

# Package ‘autonomics’

November 1, 2022

**Type** Package

**Title** Generifying and intuifying cross-platform omics analysis

**Version** 1.7.0

**Description**

This package offers a generic and intuitive solution for cross-platform omics data analysis. It has functions for import, preprocessing, exploration, contrast analysis and visualization of omics data. It follows a tidy, functional programming paradigm.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**VignetteBuilder** knitr

**biocViews** DataImport, DimensionReduction, GeneExpression, MassSpectrometry, Preprocessing, PrincipalComponent, RNASeq, Software, Transcription

**BugReports** <https://bitbucket.org/graumannlabtools/autonomics>

**URL** <https://github.com/bhagwataditya/autonomics>

**RoxygenNote** 7.1.1

**Depends** R (>= 4.0)

**Imports** abind, assertive, BiocFileCache, BiocGenerics, colorspace, data.table, edgeR, ggplot2, ggrepel, graphics, grDevices, grid, gridExtra, limma, magrittr, matrixStats, methods, MultiAssayExperiment, parallel, pcaMethods, rappdirs, rlang, R.utils, readxl, S4Vectors, scales, stats, stringi, SummarizedExperiment, tidyr, tools, utils

**Suggests** affy, AnnotationDbi, BiocManager, diagram, GenomicRanges, GEOquery, hgu95av2.db, ICSNP, knitr, lme4, lmerTest, MASS, mixOmics, mpm, nlme, org.Hs.eg.db, org.Mm.eg.db, RCurl, remotes, rmarkdown, ropls, Rsubread, rtracklayer, seqinr, statmod, testthat

**git\_url** <https://git.bioconductor.org/packages/autonomics>

**git\_branch** master

**git\_last\_commit** 648ec51

**git\_last\_commit\_date** 2022-11-01

**Date/Publication** 2022-11-01

**Author** Aditya Bhagwat [aut, cre],  
 Shahina Hayat [aut],  
 Anna Halama [ctb],  
 Richard Cotton [ctb],  
 Laure Cougnaud [ctb],  
 Rudolf Engelke [ctb],  
 Hinrich Goehlmann [sad],  
 Karsten Suhre [sad],  
 Johannes Graumann [aut, sad, rth]

**Maintainer** Aditya Bhagwat <aditya.bhagwat@uni-marburg.de>

## R topics documented:

.read_maxquant . . . . .	4
.read_metabolon . . . . .	6
.read_rectangles . . . . .	8
.read_rnaseq_bams . . . . .	11
.read_somascan . . . . .	14
add_smiles . . . . .	15
analysis . . . . .	16
analyze . . . . .	17
assert_is_valid_sumexp . . . . .	18
AUTONOMICS_DATASETS . . . . .	19
biplot . . . . .	19
center . . . . .	20
contrastdefs . . . . .	21
contrast_subgroup_cols . . . . .	22
counts . . . . .	23
counts2cpm . . . . .	24
counts2tpm . . . . .	24
cpm . . . . .	25
create_design . . . . .	26
create_sfile . . . . .	27
default_sfile . . . . .	27
default_subgroupvar . . . . .	28
download_data . . . . .	29
download_gtf . . . . .	30
dt2mat . . . . .	31
explore_imputations . . . . .	31
explore_transformations . . . . .	32
extract_features . . . . .	33
extract_rectangle . . . . .	34

fdata . . . . .	36
filter_exprs_replicated_in_some_subgroup . . . . .	37
filter_features . . . . .	38
filter_medoid . . . . .	38
filter_replicated . . . . .	39
filter_samples . . . . .	40
fit_limma . . . . .	40
flevels . . . . .	42
fnames . . . . .	43
formula2str . . . . .	44
fvalues . . . . .	44
fvars . . . . .	45
guess_maxquant_quantity . . . . .	46
guess_sep . . . . .	47
impute_systematic_nondetects . . . . .	48
invert . . . . .	49
is_imputed . . . . .	50
is_sig . . . . .	51
limma . . . . .	52
log2counts . . . . .	53
log2countsratios . . . . .	54
log2cpm . . . . .	55
log2cpmratios . . . . .	56
log2tpm . . . . .	57
log2tpmratios . . . . .	58
log2transform . . . . .	59
make_volcano_dt . . . . .	60
matrix2sumexp . . . . .	60
MAXQUANT_PATTERNS_PEPCounts . . . . .	61
MAXQUANT_PATTERNS_QUANTITY . . . . .	62
merge_sdata . . . . .	62
merge_sfile . . . . .	63
message_df . . . . .	65
nfactors . . . . .	65
normimpute . . . . .	66
occupancies . . . . .	67
pca . . . . .	68
plot_boxplots . . . . .	69
plot_contrastogram . . . . .	71
plot_corrections . . . . .	72
plot_covariates . . . . .	73
plot_data . . . . .	74
plot_densities . . . . .	75
plot_detects . . . . .	76
plot_features . . . . .	78
plot_venn . . . . .	79
plot_violins . . . . .	80
plot_volcano . . . . .	81

preprocess_rnaseq_counts . . . . .	82
proteingroups . . . . .	83
read_affymetrix . . . . .	84
rm_singleton_samples . . . . .	85
scaledlibsizes . . . . .	86
sdata . . . . .	86
slevels . . . . .	87
snames . . . . .	88
split_by_svar . . . . .	89
standardize_maxquant_snames . . . . .	89
subgroup_array . . . . .	91
subtract_baseline . . . . .	91
sumexp2mae . . . . .	93
sumexp_to_wide_dt . . . . .	94
summarize_fit . . . . .	95
svalues . . . . .	96
svars . . . . .	97
TESTS . . . . .	98
tpm . . . . .	98
values . . . . .	99
venn_detects . . . . .	100
weights . . . . .	100
zero_to_na . . . . .	101

**Index****103**


---

<i>.read_maxquant</i>	<i>Read/Analyze proteingroups/phosphosites</i>
-----------------------	--

---

**Description**

Read/Analyze proteingroups/phosphosites

**Usage**

```
.read_maxquant(
  file,
  quantity = guess_maxquant_quantity(file),
  sfile = NULL,
  sfileby = NULL,
  subgroupvar = "subgroup",
  select_subgroups = NULL,
  invert_subgroups = character(0),
  pepcountpattern = MAXQUANT_PATTERNS_PEP COUNTS[1],
  verbose = TRUE
)

read_proteingroups(
```

```
file,  
quantity = guess_maxquant_quantity(file),  
sfile = NULL,  
sfileby = NULL,  
select_subgroups = NULL,  
contaminants = FALSE,  
reverse = FALSE,  
fastafilename = NULL,  
invert_subgroups = character(0),  
impute = stri_detect_regex(quantity, "[Ii]ntensity"),  
pepcountpattern = MAXQUANT_PATTERNS_PEP_COUNTS[1],  
subgroupvar = NULL,  
formula = NULL,  
block = NULL,  
contrastdefs = NULL,  
pca = FALSE,  
fit = NULL,  
verbose = TRUE,  
plot = TRUE  
)  
  
read_phosphosites(  
file,  
proteinfile = paste0(dirname(file), "/proteinGroups.txt"),  
quantity = guess_maxquant_quantity(file),  
sfile = NULL,  
sfileby = NULL,  
select_subgroups = NULL,  
contaminants = FALSE,  
reverse = FALSE,  
min_localization_prob = 0.75,  
fastafilename = NULL,  
invert_subgroups = character(0),  
pca = FALSE,  
fit = NULL,  
subgroupvar = NULL,  
formula = NULL,  
block = NULL,  
contrastdefs = NULL,  
verbose = TRUE,  
plot = TRUE  
)
```

### Arguments

file	proteinGroups/phosphosites file
quantity	string: "Ratio normalized", "Ratio", "LFQ intensity", "Reporter intensity corrected", "Reporter intensity", "Intensity labeled", "Intensity"

sfile	sample file
sfileby	sample file mergeby column
subgroupvar	subgroup svar
select_subgroups	subgroups to be selected (character vector)
invert_subgroups	subgroups to be inverted (character vector)
pepcountpattern	value in MAXQUANT_PATTERNS_PEPCOUNTS
verbose	whether to message
contaminants	whether to return contaminants
reverse	whether to return reverse peptides
fastafilename	NULL or fastafilename (to deconvolute proteingroups)
impute	whether to impute consistent nondetects
formula	designmat formula
block	block svar
contrastdefs	contrastdef vector/matrix/list
pca	whether to pca
fit	fit model: NULL, 'limma', 'lm', 'lme', 'lmer', 'wilcoxon'
plot	whether to plot
proteinfile	proteingroups file
min_localization_prob	min site localization probability (number)

**Value**

SummarizedExperiment

**Examples**

```
file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, pca=TRUE, fit='limma')
```

---

*.read\_metabolon**Read metabolon*

---

**Description**

Read metabolon

### Usage

```
.read_metabolon(  
  file,  
  sheet = "OrigScale",  
  fid_var = "(COMP|COMP_ID)",  
  sid_var = "(CLIENT_IDENTIFIER|Client ID)",  
  sfile = NULL,  
  sfileby = NULL,  
  by = NULL,  
  subgroupvar = "Group"  
)
```

```
read_metabolon(  
  file,  
  sheet = "OrigScale",  
  fid_var = "(COMP|COMP_ID)",  
  sid_var = "(CLIENT_IDENTIFIER|Client ID)",  
  sfile = NULL,  
  sfileby = NULL,  
  by = NULL,  
  subgroupvar = "Group",  
  fname_var = "BIOCHEMICAL",  
  impute = FALSE,  
  add_kegg_pathways = FALSE,  
  add_smiles = FALSE,  
  pca = FALSE,  
  fit = NULL,  
  formula = NULL,  
  block = NULL,  
  contrastdefs = NULL,  
  verbose = TRUE,  
  plot = TRUE  
)
```

### Arguments

file	metabolon xlsx filepath
sheet	xls sheet number or name
fid_var	feature_id fvar
sid_var	sampleid svar
sfile	sample file
sfileby	sample file mergeby column
by	metabolon file mergeby column
subgroupvar	subgroup svar
fname_var	featurename fvar
impute	whether to impute

add_kegg_pathways	whether to add kegg pathways
add_smiles	whether to add smiles
pca	whether to pca
fit	fit model: NULL, 'limma', 'lm', 'lme', 'lmer', 'wilcoxon'
formula	designmat formula
block	block svar
contrastdefs	contrastdef vector/matrix/list
verbose	whether to msg
plot	whether to plot

**Value**

SummarizedExperiment

**Examples**

```
file <- download_data('atkin18.metabolon.xlsx')
read_metabolon(file, pca = TRUE, fit = 'limma', block='SUB')
```

---

.read_rectangles	<i>Read omics data from rectangular file</i>
------------------	--

---

**Description**

Read omics data from rectangular file

**Usage**

```
.read_rectangles(
  file,
  sheet = 1,
  fid_rows,
  fid_cols,
  sid_rows,
  sid_cols,
  expr_rows,
  expr_cols,
  fvar_rows = NULL,
  fvar_cols = NULL,
  svar_rows = NULL,
  svar_cols = NULL,
  fdata_rows = NULL,
  fdata_cols = NULL,
  sdata_rows = NULL,
```



```
    sdata_cols = NULL,  
    transpose = FALSE,  
    verbose = TRUE  
  )  
  
read_rectangles(  
  file,  
  sheet = 1,  
  fid_rows,  
  fid_cols,  
  sid_rows,  
  sid_cols,  
  expr_rows,  
  expr_cols,  
  fvar_rows = NULL,  
  fvar_cols = NULL,  
  svar_rows = NULL,  
  svar_cols = NULL,  
  fdata_rows = NULL,  
  fdata_cols = NULL,  
  sdata_rows = NULL,  
  sdata_cols = NULL,  
  transpose = FALSE,  
  sfile = NULL,  
  sfileby = NULL,  
  subgroupvar = character(0),  
  verbose = TRUE  
)
```

**Arguments**

file	string: name of text (txt, csv, tsv, adat) or excel (xls, xlsx) file
sheet	integer/string: only relevant for excel files
fid_rows	numeric vector: featureid rows
fid_cols	numeric vector: featureid cols
sid_rows	numeric vector: sampleid rows
sid_cols	numeric vector: sampleid cols
expr_rows	numeric vector: expr rows
expr_cols	numeric vector: expr cols
fvar_rows	numeric vector: fvar rows
fvar_cols	numeric vector: fvar cols
svar_rows	numeric vector: svar rows
svar_cols	numeric vector: svar cols
fdata_rows	numeric vector: fdata rows
fdata_cols	numeric vector: fdata cols

<code>sdata_rows</code>	numeric vector: sdata rows
<code>sdata_cols</code>	numeric vector: sdata cols
<code>transpose</code>	TRUE or FALSE (default)
<code>verbose</code>	TRUE (default) or FALSE
<code>sfile</code>	sample file
<code>sfileby</code>	sample file mergeby column
<code>subgroupvar</code>	subgroupvar in sfile

**Value**

SummarizedExperiment

**Examples**

```
# RNASEQ
file <- download_data('billing16.rnacounts.txt')
read_rectangles(file, fid_rows = 2:58736, fid_cols = 1,
                sid_rows = 1, sid_cols = 4:14,
                expr_rows = 2:58736, expr_cols = 4:14,
                fvar_rows = 1, fvar_cols = 1:3,
                fdata_rows = 2:58736, fdata_cols = 1:3,
                transpose = FALSE)

# LCMSMS PROTEINGROUPS
file <- download_data('billing19.proteingroups.txt')
read_rectangles(file, fid_rows = 2:9044, fid_cols = 383,
                sid_rows = 1, sid_cols = seq(124, 316, by = 6),
                expr_rows = 2:9044, expr_cols = seq(124, 316, by = 6),
                fvar_rows = 1, fvar_cols = c(2, 6, 7, 383),
                fdata_rows = 2:9044, fdata_cols = c(2, 6, 7, 383),
                transpose = FALSE)

# SOMASCAN
file <- download_data('billing16.somascan.adat')
read_rectangles(file, fid_rows = 21, fid_cols = 19:1146,
                sid_rows = 30:40, sid_cols = 4,
                expr_rows = 30:40, expr_cols = 19:1146,
                fvar_rows = 21:28, fvar_cols = 18,
                svar_rows = 29, svar_cols = 1:17,
                fdata_rows = 21:28, fdata_cols = 19:1146,
                sdata_rows = 30:40, sdata_cols = 1:17,
                transpose = TRUE)

# METABOLON
file <- download_data('halama18.metabolon.xlsx')
read_rectangles(file, sheet = 2,
                fid_rows = 11:401, fid_cols = 5,
                sid_rows = 3, sid_cols = 15:86,
                expr_rows = 11:401, expr_cols = 15:86,
                fvar_rows = 10, fvar_cols = 1:14,
                svar_rows = 1:10, svar_cols = 14,
                fdata_rows = 11:401, fdata_cols = 1:14,
                sdata_rows = 1:10, sdata_cols = 15:86,
                transpose = FALSE)
```

---

.read\_rnaseq\_bams      *Read rnaseq*

---

### **Description**

Read/analyze rnaseq counts / bamfiles

### **Usage**

```
.read_rnaseq_bams(  
  dir,  
  paired,  
  genome,  
  nthreads = detectCores(),  
  sfile = NULL,  
  sfileby = NULL,  
  subgroupvar = NULL,  
  ffile = NULL,  
  ffileby = NULL,  
  fnamevar = NULL,  
  verbose = TRUE  
)
```

```
.read_rnaseq_counts(  
  file,  
  fid_col = 1,  
  sfile = NULL,  
  sfileby = NULL,  
  ffile = NULL,  
  ffileby = NULL,  
  subgroupvar = NULL,  
  verbose = TRUE  
)
```

```
read_rnaseq_bams(  
  dir,  
  paired,  
  genome,  
  nthreads = detectCores(),  
  sfile = NULL,  
  sfileby = NULL,  
  subgroupvar = NULL,  
  block = NULL,  
  ffile = NULL,  
  ffileby = NULL,  
  fnamevar = NULL,  
  formula = NULL,
```

```

min_count = 10,
pseudocount = 0.5,
genesize = NULL,
cpm = TRUE,
tmm = cpm,
log2 = TRUE,
pca = FALSE,
fit = NULL,
voom = !is.null(fit),
contrastdefs = NULL,
verbose = TRUE,
plot = TRUE
)

```

```

read_rnaseq_counts(
  file,
  fid_col = 1,
  sfile = NULL,
  sfileby = NULL,
  subgroupvar = NULL,
  block = NULL,
  ffile = NULL,
  ffileby = NULL,
  fnamevar = NULL,
  formula = NULL,
  min_count = 10,
  pseudocount = 0.5,
  genesize = NULL,
  cpm = TRUE,
  tmm = cpm,
  log2 = TRUE,
  pca = FALSE,
  fit = NULL,
  voom = !is.null(fit),
  contrastdefs = NULL,
  verbose = TRUE,
  plot = TRUE
)

```

### Arguments

<code>dir</code>	<code>read_rnaseq_bams</code> : bam/samfile dir
<code>paired</code>	<code>read_rnaseq_bams</code> : whether paired end reads
<code>genome</code>	<code>read_rnaseq_bams</code> : mm10/"hg38"/etc. or GTF file
<code>nthreads</code>	<code>read_rnaseq_bams</code> : nthreads used by Rsubread::featureCounts()
<code>sfile</code>	sample file
<code>sfileby</code>	sample file mergeby column

subgroupvar	subgroup svar
ffile	feature file
ffileby	feature file mergeby column
fnamevar	featurename fvar
verbose	whether to message
file	read_rnaseq_counts: count file
fid_col	featureid fvar
block	block svar
formula	designmat formula
min_count	min feature count required in some samples
pseudocount	added pseudocount to prevent -Inf log2 values
genesize	genesize fvar for tpm
cpm	whether to compute cpm
tmm	whether to tmm-scale library sizes
log2	whether to log2 transform
pca	whether to pca
fit	fit model: NULL, 'limma', 'lm', 'lme', 'lmer', 'wilcoxon'
voom	whether to compute voom precision weights
contrastdefs	contrastdef vector/matrix/list
plot	whether to plot

**Value**

SummarizedExperiment

**Author(s)**

Aditya Bhagwat, Shahina Hayat

**Examples**

```
file <- download_data('billing19.rnacounts.txt')
object <- read_rnaseq_counts(file, pca= TRUE, fit='limma')

# requires Rsubread
# file <- download_data('billing16.bam.zip')
# object <- read_rnaseq_bams(file, paired=TRUE, genome='hg38', pca=TRUE,
#                             fit='limma', plot=TRUE)
```

---

<code>.read_somascan</code>	<i>Read somascan</i>
-----------------------------	----------------------

---

**Description**

Read data from somascan adat file

**Usage**

```
.read_somascan(  
  file,  
  fidvar = "SeqId",  
  sidvar = "SampleId",  
  sfile = NULL,  
  sfileby = NULL,  
  by = NULL,  
  subgroupvar = "SampleGroup"  
)  
  
read_somascan(  
  file,  
  fidvar = "SeqId",  
  sidvar = "SampleId",  
  sfile = NULL,  
  sfileby = NULL,  
  by = NULL,  
  subgroupvar = "SampleGroup",  
  fname_var = "EntrezGeneSymbol",  
  sample_type = "Sample",  
  feature_type = "Protein",  
  sample_quality = c("FLAG", "PASS"),  
  feature_quality = c("FLAG", "PASS"),  
  rm_na_svars = FALSE,  
  rm_single_value_svars = FALSE,  
  pca = FALSE,  
  fit = NULL,  
  formula = NULL,  
  block = NULL,  
  contrastdefs = NULL,  
  verbose = TRUE,  
  plot = TRUE  
)
```

**Arguments**

<code>file</code>	*.adat file path (string)
<code>fidvar</code>	featureid fvar (string)

sidvar	sampleid svar (string)
sfile	sample file
sfileby	sample file mergeby column
by	metabolon file mergeby column
subgroupvar	subgroup svar (string)
fname_var	featurename fvar (string)
sample_type	subset of c('Sample', 'QC', 'Buffer', 'Calibrator')
feature_type	subset of c('Protein', 'Hybridization Control Elution', 'Rat Protein')
sample_quality	subset of c('PASS', 'FLAG', 'FAIL')
feature_quality	subset of c('PASS', 'FLAG', 'FAIL')
rm_na_svars	whether to rm NA svars
rm_single_value_svars	whether to rm single value svars
pca	whether to pca
fit	fit model: NULL, 'limma', 'lm', 'lme', 'lmer', 'wilcoxon'
formula	design formula (using svars)
block	block var
contrastdefs	contrastdef vector/matrix/list
verbose	whether to msg
plot	whether to plot

**Value**

Summarizedexperiment

**Examples**

```
file <- download_data('atkin18.somascan.adat')
read_somascan(file, pca = TRUE, fit = 'limma', block = 'Subject_ID')
```

---

add\_smiles

*Add smiles*

---

**Description**

Add smiles

**Usage**

```
add_smiles(object)
```

**Arguments**

object            character/factor vector with pubchem ids

**Value**

character/factor vector

**References**

<https://pubchemdocs.ncbi.nlm.nih.gov/pug-rest-tutorial>

**Examples**

```
file <- download_data('atkin18.metabolon.xlsx')
object <- read_metabolon(file, plot=FALSE)
add_smiles(object[1:10, ])
```

---

analysis

*Get/set analysis*

---

**Description**

Get/set analysis

**Usage**

```
analysis(object)
```

```
## S4 method for signature 'SummarizedExperiment'
analysis(object)
```

```
analysis(object) <- value
```

```
## S4 replacement method for signature 'SummarizedExperiment,list'
analysis(object) <- value
```

**Arguments**

object            SummarizedExperiment

value            list

**Value**

analysis details (get) or updated object (set)



**Examples**

```
file <- download_data('billing16.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
analysis(object)
```

---

analyze

*Analyze*


---

**Description**

Analyze

**Usage**

```
analyze(
  object,
  pca = FALSE,
  fit = NULL,
  subgroupvar = default_subgroupvar(object),
  formula = default_formula(object, subgroupvar, fit),
  block = NULL,
  weightvar = if ("weights" %in% assayNames(object)) "weights" else NULL,
  contrastdefs = contrast_coefs(object, formula),
  verbose = TRUE,
  plot = TRUE
)
```

**Arguments**

object	SummarizedExperiment
pca	whether to perform pca
fit	NULL, 'limma', 'lm', 'lme', 'lmer', or 'wilcoxon'
subgroupvar	subgroup svar
formula	model formula
block	block svar
weightvar	NULL or name of weight matrix in assays(object)
contrastdefs	contrastdefs vector/matrix/list
verbose	whether to msg
plot	whether to plot

**Value**

SummarizedExperiment

**Examples**

```
require(magrittr)
file <- download_data('atkin18.metabolon.xlsx')
object <- read_metabolon(file, plot=FALSE)
object %<>% analyze(pca=TRUE, subgroupvar = 'Group', fit='limma')
```

---

```
assert_is_valid_sumexp
```

*Assert that x is a valid SummarizedExperiment*

---

**Description**

Assert that x is a valid SummarizedExperiment

Assert that x is a valid SummarizedExperiment

**Usage**

```
assert_is_valid_sumexp(x, .xname = get_name_in_parent(x))
```

```
assert_is_valid_sumexp(x, .xname = get_name_in_parent(x))
```

**Arguments**

x	SummarizedExperiment
.xname	see assertive.base::get_name_in_parent

**Value**

TRUE or FALSE

TRUE or FALSE

**Examples**

```
# VALID
file <- download_data('halama18.metabolon.xlsx')
x <- read_metabolon(file, plot = FALSE)
assert_is_valid_sumexp(x)
# NOT VALID
rownames(SummarizedExperiment::colData(x)) <- NULL
# assert_is_valid_sumexp(x)
# VALID
file <- download_data('halama18.metabolon.xlsx')
x <- read_metabolon(file, plot = FALSE)
assert_is_valid_sumexp(x)
# NOT VALID
rownames(SummarizedExperiment::colData(x)) <- NULL
# assert_is_valid_sumexp(x)
```

---

AUTONOMICS\_DATASETS    *Data used in examples/vignette/tests/longtests*

---

**Description**

Data used in examples/vignette/tests/longtests

**Usage**

AUTONOMICS\_DATASETS

**Format**

An object of class character of length 12.

**Examples**

AUTONOMICS\_DATASETS

---

biplot                      *Biplot*

---

**Description**

Biplot

**Usage**

```
biplot(  
  object,  
  x = pca1,  
  y = pca2,  
  color = NULL,  
  group = NULL,  
  label = NULL,  
  feature_label = feature_name,  
  ...,  
  fixed = list(shape = 15, size = 3),  
  nloadings = 0  
)  
  
plot_biplot(...)
```

**Arguments**

object	SummarizedExperiment
x	pca1, etc.
y	pca2, etc.
color	svar mapped to color (symbol)
group	svar mapped to group
label	svar mapped to label (symbol)
feature_label	fvar mapped to (loadings) label
...	additional svars mapped to aesthetics
fixed	fixed plot aesthetics
nloadings	number of loadings per half-axis to plot

**Value**

ggplot object

**Examples**

```
require(magrittr)
file <- download_data('atkin18.metabolon.xlsx')
object <- read_metabolon(file, plot = FALSE)
object %<>% pca(ndim=4)
biplot(object)
biplot(object, color=SUB, group=SUB)
biplot(object, color=SUB, nloadings=1)
biplot(object, pca3, pca4, color=SUB, nloadings=1)
```

---

center

*Center samples*

---

**Description**

Center samples

**Usage**

```
center(
  object,
  selector = rep(TRUE, nrow(object)) == TRUE,
  fun = "median",
  verbose = TRUE
)
```

**Arguments**

object	SummarizedExperiment
selector	logical vector (length = nrow(object))
fun	aggregation function (string)
verbose	TRUE/FALSE

**Value**

SummarizedExperiment

**Examples**

```
require(magrittr)
require(matrixStats)
file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE, impute=FALSE)
fdata(object)$housekeeping <- FALSE
fdata(object)$housekeeping[order(rowVars(values(object)))[1:100]] <- TRUE
values(object)[, object$subgroup=='Adult'] %<>% add(5)
plot_sample_densities(object)
plot_sample_densities(center(object))
plot_sample_densities(center(object, housekeeping))
```

---

contrastdefs

*Get/set contrastdefs*

---

**Description**

Get/set contrastdefs

**Usage**

```
contrastdefs(object)

## S4 method for signature 'SummarizedExperiment'
contrastdefs(object)

contrastdefs(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,list'
contrastdefs(object) <- value
```

**Arguments**

object	SummarizedExperiment
value	list

**Value**

contrastdefs (get) or SummarizedExperiment (set)

**Examples**

```
file <- download_data('billing16.proteingroups.txt')
inv <- c('EM_E', 'BM_E', 'BM_EM')
object <- read_proteingroups(
  file, invert_subgroups=inv, fit='limma', plot=FALSE)
contrastdefs(object)
```

---

contrast\_subgroup\_cols

*Row/Col contrasts*

---

**Description**

Row/Col contrasts

**Usage**

```
contrast_subgroup_cols(object, subgroupvar)
```

```
contrast_subgroup_rows(object, subgroupvar)
```

**Arguments**

object	SummarizedExperiment
subgroupvar	subgroup svar

**Value**

matrix

**Examples**

```
file <- download_data('halama18.metabolon.xlsx')
object <- read_metabolon(file, plot=FALSE)
subgroup_matrix(object, subgroupvar = 'Group')
contrast_subgroup_cols(object, subgroupvar = 'Group')
contrast_subgroup_rows(object, subgroupvar = 'Group')
```

---

counts	<i>Get/Set counts</i>
--------	-----------------------

---

## Description

Get / Set counts matrix

## Usage

```
counts(object)

## S4 method for signature 'SummarizedExperiment'
counts(object)

counts(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,matrix'
counts(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,numeric'
counts(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,`NULL`'
counts(object) <- value
```

## Arguments

object	SummarizedExperiment
value	count matrix (features x samples)

## Value

count matrix (get) or updated object (set)

## Examples

```
file <- download_data('billing19.rnacounts.txt')
object <- read_rnaseq_counts(file, plot=FALSE)
counts(object) <- values(object)
counts(object)[1:3, 1:3]
```

---

counts2cpm                      *Convert between counts and cpm*

---

### Description

Convert between counts and cpm

### Usage

```
counts2cpm(x, libsize = scaledlibsizes(x))

cpm2counts(x, libsize)
```

### Arguments

x                              count/cpm matrix  
libsize                        (scaled) libsize vector

### Value

cpm/tpm/count matrix

### Examples

```
file <- download_data('billing19.rnacounts.txt')
object <- read_rnaseq_counts(file, cpm=FALSE, log2=FALSE, plot=FALSE)
libsize <- scaledlibsizes(values(object))
tpm <- counts2tpm(counts(object), genesize = 1)
cpm <- counts2cpm(counts(object), libsize)
counts <- cpm2counts(cpm, libsize)
sum(counts(object) - counts)
```

---

counts2tpm                      *counts to tpm*

---

### Description

counts to tpm

### Usage

```
counts2tpm(x, genesize)
```

### Arguments

x                              count matrix  
genesize                        genesize vector (kilobase)



**Value**

tpm matrix

**Examples**

```
file <- download_data('billing19.rnacounts.txt')
object <- read_rnaseq_counts(file, cpm=FALSE, log2=FALSE, plot=FALSE)
counts2tpm(counts(object), genesize=1)[1:3, 1:3]
```

---

cpm

*Get/Set cpm*

---

**Description**

Get / Set cpm matrix

**Usage**

```
cpm(object)

## S4 method for signature 'SummarizedExperiment'
cpm(object)

cpm(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,matrix'
cpm(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,numeric'
cpm(object) <- value
```

**Arguments**

object	SummarizedExperiment
value	cpm matrix (features x samples)

**Value**

cpm matrix (get) or updated object (set)

**Examples**

```
file <- download_data('billing19.rnacounts.txt')
object <- read_rnaseq_counts(file, plot=FALSE)
cpm(object) <- values(object)
cpm(object)[1:3, 1:3]
```

---

create_design	<i>Create design</i>
---------------	----------------------

---

## Description

Create design matrix for statistical analysis

## Usage

```
create_design(  
  object,  
  subgroupvar = if ("subgroup" %in% svars(object)) "subgroup" else NULL,  
  formula = default_formula(object, subgroupvar, fit = "limma"),  
  verbose = TRUE  
)
```

## Arguments

object	SummarizedExperiment
subgroupvar	subgroup svar
formula	formula with svars
verbose	whether to message

## Value

design matrix

## Examples

```
file <- download_data('billing19.rnacounts.txt')  
object <- read_rnaseq_counts(file, plot=FALSE)  
unique(create_design(object))  
  
object$subgroup <- 'billing19'  
unique(create_design(object))  
  
file <- download_data('atkin18.somascan.adat')  
object <- read_somascan(file, plot=FALSE)  
unique(create_design(object))  
create_design(object, formula= ~ 0 + SampleGroup + Sex + T2D + age + bmi)  
object$subgroup <- 'atkin18'  
unique(create_design(object))
```

---

create_sfile	<i>Create sfile</i>
--------------	---------------------

---

**Description**

Create sfile

**Usage**

```
create_sfile(object, sfile, verbose = TRUE)
```

**Arguments**

object	SummarizedExperiment
sfile	sample file
verbose	TRUE/FALSE

**Value**

sample file path

**Examples**

```
file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
create_sfile(object, paste0(tempfile(), '.tsv'))
```

---

default_sfile	<i>Default sfile</i>
---------------	----------------------

---

**Description**

Default sfile

**Usage**

```
default_sfile(file)
```

**Arguments**

file	data file
------	-----------

**Value**

sample file

**Examples**

```
file <- download_data('billing19.proteingroups.txt')
default_sfile(file)
```

---

default\_subgroupvar    *Create default formula*

---

**Description**

Create default formula

**Usage**

```
default_subgroupvar(object)

default_formula(object, subgroupvar = default_subgroupvar(object), fit)
```

**Arguments**

object	SummarizedExperiment
subgroupvar	string
fit	'limma', 'lm', 'lme', 'lmer'

**Value**

formula

**Examples**

```
file <- download_data('atkin18.metabolon.xlsx')
object <- .read_metabolon(file)
default_subgroupvar(object)
default_formula(object, fit = 'limma')
default_formula(object, fit = 'lm')
```

---

download_data	<i>Download autonomics example data</i>
---------------	---

---

**Description**

Download autonomics example data

**Usage**

```
download_data(
  filename,
  url = paste0("https://bitbucket.org/graumannlabtools/autonomics/downloads/",
    filename),
  verbose = TRUE
)
```

**Arguments**

filename	file name
url	web url

- **Billing 2016: stemcell comparison:** E, EM, BM
  - 'billing16.bam.zip'
  - 'billing16.rnacounts.txt'
  - 'billing16.somascan.adat'
  - 'billing16.proteingroups.txt'
- **Atkin 2018: hypoglycemia:** t0, t1, t2, t3
  - 'atkin18.somascan.adat'
  - 'atkin18.metbolon.xlsx'
- **Halama 2018: glutaminase inhibition:** 4 conc, 4 timepoints
  - 'halama18.metabolon.xlsx'
- **Billing 2019: stemcell differentiation:** E00, E01, E02, E05, EM15, EM30, M00
  - 'billing19.rnacounts.txt'
  - 'billing19.proteingroups.txt'
  - 'billing19.phosphosites.txt'
- **Fukuda 2020: zebrafish development:** X30dpt, Adult
  - 'fukuda20.proteingroups.txt'

verbose	TRUE / FALSE
---------	--------------

**Value**

local file path

**Examples**

```

# atkin18 - hypoglycemia - pubmed 30525282
  download_data('atkin18.somascan.adat')      # somascan intensities
  download_data('atkin18.metabolon.xlsx')     # metabolon intensities

# billing16 - stemcell characterization - pubmed 26857143
  download_data('billing16.proteingroups.txt') # proteingroup ratios
  download_data('billing16.somascan.adat')     # somascan intensities
  download_data('billing16.rnacounts.txt')     # rnaseq counts
  download_data('billing16.bam.zip')           # rnaseq alignments

# billing19 - stemcell differentiation - pubmed 31332097
  # download_data('billing19.proteingroups.txt') # proteingroup ratios
  # download_data('billing19.phosphosites.txt')  # phosphosite ratios
  # download_data('billing19.rnacounts.txt')     # rnaseq counts

# fukuda20 - heart regeneration - pubmed PXD016235
  download_data('fukuda20.proteingroups.txt') # proteingroup LFQ

# halama18 - glutaminase inhibition - pubmed 30525282
  download_data('halama18.metabolon.xlsx')     # metabolon intensities

```

---

download\_gtf

*Download GTF file*


---

**Description**

Download GTF file with feature annotations

**Usage**

```

download_gtf(
  organism,
  release = 100,
  gtffile = sprintf("%s/gtf/%s", rappdirs::user_cache_dir(appname = "autonomics"),
    basename(make_gtf_url(organism, release) %>% substr(1, nchar(.) - 3)))
)

```

**Arguments**

organism	'Homo sapiens', 'Mus musculus' or 'Rattus norvegicus'
release	GTF release (number)
gtffile	string: path to local GTF file

**Value**

gtffile path

**Examples**

```
organism <- 'Homo sapiens'  
# download_gtf(organism)
```

---

dt2mat	<i>'data.table' to 'matrix'</i>
--------	---------------------------------

---

**Description**

Convert between 'data.table' and 'matrix'

**Usage**

```
dt2mat(x)  
  
mat2dt(x, idvar)
```

**Arguments**

x	data.table / matrix
idvar	idvar string

**Value**

matrix / data.table

**Examples**

```
x <- data.table::data.table(  
  gene = c('ENSG001', 'ENSG002', 'ENSG003'),  
  sampleA = c(1787, 10, 432),  
  sampleB = c(1143, 3, 268))  
dt2mat(x)  
mat2dt(dt2mat(x), 'gene')
```

---

explore_imputations	<i>Explore imputations</i>
---------------------	----------------------------

---

**Description**

Explore imputations

**Usage**

```
explore_imputations(object, subgroup, xbiplot = pca1, ybiplot = pca2, ...)
```

**Arguments**

object	SummarizedExperiment
subgroup	subgroup (sym)
xbiplot	biplot x axis. Default pca1 (symbol)
ybiplot	biplot y axis. Default pca2 (symbol)
...	aesthetic mappings

**Value**

ggplot object

**Examples**

```
file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, impute = FALSE, pca = TRUE, plot = FALSE)
explore_imputations(object, subgroup=subgroup)
explore_transformations(object, subgroup=subgroup)
```

---

explore\_transformations

*Explore transformations*

---

**Description**

Explore transformations

**Usage**

```
explore_transformations(
  object,
  subgroup = subgroup,
  transformations = c("quantnorm", "zscore", "invnorm"),
  method = "pca",
  xdim = 1,
  ydim = 2,
  ...
)
```

**Arguments**

object	SummarizedExperiment
subgroup	subgroup (sym)
transformations	vector
method	'pca', 'pls', 'sma', or 'lda'



xdim	number (default 1)
ydim	number (default 2)
...	passed to plot_data

**Value**

grid object

**Examples**

```
file <- download_data('billing16.proteingroups.txt')
invert <- c('EM_E', 'EM_BM', 'BM_E')
object <- read_proteingroups(file, invert_subgroups = invert, plot=FALSE)
explore_transformations(object)
```

---

extract_features	<i>Extract features</i>
------------------	-------------------------

---

**Description**

Extract features

**Usage**

```
extract_features(object, extractor)
```

**Arguments**

object	SummarizedExperiment
extractor	logical/numeric vector

**Value**

SummarizedExperiment

**Examples**

```
require(magrittr)
file <- download_data('billing16.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
(object %<>% extract_features(c(5,4)))
```

---

extract\_rectangle      *Extract rectangle from omics file, data.table, or matrix*

---

### Description

Extract rectangle from omics file, data.table, or matrix

### Usage

```
extract_rectangle(x, ...)  
  
## S3 method for class 'character'  
extract_rectangle(  
  x,  
  rows = seq_len(nrows(x, sheet = sheet)),  
  cols = seq_len(ncols(x, sheet = sheet)),  
  verbose = FALSE,  
  transpose = FALSE,  
  drop = FALSE,  
  sheet = 1,  
  ...  
)  
  
## S3 method for class 'data.table'  
extract_rectangle(  
  x,  
  rows = seq_len(nrow(x)),  
  cols = seq_len(ncol(x)),  
  transpose = FALSE,  
  drop = FALSE,  
  ...  
)  
  
## S3 method for class 'matrix'  
extract_rectangle(  
  x,  
  rows = seq_len(nrow(x)),  
  cols = seq_len(ncol(x)),  
  transpose = FALSE,  
  drop = FALSE,  
  ...  
)
```

### Arguments

x	omics datafile or datatable
...	allow for S3 method dispatch

rows	numeric vector
cols	numeric vector
verbose	logical
transpose	logical
drop	logical
sheet	numeric or string

**Value**

matrix

**Examples**

```
# FROM FILE: extract_rectangle.character
#=====
# exprs
  require(magrittr)
  x <- download_data('halama18.metabolon.xlsx')
  extract_rectangle(x, rows = 11:401, cols = 15:86, sheet = 2) %>%
  extract(1:3, 1:3)

# fids
  extract_rectangle(x, rows = 11:401, cols = 5, sheet = 2) %>%
  extract(1:3, )

# sids
  extract_rectangle(x, rows = 2, cols = 15:86, sheet = 2) %>%
  extract(,1:3)

# fdata
  extract_rectangle(x, rows = 10:401, cols = 1:14, sheet = 2) %>%
  extract(1:3, 1:3)

# sdata
  extract_rectangle(x, rows = 1:10, cols = 14:86, sheet = 2,
  transpose = TRUE) %>% extract(1:3, 1:3)

# FROM MATRIX: extract_rectangle.matrix
#=====
# exprs
  x <-download_data('halama18.metabolon.xlsx') %>%
  extract_rectangle(sheet = 2)
  extract_rectangle(x, rows = 11:401, cols = 15:86, sheet = 2) %>%
  extract(1:3, 1:3)

# fids
  extract_rectangle(x, rows = 11:401, cols = 5, sheet = 2) %>%
  extract(1:3, )

# sids
  extract_rectangle(x, rows = 2, cols = 15:86, sheet = 2) %>%
```

```

extract(,1:3)

# fdata
extract_rectangle(x, rows = 10:401, cols = 1:14, sheet = 2) %>%
extract(1:3, 1:3)

# sdata
extract_rectangle(x, rows = 1:10, cols = 14:86, sheet = 2,
transpose = TRUE) %>% extract(1:3, 1:3)

```

---

fdata

*Get/Set fdata*


---

## Description

Get/Set feature data

## Usage

```

fdata(object)

## S4 method for signature 'SummarizedExperiment'
fdata(object)

fdata(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,data.frame'
fdata(object) <- value

```

## Arguments

object	SummarizedExperiment, eSet, or EList
value	data.frame

## Value

feature dataframe (get) or updated object (set)

## Examples

```

require(magrittr)
file <- download_data('billing16.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
head(fdata(object)) # Getter
fdata(object) %<>% cbind(z=1)
head(fdata(object)) # Setter

```

---

filter\_exprs\_replicated\_in\_some\_subgroup  
*Filter features with replicated expression in some subgroup*

---

## Description

Filter features with replicated expression in some subgroup

## Usage

```
filter_exprs_replicated_in_some_subgroup(  
  object,  
  subgroupvar = "subgroup",  
  comparator = if (contains_ratios(object)) "!=" else ">",  
  lod = 0,  
  verbose = TRUE  
)
```

## Arguments

object	SummarizedExperiment
subgroupvar	subgroup svar
comparator	'>' or '!='
lod	number: limit of detection
verbose	TRUE or FALSE

## Value

Filtered SummarizedExperiment

## Examples

```
require(magrittr)  
file <- download_data('atkin18.metabolon.xlsx')  
object <- read_metabolon(file, plot=FALSE)  
object %<>% filter_exprs_replicated_in_some_subgroup(subgroupvar = 'Group')  
filter_exprs_replicated_in_some_subgroup(object, character(0))  
filter_exprs_replicated_in_some_subgroup(object, NULL)
```

---

filter_features	<i>Filter features on condition</i>
-----------------	-------------------------------------

---

**Description**

Filter features on condition

**Usage**

```
filter_features(object, condition, verbose = FALSE)
```

**Arguments**

object	SummarizedExperiment
condition	filter condition
verbose	logical

**Value**

filtered eSet

**Examples**

```
file <- download_data('atkin18.metabolon.xlsx')
object <- read_metabolon(file, plot=FALSE)
filter_features(object, SUPER_PATHWAY=='Lipid', verbose = TRUE)
```

---

filter_medoid	<i>Filter medoid sample</i>
---------------	-----------------------------

---

**Description**

Filter medoid sample

**Usage**

```
filter_medoid(object, by = NULL, verbose = FALSE)
```

**Arguments**

object	SummarizedExperiment
by	svar
verbose	whether to message

**Value**

SummarizedExperiment

**Examples**

```
require(magrittr)
file <- download_data('billing19.rnacounts.txt')
object <- read_rnaseq_counts(file, plot=FALSE)
object %<>% filter_medoid(by = 'subgroup', verbose=TRUE)
```

---

filter_replicated	<i>Filter for replicated features</i>
-------------------	---------------------------------------

---

**Description**

Filter for replicated features

**Usage**

```
filter_replicated(object, comparator = `>`, lod = 0, n = 2, verbose = TRUE)
```

**Arguments**

object	SummarizedExperiment
comparator	string
lod	number: limit of detection
n	number: number of replicates above lod
verbose	TRUE/FALSE

**Value**

SummarizedExperiment

**Examples**

```
require(magrittr)
file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
object %<>% filter_replicated()
```

---

filter_samples	<i>Filter samples on condition</i>
----------------	------------------------------------

---

**Description**

Filter samples on condition

**Usage**

```
filter_samples(object, condition, verbose = FALSE, record = TRUE)
```

**Arguments**

object	SummarizedExperiment
condition	filter condition
verbose	TRUE or FALSE (default)
record	TRUE (default) or FALSE

**Value**

filtered SummarizedExperiment

**Examples**

```
file <- download_data('atkin18.metabolon.xlsx')
object <- read_metabolon(file, plot=FALSE)
filter_samples(object, Group != 't0', verbose = TRUE)
```

---

fit_limma	<i>Fit model and test for differential expression</i>
-----------	---

---

**Description**

Fit model and test for differential expression

**Usage**

```
fit_limma(
  object,
  subgroupvar = if ("subgroup" %in% svars(object)) "subgroup" else NULL,
  formula = default_formula(object, subgroupvar, "limma"),
  contrastdefs = contrast_coefs(object, formula),
  block = NULL,
  weightvar = if ("weights" %in% assayNames(object)) "weights" else NULL,
  verbose = TRUE,
```



```
    plot = FALSE
  )

fit_lm(
  object,
  subgroupvar = if ("subgroup" %in% svars(object)) "subgroup" else NULL,
  formula = default_formula(object, subgroupvar, fit = "lm"),
  block = NULL,
  weightvar = if ("weights" %in% assayNames(object)) "weights" else NULL,
  contrastdefs = NULL,
  verbose = TRUE,
  plot = FALSE
)

fit_lme(
  object,
  subgroupvar = if ("subgroup" %in% svars(object)) "subgroup" else NULL,
  formula = default_formula(object, subgroupvar, fit = "lme"),
  block = NULL,
  weightvar = if ("weights" %in% assayNames(object)) "weights" else NULL,
  contrastdefs = NULL,
  verbose = TRUE,
  plot = FALSE
)

fit_lmer(
  object,
  subgroupvar = if ("subgroup" %in% svars(object)) "subgroup" else NULL,
  formula = default_formula(object, subgroupvar, fit = "lmer"),
  block = NULL,
  weightvar = if ("weights" %in% assayNames(object)) "weights" else NULL,
  contrastdefs = NULL,
  verbose = TRUE,
  plot = FALSE
)

fit_wilcoxon(
  object,
  subgroupvar = if ("subgroup" %in% svars(object)) "subgroup" else NULL,
  formula = default_formula(object, subgroupvar, fit = "wilcoxon"),
  contrastdefs = contrast_coefs(object, formula = formula),
  block = NULL,
  weightvar = NULL,
  verbose = TRUE,
  plot = FALSE
)
```

**Arguments**

object	SummarizedExperiment
subgroupvar	subgroup variable
formula	modeling formula
contrastdefs	contrastdef vector / matrix / list <ul style="list-style-type: none"> <li>• <code>c("t1-t0", "t2-t1", "t3-t2")</code></li> <li>• <code>matrix(c("WT.t1-WT.t0", "WT.t2-WT.t1", "WT.t3-WT.t2"), c("KD.t1-KD.t0", "KD.t2-KD.t1", "KD.t3-KD.t2"), nrow=2, byrow=TRUE)</code></li> <li>• <code>list(matrix(c("WT.t1-WT.t0", "WT.t2-WT.t1", "WT.t3-WT.t2"), c("KD.t1-KD.t0", "KD.t2-KD.t1", "KD.t3-KD.t2"), nrow=2, byrow=TRUE), matrix(c("KD.t0-WT.t0", "KD.t1-WT.t1", "KD.t2-WT.t2", "KD.t3-WT.t3"), nrow=1, byrow=TRUE))</code></li> </ul>
block	block svar (or NULL)
weightvar	NULL or name of weight matrix in <code>assays(object)</code>
verbose	whether to msg
plot	whether to plot

**Value**

Updated SummarizedExperiment

**Examples**

```
require(magrittr)
file <- download_data('atkin18.somascan.adat')
object <- read_somascan(file, plot=FALSE)
object %<>% fit_limma(subgroupvar = 'SampleGroup')
object %<>% fit_lm( subgroupvar = 'SampleGroup')
plot_venn(is_sig(object, contrast='t3-t2'))

S4Vectors::metadata(object)$limma <- S4Vectors::metadata(object)$lm <- NULL
object %<>% fit_limma( subgroupvar = 'SampleGroup', block = 'Subject_ID')
object %<>% fit_wilcoxon(subgroupvar = 'SampleGroup', block = 'Subject_ID')
# object %<>% fit_lme( subgroupvar = 'SampleGroup', block = 'Subject_ID')
# object %<>% fit_lmer( subgroupvar = 'SampleGroup', block = 'Subject_ID')
plot_venn(is_sig(object, contrast='t3-t2'))
```

---

flevels

*Get fvar levels*


---

**Description**

Get fvar levels

**Usage**

```
flevels(object, fvar)
```

**Arguments**

object	SummarizedExperiment
fvar	feature variable

**Value**

fvar values

**Examples**

```
file <- download_data('billing16.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
head(flevels(object, 'feature_name'))
```

---

fnames	<i>Get/Set fnames</i>
--------	-----------------------

---

**Description**

Get/Set feature names

**Usage**

```
fnames(object)

## S4 method for signature 'SummarizedExperiment'
fnames(object)

fnames(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,character'
fnames(object) <- value
```

**Arguments**

object	SummarizedExperiment, eSet, or EList
value	character vector with feature names

**Value**

feature name vector (get) or updated object (set)

**Examples**

```
require(magrittr)
file <- download_data('billing16.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
fnames(object) %<>% paste0('PG', .)
object
```

---

formula2str	<i>formula to string</i>
-------------	--------------------------

---

**Description**

formula to string

**Usage**

```
formula2str(formula)
```

**Arguments**

formula	formula
---------	---------

**Value**

string

**Examples**

```
formula2str(~0+subgroup)
```

---

fvalues	<i>Get fvalues</i>
---------	--------------------

---

**Description**

Get fvar values

**Usage**

```
fvalues(object, fvar)
```

**Arguments**

object	SummarizedExperiment
fvar	feature variable

**Value**

fvar values

**Examples**

```
file <- download_data('billing16.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
head(fvalues(object, 'feature_name'))
fvalues(object, NULL)
```

---

fvars

*Get/Set fvars*

---

**Description**

Get/Set feature variables

**Usage**

```
fvars(object)

## S4 method for signature 'SummarizedExperiment'
fvars(object)

fvars(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,character'
fvars(object) <- value
```

**Arguments**

object	SummarizedExperiment
value	character vector with feature variables

**Value**

feature variables vector (get) or updated object (set)

**Examples**

```
require(magrittr)
file <- download_data('billing16.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
fvars(object)[1] %<>% paste0('1')
fvars(object)[1]
```

---

`guess_maxquant_quantity`*Guess maxquant quantity from snames*

---

## Description

character vector, dataframe, or SummarizedExperiment.

## Usage

```
guess_maxquant_quantity(x, ...)
```

```
## S3 method for class 'character'
```

```
guess_maxquant_quantity(x, ...)
```

```
## S3 method for class 'data.frame'
```

```
guess_maxquant_quantity(x, ...)
```

```
## S3 method for class 'SummarizedExperiment'
```

```
guess_maxquant_quantity(x, ...)
```

## Arguments

`x` character vector, dataframe, or SummarizedExperiment

`...` used for proper S3 method dispatch

## Value

string: value from names(MAXQUANT\_PATTERNS\_QUANTITY)

## Examples

```
# file
file <- download_data('fukuda20.proteingroups.txt')
guess_maxquant_quantity(file)

# character vector
x <- "Ratio M/L normalized STD(L)_E00(M)_E01(H)_R1"
guess_maxquant_quantity(x)

x <- "Ratio M/L STD(L)_E00(M)_E01(H)_R1"
guess_maxquant_quantity(x)

x <- "LFQ intensity E00.R1"
guess_maxquant_quantity(x)

x <- "Reporter intensity corrected 0 STD(0)E00(1)E01(2)_R1"
guess_maxquant_quantity(x)
```

```

x <- "Reporter intensity 0 STD(0)E00(1)E01(2)_R1"
guess_maxquant_quantity(x)

x <- "Intensity H STD(L)_E00(M)_E01(H)_R1"
guess_maxquant_quantity(x)

# dataframe
file <- download_data('fukuda20.proteingroups.txt')
x <- data.table::fread(file)
guess_maxquant_quantity(x)

# SummarizedExperiment
file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
guess_maxquant_quantity(file)

```

---

guess\_sep

*Guess separator*

---

## Description

Guess separator

## Usage

```

guess_sep(x, ...)

## S3 method for class 'character'
guess_sep(x, separators = c(".", "_"), verbose = FALSE, ...)

## S3 method for class 'factor'
guess_sep(x, ...)

## S3 method for class 'SummarizedExperiment'
guess_sep(x, var = "sample_id", separators = c(".", "_"), verbose = FALSE, ...)

```

## Arguments

x	character vector or SummarizedExperiment
...	used for proper S3 method dispatch
separators	character vector: possible separators to look for
verbose	TRUE or FALSE
var	svar or fvar

## Value

separator (string) or NULL (if no separator could be identified)

**Examples**

```
# charactervector
x <- c('PERM_NON.R1[H/L]', 'PERM_NON.R2[H/L]', 'PERM_NON.R3[H/L]')
guess_sep(x)

x <- c('WT untreated 1', 'WT untreated 2', 'WT treated 1')
guess_sep(x)

x <- c('group1', 'group2', 'group3.R1')
guess_sep(x)

# SummarizedExperiment
# file <- download_data('halama18.metabolon.xlsx')
# object <- read_metabolon(file, plot=FALSE)
# guess_sep(object)

# file <- download_data('billing16.proteingroups.txt')
# object <- read_proteingroups(file, plot=FALSE)
# guess_sep(object)
```

---

impute\_systematic\_nondetects

*Impute systematic nondetects*

---

**Description**

Impute systematic nondetects

**Usage**

```
impute_systematic_nondetects(
  object,
  subgroup = subgroup,
  fun = halfnormimpute,
  plot = TRUE,
  verbose = TRUE,
  ...
)
```

**Arguments**

object	SummarizedExperiment
subgroup	subgroup svar
fun	imputation function
plot	TRUE or FALSE
verbose	TRUE or FALSE
...	passed to 'fun'



**Value**

SummarizedExperiment

**Examples**

```
file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, impute = FALSE, plot = FALSE)
impute_systematic_nondetects(object)
```

invert

*Invert***Description**

For character vectors: invert collapsed strings. For SummarizedExperiments: invert expressions , subgroups, and sample ids

**Usage**

```
invert(x, ...)

## S3 method for class 'character'
invert(x, sep = guess_sep(x), ...)

## S3 method for class 'SummarizedExperiment'
invert(
  x,
  subgroups = slevels(x, "subgroup"),
  sep = guess_sep(x, "subgroup"),
  ...
)
```

**Arguments**

x	character vector or SummarizedExperiment
...	to enable S3 method dispatch
sep	string: collapsed string separator
subgroups	character vector: subgroup levels to be inverted

**Value**

character vector or SummarizedExperiment

**Examples**

```
# character
x <- c('Ctrl_A', 'Ctrl_B')
invert(x)

# SummarizedExperiment
file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
invert(object)
```

---

is_imputed	<i>Get/set is_imputed</i>
------------	---------------------------

---

**Description**

Get/Set is\_imputed

**Usage**

```
is_imputed(object)

## S4 method for signature 'SummarizedExperiment'
is_imputed(object)

is_imputed(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,matrix'
is_imputed(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,`NULL`'
is_imputed(object) <- value
```

**Arguments**

object	SummarizedExperiment
value	matrix

**Value**

matrix (get) or updated object (set)

**Examples**

```
file <- download_data('billing16.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
sum(is_imputed(object))
```

---

is_sig	<i>Is significant?</i>
--------	------------------------

---

## Description

Is significant?

## Usage

```
is_sig(  
  object,  
  fit = intersect(names(metadata(object)), TESTS),  
  contrast = if (is_scalar(fit)) colnames(metadata(object)[[fit]]) else 1,  
  quantity = "fdr"  
)
```

## Arguments

object	SummarizedExperiment
fit	subset of autonomics::TESTS
contrast	subset of colnames(metadata(object)[[fit]])
quantity	value in dimnames(metadata(object)[[fit]])[3]

## Value

matrix: -1 (downregulated), +1 (upregulatd), 0 (not fdr significant)

## Examples

```
require(magrittr)  
file <- download_data('fukuda20.proteingroups.txt')  
object <- read_proteingroups(file, plot=FALSE)  
object %<>% fit_lm()  
object %<>% fit_limma()  
issig <- is_sig(object, fit = c('lm', 'limma'), contrast = 'Adult-X30dpt')  
plot_venn(issig)
```

---

limma	<i>Get/set limma results</i>
-------	------------------------------

---

## Description

Get/Set limma results

## Usage

```
limma(object)

## S4 method for signature 'SummarizedExperiment'
limma(object)

limma(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,array'
limma(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,`NULL`'
limma(object) <- value
```

## Arguments

object	SummarizedExperiment
value	list

## Value

limma results (get) or updated object (set)

## Examples

```
file <- download_data('billing16.proteingroups.txt')
inv <- c('EM_E', 'BM_E', 'BM_EM')
object <- read_proteingroups(
  file, invert_subgroups=inv, fit='limma', plot=FALSE)
dim(limma(object))
dim(limma(object[1:5, ]))
```

---

log2counts	<i>Get/Set log2counts</i>
------------	---------------------------

---

**Description**

Get / Set log2counts matrix

**Usage**

```
log2counts(object)

## S4 method for signature 'SummarizedExperiment'
log2counts(object)

log2counts(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,matrix'
log2counts(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,numeric'
log2counts(object) <- value
```

**Arguments**

object	SummarizedExperiment
value	log2count matrix (features x samples)

**Value**

log2count matrix (get) or updated object (set)

**Examples**

```
file <- download_data('billing19.rnacounts.txt')
object <- read_rnaseq_counts(file, plot=FALSE)
log2counts(object) <- values(object)
log2counts(object)[1:3, 1:3]
```

---

log2countsratios	<i>Get/Set log2countsratios</i>
------------------	---------------------------------

---

### Description

Get / Set log2countsratios matrix

### Usage

```
log2countsratios(object)

## S4 method for signature 'SummarizedExperiment'
log2countsratios(object)

log2countsratios(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,matrix'
log2countsratios(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,numeric'
log2countsratios(object) <- value
```

### Arguments

object	SummarizedExperiment
value	log2countsratios matrix (features x samples)

### Value

log2countsratios matrix (get) or updated object (set)

### Examples

```
file <- download_data('billing19.rnacounts.txt')
object <- read_rnaseq_counts(file, plot=FALSE)
log2countsratios(object) <- values(object)
log2countsratios(object)[1:3, 1:3]
```

---

log2cpm	<i>Get/Set log2cpm</i>
---------	------------------------

---

**Description**

Get / Set log2cpm matrix

**Usage**

```
log2cpm(object)

## S4 method for signature 'SummarizedExperiment'
log2cpm(object)

log2cpm(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,matrix'
log2cpm(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,numeric'
log2cpm(object) <- value
```

**Arguments**

object	SummarizedExperiment
value	log2cpm matrix (features x samples)

**Value**

log2cpm matrix (get) or updated object (set)

**Examples**

```
file <- download_data('billing19.rnacounts.txt')
object <- read_rnaseq_counts(file, plot=FALSE)
log2cpm(object) <- values(object)
log2cpm(object)[1:3, 1:3]
```

---

log2cpmratios	<i>Get/Set log2cpmratios</i>
---------------	------------------------------

---

**Description**

Get / Set log2cpmratios matrix

**Usage**

```
log2cpmratios(object)

## S4 method for signature 'SummarizedExperiment'
log2cpmratios(object)

log2cpmratios(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,matrix'
log2cpmratios(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,numeric'
log2cpmratios(object) <- value
```

**Arguments**

object	SummarizedExperiment
value	log2cpmratios matrix (features x samples)

**Value**

log2cpmratios matrix (get) or updated object (set)

**Examples**

```
file <- download_data('billing19.rnacounts.txt')
object <- read_rnaseq_counts(file, plot=FALSE)
log2cpmratios(object) <- values(object)
log2cpmratios(object)[1:3, 1:3]
```



---

log2tpm	<i>Get/Set log2tpm</i>
---------	------------------------

---

**Description**

Get / Set log2tpm matrix

**Usage**

```
log2tpm(object)

## S4 method for signature 'SummarizedExperiment'
log2tpm(object)

log2tpm(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,matrix'
log2tpm(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,numeric'
log2tpm(object) <- value
```

**Arguments**

object	SummarizedExperiment
value	log2tpm matrix (features x samples)

**Value**

log2tpm matrix (get) or updated object (set)

**Examples**

```
file <- download_data('billing19.rnacounts.txt')
object <- read_rnaseq_counts(file, plot=FALSE)
log2tpm(object) <- values(object)
log2tpm(object)[1:3, 1:3]
```

---

log2tpmratios	<i>Get/Set log2tpmratios</i>
---------------	------------------------------

---

**Description**

Get / Set log2tpmratios matrix

**Usage**

```
log2tpmratios(object)

## S4 method for signature 'SummarizedExperiment'
log2tpmratios(object)

log2tpmratios(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,matrix'
log2tpmratios(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,numeric'
log2tpmratios(object) <- value
```

**Arguments**

object	SummarizedExperiment
value	log2tpmratios matrix (features x samples)

**Value**

log2tpmratios matrix (get) or updated object (set)

**Examples**

```
file <- download_data('billing19.rnacounts.txt')
object <- read_rnaseq_counts(file, plot=FALSE)
log2tpmratios(object) <- values(object)
log2tpmratios(object)[1:3, 1:3]
```

---

log2transform	<i>Transform values</i>
---------------	-------------------------

---

**Description**

Transform values

**Usage**

```
log2transform(object, verbose = FALSE)
exp2(object, verbose = FALSE)
zscore(object, verbose = FALSE)
quantnorm(object, verbose = FALSE)
invnorm(object, verbose = FALSE)
```

**Arguments**

object	SummarizedExperiment
verbose	TRUE or FALSE

**Value**

Transformed sumexp

**Examples**

```
require(magrittr)
file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE, impute=FALSE)

object %>% plot_sample_densities()
invnorm(object) %>% plot_sample_densities()

object %>% plot_sample_densities()
quantnorm(object) %>% plot_sample_densities()

object %>% plot_sample_densities()
zscore(object) %>% plot_sample_densities()

object %>% plot_sample_densities()
exp2(object) %>% plot_sample_densities()
log2transform(exp2(object)) %>% plot_sample_densities()
```

---

make_volcano_dt	<i>Create volcano datatable</i>
-----------------	---------------------------------

---

**Description**

Create volcano datatable

**Usage**

```
make_volcano_dt(  
  object,  
  fit,  
  contrastdefmat = contrastdefs(object)[[1]],  
  ntop = 3  
)
```

**Arguments**

object	SummarizedExperiment
fit	'limma', 'lme', 'lm', 'wilcoxon'
contrastdefmat	contrastdef matrix
ntop	no of top features to be annotated

**Value**

data.table

**Examples**

```
file <- download_data('fukuda20.proteingroups.txt')  
object <- read_proteingroups(file, fit='limma', plot=FALSE)  
make_volcano_dt(object, fit = 'limma')
```

---

matrix2sumexp	<i>Convert matrix into SummarizedExperiment</i>
---------------	---

---

**Description**

Convert matrix into SummarizedExperiment

**Usage**

```
matrix2sumexp(  
  x,  
  sdt = NULL,  
  sdtby = if (is.null(sdt)) NULL else names(sdt)[1],  
  subgroupvar = NULL,  
  fdt = NULL,  
  fdtby = if (is.null(fdt)) NULL else names(fdt)[1],  
  fnamevar = NULL,  
  verbose = TRUE  
)
```

**Arguments**

x	matrix
sdt	sample data.table / data.frame / DataFrame
sdtby	sample data mergeby column
subgroupvar	string / NULL
fdt	feature data.table / data.frame / DataFrame
fdtby	feature data mergeby column
fnamevar	string / NULL
verbose	TRUE/FALSE

**Value**

SummarizedExperiment

**Examples**

```
require(magrittr)  
file <- download_data('atkin18.metabolon.xlsx')  
x <- values(read_metabolon(file, plot=FALSE))  
object <- matrix2sumexp(x)  
object %<>% pca()  
biplot(object, nloadings=0, color=subgroup)
```

---

MAXQUANT\_PATTERNS\_PEP COUNTS

*maxquant peptide count patterns*

---

**Description**

maxquant peptide count patterns

**Usage**

MAXQUANT\_PATTERNS\_PEP COUNTS

**Format**

An object of class character of length 3.

**Examples**

MAXQUANT\_PATTERNS\_PEP COUNTS

---

MAXQUANT\_PATTERNS\_QUANTITY  
*maxquant quantity patterns*

---

**Description**

maxquant quantity patterns

**Usage**

MAXQUANT\_PATTERNS\_QUANTITY

**Format**

An object of class character of length 7.

**Examples**

MAXQUANT\_PATTERNS\_QUANTITY

---

merge\_sdata      *Merge sample/feature data*

---

**Description**

Merge sample/feature data

**Usage**

```
merge_sdata(
  object,
  dt,
  by.x = "sample_id",
  by.y = names(dt)[1],
  verbose = TRUE
)

merge_fdata(
  object,
  dt,
  by.x = "feature_id",
  by.y = names(dt)[1],
  verbose = TRUE
)
```

**Arguments**

object	SummarizedExperiment
dt	data.frame, data.table, DataFrame
by.x	object mergevar
by.y	df mergevar
verbose	TRUE/FALSE

**Value**

SummarizedExperiment

**Examples**

```
require(magrittr)
file <- download_data('halama18.metabolon.xlsx')
object <- read_metabolon(file, plot=FALSE)
object %<>% merge_sdata( data.frame(sample_id = object$sample_id,
                                   number = seq_along(object$sample_id)))
head(sdata(object))
```

---

merge\_sfile

*Merge sample/feature file*

---

**Description**

Merge sample/feature file

**Usage**

```
merge_sfile(  
  object,  
  sfile = NULL,  
  by.x = "sample_id",  
  by.y = NULL,  
  stringsAsFactors = TRUE,  
  verbose = TRUE  
)
```

```
merge_ffile(  
  object,  
  ffile = NULL,  
  by.x = "feature_id",  
  by.y = NULL,  
  stringsAsFactors = TRUE,  
  verbose = TRUE  
)
```

**Arguments**

object	SummarizedExperiment
sfile	sample file path
by.x	object mergevar
by.y	file mergevar
stringsAsFactors	TRUE or FALSE
verbose	TRUE (default) or FALSE
ffile	ffile path

**Value**

SummarizedExperiment

**Examples**

```
require(magrittr)  
file <- download_data('billing19.proteingroups.txt')  
select <- c('E00','E01', 'E02','E05','E15','E30', 'M00')  
select %<>% paste0('_STD')  
object <- read_proteingroups(file, select_subgroups = select, plot=FALSE)  
sfile <- paste0(tempdir(), '/', basename(tools::file_path_sans_ext(file)))  
sfile %<>% paste0('.samples.txt')  
invisible(create_sfile(object, sfile))  
merge_sfile(object, sfile)
```



---

message_df	<i>message dataframe</i>
------------	--------------------------

---

**Description**

message dataframe using sprintf syntax. Use place holder `

**Usage**

```
message_df(format_string, x)
```

**Arguments**

format_string	sprintf style format string
x	data.frame

**Value**

nothing returned

**Examples**

```
x <- data.frame(feature_id = c('F001', 'F002'), symbol = c('FEAT1', 'FEAT2'))
message_df('\t%s', x)
```

```
x <- c(rep('PASS', 25), rep('FAIL', 25))
message_df(format_string = '%s', table(x))
```

---

nfactors	<i>stri_split and extract</i>
----------	-------------------------------

---

**Description**

stri\_split and extract

**Usage**

```
nfactors(x, sep = guess_sep(x))
```

```
split_extract(x, i, sep = guess_sep(x))
```

**Arguments**

x	string
sep	string
i	integer

**Value**

character

**Examples**

```
require(magrittr)
file <- download_data('halama18.metabolon.xlsx')
object <- read_metabolon(file, plot=FALSE)
x <- object$sample_id[1:5]
nfactors(x)
split_extract(x, 1:2)
split_extract(x, seq_len(nfactors(x)-1))
split_extract(x, nfactors(x))

# With NA values
split_extract(fdata(object)$PUBCHEM, 1, ';')
```

---

normimpute

*Impute from half-normal distribution around 0*

---

**Description**

Impute from half-normal distribution around 0

**Usage**

```
normimpute(x, selector = is.na(x), mean = 0)
```

```
halfnormimpute(x, selector = is.na(x))
```

```
zeroimpute(x, selector = is.na(x))
```

```
translate(
  x,
  ref = c(min, mean, median, max)[[1]],
  pos = 3 * sd(x, na.rm = TRUE)
)
```

**Arguments**

x	NA-containing numeric vector
selector	which values to impute
mean	which mean to impute around

**Value**

numeric vector of same length

**Examples**

```

require(data.table)
x <- rnorm(1e5)
idx <- runif(length(x))>0.9
x[idx] <- NA
dt1 <- data.table(value = normimpute(x), distr = 'norm')

x <- abs(rnorm(1e5)); x[idx] <- NA
dt2 <- data.table(value = halfnormimpute(x), distr = 'halfnorm')

x <- abs(rnorm(1e5)); x[idx] <- NA
dt3 <- data.table(value = zeroimpute(x), distr = 'zero')

x <- abs(rnorm(1e5)); x[idx] <- NA
dt4 <- data.table(value = translate(x), distr = 'translate')

require(ggplot2)
ggplot(rbind(dt1,dt2,dt3, dt4), aes(x=value, fill=distr)) +
  geom_density(alpha=0.5)

```

---

occupancies

*Get/Set occupancies*


---

**Description**

Get / Set phosphosite occupancies matrix

**Usage**

```

occupancies(object)

## S4 method for signature 'SummarizedExperiment'
occupancies(object)

occupancies(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,matrix'
occupancies(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,numeric'
occupancies(object) <- value

```

**Arguments**

object	SummarizedExperiment
value	occupancy matrix (features x samples)

**Value**

occupancy matrix (get) or updated object (set)

**Examples**

```
file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
occupancies(object)
occupancies(object) <- values(object)
occupancies(object)[1:3, 1:3]
```

---

pca

*Add PCA, SMA, LDA, PLS*

---

**Description**

Perform a dimension reduction. Add sample scores, feature loadings, and dimension variances to object.

**Usage**

```
pca(object, ndim = 2, minvar = 0, verbose = TRUE, plot = FALSE, ...)
```

```
pls(
  object,
  subgroupvar = "subgroup",
  ndim = 2,
  minvar = 0,
  verbose = FALSE,
  plot = FALSE,
  ...
)
```

```
sma(object, ndim = 2, minvar = 0, verbose = TRUE, plot = FALSE, ...)
```

```
lda(
  object,
  subgroupvar = "subgroup",
  ndim = 2,
  minvar = 0,
  verbose = TRUE,
  plot = FALSE,
  ...
)
```

**Arguments**

object	SummarizedExperiment
ndim	number
minvar	number
verbose	TRUE (default) or FALSE
plot	TRUE/FALSE
...	passed to biplot
subgroupvar	subgroup svar

**Value**

SummarizedExperiment

**Author(s)**

Aditya Bhagwat, Laure Cougnard (LDA)

**Examples**

```
file <- download_data('atkin18.metabolon.xlsx')
object <- read_metabolon(file, plot = FALSE)
pca(object, plot=TRUE, color = Group) # Principal Component Analysis
pls(object, subgroupvar = 'Group') # Partial Least Squares
lda(object, subgroupvar = 'Group') # Linear Discriminant Analysis
sma(object) # Spectral Map Analysis
pca(object, ndim=3)
pca(object, ndim=Inf, minvar=5)
```

---

plot\_boxplots                      *Plot sample/feature boxplots*

---

**Description**

Plot sample/feature boxplots

**Usage**

```
plot_boxplots(
  object,
  x,
  fill,
  color = NULL,
  facet = NULL,
  highlight = NULL,
  fixed = list(na.rm = TRUE)
)
```

```
plot_sample_boxplots(  
  object,  
  x = sample_id,  
  fill = sample_id,  
  color = NULL,  
  highlight = NULL,  
  fixed = list(na.rm = TRUE)  
)  
  
plot_feature_boxplots(  
  object,  
  x = feature_id,  
  fill = feature_id,  
  color = NULL,  
  highlight = NULL,  
  fixed = list(na.rm = TRUE)  
)  
  
plot_subgroup_boxplots(  
  object,  
  subgroup,  
  x = !!enquo(subgroup),  
  fill = !!enquo(subgroup),  
  color = NULL,  
  highlight = NULL,  
  facet = feature_id,  
  fixed = list(na.rm = TRUE)  
)
```

### Arguments

object	SummarizedExperiment
x	svar mapped to x
fill	svar mapped to fill
color	svar mapped to color
facet	svar mapped to facet
highlight	fvar expressing which feature should be highlighted
fixed	fixed aesthetics
subgroup	subgroup svar symbol

### Value

ggplot object

### See Also

[plot\\_sample\\_densities](#), [plot\\_sample\\_violins](#)

**Examples**

```

# data
require(magrittr)
file <- download_data('halama18.metabolon.xlsx')
object <- read_metabolon(file, plot = FALSE)
object %<>% extract(, order(.$Group))
fdata(object) %<>% cbind(
  control=.$feature_name %in% c('biotin','phosphate'))

# plot
plot_boxplots(object[1:9,], x = feature_id, fill = feature_id)
plot_boxplots(object[,1:9], x = sample_id, fill = sample_id )
plot_feature_boxplots(object[1:9, ])
plot_sample_boxplots(object[, 1:12])
plot_sample_boxplots(object[, 1:12], highlight = control)
plot_subgroup_boxplots(object[1:2, ], subgroup = Group)

```

---

plot\_contrastogram      *Plot contrastogram*

---

**Description**

Plot contrastogram

**Usage**

```

plot_contrastogram(
  object,
  subgroupvar,
  formula = default_formula(object, subgroupvar, "limma"),
  colors = make_colors(slevels(object, subgroupvar), guess_sep(object)),
  curve = 0.1
)

```

**Arguments**

object	SummarizedExperiment
subgroupvar	subgroup svar
formula	formula
colors	named color vector (names = subgroups)
curve	arrow curvature

**Value**

list returned by [plotmat](#)

## Examples

```
if (requireNamespace('diagram', quietly = TRUE)){
  file <- download_data('halama18.metabolon.xlsx')
  object <- read_metabolon(file, fit='limma', plot=FALSE)
  plot_contrastogram(object, subgroupvar = 'Group')
}
```

---

plot\_corrections      *Biplot batch corrections*

---

## Description

Biplot batch corrections

## Usage

```
plot_corrections(...)
```

```
biplot_corrections(
  object,
  method = "pca",
  color = subgroup,
  covariates = character(0),
  varcols = ceiling(sqrt(1 + length(covariates))),
  plot = TRUE
)
```

## Arguments

...	used to maintain deprecated functions
object	SummarizedExperiment
method	'pca', 'pls', 'lda', or 'sma'
color	variable mapped to color (symbol)
covariates	covariates to be batch-corrected
varcols	number of covariate columns
plot	TRUE/FALSE: plot?

## Value

grid object

## See Also

biplot\_covariates



**Examples**

```
file <- download_data('atkin18.metabolon.xlsx')
object <- read_metabolon(file, pca=TRUE, plot = FALSE)
biplot_corrections(
  object, color = Group, covariates = c('SEX', 'T2D', 'SUB', 'SET'))
```

---

plot_covariates	<i>Biplot covariates</i>
-----------------	--------------------------

---

**Description**

Biplot covariates

**Usage**

```
plot_covariates(...)

biplot_covariates(
  object,
  method = "pca",
  covariates = "subgroup",
  ndim = 6,
  dimcols = 1,
  varcols = length(covariates),
  plot = TRUE
)
```

**Arguments**

...	used to maintain deprecated functions
object	SummarizedExperiment
method	'pca', 'pls', 'lda', or 'sma'
covariates	covariates: mapped to color or batch-corrected
ndim	number of dimensions to plot
dimcols	number of dimension columns
varcols	number of covariate columns
plot	TRUE or FALSE: whether to plot

**Value**

ggplot object

**See Also**

biplot\_corrections

**Examples**

```

file <- download_data('atkin18.metabolon.xlsx')
object <- read_metabolon(file, pca = TRUE, plot = FALSE)
biplot_covariates(object, covariates = 'Group', ndim = 12, dimcols = 3)
biplot_covariates(object, covariates = c('SEX', 'T2D', 'SUB', 'SET'))
biplot_covariates(object, covariates = c('SEX', 'T2D', 'SUB', 'SET'), ndim=2)
biplot_covariates(object, covariates = c('Group'), dimcols = 3)

```

---

plot\_data

*Plot data*


---

**Description**

Plot data

**Usage**

```

plot_data(
  data,
  geom = geom_point,
  color = NULL,
  fill = !enquo(color),
  ...,
  fixed = list(),
  theme = list()
)

```

**Arguments**

data	data.frame'
geom	geom_point, etc.
color	variable mapped to color (symbol)
fill	variable mapped to fill (symbol)
...	mapped aesthetics
fixed	fixed aesthetics (list)
theme	list with ggplot theme specifications

**Value**

ggplot object

**Author(s)**

Aditya Bhagwat, Johannes Graumann

**Examples**

```

require(magrittr)
file <- download_data('halama18.metabolon.xlsx')
object <- read_metabolon(file, plot = FALSE)
object %<>% pca()
data <- sdata(object)
plot_data(data, x = pca1, y = pca2)
plot_data(data, x = pca1, y = pca2, color = TIME_POINT)
data$TIME <- as.numeric(substr(data$TIME_POINT, 2, 3))
plot_data(data, x = pca1, y = pca2, color = TIME)
plot_data(data, x = pca1, y = pca2, color = NULL)

fixed <- list(shape = 15, size = 3)
plot_data(data, x = pca1, y = pca2, fixed=fixed)

```

---

plot\_densities

*Plot sample/feature densities*


---

**Description**

Plot sample/feature densities

**Usage**

```

plot_densities(
  object,
  group,
  fill,
  color = NULL,
  fixed = list(alpha = 0.5, na.rm = TRUE)
)

plot_sample_densities(
  object,
  fill = sample_id,
  color = NULL,
  group = sample_id,
  fixed = list(alpha = 0.5, na.rm = TRUE),
  subsetter = if (ncol(object) < 100) { seq_len(ncol(object)) } else {
    sample(ncol(object), 9) }
)

plot_feature_densities(
  object,
  fill = feature_id,
  color = NULL,
  group = feature_id,

```

```
fixed = list(alpha = 0.5, na.rm = TRUE),
subsetter = if (nrow(object) < 100) { seq_len(nrow(object)) } else {
  sample(nrow(object), 9) }
)
```

### Arguments

object	SummarizedExperiment
group	svar mapped to group
fill	svar mapped to fill
color	svar mapped to color
fixed	fixed aesthetics
subsetter	subsetter for showing a subset of samples/features

### Value

ggplot object

### See Also

[plot\\_sample\\_violins](#), [plot\\_sample\\_boxplots](#)

### Examples

```
# Read data
require(magrittr)
file <- download_data('atkin18.metabolon.xlsx')
object <- read_metabolon(file, plot = FALSE)
object %<>% extract(, order(.$Group))
# Plot distributions
plot_sample_densities(object, fill = Group)
plot_feature_densities(object)
```

---

plot\_detects

*Plot detections*

---

### Description

Plot detections

**Usage**

```

plot_detects(...)

plot_detections(object, subgroup = subgroup, fill = !!enquo(subgroup))

plot_quantifications(...)

plot_summarized_detections(
  object,
  subgroup = subgroup,
  fill = !!enquo(subgroup),
  na_imputes = TRUE
)

```

**Arguments**

...	for backward compatilby
object	SummarizedExperiment
subgroup	subgroup var (sym)
fill	fill var (sym)
na_imputes	whether to NA imputes prior to plottin (TRUE/FALSE)g

**Details**

plot\_detections plots feature x sample detections. It shows per feature/sample nondetects (white), imputes (light colored), and detects (full color).

plot\_summarized\_detections gives an summarized view, plotting featurtype x subgroup detections. It visualizes the subgroup-wise nondetect structure often seen in mass spectrometry proteomics data (across e.g. different cell types)

**Value**

ggplot object

**Examples**

```

require(magrittr)
file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, impute=FALSE, plot = FALSE)
plot_summarized_detections(object)
plot_detections(object)
plot_detections(impute_systematic_nondetects(object, plot=FALSE))

file <- download_data('halama18.metabolon.xlsx')
object <- read_metabolon(file, impute = FALSE, plot = FALSE)
plot_summarized_detections(object, Group)
plot_detections(object, Group)

```

---

plot_features	<i>Plot features</i>
---------------	----------------------

---

## Description

Plot features

## Usage

```
plot_features(  
  object,  
  geom,  
  subgroup,  
  x = !!enquo(subgroup),  
  fill = !!enquo(subgroup),  
  color = !!enquo(subgroup),  
  ...,  
  fixed = list(na.rm = TRUE),  
  theme = list(axis.text.x = element_blank(), axis.title.x = element_blank(),  
               axis.ticks.x = element_blank())  
)  
  
plot_feature_profiles(...)
```

## Arguments

object	SummarizedExperiment
geom	geom_point, geom_boxplot, etc.
subgroup	subgroup svar
x	svar mapped to x
fill	svar mapped to fill
color	svar mapped to color
...	mapped aesthetics
fixed	fixed aesthetics
theme	ggplot theme specifications

## Value

ggplot object

**Examples**

```
require(magrittr)
file <- download_data('halama18.metabolon.xlsx')
object <- read_metabolon(file, pca=TRUE, plot = FALSE)
idx <- order(abs(fdata(object)$pca1), decreasing=TRUE)[1:9]
object %<>% extract(idx, )
plot_feature_boxplots(object)
plot_subgroup_boxplots(object, subgroup=Group)
plot_feature_profiles( object, subgroup=Group)
```

---

plot\_venn

*Plot venn*

---

**Description**

Plot venn

**Usage**

```
plot_venn(isfdr)
```

**Arguments**

isfdr            matrix(nrow, ncontrast): -1 (down), +1 (up)

**Value**

nothing returned

**Examples**

```
require(magrittr)
file <- download_data('atkin18.somascan.adat')
object <- read_somascan(file, plot=FALSE)
object %<>% fit_wilcoxon(subgroupvar='SampleGroup', block = 'Subject_ID')
object %<>% fit_limma(  subgroupvar='SampleGroup', block = 'Subject_ID')
isfdr <- is_sig(object, contrast = 't3-t2')
plot_venn(isfdr)
```

---

plot_violins	<i>Plot sample/feature violins</i>
--------------	------------------------------------

---

**Description**

Plot sample/feature violins

**Usage**

```
plot_violins(  
  object,  
  x,  
  fill,  
  color = NULL,  
  group = NULL,  
  facet = NULL,  
  highlight = NULL,  
  fixed = list(na.rm = TRUE)  
)
```

```
plot_sample_violins(  
  object,  
  x = sample_id,  
  fill = sample_id,  
  color = NULL,  
  highlight = NULL,  
  fixed = list(na.rm = TRUE)  
)
```

```
plot_feature_violins(  
  object,  
  x = feature_id,  
  fill = feature_name,  
  color = NULL,  
  highlight = NULL,  
  fixed = list(na.rm = TRUE)  
)
```

```
plot_subgroup_violins(  
  object,  
  subgroup,  
  x = !!enquo(subgroup),  
  fill = !!enquo(subgroup),  
  color = NULL,  
  highlight = NULL,  
  facet = feature_id,  
  fixed = list(na.rm = TRUE)
```



)

**Arguments**

object	SummarizedExperiment
x	svar mapped to x
fill	svar mapped to fill
color	svar mapped to color
group	svar mapped to group
facet	svar mapped to facets
highlight	fvar expressing which feature should be highlighted
fixed	fixed aesthetics
subgroup	subgroup svar

**Value**

ggplot object

**See Also**[plot\\_sample\\_densities](#), [plot\\_sample\\_boxplots](#)**Examples**

```
# data
require(magrittr)
file <- download_data('halama18.metabolon.xlsx')
object <- read_metabolon(file, plot = FALSE)
object %<>% extract(, order(.$Group))
control_features <- c('biotin', 'phosphate')
fdata(object) %<>% cbind(control=.$feature_name %in% control_features)

# plot
plot_violins(object[1:12, ], x=feature_id, fill=feature_id)
plot_feature_violins(object[1:12, ])
plot_sample_violins(object[, 1:12], highlight = control)
plot_subgroup_violins(object[1:4, ], subgroup = Group)
```

plot\_volcano

*Plot volcano***Description**

Plot volcano

**Usage**

```
plot_volcano(
  object,
  fit = intersect(names(metadata(object)), TESTS)[1],
  contrastdefs = autonomics::contrastdefs(object)[[1]],
  label = feature_name,
  ntop = 1
)
```

**Arguments**

object	SummarizedExperiment
fit	'limma', 'lme', 'lm', 'wilcoxon'
contrastdefs	contrastdef vector / matrix / list
label	fvar for labeling top features
ntop	number: n top features to be annotated

**Value**

ggplot object

**Examples**

```
file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, fit='limma', plot=FALSE)
plot_volcano(object)
```

---

```
preprocess_rnaseq_counts
```

*Preprocess RNAseq counts*

---

**Description**

Preprocess RNAseq counts

**Usage**

```
preprocess_rnaseq_counts(
  object,
  subgroupvar = if ("subgroup" %in% svars(object)) "subgroup" else NULL,
  formula = default_formula(object, subgroupvar, "limma"),
  block = NULL,
  min_count = 10,
  pseudocount = 0.5,
  genesize = NULL,
  cpm = TRUE,
```

```
tmm = cpm,
voom = TRUE,
log2 = TRUE,
verbose = TRUE,
plot = TRUE
)
```

**Arguments**

object	SummarizedExperiment
subgroupvar	subgroup svar
formula	designmat formula
block	block svar
min_count	min count required in some samples
pseudocount	added pseudocount to avoid log(x)=-Inf
genesize	genesize fvar to compute tpm
cpm	whether to compute counts per million (scaled) reads
tmm	whether to tmm normalize
voom	whether to voom weight
log2	whether to log2
verbose	whether to msg
plot	whether to plot

**Value**

SummarizedExperiment

**Examples**

```
require(magrittr)
file <- download_data('billing19.rnacounts.txt')
object <- .read_rnaseq_counts(file)
object$subgroup
object %<>% preprocess_rnaseq_counts()
```

---

proteingroups	<i>Get/Set proteingroups</i>
---------------	------------------------------

---

**Description**

Get / Set proteingroups matrix

**Usage**

```

proteingroups(object)

## S4 method for signature 'SummarizedExperiment'
proteingroups(object)

proteingroups(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,matrix'
proteingroups(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,numeric'
proteingroups(object) <- value

```

**Arguments**

object	SummarizedExperiment
value	occupancy matrix (features x samples)

**Value**

occupancy matrix (get) or updated object (set)

**Examples**

```

file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
proteingroups(object)[1:3, 1:3]

```

---

read_affymetrix	<i>Read affymetrix microarray</i>
-----------------	-----------------------------------

---

**Description**

Read affymetrix microarray

**Usage**

```
read_affymetrix(celfiles)
```

**Arguments**

celfiles	string vector: CEL file paths
----------	-------------------------------

**Value**

RangedSummarizedExperiment

**Examples**

```

require(magrittr)
url <- paste0('http://www.bioconductor.org/help/publications/2003/',
             'Chiaretti/chiaretti2/T33.tgz')
localdir <- file.path(rappdirs::user_cache_dir(appname = 'autonomics'), 'T33')
dir.create(localdir, showWarnings=FALSE)
localfile <- file.path(localdir, basename(url))
if (!file.exists(localfile)){
  download.file(url, destfile = localfile)
  untar(localfile, exdir = path.expand(localdir))
}
localfile %<>% substr(1, nchar(.)-4)
if (!requireNamespace("BiocManager", quietly = TRUE)) install.packages(
  'BiocManager')
if (!requireNamespace("hgu95av2.db", quietly = TRUE)) BiocManager::install(
  'hgu95av2.db')
# read_affymetrix(cefiles = list.files(localfile, full.names = TRUE))
# currently openblas issue: https://stackoverflow.com/questions/61629861/

```

---

rm\_singleton\_samples *Rm singleton samples*

---

**Description**

Rm singleton samples

**Usage**

```
rm_singleton_samples(object, svar = "subgroup", verbose = TRUE)
```

**Arguments**

object	SummarizedExperiment
svar	sample var
verbose	TRUE/FALSE

**Value**

SummarizedExperiment

**Examples**

```

require(magrittr)
file <- download_data('atkin18.somascan.adat')
object <- read_somascan(file, plot=FALSE)
object %<>% filter_samples(SampleGroup %in% c('t1', 't2'), verbose = TRUE)
rm_singleton_samples(object, 'Subject_ID')

```

---

scaledlibsizes	<i>Get tmm-scaled libsizes</i>
----------------	--------------------------------

---

**Description**

Get tmm-scaled libsizes

**Usage**

```
scaledlibsizes(counts)
```

**Arguments**

counts            counts matrix

**Value**

scaled libsize vector

**Examples**

```
file <- download_data('billing19.rnacounts.txt')
object <- read_rnaseq_counts(file, cpm=FALSE, log2=FALSE, plot=FALSE)
scaledlibsizes(counts(object))
```

---

sdata	<i>Get/Set sdata</i>
-------	----------------------

---

**Description**

Get/Set sample data

**Usage**

```
sdata(object)

## S4 method for signature 'SummarizedExperiment'
sdata(object)

## S4 method for signature 'MultiAssayExperiment'
sdata(object)

sdata(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,data.frame'
sdata(object) <- value
```

```
## S4 replacement method for signature 'SummarizedExperiment,DataFrame'
sdata(object) <- value

## S4 replacement method for signature 'MultiAssayExperiment,data.frame'
sdata(object) <- value

## S4 replacement method for signature 'MultiAssayExperiment,DataFrame'
sdata(object) <- value
```

**Arguments**

object	SummarizedExperiment, eSet, or EList
value	dataframe

**Value**

sample dataframe (get) or updated object (set)

**Examples**

```
require(magrittr)
file <- download_data('billing16.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
head(sdata(object))
head(sdata(object) %<>% cbind(z=1))
```

---

slevels

*Get slevels*


---

**Description**

Get svar levels

**Usage**

```
slevels(object, svar)

subgroup_levels(object)
```

**Arguments**

object	SummarizedExperiment, eSet, or eList
svar	sample var (character)

**Value**

svar values (character)

### Examples

```
file <- download_data('billing16.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
slevels(object, 'subgroup')
subgroup_levels(object)
```

---

snames

*Get/Set snames*

---

### Description

Get/Set sample names

### Usage

```
snames(object)

## S4 method for signature 'SummarizedExperiment'
snames(object)

snames(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,character'
snames(object) <- value
```

### Arguments

object	SummarizedExperiment
value	string vector with sample names

### Value

sample names vector (get) or updated eSet (set)

### Examples

```
require(magrittr)
file <- download_data('billing16.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
head(snames(object))
head(snames(object) %<>% paste0('SAMPLE_', .))
```



---

split_by_svar	<i>Split by svar</i>
---------------	----------------------

---

**Description**

Split by svar

**Usage**

```
split_by_svar(object, svar = subgroup)
```

**Arguments**

object	SummarizedExperiment
svar	svar to split on

**Value**

list of SummarizedExperiment

**Examples**

```
file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, impute=FALSE, plot = FALSE)
split_by_svar(object)
```

---

standardize_maxquant_snames	<i>Standardize maxquant snames</i>
-----------------------------	------------------------------------

---

**Description**

Standardize maxquant sample names

**Usage**

```
standardize_maxquant_snames(x, ...)

## S3 method for class 'character'
standardize_maxquant_snames(
  x,
  quantity = guess_maxquant_quantity(x),
  verbose = FALSE,
  ...
)
```

```
## S3 method for class 'SummarizedExperiment'
standardize_maxquant_snames(
  x,
  quantity = guess_maxquant_quantity(x),
  verbose = FALSE,
  ...
)
```

### Arguments

x	character vector or SummarizedExperiment
...	allow for proper S3 method dispatch
quantity	maxquant quantity
verbose	TRUE (default) or FALSE

### Details

Drop "Ratio normalized", "LFQ intensity" etc from maxquant sample names

### Value

character vector or SummarizedExperiment

### Examples

```
# character vector
x <- "Ratio M/L normalized STD(L)_E00(M)_E01(H)_R1"
standardize_maxquant_snames(x)

x <- "Ratio M/L STD(L)_E00(M)_E01(H)_R1"
standardize_maxquant_snames(x)

x <- 'LFQ intensity STD_R1'
standardize_maxquant_snames(x)

x <- 'LFQ intensity L STD(L)_E00(M)_E01(H)_R1'
standardize_maxquant_snames(x)

x <- 'Reporter intensity 0 A(0)_B(1)_C(2)_D(3)_E(4)_F(5)_R1'
standardize_maxquant_snames(x)

x <- 'Reporter intensity corrected 0 A(0)_B(1)_C(2)_D(3)_E(4)_F(5)_R1'
standardize_maxquant_snames(x)
```

---

subgroup_array	<i>Get subgroup matrix</i>
----------------	----------------------------

---

**Description**

Arrange (subgroup)levels in matrix

**Usage**

```
subgroup_array(object, subgroupvar)
subgroup_matrix(object, subgroupvar)
```

**Arguments**

object	SummarizedExperiment
subgroupvar	subgroup svar

**Value**

matrix

**Examples**

```
file <- download_data('halama18.metabolon.xlsx')
object <- read_metabolon(file, plot=FALSE)
subgroup_matrix(object, 'Group')
```

---

subtract_baseline	<i>Subtract baseline</i>
-------------------	--------------------------

---

**Description**

Subtract baseline level within block

**Usage**

```
subtract_baseline(
  object,
  subgroupvar,
  subgroupctr = slevels(object, subgroupvar)[1],
  block = NULL,
  assaynames = setdiff(assayNames(object), "weights"),
  verbose = TRUE
)
```

```

subtract_pairs(
  object,
  subgroupvar,
  subgroupctr = slevels(object, subgroupvar)[1],
  block,
  assaynames = setdiff(assayNames(object), "weights"),
  verbose = TRUE
)

subtract_differences(object, block, subgroupvar, verbose = TRUE)

```

### Arguments

object	SummarizedExperiment
subgroupvar	subgroup svar
subgroupctr	control subgroup
block	block svar (within which subtraction is performed)
assaynames	which assays to subtract for
verbose	TRUE/FALSE

### Details

subtract\_baseline subtracts baseline levels within block, using the medoid baseline sample if multiple exist.

subtract\_pairs also subtracts baseline level within block. It cannot handle multiple baseline samples, but has instead been optimized for many blocks

subtract\_differences subtracts differences between subsequent levels, again within block

### Value

SummarizedExperiment

### Examples

```

# read
require(magrittr)
file <- download_data('atkin18.metabolon.xlsx')
object0 <- read_metabolon(file, plot=FALSE)
pca(object0, plot=TRUE, color=SET)

# subtract_baseline: takes medoid of baseline samples if multiple
object <- subtract_baseline(object0, block='SUB', subgroupvar='SET')
pca(object, plot=TRUE, color=SET)

# subtract_pairs: optimized for many blocks
object <- subtract_pairs( object0, block='SUB', subgroupvar='SET')

```

```

pca(object, plot=TRUE, color=SET)

# subtract differences
object <- subtract_differences(object0, block='SUB', subgroupvar='SET')
values(object) %<>% na_to_zero()
pca(object, plot=TRUE, color=SET)

```

---

sumexp2mae

*Create MultiAssayExperiment from SummarizedExperiment list*


---

## Description

Create MultiAssayExperiment from SummarizedExperiment list

## Usage

```
sumexp2mae(experiments)
```

## Arguments

experiments      named list of SummarizedExperiments

## Value

MultiAssayExperiment

## Examples

```

require(magrittr)
somascanfile <- download_data('atkin18.somascan.adat')
metabolonfile <- download_data('atkin18.metabolon.xlsx')
somascan <- read_somascan(somascanfile, plot=FALSE)
metabolon <- read_metabolon(metabolonfile, plot=FALSE)
svars(somascan) %<>% stringi::stri_replace_first_fixed(
  'SampleGroup', 'subgroup')
svars(metabolon) %<>% stringi::stri_replace_first_fixed(
  'Group', 'subgroup')
metabolon$replicate <- NULL
object <- sumexp2mae(list(somascan=somascan, metabolon=metabolon))

```

---

sumexp\_to\_wide\_dt      *Convert SummarizedExperiment into data.table*

---

### Description

Convert SummarizedExperiment into data.table

### Usage

```
sumexp_to_wide_dt(
  object,
  fid = "feature_id",
  fvars = intersect("feature_name", autonomics::fvars(object)),
  assay = assayNames(object)[1]
)

sumexp_to_long_dt(
  object,
  fid = "feature_id",
  fvars = intersect("feature_name", autonomics::fvars(object)),
  sid = "sample_id",
  svars = intersect("subgroup", autonomics::svars(object)),
  assay = assayNames(object) %>% intersect(c(.[1], "is_imputed"))
)

sumexp_to_subrep_dt(object, subgroup = subgroup)
```

### Arguments

object	sumexp
fid	fvar carrying feature id
fvars	additional fvars to include in table
assay	matrix in assays(object) to be used
sid	svar carrying sample id
svars	additional svars to include in table
subgroup	subgroup (sym)

### Details

- sumexp\_to\_wide\_dt: feature x sample
- sumexp\_to\_subrep\_dt: feature.subgroup x replicate
- sumexp\_to\_long\_dt: feature.sample

### Value

data.table

**Examples**

```

# Stem cell comparison
file <- download_data('billing16.proteingroups.txt')
invert_subgroups <- c('EM_E', 'BM_E', 'EM_BM')
object <- read_proteingroups(file, invert_subgroups = invert_subgroups,
                             plot=FALSE)
sumexp_to_wide_dt(object)
sumexp_to_long_dt(object)
sumexp_to_subrep_dt(object)

# Glutaminase
require(magrittr)
file <- download_data('atkin18.metabolon.xlsx')
object <- read_metabolon(file, plot=FALSE)
sumexp_to_wide_dt(object)
sumexp_to_long_dt(object)
sumexp_to_subrep_dt(object, Group)

# Fukuda
require(magrittr)
file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, impute=FALSE, plot=FALSE)
sumexp_to_long_dt(object)
object %<>% impute_systematic_nondetects(plot=FALSE)
sumexp_to_long_dt(object)

```

---

summarize\_fit

*Summarize fit*


---

**Description**

Summarize fit

**Usage**

```
summarize_fit(object, fit = intersect(names(metadata(object)), TESTS)[1])
```

**Arguments**

object	SummarizedExperiment
fit	'limma', 'lme', 'lm', 'lme', 'wilcoxon'

**Value**

data.table(contrast, nup, ndown)

**Examples**

```
file <- download_data('billing19.rnacounts.txt')
object <- read_rnaseq_counts(file, fit='limma', plot=FALSE)
summarize_fit(object, 'limma')
```

---

svalues

*Get/Set svalues*


---

**Description**

Get/Set svar values

**Usage**

```
svalues(object, svar)
```

```
subgroup_values(object)
```

```
sampleid_values(object)
```

```
svalues(object, svar) <- value
```

```
## S4 replacement method for signature 'SummarizedExperiment,character'
svalues(object, svar) <- value
```

**Arguments**

object            SummarizedExperiment

svar             sample var (character)

value            value vector

**Value**

character vector (get) or SummarizedExperiment (set)

**Examples**

```
file <- download_data('billing16.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
svalues(object, 'subgroup')
subgroup_values(object)
```



---

svars	<i>Get/Set svars</i>
-------	----------------------

---

## Description

Get/Set sample variables

## Usage

```
svars(object)

## S4 method for signature 'SummarizedExperiment'
svars(object)

## S4 method for signature 'MultiAssayExperiment'
svars(object)

svars(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,character'
svars(object) <- value

## S4 replacement method for signature 'MultiAssayExperiment,character'
svars(object) <- value
```

## Arguments

object	SummarizedExperiment
value	string factor with variable names

## Value

sample variable names (get) or updated SummarizedExperiment

## Examples

```
require(magrittr)
file <- download_data('billing16.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
svars(object)[1]
(svars(object)[1] %<>% paste0('1'))
```

---

TESTS	<i>Statistical models supported in autonomics</i>
-------	---

---

**Description**

Statistical models supported in autonomics

**Usage**

TESTS

**Format**

An object of class character of length 5.

**Examples**

TESTS

---

tpm	<i>Get/Set tpm</i>
-----	--------------------

---

**Description**

Get / Set tpm matrix

**Usage**

```
tpm(object)
```

```
## S4 method for signature 'SummarizedExperiment'  
tpm(object)
```

```
tpm(object) <- value
```

```
## S4 replacement method for signature 'SummarizedExperiment,matrix'  
tpm(object) <- value
```

```
## S4 replacement method for signature 'SummarizedExperiment,numeric'  
tpm(object) <- value
```

**Arguments**

object	SummarizedExperiment
value	tpm matrix (features x samples)

**Value**

tpm matrix (get) or updated object (set)

**Examples**

```
file <- download_data('billing19.rnacounts.txt')
object <- read_rnaseq_counts(file, plot=FALSE)
tpm(object) <- values(object)
tpm(object)[1:3, 1:3]
```

---

values

*Get/Set expr values*

---

**Description**

Get/Set value matrix

**Usage**

```
values(object)

## S4 method for signature 'SummarizedExperiment'
values(object)

values(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,matrix'
values(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,numeric'
values(object) <- value
```

**Arguments**

object	SummarizedExperiment
value	ratio matrix (features x samples)

**Value**

value matrix (get) or updated object (set)

**Examples**

```
file <- download_data('billing16.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
values(object)[1:3, 1:3]
values(object) <- 0
values(object)[1:3, 1:3]
```

venn\_detects                      *Venn detects*

---

**Description**

Venn diagram full/systematic/random detects

**Usage**

```
venn_detects(object, subgroup)
```

**Arguments**

object	SummarizedExperiment
subgroup	subgroup symbol

**Value**

NULL

**Examples**

```
file <- download_data('fukuda20.proteingroups.txt')
object <- read_proteingroups(file, impute=FALSE, plot = FALSE)
venn_detects(object, subgroup)
```

---

weights                              *Get/Set weights*

---

**Description**

Get/Set weight matrix

**Usage**

```
weights(object, ...)

## S4 method for signature 'SummarizedExperiment'
weights(object)

weights(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,matrix'
weights(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,numeric'
```

```
weights(object) <- value

## S4 replacement method for signature 'SummarizedExperiment,`NULL`'
weights(object) <- value
```

### Arguments

object	SummarizedExperiment
...	additional params
value	ratio matrix (features x samples)

### Value

weight matrix (get) or updated object (set)

### Examples

```
file <- download_data('billing19.proteingroups.txt')
object <- read_proteingroups(file, plot=FALSE)
weights(object)[1:3, 1:2]
weights(object) <- 1; weights(object)[1:3, 1:2]
```

---

zero_to_na	<i>Change nondetect representation</i>
------------	--

---

### Description

Change nondetect representation

### Usage

```
zero_to_na(x, verbose = FALSE)

nan_to_na(x, verbose = FALSE)

na_to_zero(x, verbose = FALSE)

inf_to_na(x, verbose = FALSE)

minusinf_to_na(x, verbose = FALSE)
```

### Arguments

x	matrix
verbose	logical(1)

**Value**

Updated matrix

**Examples**

```
require(magrittr)
matrix(c(0, 7), nrow=1)
matrix(c(0, 7), nrow=1) %>% zero_to_na(verbose=TRUE)

matrix(c(NA, 7), nrow=1)
matrix(c(NA, 7), nrow=1) %>% na_to_zero(verbose=TRUE)

matrix(c(NaN, 7), nrow=1)
matrix(c(NaN, 7), nrow=1) %>% nan_to_na(verbose=TRUE)

matrix(c(Inf, 7), nrow=1)
matrix(c(Inf, 7), nrow=1) %>% inf_to_na(verbose=TRUE)

matrix(c(-Inf, 7), nrow=1)
matrix(c(-Inf, 7), nrow=1) %>% minusinf_to_na(verbose=TRUE)
```

# Index

- \* **datasets**
  - AUTONOMICS\_DATASETS, [19](#)
  - MAXQUANT\_PATTERNS\_PEP\_COUNTS, [61](#)
  - MAXQUANT\_PATTERNS\_QUANTITY, [62](#)
  - TESTS, [98](#)
- .read\_maxquant, [4](#)
- .read\_metabolon, [6](#)
- .read\_rectangles, [8](#)
- .read\_rnaseq\_bams, [11](#)
- .read\_rnaseq\_counts
  - (.read\_rnaseq\_bams), [11](#)
- .read\_somascan, [14](#)
- add\_smiles, [15](#)
- analysis, [16](#)
- analysis, SummarizedExperiment-method
  - (analysis), [16](#)
- analysis<- (analysis), [16](#)
- analysis<-, SummarizedExperiment, list-method
  - (analysis), [16](#)
- analyze, [17](#)
- assert\_is\_valid\_sumexp, [18](#)
- AUTONOMICS\_DATASETS, [19](#)
  
- biplot, [19](#)
- biplot\_corrections (plot\_corrections),
  - [72](#)
- biplot\_covariates (plot\_covariates), [73](#)
  
- center, [20](#)
- contrast\_subgroup\_cols, [22](#)
- contrast\_subgroup\_rows
  - (contrast\_subgroup\_cols), [22](#)
- contrastdefs, [21](#)
- contrastdefs, SummarizedExperiment-method
  - (contrastdefs), [21](#)
- contrastdefs<- (contrastdefs), [21](#)
- contrastdefs<-, SummarizedExperiment, list-method
  - (contrastdefs), [21](#)
- counts, [23](#)
  
- counts, SummarizedExperiment-method
  - (counts), [23](#)
- counts2cpm, [24](#)
- counts2tpm, [24](#)
- counts<- (counts), [23](#)
- counts<-, SummarizedExperiment, matrix-method
  - (counts), [23](#)
- counts<-, SummarizedExperiment, NULL-method
  - (counts), [23](#)
- counts<-, SummarizedExperiment, numeric-method
  - (counts), [23](#)
- cpm, [25](#)
- cpm, SummarizedExperiment-method (cpm),
  - [25](#)
- cpm2counts (counts2cpm), [24](#)
- cpm<- (cpm), [25](#)
- cpm<-, SummarizedExperiment, matrix-method
  - (cpm), [25](#)
- cpm<-, SummarizedExperiment, numeric-method
  - (cpm), [25](#)
- create\_design, [26](#)
- create\_sfile, [27](#)
  
- default\_formula (default\_subgroupvar),
  - [28](#)
- default\_sfile, [27](#)
- default\_subgroupvar, [28](#)
- download\_data, [29](#)
- download\_gtf, [30](#)
- dt2mat, [31](#)
  
- exp2 (log2transform), [59](#)
- explore\_imputations, [31](#)
- explore\_transformations, [32](#)
- extract\_features, [33](#)
- extract\_rectangle, [34](#)
  
- fdata, [36](#)
- fdata, SummarizedExperiment-method
  - (fdata), [36](#)

- fdata<- (fdata), 36
- fdata<- , SummarizedExperiment, data.frame-method (fdata), 36
- filter\_exprs\_replicated\_in\_some\_subgroup, 37
- filter\_features, 38
- filter\_medoid, 38
- filter\_replicated, 39
- filter\_samples, 40
- fit\_limma, 40
- fit\_lm (fit\_limma), 40
- fit\_lme (fit\_limma), 40
- fit\_lmer (fit\_limma), 40
- fit\_wilcoxon (fit\_limma), 40
- flevels, 42
- fnames, 43
- fnames, SummarizedExperiment-method (fnames), 43
- fnames<- (fnames), 43
- fnames<- , SummarizedExperiment, character-method (fnames), 43
- formula2str, 44
- fvalues, 44
- fvars, 45
- fvars, SummarizedExperiment-method (fvars), 45
- fvars<- (fvars), 45
- fvars<- , SummarizedExperiment, character-method (fvars), 45
- guess\_maxquant\_quantity, 46
- guess\_sep, 47
- halfnormimpute (normimpute), 66
- impute\_systematic\_nondetects, 48
- inf\_to\_na (zero\_to\_na), 101
- invert, 49
- invnorm (log2transform), 59
- is\_imputed, 50
- is\_imputed, SummarizedExperiment-method (is\_imputed), 50
- is\_imputed<- (is\_imputed), 50
- is\_imputed<- , SummarizedExperiment, matrix-method (is\_imputed), 50
- is\_imputed<- , SummarizedExperiment, NULL-method (is\_imputed), 50
- is\_sig, 51
- lda (pca), 68
- limma, 52
- limma, SummarizedExperiment-method (limma), 52
- limma<- (limma), 52
- limma<- , SummarizedExperiment, array-method (limma), 52
- limma<- , SummarizedExperiment, NULL-method (limma), 52
- log2counts, 53
- log2counts, SummarizedExperiment-method (log2counts), 53
- log2counts<- (log2counts), 53
- log2counts<- , SummarizedExperiment, matrix-method (log2counts), 53
- log2counts<- , SummarizedExperiment, numeric-method (log2counts), 53
- log2countsratios, 54
- log2countsratios, SummarizedExperiment-method (log2countsratios), 54
- log2countsratios<- (log2countsratios), 54
- log2countsratios<- , SummarizedExperiment, matrix-method (log2countsratios), 54
- log2countsratios<- , SummarizedExperiment, numeric-method (log2countsratios), 54
- log2cpm, 55
- log2cpm, SummarizedExperiment-method (log2cpm), 55
- log2cpm<- (log2cpm), 55
- log2cpm<- , SummarizedExperiment, matrix-method (log2cpm), 55
- log2cpm<- , SummarizedExperiment, numeric-method (log2cpm), 55
- log2cpmratios, 56
- log2cpmratios, SummarizedExperiment-method (log2cpmratios), 56
- log2cpmratios<- (log2cpmratios), 56
- log2cpmratios<- , SummarizedExperiment, matrix-method (log2cpmratios), 56
- log2cpmratios<- , SummarizedExperiment, numeric-method (log2cpmratios), 56
- log2tpm, 57
- log2tpm, SummarizedExperiment-method (log2tpm), 57
- log2tpm<- (log2tpm), 57
- log2tpm<- , SummarizedExperiment, matrix-method (log2tpm), 57
- log2tpm<- , SummarizedExperiment, numeric-method



- (log2tpm), 57
- log2tpmratios, 58
- log2tpmratios, SummarizedExperiment-method (log2tpmratios), 58
- log2tpmratios<- (log2tpmratios), 58
- log2tpmratios<- , SummarizedExperiment, matrix-method (log2tpmratios), 58
- log2tpmratios<- , SummarizedExperiment, numeric-method (log2tpmratios), 58
- log2transform, 59
- make\_volcano\_dt, 60
- mat2dt (dt2mat), 31
- matrix2sumexp, 60
- MAXQUANT\_PATTERNS\_PEP COUNTS, 61
- MAXQUANT\_PATTERNS\_QUANTITY, 62
- merge\_fdata (merge\_sdata), 62
- merge\_ffile (merge\_sfile), 63
- merge\_sdata, 62
- merge\_sfile, 63
- message\_df, 65
- minusinf\_to\_na (zero\_to\_na), 101
- na\_to\_zero (zero\_to\_na), 101
- nan\_to\_na (zero\_to\_na), 101
- nfactors, 65
- normimpute, 66
- occupancies, 67
- occupancies, SummarizedExperiment-method (occupancies), 67
- occupancies<- (occupancies), 67
- occupancies<- , SummarizedExperiment, matrix-method (occupancies), 67
- occupancies<- , SummarizedExperiment, numeric-method (occupancies), 67
- pca, 68
- plot\_biplot (biplot), 19
- plot\_boxplots, 69
- plot\_contrastogram, 71
- plot\_corrections, 72
- plot\_covariates, 73
- plot\_data, 74
- plot\_densities, 75
- plot\_detections (plot\_detects), 76
- plot\_detects, 76
- plot\_feature\_boxplots (plot\_boxplots), 69
- plot\_feature\_densities (plot\_densities), 75
- plot\_feature\_profiles (plot\_features), 78
- plot\_feature\_violins (plot\_violins), 80
- plot\_features, 78
- plot\_quantifications (plot\_detects), 76
- plot\_sample\_boxplots, 76, 81
- plot\_sample\_boxplots (plot\_boxplots), 69
- plot\_sample\_densities, 70, 81
- plot\_sample\_densities (plot\_densities), 75
- plot\_sample\_violins, 70, 76
- plot\_sample\_violins (plot\_violins), 80
- plot\_subgroup\_boxplots (plot\_boxplots), 69
- plot\_subgroup\_violins (plot\_violins), 80
- plot\_summarized\_detections (plot\_detects), 76
- plot\_venn, 79
- plot\_violins, 80
- plot\_volcano, 81
- plotmat, 71
- pls (pca), 68
- preprocess\_rnaseq\_counts, 82
- proteingroups, 83
- proteingroups, SummarizedExperiment-method (proteingroups), 83
- proteingroups<- (proteingroups), 83
- proteingroups<- , SummarizedExperiment, matrix-method (proteingroups), 83
- proteingroups<- , SummarizedExperiment, numeric-method (proteingroups), 83
- quantnorm (log2transform), 59
- read\_affymetrix, 84
- read\_metabolon (.read\_metabolon), 6
- read\_phosphosites (.read\_maxquant), 4
- read\_proteingroups (.read\_maxquant), 4
- read\_rectangles (.read\_rectangles), 8
- read\_rnaseq\_bams (.read\_rnaseq\_bams), 11
- read\_rnaseq\_counts (.read\_rnaseq\_bams), 11
- read\_somascan (.read\_somascan), 14
- rm\_singleton\_samples, 85
- sampleid\_values (svalues), 96
- scaledlibsizes, 86

- sdata, 86
- sdata, MultiAssayExperiment-method (sdata), 86
- sdata, SummarizedExperiment-method (sdata), 86
- sdata<- (sdata), 86
- sdata<-, MultiAssayExperiment, data.frame-method (sdata), 86
- sdata<-, MultiAssayExperiment, DataFrame-method (sdata), 86
- sdata<-, SummarizedExperiment, data.frame-method (sdata), 86
- sdata<-, SummarizedExperiment, DataFrame-method (sdata), 86
- slevels, 87
- sma (pca), 68
- snames, 88
- snames, SummarizedExperiment-method (snames), 88
- snames<- (snames), 88
- snames<-, SummarizedExperiment, character-method (snames), 88
- split\_by\_svar, 89
- split\_extract (nfactors), 65
- standardize\_maxquant\_snames, 89
- subgroup\_array, 91
- subgroup\_levels (slevels), 87
- subgroup\_matrix (subgroup\_array), 91
- subgroup\_values (svalues), 96
- subtract\_baseline, 91
- subtract\_differences (subtract\_baseline), 91
- subtract\_pairs (subtract\_baseline), 91
- sumexp2mae, 93
- sumexp\_to\_long\_dt (sumexp\_to\_wide\_dt), 94
- sumexp\_to\_subrep\_dt (sumexp\_to\_wide\_dt), 94
- sumexp\_to\_wide\_dt, 94
- summarize\_fit, 95
- svalues, 96
- svalues<- (svalues), 96
- svalues<-, SummarizedExperiment, character-method (svalues), 96
- svars, 97
- svars, MultiAssayExperiment-method (svars), 97
- svars, SummarizedExperiment-method (svars), 97
- svars<- (svars), 97
- svars<-, MultiAssayExperiment, character-method (svars), 97
- svars<-, SummarizedExperiment, character-method (svars), 97
- TESTS, 98
- tpm, 98
- tpm, SummarizedExperiment-method (tpm), 98
- tpm<- (tpm), 98
- tpm<-, SummarizedExperiment, matrix-method (tpm), 98
- tpm<-, SummarizedExperiment, numeric-method (tpm), 98
- translate (normimpute), 66
- values, 99
- values, SummarizedExperiment-method (values), 99
- values<- (values), 99
- values<-, SummarizedExperiment, matrix-method (values), 99
- values<-, SummarizedExperiment, numeric-method (values), 99
- venn\_detects, 100
- weights, 100
- weights, SummarizedExperiment-method (weights), 100
- weights<- (weights), 100
- weights<-, SummarizedExperiment, matrix-method (weights), 100
- weights<-, SummarizedExperiment, NULL-method (weights), 100
- weights<-, SummarizedExperiment, numeric-method (weights), 100
- zero\_to\_na, 101
- zeroimpute (normimpute), 66
- zscore (log2transform), 59