

# Package ‘savR’

October 8, 2014

**Type** Package

**Title** Parse and analyze Illumina SAV files

**Version** 1.2.0

**Date** 2014-02-04

**Author** R. Brent Calder

**Maintainer** R. Brent Calder <brent.calder@einstein.yu.edu>

**Description** Parse Illumina Sequence Analysis Viewer (SAV) files, access data, and generate QC plots.

**License** AGPL-3

**URL** <https://github.com/bcalder/savR>

**BugReports** <https://github.com/bcalder/savR/issues>

**Depends** ggplot2

**Imports** methods, reshape2, scales, gridExtra, XML

**Suggests** Cairo

**biocViews** Sequencing

## R topics documented:

savR-package . . . . .	2
buildReports . . . . .	3
correctedIntensities . . . . .	3
cycles . . . . .	5
directions . . . . .	5
extractionMetrics . . . . .	6
flowcellLayout . . . . .	7
illuminaFlowCellLayout-class . . . . .	7
illuminaRead-class . . . . .	8
location . . . . .	8

pfBoxplot . . . . .	9
plotFWHM . . . . .	9
plotIntensity . . . . .	10
plotQGT30 . . . . .	11
qualityHeatmap . . . . .	11
qualityMetrics . . . . .	12
reads . . . . .	12
run . . . . .	13
savCorrectedIntensityFormat-class . . . . .	14
savExtractionFormat-class . . . . .	14
savFormat-class . . . . .	15
savProject-class . . . . .	15
savQualityFormat-class . . . . .	16
savR . . . . .	16
savTileFormat-class . . . . .	17
tileMetrics . . . . .	17

<b>Index</b>	<b>19</b>
--------------	-----------

---

savR-package	<i>Parse and analyze Illumina SAV files</i>
--------------	---

---

## Description

Parse Illumina Sequence Analysis Viewer files

## Details

Package:	savR
Type:	Package
Version:	0.99.1
Date:	2014-01-29
License:	AGPL-3
LazyLoad:	yes

Parse Illumina Sequence Analysis Viewer (SAV) files, access data, and generate QC plots.

## Author(s)

R. Brent Calder <brent.calder@einstein.yu.edu>

## References

For information about Illumina SAV, please refer to  
[http://supportres.illumina.com/documents/documentation/software\\_documentation/sav/sequencinganalysisviewer\\_userguide\\_15020619c.pdf](http://supportres.illumina.com/documents/documentation/software_documentation/sav/sequencinganalysisviewer_userguide_15020619c.pdf)

For other implementations (and inspiration) please see  
<http://search.cpan.org/dist/Bio-IlluminaSAV/Bio/IlluminaSAV.pm>  
<https://bitbucket.org/invitae/illuminate>

---

buildReports	<i>Generate Illumina reports folder</i>
--------------	---

---

### Description

Generate a folder of images that approximates the format of the folder that was superceded by InterOp. Requires the Cairo package.

### Usage

```
buildReports(project, destination)

## S4 method for signature savProject,character
buildReports(project,
  destination = "./savR-reports")

## S4 method for signature savProject,missing
buildReports(project)
```

### Arguments

project	SAV project
destination	path to save reports folder

### Examples

```
## Not run:
example(savR)
buildReports(fc, "reports")

## End(Not run)
```

---

correctedIntensities	<i>Get Corrected Intensity data</i>
----------------------	-------------------------------------

---

### Description

Returns a data frame of corrected intensity data.

**Usage**

```
correctedIntensities(project)  
  
## S4 method for signature savProject  
correctedIntensities(project)
```

**Arguments**

project            SAV project

**Details**

lane: Lane number

tile: Tile ID

cycle: Cycle number

avg\_intensity: Average intensity

avg\_cor\_[ACGT]: Average corrected intensity of channel A, C, G, or T

avg\_cor\_called\_[ACGT]: Average corrected intensity for called clusters in channel A, C, G, or T

num\_{none|[ACGT]}: Number of called bases for no-call, A, C, G, or T

sig\_noise: Signal to noise ratio

**Value**

sorted data.frame of CI data.

**Examples**

```
example(savR)  
colnames(correctedIntensities(fc))
```

---

cycles	<i>Get the total number of cycles</i>
--------	---------------------------------------

---

**Description**

Accessor to obtain the total number of cycles sequenced in an Illumina sequencing run.

**Usage**

```
cycles(project)
```

```
## S4 method for signature savProject  
cycles(project)
```

**Arguments**

project            SAV project

**Value**

total number of cycles in run, including all sequencing and index reads.

**Examples**

```
example(savR)  
cycles(fc)
```

---

directions	<i>Get the number of sequence reads</i>
------------	---

---

**Description**

Returns the number of sequencing reads (excluding index reads).

**Usage**

```
directions(project)
```

```
## S4 method for signature savProject  
directions(project)
```

**Arguments**

project            SAV project

**Value**

number of reads

**Examples**

```
example(savR)
directions(fc)
```

---

extractionMetrics      *Get Extraction Metrics*

---

**Description**

Extraction (intensity and FWHM) metrics for lane, tile, and cycle.

**Usage**

```
extractionMetrics(project)

## S4 method for signature 'savProject'
extractionMetrics(project)
```

**Arguments**

project      SAV project

**Details**

lane: Lane number  
tile: Tile ID  
cycle: Cycle number  
FWHM\_[ACGT]: Full width at half maximum for A, C, G, or T  
int\_[ACGT]: Intensity of channel A, C, G, or T  
datestamp: Time/date stamp

**Value**

sorted data.frame of Extraction metrics

**Examples**

```
example(savR)
colnames(extractionMetrics(fc))
```

---

flowcellLayout	<i>Get flowcell layout</i>
----------------	----------------------------

---

**Description**

Accessor to obtain information about the characteristics of the flowcell from an Illumina sequencing run.

**Usage**

```
flowcellLayout(project)

## S4 method for signature savProject
flowcellLayout(project)
```

**Arguments**

project            SAV project

**Value**

[illuminaFlowCellLayout-class](#) object

**Examples**

```
example(savR)
flowcellLayout(fc)
```

---

illuminaFlowCellLayout-class	<i>Layout of an Illumina flowcell</i>
------------------------------	---------------------------------------

---

**Description**

Class representation of the features of an Illumina flow cell.

**Slots**

lanecount: Number of lanes on the flowcell  
surfacecount: Number of surfaces  
swathcount: Number of imaging swaths  
tilecount: Number of tiles per swath

---

illuminaRead-class	<i>Illumina read</i>
--------------------	----------------------

---

**Description**

Class representation of the features of an Illumina sequencing read.

**Slots**

number: the index of this read in sequencing

cycles: number of cycles in this read

index: logical representing whether or not this read is an index read

---

location	<i>Get Flowcell folder location</i>
----------	-------------------------------------

---

**Description**

Accessor to obtain the path to data for a particular SAV project.

**Usage**

```
location(project)
```

```
## S4 method for signature savProject
location(project)
```

**Arguments**

project            SAV project

**Value**

normalized path to Illumina run data.

**Examples**

```
example(savR)
location(fc)
```

---

 pfBoxplot

*PF Boxplot*


---

**Description**

Generate a boxplot of the numbers of clusters and the number of Illumina pass-filter clusters per tile and lane

**Usage**

```
pfBoxplot(project)

## S4 method for signature savProject
pfBoxplot(project)
```

**Arguments**

project          SAV project

---

 plotFWHM

*Generate FWHM plots*


---

**Description**

Plots the average full width of clusters at half maximum (FWHM) of each tile for a given cycle and base.

**Usage**

```
plotFWHM(project, cycle, base)

## S4 method for signature savProject,integer,character
plotFWHM(project, cycle = 1L,
  base = c("A", "C", "G", "T"))

## S4 method for signature savProject,missing,missing
plotFWHM(project)

## S4 method for signature savProject,integer,missing
plotFWHM(project, cycle)

## S4 method for signature savProject,missing,character
plotFWHM(project, base)
```

**Arguments**

project	SAV project
cycle	sequence cycle
base	nucleotide base (ACGT)

---

<code>plotIntensity</code>	<i>Plot flowcell intensity by base and cycle</i>
----------------------------	--

---

**Description**

Draws a representation of a flowcell, showing the average corrected called intensity values.

**Usage**

```
plotIntensity(project, cycle, base)

## S4 method for signature savProject,integer,character
plotIntensity(project, cycle = 1L,
  base = c("A", "C", "G", "T"))

## S4 method for signature savProject,missing,missing
plotIntensity(project)

## S4 method for signature savProject,integer,missing
plotIntensity(project, cycle)

## S4 method for signature savProject,missing,character
plotIntensity(project, base)
```

**Arguments**

project	A <a href="#">savProject-class</a> object
cycle	integer cycle number
base	character for nucleotide

---

plotQGT30	<i>Plot Quality &gt; 30 for a flowcell</i>
-----------	--

---

**Description**

Generate a plot for a given cycle of the percentage of clusters in each tile that are  $\geq$  Q30.

**Usage**

```
plotQGT30(project, cycle)

## S4 method for signature savProject,integer
plotQGT30(project, cycle = 1L)

## S4 method for signature savProject,missing
plotQGT30(project)
```

**Arguments**

project	SAV project
cycle	sequence cycle

---

qualityHeatmap	<i>Generate a heatmap of qualities</i>
----------------	--

---

**Description**

Plots a heatmap of quality vs cycle for a given lane for 1 or more sequence reads. Read qualities include sequence + index.

**Usage**

```
qualityHeatmap(project, lane, read)

## S4 method for signature savProject,integer,integer
qualityHeatmap(project, lane, read)

## S4 method for signature savProject,numeric,numeric
qualityHeatmap(project, lane, read)
```

**Arguments**

project	SAV project
lane	integer lane specification
read	integer vector of sequence reads to include (not including index reads)

---

qualityMetrics	<i>Get Quality Metrics data</i>
----------------	---------------------------------

---

**Description**

Quality metric by lane, tile and cycle.

**Usage**

```
qualityMetrics(project)
```

```
## S4 method for signature savProject  
qualityMetrics(project)
```

**Arguments**

project            SAV project

**Details**

lane: Lane number

tile: Tile ID

cycle: Cycle number

Q1-Q50: Number of clusters with quality of indicated column

**Value**

sorted data.frame of quality data

**Examples**

```
example(savR)  
colnames(qualityMetrics(fc))
```

---

reads	<i>Get reads</i>
-------	------------------

---

**Description**

Accessor to obtain information about the reads of a particular Illumina sequencing run.

**Usage**

```
reads(project)
```

```
## S4 method for signature savProject  
reads(project)
```

**Arguments**

project            SAV project

**Value**

List of [illuminaRead-class](#) objects

**Examples**

```
example(savR)
reads(fc)
```

---

run	<i>Get the Run ID</i>
-----	-----------------------

---

**Description**

Accessor to obtain the string identifier of an Illumina sequencing run.

**Usage**

```
run(project)

## S4 method for signature savProject
run(project)
```

**Arguments**

project            SAV project

**Value**

parsed Illumina run id

**Examples**

```
example(savR)
run(fc)
```

savCorrectedIntensityFormat-class

*Corrected Intensity formatter*

---

### **Description**

Lane, tile, cycle, average intensity, corrected intensities (ACGT), average corrected called intensities (ACGT), number of no-calls, number of (ACGT) calls, and signal to noise ratio.

### **Slots**

name: vector of column names

type: vector of data types of elements

lengths: vector of byte lengths for each element

order: vector of column names for sorting

version: integer version number

---

savExtractionFormat-class

*Extraction Metrics formatter*

---

### **Description**

Lane, tile, cycle, FWHM (ACGT), intensity (ACGT), datestamp, timestamp. Datestamp and timestamp are munged at the moment because R does not have native support for 32-bit unsigned integers and I have not implemented a solution.

### **Slots**

name: vector of column names

type: vector of data types of elements

lengths: vector of byte lengths for each element

order: vector of column names for sorting

version: integer version number

---

savFormat-class	<i>Base class for formatters</i>
-----------------	----------------------------------

---

**Description**

Defines the necessary slots to create parse different binary files using the same generic parser.

**Slots**

name: vector of column names  
type: vector of data types of elements  
lengths: vector of byte lengths for each element  
order: vector of column names for sorting  
version: integer version number

---

savProject-class	<i>SAV project class</i>
------------------	--------------------------

---

**Description**

Represents a flowcell, metadata and parsed SAV information

**Slots**

location: Full path to flowcell directory  
reads: List of [illuminaRead-class](#)  
layout: [illuminaFlowCellLayout-class](#)  
runid: Run ID  
number: Run number  
flowcell: Flowcell ID  
instrument: Instrument ID  
date: Run date  
cycles: Total number of cycles  
directions: Total number of sequence runs (ends)  
parsedData: SAV data

---

`savQualityFormat-class`*Quality Metrics formatter*

---

**Description**

Lane, tile, cycle, Q1-Q50 counts

**Slots**

`name`: vector of column names

`type`: vector of data types of elements

`lengths`: vector of byte lengths for each element

`order`: vector of column names for sorting

`version`: integer version number

---

`savR`*Build a SAV project*

---

**Description**

Constructor to build a [savProject-class](#) object and populate it. A SAV project consists of binary files generated by an Illumina sequencing run and placed in a folder named "InterOp". This folder contains a number of ".bin" files that contain statistics about the run. Creating this object parses all of the files and makes the data available for analysis.

**Usage**

```
savR(object)
```

```
## S4 method for signature character  
savR(object)
```

```
## S4 method for signature missing  
savR()
```

**Arguments**

`object`                      String Path to Flowcell data

**Examples**

```
fc <- savR(system.file("extdata", "MiSeq", package="savR"))  
fc
```

---

savTileFormat-class	<i>Tile Metrics formatter</i>
---------------------	-------------------------------

---

### Description

Lane, tile, code, value. Codes are:

### Details

100	Cluster Density
101	PF Cluster Density
102	Number of clusters
103	Number of PF clusters
400	Control lane

### Slots

name: vector of column names  
 type: vector of data types of elements  
 lengths: vector of byte lengths for each element  
 order: vector of column names for sorting  
 version: integer version number

---

tileMetrics	<i>Get Tile Metrics</i>
-------------	-------------------------

---

### Description

Returns the Tile Metrics SAV data.

### Usage

```
tileMetrics(project)

## S4 method for signature savProject
tileMetrics(project)
```

### Arguments

project            SAV project

**Details**

Metrics for each tile are encoded in the following format:

cluster density:	100
PF cluster density:	101
number of clusters:	102
number of PF clusters:	103
phasing for read N:	$(200 + (N - 1) * 2)$
prephasing for read N:	$(201 + (N - 1) * 2)$
percent aligned for read N:	$(300 + N - 1)$
control lane:	400

lane: Lane number

tile: Tile ID

code: Code described above

value: Value for code key

**Value**

sorted data.frame of tile metrics

**References**

Codes for Tile Metrics were obtained from the Python Illuminate package:  
<https://bitbucket.org/invitae/illuminate>

**Examples**

```
example(savR)
colnames(tileMetrics(fc))
```

# Index

- \*Topic **package**
  - savR-package, 2
- buildReports, 3
- buildReports, savProject, character-method (buildReports), 3
- buildReports, savProject, missing-method (buildReports), 3
- correctedIntensities, 3
- correctedIntensities, savProject-method (correctedIntensities), 3
- cycles, 5
- cycles, savProject-method (cycles), 5
- directions, 5
- directions, savProject-method (directions), 5
- extractionMetrics, 6
- extractionMetrics, savProject-method (extractionMetrics), 6
- flowcellLayout, 7
- flowcellLayout, savProject-method (flowcellLayout), 7
- illuminaFlowCellLayout-class, 7, 7, 15
- illuminaRead-class, 8, 13, 15
- location, 8
- location, savProject-method (location), 8
- pfBoxplot, 9
- pfBoxplot, savProject-method (pfBoxplot), 9
- plotFWHM, 9
- plotFWHM, savProject, integer, character-method (plotFWHM), 9
- plotFWHM, savProject, integer, missing-method (plotFWHM), 9
- plotFWHM, savProject, missing, character-method (plotFWHM), 9
- plotFWHM, savProject, missing, missing-method (plotFWHM), 9
- plotIntensity, 10
- plotIntensity, savProject, integer, character-method (plotIntensity), 10
- plotIntensity, savProject, integer, missing-method (plotIntensity), 10
- plotIntensity, savProject, missing, character-method (plotIntensity), 10
- plotIntensity, savProject, missing, missing-method (plotIntensity), 10
- plotQGT30, 11
- plotQGT30, savProject, integer-method (plotQGT30), 11
- plotQGT30, savProject, missing-method (plotQGT30), 11
- qualityHeatmap, 11
- qualityHeatmap, savProject, integer, integer-method (qualityHeatmap), 11
- qualityHeatmap, savProject, numeric, numeric-method (qualityHeatmap), 11
- qualityMetrics, 12
- qualityMetrics, savProject-method (qualityMetrics), 12
- reads, 12
- reads, savProject-method (reads), 12
- run, 13
- run, savProject-method (run), 13
- savCorrectedIntensityFormat-class, 14
- savExtractionFormat-class, 14
- savFormat-class, 15
- savProject-class, 10, 15, 16
- savQualityFormat-class, 16
- savR, 16
- savR, character-method (savR), 16

savR,missing-method (savR), [16](#)  
savR-package, [2](#)  
savTileFormat-class, [17](#)  
  
tileMetrics, [17](#)  
tileMetrics,savProject-method  
    (tileMetrics), [17](#)