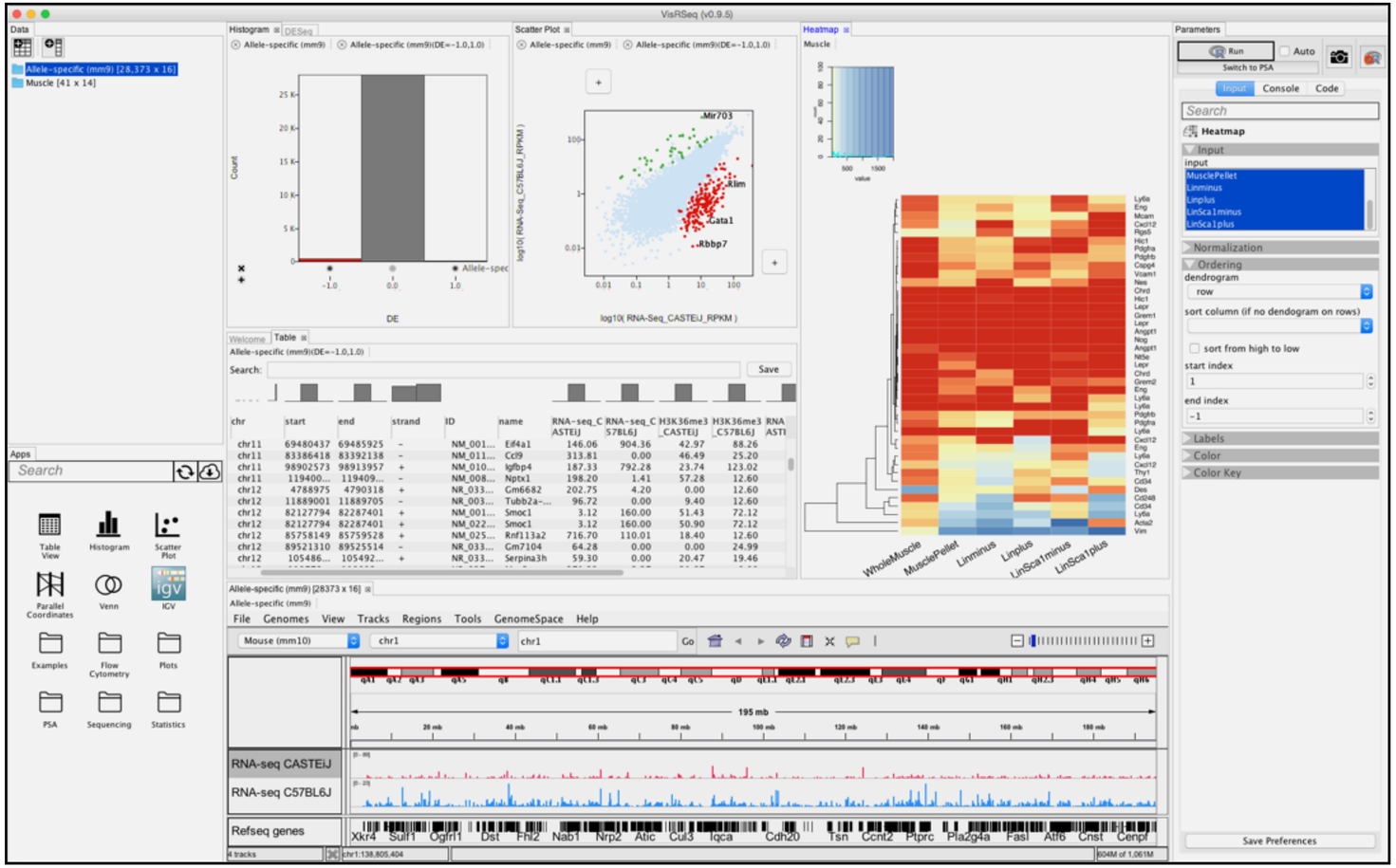
VisRseq: R-based visual analytics software for sequencing data

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*

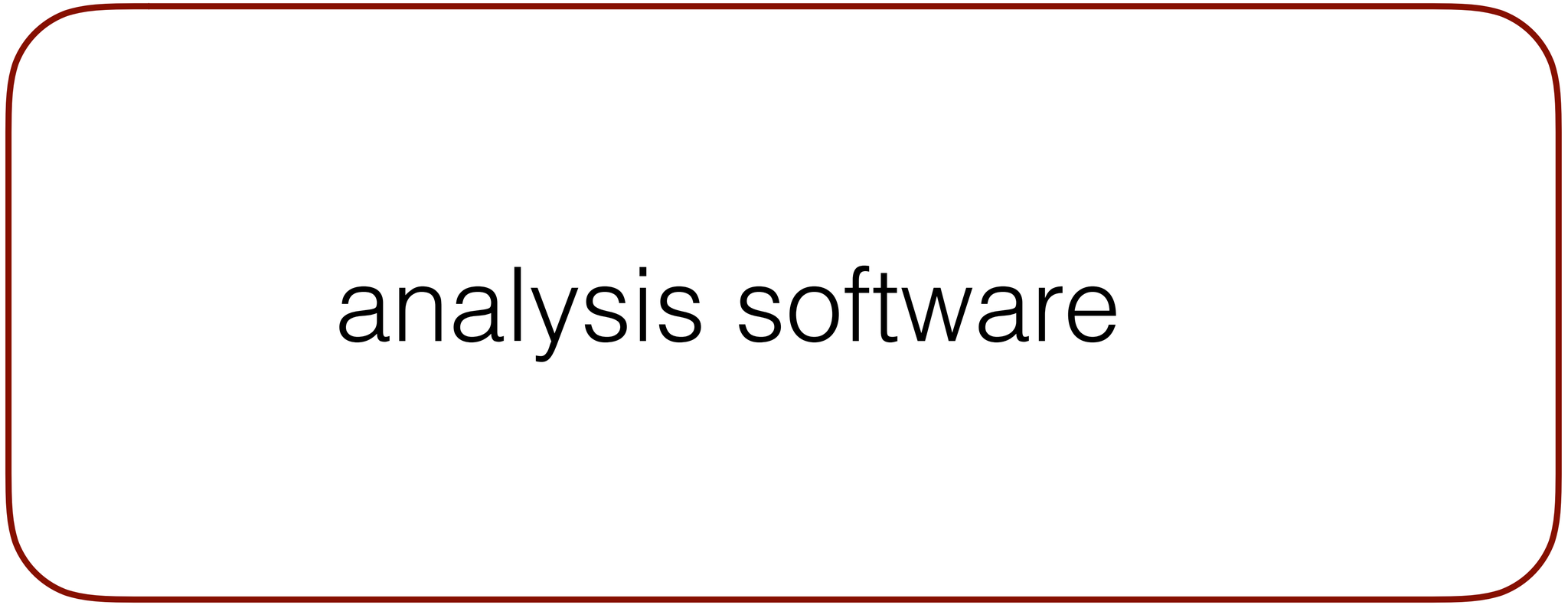
Overview

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

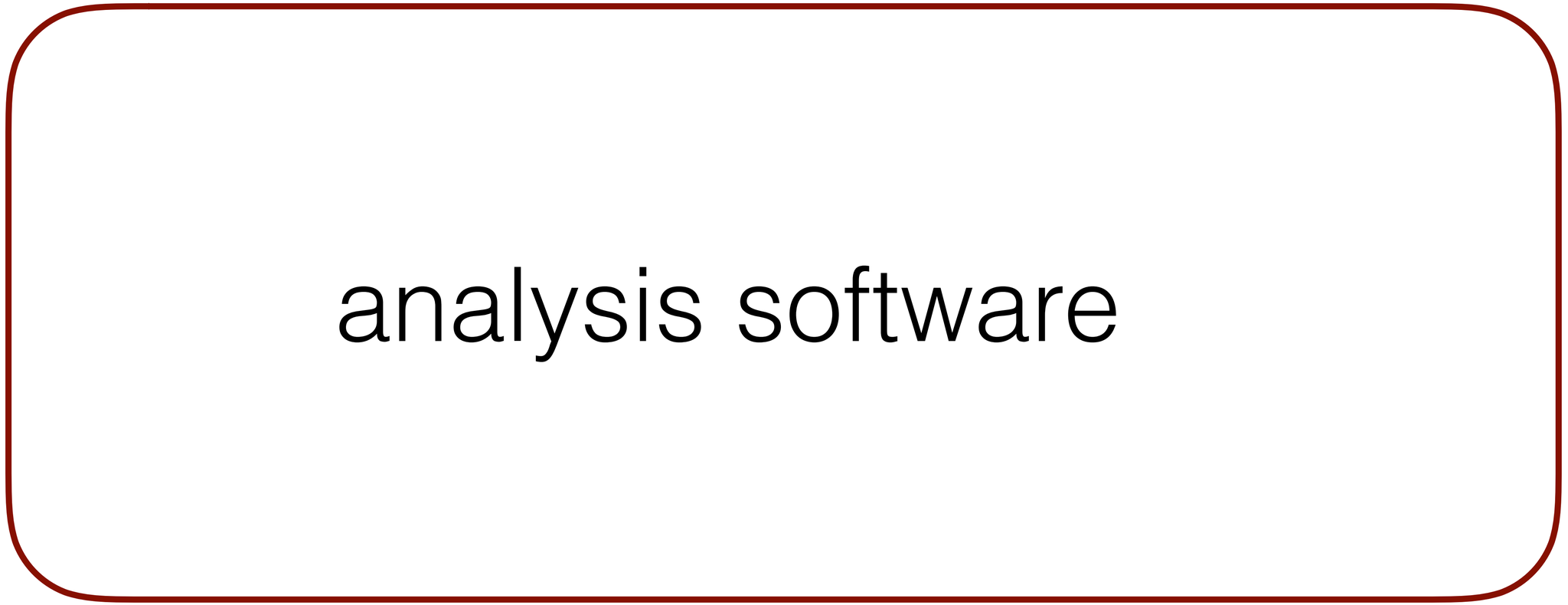


motivation

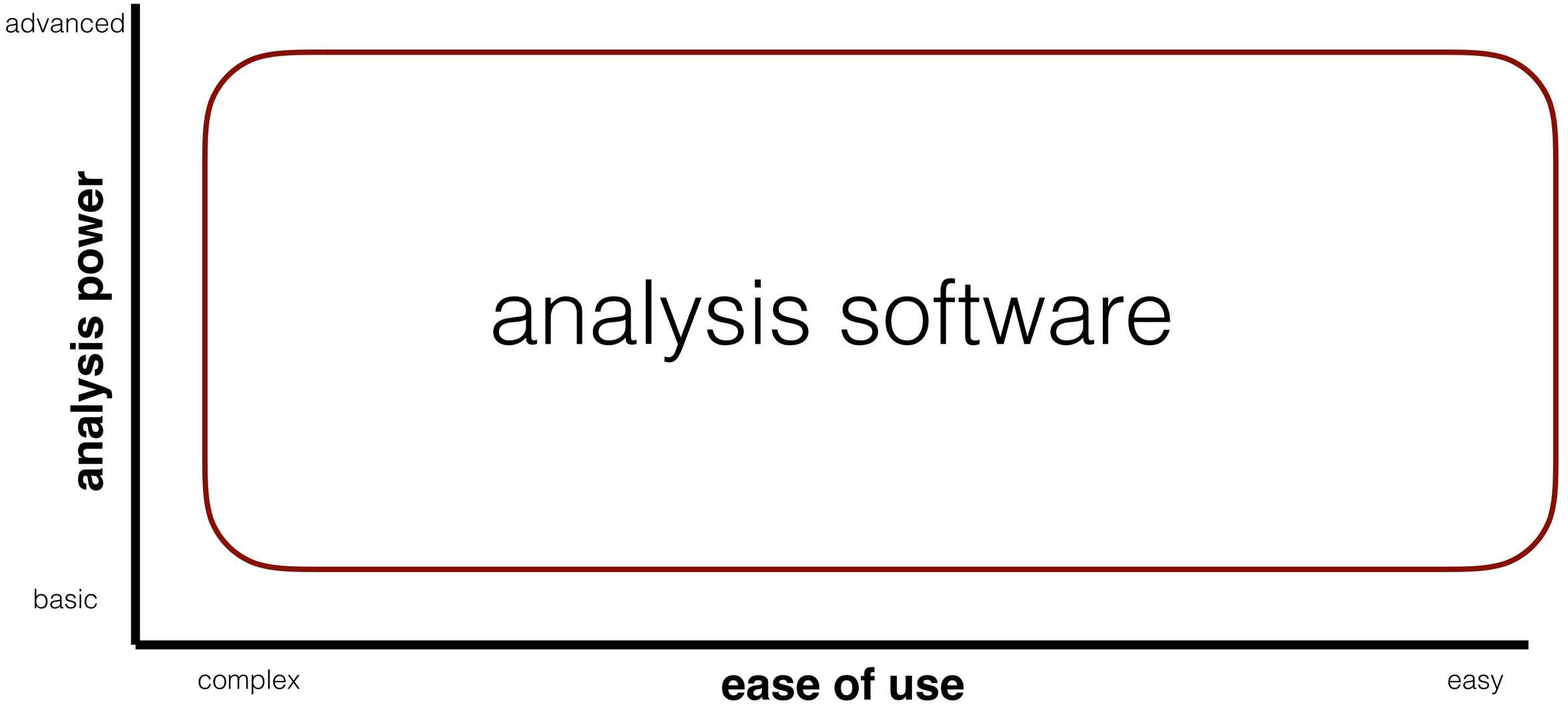
motivation



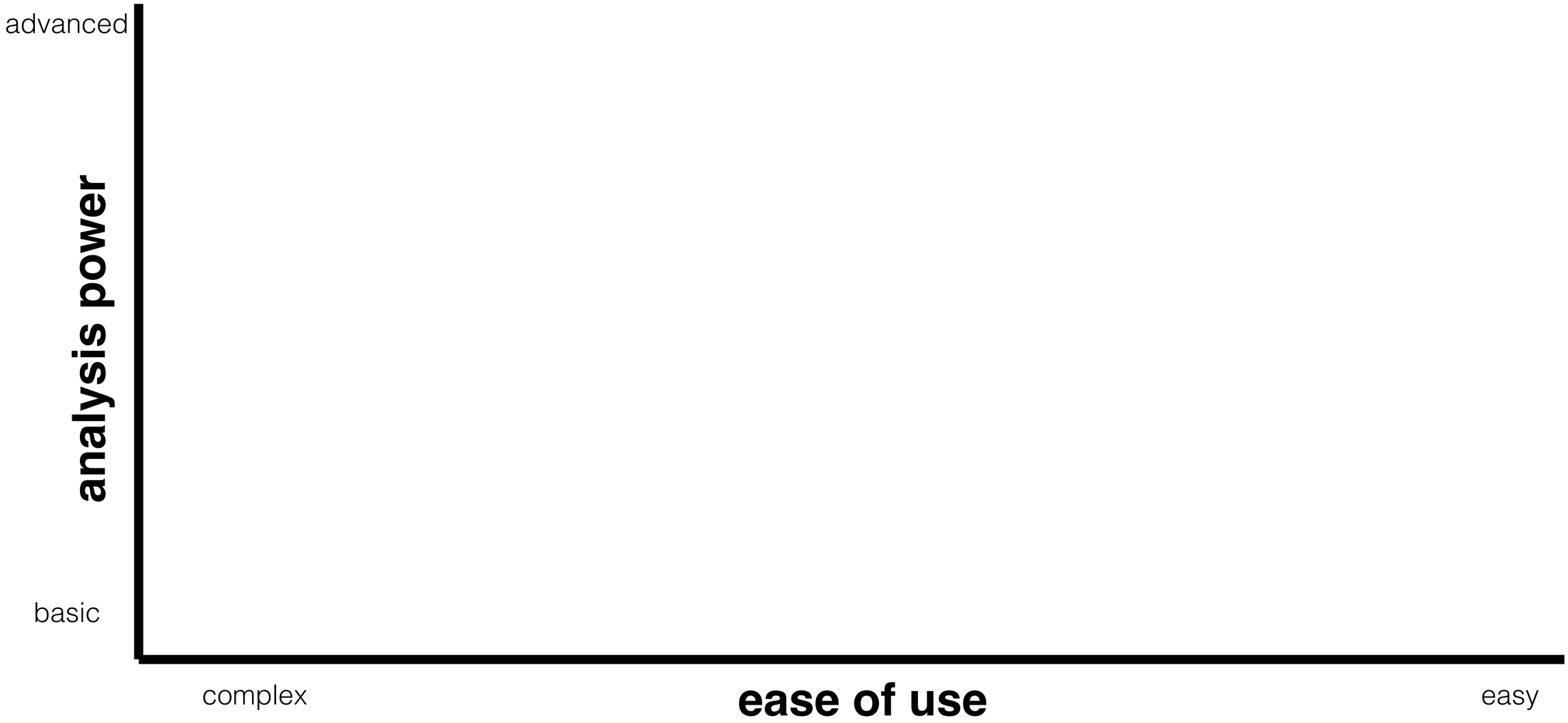
motivation



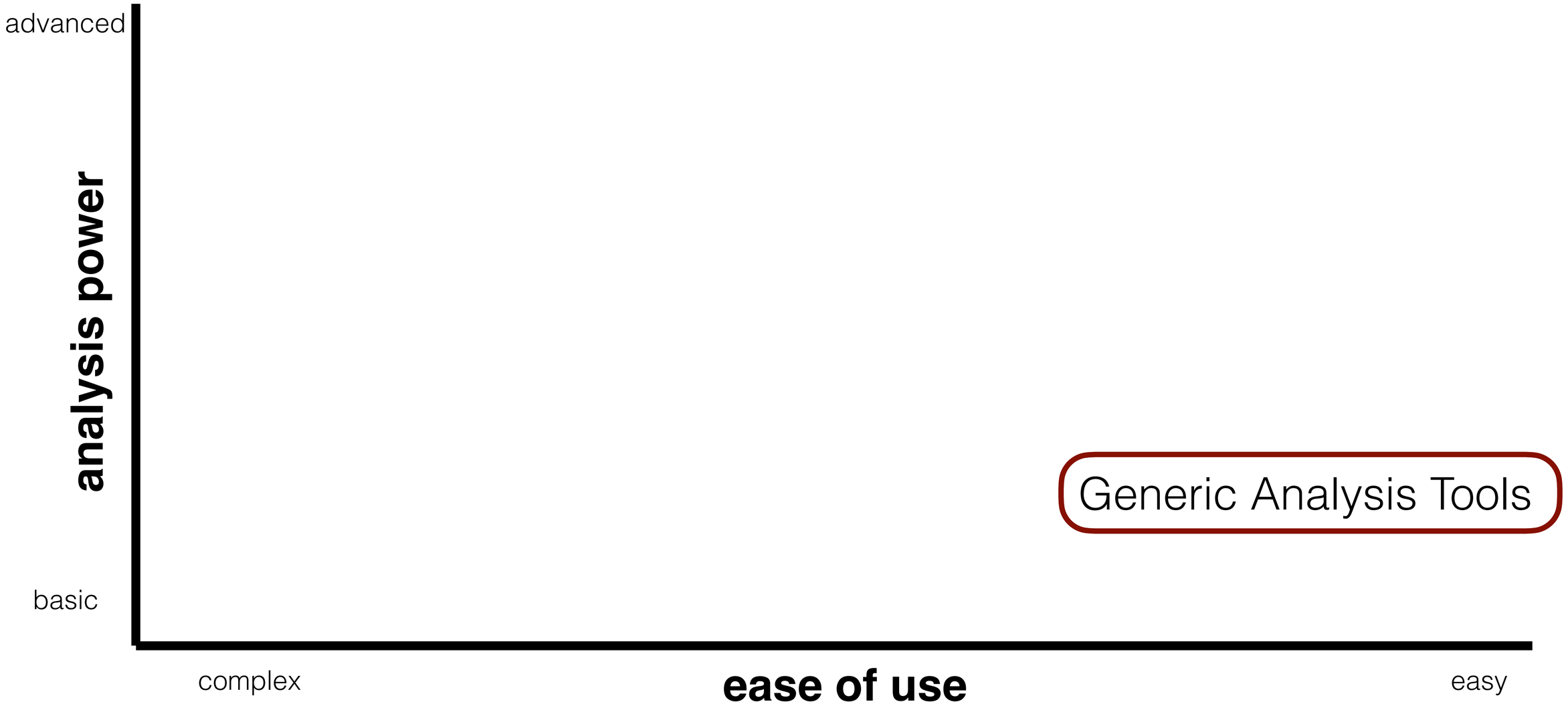
motivation


advanced
analysis power
basic
analysis software
complex
ease of use
easy

motivation



motivation

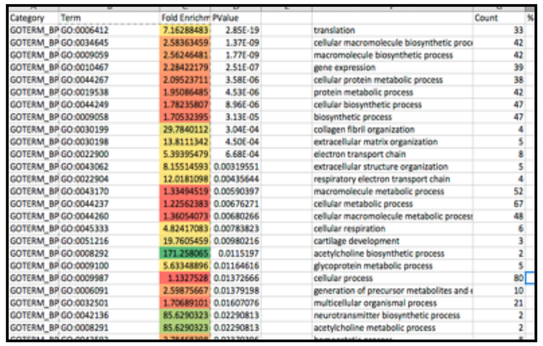

advanced
analysis power
basic
Generic Analysis Tools
complex
ease of use
easy

motivation

advanced

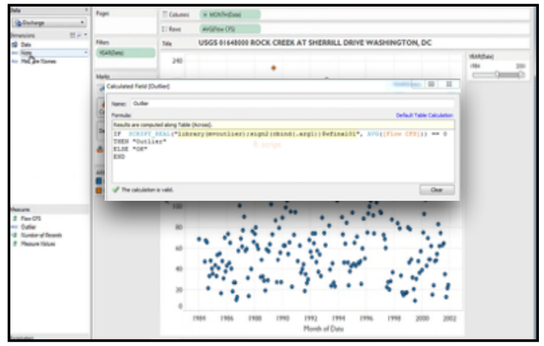
analysis power

basic



Category	Term	Fold Enrichm	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		13.8111342	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	0.00435644	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 process	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:0009100		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	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

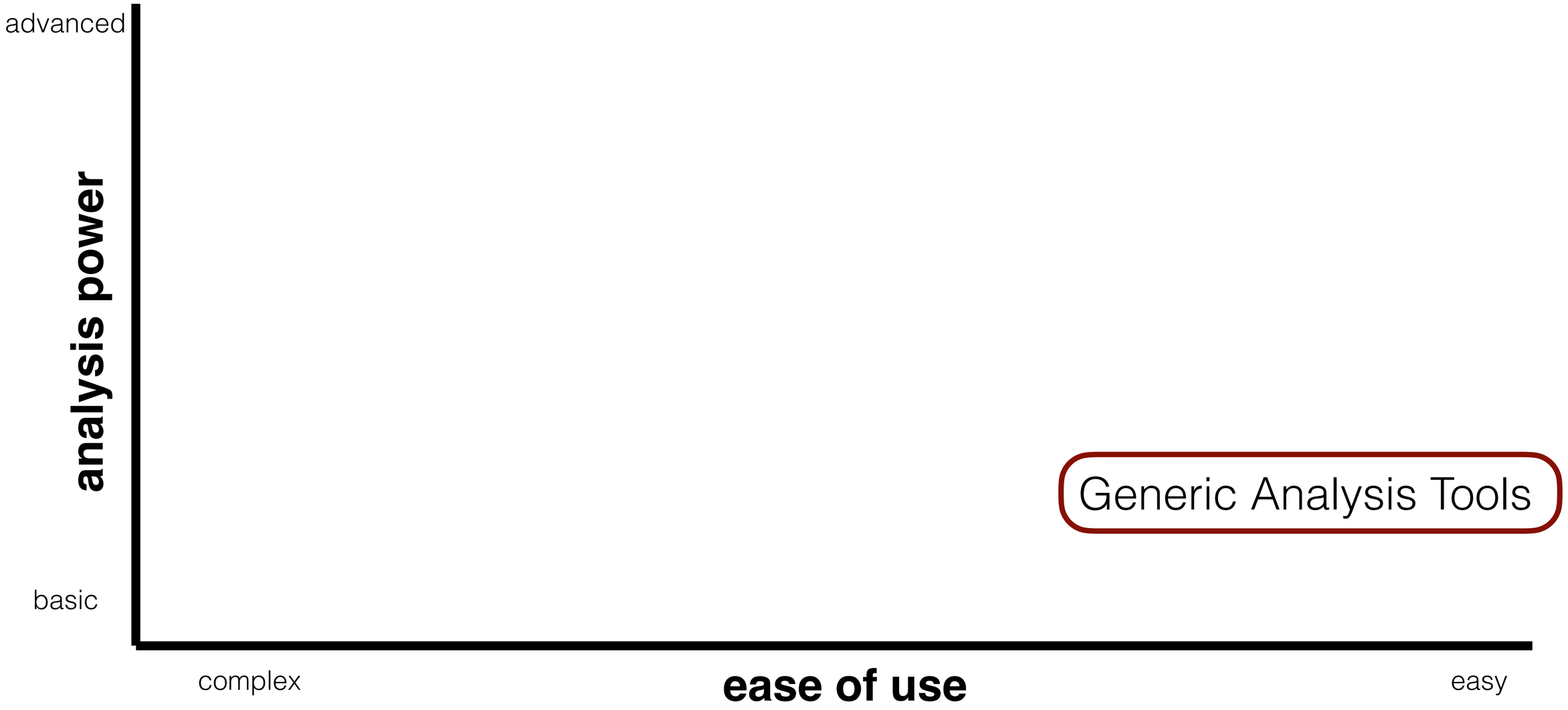
Generic Analysis Tools

complex

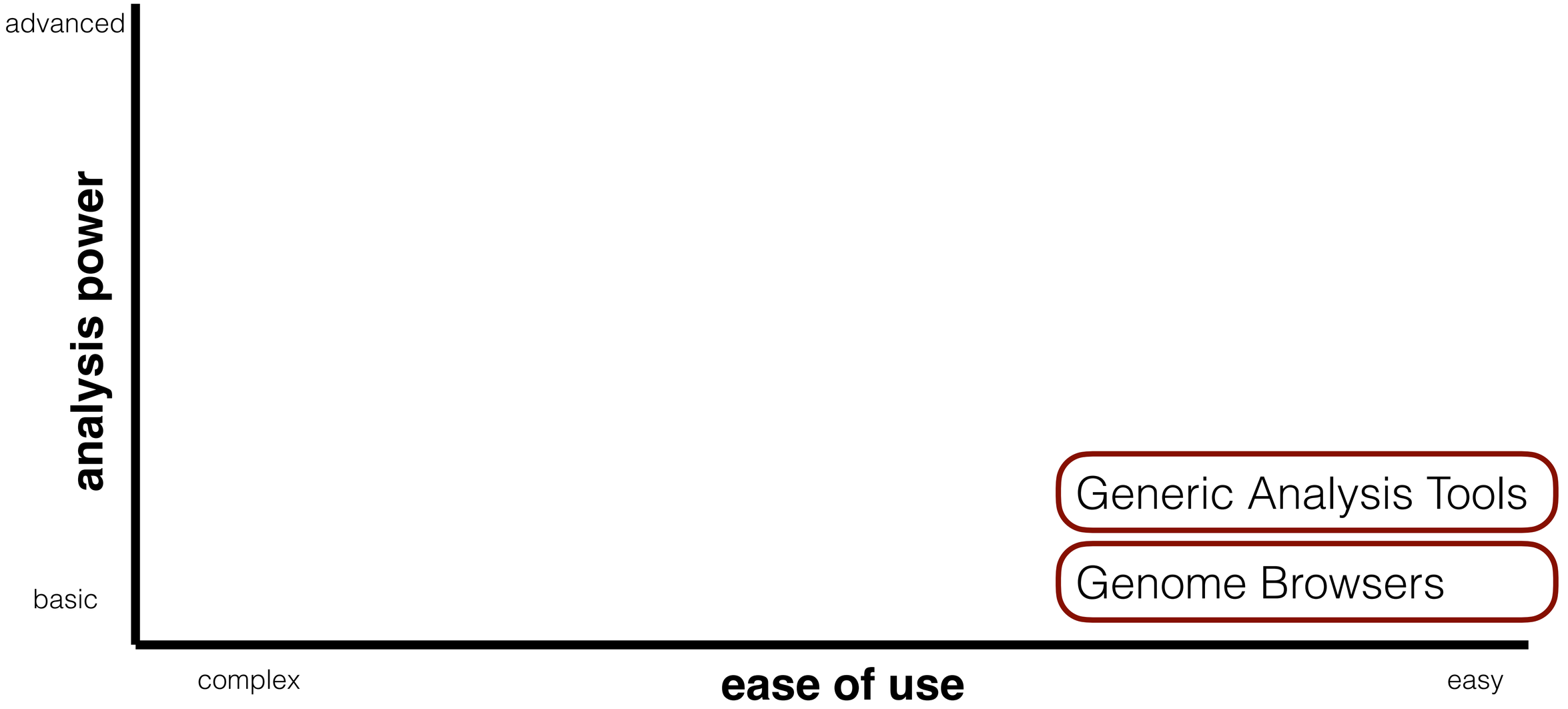
ease of use

easy

motivation


advanced
analysis power
basic
Generic Analysis Tools
complex
ease of use
easy

motivation

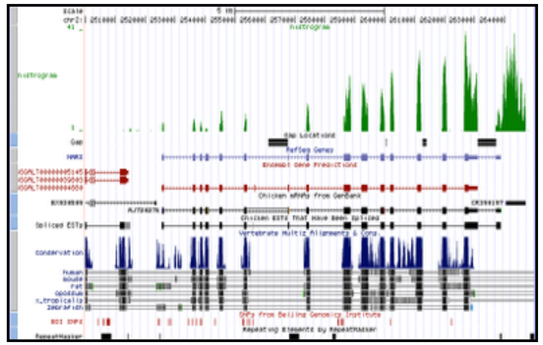

advanced
analysis power
basic
Generic Analysis Tools
Genome Browsers
complex
ease of use
easy

motivation

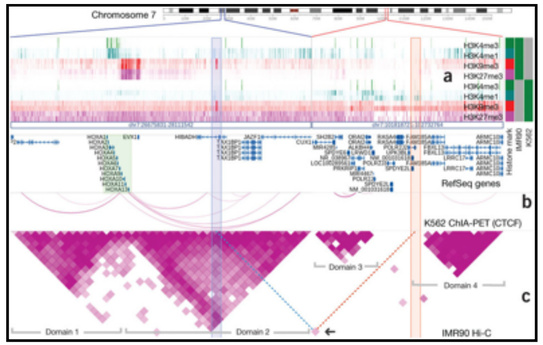
advanced

analysis power

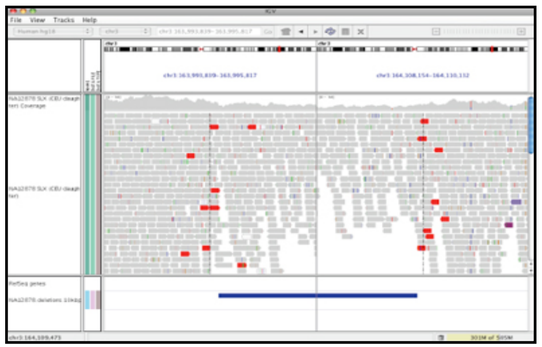
basic



UCSC Genome Browser



WashU Epigenome Browser



Integrated Genome Viewer

Generic Analysis Tools

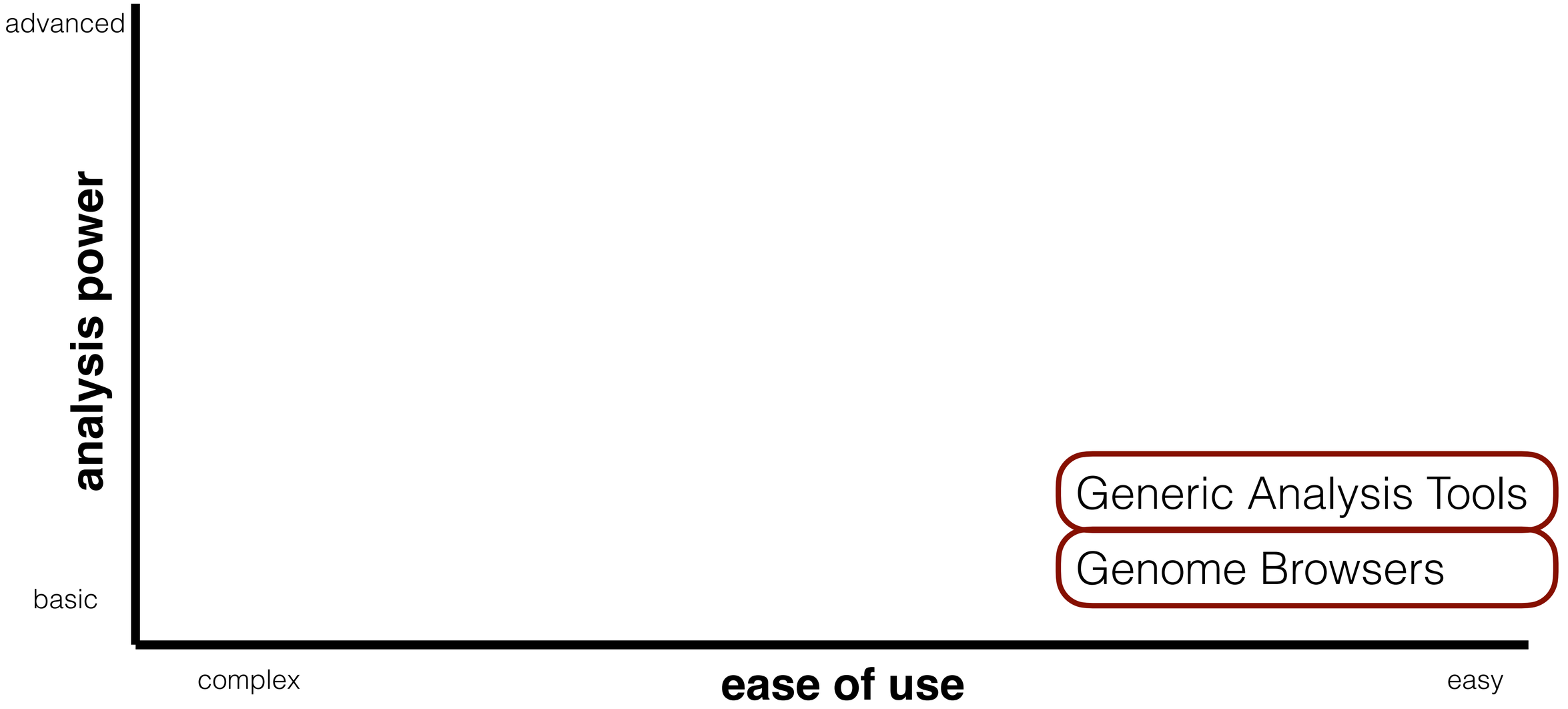
Genome Browsers

complex

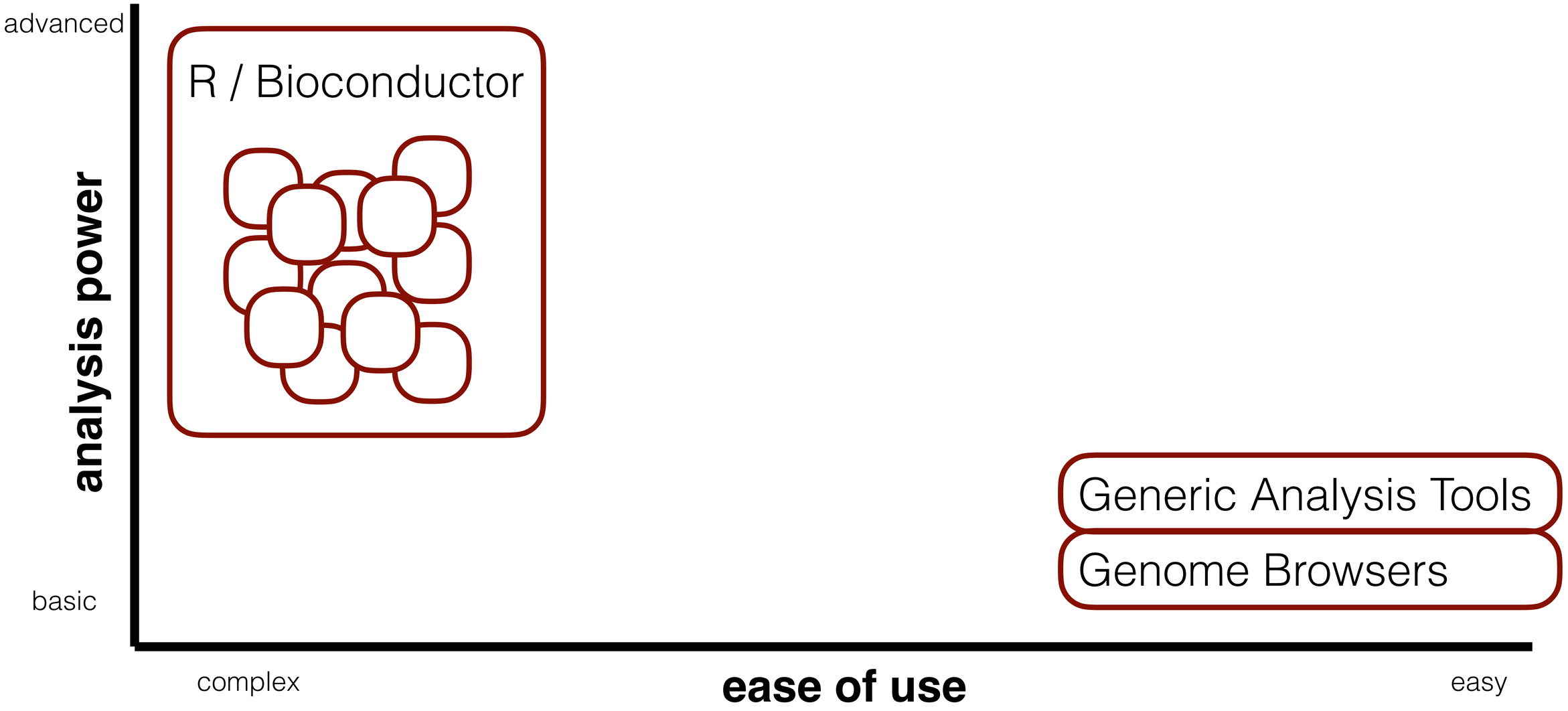
ease of use

easy

motivation


advanced
analysis power
basic
Generic Analysis Tools
Genome Browsers
complex
ease of use
easy

motivation


advanced
analysis power
basic
R / Bioconductor
Generic Analysis Tools
Genome Browsers
complex
ease of use
easy

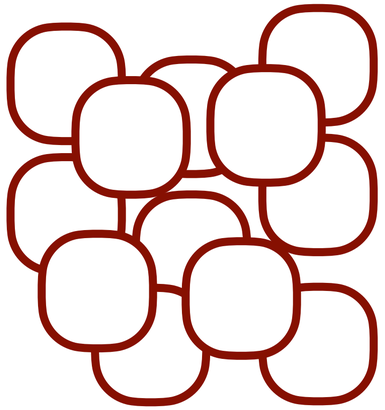
motivation

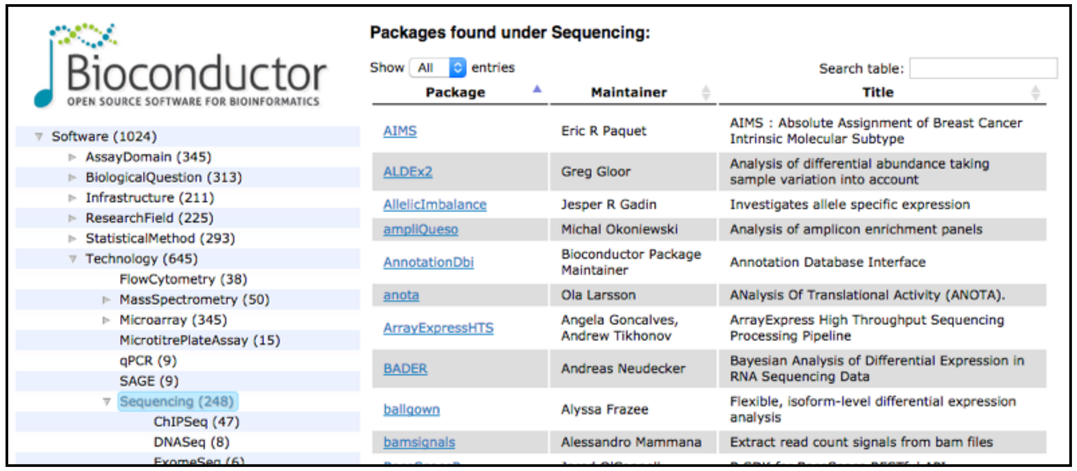
advanced

analysis power

basic

R / Bioconductor





Bioconductor Repository

Generic Analysis Tools

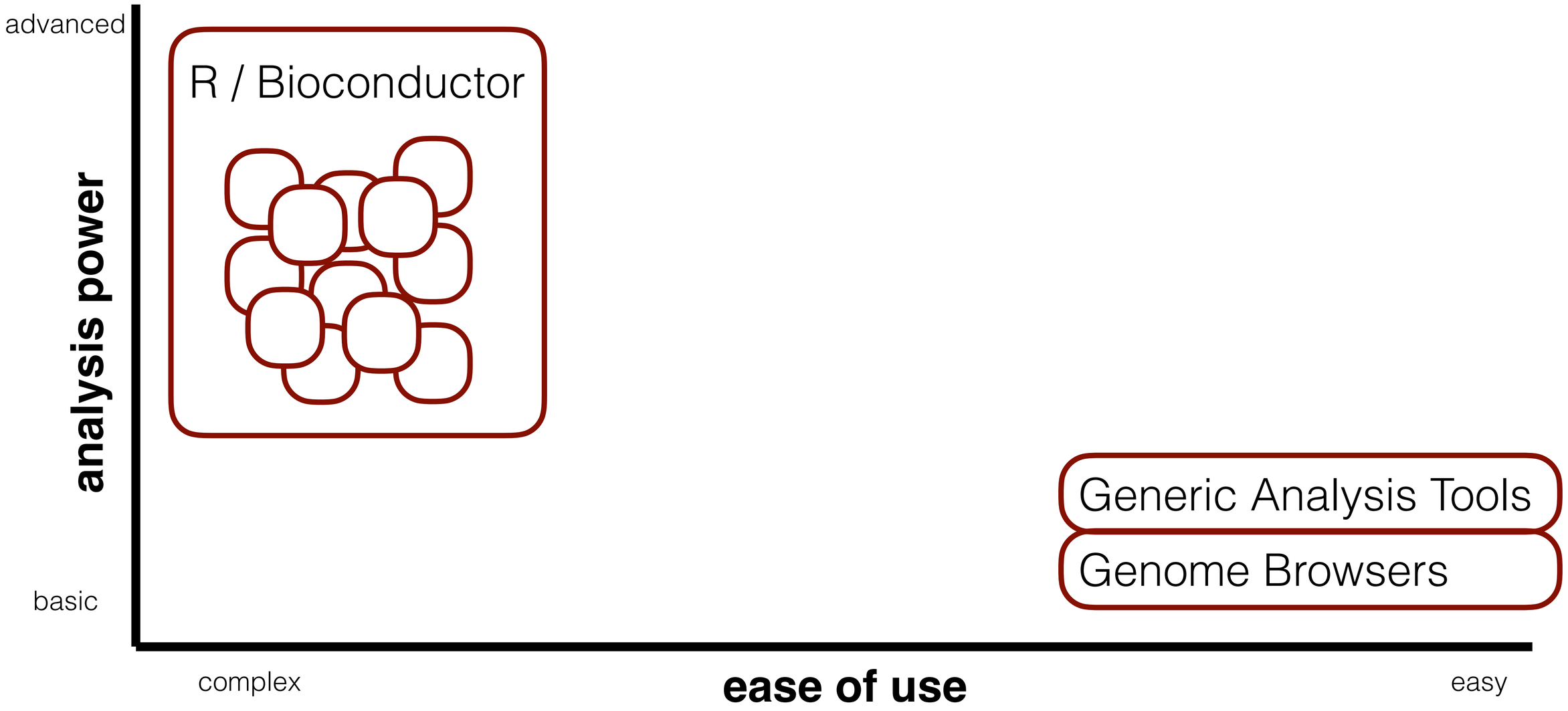
Genome Browsers

complex

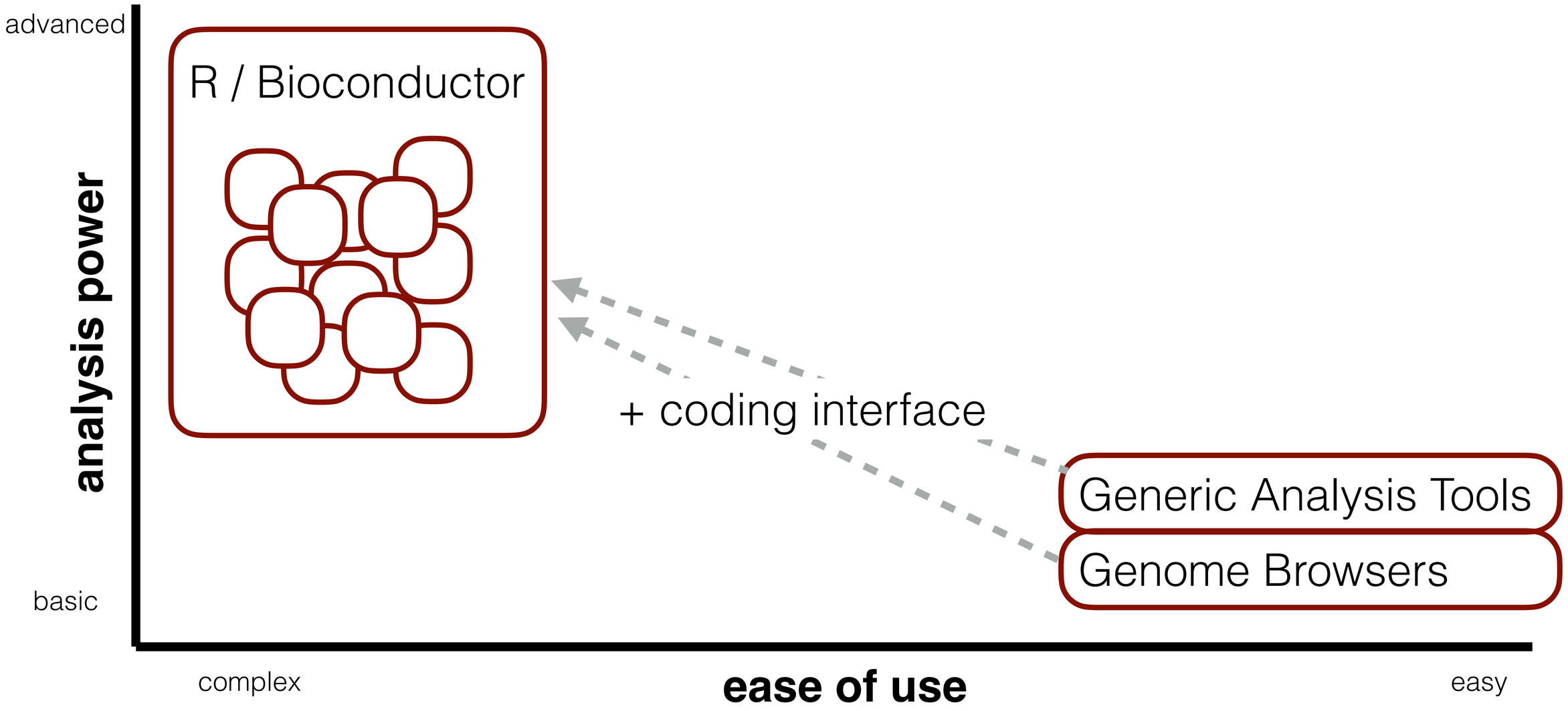
ease of use

easy

motivation


advanced
analysis power
basic
R / Bioconductor
Generic Analysis Tools
Genome Browsers
complex
ease of use
easy

motivation


advanced
analysis power
basic
R / Bioconductor
+ coding interface
Generic Analysis Tools
Genome Browsers
complex
ease of use
easy

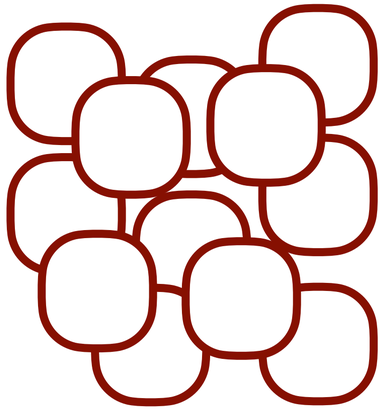
motivation

advanced

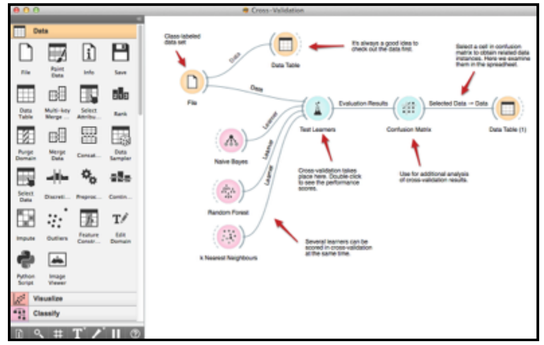
analysis power

basic

R / Bioconductor



KNIME



Orange

+ coding interface

Generic Analysis Tools

Genome Browsers

complex

ease of use

easy

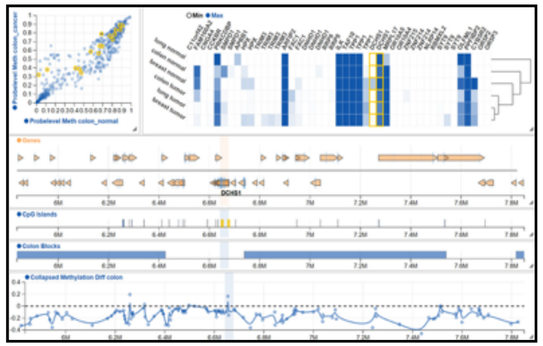
motivation

advanced

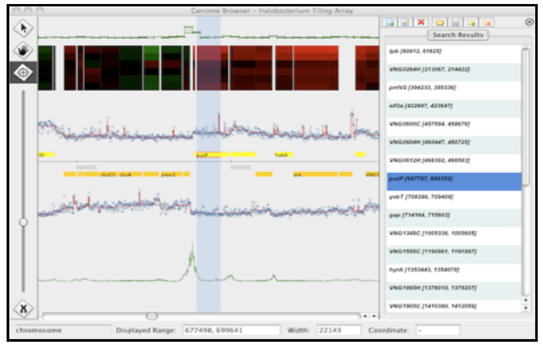
analysis power

basic

R / Bioconductor

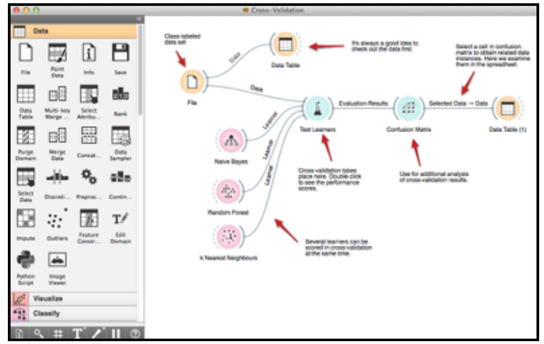


Epiviz



Gaggle Genome Browser

KNIME



Orange

+ coding interface

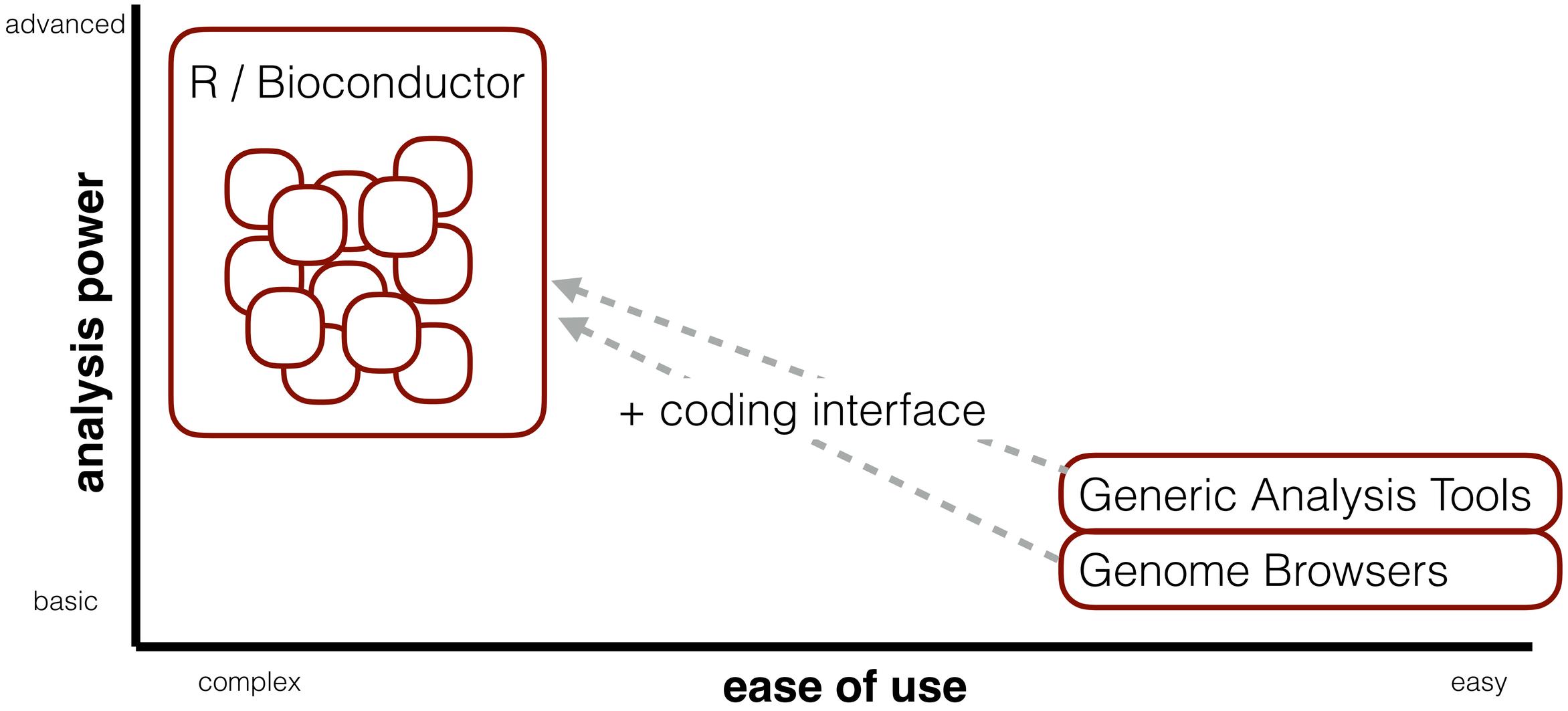
Generic Analysis Tools

Genome Browsers

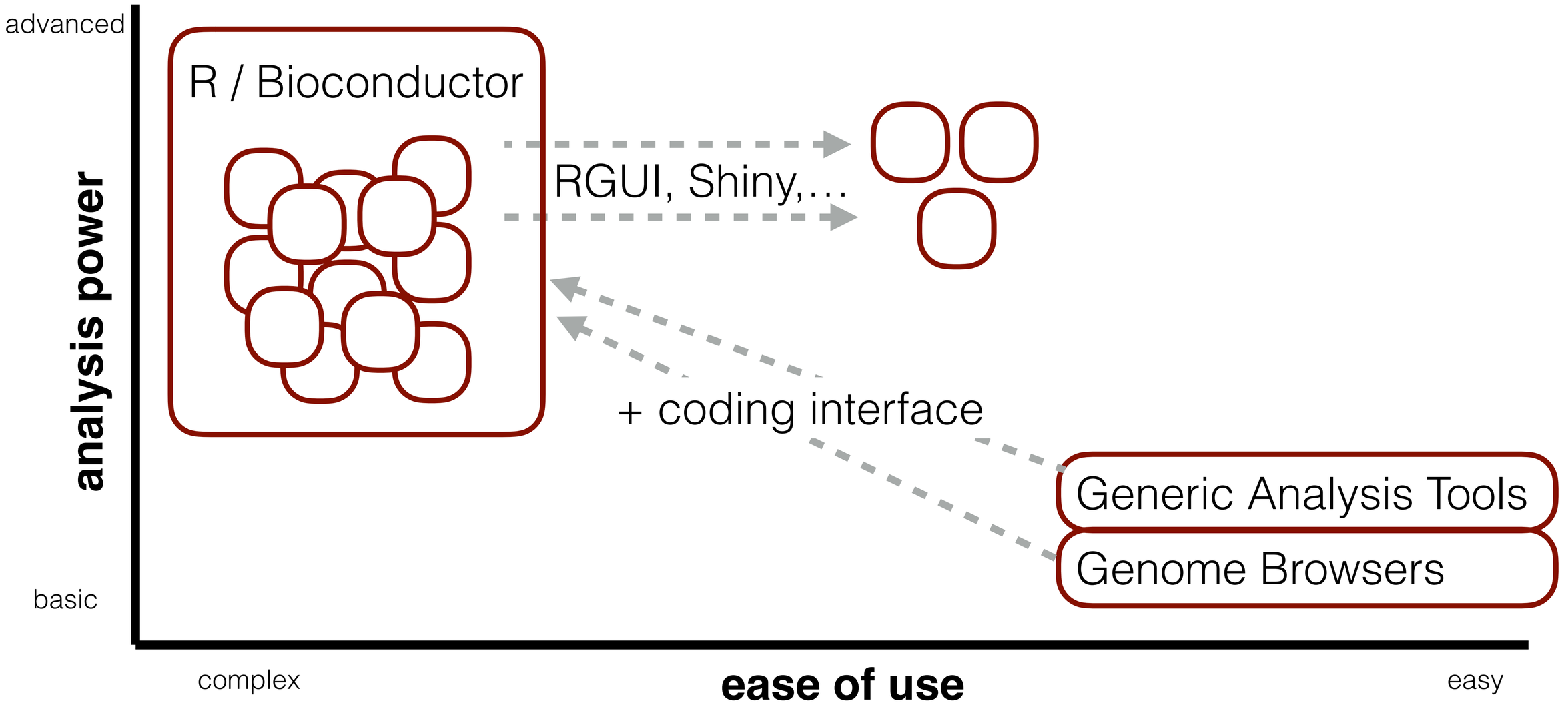
ease of use

easy

motivation


advanced
analysis power
basic
R / Bioconductor
+ coding interface
Generic Analysis Tools
Genome Browsers
complex
ease of use
easy

motivation


advanced
analysis power
basic
R / Bioconductor
RGUI, Shiny,…
+ coding interface
Generic Analysis Tools
Genome Browsers
complex
ease of use
easy

motivation

advanced

basic

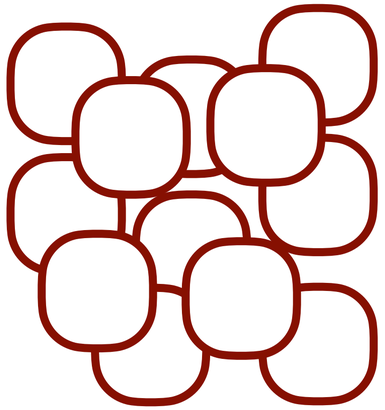
analysis power

complex

ease of use

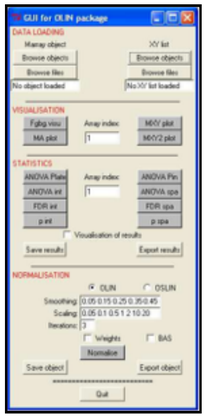
easy

R / Bioconductor

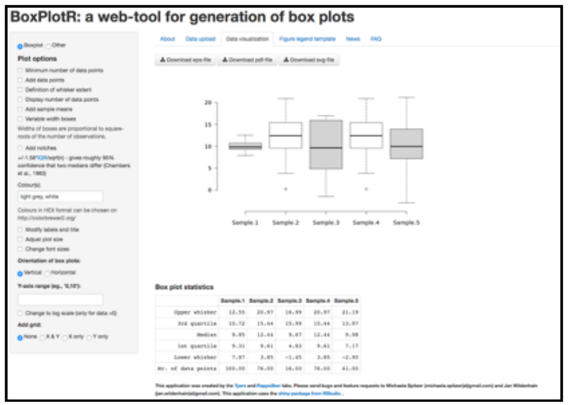


RGUI, Shiny,…

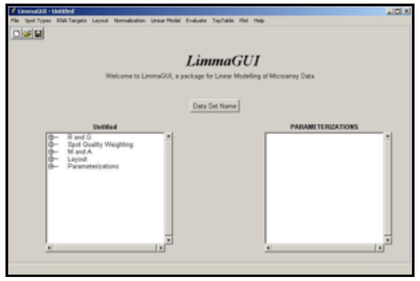
+ coding interface



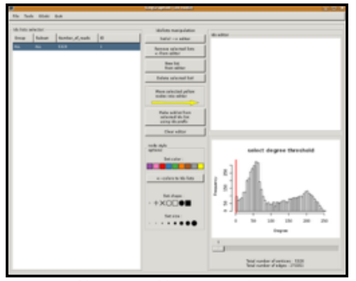
OLINgui



BoxPlotR



LimmaGUI

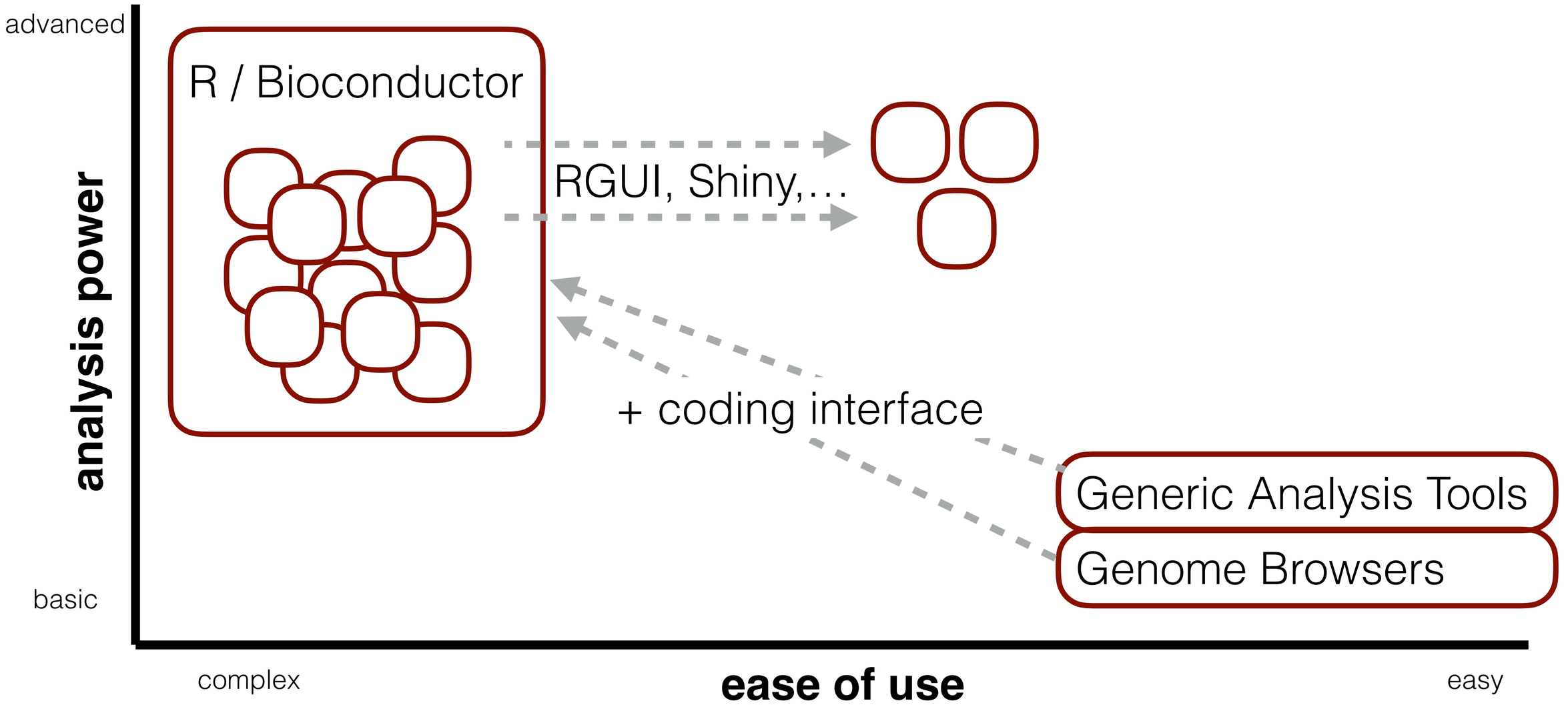


SeqGraphR

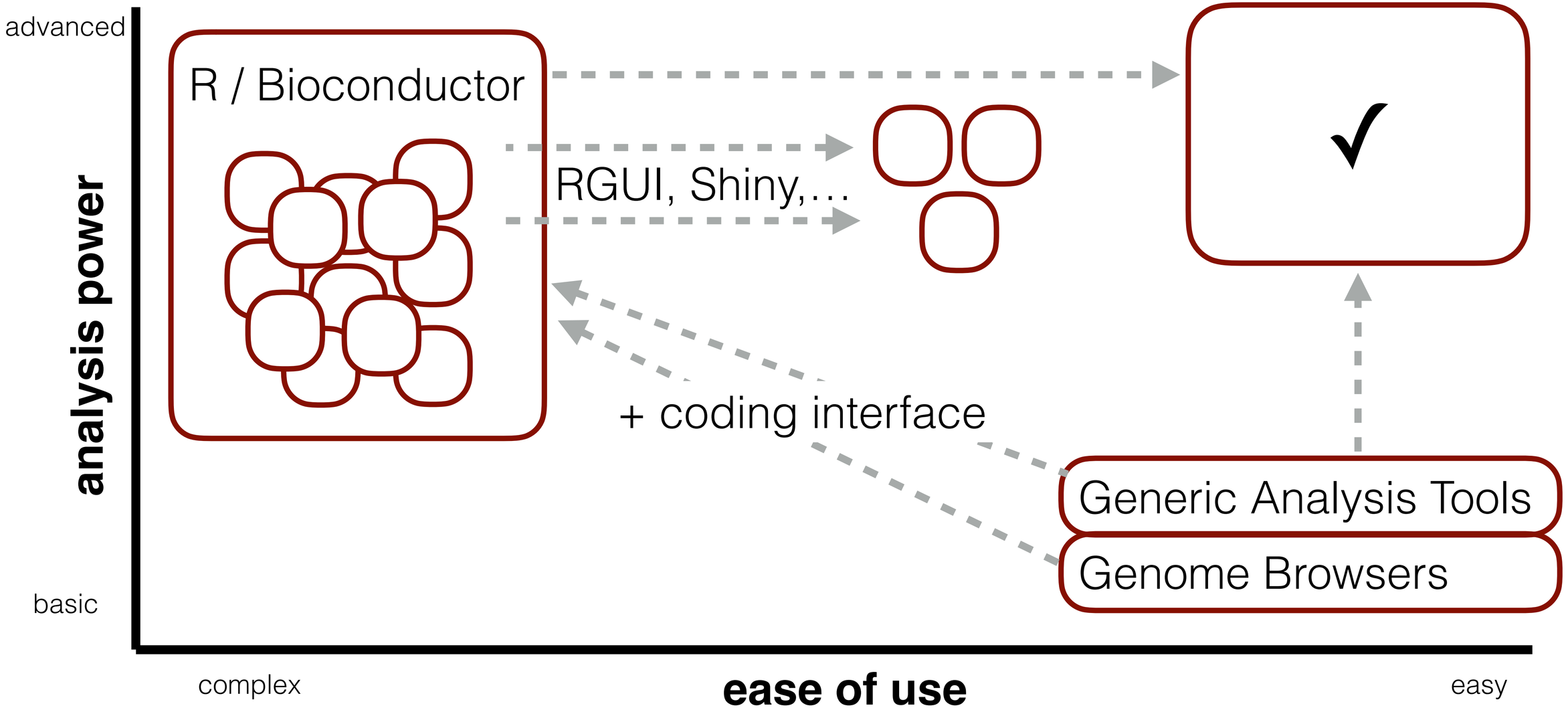
Generic Analysis Tools

Genome Browsers

motivation

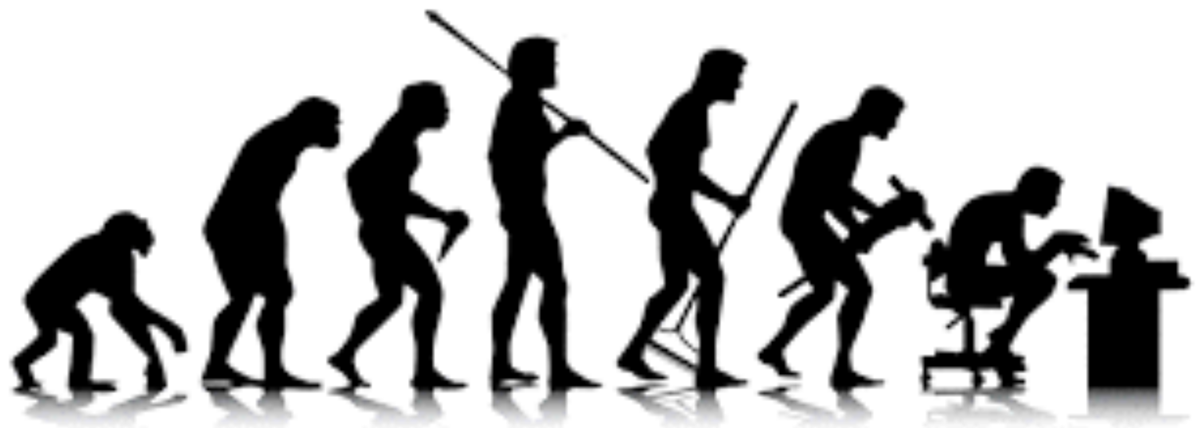

advanced
analysis power
basic
R / Bioconductor
RGUI, Shiny,…
+ coding interface
Generic Analysis Tools
Genome Browsers
complex
ease of use
easy

motivation

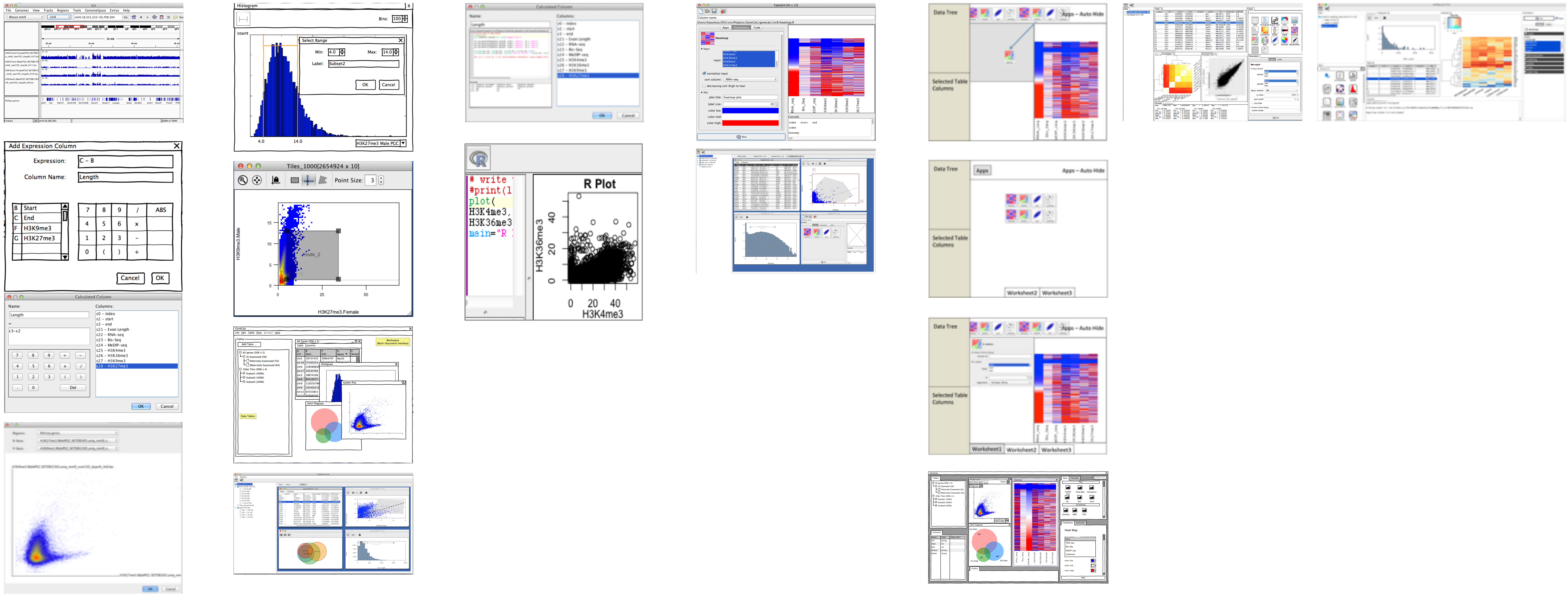

advanced
analysis power
basic
R / Bioconductor
RGUI, Shiny,…
✓
+ coding interface
Generic Analysis Tools
Genome Browsers
complex
ease of use
easy

VisRseq

design evolution

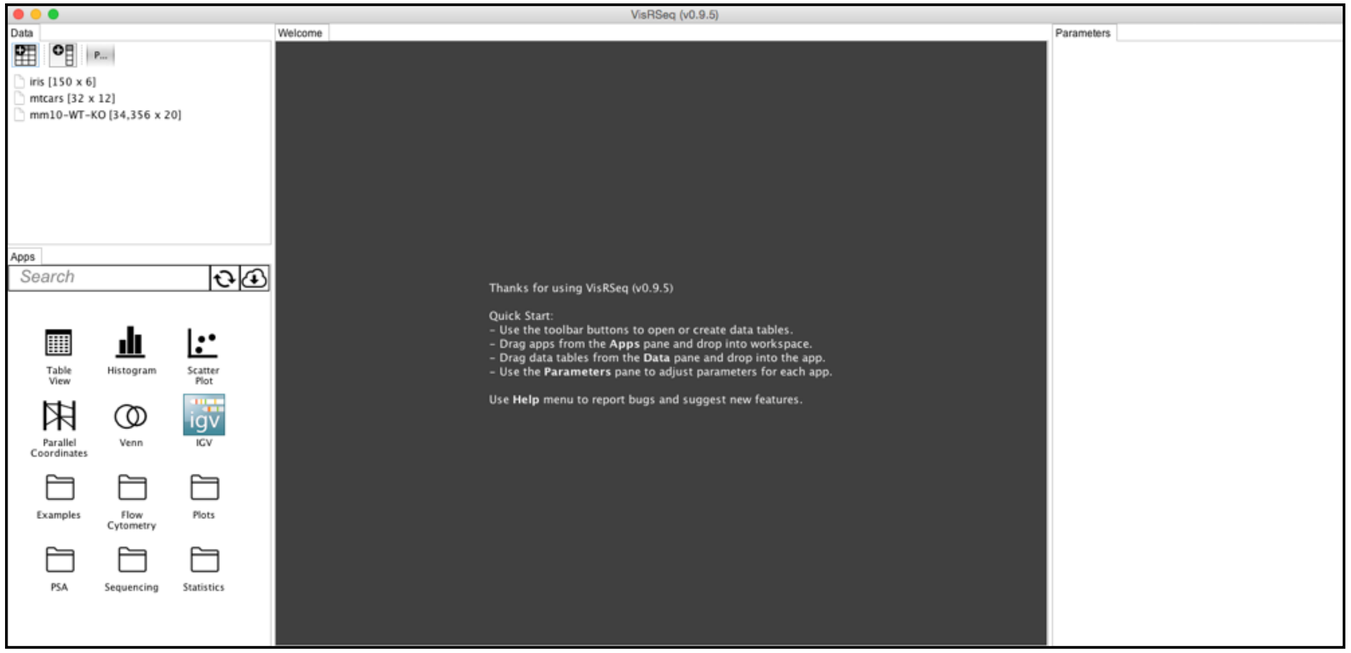


IGV plugin (2013) — workspace — R integration — BioVis Poster (2014) — Redesigned Layout — BMC Bioinf (2015) — **open beta**

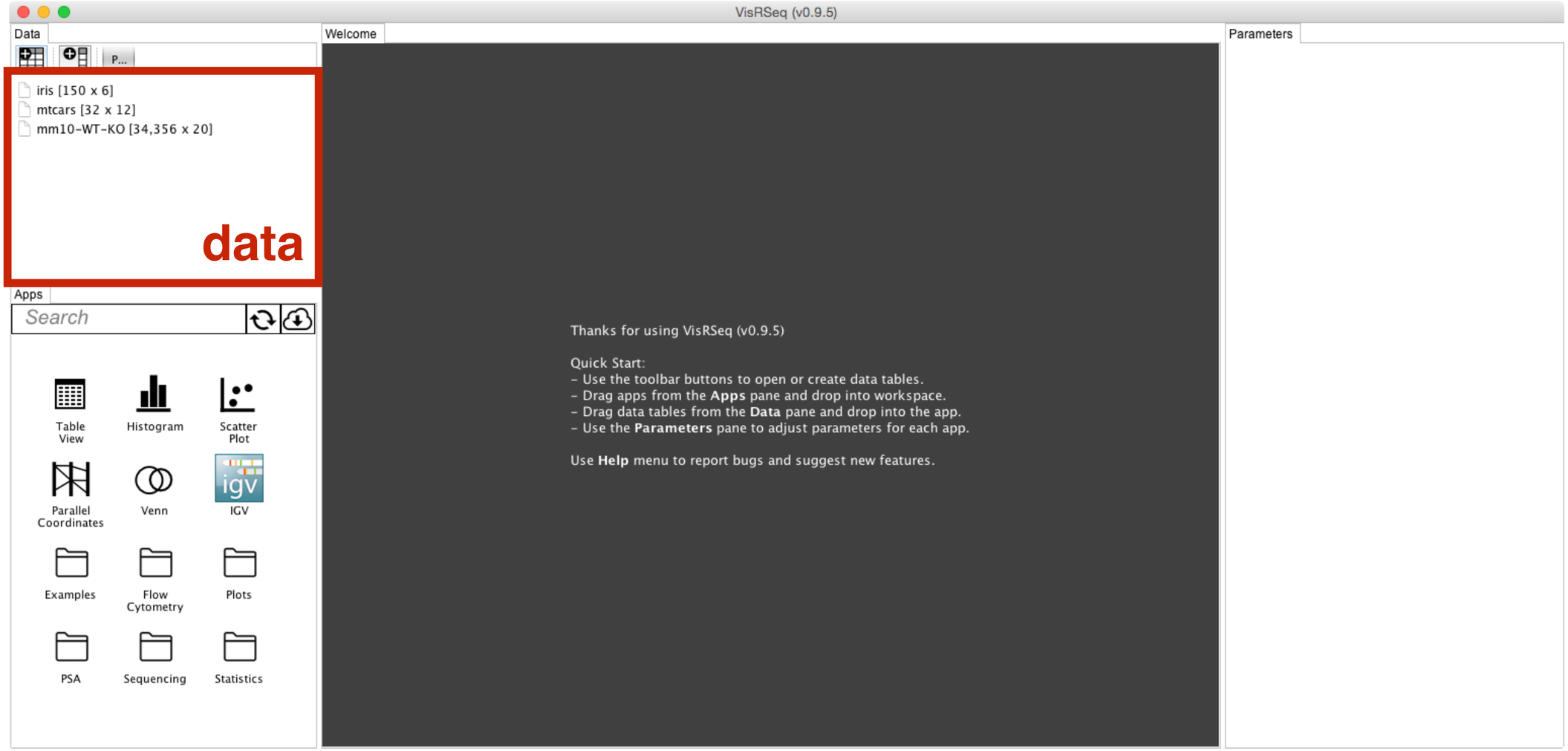


interface

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

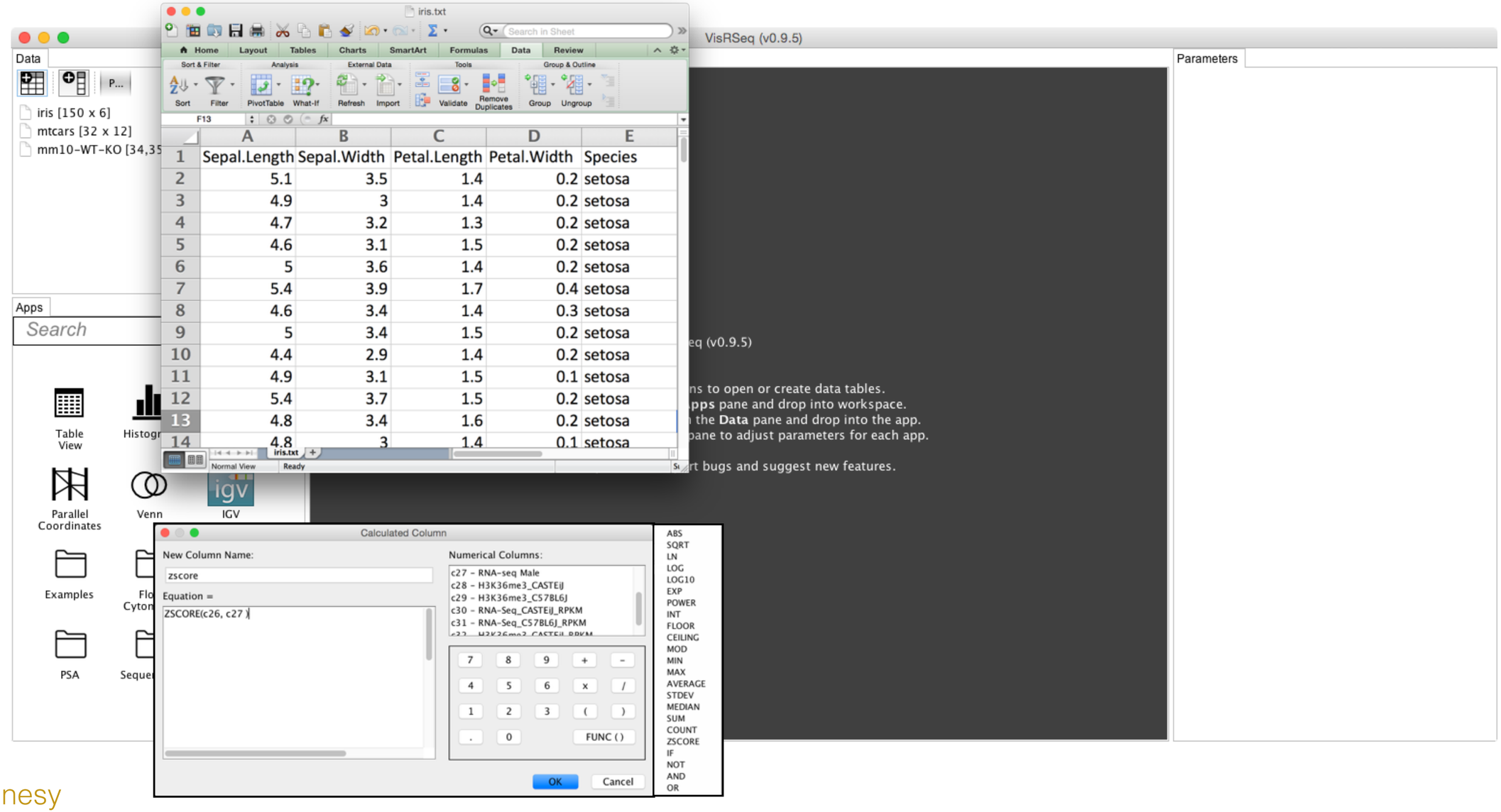


interface: data



interface: data

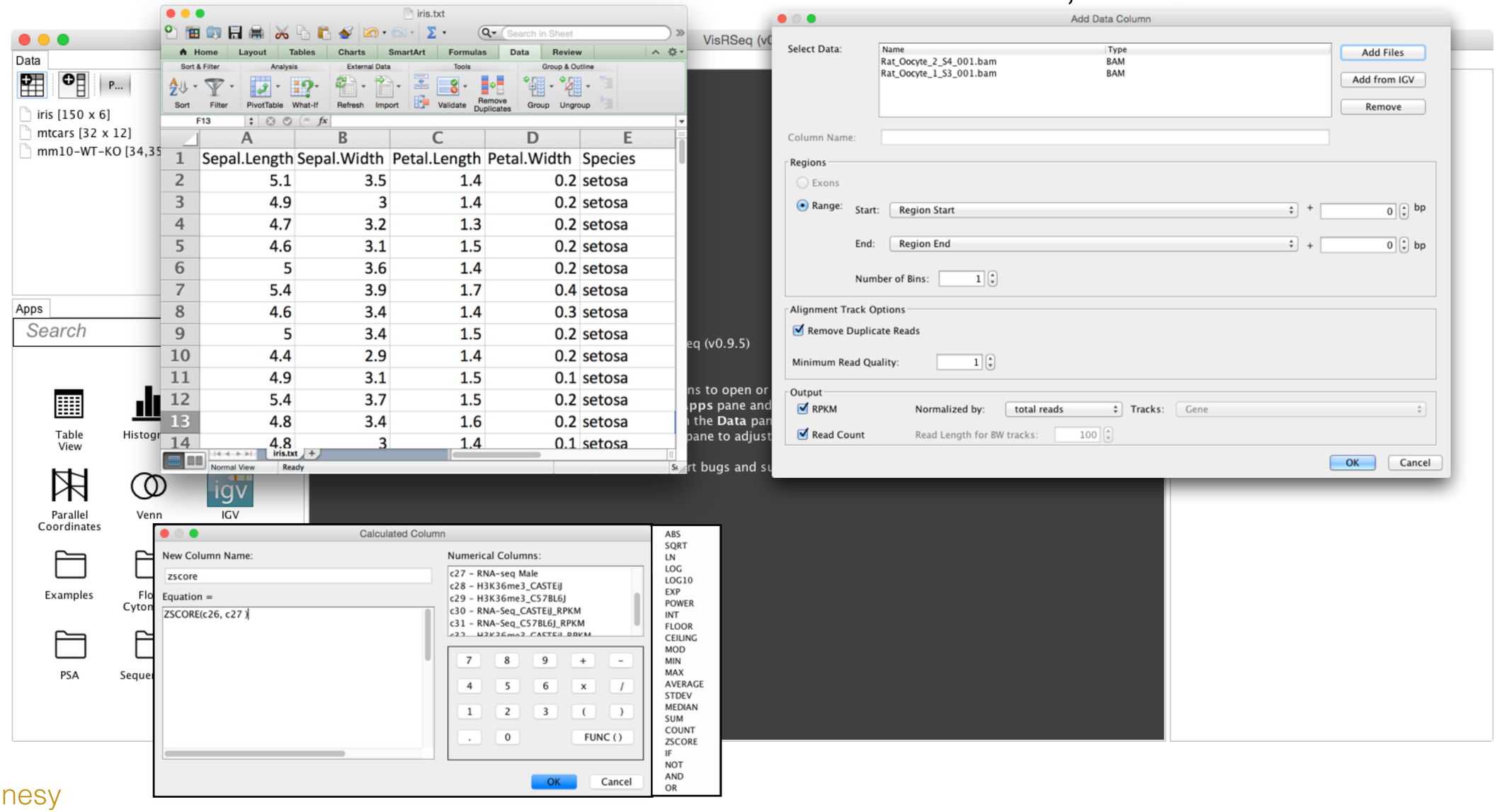
Tab delimited Text



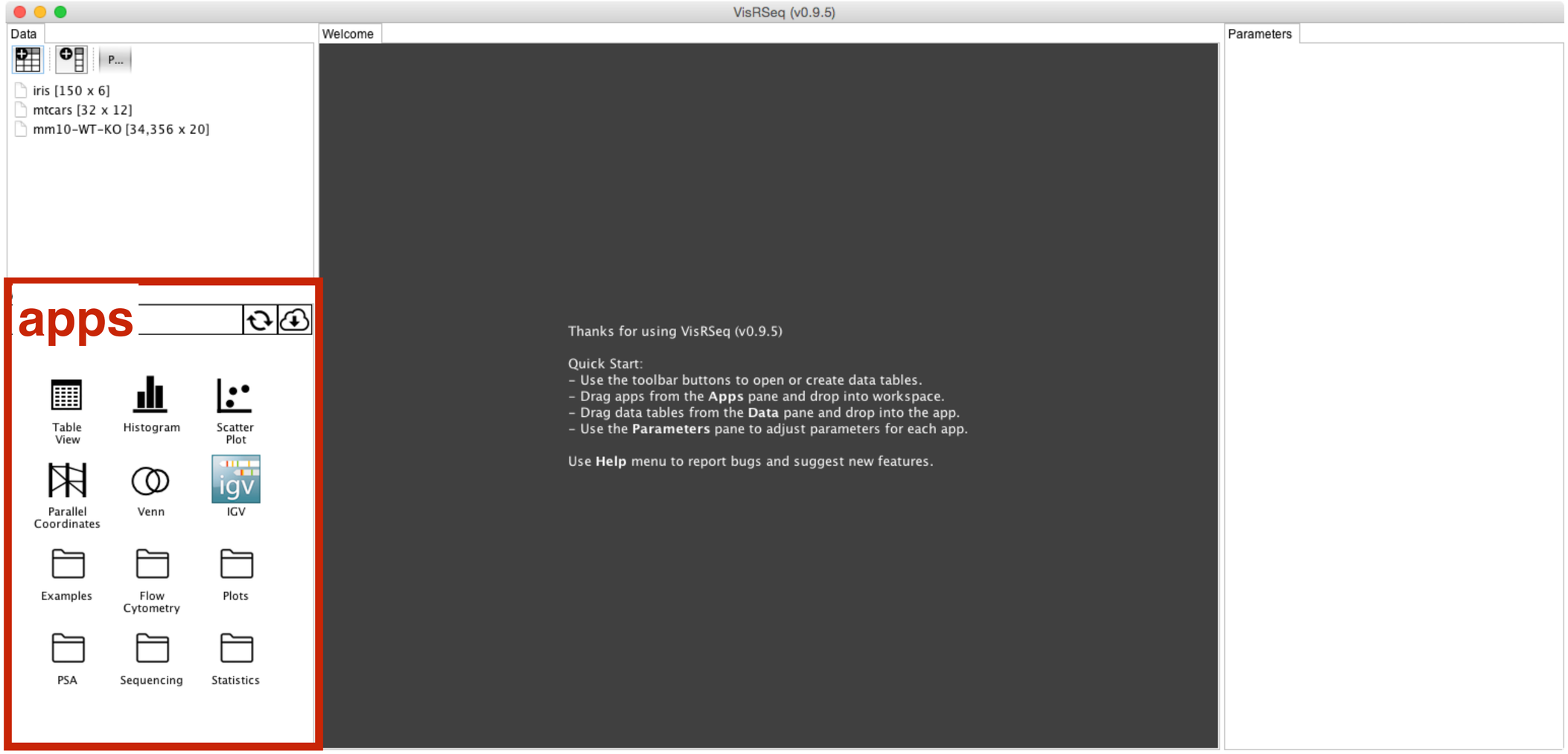
interface: data

Tab delimited Text

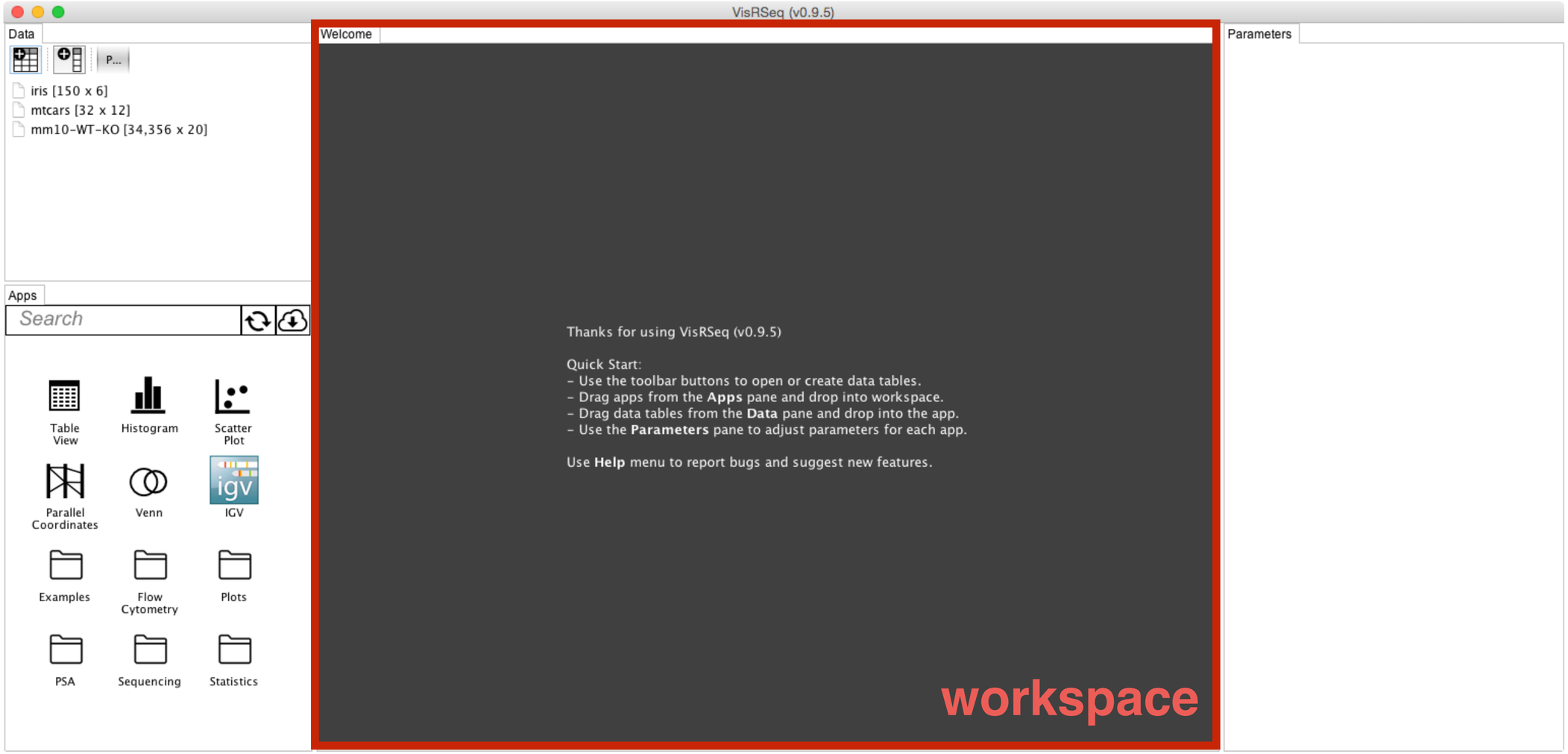
.BAM, .WIG



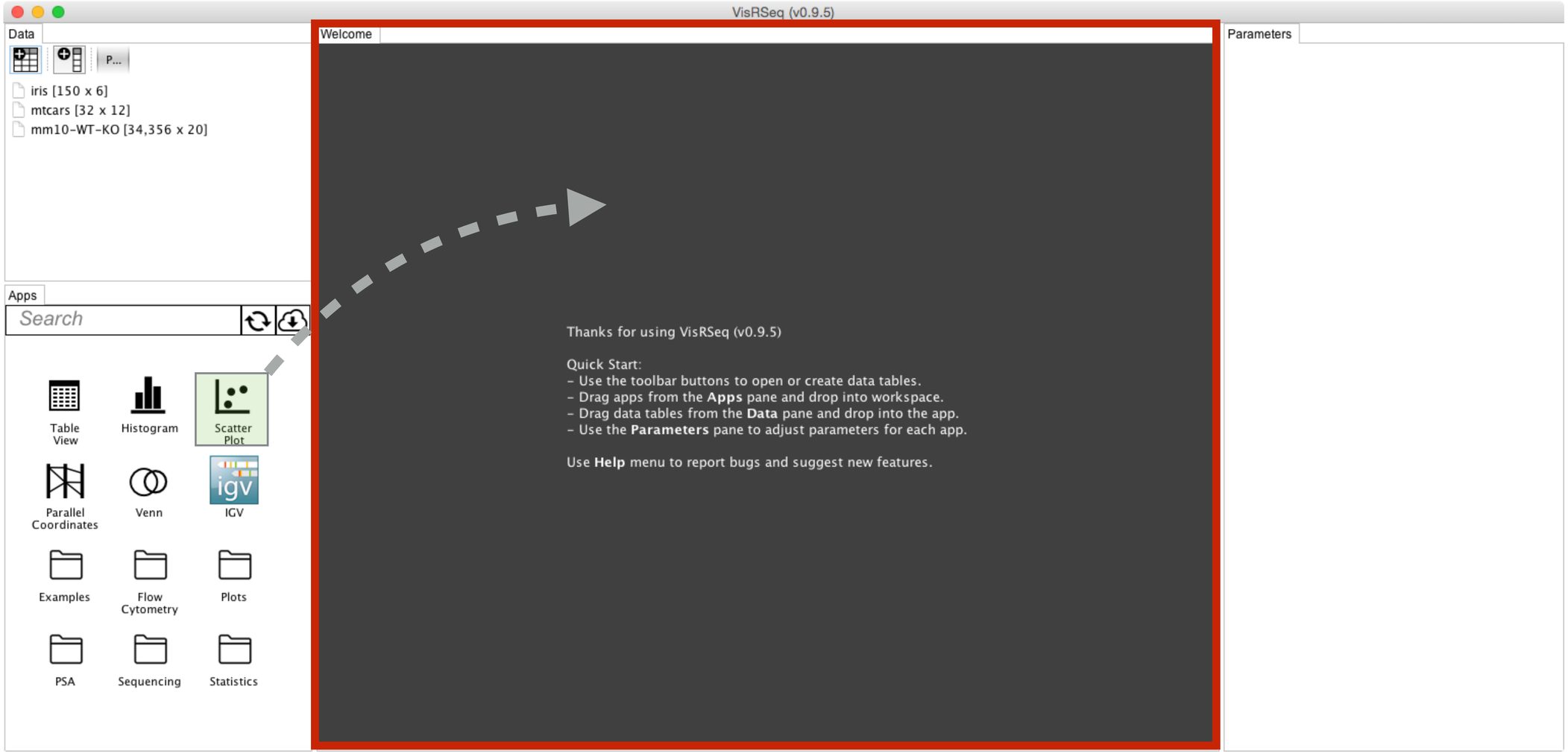
interface: apps



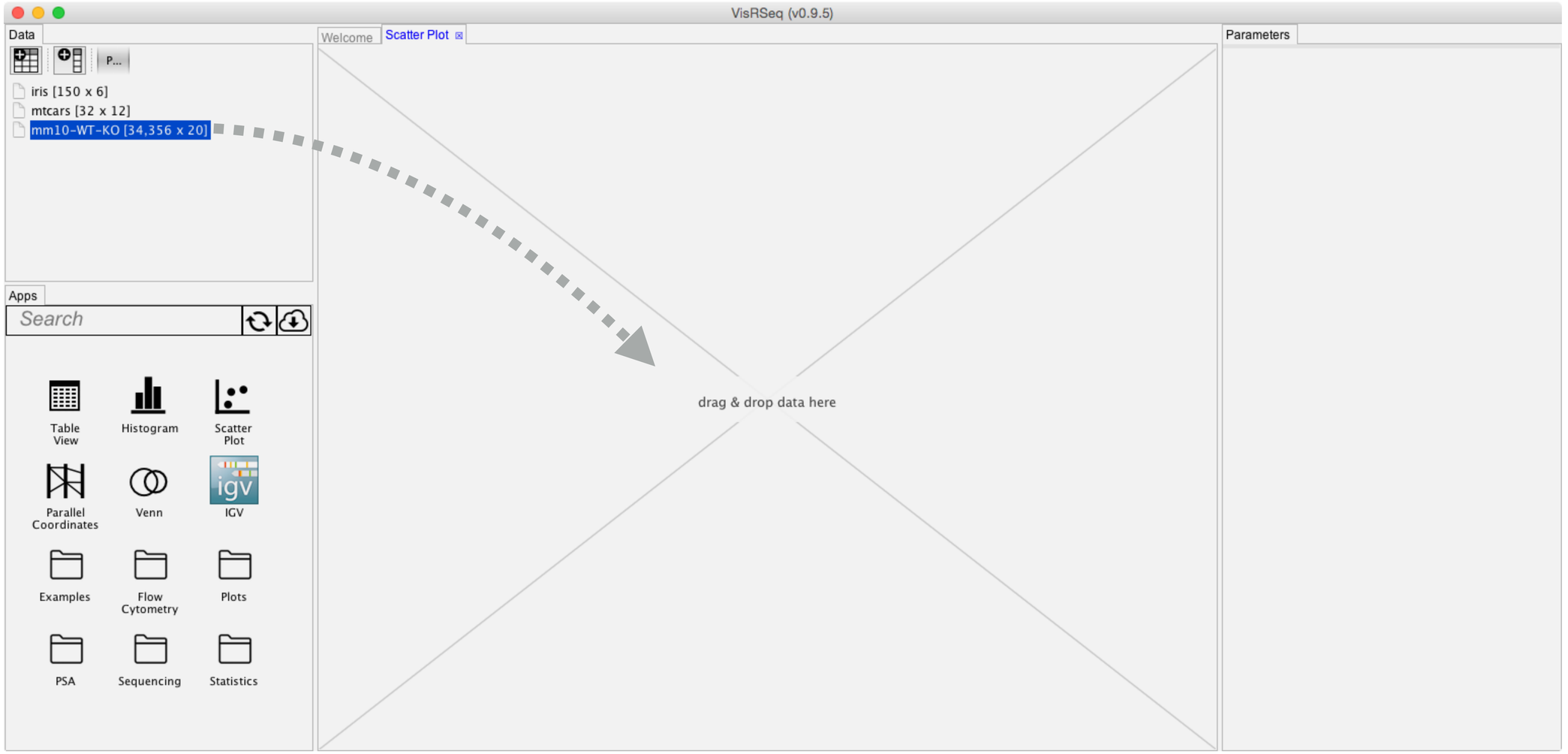
interface: workspace



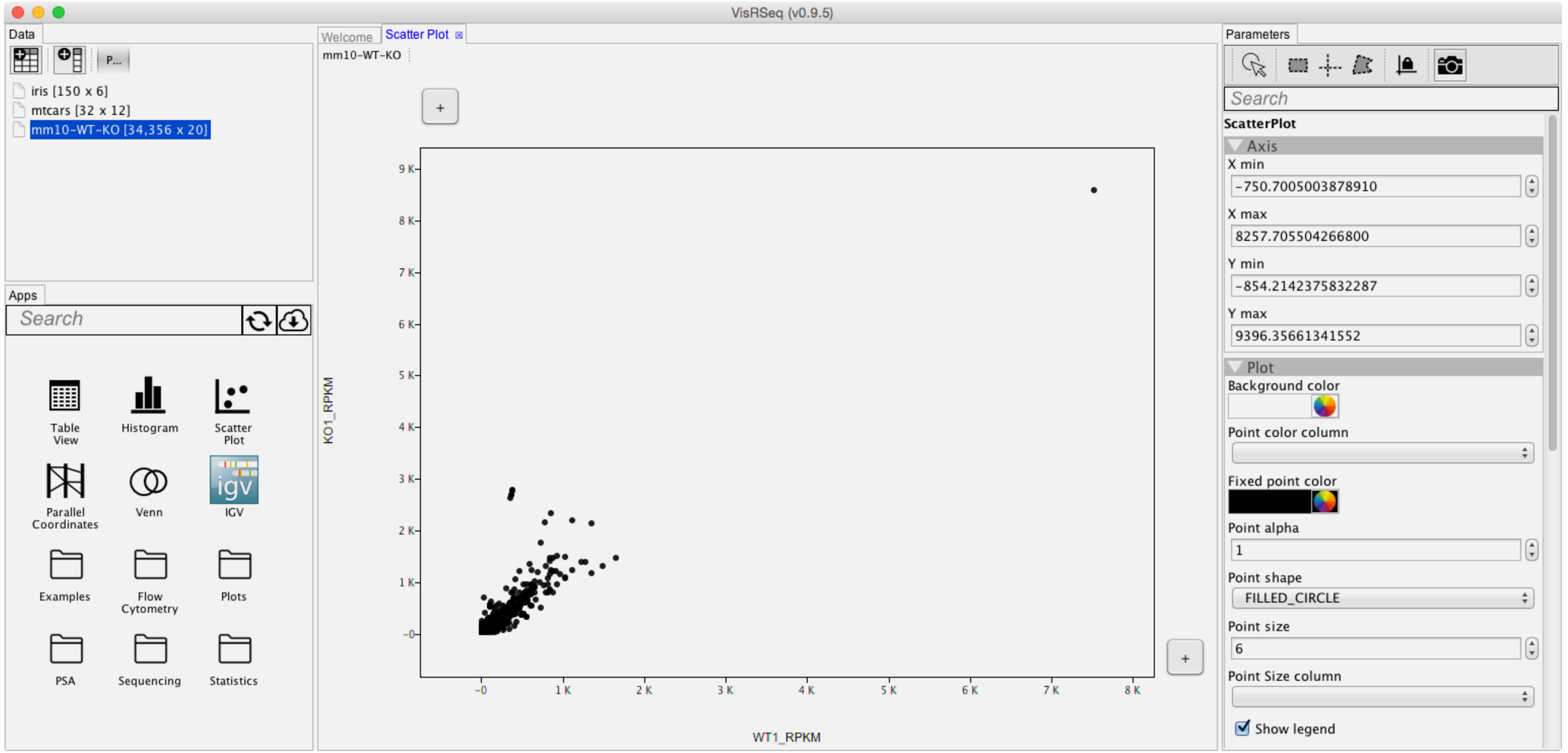
interface: workspace



interface: workspace

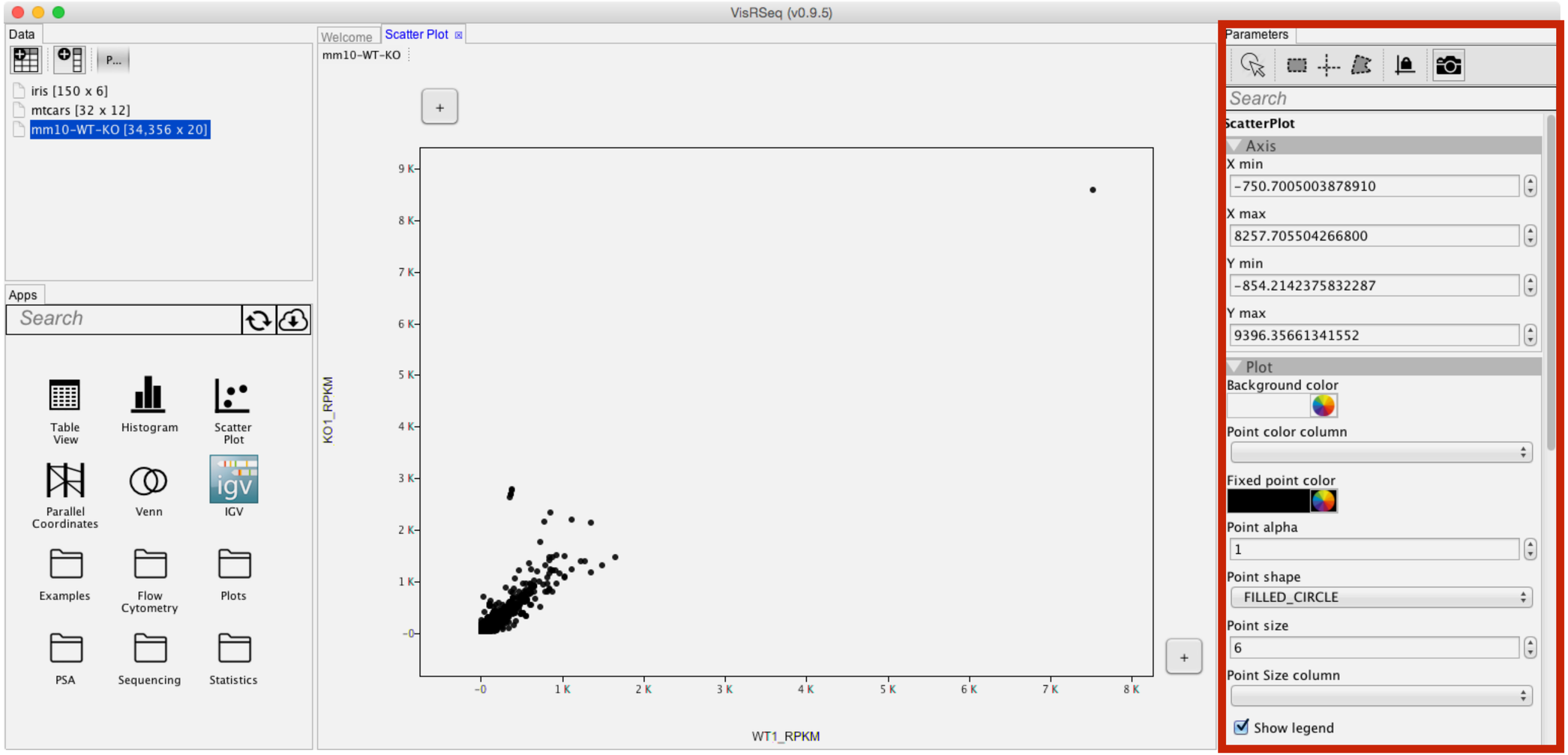


interface: workspace

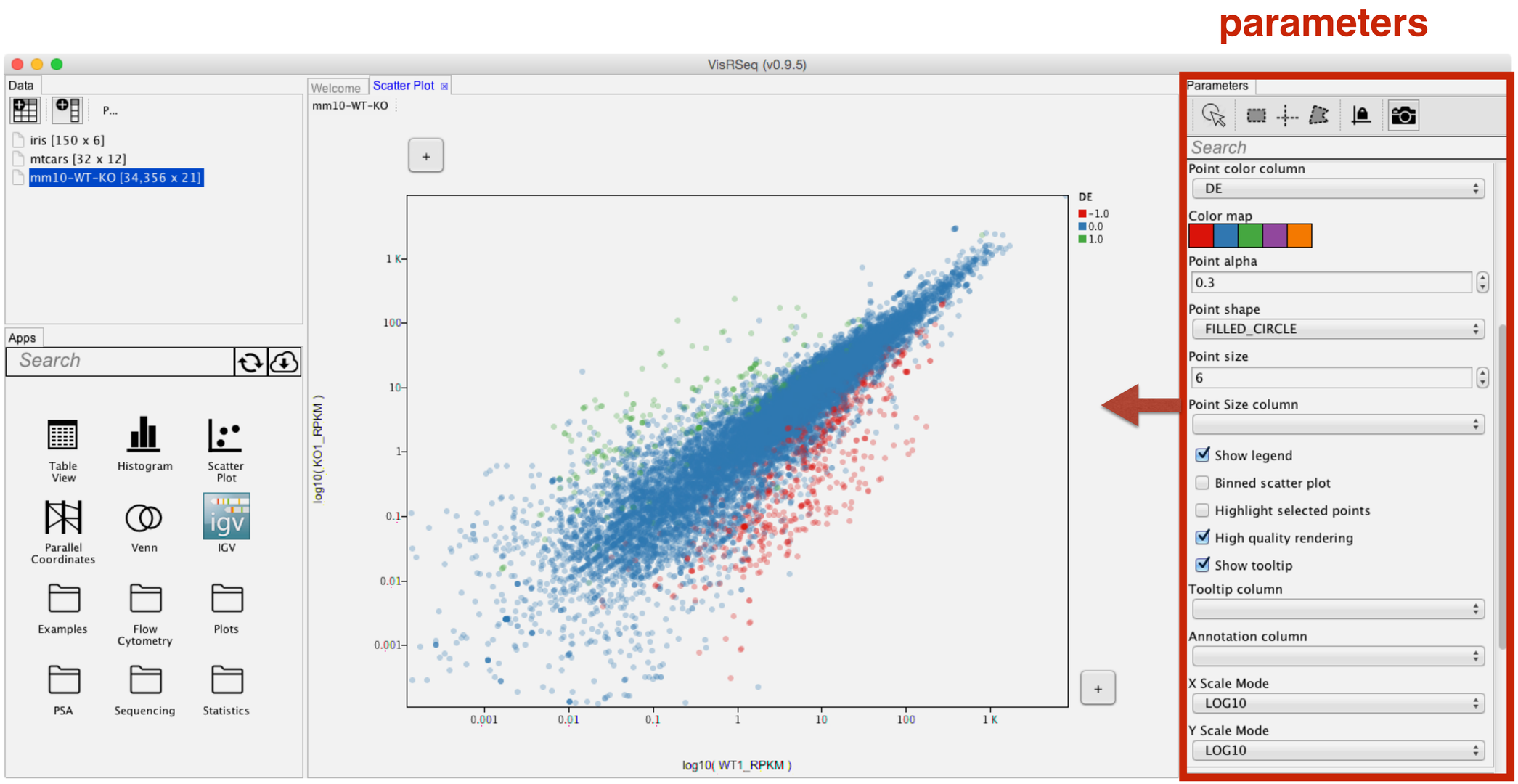


interface: parameters

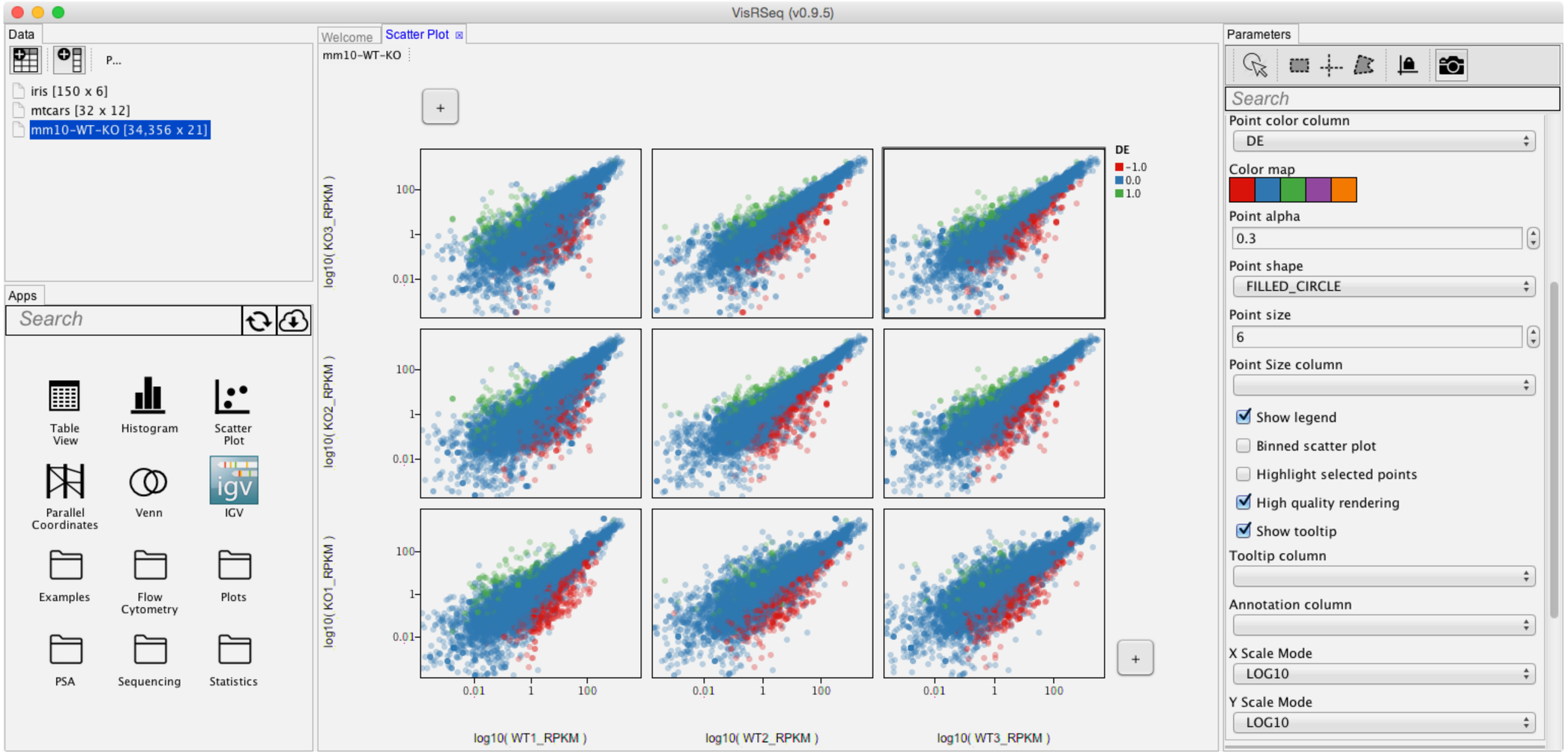
parameters



interface: parameters

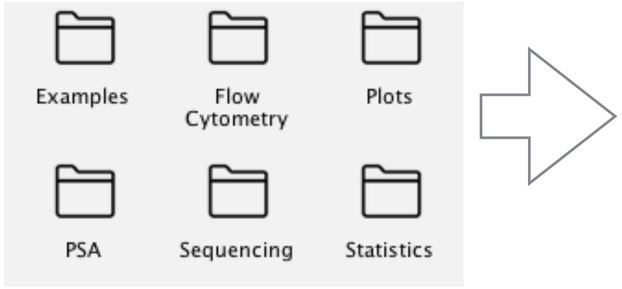


Apps: interactive

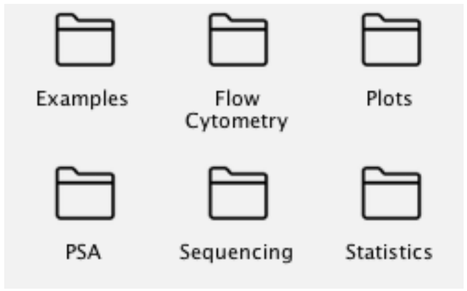


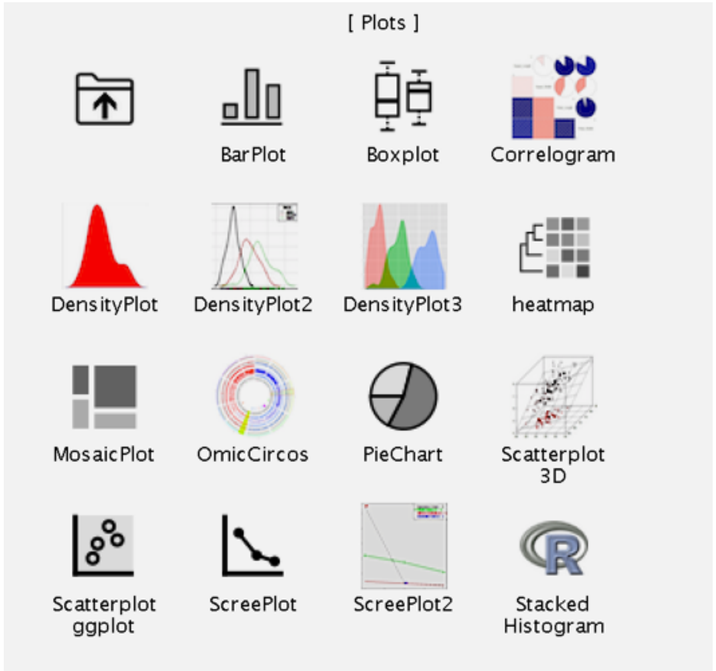
R - apps

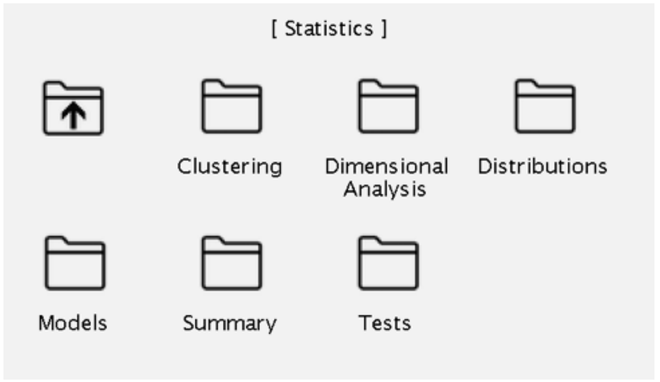
R Apps

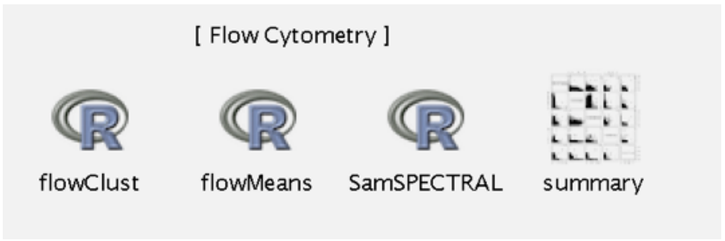


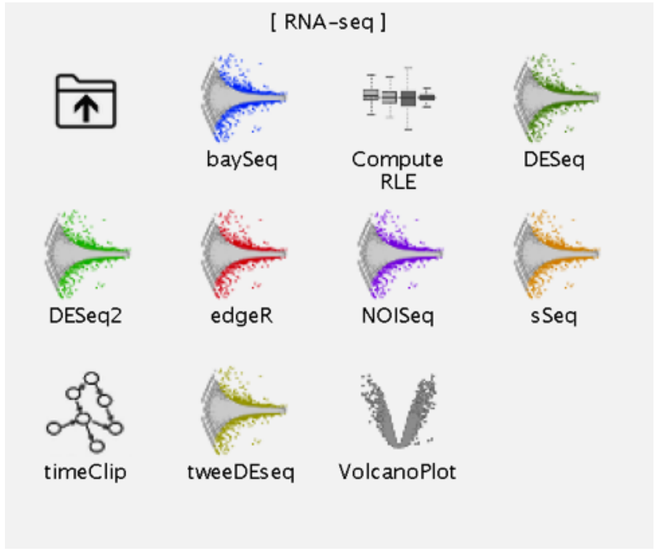
R Apps


Examples
Flow Cytometry
Plots
PSA
Sequencing
Statistics

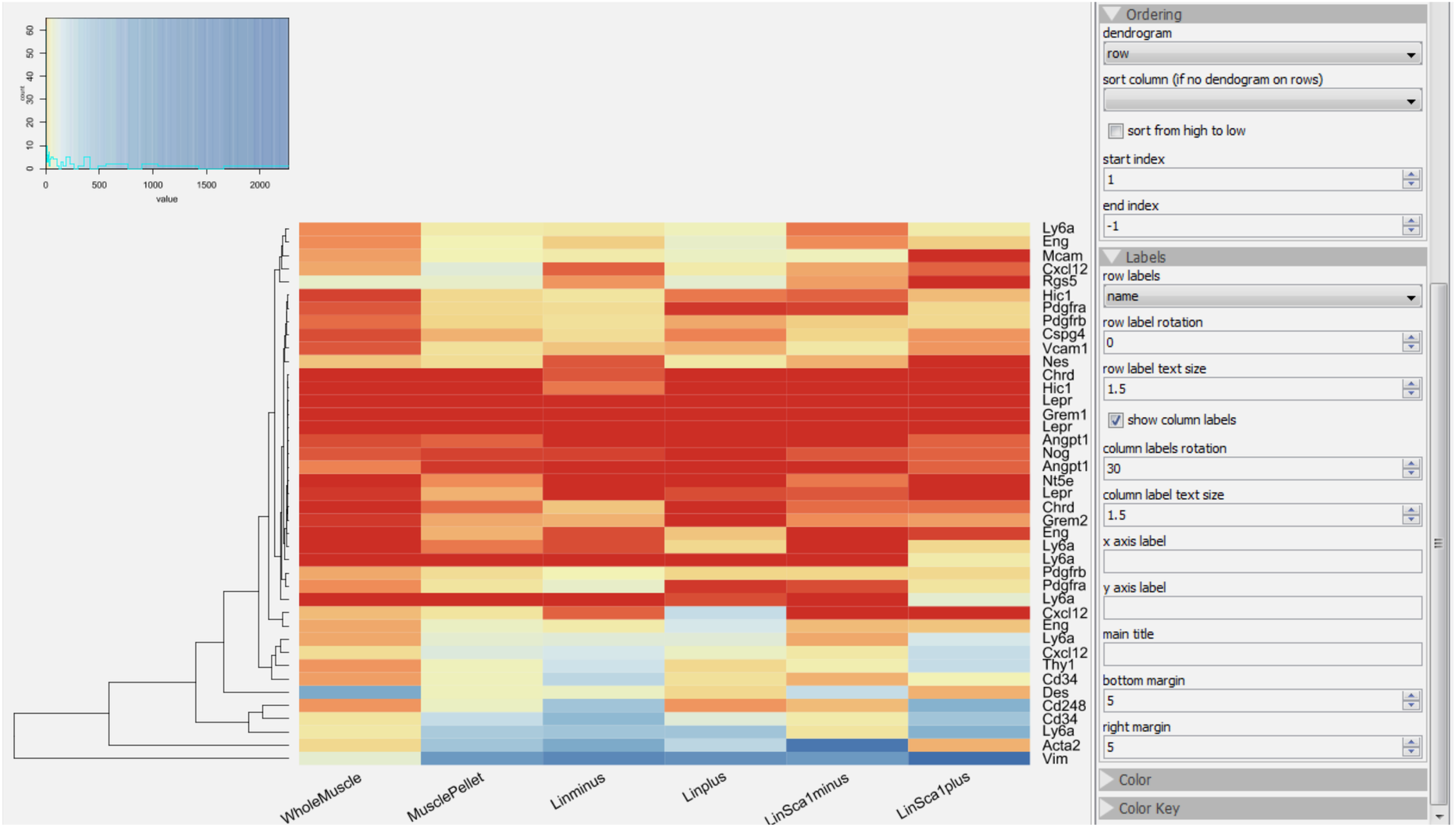

[Plots]
BarPlot
Boxplot
Correlogram
DensityPlot
DensityPlot2
DensityPlot3
heatmap
MosaicPlot
OmicCircos
PieChart
Scatterplot 3D
Scatterplot ggplot
ScreePlot
ScreePlot2
Stacked Histogram


[Statistics]
Clustering
Dimensional Analysis
Distributions
Models
Summary
Tests


[Flow Cytometry]
flowClust
flowMeans
SamSPECTRAL
summary


[RNA-seq]
baySeq
Compute RLE
DESeq
DESeq2
edgeR
NOISeq
sSeq
timeClip
tweeDEseq
VolcanoPlot

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

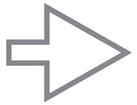
kmeans.R

```
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))
```

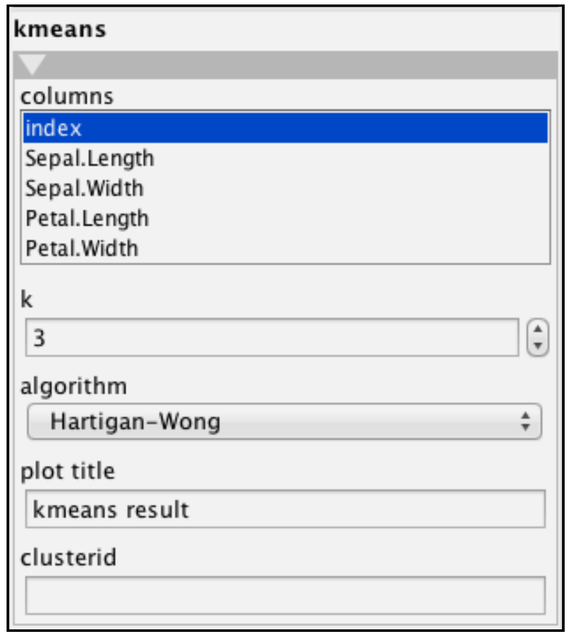
parameters

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

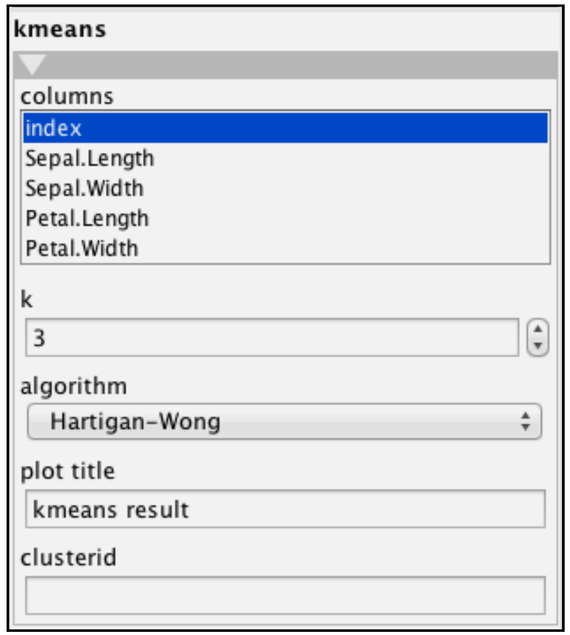
```
source("visrutils.R")

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

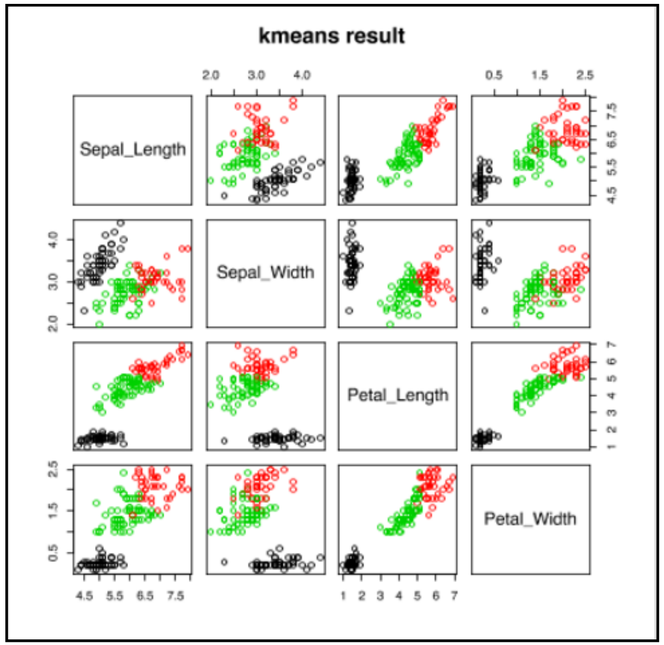
kmeans.json

```
{  "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

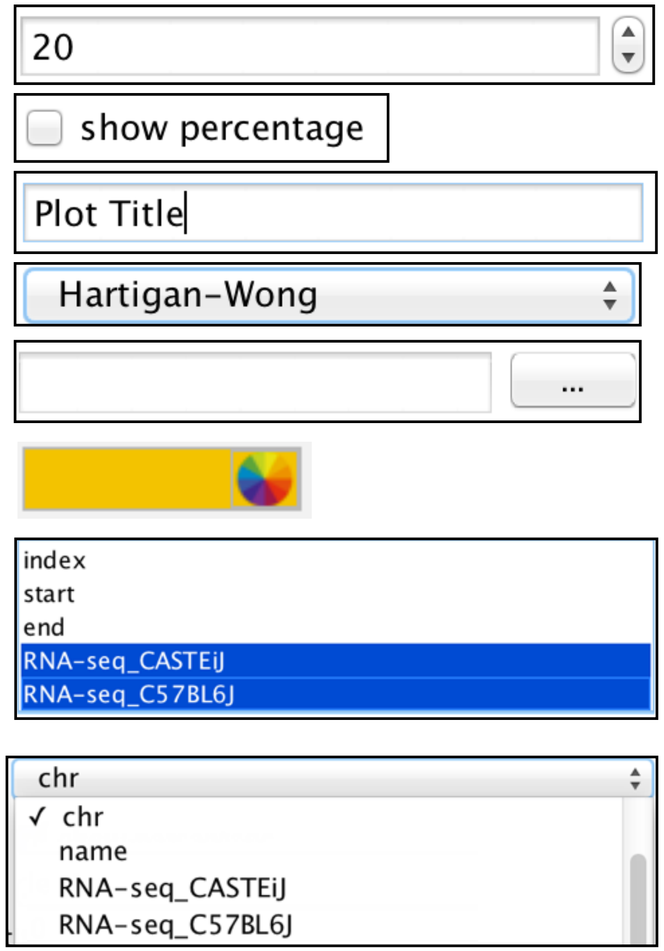


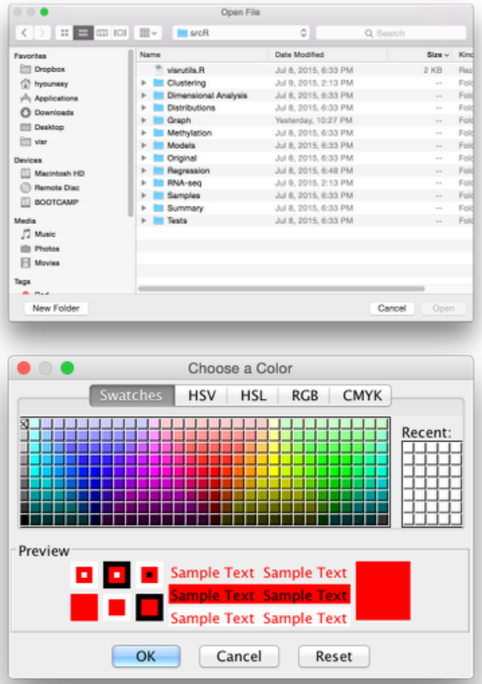
output



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
ouput-column
output-table






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

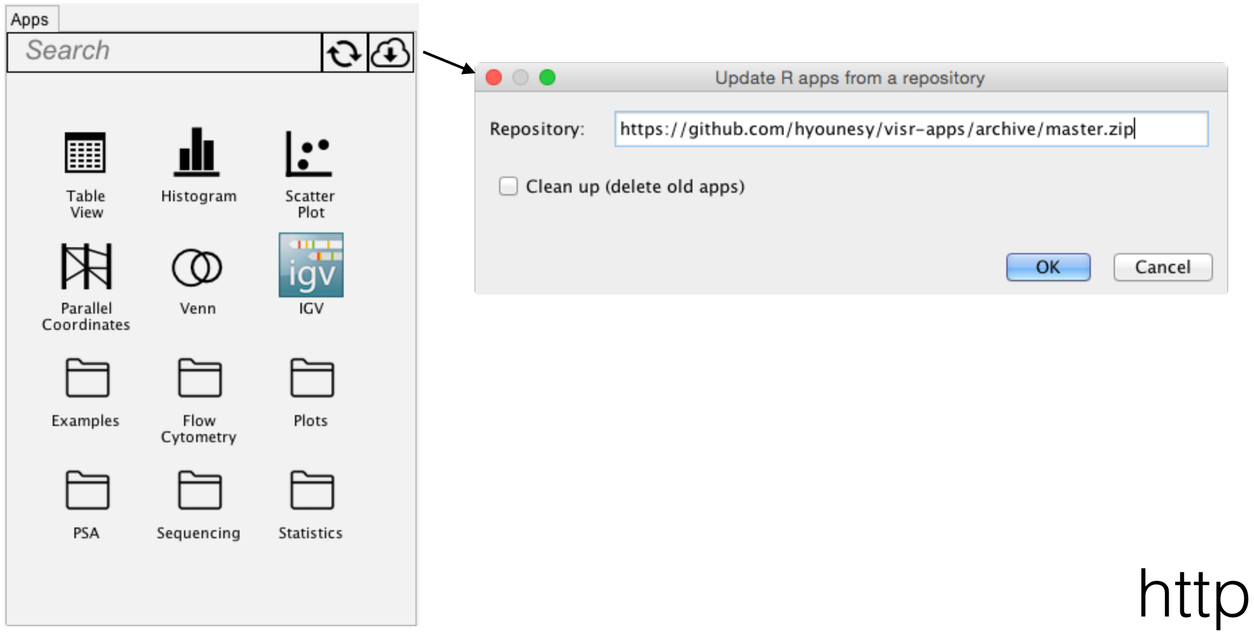
creating R apps

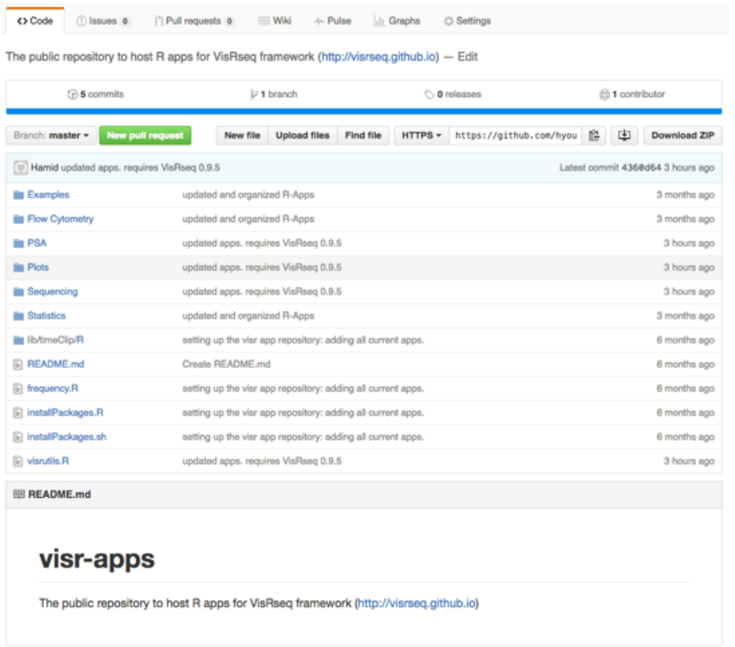
- Entire app may also be coded and debugged in R (/ RStudio)
- The .json file will be generated from the R code

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

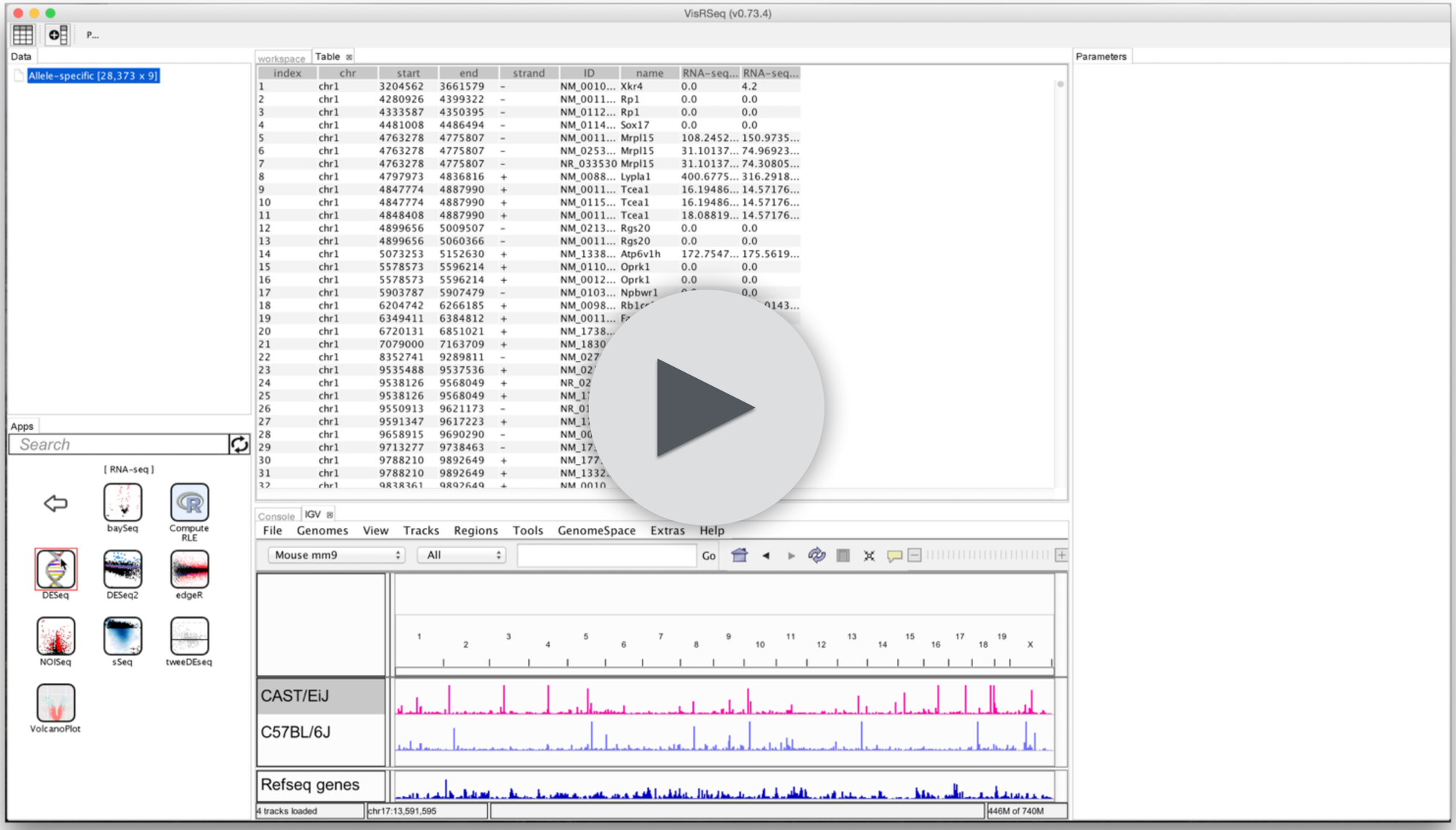




https://github.com/hyounesy/visr-apps/

Demo

Demo


VisRSeq (v0.73.4)
Data
Allele-specific [28,373 x 9]
workspace
Table
index chr start end strand ID name RNA-seq... RNA-seq...
Parameters
Apps
Search
[RNA-seq]
baySeq
Compute RLE
DESeq
DESeq2
edgeR
NOISeq
sSeq
tweeDEseq
VolcanoPlot
Console
IGV
File Genomes View Tracks Regions Tools GenomeSpace Extras Help
Mouse mm9
All
Go
CAST/EiJ
C57BL/6J
Refseq genes
4 tracks loaded
chr17:13,591,595
446M of 740M

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

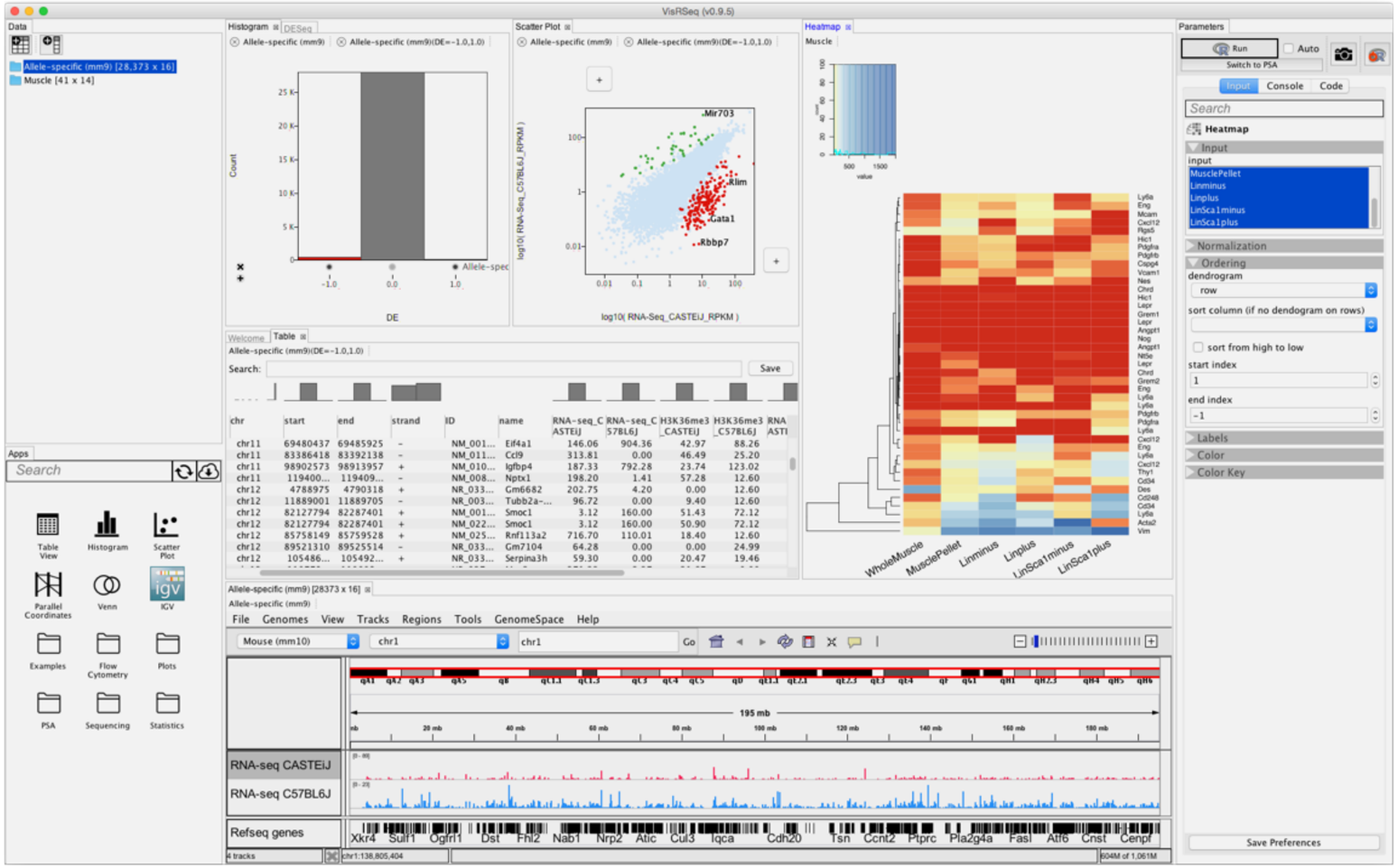
future work

- 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

questions?



visrseq.github.io

(please email for latest version)