



Project Deliverable

Project acronym: SOUND	GA number: 633974
Project title: Statistical Multi-Omics Understanding of Patient Data	
Funding Scheme: Collaborative Project (H2020-PHC-2014-2015/H2020-PHC-2014-two-stage) Health, novel medical developments	
Project start date: 01 September 2015	Duration: 36 months
Project's coordinator: Dr Wolfgang Huber (European Molecular Biology Laboratory, Heidelberg)	

D10.1 Open-source software implementing *InteractiveReports* for displaying and scripting reproducible workflows; and a software catalogue of standard templates for informed decision-making

Due date of deliverable: Month 18 - 28.02.2017

Actual submission date: 23.08.2018

Organization name of lead contractor for this deliverable: Roswell Park Cancer Institute (RPCI)

Organization name of other involved partners:

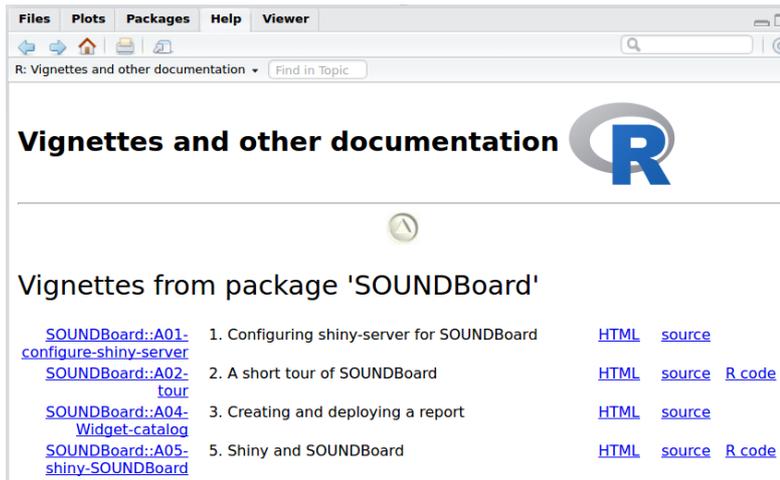
Personnel involved: Morgan, Martin. Turaga, Nitesh.

Project co-funded by the European Commission within the H2020 Program (2015-2018)		
Dissemination Level		
PU	Public	x
PP	Restricted to other program participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Deliverable description and summary

Implementation of InteractiveReports (implemented as SOUNDBoard) for displaying and scripting reproducible workflows, including an initial catalog of standard tools. Software is available in pre-release form at <https://github.com/Bioconductor/SOUNDBoard> ; install with `BiocManager::install("Bioconductor/SOUNDBoard")`.

The package contains a collection of vignettes to help the user configure their own SOUNDBoard server, get started using SOUNDBoard, exploring available widgets, and developing new widgets

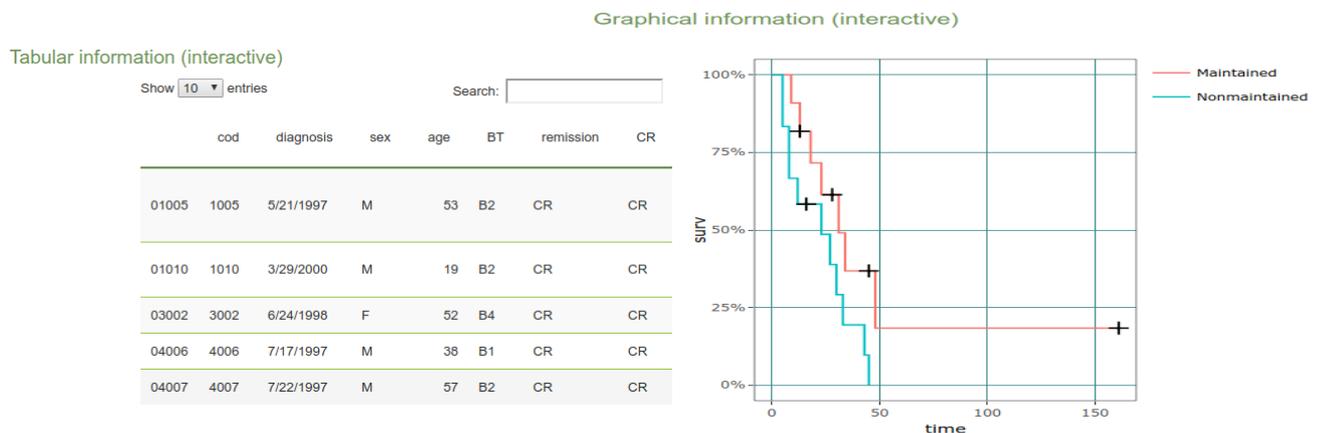


The package includes help pages for developers to create custom reports.

Help Pages

.SOUNDManager	Managing SOUNDBoard Reports
.SOUNDWidget	Create and use 'widgets' for consistent data presentation
deploy	Managing SOUNDBoard Reports
deploy_path	Managing SOUNDBoard Reports
sblog	Save, load, and report on SOUNDWidget instances.
sblogreport	Save, load, and report on SOUNDWidget instances.
sblogreport-method	Save, load, and report on SOUNDWidget instances.
sblogresource	Create and use 'widgets' for consistent data presentation
sblogsave	Save, load, and report on SOUNDWidget instances.
show-method	Managing SOUNDBoard Reports
SOUNDBoardWidget	Interactive display of SOUND resources through a the ShinyAppWidget class
SOUNDManager	Managing SOUNDBoard Reports
SOUNDManager-class	Managing SOUNDBoard Reports
SOUNDWidget	Create and use 'widgets' for consistent data presentation
SOUNDWidget-class	Create and use 'widgets' for consistent data presentation
urls	Managing SOUNDBoard Reports

Reports include static as well as dynamic (e.g., sortable tables, interactive graphic) elements



The reports are created entirely through R-based scripts, providing arbitrary flexibility in data processing. The following illustrates a code snippet creating an interactive 'ggplot' survival analysis plot.

```
library(ggplot2)
library(survival)
library(ggfortify)

fit <- survfit(Surv(time, status) ~ x, data = aml)
plot <- autoplot(fit, conf.int=FALSE)

tbl(x, "assay") <- list(
  case_uid = "PATIENT_1",
  assay = "ggplot",
  description = "Demonstrating ggplot management",
  resource = manage(x, plot)
)
```

The scripted nature of report production provide a high degree of reproducibility. The software contains a catalog of pre-defined 'widgets' including those illustrated above (for static text, interactive tables, interactive graphics), 'shiny' interactive embedded applets, an interactive heatmap widget, an OncoPrint widget, and a drug/gene interaction widget. Custom widgets are readily created by users implementing their own reports.

The SOUNDBoard application has been used in a preliminary fashion by at least two addition SOUND consortium partners, although no group has engaged in extensive adoption.