



## Project Deliverable

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

### D10.2 Website of Interactive Reports for select outcomes of WP2

Due date of deliverable: Month 36 – 30.08.2018

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: Turaga, Nitesh. Morgan, Martin.

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

## Deliverable description and summary

The deliverable targeted a web site of interactive reports. Software for production of these reports is available. The software is fully documented, including vignettes describing deployment, report development, use of standard widgets for display of interactive report elements, and customization through new widget development. While two SOUND partners expressed initial interest in presenting results through SOUNDBoard, the needs and / or technical expertise in these groups did not grow into active adoption. This was in part because of the need for additional software to rapidly process data for interactive presentation, and for standardized representation of complex, integrative data objects.

Our efforts associated with this deliverable have therefore focused on development of supporting infrastructure. The *BiocParallel* package has been enhanced to support a broader range of computational back-ends, especially standard cluster-based computing facilities that are likely to be accessible and relevant to academic and other research groups in the process of developing resources primarily for internal consumption. We also contributed to the development of the *GenomicRanges*, *RaggedExperiment* and *MultiAssayExperiment* packages for representing and conveniently manipulating bioinformatic data types relevant to SOUND participants.

## Software availability

SOUNDBoard software, including help pages and end-user vignettes, is available through a *Bioconductor* GitHub repository.

- <https://github.com/Bioconductor/SOUNDBoard>

Supporting software innovations are available in the following *Bioconductor* packages, available through the *Bioconductor* GitHub repository, the *Bioconductor* git repository, and through package 'landing pages'.

- <https://github.com/Bioconductor/GenomicRanges>

- <https://github.com/Bioconductor/RaggedExperiment>

- <https://github.com/Bioconductor/MultiAssayExperiment>

- git clone [git@git.bioconductor.org](mailto:git@git.bioconductor.org):packages/GenomicRanges

- git clone [git@git.bioconductor.org](mailto:git@git.bioconductor.org):packages/RaggedExperiment

- git clone [git@git.bioconductor.org](mailto:git@git.bioconductor.org):packages/MultiAssayExperiment

- BiocManager::install("GenomicRanges")

- BiocManager::install("RaggedExperiment")

- BiocManager::install("MultiAssayExperiment")

## Status

The SOUNDBoard package is mature and capable. It 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

Vignettes and other documentation 

---

Vignettes from package 'SOUNDBoard'

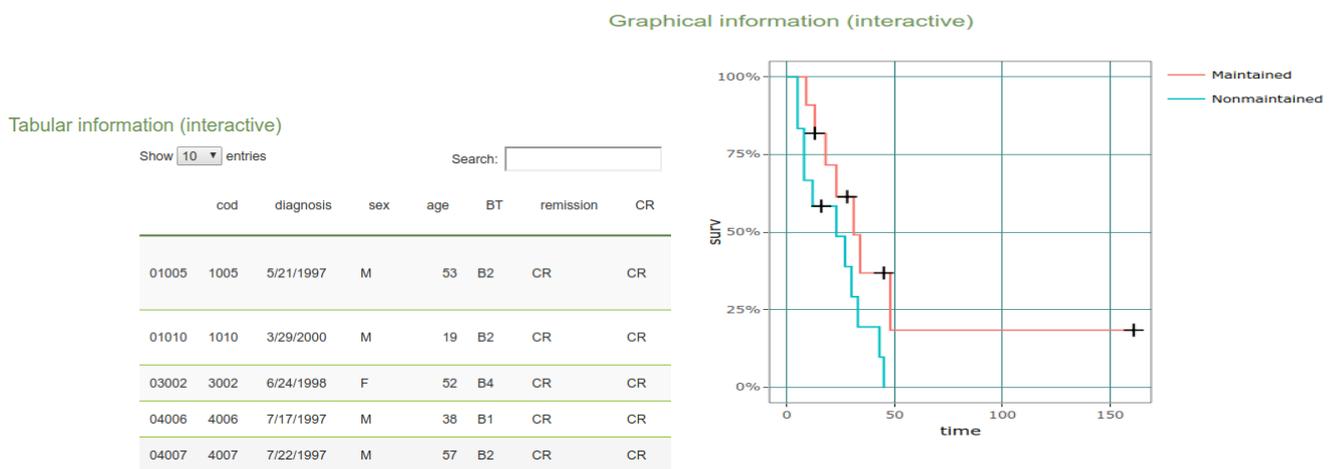
- [SOUNDBoard::A01-configure-shiny-server](#) 1. Configuring shiny-server for SOUNDBoard [HTML](#) [source](#)
- [SOUNDBoard::A02-tour](#) 2. A short tour of SOUNDBoard [HTML](#) [source](#) [R code](#)
- [SOUNDBoard::A04-Widget-catalog](#) 3. Creating and deploying a report [HTML](#) [source](#)
- [SOUNDBoard::A05-shiny-SOUNDBoard](#) 5. Shiny and SOUNDBoard [HTML](#) [source](#) [R code](#)

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

### Help Pages

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

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



SOUNDBoard reports often require access to bioinformatic data formats that can be accessed conveniently and manipulated in a performant manner.

The *GenomicRanges* package has been updated to include the *GPos* class for representing large collections of scores summarizing adjacent genomic positions (e.g., coverage profiles in copy-number or ChIP-seq studies). The container handles large genomic data, e.g., entire human genomes. It was developed through interaction with SOUND partner Dr. Julien Gagneur and his student Georg Stricker; it is used in the *GenoGAM* package (<https://bioconductor.org/packages/GenoGAM>) for the analysis of ChIP-seq data produced by these authors, as well as 14 other *Bioconductor* packages.

The *RaggedExperiment* package and class has been developed to represent genomic range-based data summarizes across samples, for the case where ranges are not shared across samples, e.g., with copy-number data. The *RaggedExperiment* package is used by 6 other packages directly, including the important *MultiAssayExperiment* infrastructure package. *RaggedExperiment*, introduced in 2017, is currently downloaded to more than 600 unique IP addresses per month.

The SOUND project has contributed to development of the *MultiAssayExperiment* package and class for integrating diverse data types into an object that enables convenient, coordinated manipulation across data types while minimizing the need for explicit 'book keeping' of, e.g., different sample identifier schemes used in different assays. In particular the project has contributed to some aspects of scalable data representation, and of representing copy-number and similar data types in the multi-assay context. *MultiAssayExperiment* is downloaded to approximately 800 unique IP addresses per month, and is a key component of some of the most important *ExperimentHub* resources described under Deliverable 8.2.

An important need for effective SOUNDBoard use is the ability to process large data to summarize suitable for presentation. The *BiocParallel* package provides a standard interface for parallel processing in *R* / *Bioconductor*, and is widely used (>17,000 unique IP downloads per month; >50 packages using *BiocParallel*). While existing *BiocParallel* functionality had good support for single-computer (multi-core) and light-weight socket-based multi-computer parallel computation, it did not have up-to-date facilities for parallel computation in clustered computing (e.g., managed via *slurm* or the Sun Grid Engine) environments common at academic and other research centers. Thus we have developed facilities to support cluster-based computing through the *BiocParallel BatchToolsParam()* interface to the *BatchTools* package. These facilities are fully developed and available in the current *BiocParallel* package. The *BiocParallel* package now provides very convenient facilities for candidate SOUNDBoard adopters to develop high-throughput computing work flows that can be easily scaled from single processor, to multiple cores, to cluster-based computing; *BiocParallel* provides particularly flexible facilities for logging, error recovery, and debugging during the software development process.