Memo R basics

université de BORDEAUX

Plan

I. R Basics
   A. What is R
   B. Ressources
   C. Basic knowledge (variable type, classic functions)

II. R objects for NGS
   A. Data bases
   B. Genomic object
   C. Specific tools

III. R plots

R Basics

What is R?

- A calculator

```
## All mathematical / statistical functions implemented in the basics
(5+5)*10^4
sqrt(16) / factorial(50)
```

- A programmation langage

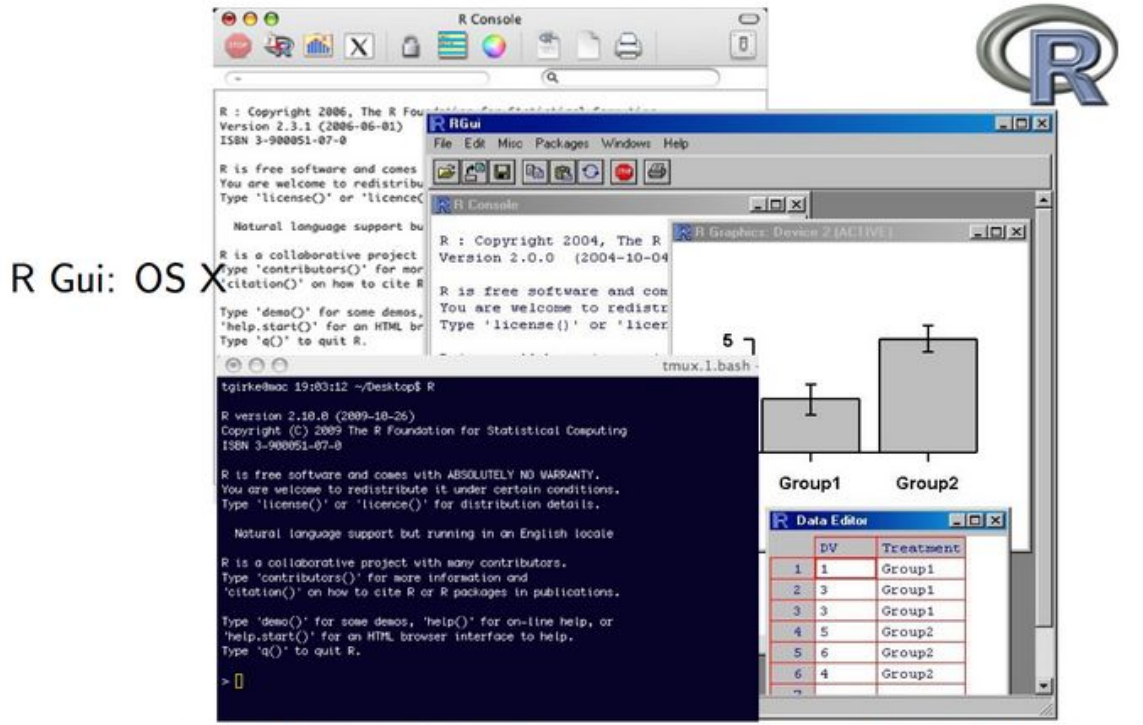
```
## Automatisation of repetitive tasks / functions
for (i in 1:50){
        print(i^2)
}
```

```
hello.world <- function(){
        print("Hello World")
}
```

- An open source software

- A very active community that have already provided > 4000 extensions.

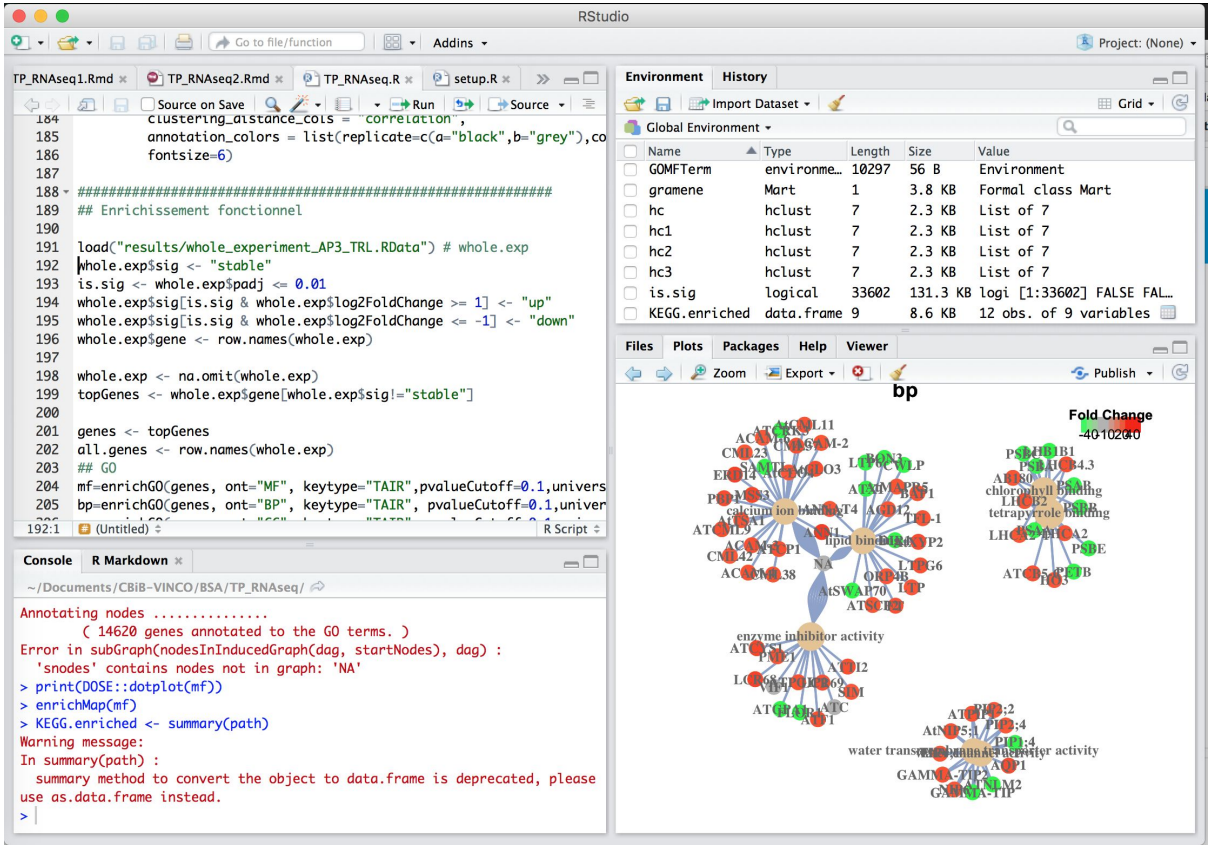
How to play with R? Console



Environnements R par défaut

How to play with R? RStudio

Interactive, integrated, visual environment



Ressources

- R Packages
  - When downloading R, it comes with “basic” functions
  - Due to R popularity, thousands of packages **freely** available
- Where to get the packages
  - Principal repository: **CRAN** (https://cran.r-project.org/)
    - ~ 10,000 packages organized by views

Ressources

- R Packages
  - When dow… functions
  - Due to R p… **freely** available
- Where to get the pa…
  - Principal re… oject.org/)
    - ~ 10,0…

CRAN Task Views

Bayesian	Bayesian Inference
ChemPhys	Chemometrics and Computational Physics
ClinicalTrials	Clinical Trial Design, Monitoring, and Analysis
Cluster	Cluster Analysis & Finite Mixture Models
DifferentialEquations	Differential Equations
Distributions	Probability Distributions
Econometrics	Econometrics
Environmetrics	Analysis of Ecological and Environmental Data
ExperimentalDesign	Design of Experiments (DoE) & Analysis of Experimental Data
ExtremeValue	Extreme Value Analysis
Finance	Empirical Finance
FunctionalData	Functional Data Analysis
Genetics	Statistical Genetics
Graphics	Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization
HighPerformanceComputing	High-Performance and Parallel Computing with R
MachineLearning	Machine Learning & Statistical Learning
MedicalImaging	Medical Image Analysis
MetaAnalysis	Meta-Analysis
Multivariate	Multivariate Statistics
NaturalLanguageProcessing	Natural Language Processing
NumericalMathematics	Numerical Mathematics
OfficialStatistics	Official Statistics & Survey Methodology
Optimization	Optimization and Mathematical Programming
Pharmacokinetics	Analysis of Pharmacokinetic Data
Phylogenetics	Phylogenetics, Especially Comparative Methods
Psychometrics	Psychometric Models and Methods
ReproducibleResearch	Reproducible Research
Robust	Robust Statistical Methods
SocialSciences	Statistics for the Social Sciences
Spatial	Analysis of Spatial Data
SpatioTemporal	Handling and Analyzing Spatio-Temporal Data
Survival	Survival Analysis
TimeSeries	Time Series Analysis
WebTechnologies	Web Technologies and Services
gR	gRaphical Models in R

Ressources

- R Packages
  - When downloading R, it comes with “basic” functions
  - Due to R popularity, thousands of packages **freely** available

- Where to get the packages?
  - Principal: **CRAN** (https://cran.r-project.org/)
    - ~ 10,000 packages organized by views
    - principal function is install.packages()

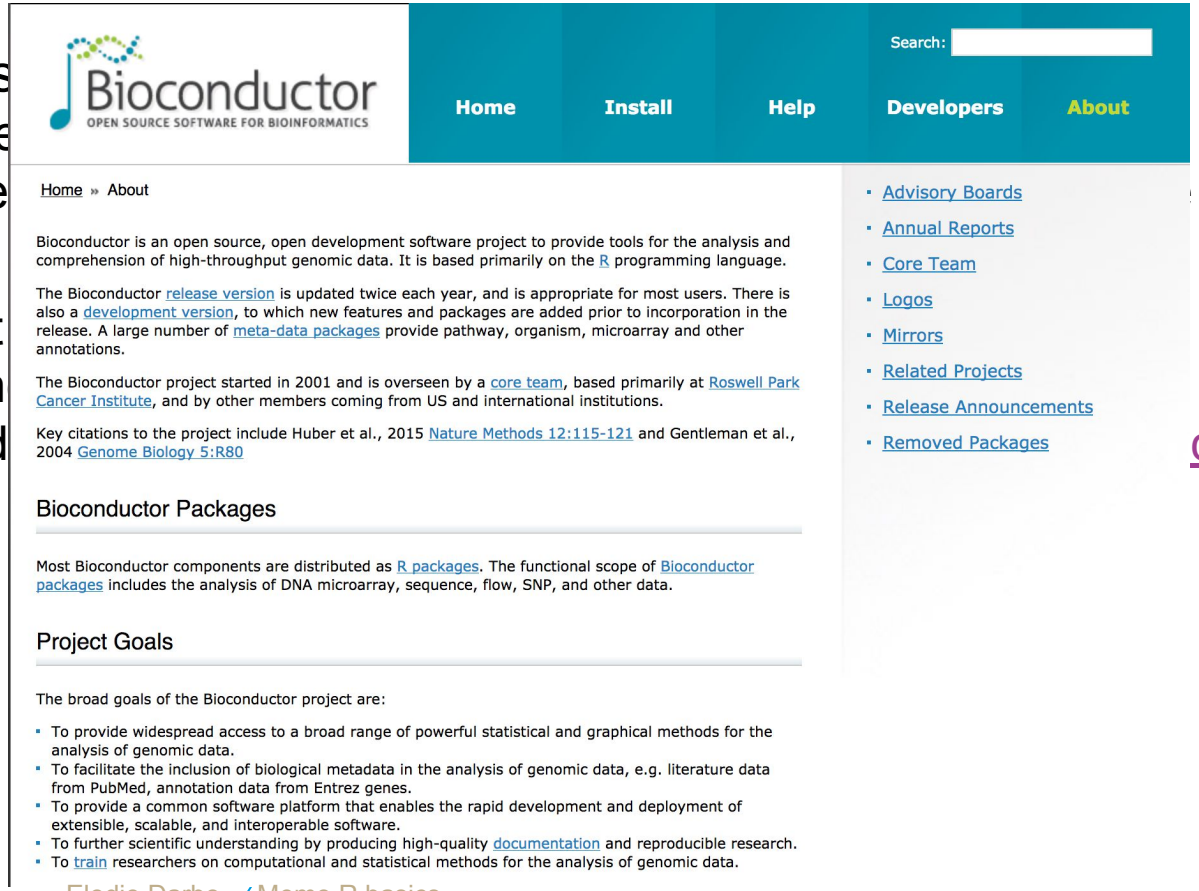
Ressources

- R Packages
  - When downloading R, it comes with “basic” functions
  - Due to R popularity, thousands of packages **freely** available

- Where to get the packages?
  - Principal: **CRAN** (https://cran.r-project.org/)
  - Dedicated to bioinformatics: **BioConductor** (http://bioconductor.org/)
    - More than 10 years experience and 1000 packages

Ressources

- R Packages
  - Whe
  - Due
- Where to get
  - Prin
  - Ded
    -



ctor.org/)

Ressources

- R Packages
  - When downloading R, it comes with “basic” functions
  - Due to R popularity, thousands of packages **freely** available

- Where to get the packages?
  - Principal: **CRAN** (https://cran.r-project.org/)
  - Dedicated to bioinformatics: **BioConductor** (http://bioconductor.org/)
    - More than 10 years experience and 1000 packages
    - Need to source the biocLite() function from BioConductor

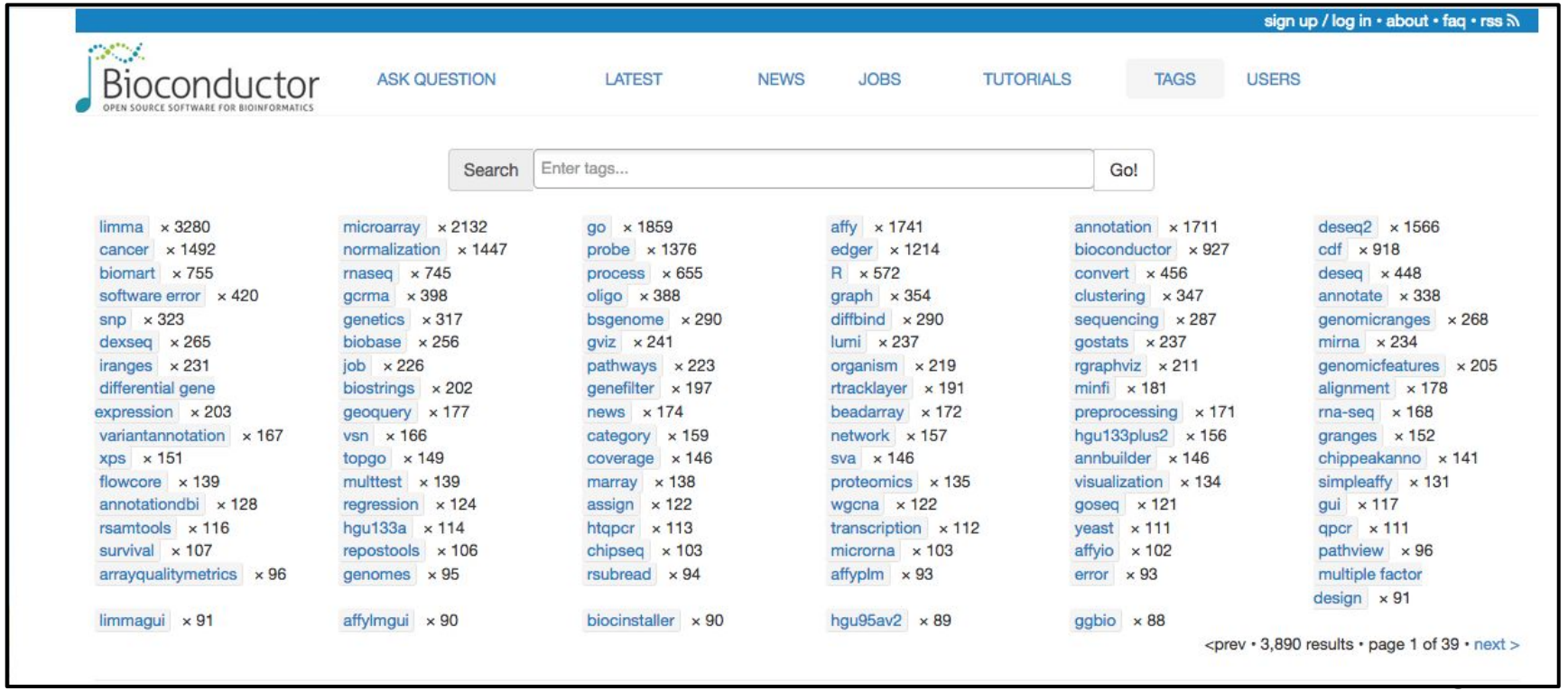
```
source("http://bioconductor.org/biocLite.R")
biocLite("package.name")
```

Ressources

- R Packages
  - When downloading R, it comes with “basic” functions
  - Due to R popularity, thousands of packages **freely** available

- Where to get the packages?
  - Principal: **CRAN** (https://cran.r-project.org/)
  - Dedicated to bioinformatics: **BioConductor** (http://bioconductor.org/)
    - More than 10 years experience and 1000 packages
    - Need to source the `biocLite()` function from BioConductor
    - Forum dedicated to bioinformatics and BioConductor packages (https://support.bioconductor.org/t/) - No one can know all packages.

Ressources



(https://support.bioconductor.org/t/) - No one can know all packages.

Which type of variable?

No dimension

- Numeric (float)

```
x <- c(3,2,0,5);y <- "85"
is.numeric(x); as.numeric(y)
```

- Integer

```
is.integer(x); as.integer(x)
```

- Character / String

```
as.character(x); nchar("Boo")
paste("Boo", "Boo")
```

- Logical (T/F)

```
3 %in% x; !(x < 3); TRUE==T
which(x < 3); T+TRUE+F+FALSE
```

Undefined values

- Non existing number

```
## NaN, Inf
x / 0
```

- Null data (NULL)

```
z <- NULL; length(z); is.null(z)
z + 2
```

- Missing data (NA)

```
x <- c(3,2,0,5,NA); class(x)
is.na(x); na.omit(x)
mean(x); mean(x,na.rm=T)
```

Combine elements: Constraint on type

Vector (1D)

```
x <- c(3,2,0,5)
y <- c("85",8,9)
x[2]
seq(1,6,1)
rep(y,3); 1:6; x[2:3]
x^2; x + x[2:3]
```

matrix (2D) / Arrays (ND)

```
# 2 dimensions
x2 <- matrix(1:12,nrow=4,ncol=3,byrow=TRUE)
x2[3,2]; x2[,2]
x2 + x2
x2 + c(3,2,5)
x2[,1] <- c(3,2,5)
dim(x2); row.names(x2)
# 3 dimensions
x3 <- array(1:12,dim=c(2,2,3))
x3[1,2,2] ; x3[1,2,]
```

Factor: categorical vector(1D)

```
# limited number of different values, encoded in integer, used by read.table()
data <- factor(c(3,2,0,5,3,2,0,5,3,2,0,5))
as.integer(data)
data==5; data=="5"
sum(data)
levels(data) <- c("bleu","jaune","rouge","gris")
```

In a general manner, to access the data:

```
x[indexes.dim1]
x2[indexes.dim1,indexes.dim2]
x3[indexes.dim1,indexes.dim2,indexes.dim3]
xN[indexes.dim1,indexes.dim2,indexes.dim3,...,indexes.NDim]
```

Combine elements: Not constraint on type

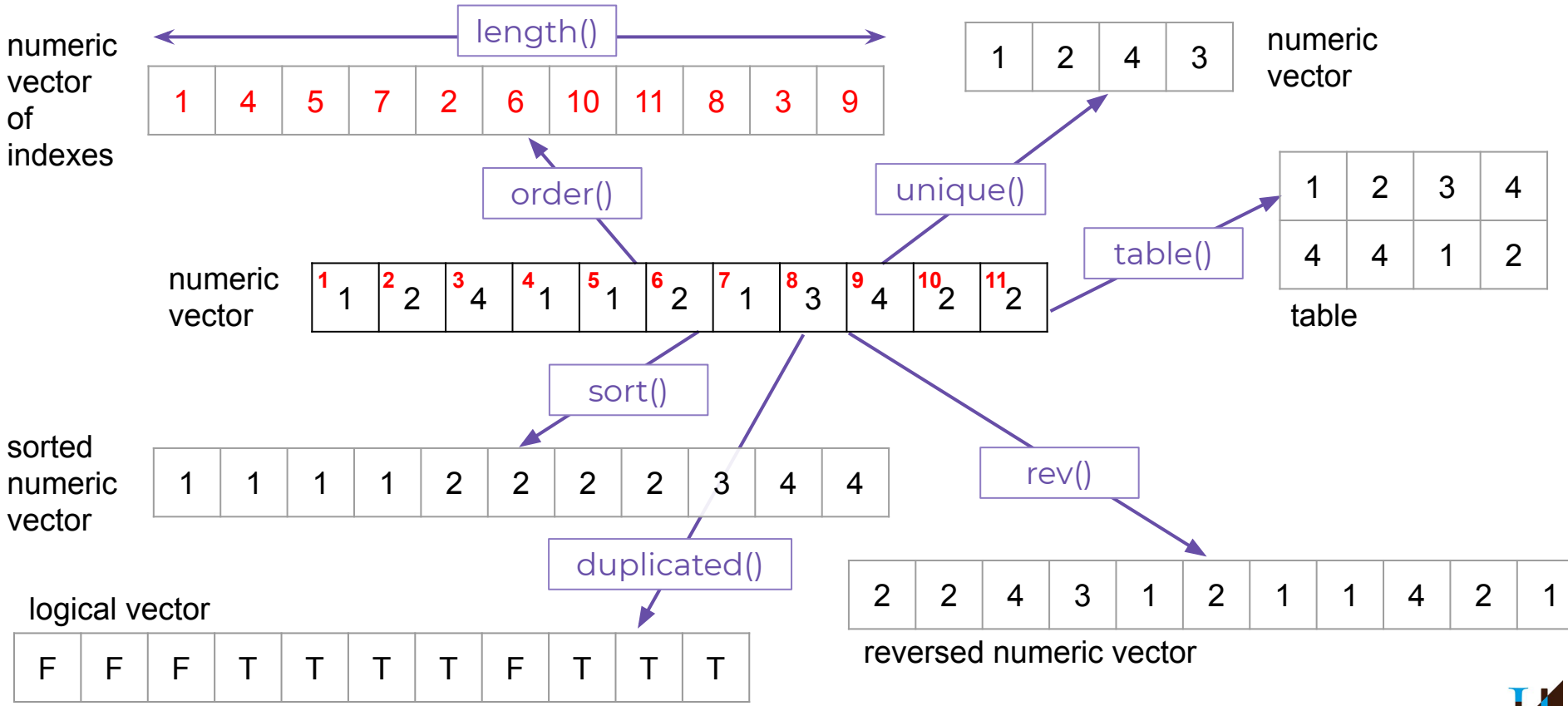
- data frame (2D)

```
# collection of vectors and/or factors
# constrained by column
firstNames <- c("Remy", "Lol", "Pierre", "Domi", "Ben")
IMC <- data.frame(sex=c("H", "F", "H", "F", "H") ,
                  height=c(1.83,1.76,1.82,1.60,1.90),
                  weight=c(67,58,66,48,75),
                  row.names=firstNames)

IMC$height
IMC[,"sex"]
IMC["Remy",]
IMC[3,2]
head(IMC)
colnames(IMC)
```

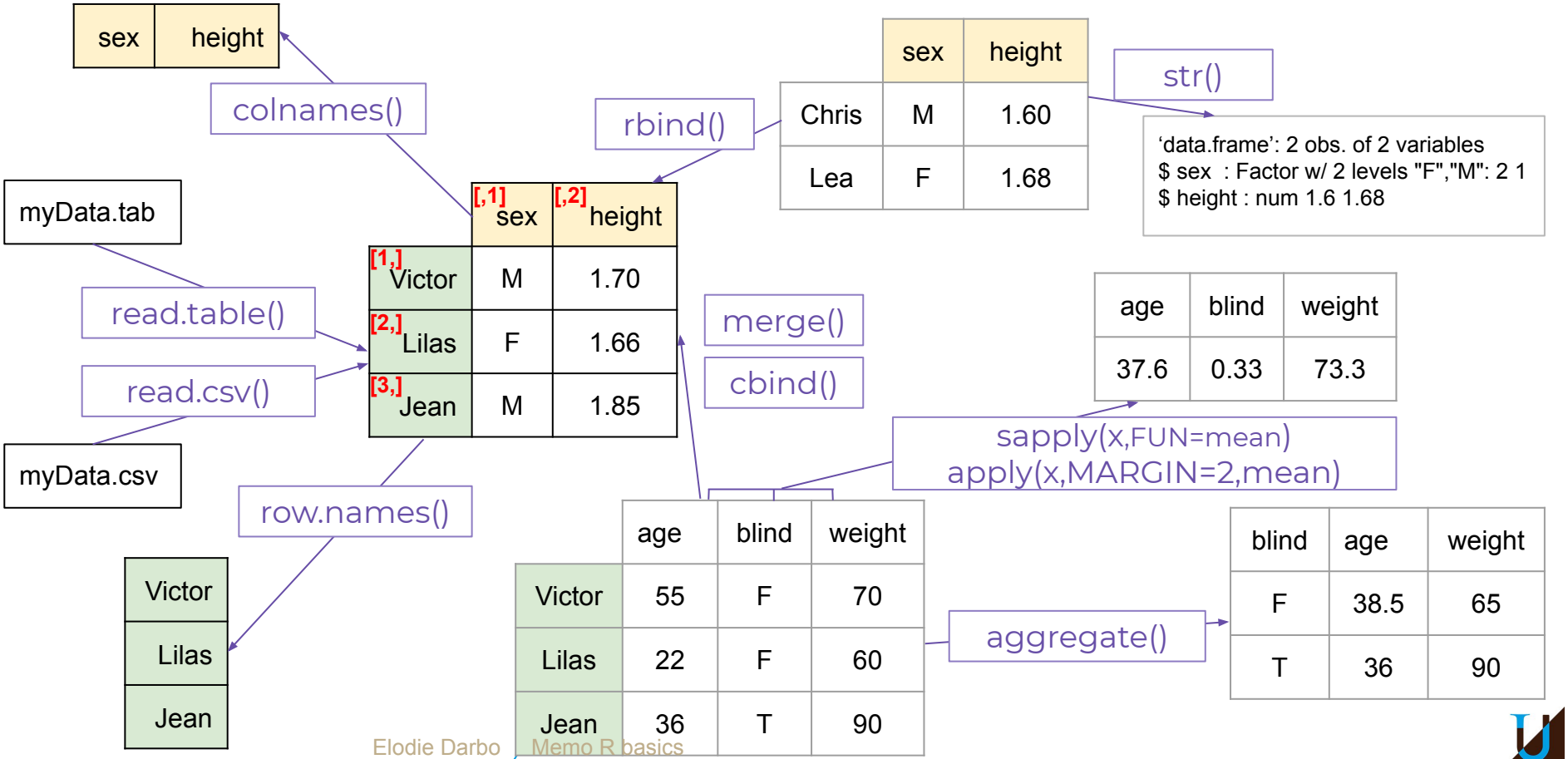
- list

```
# very flexible, store everything
list.ex <- list(one_vec=1:12,
        one_name="Boo",
        one_tab=matrix(1:4,nrow=2))

list.ex$one_tab
list.ex[[1]]
list.ex$new <- list(a="a",b="b")
list.ex$new$a
```

Manipulate vectors



Manipulate data frames



How would you?

```
load("TP_RNAseq/R_course_objects.RData")
# We will use the data.frame objects imc and imc_age
```

- Assemble the 2 data frames?
    - What do they have in commun?
- Get the mean weight per gender?
- Get the mean values for all numeric information?

How would you? … Answer

```
load("TP_RNAseq/R_course_objects.RData")
# We will use the data.frame objects imc and imc_age
```

- Assemble the 2 data frames?
  - What do they have in commun?
- Get the mean weight per gender?
- Get the mean values for all numeric information?

```
# merge the data frames
imc_merged=merge(imc, imc_age, by=c( "prenom", "sexe"))
# Aggregate
aggregate(poids~sexe,imc_merged,mean)
# get numeric columns
numCol <- sapply(imc_merged,is.numeric) # returns logical
imc_num <- imc_merged[,numCol]
# apply on columns the function mean
apply(imc_num,2,mean)
sapply(imc_merged[,numCol],mean)
```

R Objects for NGS

Why using R for NGS analysis?

To replace existing command-line tools?

- ❏ R NGS treatment built on existing tools (BWA, BowTie, Samtools ...) and data bases (EnsEMBL ...)

To have a analysis workflow?

- ❏ No magic ready-to-use solution, constant evolving tools, constant new, more performant methods. Package documentation **must** be consulted !

To take advantage of statistical methods?

To manipulate large data (consulting, filtering, modifying)?

- ❏ Import/export of NGS format files (BAM, vcf etc) in specific R objects

To visualise graphically data and results?

NGS oriented packages & functions

nucleic/protein sequences (FASTA)
GC content, pattern matching
Biostring: e.g. getSeq(), vcoutPattern()

reads (FASTQ)
Quality control, cleaning
ShortRead: e.g. FastqSampler(), readFastq(), qa(), report(), sread()

alignment (BAM/SAM)
Manipulate, storage
GenomicAlignment: e.g. filterBam(), readAlignment(), scanBam()

SNPs (VCF)
Predict protein impact, prioritize mutations (drivers)
VariantAnnotation e.g. readVcf(), rowRanges(), predictCoding()

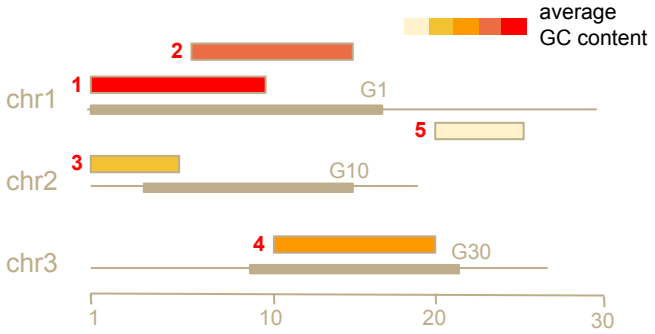
regions (GFF/BED)
Genomic interval overlap, storage
GenomicRanges e.g. findOverlaps(), intersect(),reduce()

metadata (GFF/BED/WIG/ucsc tracks)
Annotations with informations on the biological objects (gene, genomic region)
BiomaRt e.g. getBM()
rtracklayer e.g. import(), export()
GenomicFeatures e.g. makeTxDB()

all-in-one (from FASTQ to specific results)
Build a workflow with no problem of compatibility
QuasR e.g. qa(), qAlign(), qCount()
Rsubread e.g. buildindex(), align(), featureCounts(), extractSNP()

Genomic regions: extract info from *GRanges*

GRanges(seqnames, range=IRanges(start,end), strand, gene, GC)



numeric vectors

start()

end()

DataFrame with 10 rows and 2 columns

values()
mcols()

	seqnames <Rle>	start <IRanges>	end <IRanges>	strand <Rle>	gene <character>	GC <numeric>
1	chr1	1	10	+	G1	40
2	chr1	6	15	+	G1	35
3	chr2	1	5	+	G10	26
4	chr3	10	20	*	G30	30
5	chr1	20	25	-	NA	23

character vectors

“chr1”	“chr1”	“chr2”	“chr3”	“chr1”

as.vector()

character-Rle of length 5 with 4 runs

Lengths	2	1	1	1
Values	“chr1”	“chr2”	“chr3”	“chr1”

seqnames()

strand()

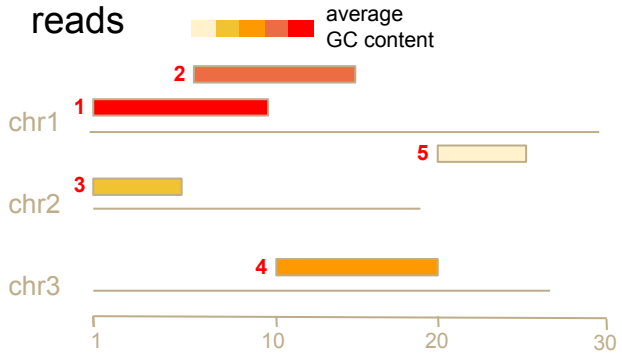
range()

character-Rle of length 5 with 3 runs

Lengths	3	1	1
Values	“+”	“*”	“-”

GRanges object with 7 ranges and 0 metadata columns

Genomic regions: manipulate *GRanges*

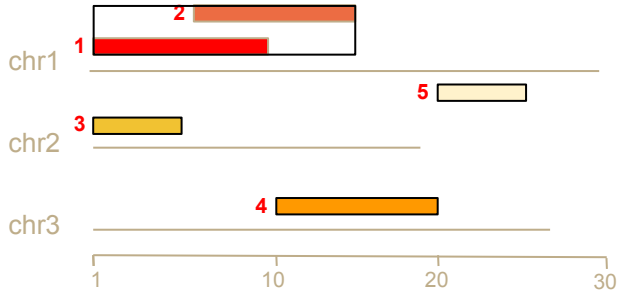


GRanges object with 5 ranges and 1 metadata columns

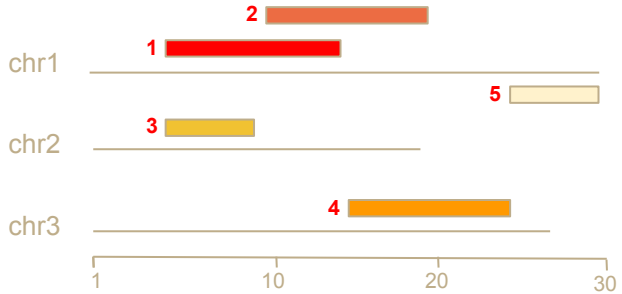
reduce(reads)

disjoin(reads)

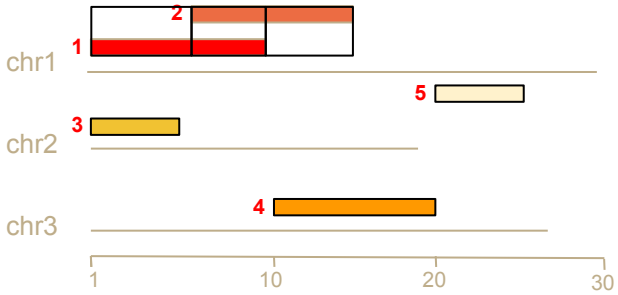
shift(reads,5)



GRanges object with 4 ranges and 1 metadata columns



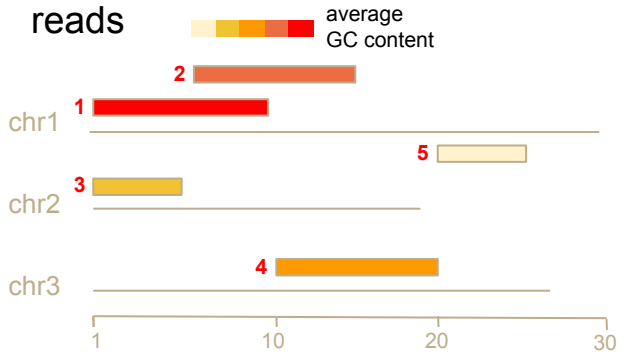
GRanges object with 5 ranges and 1 metadata columns



GRanges object with 6 ranges and 1 metadata columns

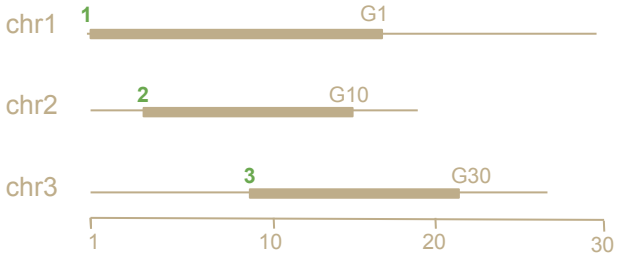
Genomic regions: manipulate *GRanges*

reads



GRanges object with 5 ranges and 1 metadata columns

genomic_features



GRanges object with 3 ranges and 1 metadata columns

	queryHits <integer>	subjectHits <integer>
1	1	1
2	2	1
3	3	2
4	4	3

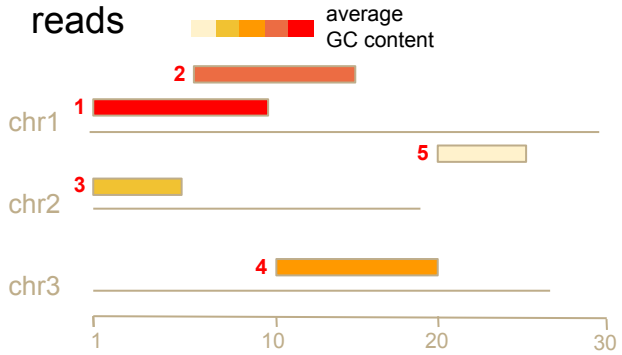
queryLength: 5 / subjectLength: 3

findOverlaps(reads,genomic_features)

Hits object with 4 hits and 0 metadata columns

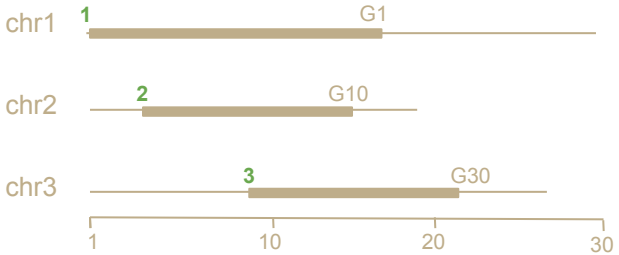
Genomic regions: manipulate *GRanges*

reads



GRanges object with 5 ranges and 1 metadata columns

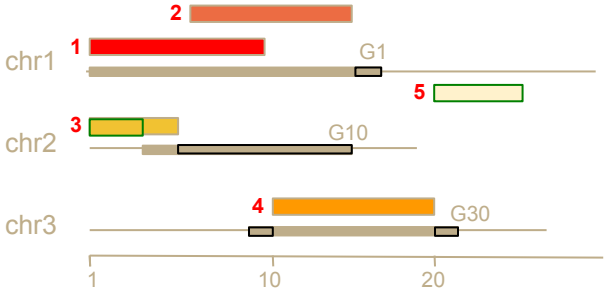
genomic_features



GRanges object with 3 ranges and 1 metadata columns

setdiff(genomic_features,reads)

GRanges object with 4 ranges and 0 metadata columns

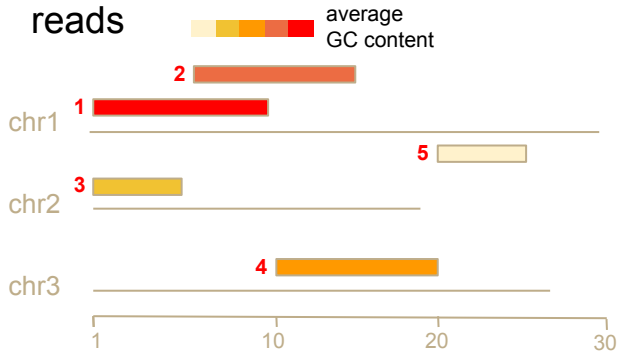


setdiff(reads,genomic_features)

GRanges object with 2 ranges and 0 metadata columns

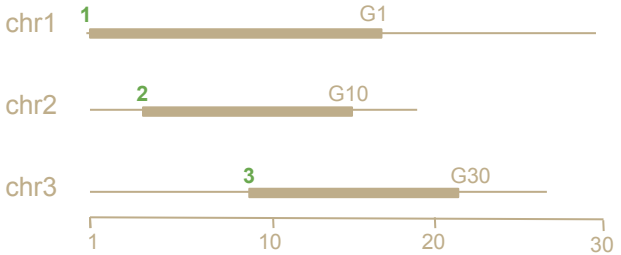
Genomic regions: manipulate *GRanges*

reads



GRanges object with 5 ranges and 1 metadata columns

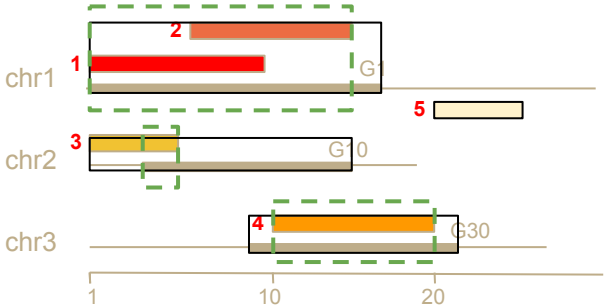
genomic_features



GRanges object with 3 ranges and 1 metadata columns

intersect(reads,genomic_features)

GRanges object with 3 ranges and 0 metadata columns



union(reads,genomic_features)

GRanges object with 4 ranges and 0 metadata columns

Annotation Objects

By species and genome version

Give access to genomic sequences

Give access to gene, transcript genomic positions

Give access to gene functional annotations

GenomicFeatures: build TxDB objects from scratch

makeTxDb(transcripts, splicings, genes)

data.frame with 6 columns and a row per exon

tx_id	exon_rank	exon_start	exon_end	cds_start	cds_end
1	1	1	999	1	999
2	1	2001	2085	2022	2085
2	2	2101	2144	2101	2144
2	3	2131	2199	2131	2193
3	1	2001	2085	NA	NA
3	2	2131	2199	NA	NA

data.frame with 5 columns and a row per transcript

tx_id	tx_chrom	tx_strand	tx_start	tx_end
1	chr1	-	1	999
2	chr1	+	2001	2199
3	chr1	+	2001	2199

data.frame with 2 columns and a row per gene

tx_id	gene_id
1	gene1
2	gene2
3	gene3

GenomicFeatures: build TxDB objects from known annotations

TAIR10_GFF3_trunc.gff

makeTxDbFromGFF()

rtracklayer::import()

GRanges object with 1941 ranges
and 5 metadata columns

makeTxDbFromGRanges()

bio mart

(http://www.biomart.org/)
(http://www.ensembl.org/biomart/martview/)

makeTxDbFromBiomart()

Needs internet connection

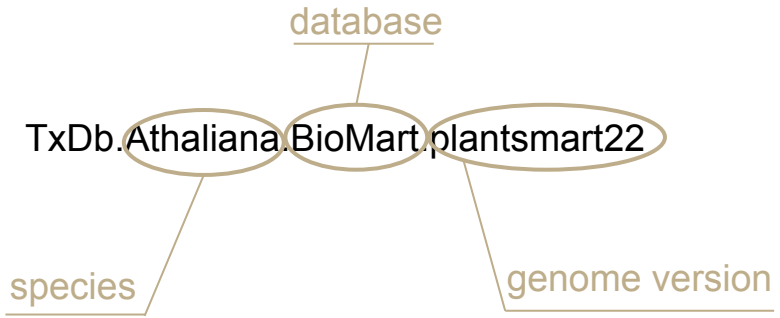
UCSC

makeTxDbFromUCSC()

GenomicFeatures: Ready-to-use TxDB objects

There is a collection of packages containing TxDB objects in BioConductor. You can find a list in the BioConductor GenomicFeatures page.
https://bioconductor.org/packages/release/bioc/html/GenomicFeatures.html

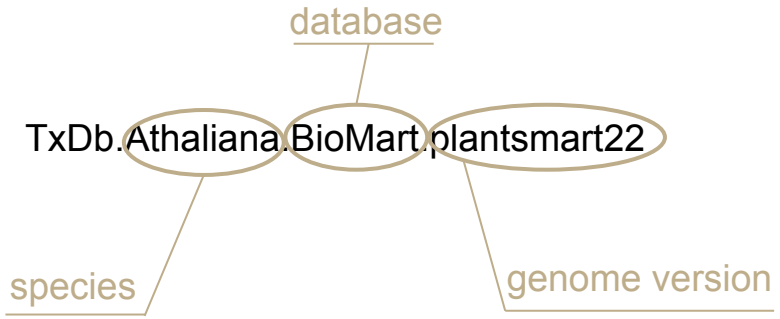
TxDb.Athaliana.BioMart.plantsmart25, TxDb.Athaliana.BioMart.plantsmart28, TxDb.Btaurus.UCSC.bosTau8.refGene, TxDb.Celegans.UCSC.ce11.refGene, TxDb.Celegans.UCSC.ce6.ensGene, TxDb.Cfamiliaris.UCSC.canFam3.refGene, TxDb.Dmelanogaster.UCSC.dm3.ensGene, TxDb.Dmelanogaster.UCSC.dm6.ensGene, TxDb.Drerio.UCSC.danRer10.refGene, TxDb.Ggallus.UCSC.galGal4.refGene, TxDb.Ggallus.UCSC.galGal5.refGene, TxDb.Hsapiens.BioMart.igis, TxDb.Hsapiens.UCSC.hg18.knownGene, TxDb.Hsapiens.UCSC.hg19.knownGene, TxDb.Hsapiens.UCSC.hg19.lincRNAsTranscripts, TxDb.Hsapiens.UCSC.hg38.knownGene, TxDb.Mmulatta.UCSC.rheMac3.refGene, TxDb.Mmulatta.UCSC.rheMac8.refGene, TxDb.Mmusculus.UCSC.mm10.ensGene, TxDb.Mmusculus.UCSC.mm10.knownGene, TxDb.Mmusculus.UCSC.mm9.knownGene, TxDb.Ptroglodytes.UCSC.panTro4.refGene, TxDb.Rnorvegicus.BioMart.igis, TxDb.Rnorvegicus.UCSC.rn4.ensGene, TxDb.Rnorvegicus.UCSC.rn5.refGene, TxDb.Rnorvegicus.UCSC.rn6.refGene, TxDb.Scerevisiae.UCSC.sacCer2.sgdGene, TxDb.Scerevisiae.UCSC.sacCer3.sgdGene, TxDb.Sscrofa.UCSC.susScr3.refGene



GenomicFeatures: Ready-to-use TxDB objects

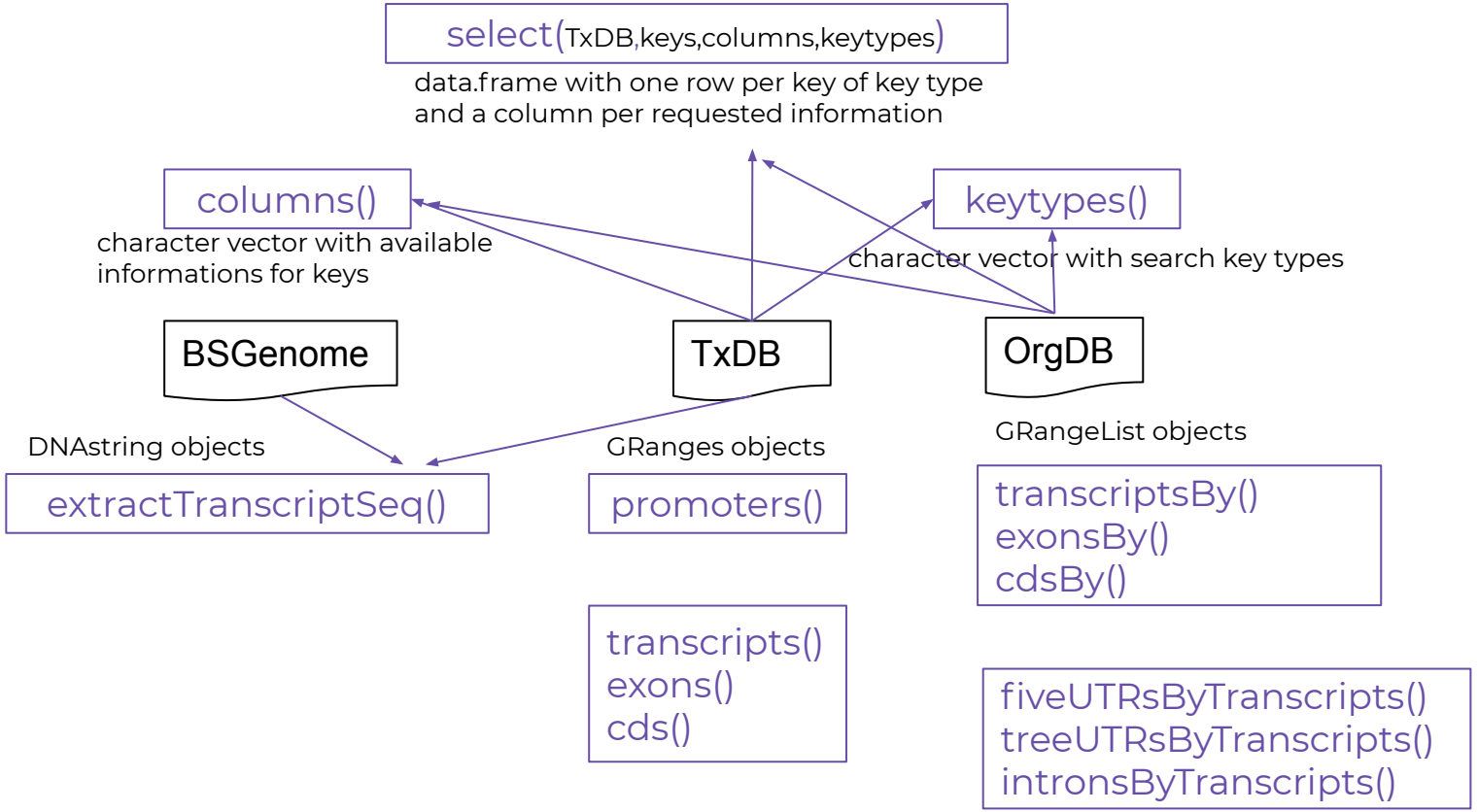
There is a collection of packages containing TxDB objects in BioConductor. You can find a list in the BioConductor GenomicFeatures page.
https://bioconductor.org/packages/release/bioc/html/GenomicFeatures.html

TxDb.Athaliana.BioMart.plantsmart25, TxDb.Athaliana.BioMart.plantsmart28, TxDb.Btaurus.UCSC.bosTau8.refGene, TxDb.Celegans.UCSC.ce11.refGene, TxDb.Celegans.UCSC.ce6.ensGene, TxDb.Cfamiliaris.UCSC.canFam3.refGene, TxDb.Dmelanogaster.UCSC.dm3.ensGene, TxDb.Dmelanogaster.UCSC.dm6.ensGene, TxDb.Drerio.UCSC.danRer10.refGene, TxDb.Ggallus.UCSC.galGal4.refGene, TxDb.Ggallus.UCSC.galGal5.refGene, TxDb.Hsapiens.BioMart.igis, TxDb.Hsapiens.UCSC.hg18.knownGene, TxDb.Hsapiens.UCSC.hg19.knownGene, TxDb.Hsapiens.UCSC.hg19.lincRNAsTranscripts, TxDb.Hsapiens.UCSC.hg38.knownGene, TxDb.Mmulatta.UCSC.rheMac3.refGene, TxDb.Mmulatta.UCSC.rheMac8.refGene, TxDb.Mmusculus.UCSC.mm10.ensGene, TxDb.Mmusculus.UCSC.mm10.knownGene, TxDb.Mmusculus.UCSC.mm9.knownGene, TxDb.Ptroglodytes.UCSC.panTro4.refGene, TxDb.Rnorvegicus.BioMart.igis, TxDb.Rnorvegicus.UCSC.rn4.ensGene, TxDb.Rnorvegicus.UCSC.rn5.refGene, TxDb.Rnorvegicus.UCSC.rn6.refGene, TxDb.Scerevisiae.UCSC.sacCer2.sgdGene, TxDb.Scerevisiae.UCSC.sacCer3.sgdGene, TxDb.Sscrofa.UCSC.susScr3.refGene




```
source("https://bioconductor.org/biocLite.R")
biocLite("TxDb.Athaliana.BioMart.plantsmart22")
library(TxDb.Athaliana.BioMart.plantsmart22)
```

GenomicFeatures: Extract informations from TxDB objects



https://bioconductor.org/packages/3.7/bioc/vignettes/GenomicFeatures/inst/doc/GenomicFeatures.pdf

How would you?

```
library(org.Hs.eg.db)
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
```

- Get CDS from BRCA1 (key type is SYMBOL) in human genome hg19?
  - Does it exist in both OrgDB and TxDB objects?
  - Knowing ENTREZID from OrgDB correspond to GENEID in TxDB, how to get CDS list by transcripts?

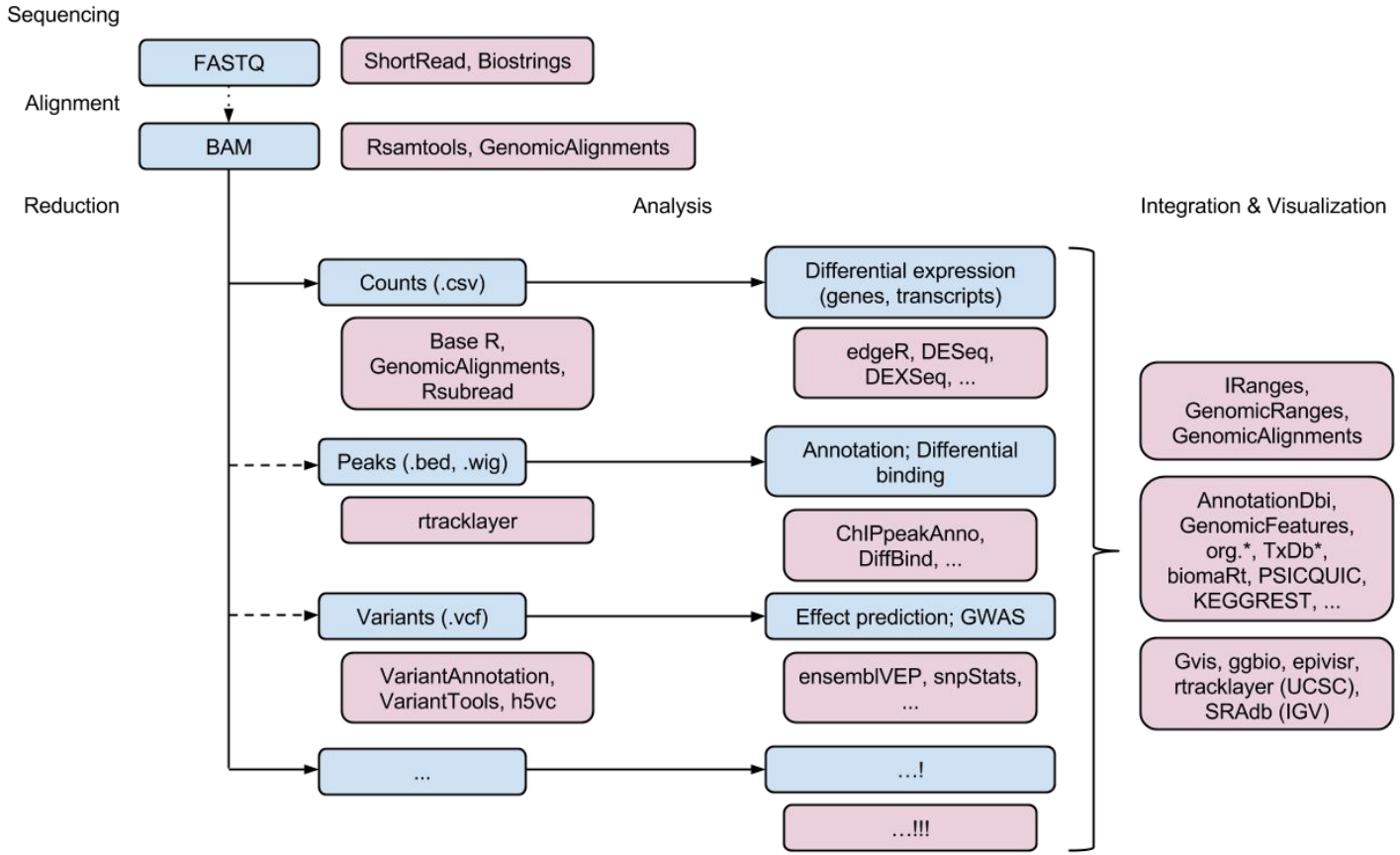
How would you? … Answers

```
library(org.Hs.eg.db)
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
```

- Get CDS from BRCA1 (key type is SYMBOL) in human genome hg19?
  - Does it exist in both OrgDB and TxDB objects?
  - Knowing ENTREZID from OrgDB correspond to GENEID in TxDB, how to get CDS list by transcripts?

```
# The SYMBOLs are not present in the TxDB, from OrgDB find the corresponding ENTREZID. The key is the gene symbol, the required information in the ENTREZID.
eid=select(org.Hs.eg.db, "BRCA1", "ENTREZID", "SYMBOL")[["ENTREZID"]]
# Now you can get the transcript names (TXNAME) from the GENEID in TxDB
txid <- select(x=TxDb.Hsapiens.UCSC.hg19.knownGene, eid, "TXNAME","GENEID")$TXNAME
# Extraction of CDS by transcripts specifying it transcript names
cds <- cdsBy(TxDb.Hsapiens.UCSC.hg19.knownGene, by="tx", use.names=TRUE)
# select in the cds object (GRanges) the cds relative to the BRCA1 transcripts
brca1cds <- cds[names(cds) %in% txid]
```

Specific tools



R plots

Questions about data analysis

Descriptive

Summary of the data

data type, sample number, MAF, mean gene expression

Exploratory

Behaviour of the data
Not to be generalized / Not predictive
Unsupervised and supervised

discover new connections
determine further analyses

Predictive

Use measures to predict other measure values
Not necessary causal

Amazon recommandations
Transcriptional signature

Inferential

Generalise to population from representative subset
Statistics and confidence

survey, poll

Causal

Understand impact of a variable on an other
Need of randomized and prospective experiments

Clinical test
phenotypic consequence of environmental factors

Questions about data analysis

Descriptive

Summary of the data

data type, sample number, MAF, mean gene expression

Exploratory

Behaviour of the data
Not to be generalized / Not predictive
Unsupervised and supervised

discover new connections
determine further analyses

Predictive

Use measures to predict other measure values
Not necessary causal

Amazon recommandations
Transcriptional signature

Inferential

Generalise to population from representative subset
Statistics and confidence

survey, poll

Causal

Understand impact of a variable on an other
Need of randomized and prospective experiments

Clinical test
phenotypic consequence of environmental factors

Why R plots?

- Customized to infinity and beyond
- Generate high resolution graphs (publication, printing ...)
- Completely reproducible and automatizable
- In base R functions & dedicated packages (ggplot2)

Some useful links

CRAN Task View: Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization

Maintainer: Nicholas Lewin-Koh
Contact: nikko at hailmail.net
Version: 2015-01-07
URL: https://CRAN.R-project.org/view=Graphics

R is rich with facilities for creating and developing interesting graphics. Base R contains functionality for many plot types including
as device drivers for all platforms running R. lattice and grid are supplied with R's recommended packages and are included in every
graphics environment than the base R graphics.

R's base graphics are implemented in the same way as in the S3 system developed by Becker, Chambers, and Wilks. There is a static
global parameters such as margins and layouts which can be manipulated by the user using `par()` commands. The R graphics engine
redrawing a whole plot. This situation may change in R 2.7.x, where developers are working on double buffering for R devices. Even

One can quickly run into trouble with R's base graphic system if one wants to design complex layouts where scaling is maintained pr
limitations and as a result packages like lattice, ggplot2, vcd or hexbin use grid for the underlying primitives. When using plots desig
grid commands, e.g., `grid.polygon()` rather than `polygon()`. Also grid maintains a stack of viewports from the device and one neec
vignettes.

The graphics packages in R can be organized roughly into the following topics, which range from the more user oriented at the top to

- *Plotting* : Enhancements for specialized plots can be found in plotrix, for polar plotting, vcd for categorical data, hexbin for he
  implemented in aplpack, which also has a nice implementation of Tukey's bag plot. For 3D plots lattice, scatterplot3d and misc
  on rgl. The package onion for visualizing quaternions and octonions is well suited to display 3D graphics based on derived me
- *Graphic Applications* : This area is not much different from the plotting section except that these packages have tools that may
  subject areas, like maps, or clustering the excellent task views contributed by other dedicated useRs is an excellent place to sta
  - *Effect ordering* : The gclus package focuses on the ordering of graphs to accentuate cluster structure or natural ordering
    criteria. For ordering an array of displays, biclust can be useful.
  - *Large Data Sets* : Large data sets can present very different challenges from moderate and small datasets. Aside from ov
    very quickly, and hexbin can bin bivariate data onto a hexagonal lattice, the advantage being that the irregular lines and
    with lattice. An alternative is to use scagnostics to produce a scatterplot matrix of "data about the data", and look for inte
  - *Trees and Graphs* : ape and ade4 have functions for plotting phylogenetic trees, which can be used for plotting dendrogr
    placement, so may not be useful for very large trees. igraph has the Tilford-Rheingold algorithm implemented and is use
    igraph have functions for plotting and layout, especially useful for representing large networks.
- *Graphics Systems* : lattice is built on top of the grid graphics system and is an R implementation of William Cleveland's trellis
  implementation of the system described in "A Grammar of Graphics" by Leland Wilkinson. Like lattice, ggplot2 (also built on
  more emphasis on reshaping data, transformation, and assembling the elements of a plot.
- *Devices* : Whereas grid is built on top of the R graphics engine, many in the R community have found the R graphics engine sc
  base supplies devices for PostScript, PDF, JPEG and other formats. Devices on CRAN include cairoDevice which is a device b
  the Gimp Tool Kit, similar to pyGTK2. RSvgDevice is an SVG device driver and interfaces well with with vector drawing pro
  standard. Trust Microsoft. rgl provides a device driver based on OpenGL, and is good for 3D and interactive development. Las
- *Colors* : The package colorspace provides a set of functions for transforming between color spaces and `mixcolor()` for mixing
  suitable for coding categorical variables (`rainbow_hcl()`) and numerical information (`sequential_hcl()`,`diverge_hcl()`).
- *Interactive Graphics* : There are several efforts to implement interactive graphics systems that interface well with R. In an inte
  the device as well as link with other views of the data. rggobi embeds the GGobi interactive graphics system within R, so that
  GGobi's edge set functionality. The RoSuDA repository maintained and developed by the University of Augsburg group has tw
  graphics tools contain functions for alpha blending, which produces darker shading around areas with more data. This is excep
  versions of R graphics using the cairoDevice and RGtk2. Lastly, the rgl package has mechanisms for interactive manipulation
- *Development* : For development of specialized graphics packages in R, grid should probably be the first consideration for any
  the Java device in the RoSuDA packages, though Java has its own drawbacks. For porting plotting code to grid, using the pack

CRAN packages:

- ade4
- animation
- ape
- aplpack

https://cran.r-project.org/web/views/Graphics.html

R graph gallery

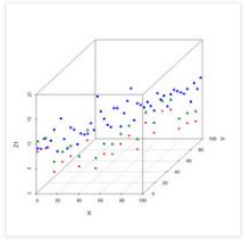
The blog is a collection of script examples with example data and output plots. R produce excellent quality graphs for data analysis, science and business presentation, publications and other purposes. Self-help codes and examples are provided. Enjoy nice graphs !!

3 variable plots (5) 3D plots (8) association plot (4) barchart (13) boxplot (10) bubble plot (5) categorical data (6) circular (1) cluster (4) combination plot (10) density (13) distribution (9) heatmap (20) histogram (11) map (13) pie (1) polygon (1) scatterplot (25) trellis plot (8) xy line (10) xy points (25)

Wednesday, May 22, 2013

RG #110: 3D scatter plot with multiple series in Y axis

```
X = seq(1, 100, 5)
Y = seq (1, 100, 5)
Z = rnorm (length (X), 10, 2)
data1 <- data.frame (X, Y, )
data2 <- data.frame (X, Y, Z1 = Z - 5)
data3 <- data.frame (X, Y, Z1 = Z - 3)

require(scatterplot3d)
s3d <- scatterplot3d(data1, color = "blue", pch = 19, xlim=NULL,
ylim=NULL, zlim= c(0, 20))
s3d$points3d(data2, col = "red", pch = 18)
s3d$points3d(data3, col = "green4", pch = 17)
```



Translate

Sélectionner une langue

Powered by Google Translation

Search This Blog

Search

Popular Posts (Last 30 days)

RG#35: Stacked bar chart (number and percent)

RG#80: Plotting boxplot and histogram (overlayed or in margin)

RG#96: Basic point and line graph with error bars (duplication purpose)

RG#81: plotting scatter plot with means and samples (means are connected with line while all samples as scatter plot)

RG#99: cloud 3D bars with heatmap

Popular Posts (All time)

RG#35: Stacked bar chart (number and percent)

http://rgraphgallery.blogspot.fr/

Fiche TD avec le logiciel R : tdr75

Les paramètres graphiques

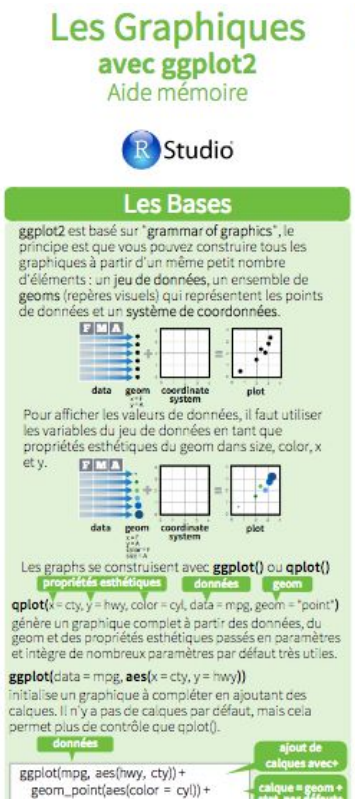
J.R. Lobry, A.B. Dufour & D. Chessel

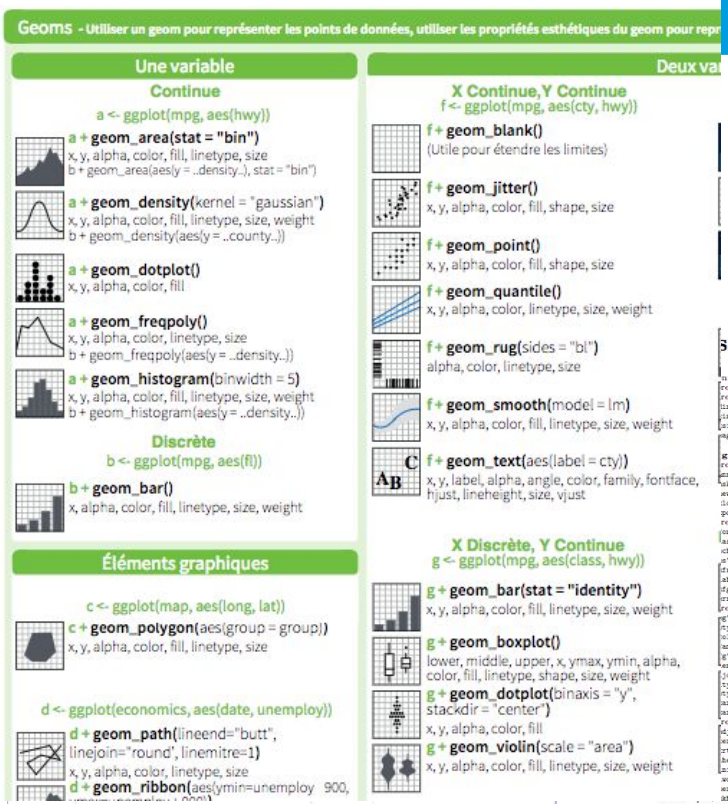
Table des matières

http://pbil.univ-lyon1.fr/R/pdf/tdr75.pdf

Some us

https://www.rstudio.com/wp-content/uploads/2015/06/ggplot2-french.pdf





CRAN Task View: Graphic Displays & Dyna

Maintainer: Nicholas Lewin-Koh
Contact: nikko at hailmail.net
Version: 2015-01-07
URL: https://CRAN.R-project.org/view=Graphics

R is rich with facilities for creating and developing interes
as device drivers for all platforms running R. lattice and g
graphics environment than the base R graphics.

R's base graphics are implemented in the same way as in t
global parameters such as margins and layouts which can
redrawing a whole plot. This situation may change in R 2.

One can quickly run into trouble with R's base graphic sys
limitations and as a result packages like lattice, ggplot2,
grid commands, e.g., `grid.polygon()` rather than `polygo`
vignettes.

The graphics packages in R can be organized roughly into

- *Plotting* : Enhancements for specialized plots can be
  implemented in aplpack, which also has a nice impl
  on rgl. The package onion for visualizing quaternio
- *Graphic Applications* : This area is not much differe
  subject areas, like maps, or clustering the excellent
  - *Effect ordering* : The gclus package focuses o
    criteria. For ordering an array of displays, bic
  - *Large Data Sets* : Large data sets can present
    very quickly, and hexbin can bin bivariate dat
    with lattice. An alternative is to use scagnosti
  - *Trees and Graphs* : ape and ade4 have functic
    placement, so may not be useful for very larg
    igraph have functions for plotting and layout,
- *Graphics Systems* : lattice is built on top of the grid
  implementation of the system described in "A Gram
  more emphasis on reshaping data, transformation, a
- *Devices* : Whereas grid is built on top of the R grap
  base supplies devices for PostScript, PDF, JPEG an
  the Gimp Tool Kit, similar to pyGTK2. RSvgDevic
  standard. Trust Microsoft. rgl provides a device driv
- *Colors* : The package colorspace provides a set of f
  suitable for coding categorical variables (`rainbow_`
- *Interactive Graphics* : There are several efforts to in
  the device as well as link with other views of the da
  GGobi's edge set functionality. The RoSuDA reposi
  graphics tools contain functions for alpha blending,
  versions of R graphics using the cairoDevice and R
- *Development* : For development of specialized grap
  the Java device in the RoSuDA packages, though Ja

CRAN packages:

- ade4
- animation
- ape
- aplpack

https://cran.r-project.org/web/views/Graphics.html

http://rgraphgallery.blogspot.fr/

Fiche TD avec le logiciel R : tdr75

Les paramètres graphiques

J.R. Lobry, A.B. Dufour & D. Chessel

s matières

http://pbil.univ-lyon1.fr/R/pdf/tdr75.pdf

GGPlot

http://ggplot2.org/

Statistiques

length	width	depth	trt
2	3	4	a
1	2	1	a
4	5	15	b
9	10	80	b

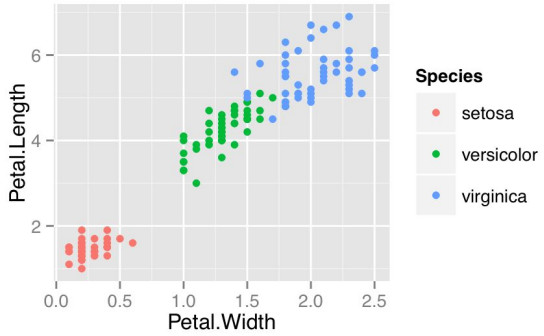
Mapping aesthetetics

x	y	colour
2	3	a
1	2	a
4	5	b
9	10	b

scales

x	y	colour
25	11	red
0	0	red
75	53	blue
200	300	blue

geom



Frequently asked questions

Is there any correlation between these variables, samples ?

XY plots

heatmaps

How is the distribution of the data?

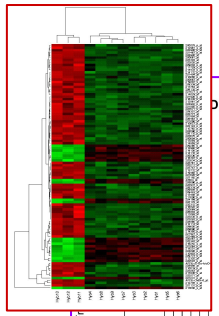
Histograms

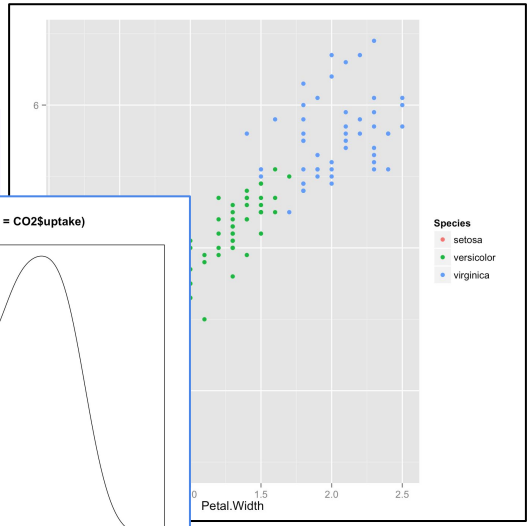
Densities

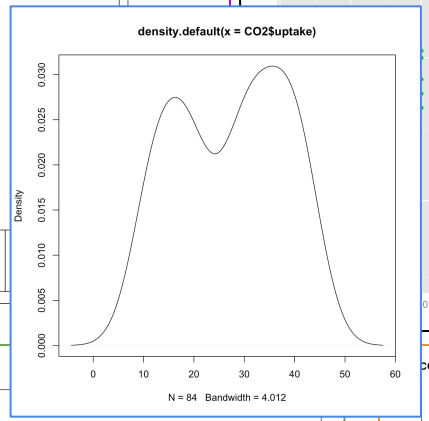
Is there a difference of value distribution between groups ?

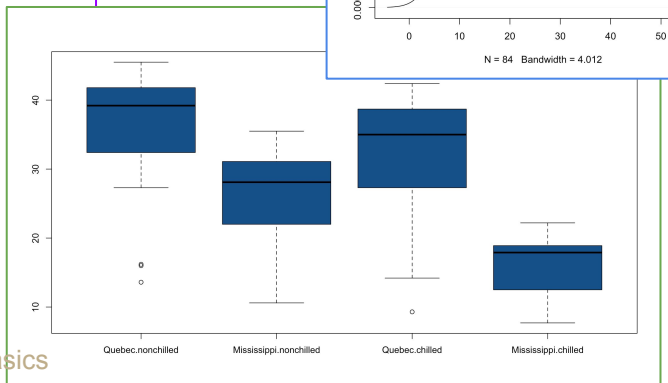
Boxplots

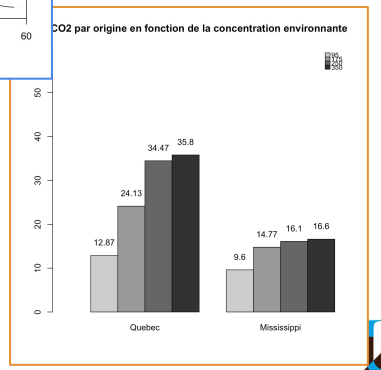
Barplots











Example data

```
library(ggplot2)
data(CO2)
```

How would you?

```
library(ggplot2)
data(CO2)
```

- Visualise if there is a correlation between CO_2 plant uptake and atmospheric CO_2 concentration
    - Which informations are contained in CO2?
    - Which kind of plot?

How would you?

```
library(ggplot2)
data(CO2)
```

- Visualise if there is a correlation between CO_2 plant uptake and atmospheric CO_2 concentration
    - Which informations are contained in CO2?
    - Which kind of plot?

```
head(CO2)
summary(CO2)
str(CO2)
```

How would you?

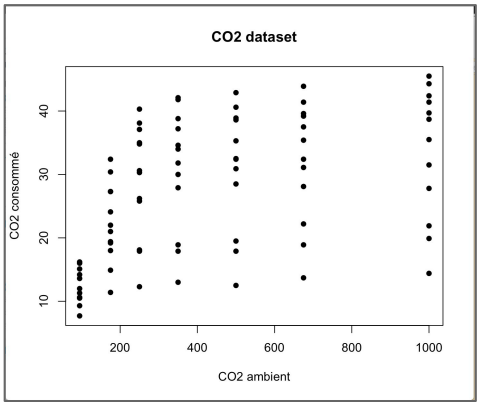
```
library(ggplot2)
data(CO2)
```

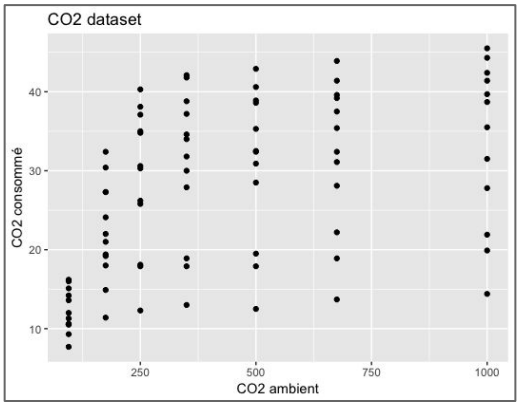
- Visualise if there is a correlation between CO_2 plant uptake and atmospheric CO_2 concentration
  - Which informations are con
  - Which kind of plot?

```
plot(CO2$conc,CO2$uptake,
     pch=16,
     xlab="CO2 ambient",
     ylab="CO2 consommé",
     main="CO2 dataset")
```

```
ggplot(CO2, aes(conc,uptake)) +
     geom_point() +
     xlab("CO2 ambient") +
     ylab("CO2 consommé") +
     ggtitle("CO2 dataset")
```

```
head(CO2)
summary(CO2)
str(CO2)
```





How would you?

```
library(ggplot2)
data(CO2)
```

- Visualise if there is a correlation between CO_2 plant uptake and atmospheric CO_2 concentration
    - Which informations are contained in CO2?
    - Which kind of plot?
- Is there a difference between the two locations?
    - Add color by location

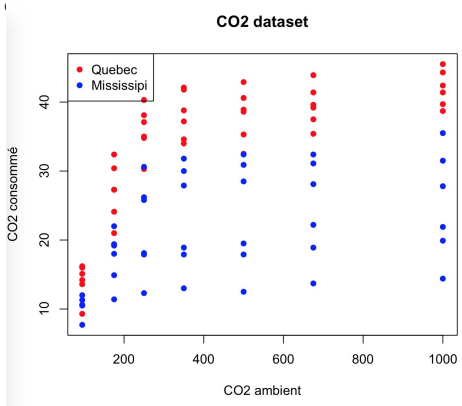
How would you?

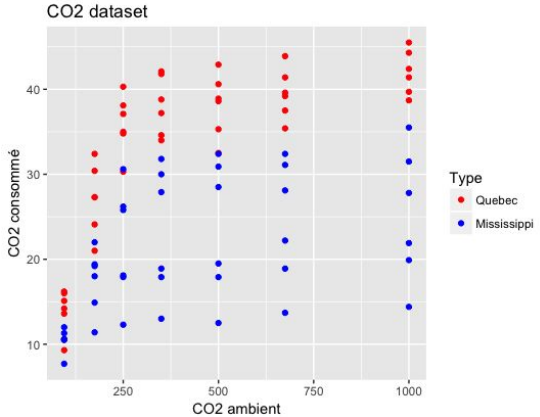
```
library(ggplot2)
data(CO2)
```

```
point_color <- ifelse(CO2$Type=="Quebec","red","blue")
plot(CO2$conc,CO2$uptake,
        pch=16,
        col= point_color,
        xlab="CO2 ambient",
        ylab="CO2 consommé",
        main="CO2 dataset")
legend("topleft", legend=c("Quebec","Mississipi"),
        col=c("red","blue"),pch=16)
```

ptake

```
ggplot(CO2, aes(conc,uptake)) +
        geom_point(aes(colour=Type)) +
        xlab("CO2 ambient") +
        ylab("CO2 consommé") +
        ggtitle("CO2 dataset") +
        scale_color_manual(values=c("red","blue"))
```





How would you?

```
library(ggplot2)
data(CO2)
```

- Visualise if there is a correlation between CO_2 plant uptake and atmospheric CO_2 concentration
    - Which informations are contained in CO2?
    - Which kind of plot?
- Is there a difference between the two locations?
    - Add color by location
    - Add the names of the plant

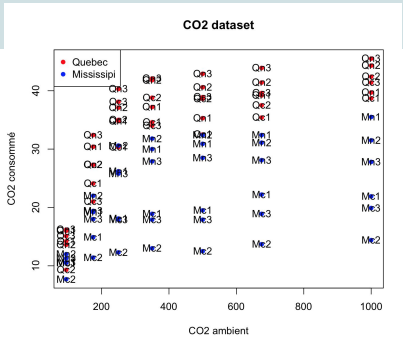
How would you?

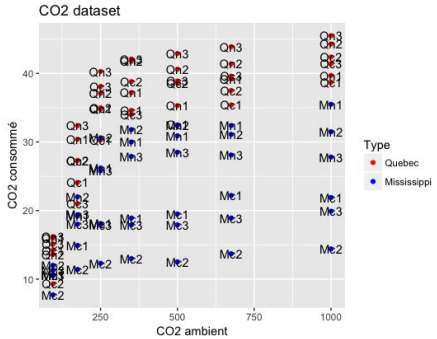
```
library(ggplot2)
data(CO2)
```

```
point_color <- ifelse(CO2$Type=="Quebec","red","blue")
plot(CO2$conc,CO2$uptake,
        pch=16,
        col= point_color,
        xlab="CO2 ambient",
        ylab="CO2 consommé",
        main="CO2 dataset")
legend("topleft", legend=c("Quebec","Mississipi"),
        col=c("red","blue"),pch=16)
text(CO2$conc,CO2$uptake, labels=CO2$Plant)
```

ptake

```
ggplot(CO2, aes(conc,uptake)) +
        geom_point(aes(colour=Type)) +
        xlab("CO2 ambient") +
        ylab("CO2 consommé") +
        ggtitle("CO2 dataset") +
        scale_color_manual(values=c("red","blue")) +
        geom_text(label=Plant)
```





How would you?

```
library(ggplot2)
data(CO2)
```

- Visualise if there is a correlation between CO_2 plant uptake and atmospheric CO_2 concentration
    - Which informations are contained in CO2?
    - Which kind of plot?
- Is there a difference between the two locations?
    - Add color by location
    - Add the names of the plant
    - Plot a linear regression curve per location

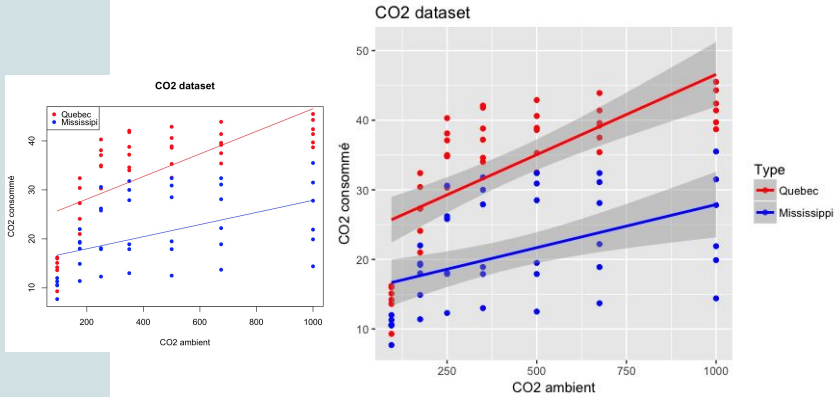
How would you?

```
data(iris)
data(CO2)
```

```
point_color <- ifelse(CO2$Type=="Quebec","red","blue")
plot(CO2$conc,CO2$uptake,
        pch=16,
        col= point_color,
        xlab="CO2 ambient",
        ylab="CO2 consommé",
        main="CO2 dataset")
legend("topleft", legend=c("Quebec","Mississipi"),
        col=c("red","blue"),pch=16)

CO2.miss <- CO2[CO2$Type=="Mississippi",]
lm.miss <- lm(CO2.miss$uptake~CO2.miss$conc)
CO2.miss$fitted <- lm.miss$fitted.values

CO2.quebec <- CO2[CO2$Type=="Quebec",]
lm.quebec <- lm(CO2.quebec$uptake~CO2.quebec$conc)
CO2.quebec$fitted <- lm.quebec$fitted.values

lines(CO2.miss$conc,CO2.miss$fitted,col="blue")
lines(CO2.quebec$conc,CO2.quebec$fitted,col="red")
```

ptake

```
ggplot(CO2, aes(conc,uptake)) +
        geom_point(aes(colour=Type)) +
        xlab("CO2 ambient") +
        ylab("CO2 consommé") +
        ggtitle("CO2 dataset") +
        scale_color_manual(values=c("red","blue")) +
        geom_smooth(method="lm",aes(colour=Type))
```



How would you?

```
library(ggplot2)
data(CO2)
```

- Visualise if there is a correlation between CO_2 plant uptake and atmospheric CO_2 concentration
    - Which informations are contained in CO2?
    - Which kind of plot?
- Is there a difference between the two locations?
    - Add color by location
    - Add the names of the plant
    - Plot a linear regression curve per location
- Is there a difference between plants?

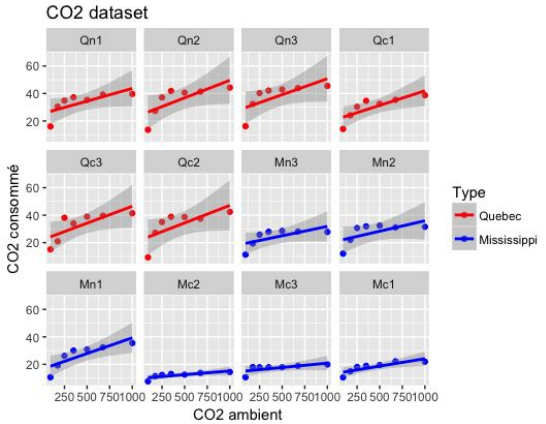
How would you?

```
library(ggplot2)
data(CO2)
```

```
point_color <- ifelse(CO2$Type=="Quebec","red","blue")
plot(CO2$conc,CO2$uptake,
        pch=16,
        col= point_color,
        xlab="CO2 ambient",
        ylab="CO2 consommé",
        main="CO2 dataset")
legend("topleft", legend=c("Quebec","Mississipi"),
        col=c("red","blue"),pch=16)

CO2.miss <- CO2[CO2$Type=="Mississippi",]
lm.miss <- lm(CO2.miss$uptake~CO2.miss$conc)
CO2.miss$fitted <- lm.miss$fitted.values

CO2.quebec <- CO2[CO2$Type=="Quebec",]
lm.quebec <- lm(CO2.quebec$uptake~CO2.quebec$conc)
CO2.quebec$fitted <- lm.quebec$fitted.values

lines(CO2.miss$conc,CO2.miss$fitted,col="blue")
lines(CO2.quebec$conc,CO2.quebec$fitted,col="red")
```

ptake

```
ggplot(CO2, aes(conc,uptake)) +
        geom_point(aes(colour=Type)) +
        xlab("CO2 ambient") +
        ylab("CO2 consommé") +
        ggtitle("CO2 dataset") +
        scale_color_manual(values=c("red","blue")) +
        geom_smooth(method="lm",aes(colour=Type)) +
        facet_wrap(~Plant)
```



How would you?

```
library(ggplot2)
data(CO2)
```

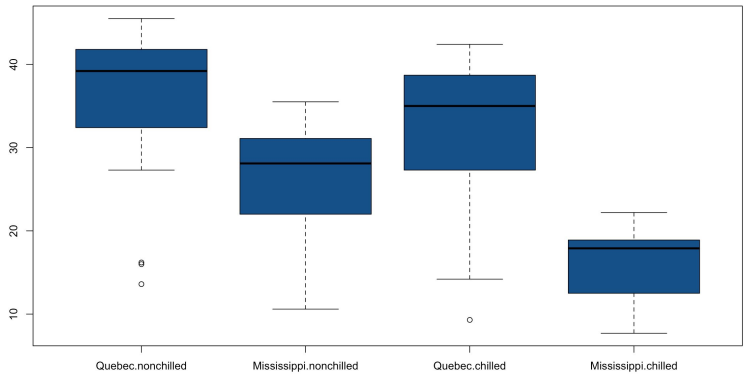
- Visualise if there is a difference in the plant CO_2 uptake according to the location and the treatment

How would you?

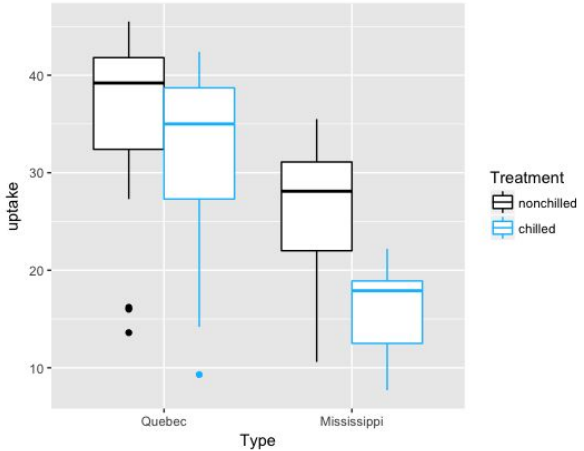
```
library(ggplot2)
data(CO2)
```

- Visualise if there is a difference in the plant CO_2 uptake according to the location and the treatment

```
boxplot(CO2$uptake~CO2$Type*CO2$Treatment,
        col="dodgerblue4")
```




```
ggplot(CO2, aes(x=Type,y=uptake)) +
        geom_boxplot(aes(colour=Treatment))+
        scale_color_manual(values=c("black",colours()[121]))
```



How would you?

```
library(ggplot2)
data(CO2)
```

- Visualise if there is a difference in the plant CO_2 uptake according to the location and the treatment
- Check if the distribution on the plant C02 uptake looks normal

How would you?

```
library(ggplot2)
data(CO2)
```

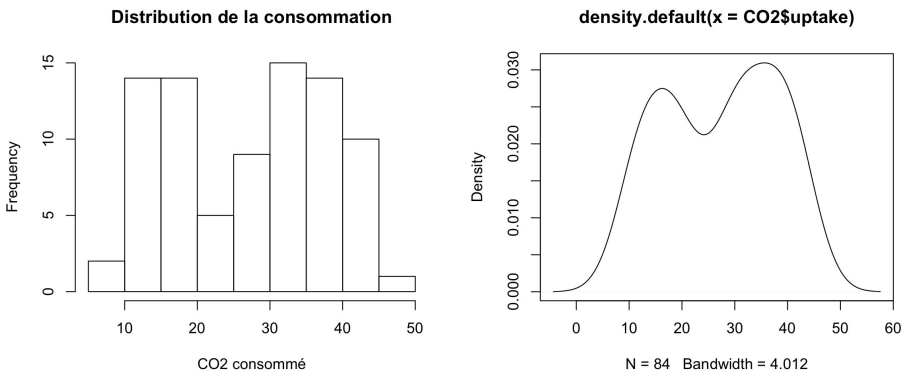
- Visualise if there is a difference in the plant CO_2 uptake according to the

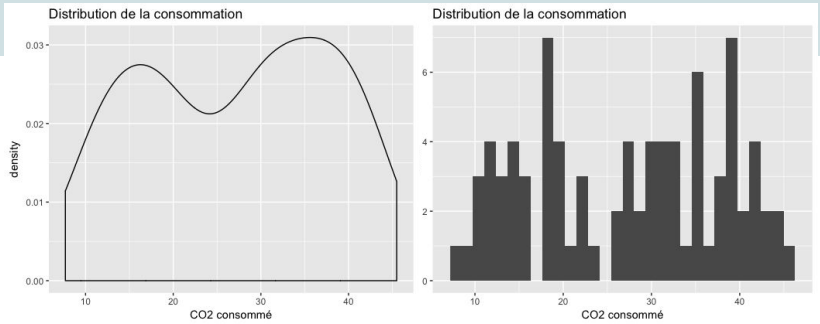
```
par(mfrow=c(1,2))
hist(CO2$uptake,
xlab="CO2 consommé",
main="Distribution de la consommation")
plot(density(CO2$uptake))
par(mfrow=c(1,1))
```

e plant C02 uptake looks

```
ggplot(CO2, aes(x=uptake)) +
        geom_histogram() +
        ggtitle("Distribution de la consommation") +
        xlab("CO2 consommé")
```

```
ggplot(CO2, aes(x=uptake)) +
        geom_density() +
        ggtitle("Distribution de la consommation") +
        xlab("CO2 consommé")
```



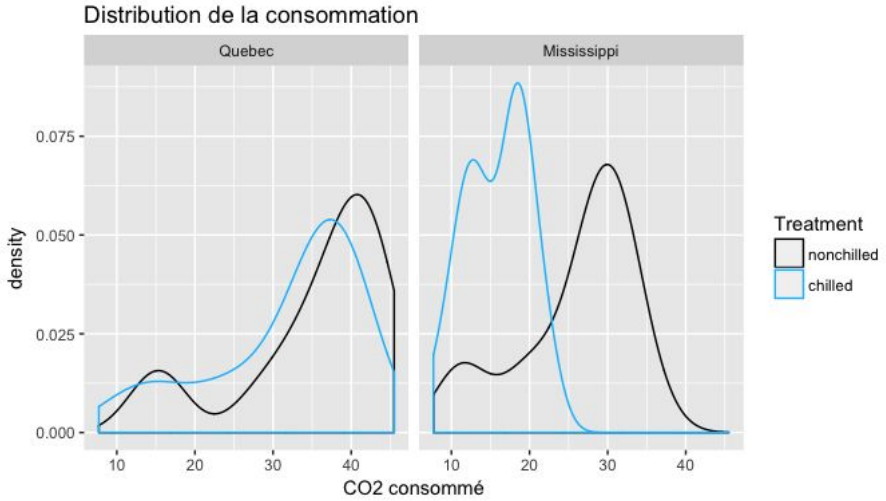


How would you?

```
library(ggplot2)
data(CO2)
```

- Visualise if there is a difference in the plant CO_2 uptake according to the location and the treatment
- Check if the distribution on the plant C02 uptake looks normal
- Is this distribution due to some variables?

```
ggplot(CO2, aes(x=uptake)) +
        geom_density(aes(colour=Treatment)) +
        scale_color_manual(values=c("black",colours()[121])) +
        ggtitle("Distribution de la consommation") +
        xlab("CO2 consommé") +
        facet_wrap(~Type)
```



GGBio

- ggplot quickly adopted by analyst
- ggbio uses ggplot to offer genomic specificities
- Adapted to read NGS data (BAM/GFF/VCF) in object as GRanges, GAlignments etc.
- The generic function autoplot() works often very well
- Combine several tracks of the same regions (reads, coverage, transcripts) using the tracks() function from rtracklayer package

http://www.tengfei.name/ggbio/docs/man/autoplot-method.html