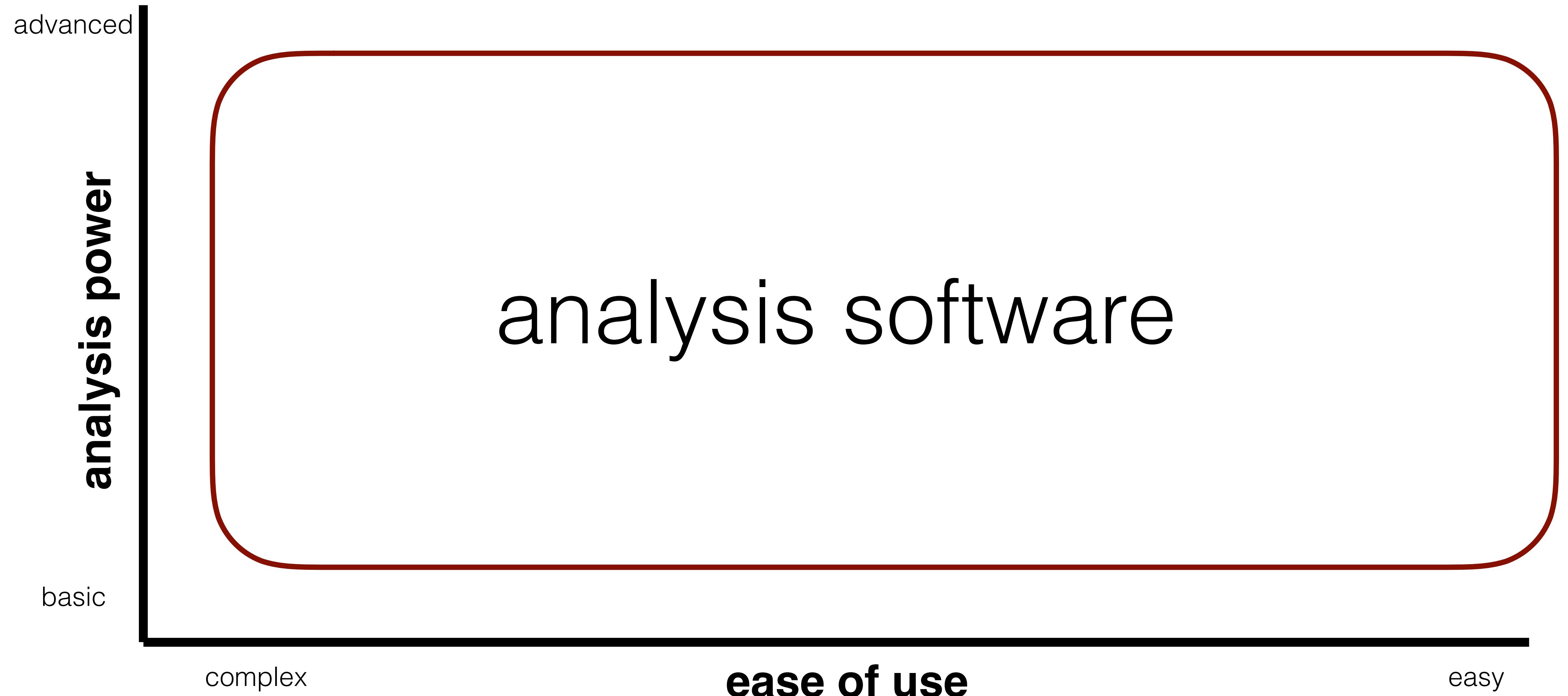
analysis software

complex

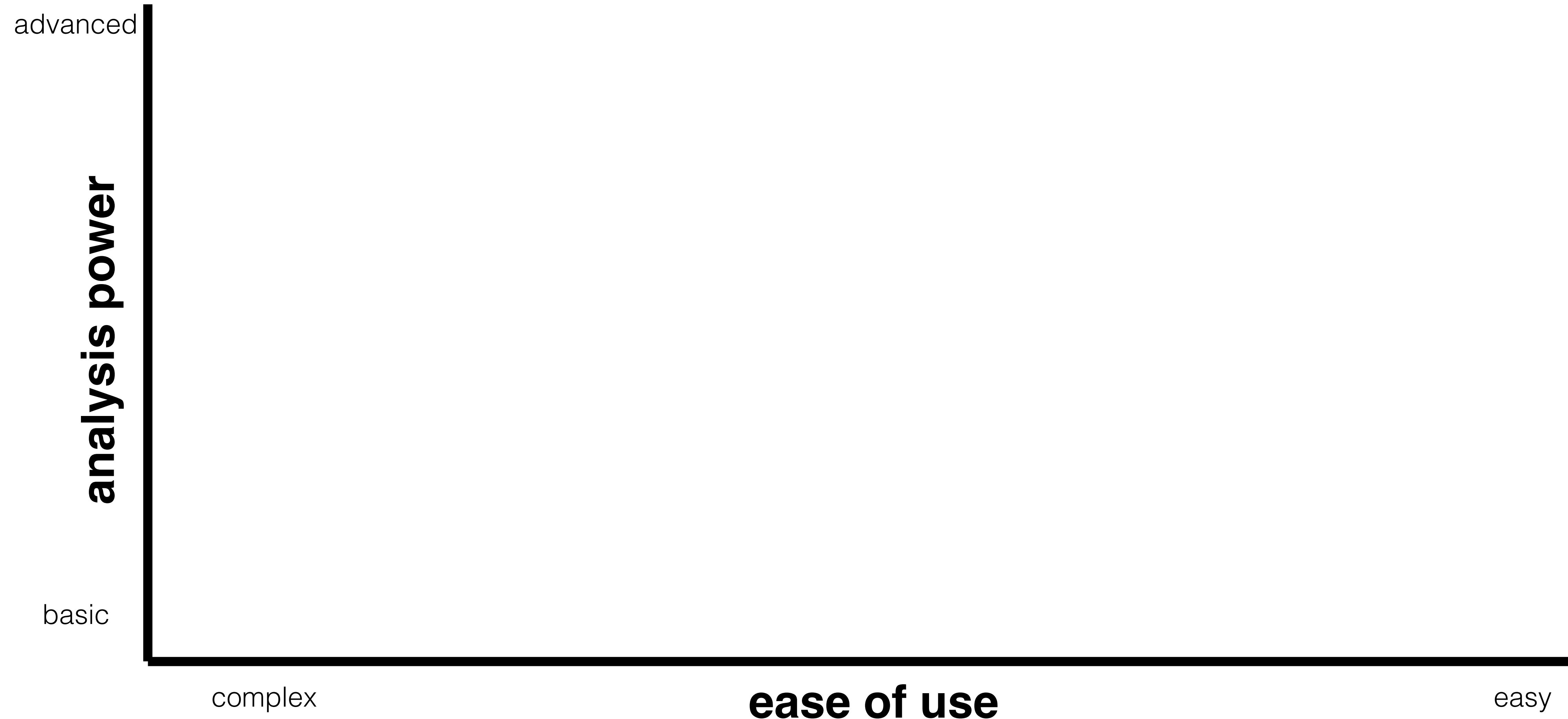
**ease of use**

easy

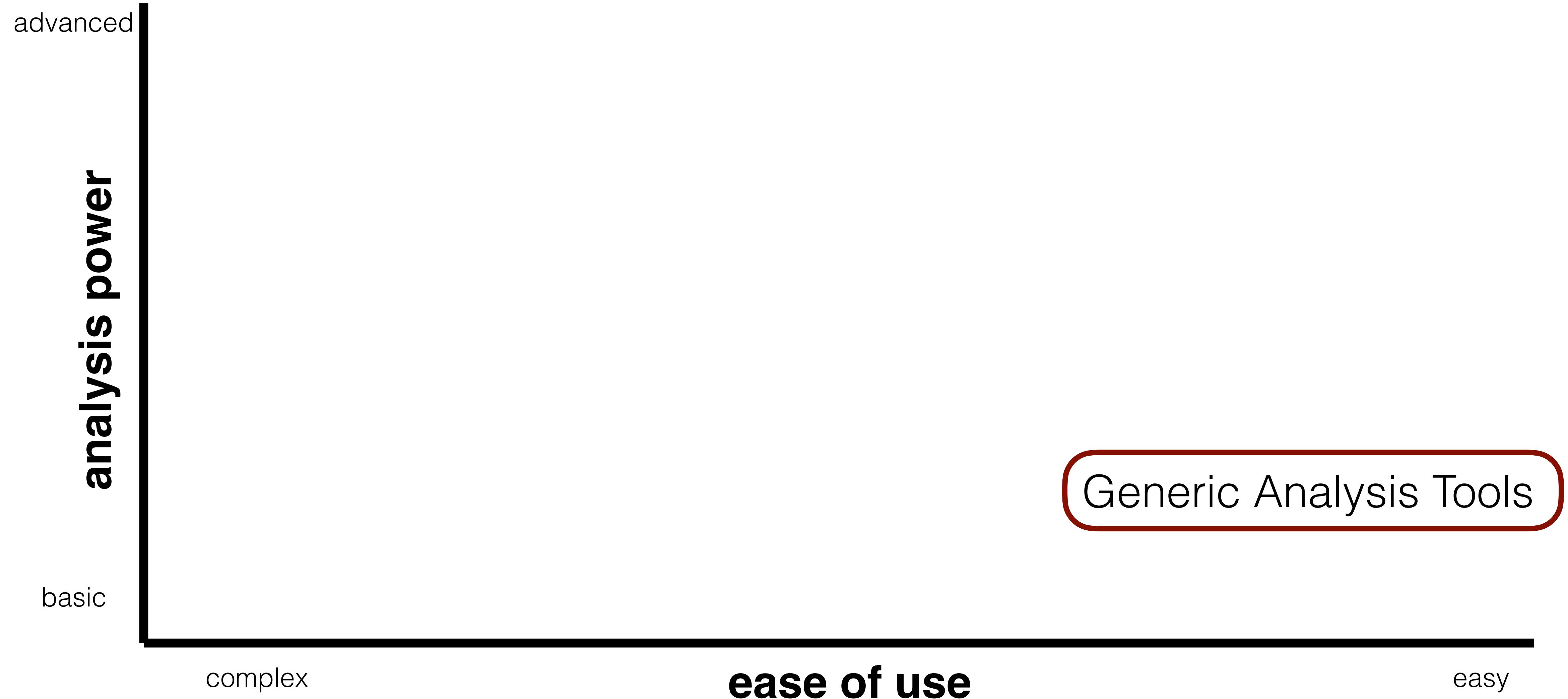
# motivation



# motivation



# motivation



# motivation

advanced

analysis power

basic

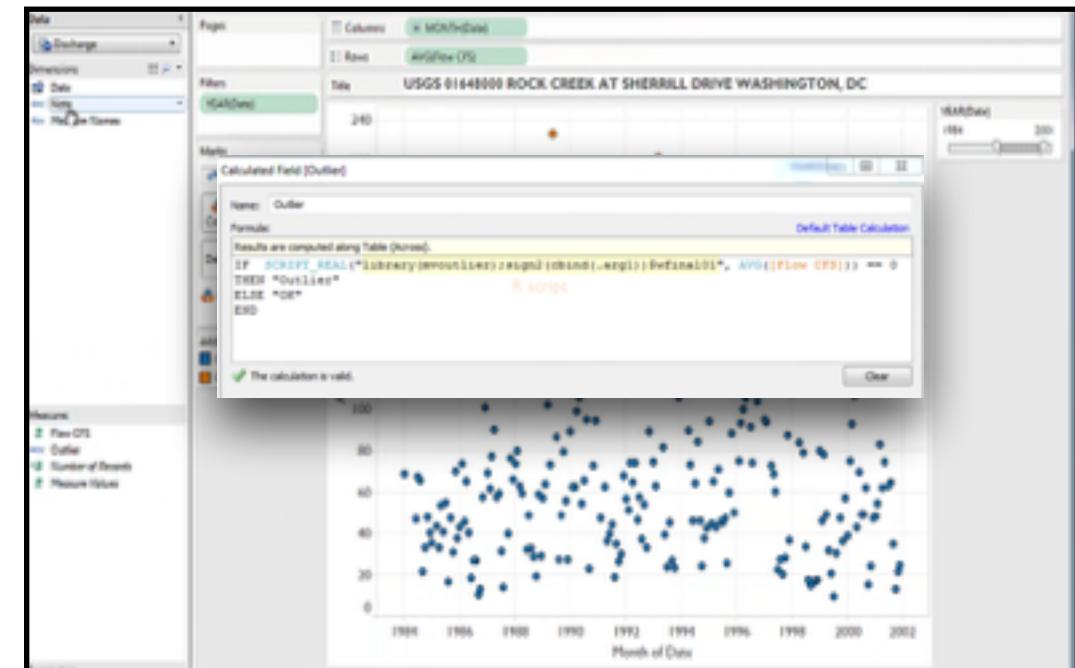
complex

ease of use

easy

Category	Term	Fold Enrichr	PValue	Count	%
GOTERM_BP_GO_0006412	7.16288483	2.85E-19	translation	33	
GOTERM_BP_GO_0034645	2.58363459	1.37E-09	cellular macromolecule biosynthetic proc	42	
GOTERM_BP_GO_0009059	2.56246481	1.77E-09	macromolecule biosynthetic process	42	
GOTERM_BP_GO_0010467	2.28422179	2.51E-07	gene expression	39	
GOTERM_BP_GO_0044267	2.09523711	3.58E-06	cellular protein metabolic process	38	
GOTERM_BP_GO_0019538	1.95086485	4.53E-06	protein metabolic process	42	
GOTERM_BP_GO_0044249	1.78235807	8.96E-06	cellular biosynthetic process	47	
GOTERM_BP_GO_0009058	1.70532395	3.13E-05	biosynthetic process	47	
GOTERM_BP_GO_0030199	29.7840112	3.04E-04	collagen fibril organization	4	
GOTERM_BP_GO_0030198	18.8111342	4.50E-04	extracellular matrix organization	5	
GOTERM_BP_GO_0022900	5.39395479	6.68E-04	electron transport chain	8	
GOTERM_BP_GO_0043062	8.15514593	0.00319551	extracellular structure organization	5	
GOTERM_BP_GO_0022904	12.0181098	0.00415644	respiratory electron transport chain	4	
GOTERM_BP_GO_0043170	1.33494519	0.00590397	macromolecule metabolic process	52	
GOTERM_BP_GO_0044237	1.22562383	0.00676271	cellular metabolic process	67	
GOTERM_BP_GO_0044260	1.36054073	0.00680266	cellular macromolecule metabolic proces	48	
GOTERM_BP_GO_0045333	4.82417083	0.00783823	cellular respiration	6	
GOTERM_BP_GO_0051216	19.7605459	0.00980216	cartilage development	3	
GOTERM_BP_GO_0008292	171.258065	0.0115197	acetylcholine biosynthetic process	2	
GOTERM_BP_GO_0009102	5.63348896	0.01164616	glycoprotein metabolic process	5	
GOTERM_BP_GO_0009987	1.1327528	0.01372666	cellular process	80	
GOTERM_BP_GO_0006091	2.59875667	0.01379198	generation of precursor metabolites and i	10	
GOTERM_BP_GO_0032501	1.70689101	0.01607076	multicellular organismal process	21	
GOTERM_BP_GO_0042136	85.6290323	0.02290813	neurotransmitter biosynthetic process	2	
GOTERM_BP_GO_0008291	85.6290323	0.02290813	acetylcholine metabolic process	2	

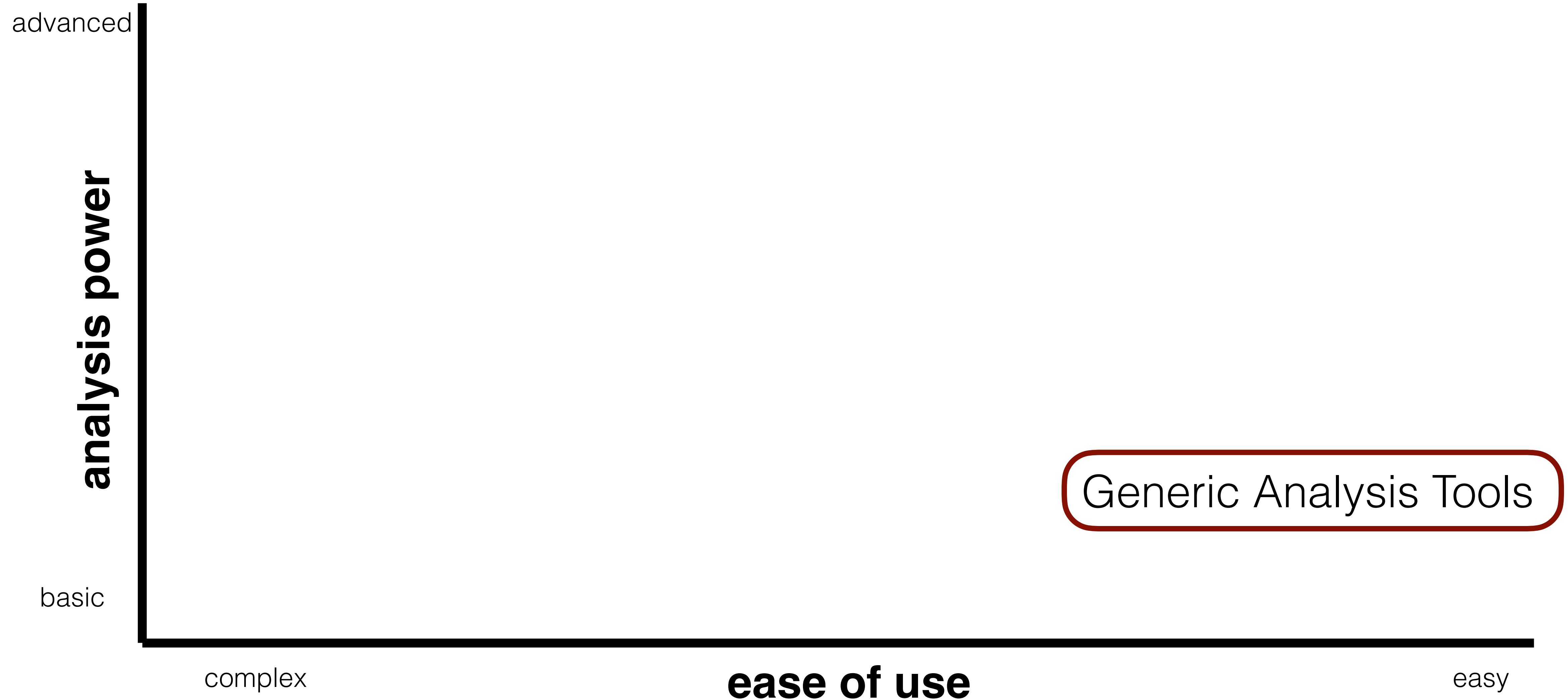
Excel



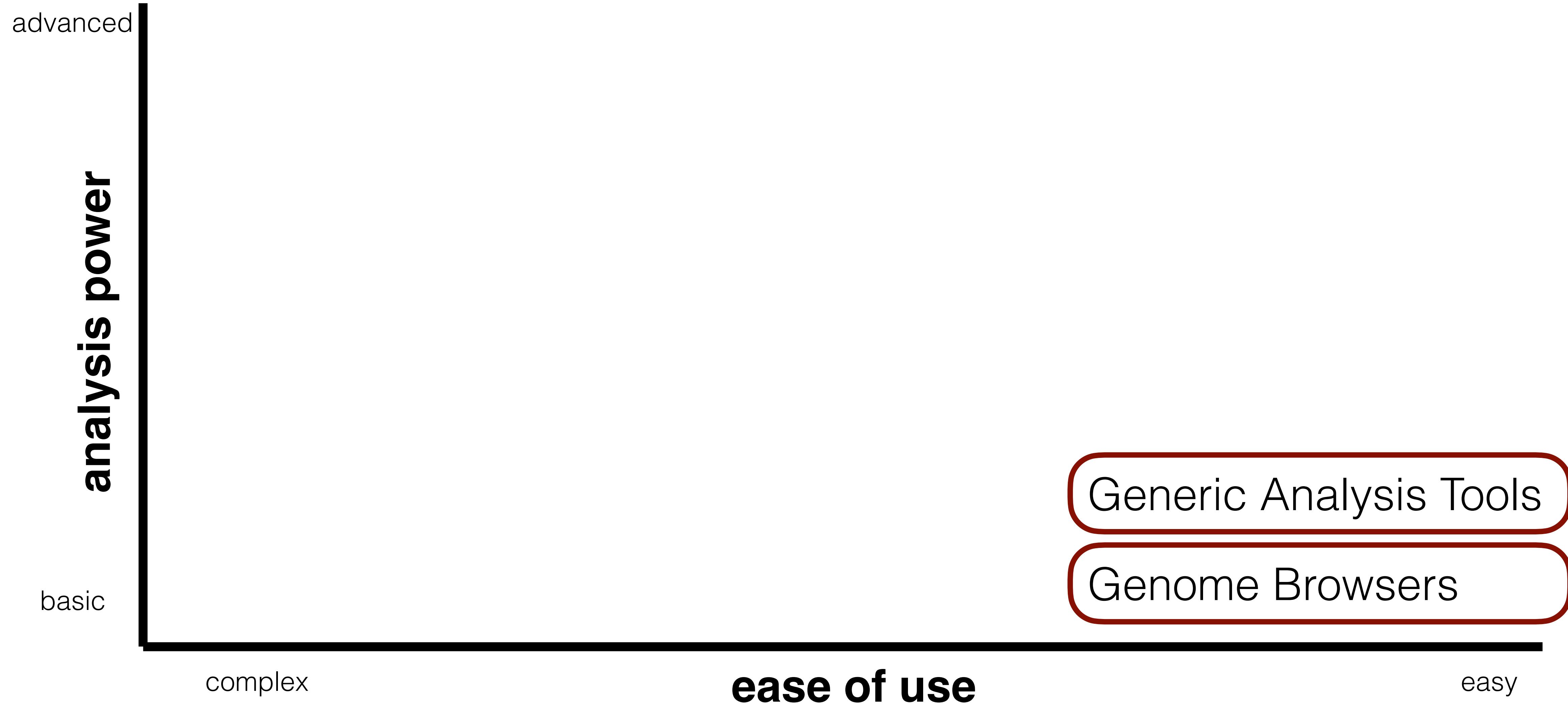
Tableau

Generic Analysis Tools

# motivation

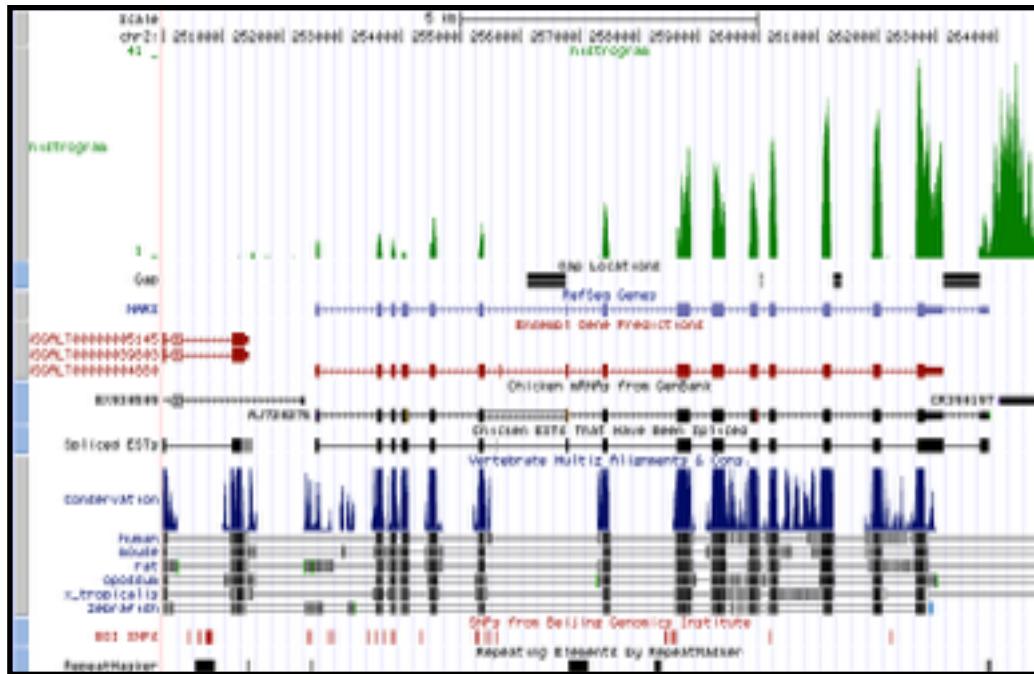


# motivation

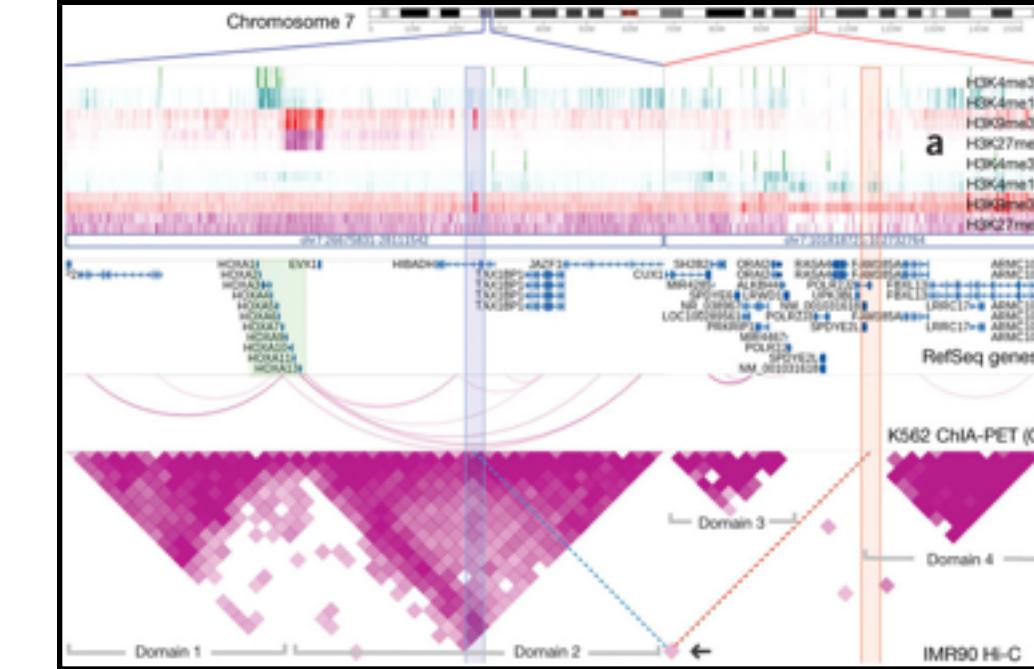


# motivation

advanced



*UCSC Genome Browser*



*WashU Epigenome Browser*

analysis power

basic

complex

**ease of use**

easy

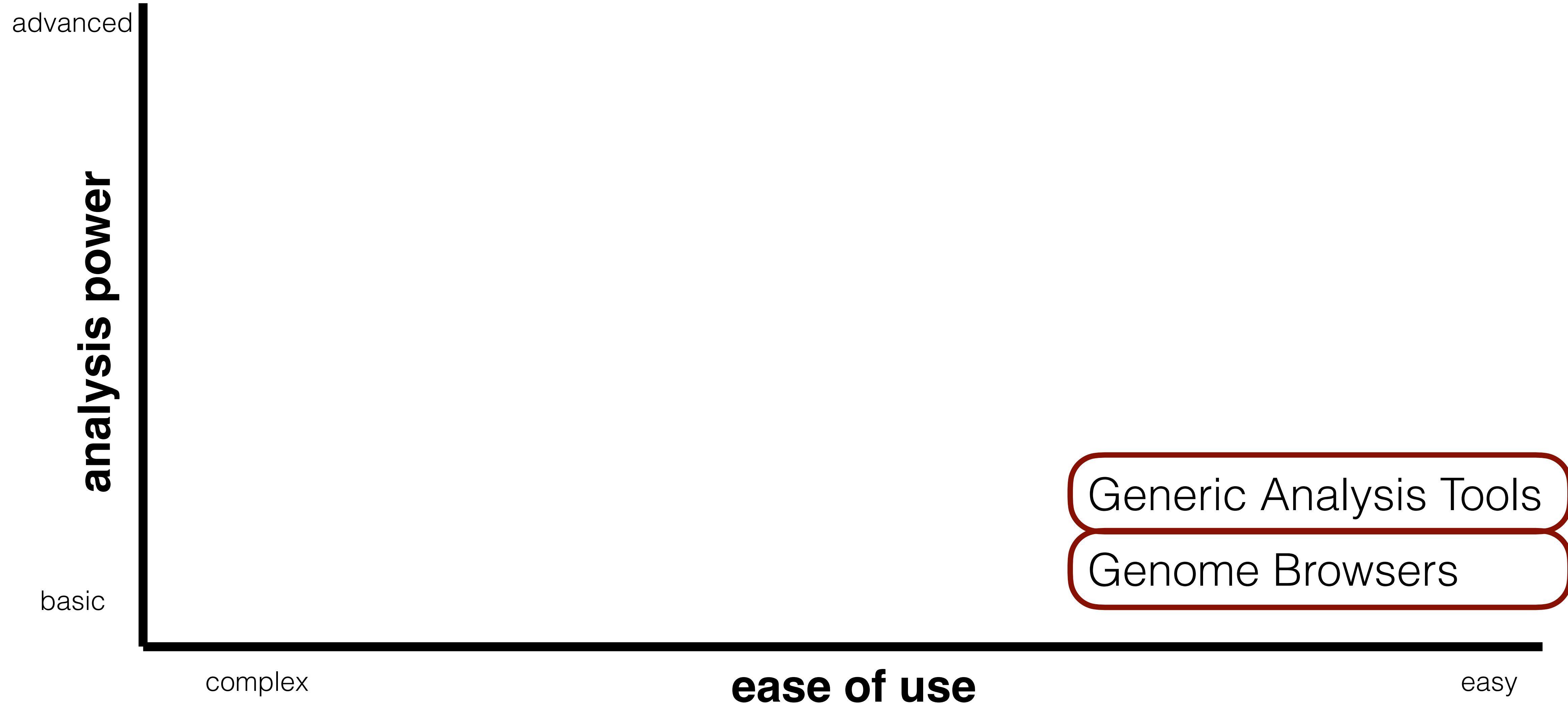


*Integrated Genome Viewer*

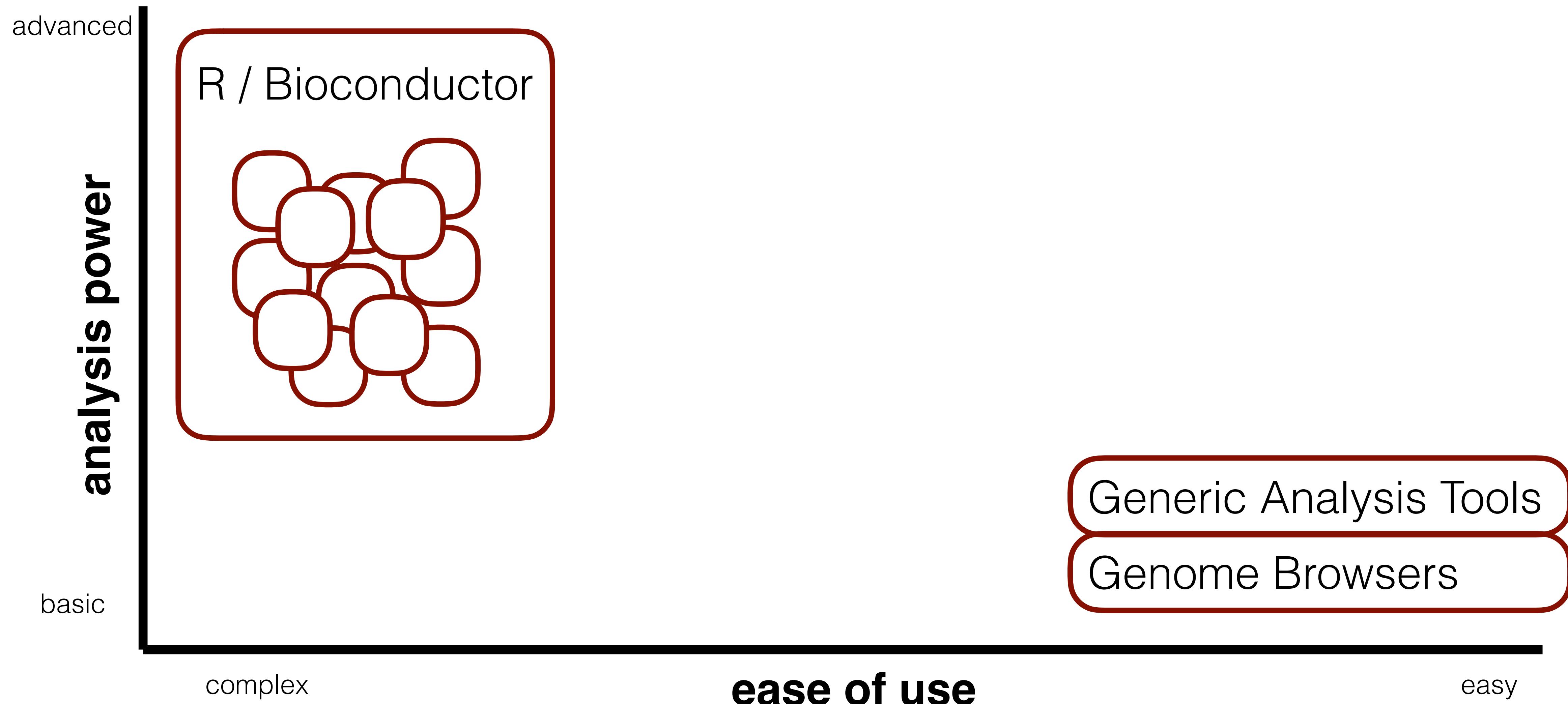
Generic Analysis Tools

Genome Browsers

# motivation



# motivation



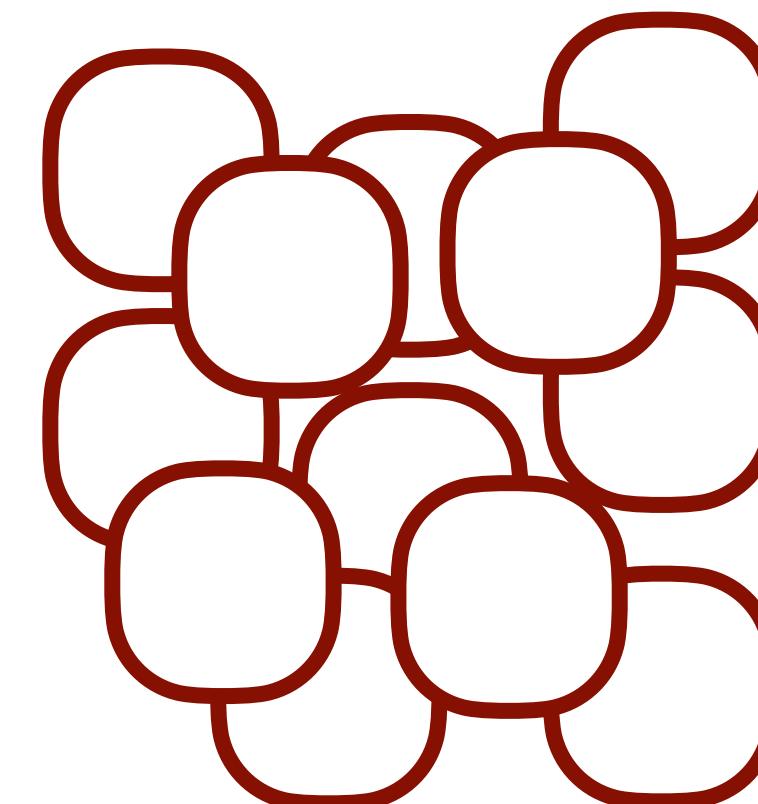
# motivation

advanced

analysis power

basic

R / Bioconductor



complex

ease of use

easy

Packages found under Sequencing:

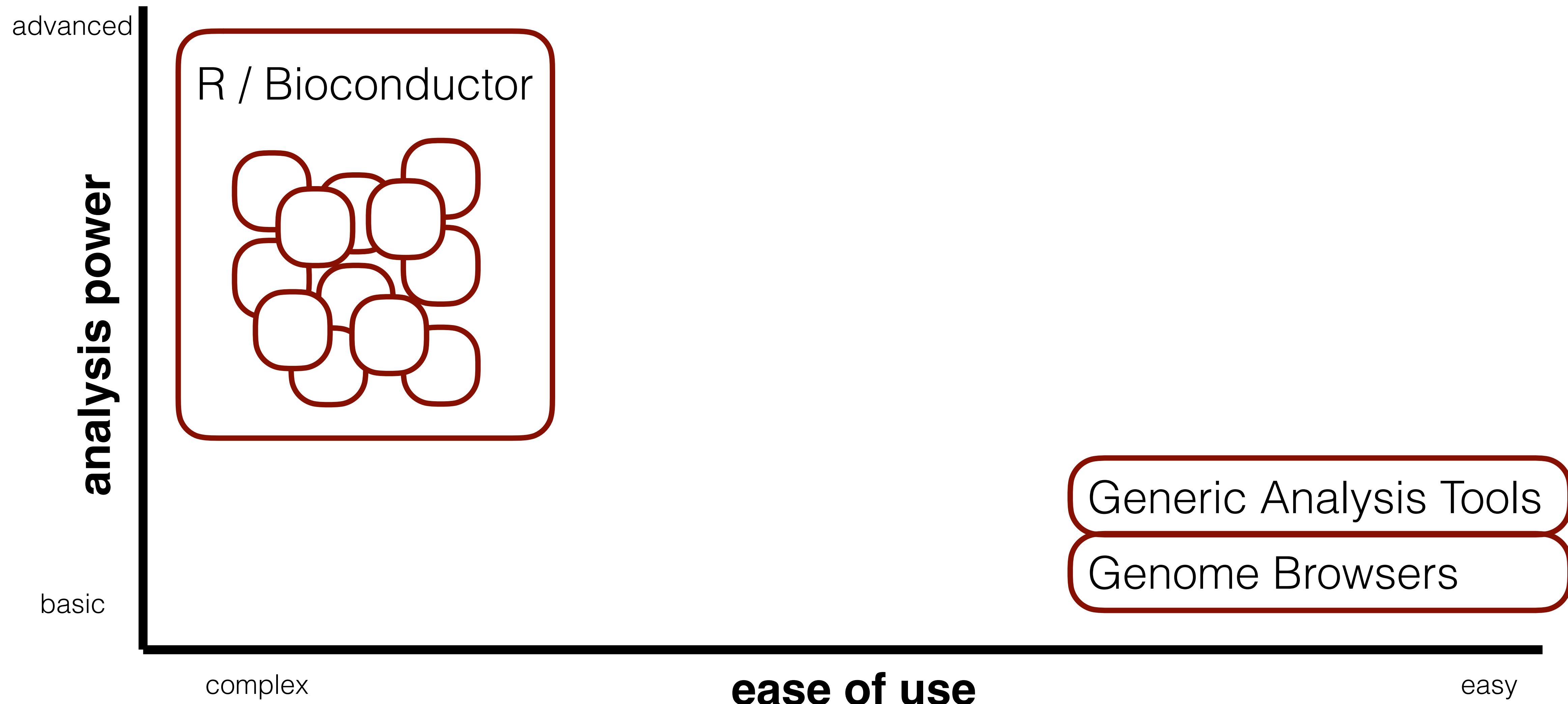
Package	Maintainer	Title
AIMS	Eric R. Paquet	AIMS : Absolute Assignment of Breast Cancer Intrinsic Molecular Subtype
ALDEX2	Greg Gloor	Analysis of differential abundance taking sample variation into account
AllelicImbalance	Jesper R. Gadin	Investigates allele specific expression
ampliQueso	Michał Okoniewski	Analysis of amplicon enrichment panels
AnnotationDbi	Bioconductor Package Maintainer	Annotation Database Interface
anota	Ola Larsson	ANalysis Of Translational Activity (ANOTA).
ArrayExpressHTS	Angela Goncalves, Andrew Tikhonov	ArrayExpress High Throughput Sequencing Processing Pipeline
BADER	Andreas Neudecker	Bayesian Analysis of Differential Expression in RNA Sequencing Data
ballgown	Alyssa Frazee	Flexible, isoform-level differential expression analysis
bamsignals	Alessandro Mammana	Extract read count signals from bam files

*Bioconductor Repository*

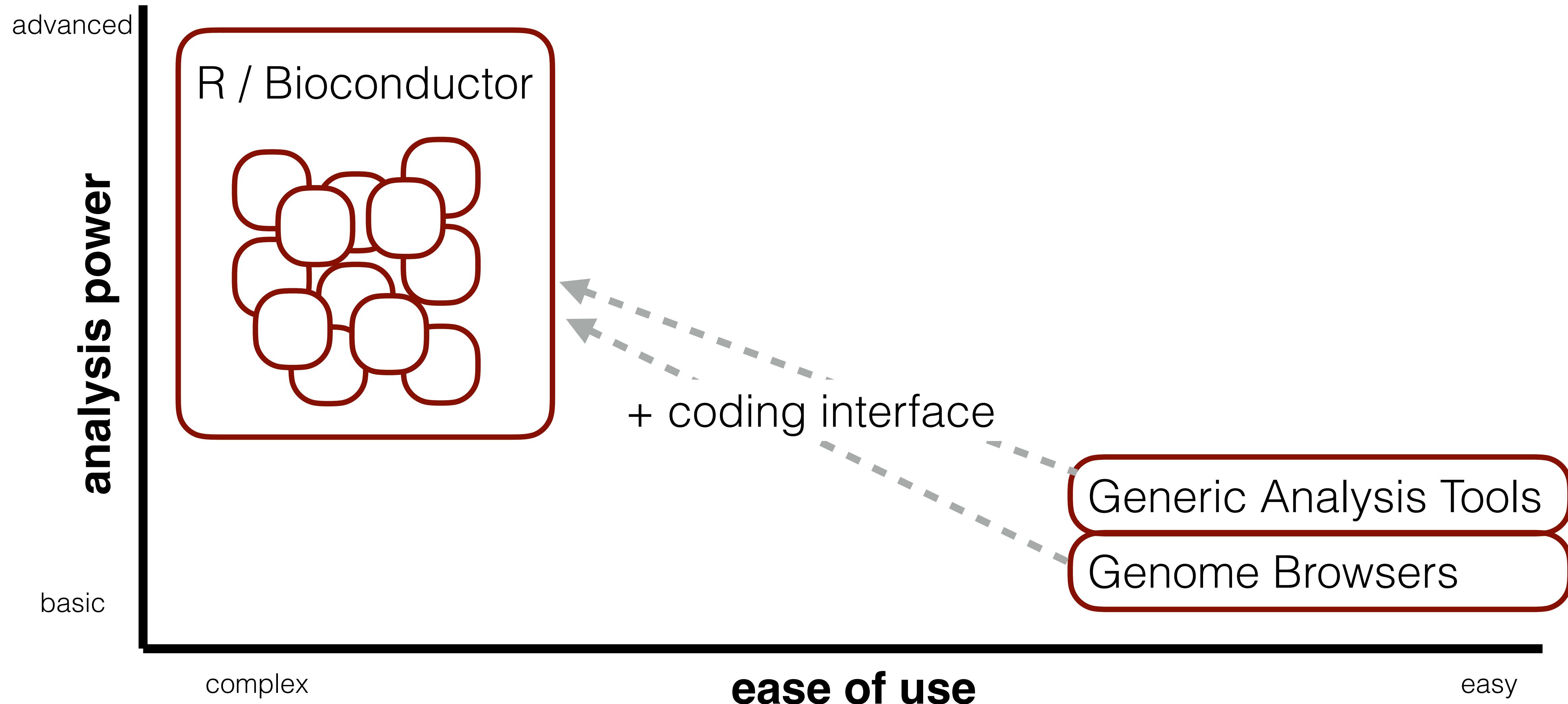
Generic Analysis Tools

Genome Browsers

# motivation



# motivation

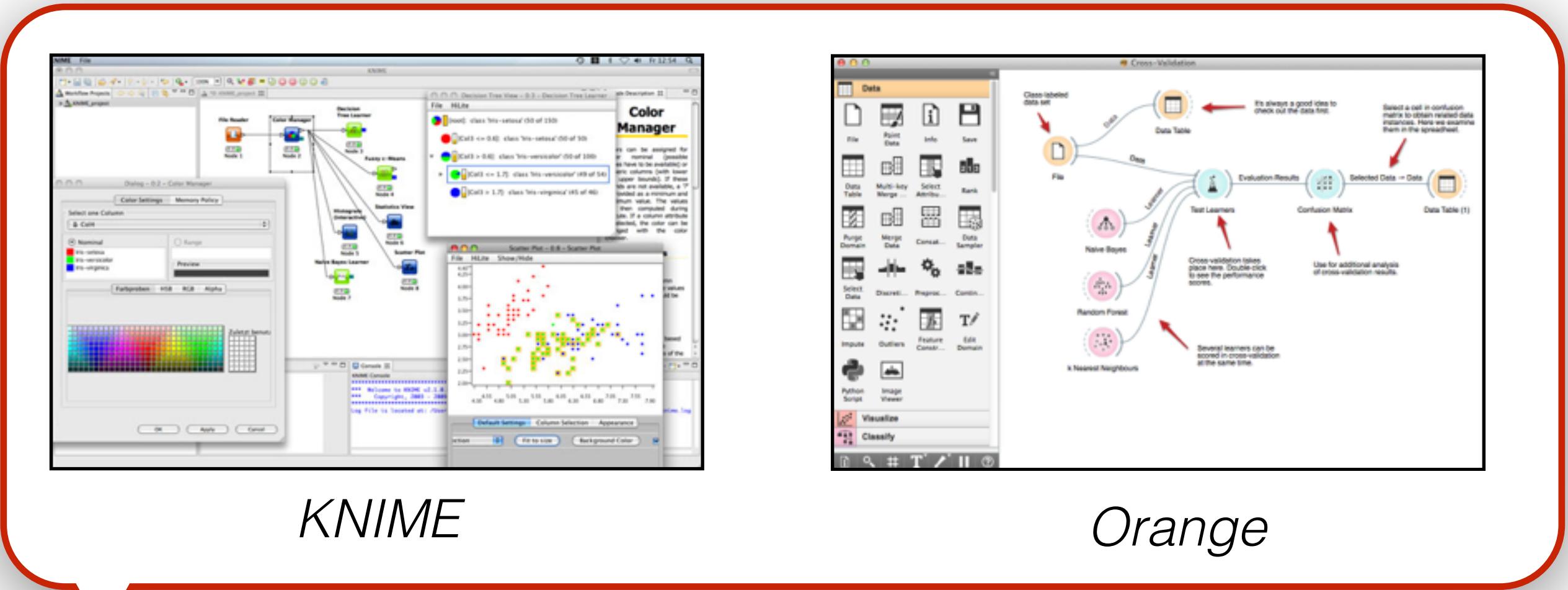
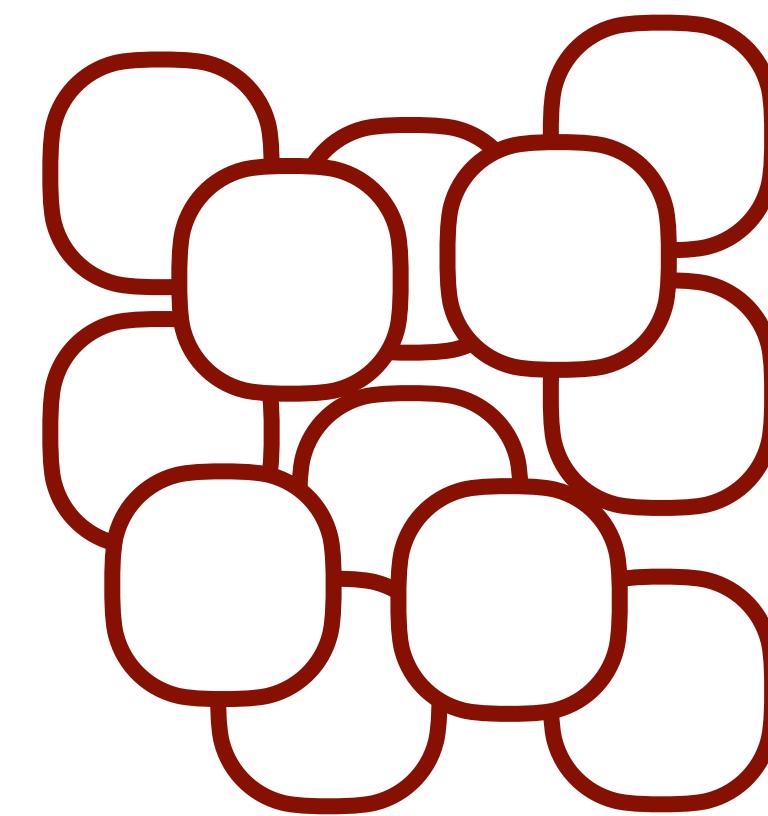


# motivation

advanced

analysis power

R / Bioconductor



KNIME

Orange

+ coding interface

basic

complex

ease of use

easy

Generic Analysis Tools  
Genome Browsers

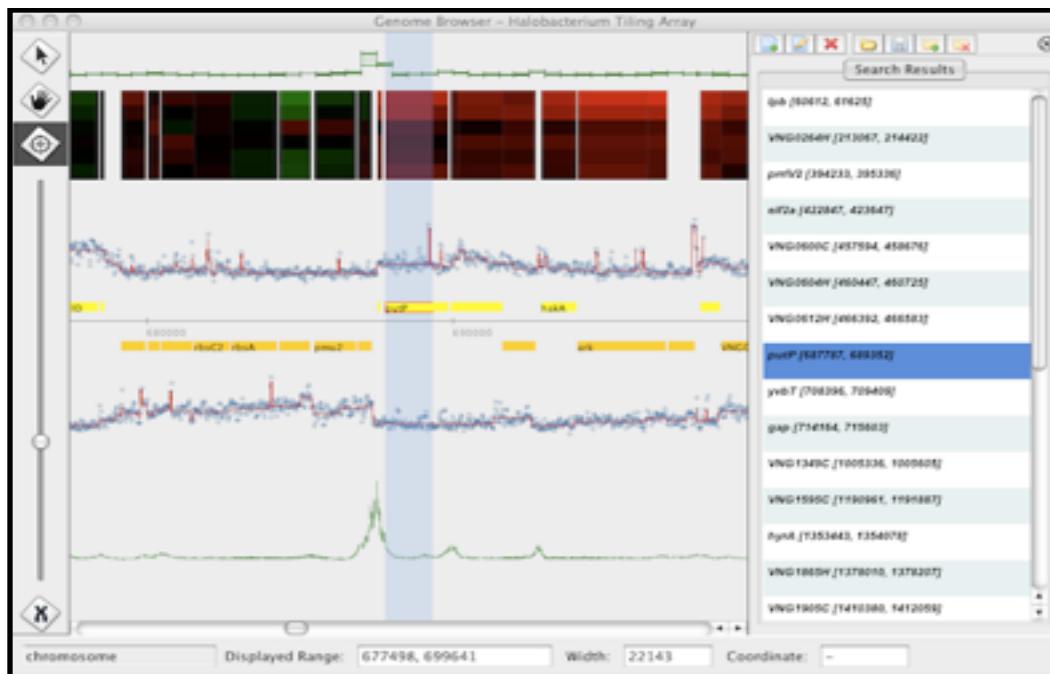
# motivation

advanced

R / Bioconductor



Epiviz



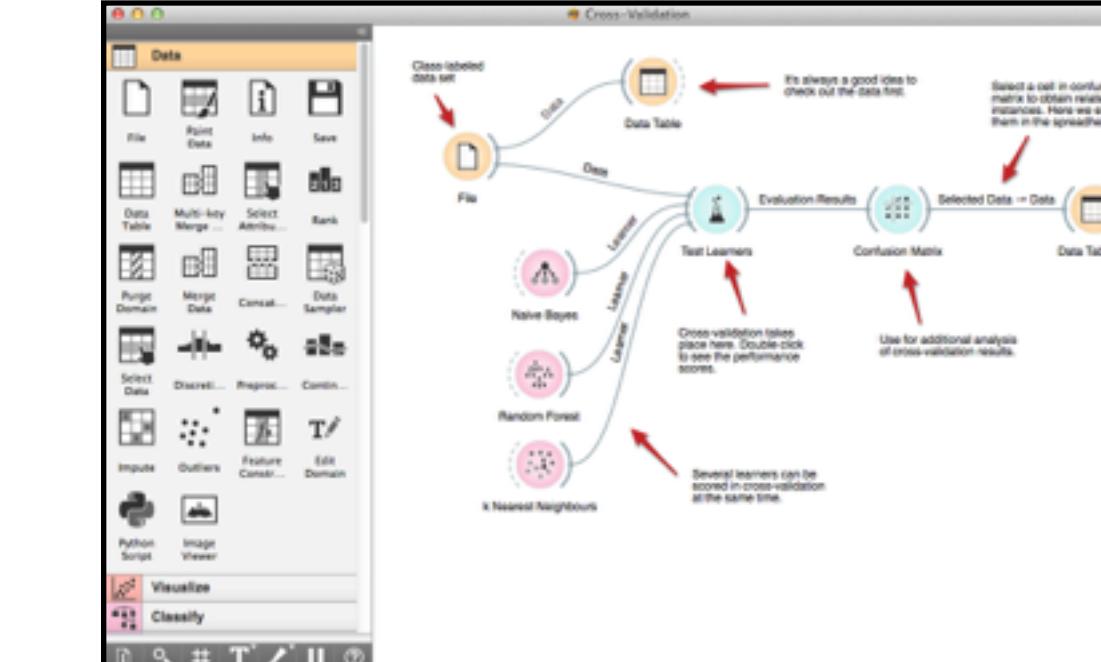
Gaggle Genome Browser

analysis power

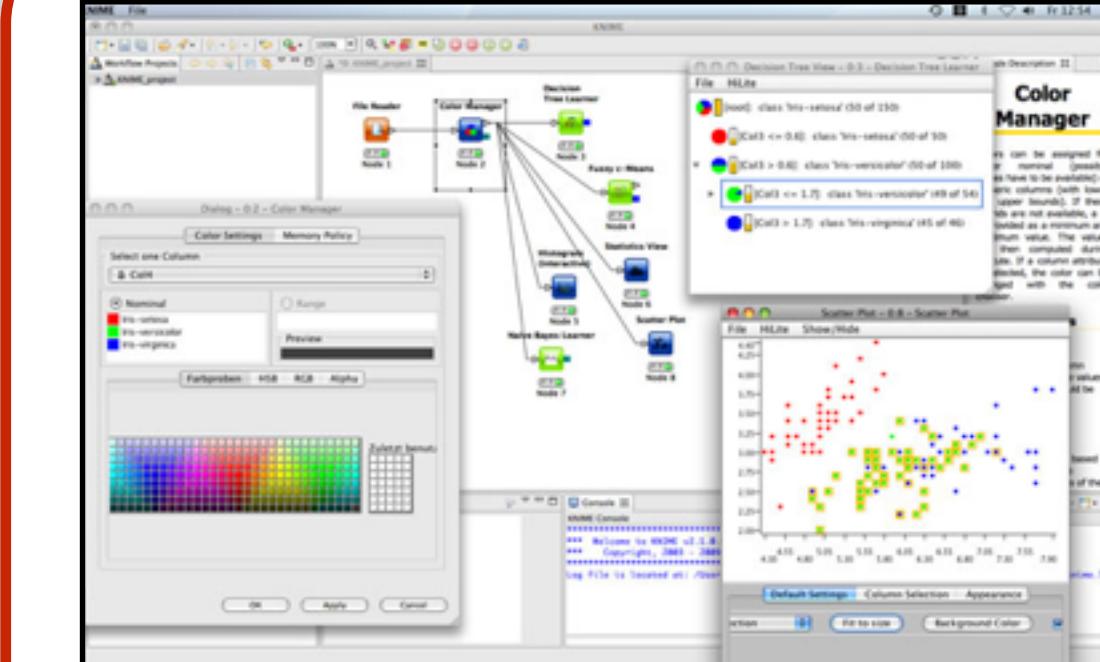
+ coding interface

ease of use

basic



Orange



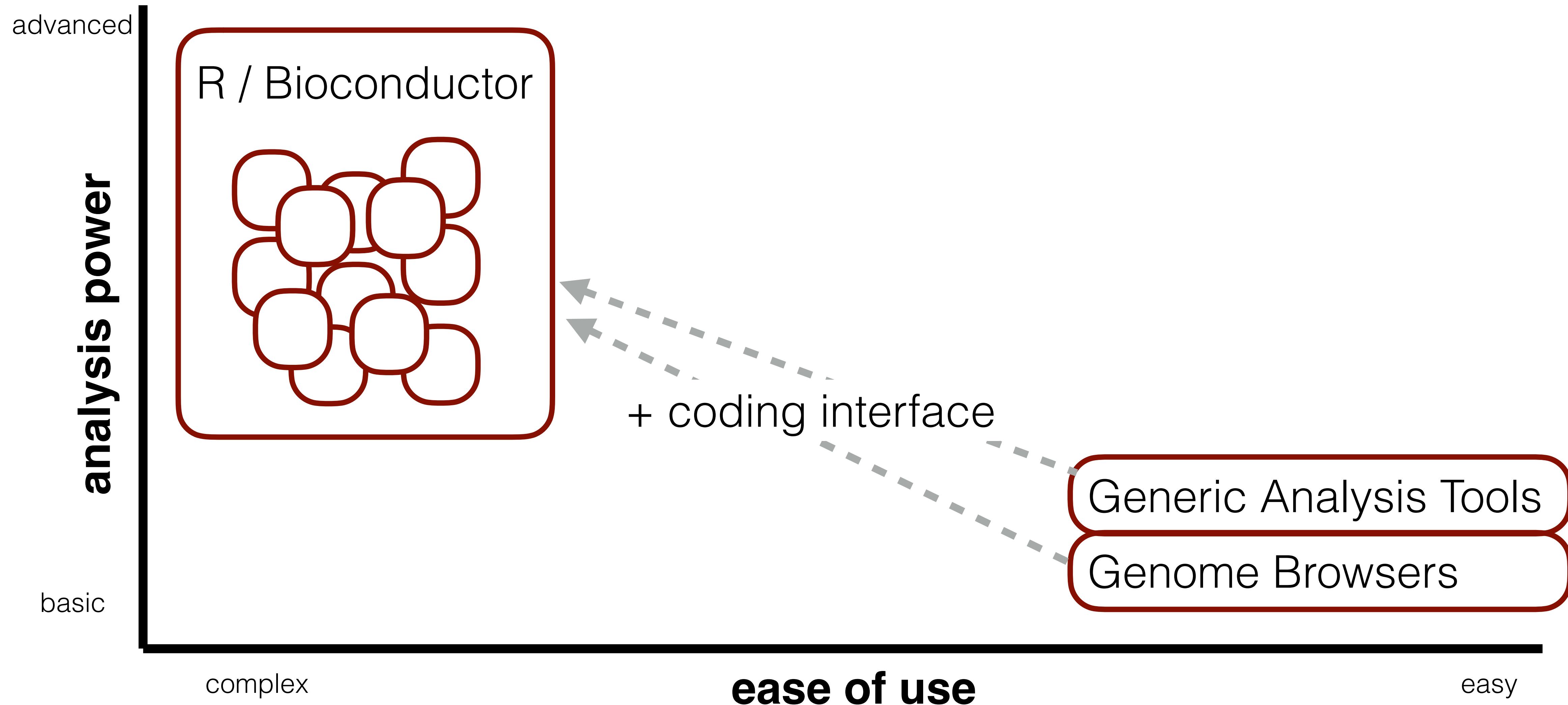
KNIME

Generic Analysis Tools

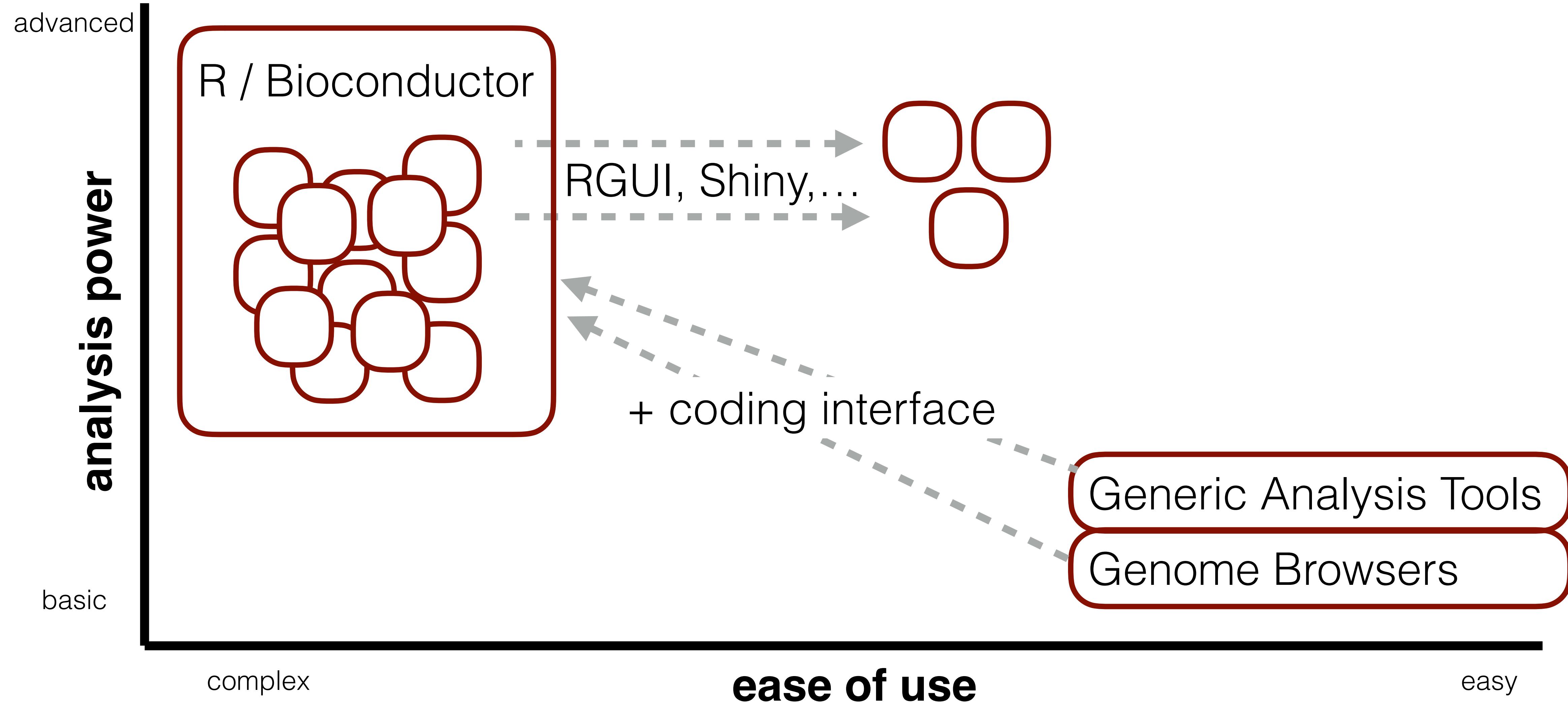
Genome Browsers

easy

# motivation



# motivation



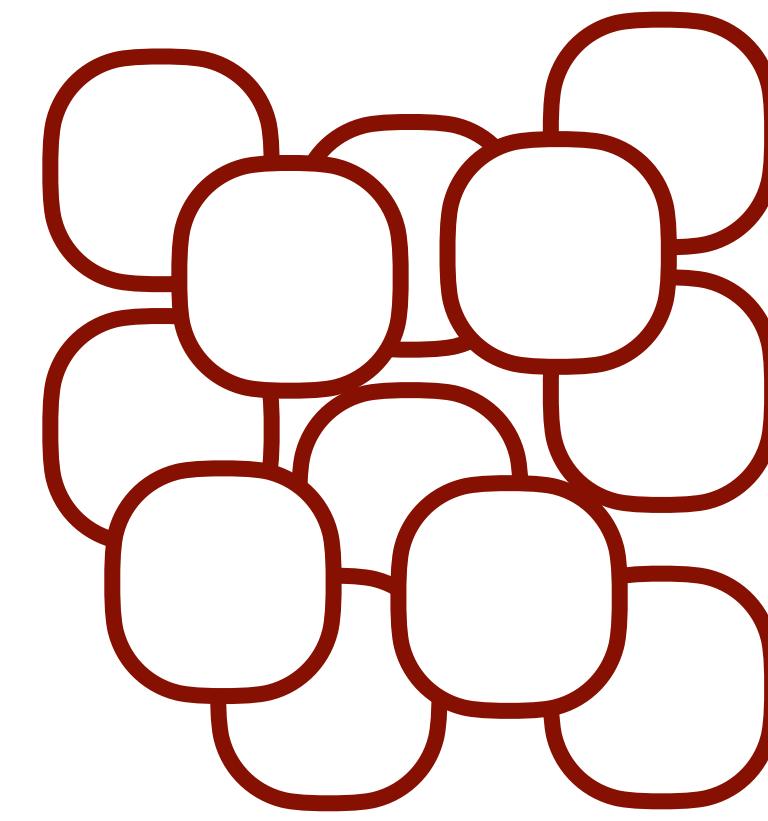
# motivation

advanced

analysis power

basic

R / Bioconductor



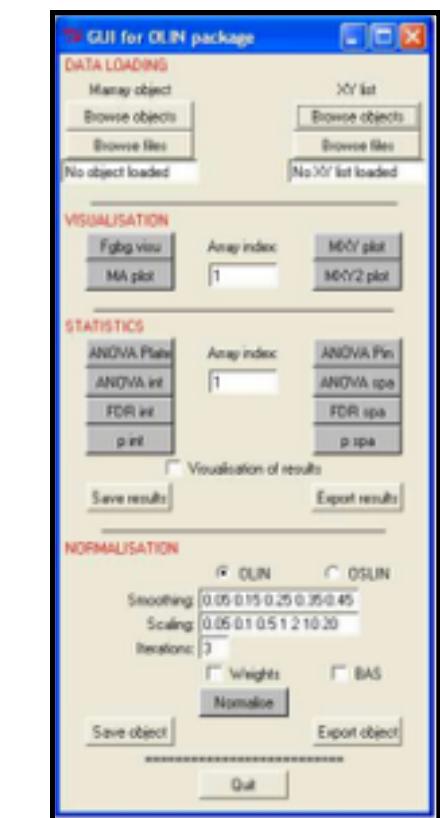
RGUI, Shiny, ...

+ coding interface

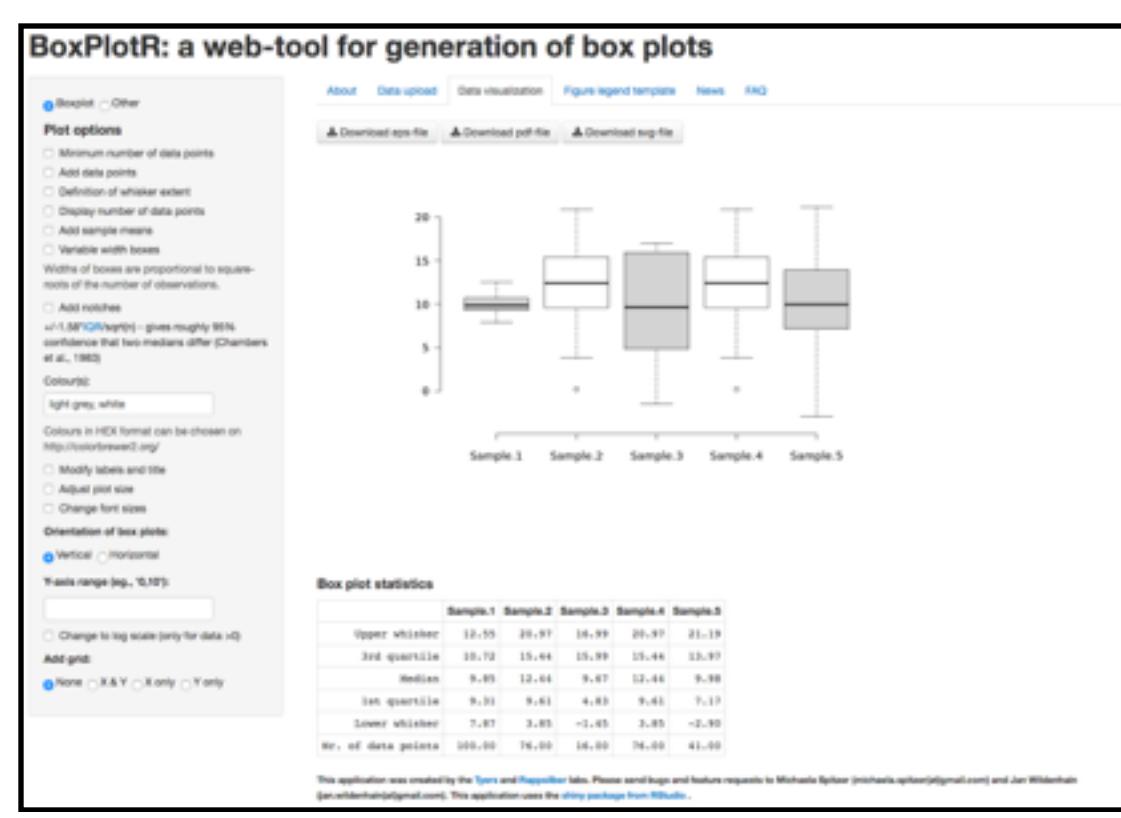
complex

ease of use

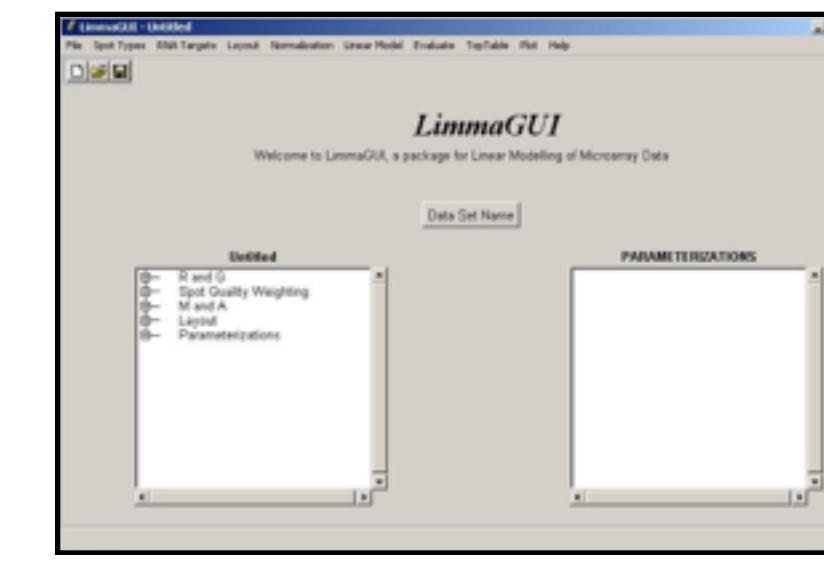
easy



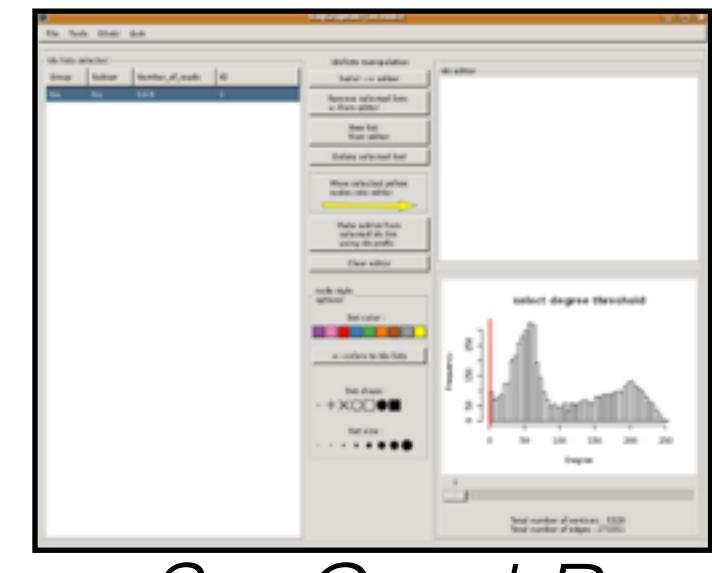
OLINgui



BoxPlotR



LimmaGUI

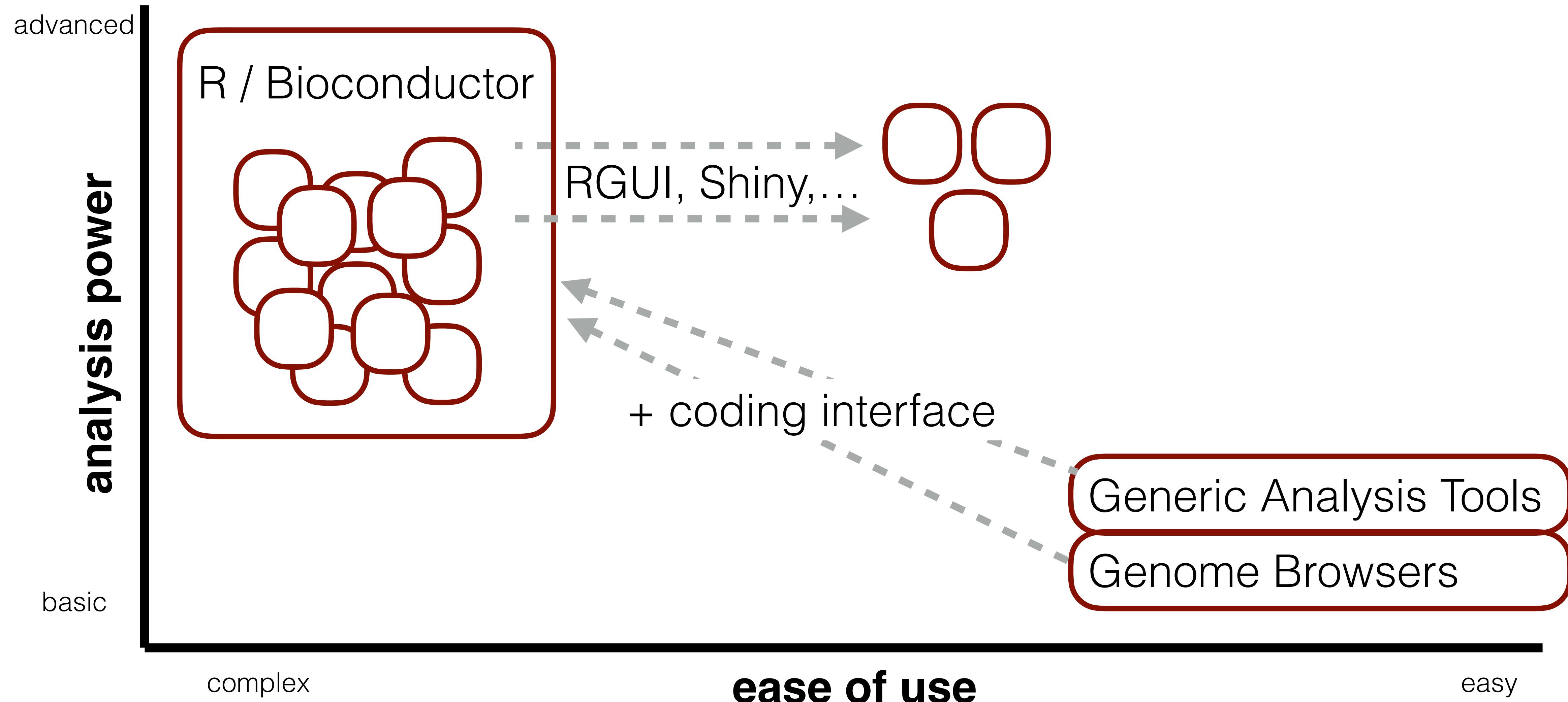


SeqGraphR

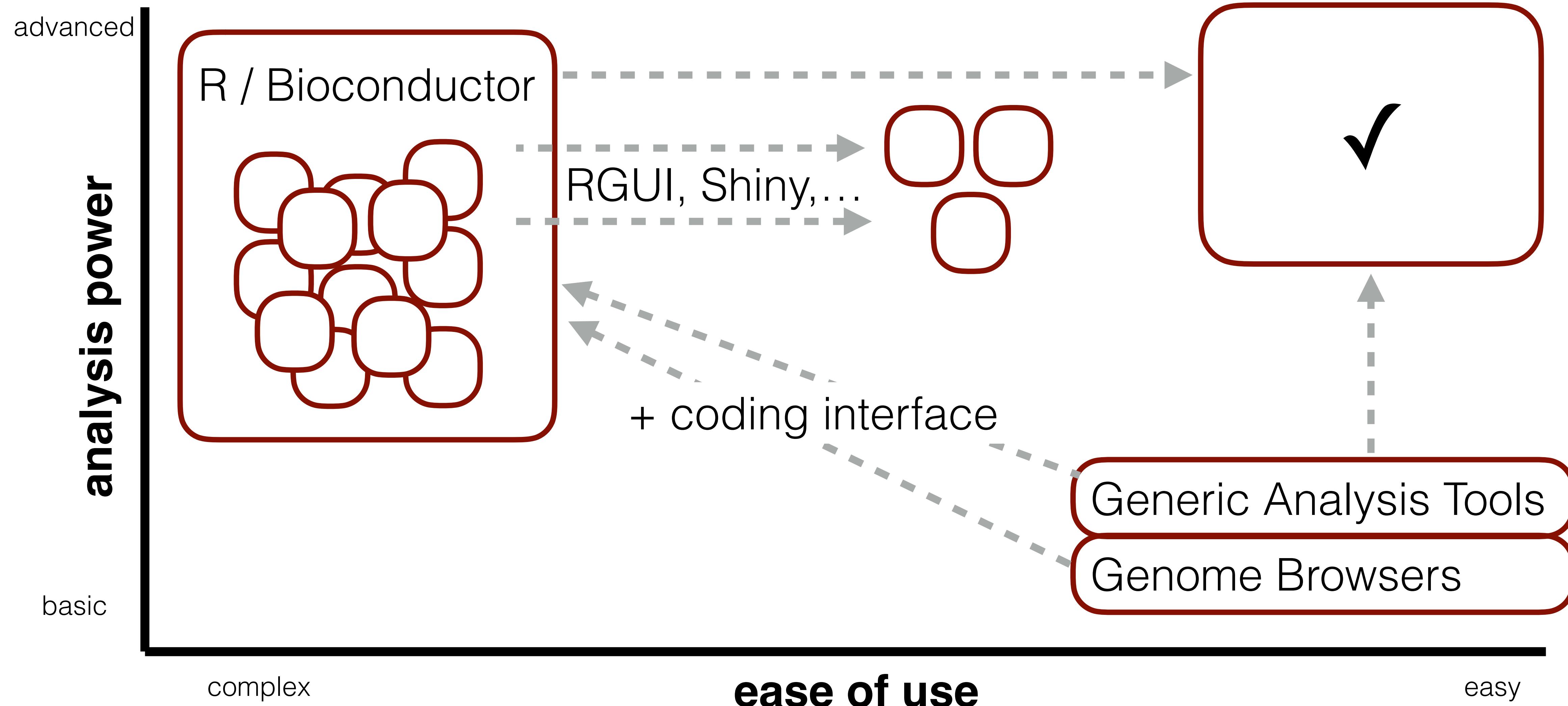
Generic Analysis Tools

Genome Browsers

# motivation

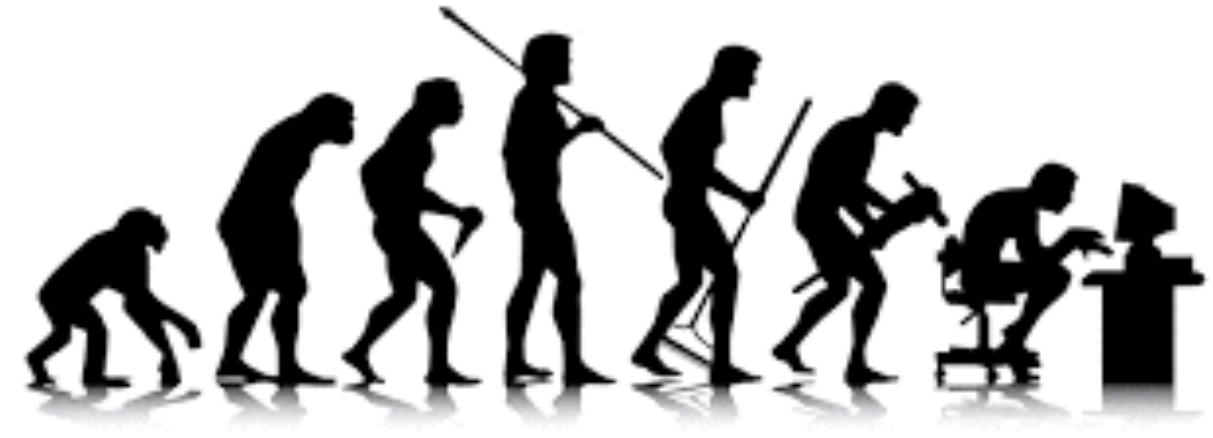


# motivation



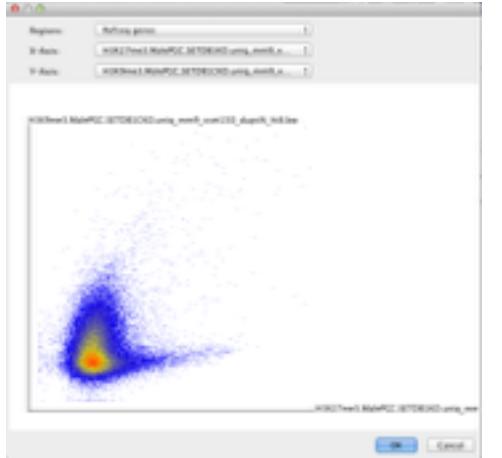
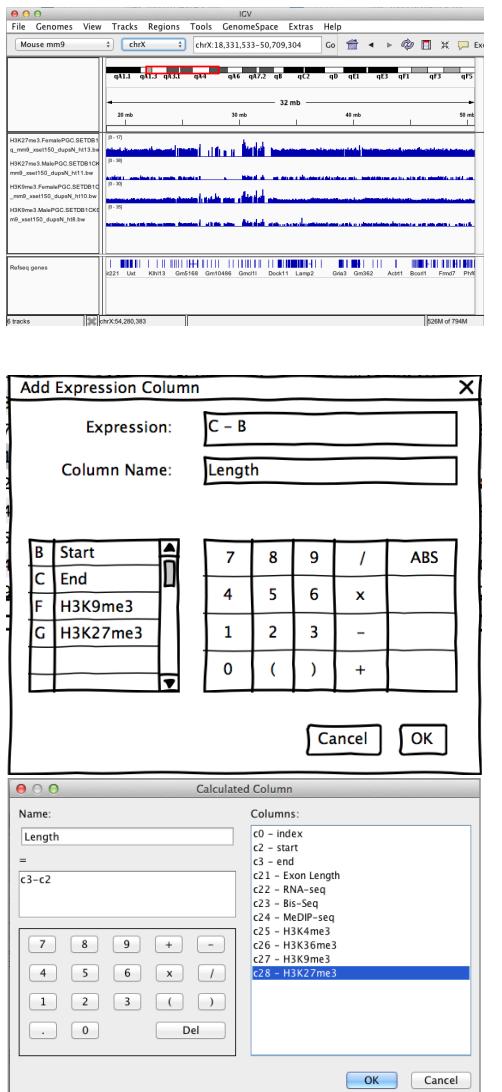
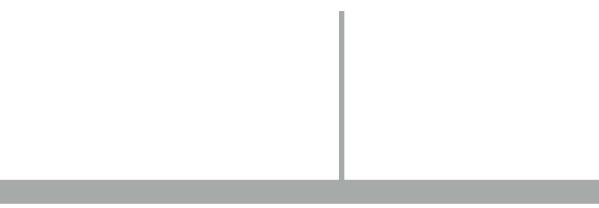
# VisRseq

# design evolution

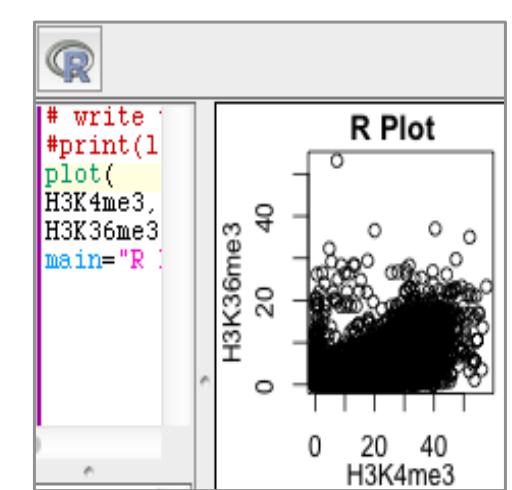


IGV plugin  
(2013)

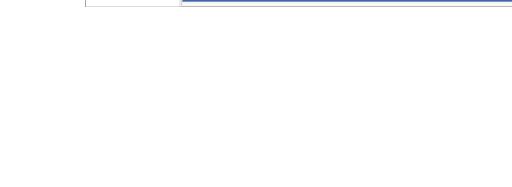
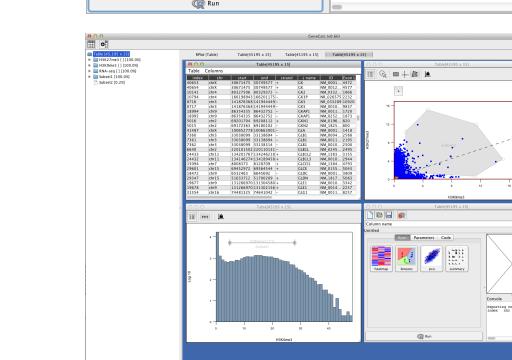
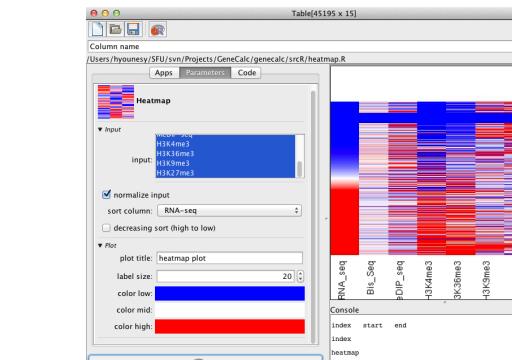
workspace



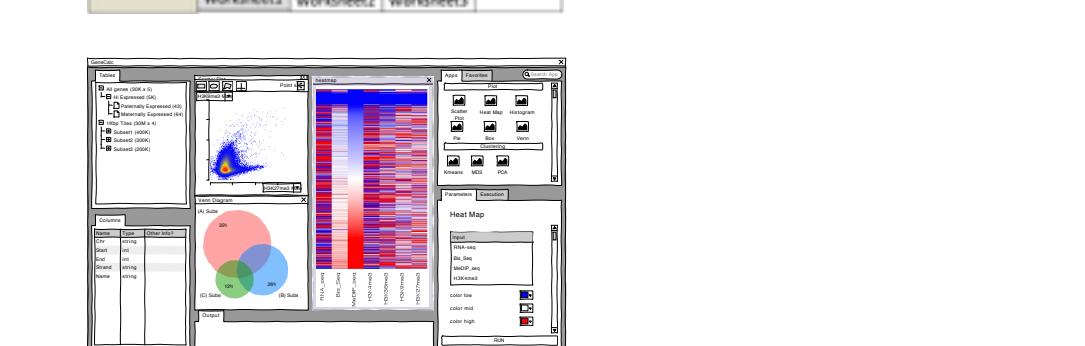
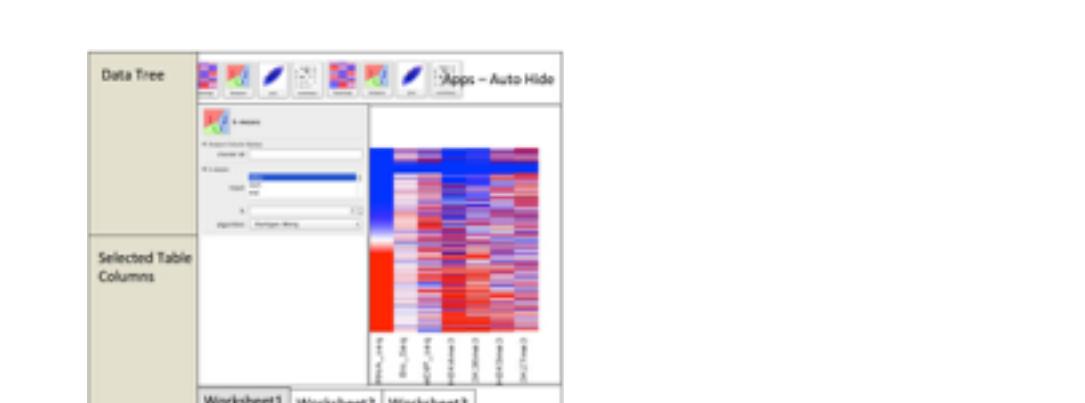
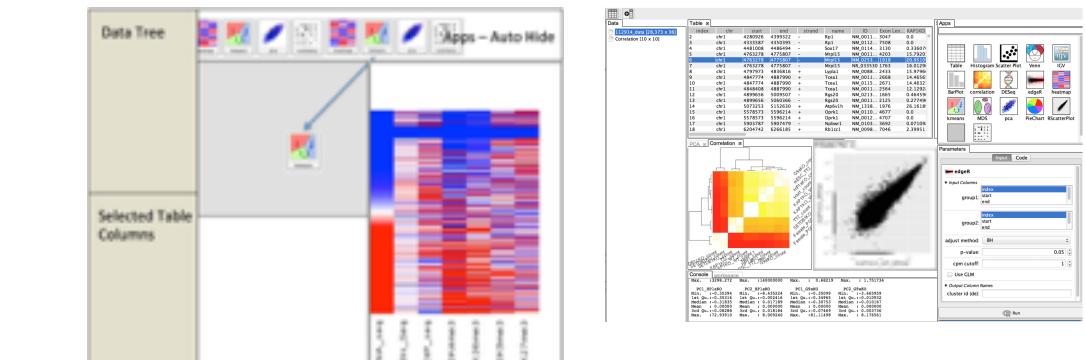
R  
integration



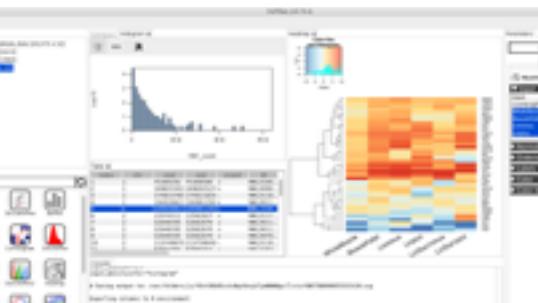
BioVis Poster  
(2014)



Redesigned  
Layout  
(2015)

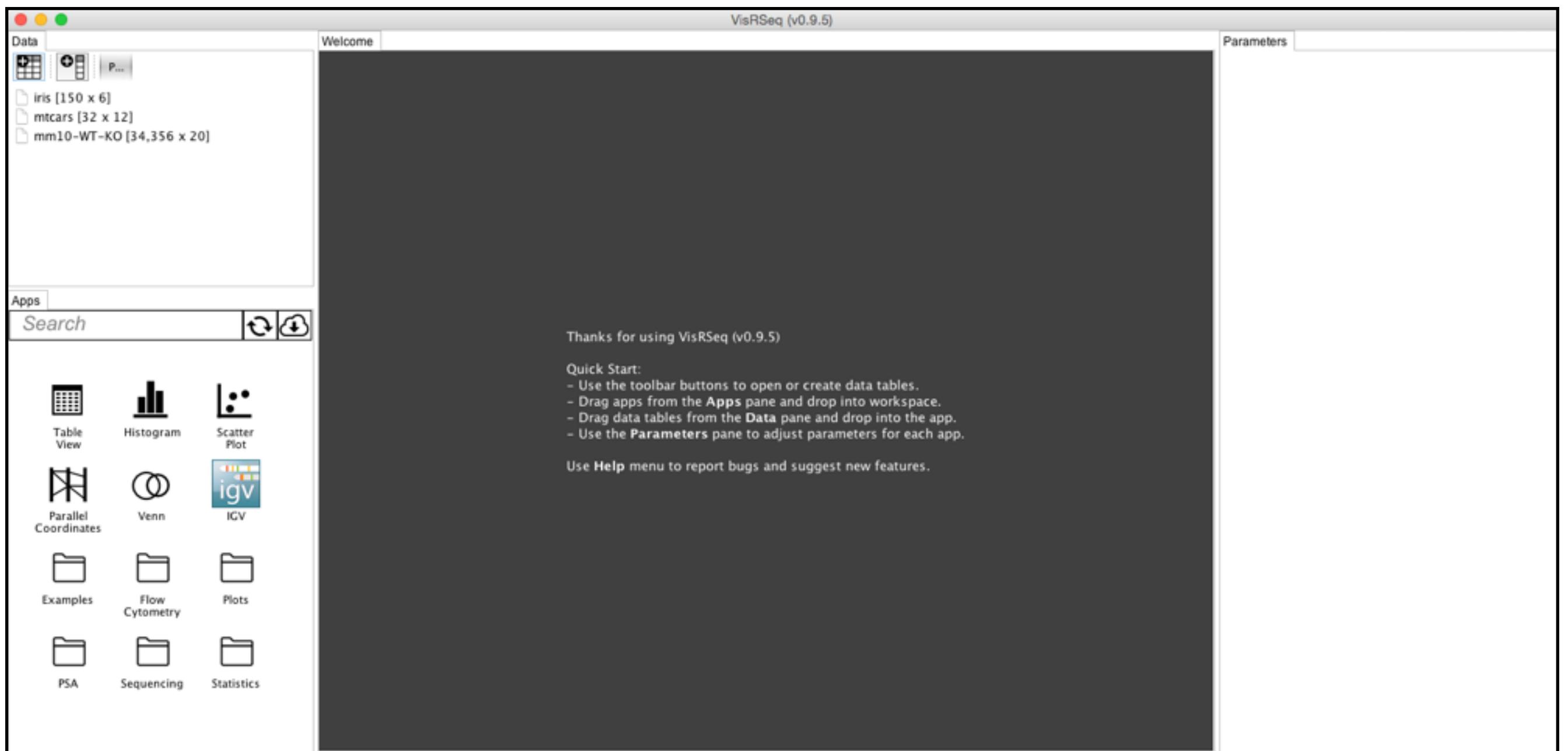


BMC Bioinf  
open beta

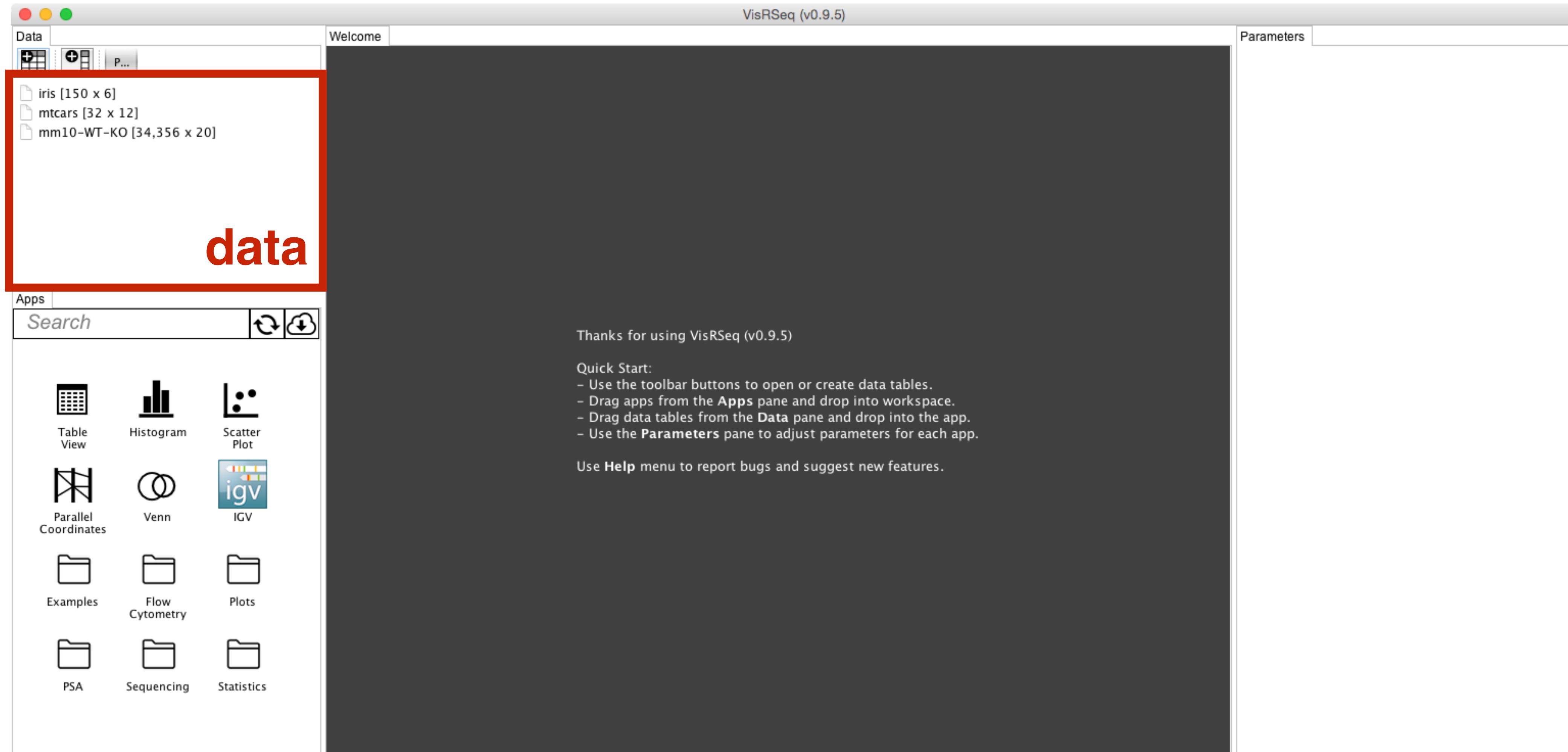


# interface

- Installation Requirements:
  - Java SE 7
  - R
- Auto installs required packages
- Open Beta: <http://visrseq.github.io>



# interface: data



# interface: data

## Tab delimited Text

The screenshot shows the VisRSeq (v0.9.5) software interface. On the left, there is a sidebar with various data analysis tools: Data (iris [150 x 6], mtcars [32 x 12], mm10-WT-KO [34,35]), Apps (Search, Table View, Histogram, Parallel Coordinates, Venn, IGV), Examples, and PSA. The main workspace displays a tab-delimited text file named "iris.txt" with the first 14 rows of the Iris dataset. The columns are labeled A through E, and the species names are listed in the last column. Below the table, there is a "Calculated Column" dialog box. The "New Column Name:" field contains "zscore". The "Equation =" field contains "ZSCORE(c26, c27)". To the right of the equation, there is a list of numerical functions: ABS, SQRT, LN, LOG, LOG10, EXP, POWER, INT, FLOOR, CEILING, MOD, MIN, MAX, AVERAGE, STDEV, MEDIAN, SUM, COUNT, ZSCORE, IF, NOT, AND, OR. At the bottom of the dialog are "OK" and "Cancel" buttons.

	A	B	C	D	E
1	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
2	5.1	3.5	1.4	0.2	setosa
3	4.9	3	1.4	0.2	setosa
4	4.7	3.2	1.3	0.2	setosa
5	4.6	3.1	1.5	0.2	setosa
6	5	3.6	1.4	0.2	setosa
7	5.4	3.9	1.7	0.4	setosa
8	4.6	3.4	1.4	0.3	setosa
9	5	3.4	1.5	0.2	setosa
10	4.4	2.9	1.4	0.2	setosa
11	4.9	3.1	1.5	0.1	setosa
12	5.4	3.7	1.5	0.2	setosa
13	4.8	3.4	1.6	0.2	setosa
14	4.8	3	1.4	0.1	setosa

# interface: data

Tab delimited Text

iris.txt

	A	B	C	D	E
1	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
2	5.1	3.5	1.4	0.2	setosa
3	4.9	3	1.4	0.2	setosa
4	4.7	3.2	1.3	0.2	setosa
5	4.6	3.1	1.5	0.2	setosa
6	5	3.6	1.4	0.2	setosa
7	5.4	3.9	1.7	0.4	setosa
8	4.6	3.4	1.4	0.3	setosa
9	5	3.4	1.5	0.2	setosa
10	4.4	2.9	1.4	0.2	setosa
11	4.9	3.1	1.5	0.1	setosa
12	5.4	3.7	1.5	0.2	setosa
13	4.8	3.4	1.6	0.2	setosa
14	4.8	3	1.4	0.1	setosa

iris.txt

Calculated Column

New Column Name: zscore

Numerical Columns:

- c27 - RNA-seq Male
- c28 - H3K36me3\_CASTEij
- c29 - H3K36me3\_C57BL6j
- c30 - RNA-Seq\_CASTEij\_RPKM
- c31 - RNA-Seq\_C57BL6j\_RPKM
- c22 - H2K26me2\_CASTEij\_RPKM

Equation = ZSCORE(c26, c27)

OK Cancel

.BAM, .WIG

Add Data Column

Select Data:

Name	Type
Rat_Oocyte_2_S4_001.bam	BAM
Rat_Oocyte_1_S3_001.bam	BAM

Column Name:

Regions

Exons

Range: Start: Region Start + 0 bp  
End: Region End + 0 bp  
Number of Bins: 1

Alignment Track Options

Remove Duplicate Reads

Minimum Read Quality: 1

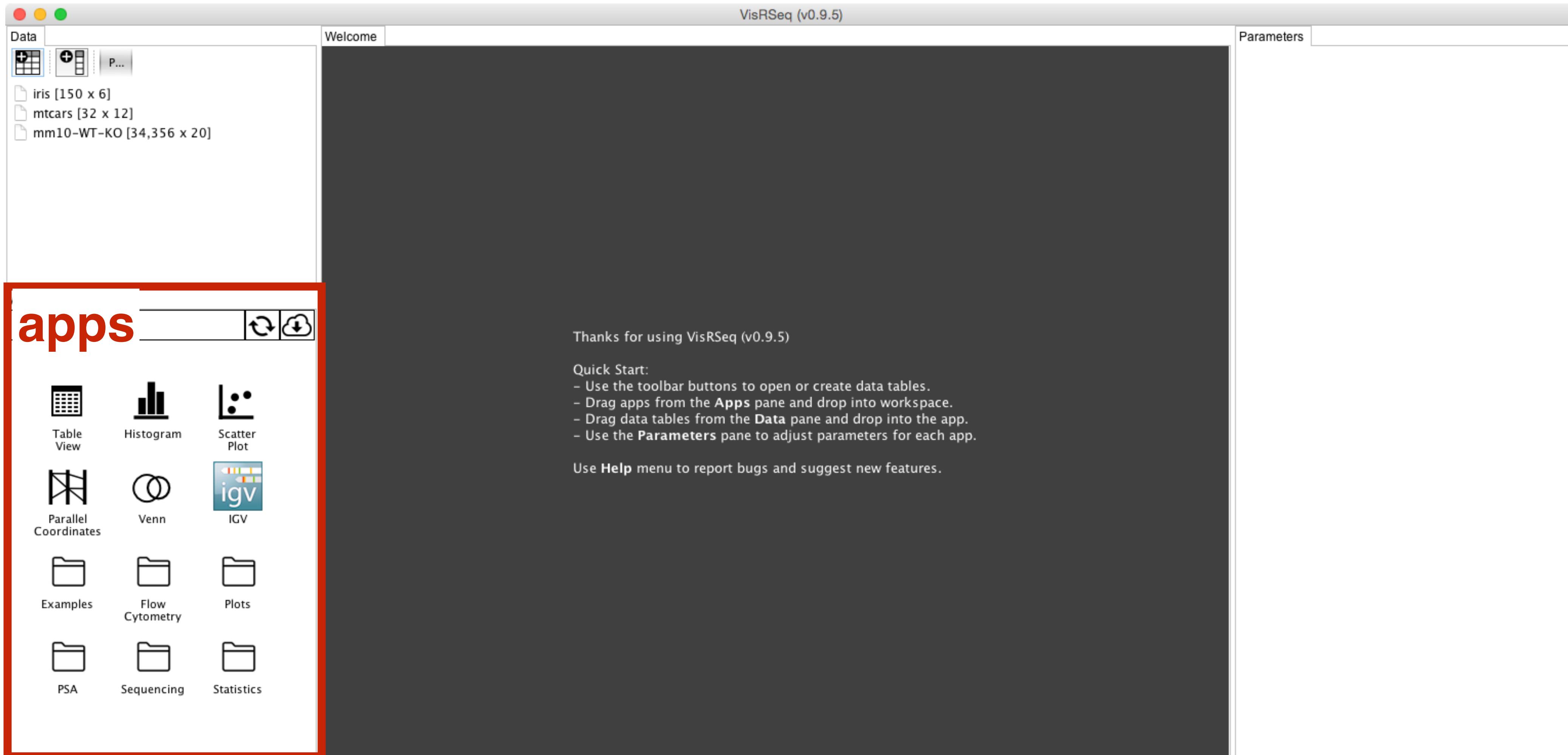
Output

RPKM  
Normalized by: total reads  
Tracks: Gene

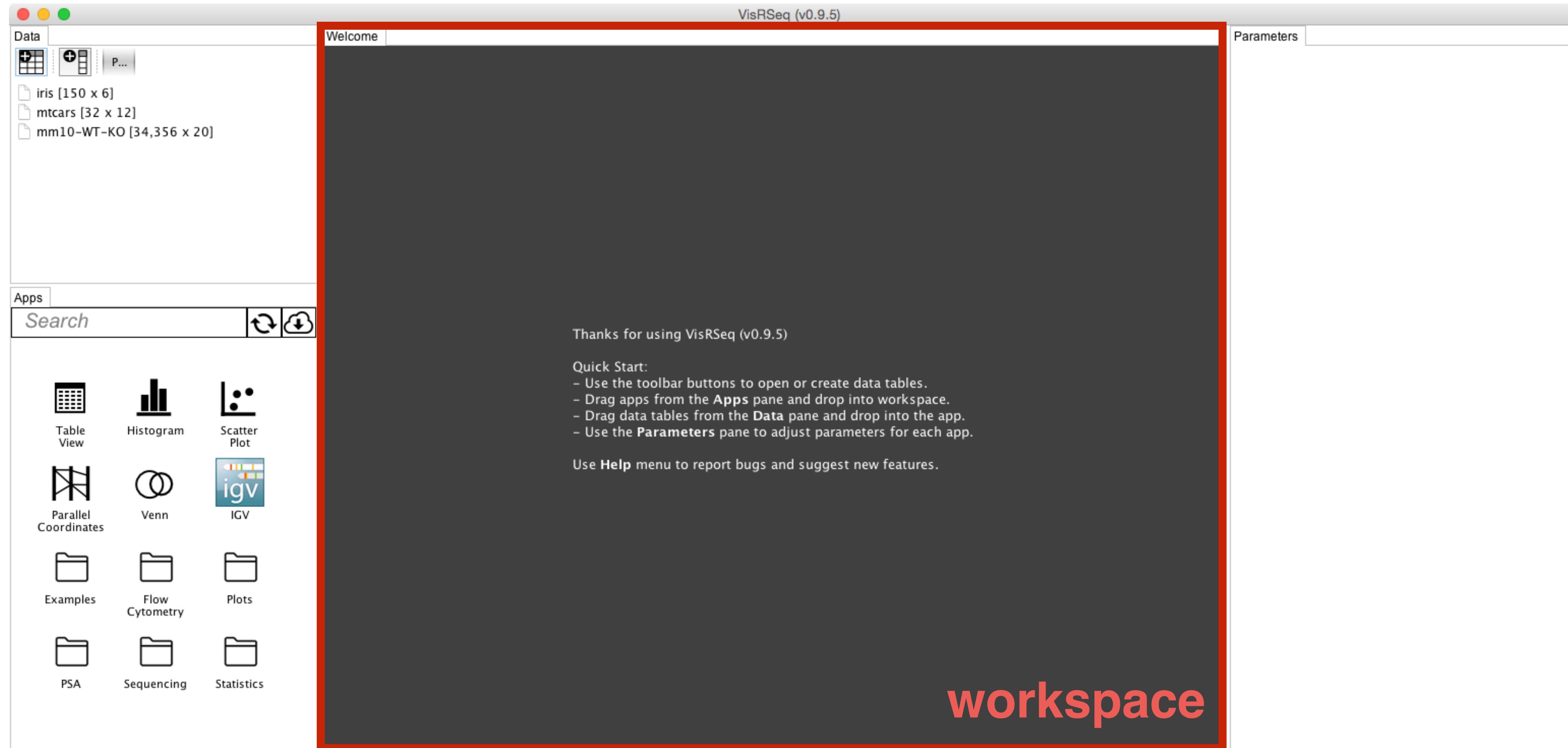
Read Count  
Read Length for BW tracks: 100

OK Cancel

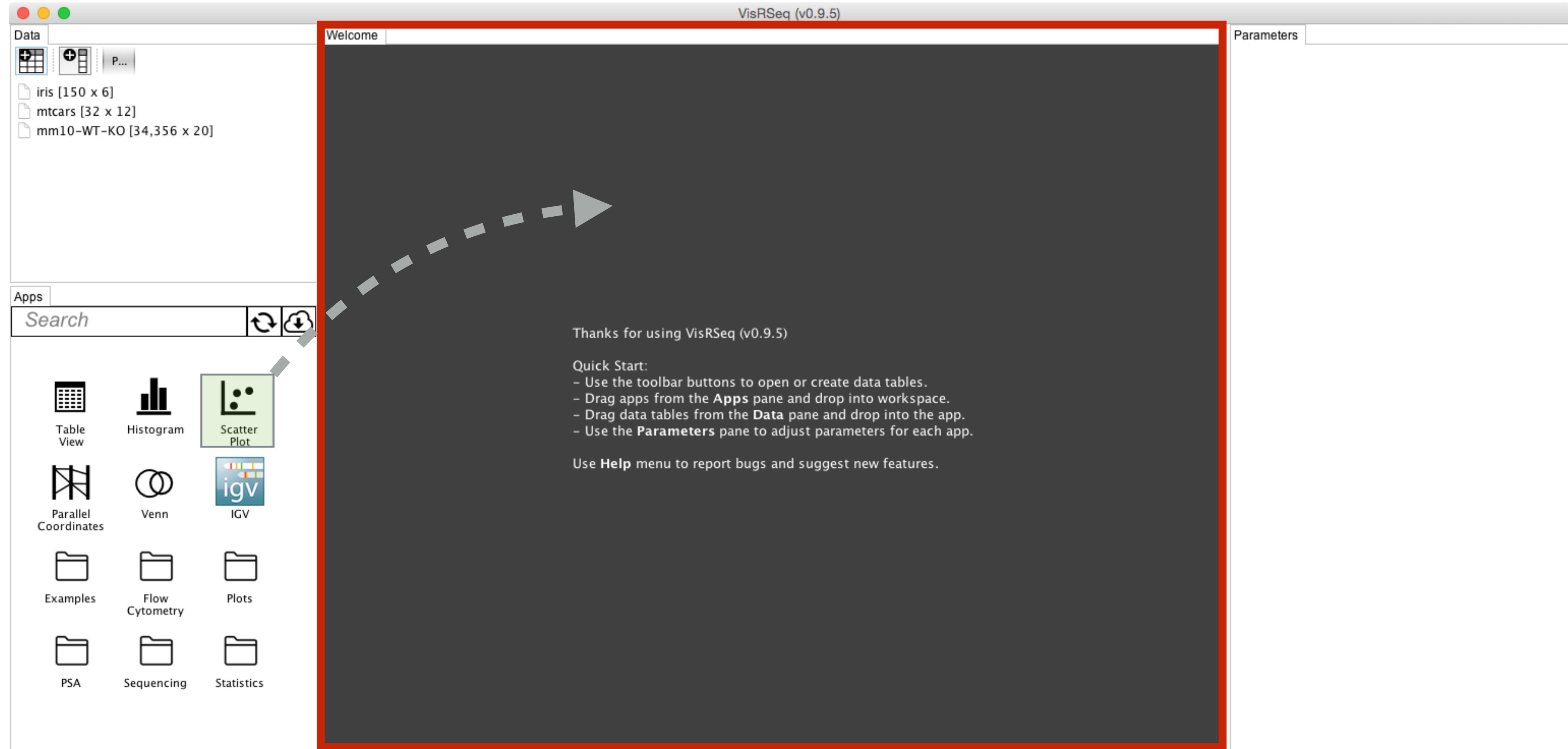
# interface: apps



# interface: workspace



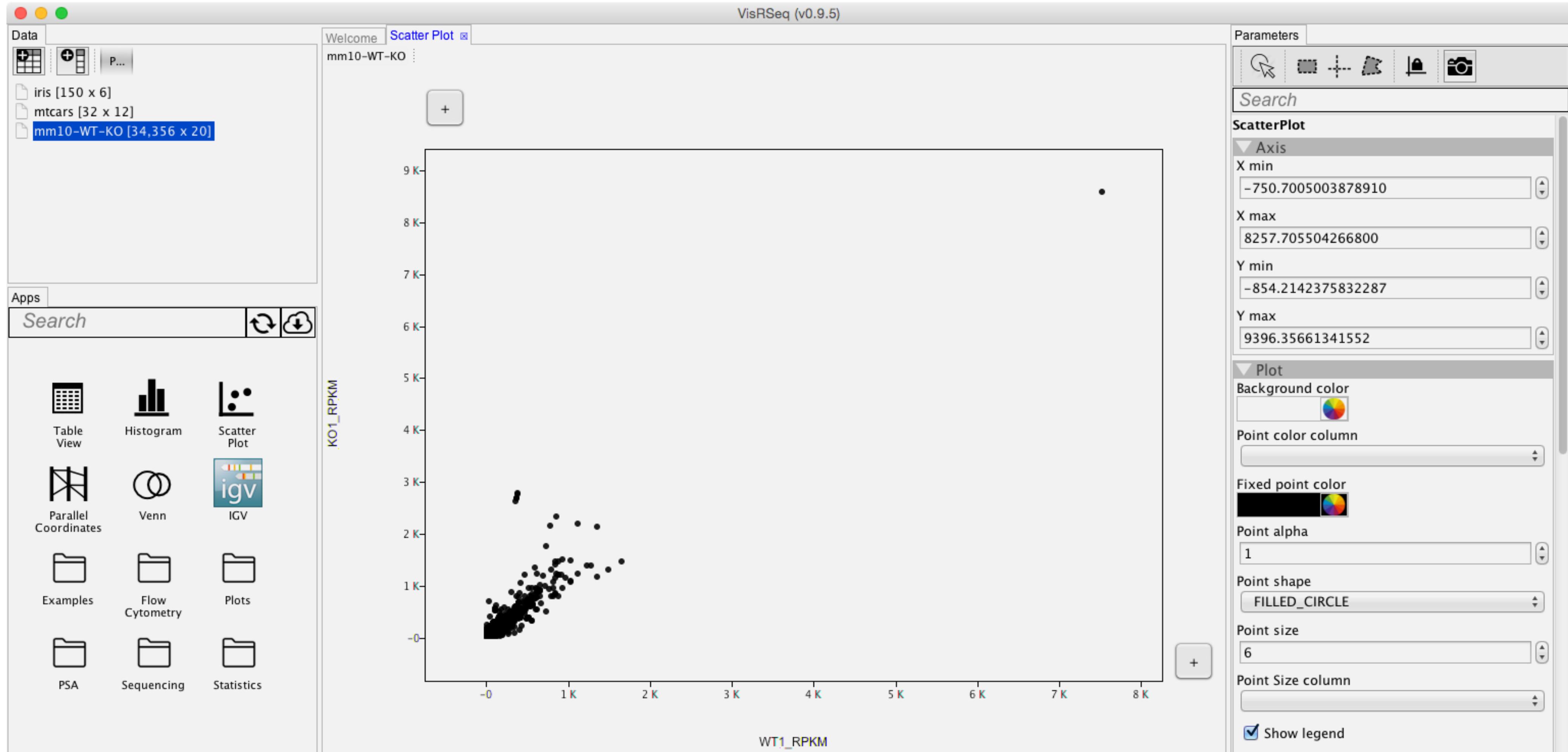
# interface: workspace



# interface: workspace

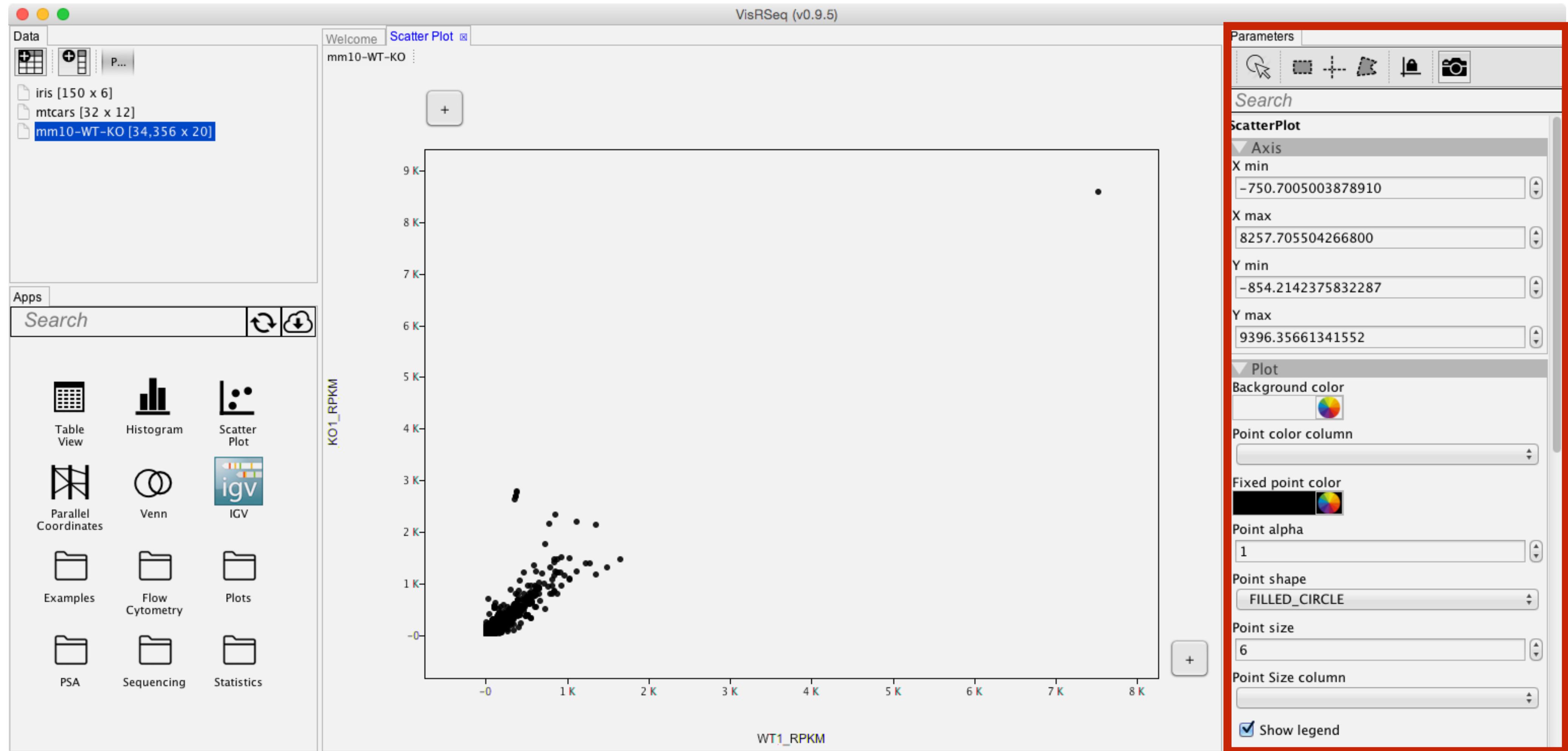


# interface: workspace



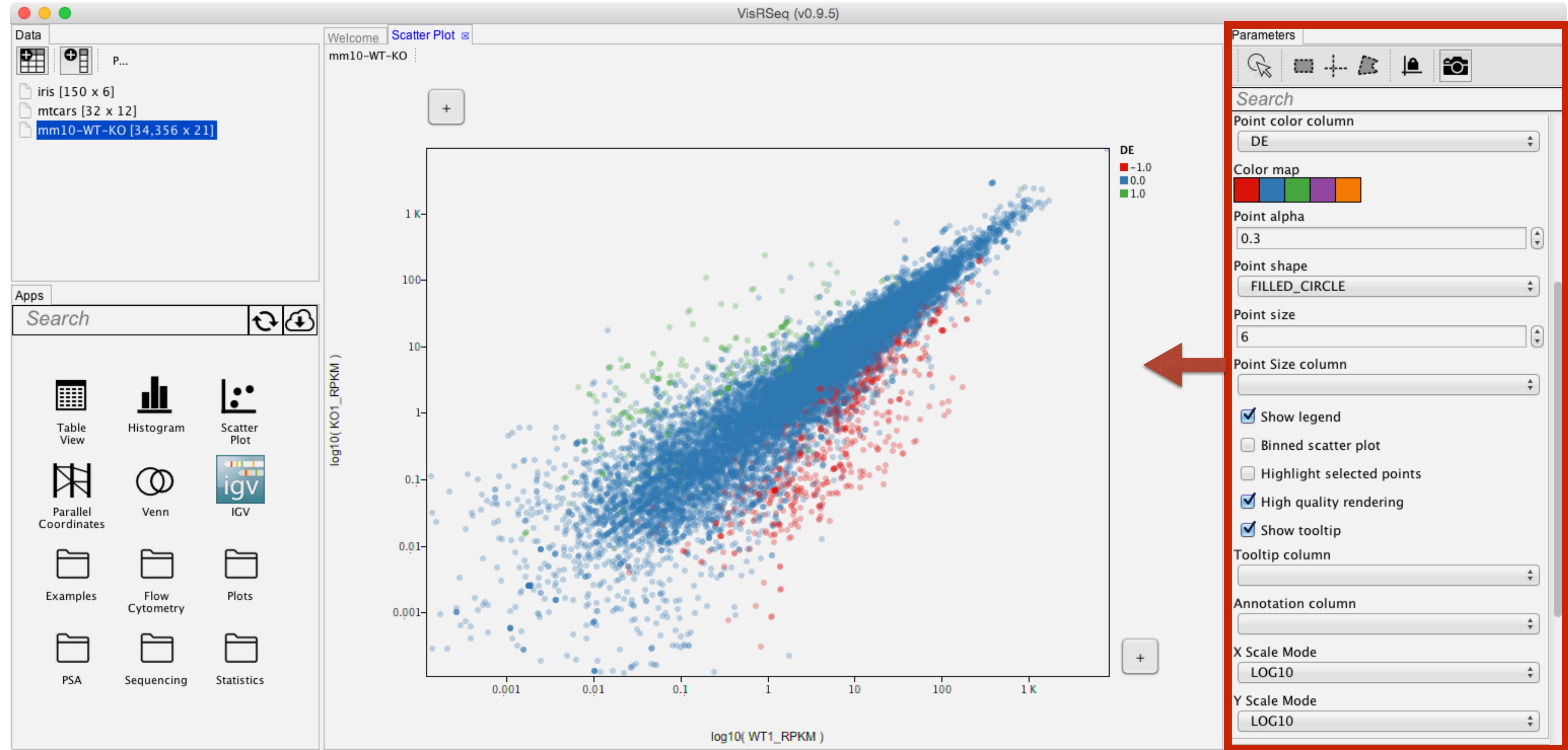
# interface: parameters

parameters

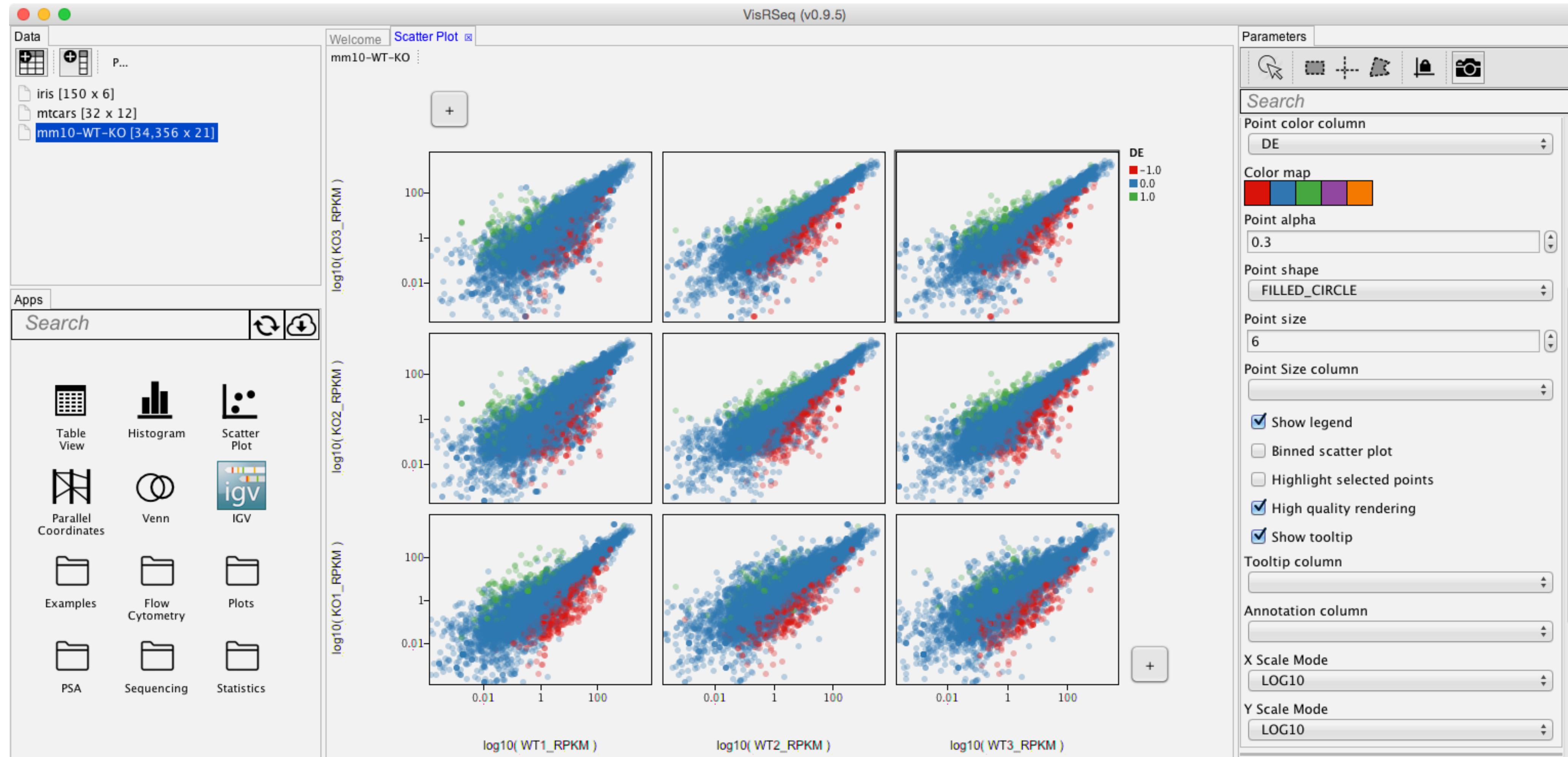


# interface: parameters

parameters

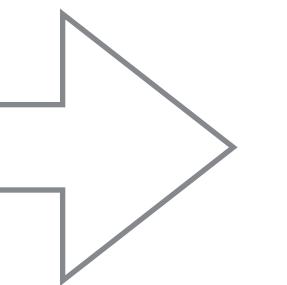
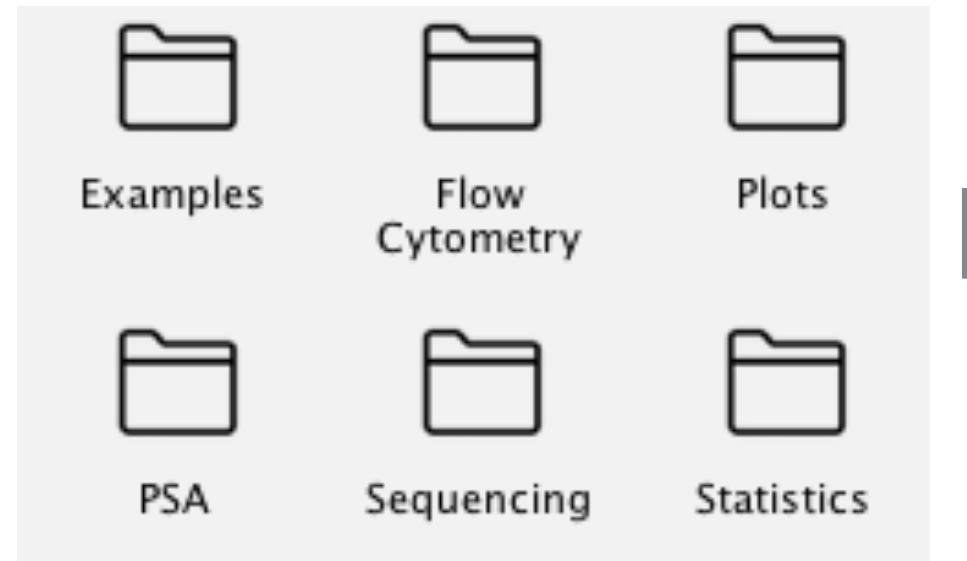


# Apps: interactive

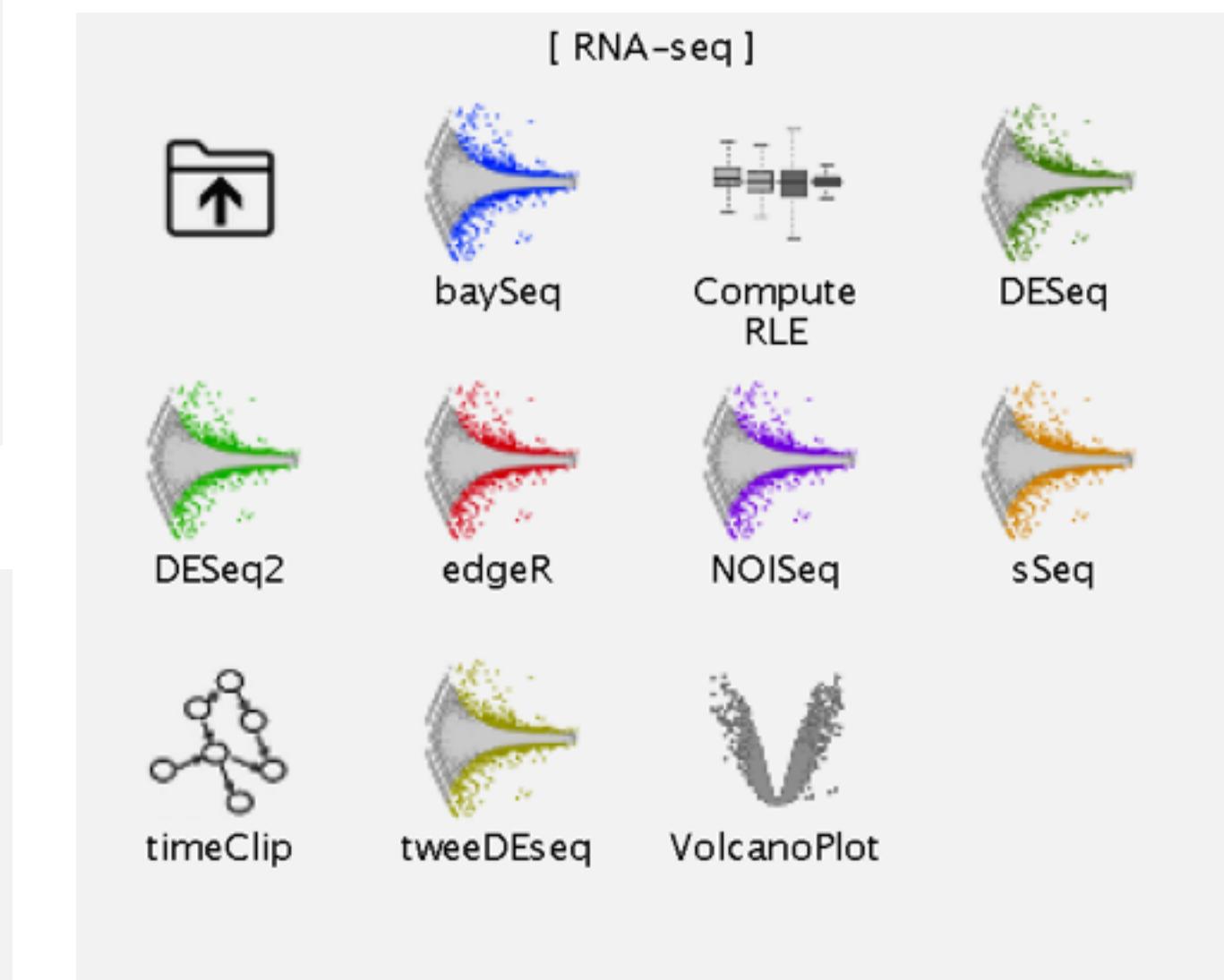
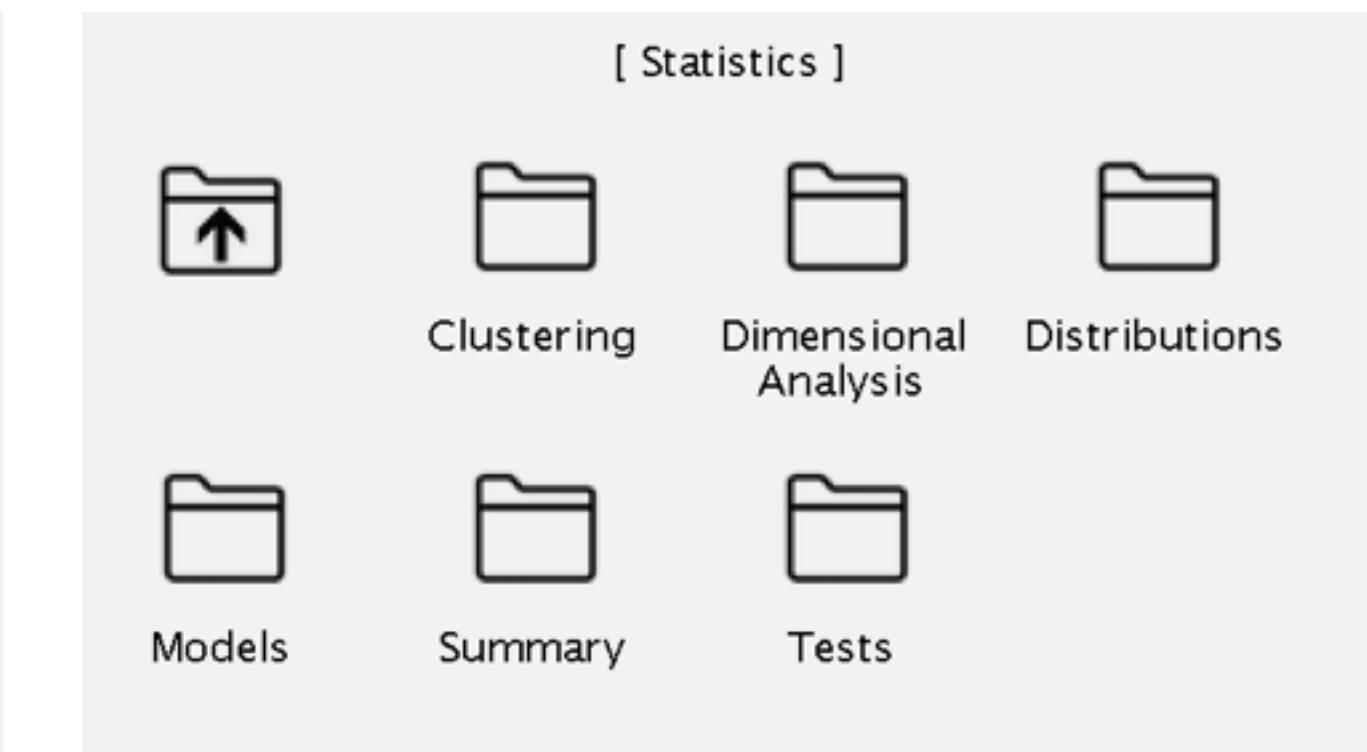
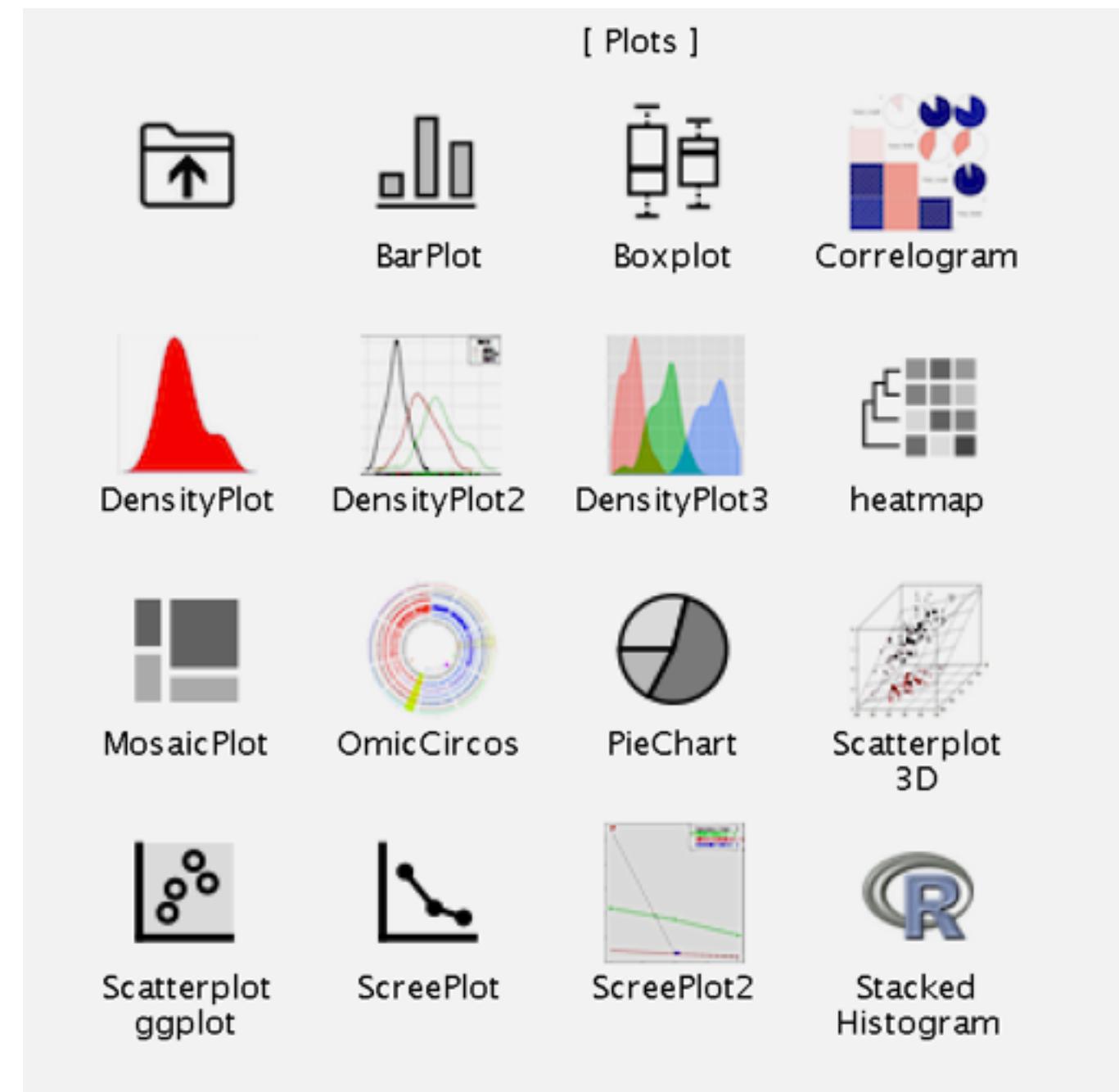
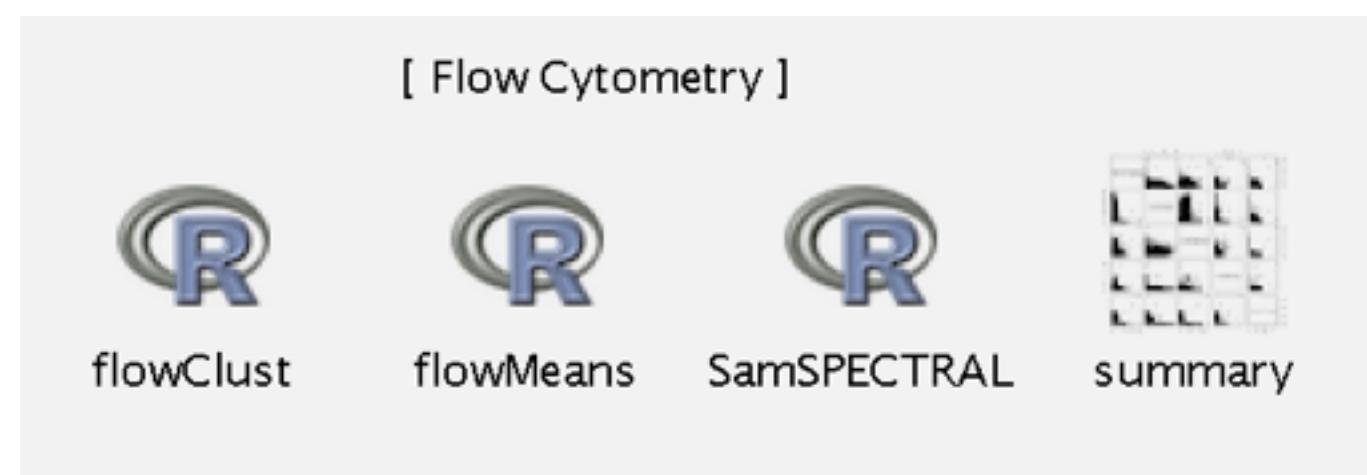
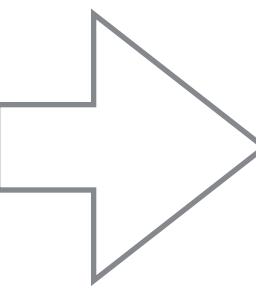
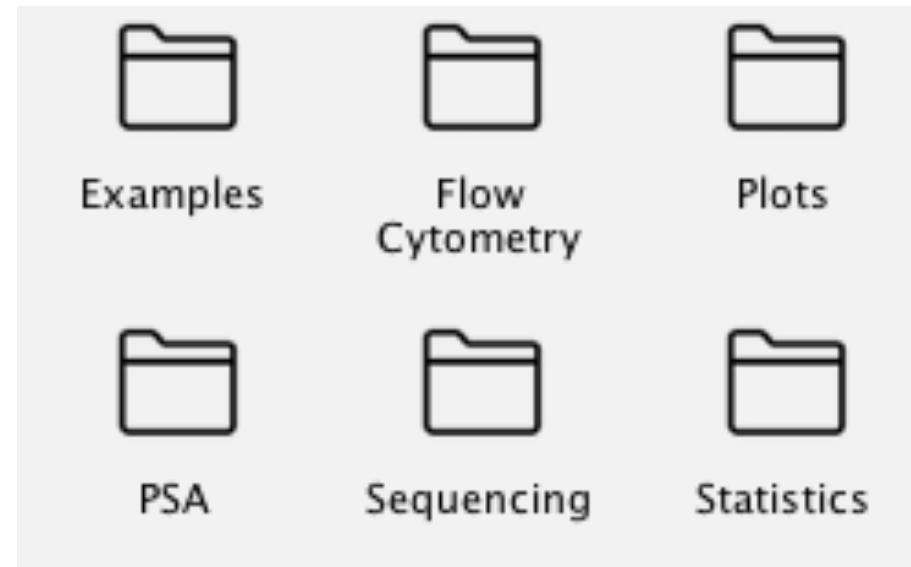


# R - apps

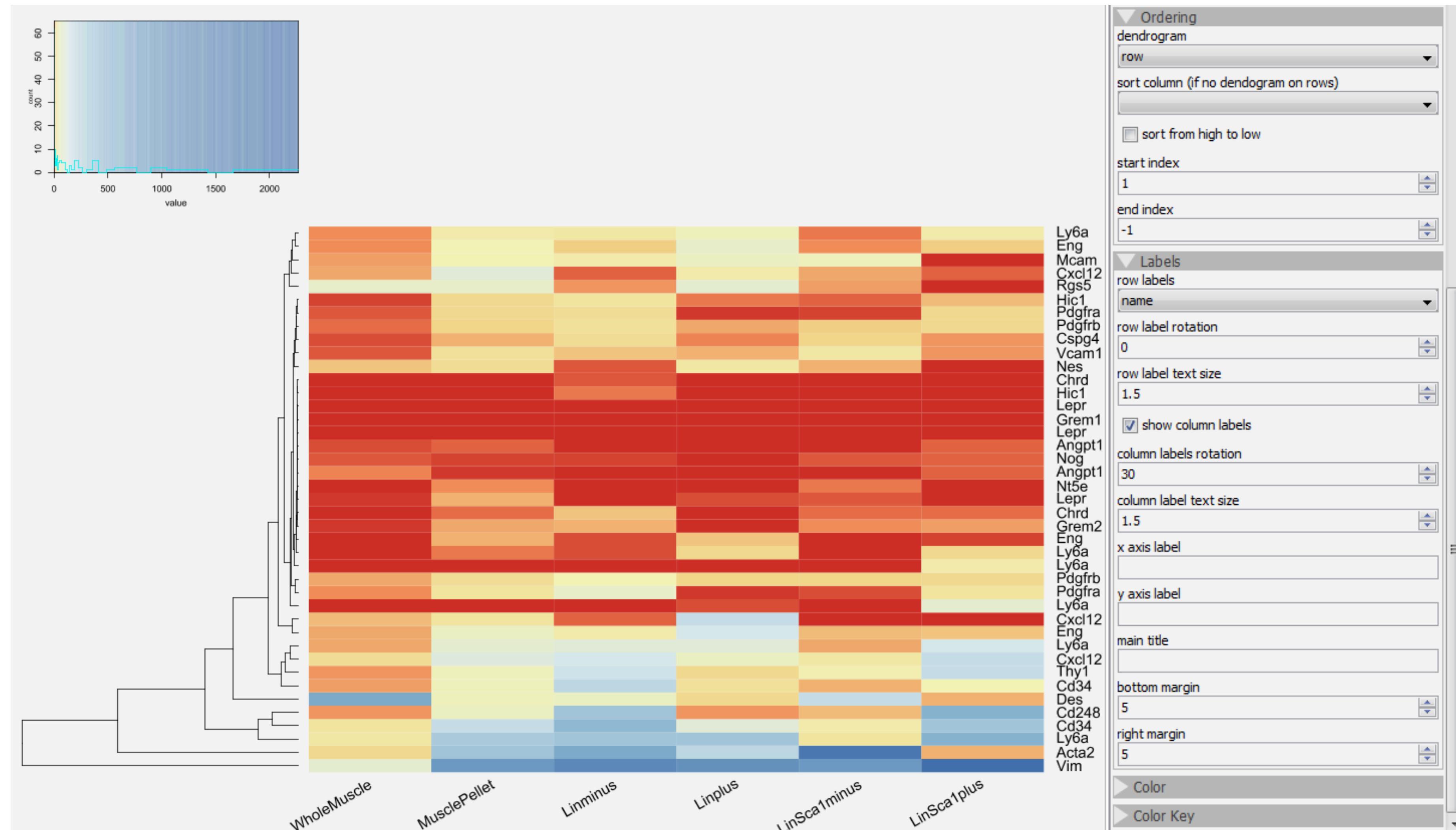
# R Apps



# R Apps



# R – Apps example: heatmap



# Anatomy of an R app

## R code

kmeans.R

```
source("visrutils.R")

visr.applyParameters()

cluster_data<-subset(visr.input, select = param.columns)

output.clusterid <- kmeans(cluster_data, param.k,
                           algorithm = param.algorithm)$cluster
plot(cluster_data, main = param.plot.title,
      col = as.integer(output.clusterid))
```

# Anatomy of an R app

## R code

kmeans.R

```
source("visrutils.R")

visr.applyParameters()

cluster_data<-subset(visr.input, select = param.columns)

output.clusterid <- kmeans(cluster_data, param.k,
                           algorithm = param.algorithm)$cluster
plot(cluster_data, main = param.plot.title,
      col = as.integer(output.clusterid))
```

## parameters

kmeans.json

```
{ "label": "kmeans", "categories": [
  { "variables": {
      "param.columns": {"type": "multi-column-numerical" },
      "param.k": {"type": "int", "default": 3 },
      "param.algorithm": {"items": ["Hartigan-Wong", "Lloyd",
                                    "Forgy", "MacQueen"]},
      "param.plot.title": {"default": "kmeans result" },
      "output.clusterid": {"type": "output-column" }
    }
  }
]}
```

# Anatomy of an R app

## R code

kmeans.R

```
source("visrutils.R")

visr.applyParameters()

cluster_data<-subset(visr.input, select = param.columns)

output.clusterid <- kmeans(cluster_data, param.k,
                            algorithm = param.algorithm)$cluster
plot(cluster_data, main = param.plot.title,
      col = as.integer(output.clusterid))
```

## parameters

kmeans.json

```
{ "label": "kmeans", "categories": [
  { "variables": {
    "param.columns": {"type": "multi-column-numerical"}, 
    "param.K": {"type": "int", "default": 3}, 
    "param.algorithm": {"items": ["Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"]}, 
    "param.plot.title": {"default": "kmeans result"}, 
    "output.clusterid": {"type": "output-column"}
  }}
]}
```

# Anatomy of an R app

## R code

kmeans.R

```
source("visrutils.R")

visr.applyParameters()

cluster_data<-subset(visr.input, select = param.columns)

output.clusterid <- kmeans(cluster_data, param.k
                            algorithm = param.algorithm$cluster
plot(cluster_data, main = param.plot.title,
      col = as.integer(output.clusterid))
```

## parameters

kmeans.json

```
{ "label": "kmeans", "categories": [
  { "variables": {
    "param.columns": {"type": "multi-column-numerical" },
    "param.k": {"type": "int", "default": 3 },
    "param.algorithm": {"items": ["Hartigan-Wong", "Lloyd",
                                 "Forgy", "MacQueen"]},
    "param.plot.title": {"default": "kmeans result" },
    "output.clusterid": {"type": "output-column" }
  }}
]}
```

# Anatomy of an R app

## R code

kmeans.R

```
source("visrutils.R")

visr.applyParameters()

cluster_data<-subset(visr.input, select = param.columns)

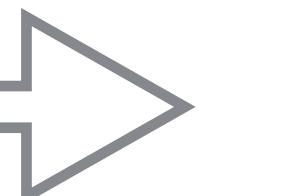
output.clusterid <- kmeans(cluster_data, param.k,
                           algorithm = param.algorithm)$cluster

plot(cluster_data, main = param.plot.title,
     col = as.integer(output.clusterid))
```

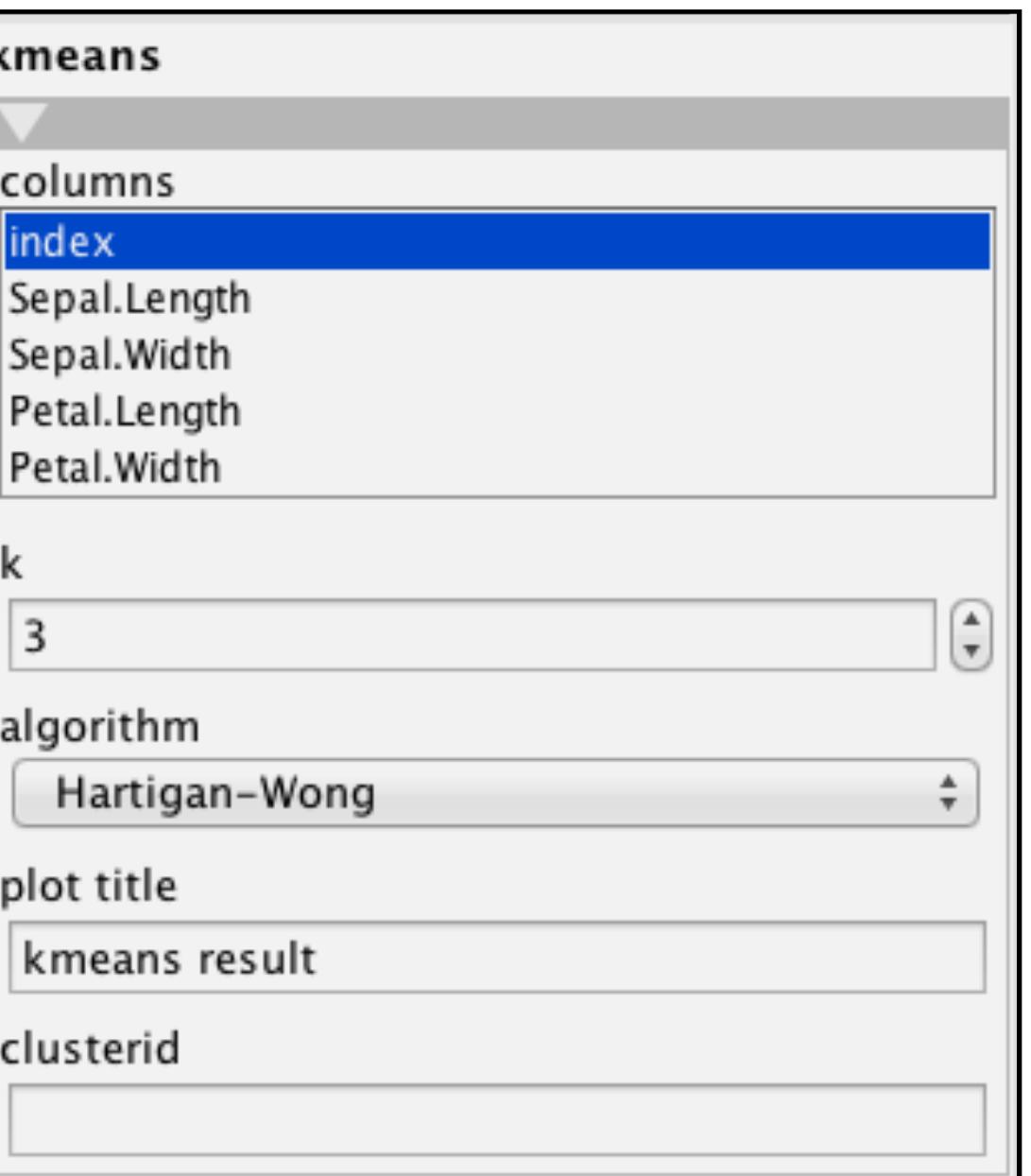
## parameters

kmeans.json

```
{ "label": "kmeans", "categories": [
  { "variables": {
      "param.columns": {"type": "multi-column-numerical" },
      "param.k": {"type": "int", "default": 3 },
      "param.algorithm": {"items": ["Hartigan-Wong", "Lloyd",
                                    "Forgy", "MacQueen"]},
      "param.plot.title": {"default": "kmeans result"},
      "output.clusterid": {"type": "output-column"}
    }
]}]
```



## auto generated UI



# Anatomy of an R app

## R code

kmeans.R

```
source("visrutils.R")

visr.applyParameters()

cluster_data<-subset(visr.input, select = param.columns)

output.clusterid <- kmeans(cluster_data, param.k,
                           algorithm = param.algorithm)$cluster

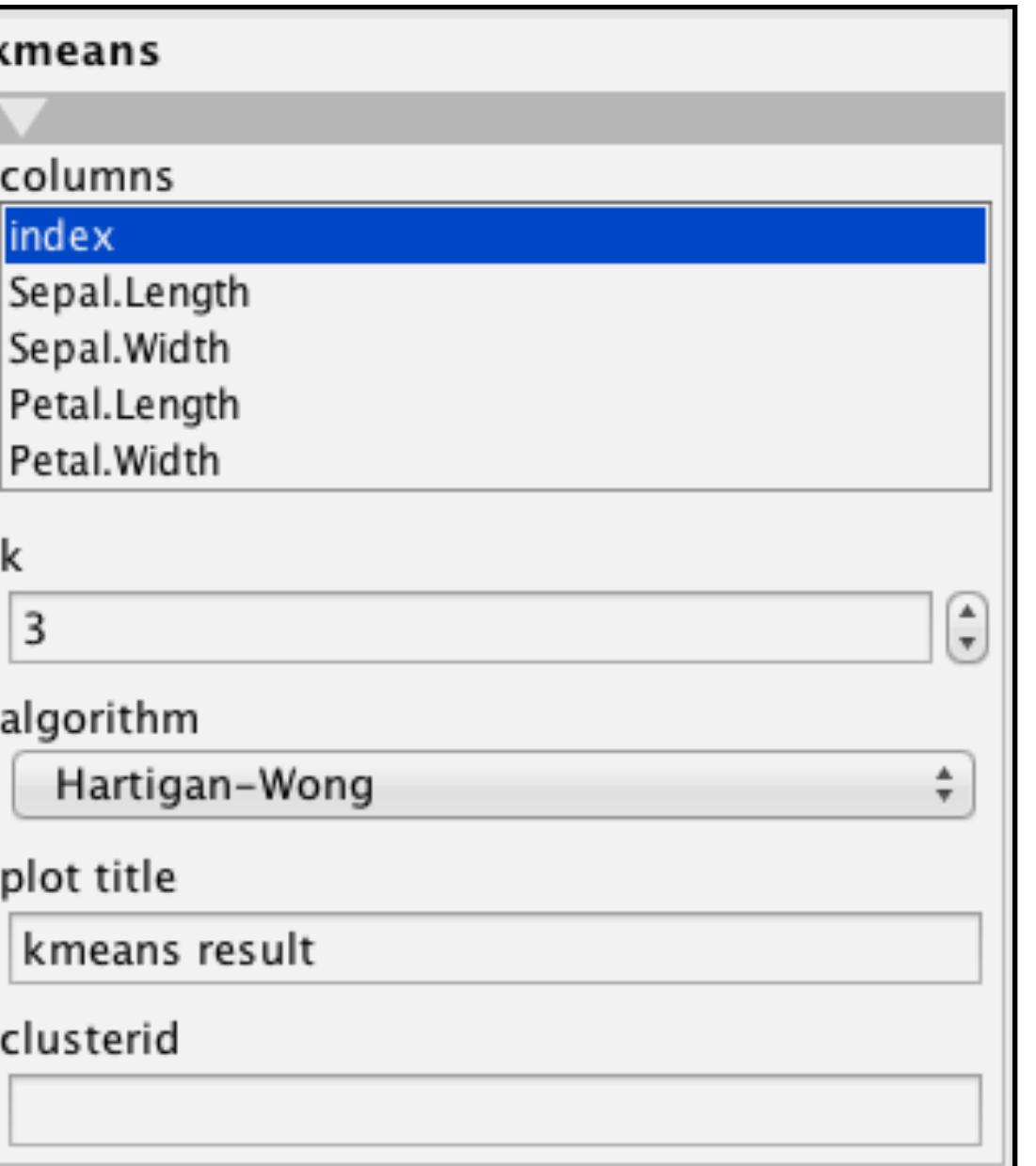
plot(cluster_data, main = param.plot.title,
     col = as.integer(output.clusterid))
```

## parameters

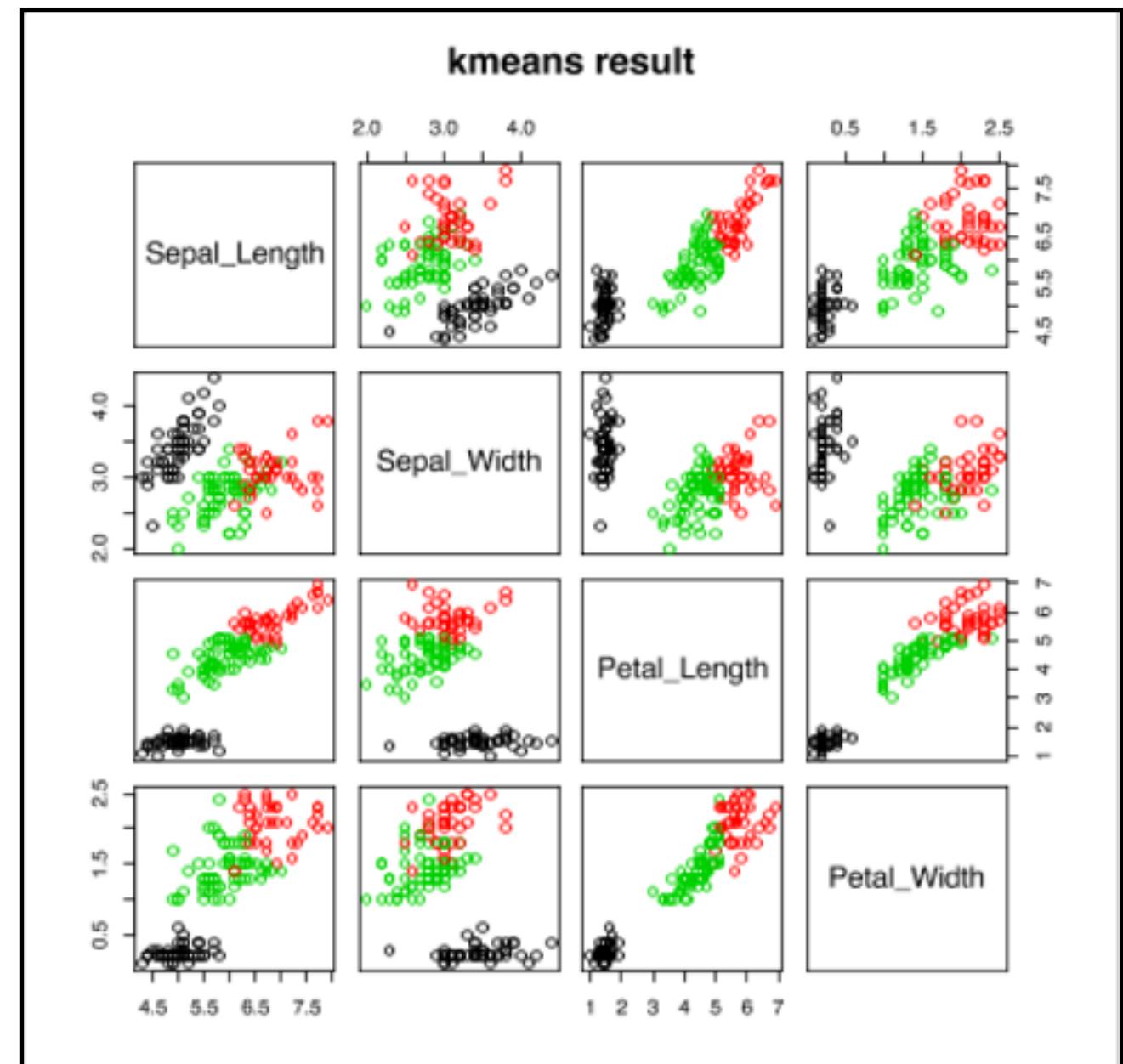
kmeans.json

```
{ "label": "kmeans", "categories": [
  { "variables": {
      "param.columns": { "type": "multi-column-numerical" },
      "param.k": { "type": "int", "default": 3 },
      "param.algorithm": { "items": ["Hartigan-Wong", "Lloyd",
                                    "Forgy", "MacQueen"] },
      "param.plot.title": { "default": "kmeans result" },
      "output.clusterid": { "type": "output-column" }
    }
  }
]}
```

## auto generated UI



## output



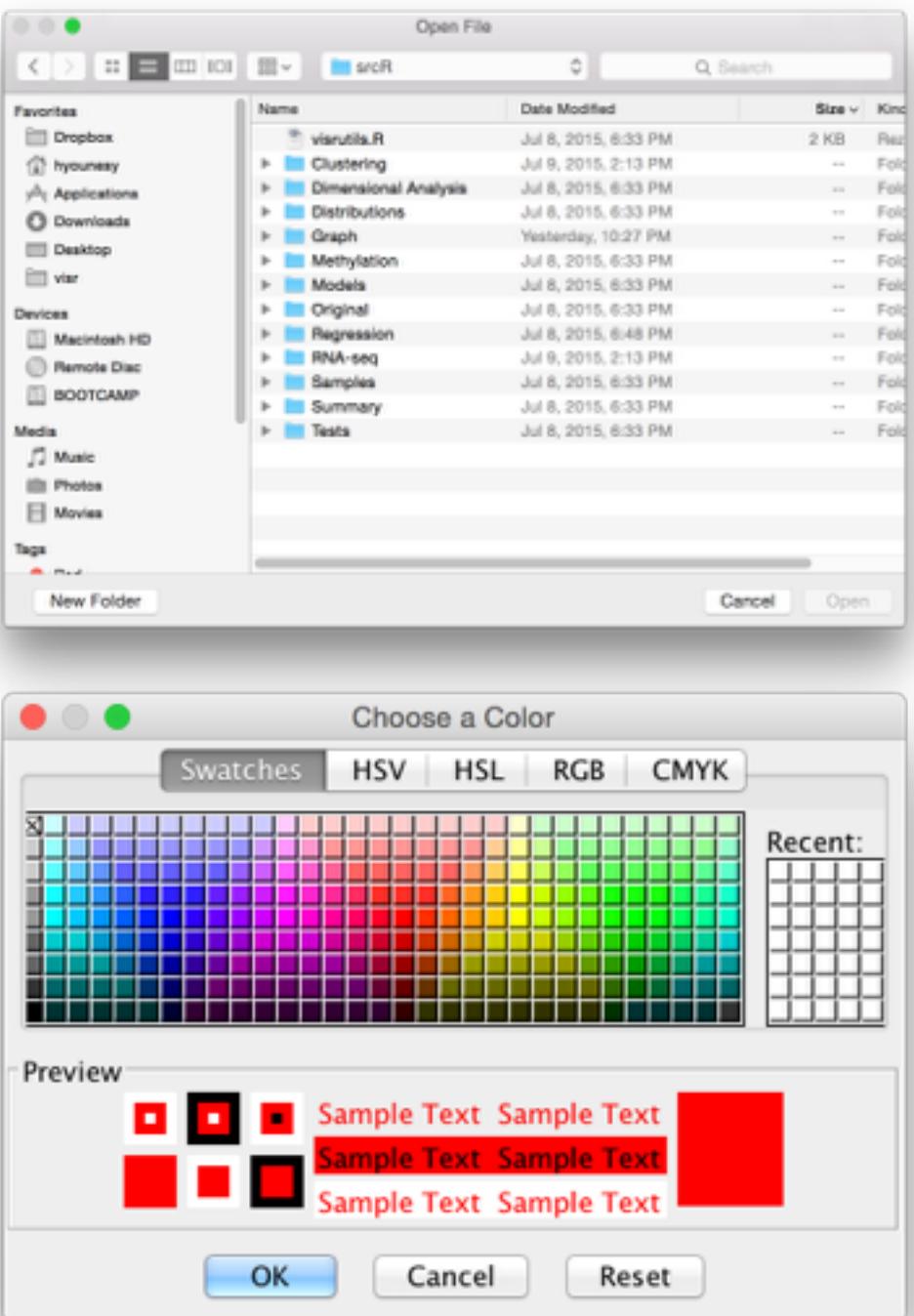
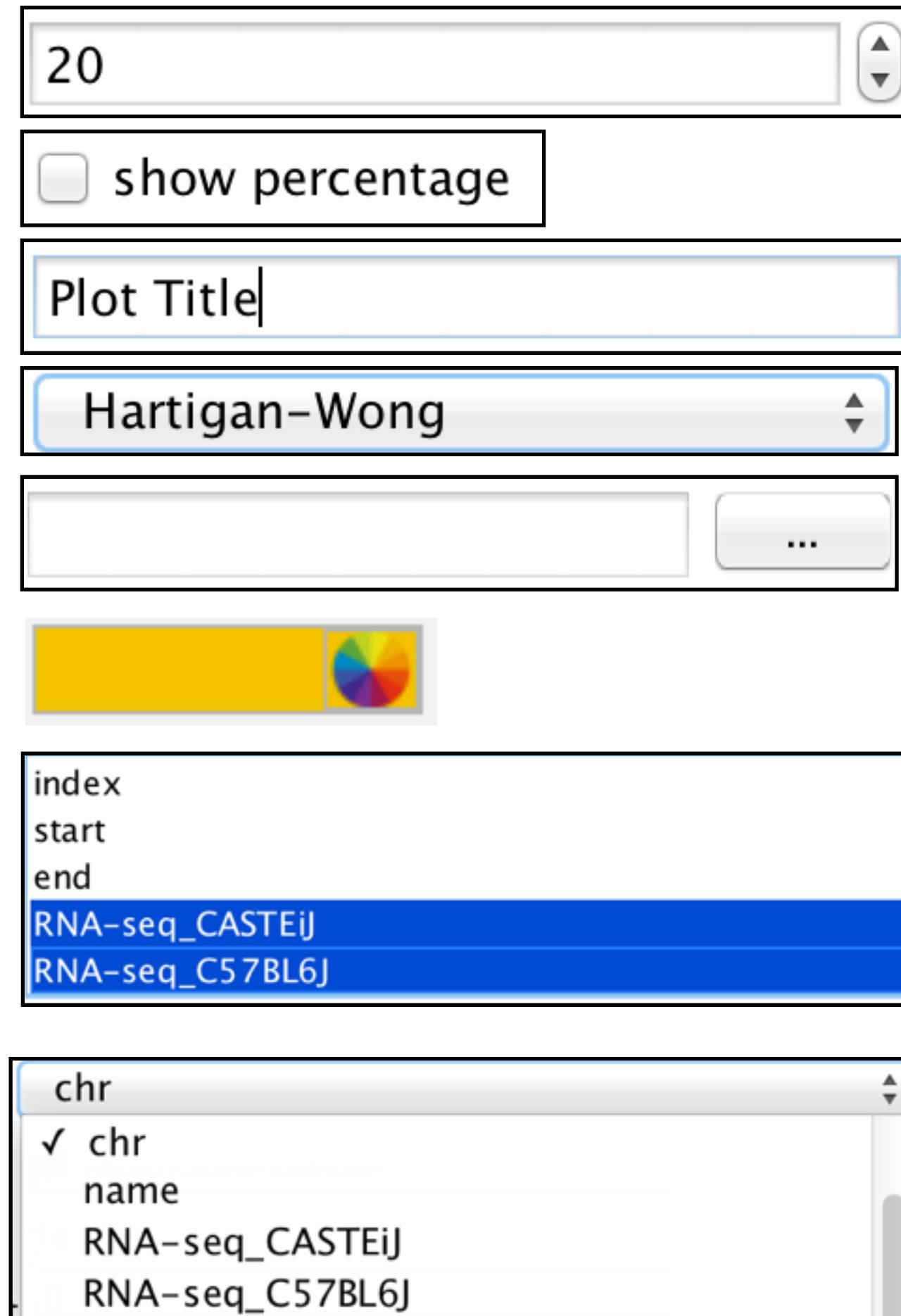
# creating R apps: variable types

**variable type**

---

- int
- double
- boolean
- string
- string with items
- filename
- color
- range-int
- range-double
- column
- column-numerical
- multi-column
- multi-column-numerical
- output-column
- output-table

---



```
visr.message(message, type)
```

# creating R apps

- Entire app may also be coded and debugged in R (/ RStudio)
- The .json file will be generated from the R code

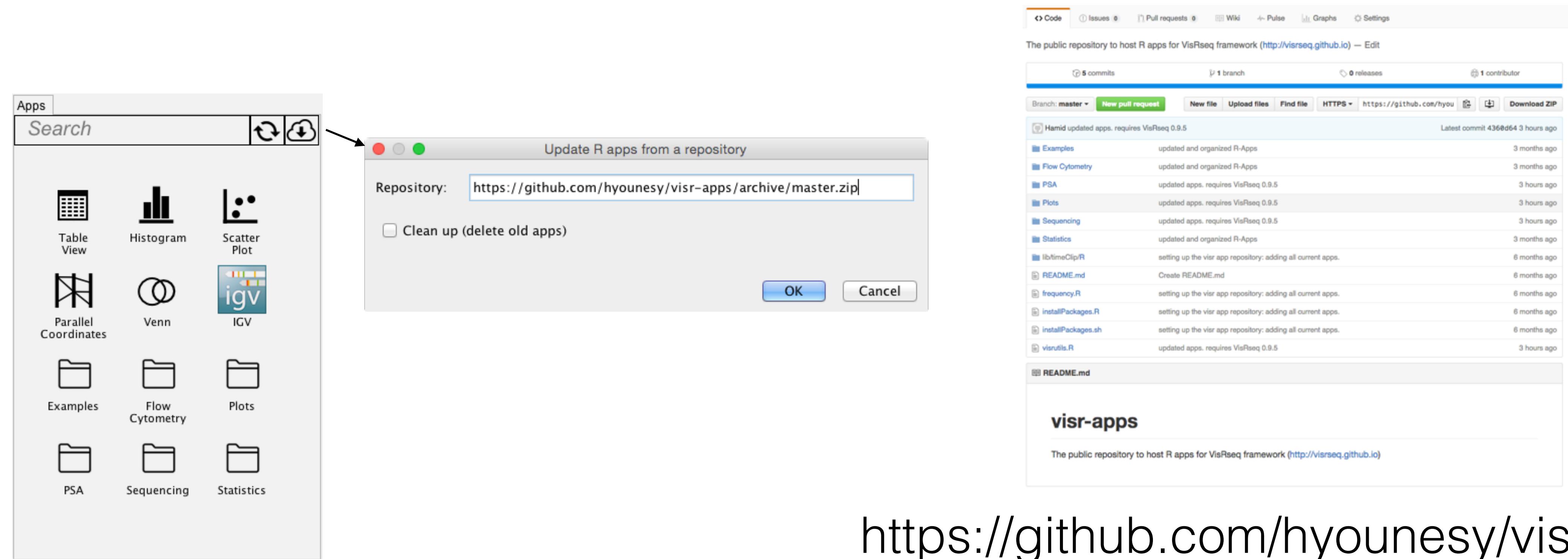
# creating R apps

```
source("visrutils.R")

# parameters
visr.app.start("Simple Kmeans", debugdata = iris)
visr.category("clustering parameters")
visr.param("columns", type = "multi-column-numerical",
           debugvalue = c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width"))
visr.param("k", default = 3)
visr.param("algorithm", items = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"))
visr.category("output")
visr.param("plot.title", default = "kmeans results")
visr.param("output.clusterid", type = "output-column")
visr.app.end(printjson=TRUE)
visr.applyParameters()

# kmeans code
cluster_data<-subset(visr.input, select = visr.param.columns)
visr.param.output.clusterid <- kmeans(cluster_data,
                                       centers = visr.param.k,
                                       algorithm = visr.param.algorithm)$cluster
plot(cluster_data, main = visr.param.plot.title,
      col = as.integer(visr.param.output.clusterid))
```

# creating R apps: sharing



<https://github.com/hyounesy/visr-apps/>

# Demo

# Demo

The screenshot displays the VisRSeq software interface (version v0.73.4) running on a Mac OS X system. The main window is divided into several panels:

- Data Panel:** Shows a table titled "Allele-specific [28,373 x 9]" containing genomic data. The columns include index, chr, start, end, strand, ID, name, RNA-seq..., and RNA-seq... . The data shows various genes across chromosomes 1 through 22.
- Apps Panel:** A sidebar titled "Search [ RNA-seq ]" lists various RNA-seq analysis tools: baySeq, Compute RLE, DESeq, DESeq2, edgeR, NOISeq, sSeq, tweeDseq, and VolcanoPlot. The DESeq icon is highlighted with a red border.
- Table Panel:** A detailed view of the "Allele-specific" data table, showing rows 1 through 32.
- Parameters Panel:** A panel on the right side of the table view.
- IGV Panel:** A genome browser interface for the Mouse mm9 genome. It shows tracks for CAST/Eij and C57BL/6J, with a Refseq genes track at the bottom. The genome scale ranges from 1 to X (chromosome 1 to X). A play button icon is overlaid on the IGV panel.

Your assignment if you  
choose to accept it...

# Data

- 58788 Rows
- **title:** Title of the movie.
- **year:** Year of release.
- **budget:** Total budget (if known) in US dollars
- **length:** Length in minutes.
- **mpaa:** MPAA rating.
- **rating:** Average IMDB user rating.
- **votes:** Number of IMDB users who rated this movie.
- **r1-r10:** Multiplying by ten gives percentile (to nearest 10%) of users who rated this movie a 1.
- **action, animation, comedy, drama, documentary, romance, short:** Binary variables representing if movie was classified as belonging to that genre.

# Assignments

- Two categories:
  1. Data Analysis
  2. App development / improvement.
- You may participate in either or both categories.

# Assignment 1: Data Analysis

- Create an infographic about the movie dataset.
- Single page PDF document or PNG image.
- Plots should be generated only using VisRseq.
- Ok to use other software (e.g. MS Word, MS Powerpoint, Photoshop, etc.) to arrange several plots and to add additional text or graphics.

# Assignment 1: Data Analysis

- The submissions will be evaluated based on:
  - functionality / the amount of information content
  - clarity (i.e. ease of interpretation)
  - usability / interaction possibilities
  - interesting findings
- Describe your interesting/surprising findings in words as well.

# Assignment 2: App Development

- Develop a new R-App or modify and improve an existing R-App.
- Tutorial: <https://github.com/hyounesy/bioc2016.visrseq>
- Goal is to enhance the current analytical power of VisRseq to allow getting more or improved results.
- App(s) may add new computational functionality (e.g. new classification method) or new plots.
- It is still in the context of movie dataset, but the app should be designed such that it can be used with any tabular data.

# Assignment 2: App Development

- Submit the .R and .json files for the new or improved app.
- App(s) should be functional in the current VisRseq version.
- Include a .pdf or .md document explaining the app functionality and example output.
- If your app is selected to be included in the VisRseq framework, you will be credited in the credits section of the webpage.

# Assignment Submission

- Due by Sunday, July 17th @11:59PM Beijing Time (CST).
- Results will be announced and discussed on Monday July 18th.
- Submission using a **git pull request** of:
  - directory with your full name (Firstname\_Lastname)
  - containing your submissions
  - a readme.txt (or readme.md) file, specifying the category(ies) you are submitting for and any additional information about your submission.

# conclusions

# limitations

- Scalability: few million rows and 30-40 columns of sequencing data
- cannot run R-apps in parallel
- desktop only

# future work

- general purpose framework: Apps for other domains (e.g. Proteomics, etc.)
- workflow designer : link several apps to create “macro” apps.
- reproducibility: provenance / parameter exploration

# acknowledgement

- Developers: Xu Fan, Kathy Cheng, Joseph Poper
- BC Genome Sciences Centre
- Lorincz lab, Underhill lab, Rossi lab, Mager lab

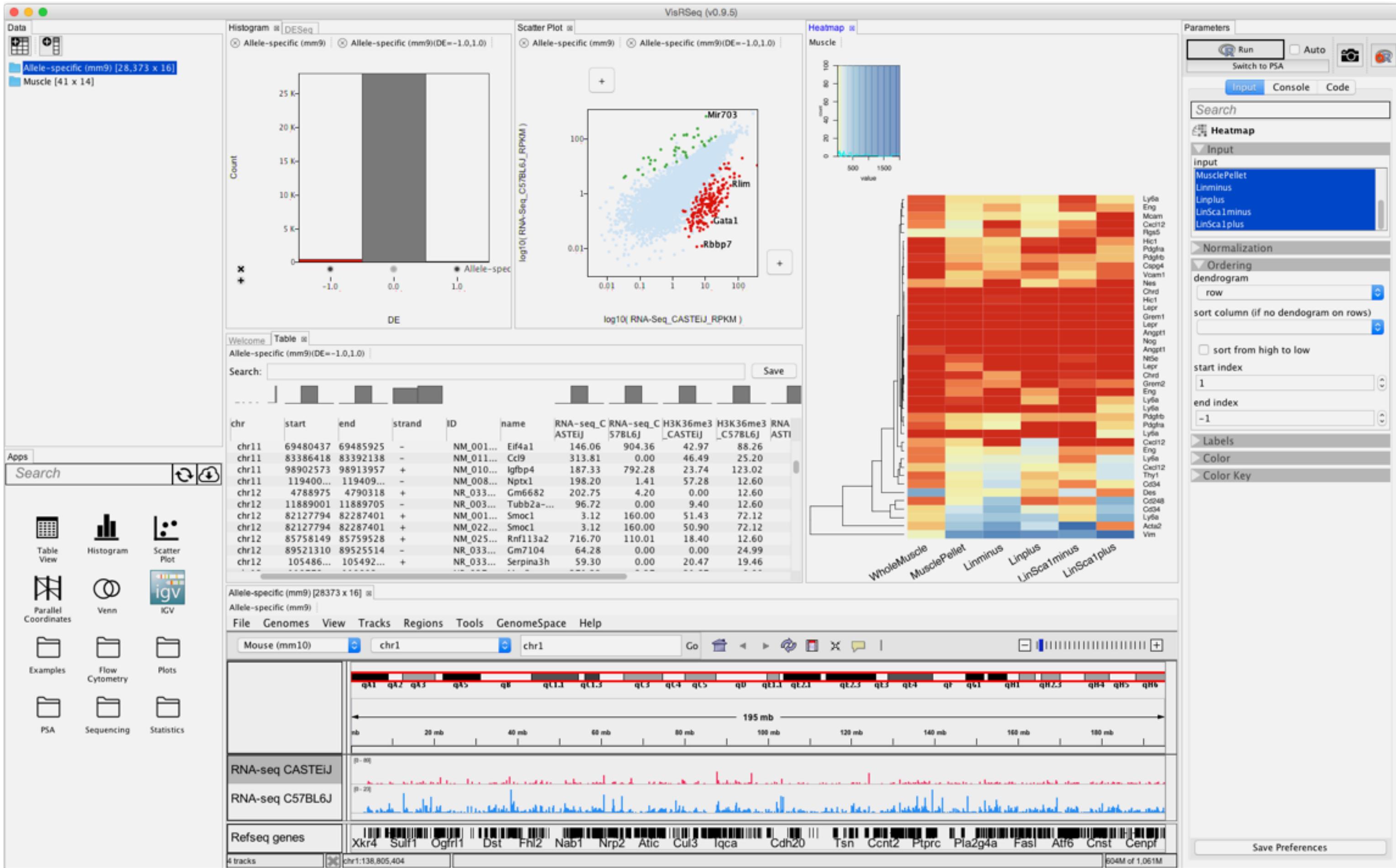


BC Cancer Agency  
CARE + RESEARCH



**NSERC**  
**CRSNG**

# questions?



[visrseq.github.io](https://visrseq.github.io)  
(please email for latest version)