

# Package ‘RoLexa’

October 8, 2014

**Type** Package

**Title** Statistical analysis of Solexa sequencing data

**Version** 1.20.0

**Date** 2009-10-06

**Author** Jacques Rougemont, Arnaud Amzallag, Christian Iseli, Laurent Farinelli, Ioannis Xenarios, Felix Naef

**Maintainer** Jacques Rougemont <jacques.rougemont@epfl.ch>

**Depends** R (>= 2.9.0), graphics, grDevices, methods, ShortRead

**Imports** mclust, Biostrings, graphics, grDevices, IRanges, methods, ShortRead, stats

**Enhances** fork

## **Description**

Provides probabilistic base calling, quality checks and diagnostic plots for Solexa sequencing data

**License** GPL-2

**biocViews** Sequencing, DataImport, Preprocessing, QualityControl

## **R topics documented:**

BatchAnalysis . . . . .	2
CombinedPlot . . . . .	3
DeCorrelateChannels . . . . .	4
FilterResults . . . . .	6
ForkBatch . . . . .	7
SaveResults . . . . .	9
SeqScore . . . . .	10
TileImage . . . . .	11

<b>Index</b>	<b>13</b>
--------------	-----------

---

BatchAnalysis

*Batch Analysis*


---

## Description

Generate summary plots of the results of a base calling batch

## Usage

```
## S4 method for signature RolexaRun
PlotCycles(run=Rolexa.env, int, seq,
cycles=c(1,11,21,31), par=list())
PlotCycles(run,...)
## S4 method for signature RolexaRun
BatchAnalysis(run=Rolexa.env, seq, scores, what=c("length","information","base","ratio","iupac"), main, par)
BatchAnalysis(run,...)
QualityBoxPlots(run=Rolexa.env, seq, cycles, par=list(las=2))
```

## Arguments

run	a RolexaRun object defining the run parameters
int	a <a href="#">SolexaIntensity</a> object
seq	a <a href="#">DNAStrngSet</a> object
scores	a matrix of base quality scores (one column per base, one row per sequence)
what	select one the plot types
main	a title for the plot
cycles	the cycles to plot
par	parameters for the plotting functions
...	additional arguments, ignored

## Details

Four types of diagnostic plots can be selected with the what argument of BatchAnalysis:

- lengthshows the histogram of tag lengths,
- informationthe distribution of information content per sequenced base, namely  $((2 \times \text{length}(\text{tag}) - \text{total\_entropy}(\text{tag})) / \text{length}(\text{tag}))$ ,
- basethe base composition of the sequences,
- ratiothe ratio of complementary bases,
- iupacthe proportion of the different classes of ambiguous bases along the sequences.

QualityBoxPlots makes boxplots of quality scores along the sequences. PlotCycles will execute [SeqScore](#) with plot=TRUE.

**Author(s)**

Jacques Rougemont, Arnaud Amzallag, Christian Iseli, Laurent Farinelli, Ioannis Xenarios, Felix Naef

**References**

Probabilistic base calling of Solexa sequencing data, BMC Bioinformatics 2008, 9:431

**See Also**

[SaveResults](#) to save the results produced by [SeqScore](#) or [FilterResults](#).

**Examples**

```
path = SolexaPath(system.file("extdata", package="ShortRead"))
rolenv = SetModel(idsep="_")
int = readIntensities(path,pattern="s_1_0001",withVariability=FALSE)
seq = CombineReads(run=rolenv,path=path,pattern="s_1_0001_seq*")
results = SeqScore(run=rolenv,int=int,seqInit=seq,cycles=1:36)
PlotCycles(run=rolenv,int=int,seq=seq,cycles=1:4)
par(ask=TRUE)
BatchAnalysis(rolenv,sread(seq),matrix(),what="iupac")
BatchAnalysis(rolenv,sread(seq),results$entropy,what="information")
results = FilterResults(run=rolenv,results=results)
BatchAnalysis(rolenv,sread(seq),results,what="length")
seq = readFastq(path)
par(mar=c(4,4,1,1),cex=1.5,lwd=2)
QualityBoxPlots(rolenv,seq,cycles=10:36)
```

---

CombinedPlot

*Diagnostic plots*

---

**Description**

Generate plots to visually assess the quality of select colonies or sequencing cycles

**Usage**

```
## S4 method for signature RolexaRun
CombinedPlot(run=Rolexa.env, int, seq, scores, colonies = 1:4, par = list())
CombinedPlot(run,...)
## S4 method for signature SolexaIntensity
ChannelHistogram(int, cycles = c(1,18,36),
threeModes = FALSE, par = list())
ChannelHistogram(int,...)
```

**Arguments**

run	a <a href="#">RoLexaRun</a> object defining the run parameters
int	a <a href="#">SolexaIntensity</a> object
seq	a <a href="#">ShortRead</a> object
scores	a matrix of base quality scores (one column per base, one row per sequence)
cycles	the list of cycles to plot
colonies	the list of rows to select for plotting
threemodes	fit and plot a mixture of 3 gaussians (2 by default)
par	parameters for the plotting function
...	additional arguments, ignored

**Details**

CombinedPlot creates one plot for each selected colony with the sequence along the x axis, the four intensities plotted as barplots above each base and the quality scores as a line plot below the sequence.

ChannelHistogram plots histograms and signal-noise thresholds for each of the four intensity channels on selected cycles. Fits to 2 or 3 gaussians are overlaid on the histograms.

**Author(s)**

Jacques Rougemont, Arnaud Amzallag, Christian Iseli, Laurent Farinelli, Ioannis Xenarios, Felix Naef

**References**

Probabilistic base calling of Solexa sequencing data, BMC Bioinformatics 2008, 9:431

**Examples**

```
path = SolexaPath(system.file("extdata", package="ShortRead"))
rolenv = SetModel(idsep="_")
int = readIntensities(path,pattern="s_1_0001",withVariability=FALSE)
seq = CombineFastQ(run=rolenv,path=path)
CombinedPlot(run=rolenv,int=int,seq=seq,scores=as(quality(seq),"matrix"),colonies=1)
```

---

DeCorrelateChannels    *Correct for global correlations and biases*

---

**Description**

Functions to correct for global correlations between color channels or between successive sequencing cycles

**Usage**

```

## S4 method for signature SolexaIntensity
DeCorrelateChannels(int,cycles=seq(1,dim(int)[3],by=1),theta=matrix(rep(c(0.8806742,1.3727418,0.883
## S4 method for signature array
DeCorrelateChannels(int,cycles=seq(1,dim(int)[3],by=1),theta=matrix(rep(c(0.8806742,1.3727418,0.883
DeCorrelateChannels(int,...)
## S4 method for signature SolexaIntensity
OptimizeAngle(int,cycles=seq(1,dim(int)[3],by=1),...)
OptimizeAngle(int,...)
## S4 method for signature SolexaIntensity
DeCorrelateCycles(int,ncycles=dim(int)[3],rate=1.8e-2)
## S4 method for signature array
DeCorrelateCycles(int,ncycles=dim(int)[3],rate=1.8e-2)
DeCorrelateCycles(int,...)
## S4 method for signature SolexaIntensity
OptimizeRate(int,ncycles=dim(int)[3],...)
OptimizeRate(int,...)
## S4 method for signature RolexaRun
TileNormalize(run=Rolexa.env,int,cycles=seq(1,dim(int)[3],by=1))
TileNormalize(run,...)

```

**Arguments**

run	a <a href="#">RolexaRun</a> object defining the run parameters
int	a <a href="#">SolexaIntensity</a> object or an array
cycles, ncycles	the cycles or the number of cycles (starting from 1) to apply the correction to
theta	a $\text{length}(\text{cycles}) \times 4$ matrix with four angles per cycle defining the coordinate changes
rate	the rate of nucleotide mis-incorporation at each cycle
...	additional arguments passed to <a href="#">optim</a>

**Details**

DeCorrelateChannels applies to coordinate transforms: one transforming the axes 1,2 to the axes with angles  $\theta[1:2]$  relative to axis 1, and similarly with axes 3,4 and angles  $\theta[3:4]$ . These angles can be calculated with `OptimizeAngle` which minimizes the correlations between channel 1 and 2, and between channel 3 and 4, for each cycle. DeCorrelateCycles assumes that at each cycles, a fraction rate of sequences fail to incorporate any nucleotides and therefore the sequence lengths at each colony display a binomial distribution which is corrected for by taking into account the intensity measured at previous cycles. OptimizeRate calculates a rate that minimizes correlations between consecutive cycles.

TileNormalize estimates the local trend by [loess](#) fitting of the model  $\text{int} \sim x+y$  and subtracts it from the intensity matrix.

**Value**

TileNormalize, DeCorrelateChannels and DeCorrelateCycles return an object of the same type as int corrected for spurious correlations. OptimizeAngle returns an length(cycles)\*4 matrix and OptimizeRate returns a single positive real number.

**Author(s)**

Jacques Rougemont, Arnaud Amzallag, Christian Iseli, Laurent Farinelli, Ioannis Xenarios, Felix Naef

**References**

Probabilistic base calling of Solexa sequencing data, BMC Bioinformatics 2008, 9:431

**See Also**

TileNormalize

**Examples**

```
path = SolexaPath(system.file("extdata", package="ShortRead"))
rolenv = SetModel(idsep="_")
int = readIntensities(path,pattern="s_1_0001",withVariability=FALSE)

int1 = DeCorrelateChannels(int=int,cycles=1:5,theta=OptimizeAngle(int=int,cycles=1:5))
int2 = DeCorrelateCycles(int=int1,ncycles=5,rate=OptimizeRate(int=int1))
int3 = TileNormalize(run=rolenv,int=int,cycles=1)
seq = CombineReads(run=rolenv,path=path,pattern="s_1_0001_seq*")
PlotCycles(run=rolenv,int=int3,seq=seq,cycles=1:4)
```

---

FilterResults

*FilterResults*

---

**Description**

Filter basecalling results to keep only high-quality bases

**Usage**

```
## S4 method for signature RolexaRun
FilterResults(run=Rolexa.env,results)
FilterResults(run,...)
```

**Arguments**

run	a <a href="#">RolexaRun</a> object defining the run parameters
results	a results object from <a href="#">SeqScore</a>
...	additional arguments, ignored

## Details

FilterResults filters the sequences according to the entropy thresholds set by [IThresholds](#) and applies the tag length cutoff [MinimumTagLength](#).

The algorithm works as follows: for each tag the base entropies are searched for a sub-vector  $k+1:l$  such that  $\text{sum}(\text{entropy}[n, 5+k+1:l]) \leq \text{IThresholds}[l]$  where  $l = \text{MinimumTagLength}$ . If such a sub-vector exists, it is then extended in both direction until the total entropy exceeds the threshold:  $\text{sum}(\text{results}[n, 5+k1:k2]) > \text{IThresholds}[k2-k1+1]$ .

The tag is then shortened: `substr(results[n,5],k1,k2)`, but [ACGT] bases to left of  $k1$  and to the right of  $k2$  are added. The [Barcode](#) first bases of the tags will always be included in a separate column if this parameter has been set. If `PET=TRUE` then the whole procedure is applied independently to each half of the sequence (and two separate sets of tags and scores are returned) and the barcode (if any) is assumed to be in-between the two paired tags.

## Value

FilterResults returns an object suitable for [SaveResults](#)

## Author(s)

Jacques Rougemont, Arnaud Amzallag, Christian Iseli, Laurent Farinelli, Ioannis Xenarios, Felix Naef

## References

Probabilistic base calling of Solexa sequencing data, BMC Bioinformatics 2008, 9:431

## See Also

[readFastq](#) to read fastq files, [SeqScore](#) and [FilterResults](#) to produce results for [SaveResults](#)

---

ForkBatch

*Multi-threaded Probabilistic Base Calling*

---

## Description

Performs multi-threaded base calling on a collection of intensity files generated by the Solexa image analysis software

## Usage

```
ForkBatch(run=Rolexa.env,path,outputpath="./",prefix="rs_",nthreads=3,nfiles=2,lane=1,tiles=1:100,...)
## S4 method for signature RolexaRun
OneBatch(run,path,lane,tiles,outputpath,prefix)
OneBatch(run,...)
```

## Arguments

run	a <a href="#">RolexaRun</a> object defining the run parameters
path	a <a href="#">SolexaPath</a> object defining providing the input paths
outpath	the path to the output directory
prefix	output file prefix, see <a href="#">SaveResults</a>
nthreads	number of threads to use
nfiles	number of input files to concatenate in one batch
lane	the lane number to analyze
tiles	a subset of tiles to read
...	further arguments passed to the <a href="#">RolexaRun</a> constructor

## Details

The function [ForkBatch](#) runs through the list of input files, concatenates them by batches of `nfiles`, then calls [OneBatch](#) in each of the `nthreads` threads until all batches have been processed. Each batch results are passed to [FilterResults](#) and saved in an output file inside `outpath`.

## Author(s)

Jacques Rougemont, Arnaud Amzallag, Christian Iseli, Laurent Farinelli, Ioannis Xenarios, Felix Naef

## References

Probabilistic base calling of Solexa sequencing data, BMC Bioinformatics 2008, 9:431

## See Also

[CombineFastQ](#), [CombineReads](#) and [SaveResults](#)

## Examples

```
path = SolexaPath(system.file("extdata", package="ShortRead"))
rolenv = SetModel(idsep="_")
## Not run:
#This will take some time to complete:
library(fork)
ForkBatch(run=rolenv,path=path,tiles=1)

## End(Not run)
```



SaveResults

*SaveResults***Description**

Read and write data in a convenient form for Rolexa base-calling

**Usage**

```
## S4 method for signature RolexaRun
SaveResults(run=Rolexa.env,results,outputpath,prefix="rs_")
SaveResults(run,...)
## S4 method for signature RolexaRun,SolexaPath
CombineReads(run=Rolexa.env,path,pattern="s_[1-8]_0[01][0-9]*_seq*")
CombineReads(run,path,...)
## S4 method for signature RolexaRun,SolexaPath
CombineFastQ(run=Rolexa.env,path,pattern="s_[1-8]_0[01][0-9]*",sext="_seq*",pext="_prb*")
CombineFastQ(run,path,...)
```

**Arguments**

run	a RolexaRun object defining the run parameters
results	a results list, as given by <a href="#">FilterResults</a> or <a href="#">SeqScore</a>
outputpath	a directory name for the output files
path	a <a href="#">SolexaPath</a> object
prefix	a prefix string for output file names
pattern	a pattern for selecting Solexa output files, see <a href="#">readXStringColumns</a>
sext	file extension tag for sequence files <a href="#">readPrb</a>
pext	file extension tag for prb files, see
...	additional arguments, ignored

**Details**

CombineReads reads "\\_seq" files and splits the columns to create a [ShortRead](#) object, CombineFastQ reads "\\_seq" and "\\_prb" files and combines them into a [ShortReadQ](#) object, SaveResults creates a [ShortReadQ](#) objects from the output of [FilterResults](#) and writes it to a file using [writeFastq](#).

**Value**

CombineReads returns a [ShortRead](#) object, CombineFastQ returns a [ShortReadQ](#) object,

**Author(s)**

Jacques Rougemont, Arnaud Amzallag, Christian Iseli, Laurent Farinelli, Ioannis Xenarios, Felix Naef

**References**

Probabilistic base calling of Solexa sequencing data, BMC Bioinformatics 2008, 9:431

**See Also**

[readFastq](#) to read fastq files, [SeqScore](#) and [FilterResults](#) to produce results for SaveResults

---

 SeqScore

*Fit and Plot intensities*


---

**Description**

Model-based classification of intensity data points, to either perform a base calling or generate diagnostic plots

**Usage**

```
## S4 method for signature RolexaRun
SeqScore(run=Rolexa.env,int,seqInit,colonies,cycles,plot=FALSE)
SeqScore(run,...)
```

**Arguments**

run	a RolexaRun object defining the run parameters
int	a <a href="#">SolexaIntensity</a> object
seqInit	a <a href="#">ShortRead</a> object
colonies	which colonies to select
cycles	which cycles to select
plot	if TRUE do a plot rather than perform a base-calling
...	additional arguments, ignored

**Details**

This will use the EEV model of [mclust](#) to fit the data clouds with a mixture of 4 gaussian distributions. and generate a list of tags and entropy scores for each sequenced colony (if plot is FALSE) or plots two 2-dimensional projections for each selected cycle with gaussian parameters represented by standard ellipses and data points colored according to the induced classification.

If [fit](#) is TRUE, then the [EM](#) algorithm is run to convergence, otherwise only an [E-step](#) and an [M-step](#) are performed to evaluate the probabilities.

The fitting procedure then uses [HThresholds](#) to decide if a base is unambiguous and if degenerate IUPAC codes will be used.

**Value**

if plot is FALSE, SeqScore returns a list with an id slot containing the colonies coordinates, an sread slot which is a [DNAStrngSet](#) object and an entropy matrix

**Author(s)**

Jacques Rougemont, Arnaud Amzallag, Christian Iseli, Laurent Farinelli, Ioannis Xenarios, Felix Naef

**References**

Probabilistic base calling of Solexa sequencing data, BMC Bioinformatics 2008, 9:431

**Examples**

```
path = SolexaPath(system.file("extdata", package="ShortRead"))
rolenv = SetModel(idsep="_")
int = readIntensities(path,pattern="s_1_0001",withVariability=FALSE)
seq = CombineReads(run=rolenv,path=path,pattern="s_1_0001_seq*")
results = SeqScore(run=rolenv,int=int,seqInit=seq,cycles=1:10)
results$sread
```

---

TileImage	<i>Reconstruct tile image</i>
-----------	-------------------------------

---

**Description**

Generate an image of the local intensity average

**Usage**

```
## S4 method for signature SolexaIntensity
TileImage(int,cycle,tile,channel=c('A','C','G','T'),ncell=30)
TileImage(int,...)
```

**Arguments**

int	a <a href="#">SolexaIntensity</a> object
cycle	the cycle to make an image of
tile	the tile to make an image of
channel	the channel ('A', 'C', 'G' or 'T') to make an image of
ncell	the number of divisions in each dimension for the image
...	additional arguments, ignored

**Details**

TileImage creates an image of the intensity on a tile, in a given channel and at a given cycle. The tile is divided into ncell\*ncell cells and the average intensity in each cell is represented on a color scale.

**Author(s)**

Jacques Rougemont, Arnaud Amzallag, Christian Iseli, Laurent Farinelli, Ioannis Xenarios, Felix Naef

**References**

Probabilistic base calling of Solexa sequencing data, BMC Bioinformatics 2008, 9:431

**Examples**

```
path = SolexaPath(system.file("extdata", package="ShortRead"))
rolenv = SetModel(idsep="_")
int = readIntensities(path,pattern="s_1_0001",withVariability=FALSE)
par(mfrow=c(2,2))
for (c in c(A,C,G,T))
  TileImage(int=int,cycle=1,tile=readInfo(int)$tile[1],channel=c,ncell=5)
int2 = TileNormalize(rolenv,int=int,cycles=1)
x11()
par(mfrow=c(2,2))
for (c in c(A,C,G,T))
  TileImage(int=int2,cycle=1,tile=readInfo(int)$tile[1],channel=c,ncell=5)
```

# Index

- \*Topic **IO**
  - SaveResults, 9
- \*Topic **cluster**
  - SeqScore, 10
- \*Topic **datagen**
  - SaveResults, 9
- \*Topic **dplot**
  - BatchAnalysis, 2
  - CombinedPlot, 3
- \*Topic **hplot**
  - TileImage, 11
- \*Topic **iteration**
  - ForkBatch, 7
- \*Topic **loess**
  - DeCorrelateChannels, 4
  - TileImage, 11
- \*Topic **manip**
  - BatchAnalysis, 2
  - CombinedPlot, 3
  - FilterResults, 6
- \*Topic **multivariate**
  - DeCorrelateChannels, 4
  - SeqScore, 10
- \*Topic **regression**
  - DeCorrelateChannels, 4
- \*Topic **utilities**
  - BatchAnalysis, 2
  - CombinedPlot, 3
  - ForkBatch, 7
- Barcode, 7
- BatchAnalysis, 2
- BatchAnalysis,RolexaRun-method  
(BatchAnalysis), 2
- ChannelHistogram (CombinedPlot), 3
- ChannelHistogram,SolexaIntensity-method  
(CombinedPlot), 3
- CombinedPlot, 3
- CombinedPlot,RolexaRun-method  
(CombinedPlot), 3
- CombineFastQ, 8
- CombineFastQ (SaveResults), 9
- CombineFastQ,RolexaRun,SolexaPath-method  
(SaveResults), 9
- CombineReads, 8
- CombineReads (SaveResults), 9
- CombineReads,RolexaRun,SolexaPath-method  
(SaveResults), 9
- DeCorrelateChannels, 4
- DeCorrelateChannels,array-method  
(DeCorrelateChannels), 4
- DeCorrelateChannels,SolexaIntensity-method  
(DeCorrelateChannels), 4
- DeCorrelateCycles  
(DeCorrelateChannels), 4
- DeCorrelateCycles,array-method  
(DeCorrelateChannels), 4
- DeCorrelateCycles,SolexaIntensity-method  
(DeCorrelateChannels), 4
- DNAStrngSet, 2, 10
- E-step, 10
- EM, 10
- FilterResults, 3, 6, 7–10
- FilterResults,RolexaRun-method  
(FilterResults), 6
- fit, 10
- ForkBatch, 7, 8
- HThresholds, 10
- IThresholds, 7
- loess, 5
- M-step, 10
- mclust, 10

MinimumTagLength, 7

OneBatch, 8

OneBatch (ForkBatch), 7

OneBatch,RolexaRun-method (ForkBatch), 7

optim, 5

OptimizeAngle (DeCorrelateChannels), 4

OptimizeAngle,SolexaIntensity-method  
(DeCorrelateChannels), 4

OptimizeRate (DeCorrelateChannels), 4

OptimizeRate,SolexaIntensity-method  
(DeCorrelateChannels), 4

PET, 7

PlotCycles (BatchAnalysis), 2

PlotCycles,RolexaRun-method  
(BatchAnalysis), 2

QualityBoxPlots (BatchAnalysis), 2

readFastq, 7, 10

readPrb, 9

readXStringColumns, 9

RolexaRun, 4–6, 8

SaveResults, 3, 7, 8, 9

SaveResults,RolexaRun-method  
(SaveResults), 9

SeqScore, 2, 3, 6, 7, 9, 10, 10

SeqScore,RolexaRun-method (SeqScore), 10

ShortRead, 4, 9, 10

ShortReadQ, 9

SolexaIntensity, 2, 4, 5, 10, 11

SolexaPath, 8, 9

TileImage, 11

TileImage,SolexaIntensity-method  
(TileImage), 11

TileNormalize (DeCorrelateChannels), 4

TileNormalize,RolexaRun-method  
(DeCorrelateChannels), 4

writeFastq, 9