



This is a case study from **Manuele Simi**,
Software Engineer at Weill Cornell Medicine

Description

[MetaR](#) has been developed to make data analysis easier for biomedical scientists with minimal computational skills.



Problem

Data analysis tools have become essential to the study of biology. The tools available today have been constructed using layers of technology developed over decades. Biologists and clinicians are often called upon to perform basic or advanced data analysis,

as their unique knowledge of the experiments that generate the data puts them in the ideal position to perform their own analysis. Yet statistical languages are not always easily accessible to them, and their limited computational experience is often an obstacle.

Solution

The [R language](#) is widely used for data analysis in biology. Expert biostatisticians and bioinformaticians have developed many R packages that implement advanced analyses for biological [high-throughput data](#). However, it takes a long time to acquire the computational and statistical knowledge required to fully benefit from the flexibility that R provides.

MetaR applies Language Workbench Technology to create a set of data analysis languages tailored to biologists. These languages automatically generate the underlying R code in order to take advantage of the packages developed in this language. MetaR is an integrated environment that makes it possible for users to write their own analyses

with minimal knowledge of the syntax of the constructs. The auto-completion features of the projectional editors, in addition to the composition of elements from different languages, offer a convenient way to set references between objects and help users avoid typos.

A key aspect of MetaR is the way it combines the user interface and scripting in a single platform. This feature makes it possible to analyze data more efficiently. Experts can design simplified data analysis languages that do not require any prior programming experience and behave like graphical user interfaces while still maintaining the advantages of scripting. MetaR also makes it possible to perform analyses in native or virtualized environments.

The MetaR Languages

Since working with high-throughput data often requires using tables of data as inputs, MetaR includes *Table* as a key element of the design. Tables are imported into the MetaR models and then analyzed with *metar- statements* inside *Analysis* elements.

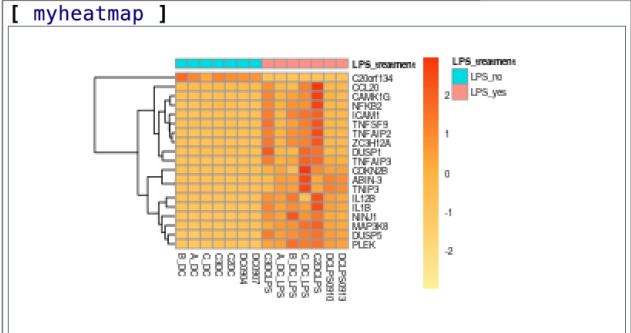
Metar-statements are declarative language constructs that remove the need for prior knowledge of the language syntax, which helps provide a smooth learning curve for beginners just starting out, who have no knowledge of programming.

Example of Table imported in MetaR:

Table GSE59364_DC_all.csv
File Path
/Users/mas2182/Docs/GSE59364_DC_all.csv
Columns
gene: *string* [ID]
mRNA len: *numeric*
genomic span: *numeric*
DC_normal: *numeric*
DC_treated: *numeric*
DC0904: *numeric* [counts, LPS=no]
DC0907: *numeric* [counts, LPS=no]
DCLPS0910: *numeric* [counts, LPS=yes]
DCLPS0913: *numeric* [counts, LPS=yes]
A_DC: *numeric* [counts, LPS=no]
A_DC_LPS: *numeric* [counts, LPS=yes]
B_DC: *numeric* [counts, LPS=no]
B_DC_LPS: *numeric* [counts, LPS=yes]
C_DC: *numeric* [counts, LPS=no]
C_DC_LPS: *numeric* [counts, LPS=yes]
C2DC: *numeric* [counts, LPS=no]
C2DCLPS: *numeric* [counts, LPS=yes]
C3DC: *numeric* [counts, LPS=no]
C3DCLPS: *numeric* [counts, LPS=yes]

Example of Analysis script:

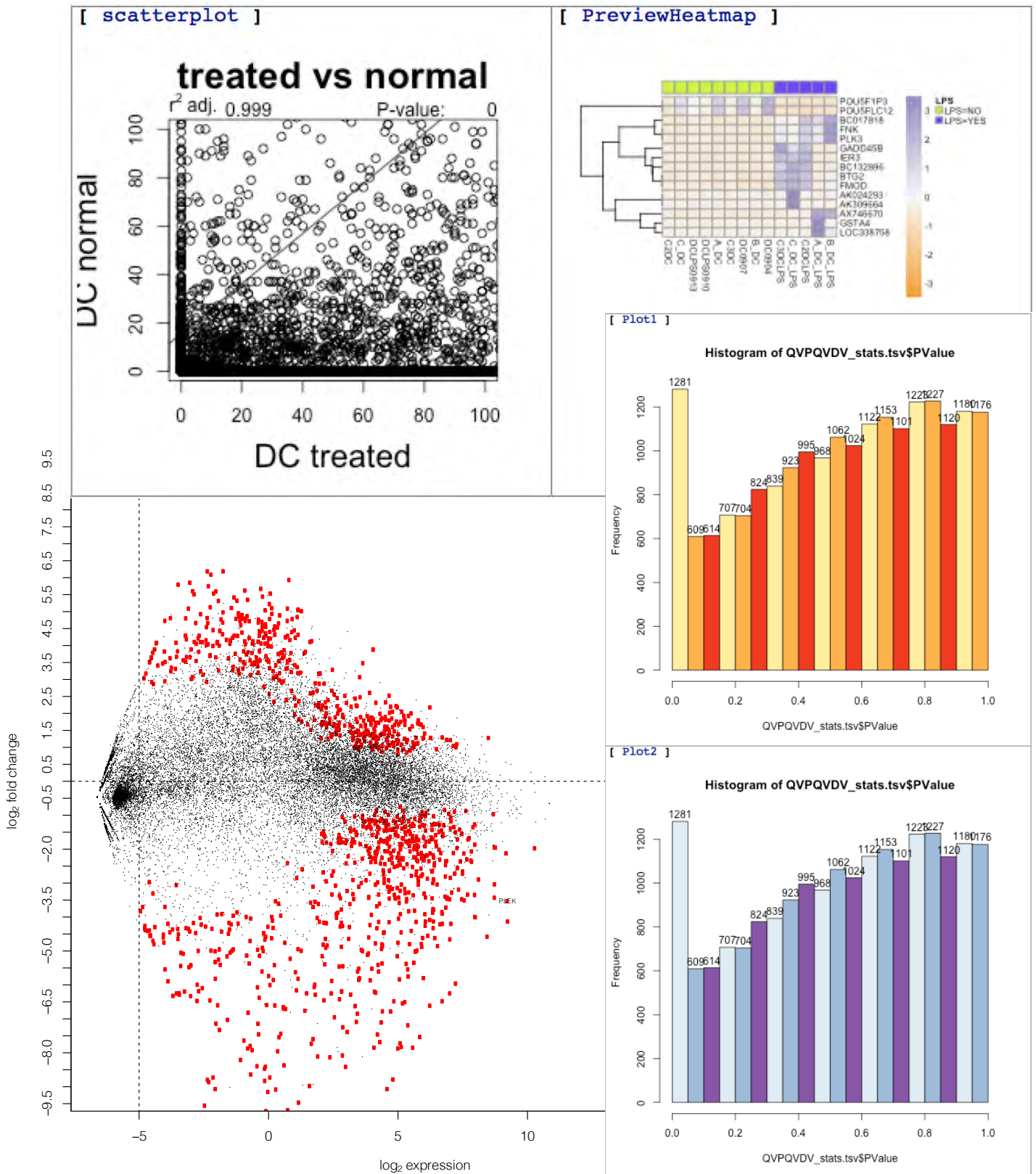
Analysis Diffexp

```
{  
  import table GSE59364_DC_all.csv  
  
  limma voom counts= GSE59364_DC_all.csv model: ~ 0 + LPS_treatment  
    comparing LPS=no - LPS=yes -> stats: results normalized: default  
  
  join ( results, GSE59364_DC_all.csv ) by group ID -> joined  
  
  subset rows joined when true: $(adj.P.Val) < 0.0001 -> subset  
  
  heatmap with subset select data by one or more group LPS=yes, group LPS=no -> myheatmap HeatmapStyle [  
    show names using group ID  
    annotate with these groups: LPS_treatment  
    scale values: scale by row  
    cluster columns: false cluster rows: true  
  ]  
  
  multiplot -> multi [ 1 cols x 1 rows ]   
  [ myheatmap ]  
    
  render myheatmap as PDF named "heatmap.pdf"  no style  
  render multi as PDF named "multi.pdf"  no style  
}
```

The Analysis script above shows how to import a table (**import** meta-statement), elaborate (**limma voom** – a popular method in statistical analysis to compare sets of genes) and transform (**join, subset rows**) its data, and finally draw (**heatmap**) and visualize/save (**multiplot, render**) a plot of the results, which is a very common set of procedures in data analysis.

This script uses only a very small subset of the meta-statements distributed with MetaR. However, the tool is general and can be readily extended to support a broad range of data analyses and visualizations. New languages can easily create and add new meta-statements that seamlessly integrate with those that already exist inside Analysis elements.

Other Examples of Data Visualization:



Target

MetaR can be used by:

- Biologists with no programming skills who want to analyze their data.
- Bioinformaticians who need to perform repetitive analyses and who find it beneficial to design and use specialized micro-languages to increase the efficiency and the consistency of their data analysis.
- R programmers who want to experiment with language composition and extension.
- Bioinformaticians who wish to package state-of-the-art analysis methods into user-friendly MetaR analysis language constructs. MetaR can act as a bridge that allows experts who develop analysis methods in R to distribute these methods to the broadest possible audience without having to invest a lot of effort into developing user interfaces.

Training Sessions

Training sessions are periodically offered to staff, students, postdocs, and investigators who hold an appointment in one of our Clinical & Translational Science Center institutions (Memorial Sloan-Kettering Cancer Center, the Hospital for Special Surgery, NewYork-Presbyterian Hospital, Hunter College and, Cornell University),

but they often include participants from other institutions in NYC. We have found that beginners can complete the assignments in session in less than 2 hours with MetaR, while more traditional training in R and its packages would require several sessions (6-24 hours) and an extensive technical background.

Why MPS

MetaR takes advantage of JetBrains MPS to make data analysis with the R language easier. MPS created brand new and unique possibilities for MetaR:

- The projectional editor's interactive features, such as auto-completion, provide guidance to beginners and experts alike when using the language to develop analyses.
- The ability to render nodes with a mix of text and graphical user interface components grants various levels of user experience.
- Language composition makes it possible for experts to extend MetaR with their own constructs and integrate them easily with the other languages.
- Run configurations allow MetaR to define and control how to execute analysis scripts in different environments and transparently install missing dependencies before the scripts are executed.
- The capability to generate R language scripts (the gold standard when it comes to data analysis) makes it possible for MetaR to build upon R's features and benefit from the vast array of packages (more than 10,000) available for R.

MetaR is distributed as a set of plugins for MPS.

References

- MetaR website: metar-languages.github.io
- Clinical & Translational Science Center website: ctscweb.weill.cornell.edu
- Clinical & Translational Science Center on Twitter: [@WCMC_CTSC](https://twitter.com/WCMC_CTSC)
- Campagne Laboratory website (where MetaR was initially developed): campagnelab.org
- Fabien Campagne, William ER Digan, Manuele Simi *MetaR: simple, high-level languages for data analysis with the R ecosystem* bioRxiv 2015 doi: [dx.doi.org/10.1101/030254](https://doi.org/10.1101/030254)