

Computing Basics for Bioinformatics in R

Yaoyu Wang

Overview

- What is R
- Basic IO and data type
- Data type manipulation
- Set operation

What is R?

- Programs for statistical analysis
 - SAS, SPSS, STATA, ...
- Programming Language
 - Assembly, FORTRAN, COBOL → B, C, S → C++, JAVA
 - S = statistical programming language
developed by Bell Labs in the late 70s
- Evolution of S
 - S-Plus
 - **R : Free & open source implementation of S**
 - **We can use R for Free!!.**

What is R?

- slow / fast
 - Interpreted language : R, BASIC
 - Compiled language : C, FORTRAN
- easy
 - Great help manual
 - Many web resource..(<https://stat.ethz.ch/mailman/listinfo/r-help>)
- Extensible
 - Cross-platform: can run anywhere
- lots of libraries
- Can execture C, Fortran through Dynamic Link Library
- **CRAN & Bioconductor**

What is R?

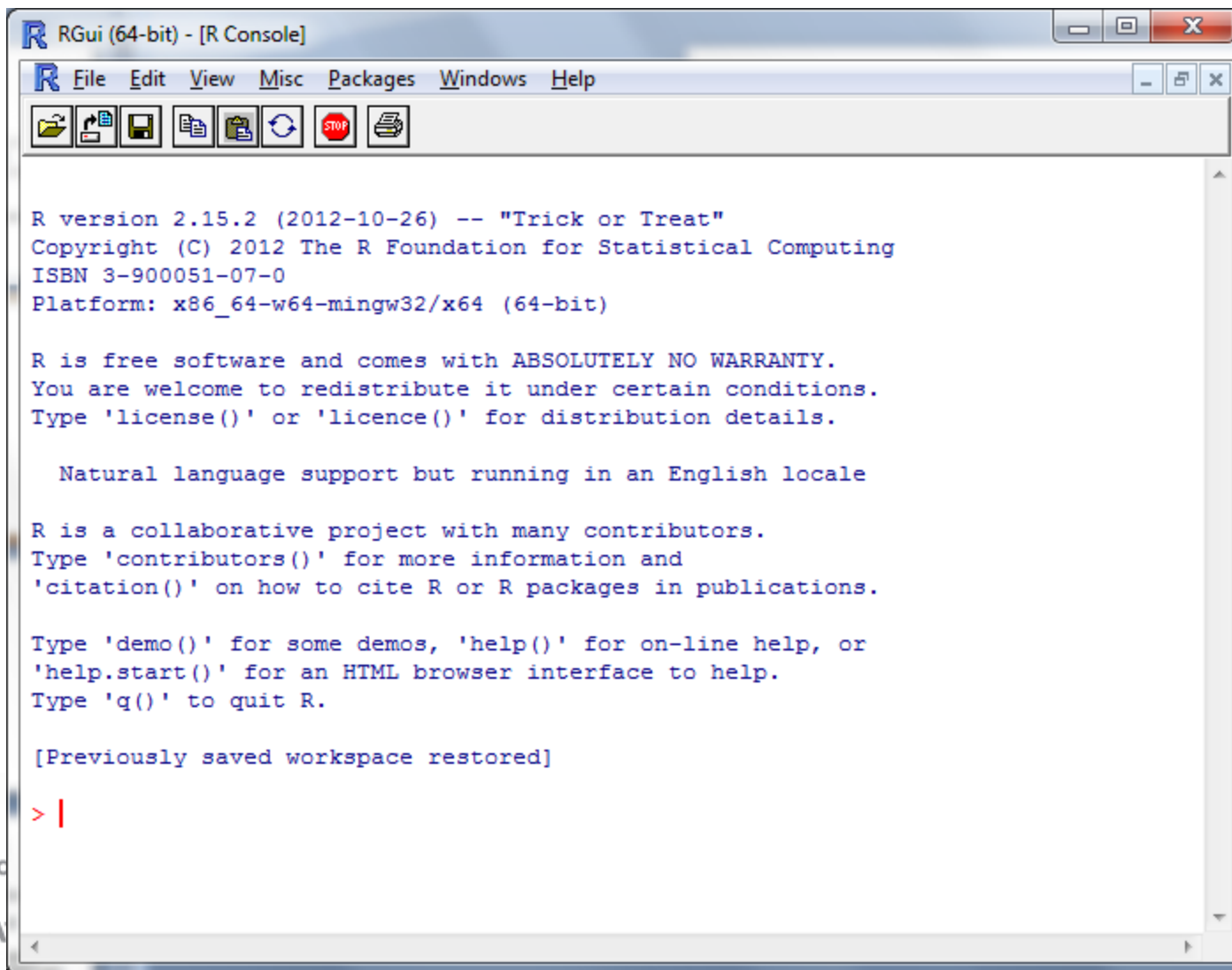
Basic of R

- R
 - Basic programming
 - Using packages
- stats
- Good reference
 - <http://www.r-project.org/>

What is R?

- Programs for statistical analysis
 - SAS, SPSS, STATA, ...
- Programming Language
 - Assembly, FORTRAN, COBOL → B, C, S → C++, JAVA
 - S = statistical programming language
developed by Bell Labs in the late 70s
- Evolution of S
 - S-Plus
 - **R : Free & open source implementation of S**
 - **We can use R for Free!!**

Starting R



```
R version 2.15.2 (2012-10-26) -- "Trick or Treat"
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> |
```

Basic Data types

- Scalar
- Vector
- Matrix
- List
- Dataframe
- Factor

Basic Data types

Data type : Scalar

- A single variable of numeric, character, and logical type

e.g. $a=10$ a variable a with value 10

Data type : Vector

- Statistical data = a set of (random) variables
- It can uniformly contain numeric, character, and logical values

e.g. $a=c(1,2,3,4)$ or $a=c("a", "b", "c", "d")$

Basic Data types

Data type : Matrix

- A set of vectors with **the same length** and same scalar data type

Example: A 2x2 matrix

```
> mdat =matrix(c(52,2,77, 11), nrow = 2, ncol=2,  
byrow=TRUE)
```

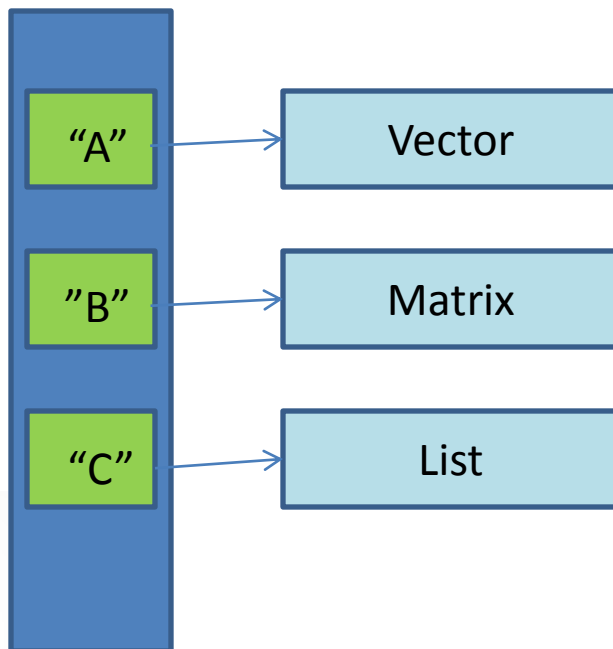
```
> mdat
```

	[,1]	[,2]
[1,]	52	2
[2,]	77	11

Basic Data types

Data type : List

- An object that contains a ***LIST*** of other objects



Example:

A=5

B=c(1,2,3,4)

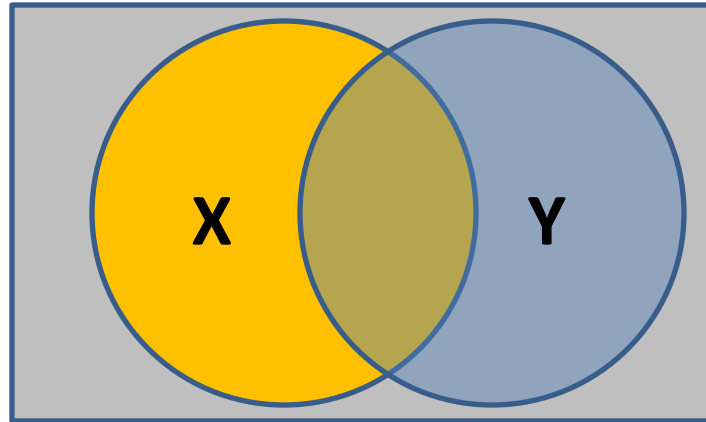
C=list()

mylist=list(A,B,C,D="test")

mylist[[1]] # equal 5

mylist["A"] # equal 5 as well

Set operation of vectors



- > $xy = \text{union}(x, y)$
- > $xy_inter = \text{intersect}(x, y)$
- > $x_diff_y = \text{setdiff}(x, y)$
- > $x_in_y = \text{is.element}(x, y)$
- > $x_in_y = x[x \%in\% y]$

R Packages and Bioconductor

- There are many contributed packages that can be used to extend R basic function
- These libraries are created and maintained by the authors.
- BioConductor is an open source and open development software project for the analysis and comprehension of genomic data.
- <http://www.bioconductor.org>
- Download > Software > Installation Instructions

```
source("http://bioconductor.org/biocLite.R")  
biocLite()
```

Basic Visualization for R

- Boxplot
- Scatterplots
- Histogram
- Heatmap

```
source("http://bioconductor.org/biocLite.R")  
biocLite("gplots")
```

(want to see what's in gplots? Do `> ls("package:gplots")`)

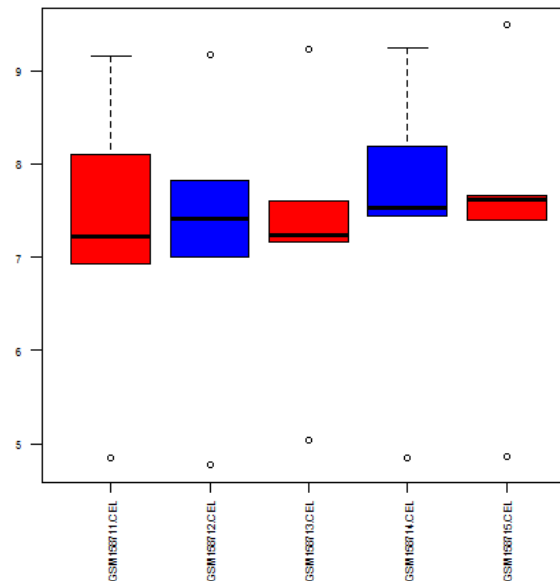
Generate Boxplot

```
boxplot(join.dat)
```

```
boxplot(join.dat, las=2)
```

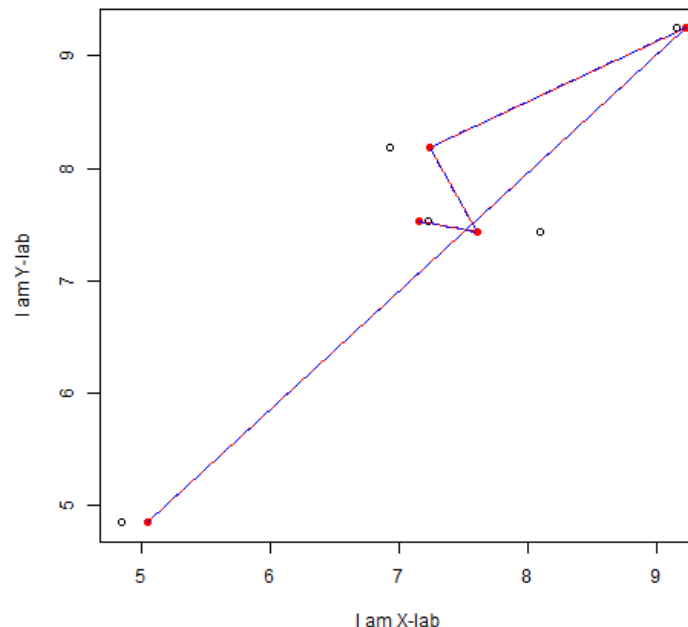
```
boxplot(join.dat, las=2, cex.axis=0.8)
```

```
boxplot(join.dat, las=2, cex.axis=0.7, col=c("red", "blue"))
```



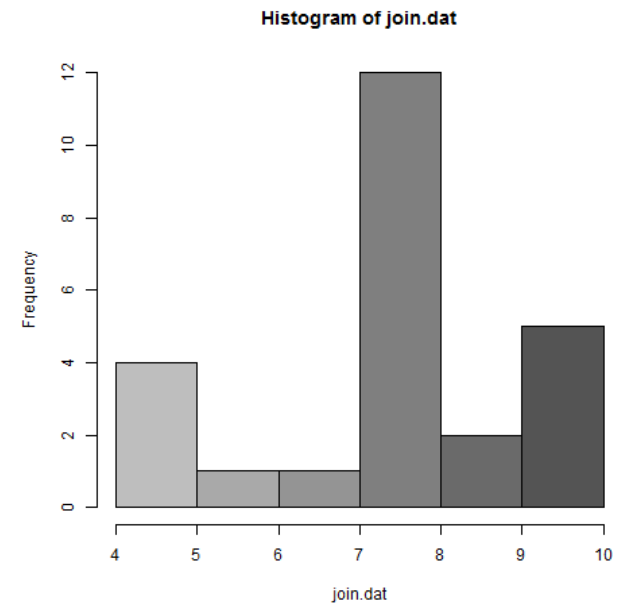
Scatter plot with connecting lines

```
plot(join.dat[,1], join.dat[,4], xlab="I am X-lab", ylab="I am Y-lab")  
points(join.dat[,3], join.dat[,4])  
points(join.dat[,3], join.dat[,4], col="red", pch=16)  
lines(join.dat[,3], join.dat[,4], col="red", pch=16)  
lines(join.dat[,3], join.dat[,4], col="blue", pch=16, lty="dashed")
```



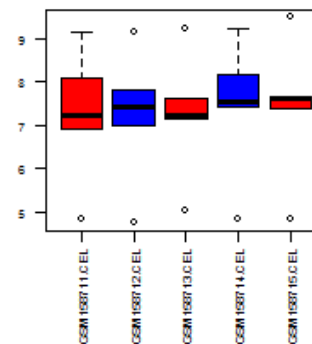
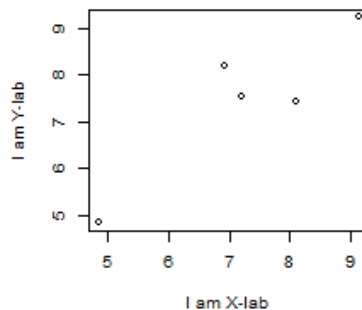
A histogram of all the probes intensity

```
join.dat=as.matrix(join.dat)  
hist(join.dat)  
hist(join.dat, col=colorpanel(10,"grey", "black"))
```

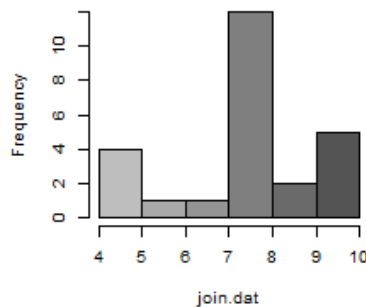


Multiple Plots Together

```
par(mfrow=c(2,2))  
plot(join.dat[,1], join.dat[,4], xlab="I am X-lab", ylab="I am Y-lab")  
boxplot(join.dat, las=2, cex.axis=0.7, col=c("red", "blue"))  
hist(join.dat, col=colorpanel(10,"grey", "black"))
```



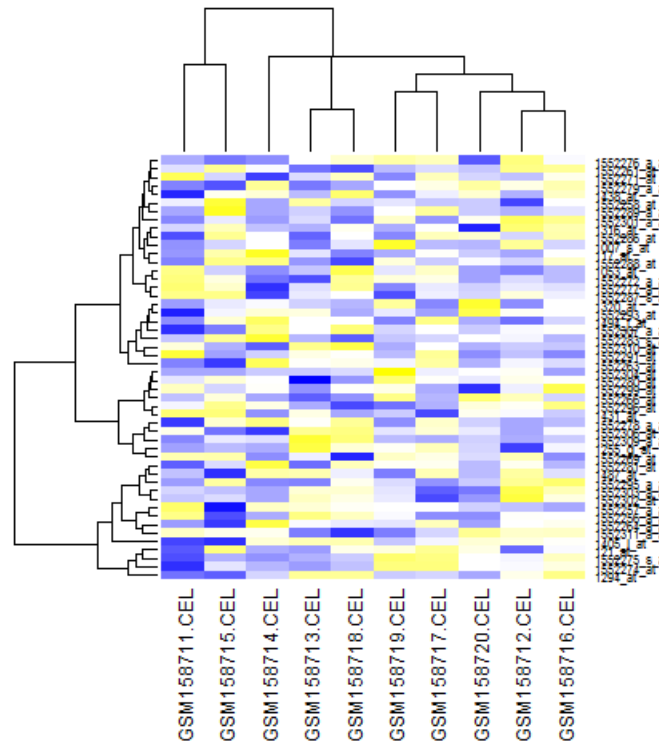
Histogram of join.dat



Heatmap

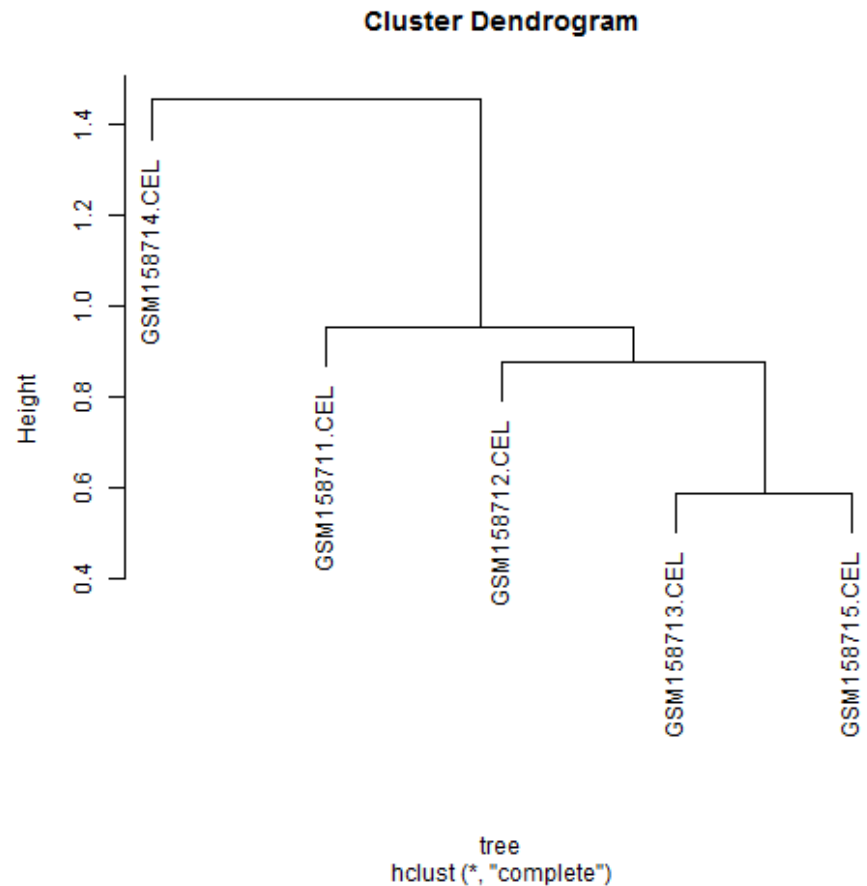
```
heatmap(join.dat)
heatmap(join.dat, col=colorpanel(100, "blue", "white", "yellow"))
heatmap(join.dat, col=colorpanel(100, "blue", "white", "yellow"),
margin=c(10,10))

heatmap(as.matrix(in.data[1:50,]), col=colorpanel(100, "blue", "white", "yellow"),
margin=c(10,10))
```



Hierarchical Cluster Tree

```
dist(join.dat)
t(join.dat)
tree=dist(t(join.dat))
hclust(tree)
```



Exercise

Let's generate boxplots and figures for the expression patterns of all the **cell cycle** genes located on **chr17**

The probe/Gene name file is at:

</apps/ComputingBasics/cellcycle.chr17.txt>

Just to get you start

set target file and read in

```
gexp.file="C:\\Users\\Yaoyu\\Desktop\\ComputingBasics\\example.rma.cll.txt"  
gexp=read.table(gexp.file, header=TRUE, sep="\t")
```

set probe file and read in

```
probe.file="C:\\Users\\Yaoyu\\Desktop\\ComputingBasics\\cellcycle.chr17.txt"  
probe=read.table(probe.file, header=TRUE, sep="\t")
```

