Reproducible Research in the era of Next Generation Sequencing:

current approaches, examples and future perspectives

Claudia Angelini, Dario Righelli, Francesco Russo

Istituto per le Applicazioni del Calcolo–CNR

Laboratory of Statistics and Computational Tools for Bioinformatics,

Napoli, Italy

http://bioinfo.na.iac.cnr.it/BioinfoLab/

SeqAHEAD

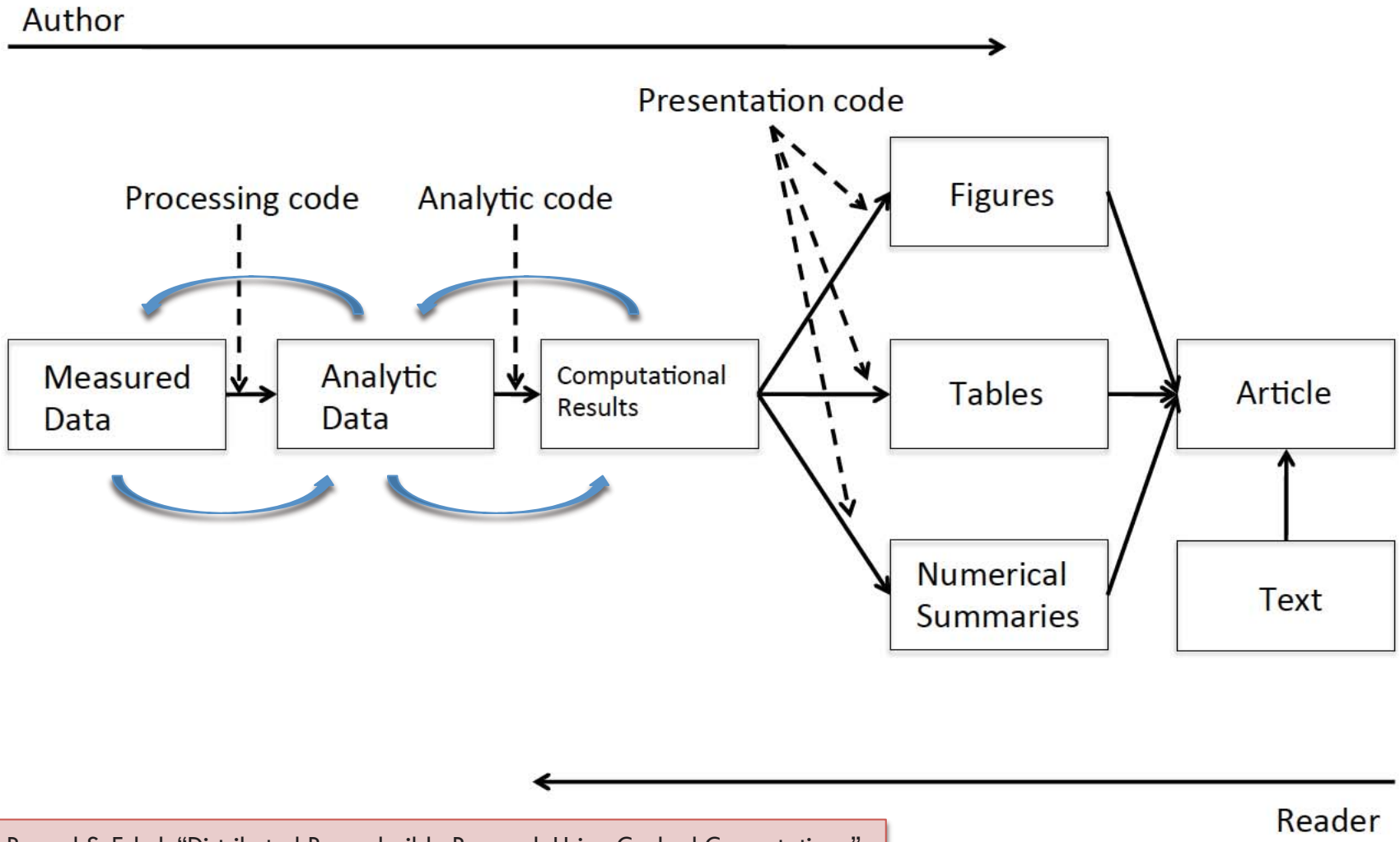
Next Generation Sequencing: a look into the future,

16-17 March 2015, Bratislava, Slovakia

Motivations

- The possibility to **replicate** scientific findings using independent investigators, methods, data, equipment and protocols is the standard approach by which scientific claims are evaluated.
- In many fields, including bio-medicine and genomics, some studies cannot be fully replicated because of a lack of time or resources (and also since journals often do not publish replicated studies).
- In such cases, it is important to be able **to inspect** and **reproduce** the entire analysis carried out in a given paper.
- Unfortunately, the description of the **data analysis is often lacking of important** (technical) **details**.
- Moreover, the analyses of **NGS "multi-omic"** data are also very **complex**.
- Therefore, **it becomes often very hard to reproduce the results**.

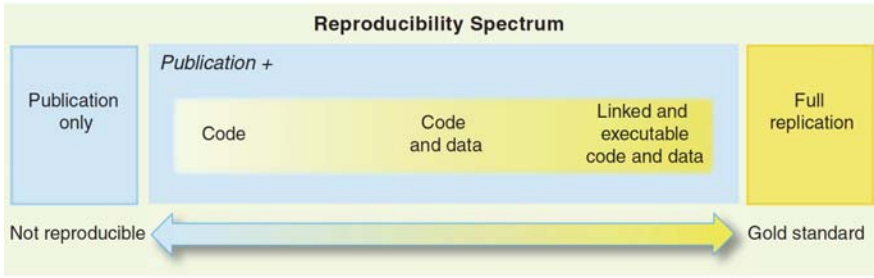
Research Pipeline



Peng, R., and S. Eckel, "Distributed Reproducible Research Using Cached Computations," *Computing in Science & Engineering*, pp. 28-34, (2009).

Reproducible Research

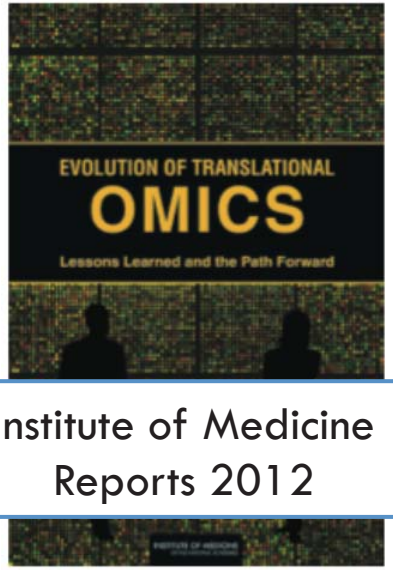
- The idea of **RR** is to make ***analytic data*** and ***code*** (and its **documentation**) available so that others may reproduce the findings.
- From a computational point of view **RR** is similar to regard **data analysis** as an "**experimental protocol**".



Science AAAS

Reproducible Research in Computational Science
Roger D. Peng
Science **334**, 1226 (2011);
DOI: 10.1126/science.1213847

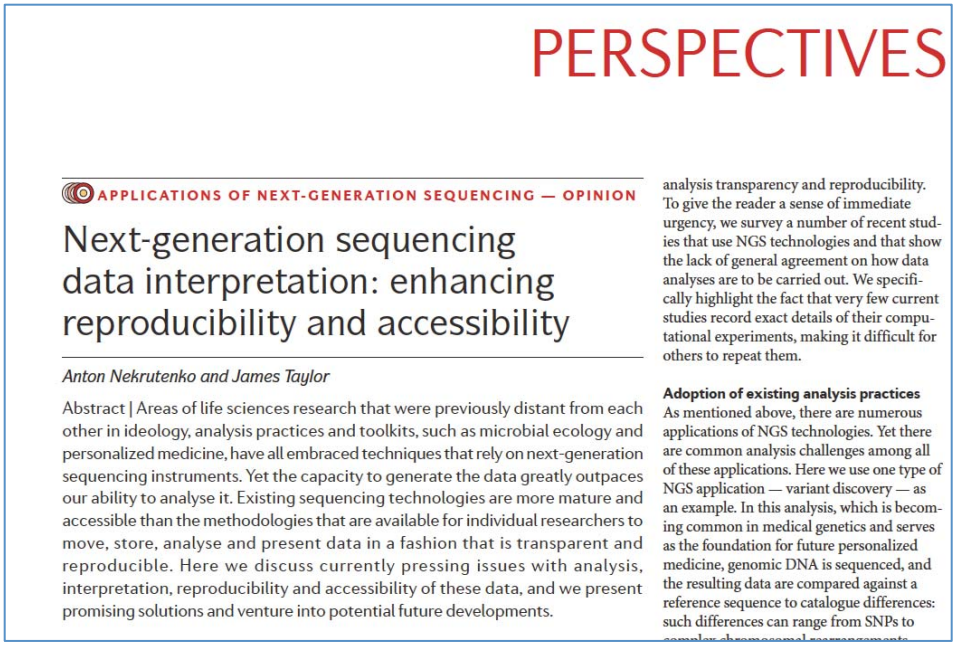
Fig. 1. The spectrum of reproducibility.



Institute of Medicine Reports 2012

- **Data/metadata** used to develop test should be made publicly available
- The **computer code** and fully specified computational procedures used for development of the omics- data analysis should be made sustainably available
- **All aspects of the analysis need to be transparently reported.**

Reproducible Research and NGS



PERSPECTIVES

APPLICATIONS OF NEXT-GENERATION SEQUENCING — OPINION

Next-generation sequencing data interpretation: enhancing reproducibility and accessibility

Anton Nekrutenko and James Taylor

Abstract | Areas of life sciences research that were previously distant from each other in ideology, analysis practices and toolkits, such as microbial ecology and personalized medicine, have all embraced techniques that rely on next-generation sequencing instruments. Yet the capacity to generate the data greatly outpaces our ability to analyse it. Existing sequencing technologies are more mature and accessible than the methodologies that are available for individual researchers to move, store, analyse and present data in a fashion that is transparent and reproducible. Here we discuss currently pressing issues with analysis, interpretation, reproducibility and accessibility of these data, and we present promising solutions and venture into potential future developments.

analysis transparency and reproducibility. To give the reader a sense of immediate urgency, we survey a number of recent studies that use NGS technologies and that show the lack of general agreement on how data analyses are to be carried out. We specifically highlight the fact that very few current studies record exact details of their computational experiments, making it difficult for others to repeat them.

Adoption of existing analysis practices

As mentioned above, there are numerous applications of NGS technologies. Yet there are common analysis challenges among all of these applications. Here we use one type of NGS application — variant discovery — as an example. In this analysis, which is becoming common in medical genetics and serves as the foundation for future personalized medicine, genomic DNA is sequenced, and the resulting data are compared against a reference sequence to catalogue differences: such differences can range from SNPs to

Box 2 | **Barriers to reproducibility are widespread**

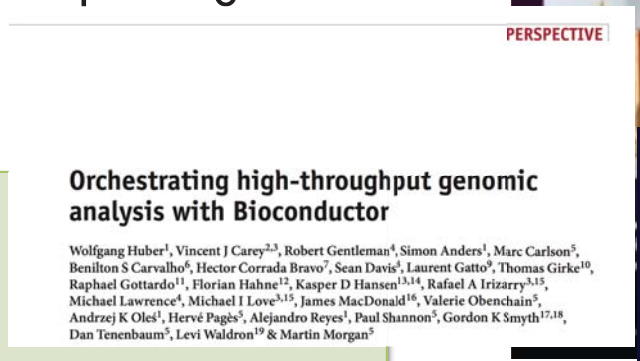
Many classical publications in life sciences have become influential because they provide complete information on how to repeat reported analyses so others can adopt these approaches in their own research, such as for chain termination sequencing technology that was developed by Sanger and colleagues[35] and for PCR[36,37]. Today's publications that include computational analyses are very different. Next-generation sequencing (NGS) technologies are undoubtedly as transformative as DNA sequencing and PCR were more than 30 years ago. As more and more researchers use high-throughput sequencing in their research, they consult other publications for examples of how to carry out computational analyses. Unfortunately, they often find that the extensive informatics component that is required to analyse NGS data makes it much more difficult to repeat studies published today. Note that the lax standards of computational reproducibility are not unique to life sciences; the importance of being able to repeat computational experiments was first brought up in geosciences[38] and became relevant in life sciences following the establishment of microarray technology and high-throughput sequencing[3,39,40]. Replication of computational experiments requires access to input data sets, source code or binaries of exact versions of software used to carry out the initial analysis (this includes all helper scripts that are used to convert formats, groom data, and so on) and knowing all parameter settings exactly as they were used. In our experience (BOX 1 and Supplementary information S1 (table)), publications rarely provide such a level of detail, making biomedical computational analyses almost irreproducible. Supplementary information S2 (reference list) lists 50 papers randomly selected from 378 manuscripts published in 2011 that use the Burrows–Wheeler Aligner[15] for mapping Illumina reads. Most papers (31) provide neither a version nor the parameters used, and neither do they provide the exact version of the genomic reference sequence. From the remaining 19 publications, only four studies provide settings, eight studies list the version, and only seven studies list all necessary details. More than half of the studies (26 out of 50) do not provide access to the primary data sets. In two cases, authors provided links to their own websites, where data were deposited; however, in both cases, links were broken.

- ❑ NGS analyses are quite **complex** and require the use of several tools
- ❑ Tools are often regularly updated, technology changes continuously
- ❑ NGS analyses are **time-consuming** and have to handle "**Big-data**"

Developing computational tools in the spirit of RR

One of the goals of modern bioinformatics should be to develop computational tools that support reproducible research

- Several instruments and have been developed to facilitate RR in different programming languages
- **R** is an open source language, particularly designed for RR. **Bioconductor** contains several hundreds of packages for the analysis of NGS

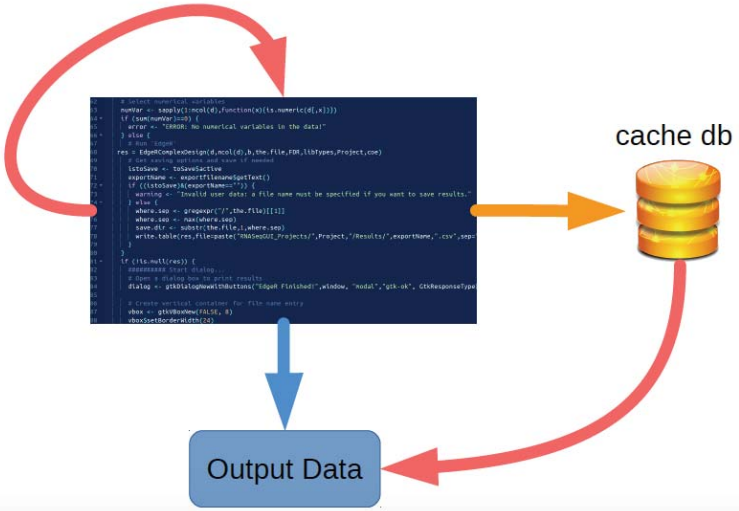


Key ingredients for RR are

- **Literate statistical** programming
- **Caching** (for handling big data)
- **Versioning control**
- Suitable result **reporting tools** and data **repositories**

Big Data Challenges & Chaching

- Cache is a module **to store** data in order **to retrieve** that data faster.
- This helps in RR serving stored data resulting from time consuming chunk of code.
  - In this way it avoids repetition of time consuming computation when they are computed again.
- Additionally permits **to share cached data through the web**.
  - In this way it is possible to reproduce the same computations using the same data or to verify the results of a third part computations.



What about GUI?

- Many tools for omics data analysis have a **graphical user interfaces** (GUIs)
- GUIs are convenient and very intuitive for biologists.
- GUIs are also interactive, so that the user can decide what kind of analysis to perform (point-and click approach) on the basis of the intermediate results
- Tools with web-interface have similar features (and share similar issues)
- **However, GUIs do not facilitate RR, since results are obtained after clicking several buttons → difficult to keep track of all performed steps**

Our Aim: To develop user friendly computational tools for NGS data analysis in the spirit of "*Reproducible Research*", *i.e., we want combine **GUI** with tools for **RR** available in R.*

RNASeqGUI

- RNASeqGUI is implemented in **R**.
- It requires the **RGTK2** graphical Library to run
- It uses **BiocParallel** to speed up the computations.

RNASeqGUI can be downloaded from http://bioinfo.na.iac.cnr.it/RNASeqGUI/

BIOINFORMATICS APPLICATIONS NOTE Vol. 30 no. 17 2014, pages 2514–2516
doi:10.1093/bioinformatics/btu308

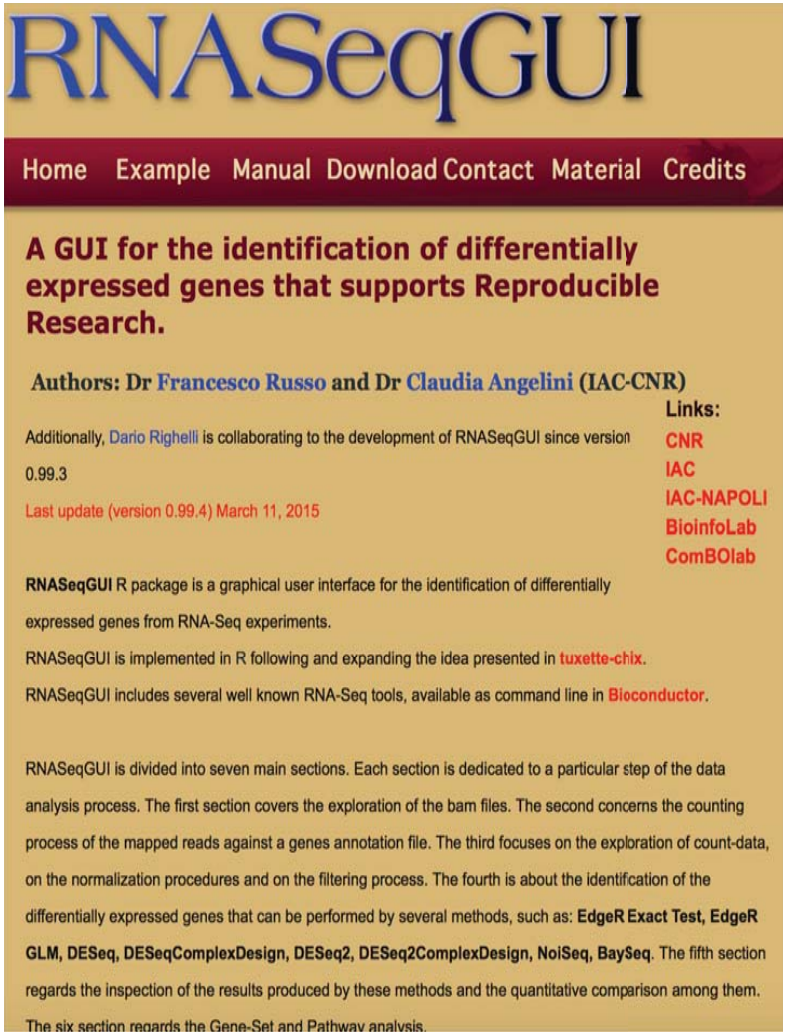
Gene expression

Advance Access publication May 7, 2014

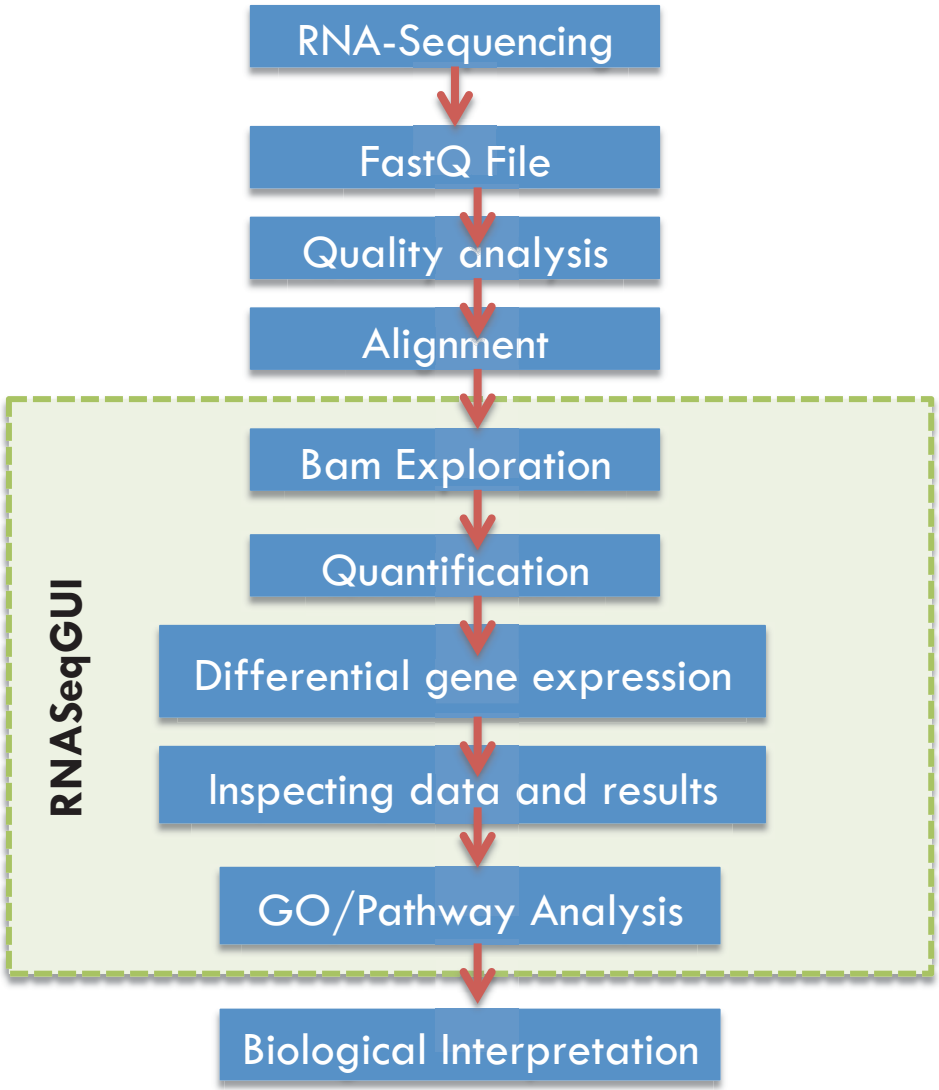
RNASeqGUI: a GUI for analysing RNA-Seq data

Francesco Russo* and Claudia Angelini
Istituto per le Applicazoni del Calcolo, CNR, 80131, Napoli, Italy

Associate Editor: Ivo Hofacker



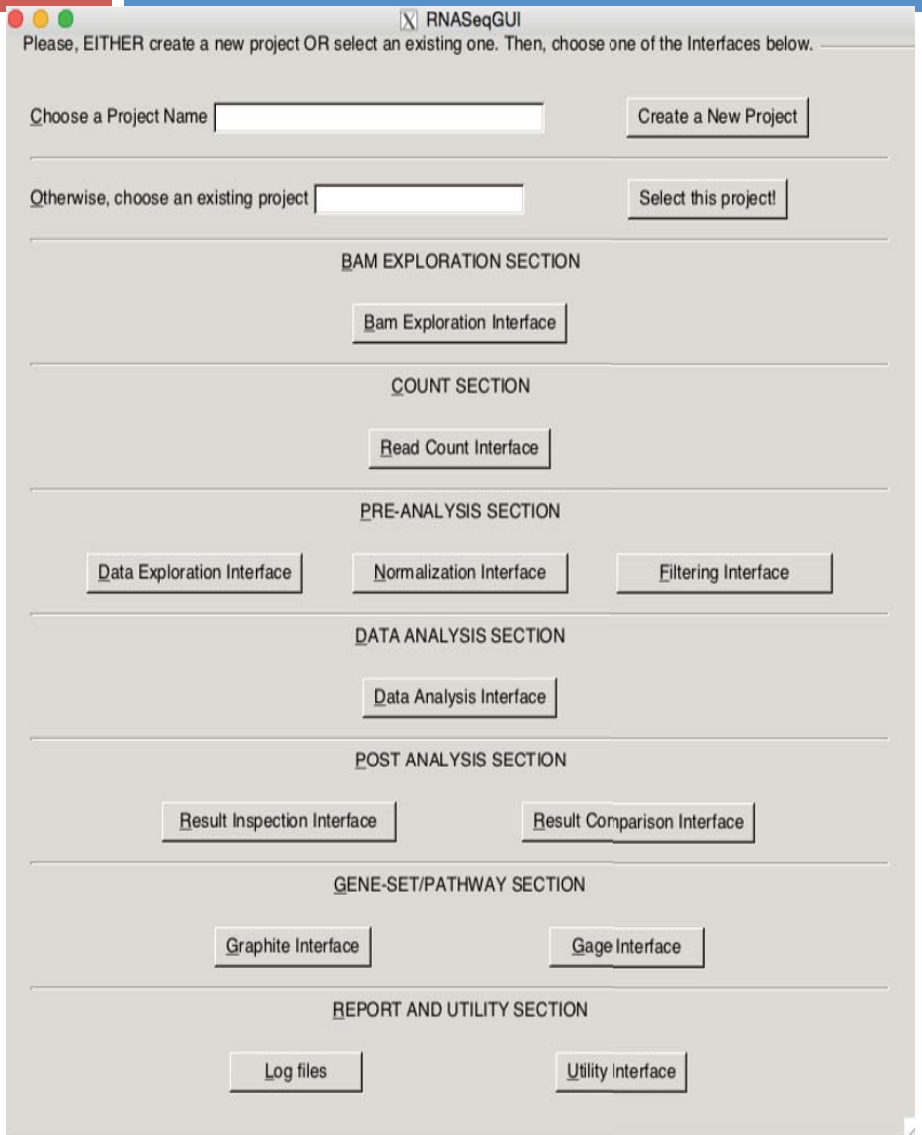
RNASeqGUI workflow



Recent update includes

- ❑ It handles technical and biological replicates
- ❑ Complex experimental designs in differential gene expression section
- ❑ Filtering and Conversion
- ❑ Pathway analysis using Gage and Graphite
- ❑ Fancy reporting using **Reportingtools**
- ❑ Advanced RR using **R markdown** and **knitr**
- ❑ Caching using **filehash**

RNASeqGUI Main Interface



The GUI is divided into several sections. Each section is dedicated to a particular step of the data analysis process.

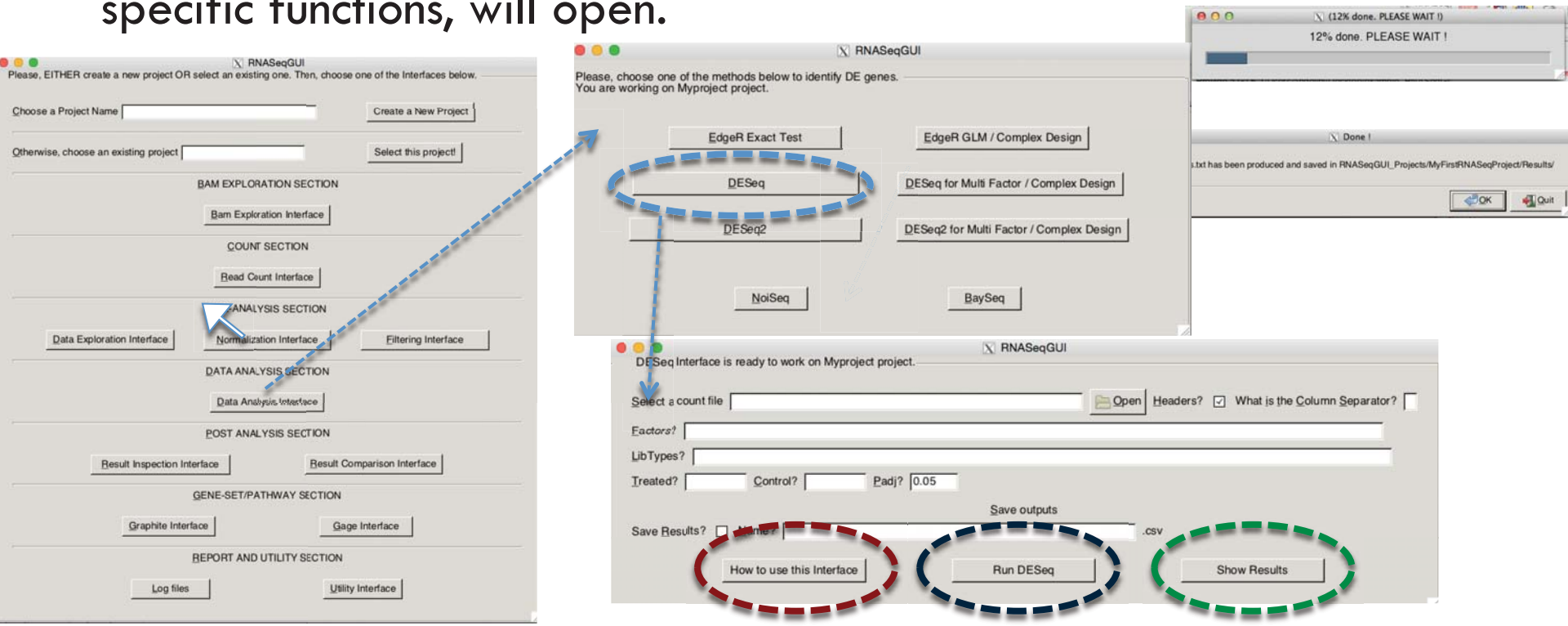
The analysis starts by creating a project or opening an existing project.

Then, the user can access any of RNASeqGUI sections.

Data Analysis Section is the core of RNASeqGUI and contains several methods to identify differentially expressed genes (DE).

Navigating RNASeqGUI

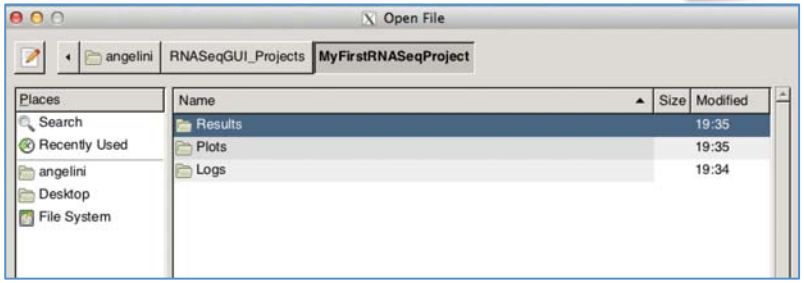
- By clicking to any specific section a new interface, that contains more specific functions, will open.



- The "*how to use*" button in each interface will guide the user in the choice of the best options and parameters.

Analyzing data with RNASeqGUI

- **BAM Exploration Section**
- **Count Section**
- **Pre-Analysis Section**
- **Data Analysis Section**
- **Post-Analysis Section**
- **GeneSet/pathway Section**
- **Report and Utility Section**



- Results are given in terms of tab-delimited-files, user friendly html-pages, summary tables, and figures→ Folder Result
- Figures are in pdf, and are customizable (in terms of colors and scale) → Folder Plots
- All actions are stored in a **html report** and **Rmd re-executble code** → Folder Logs
- Moreover, thanks to **caching, data, intermediate and final results** are directly connected with **databases.**
- **Reporting tools** provide navigable results directly linked to Ensembl

RR in RNASeqGUI (1)

The spirit of **RR** is now fully incorporated in **RNASeqGUI**.

- Thanks to the use of **R markdown** language, **RNASeqGUI automatically generates a dynamic report** of all analysis carried out on a given Project. The report includes both the R code, the figures and the summary of the results. **The report can be executed and results are being updated automatically**, if changes occur. The report also includes all versions of the R packages used, all steps, input/output parameters, file names and so on.
- **The report can be exported as HTML**
- **Caching** is used to speed up repetitive and computational expensive function calls by using results stored in pre-computed data-bases
- The report and the cached database are suitable for being submitted as documentary R file of data analysis for RR publication

RR in RNASeqGUI (2)

The Report contains "live" R code. Therefore, expert users can use RNASeqGUI to build the skeleton of their pipeline, then they can modify the code, or add their favorite method.

```
 * In the *Data Exploration Interface*, you clicked the **Plot Pairs of Counts** button at
`2014-06-21 19:52:32` and the counts.txt_1_vs_2_PlotCounts.pdf file has been saved in the
`MyFirstRNASeqProject/Plots` folder.

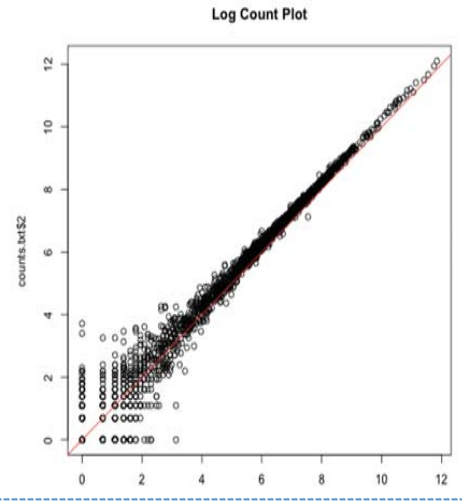
You chose the following count file: `
/Users/angelini/RNASeqGUI_Projects/MyFirstRNASeqProject/Results/demo_summarizeOverlaps/counts
.txt
` column1: `
1
` column2: `
2
` log: `
TRUE
`.
 This R code has been run:

 ```{r}
  x = read.table('/Users/angelini/RNASeqGUI_Projects/MyFirstRNASeqProject/Results/demo_summari
eOverlaps/counts.txt',header=TRUE,row.names=1)
  x = as.matrix(x)
  the.file ='/Users/angelini/RNASeqGUI_Projects/MyFirstRNASeqProject/Results/demo_summarizeOver
laps/counts.txt'
  column1 =1
  column2 =2
  log ='TRUE'
  Project ='MyFirstRNASeqProject'
   a=paste(getwd(),'/RNASeqGUI_Projects/',Project,'/Plots/',sep='')
   the.file2 = strsplit(the.file,'/')
   the.file2 = the.file2[[1]][length(the.file2[[1]])]  #estract the namefile
   outputName=paste(the.file2,'_',column1,'_vs_',column2,'_PlotCounts.pdf',sep='')
   b=paste(a,outputName,sep='')
   x_col1 = paste(the.file2,'$',column1,sep='')
   x_col2 = paste(the.file2,'$',column2,sep='')
   if (log==TRUE) { plot(log((x[,column1]) + 1), log((x[,column2]) +1), main='Log Count Plot',
xlab=x_col1, ylab=x_col2)}
   if (log==FALSE) { plot(x[,column1], x[,column2],main='Count Plot', xlab=x_col1, ylab=x_col2)
}
```

- In the *Data Exploration Interface*, you clicked the **Plot Pairs of Counts** button at 2014-06-21 19:52:32 and the counts.txt_1_vs_2_PlotCounts.pdf file has been saved in the MyFirstRNASeqProject/Plots folder.

You chose the following count file:
/Users/angelini/RNASeqGUI_Projects/MyFirstRNASeqProject/Results/demo_summarizeOverlaps/counts.txt column1: 1 column2: 2
log: TRUE . This R code has been run:

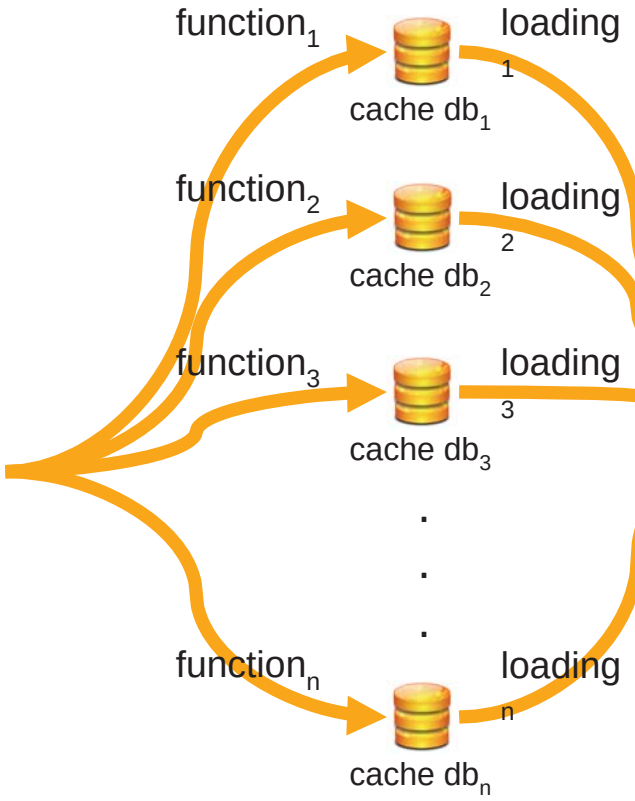
```
x = read.table('/Users/angelini/RNASeqGUI_Projects/MyFirstRNASeqProject/Results/demo_summarizeOverlaps/counts.txt',header=TRUE,row.names=1)
x = as.matrix(x)
the.file ='/Users/angelini/RNASeqGUI_Projects/MyFirstRNASeqProject/Results/demo_summarizeOverlaps/counts.txt'
column1 =1
column2 =2
log ='TRUE'
Project ='MyFirstRNASeqProject'
 a=paste(getwd(),'/RNASeqGUI_Projects/',Project,'/Plots/',sep='')
 the.file2 = strsplit(the.file,'/')
 the.file2 = the.file2[[1]][length(the.file2[[1]])]  #estract the namefile
 outputName=paste(the.file2,'_',column1,'_vs_',column2,'_PlotCounts.pdf',sep='')
 b=paste(a,outputName,sep='')
 x_col1 = paste(the.file2,'$',column1,sep='')
 x_col2 = paste(the.file2,'$',column2,sep='')
 if (log==TRUE) { plot(log((x[,column1]) + 1), log((x[,column2]) +1), main='Log Count Plot', xlab=x_col1, ylab=x_col2)}
 if (log==FALSE) { plot(x[,column1], x[,column2],main='Count Plot', xlab=x_col1, ylab=x_col2) }
 abline(a = 0, b = 1, col = 2)
```



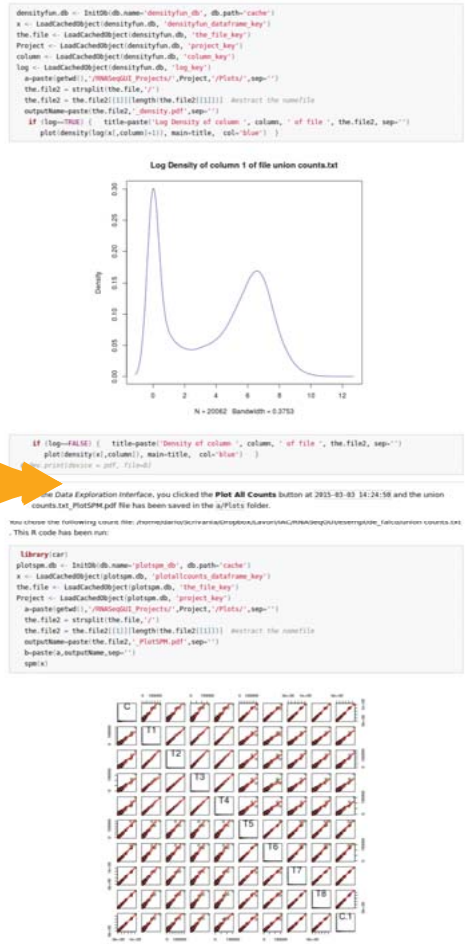
The limit of such approach is that each time the report is generated, the R code is executed and results are updated. → time consuming for NGS data
```

Caching in RNASeqGUI

Working session



Log Creation



Conclusions

- **RR** is very important for producing good Science, and it is expected that in the near future it will be mandatory.
- Editors have to encourage and promote RR.
- For those who develop computational tools it is important to provide novel software able to meet the need of RR.
- **RNAseqGUI** is one example in such directions, that combine the **GUI**s with the tools in R for **RR**. Therefore, for each project **RNAseqGU**I generates a **report** that keep track of all actions the user carried out. Moreover all data, intermediate and final results are **cached** to speed up computation, reporting layout is ameliorated and connected to database and webserver.

Take home message: As potential authors, try the best to produce RR, regardless to tool you are using. As potential developer, try the best to release tools that support RR.

Thanks to all members of

Supporting Projects

THANK YOU FOR THE ATTENTION

Questions?