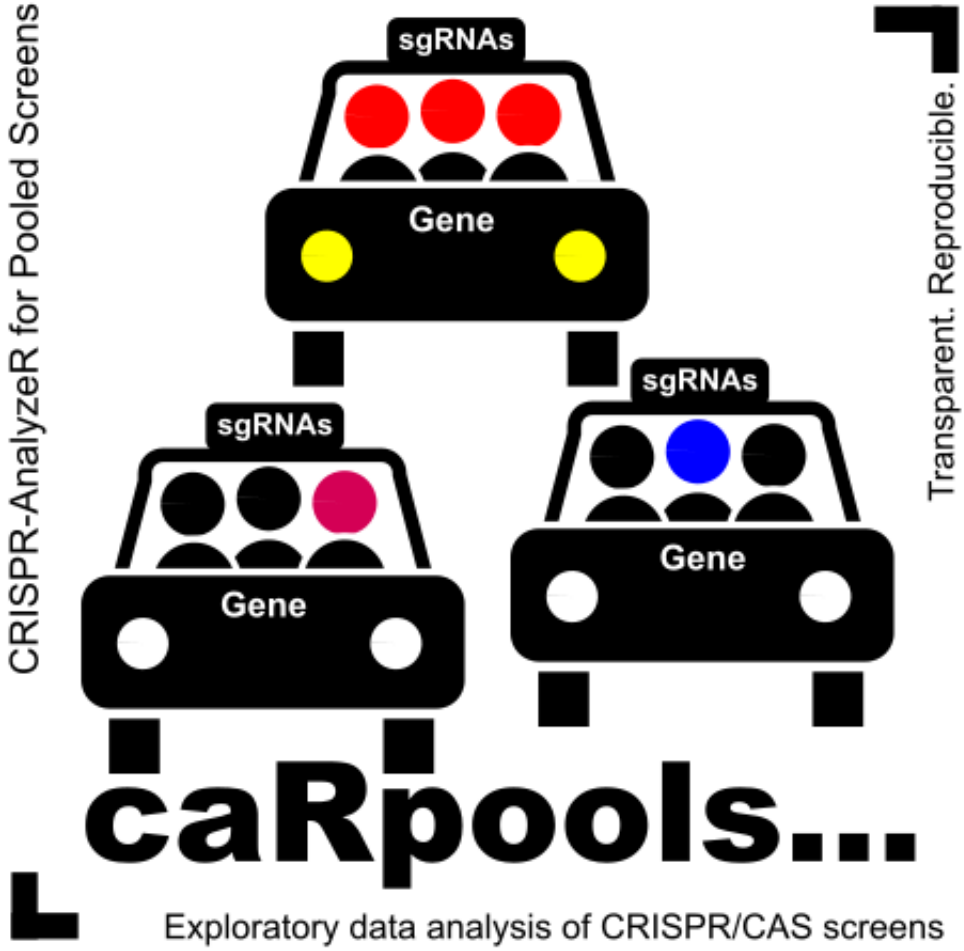
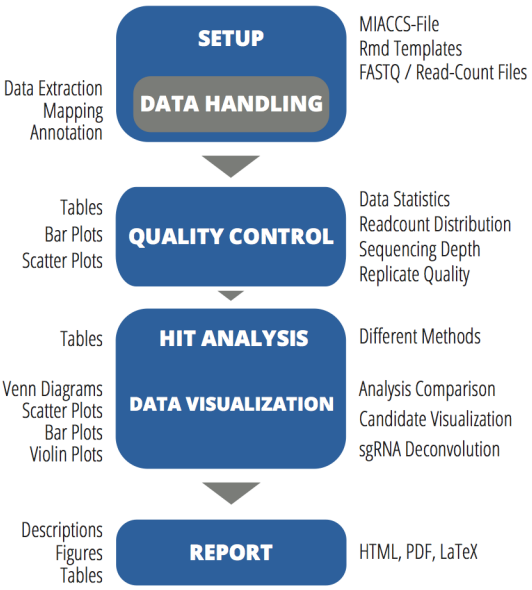
caRpools - End-To-End Analysis of Pooled CRISPR/Cas9 Screens

Contents


CRISPR-AnalyzeR for Pooled Screens
sgRNAs
Gene
sgRNAs
Gene
sgRNAs
Gene
Transparent. Reproducible.
caRpools...
Exploratory data analysis of CRISPR/CAS screens

1 Introduction

The **CaRpools** (**C**RISPR **A**nalyze**R** for **Pool**ed **S**creens) package allows users to analyze raw NGS read count data from pooled CRISPR Screens in an end-to-end fashion and serves as a basis for creating customized reports for more advanced users.
These pooled screens must contain lentivirus-based libraries as they can be obtained via Addgene e.g. . It provides functions to create different quality control plots, normalize the data, compare the data and perform hit identification via three different methods.
Furthermore it can be used to completely analyze the date in a streamlined workflow with the provided analysis template.



With **CaRpools**, the user can analyze pooled CRISPR/Cas9 screens end-to-end in a standardized fashion allowing for reproducible data analysis.
This includes:

- FASTQ data extraction (optional)
- Mapping against a reference library FASTA file using bowtie2 (optional)
- Extract Mapping data and convert it into read count files (optional)
- Generate quality control plot
- Generate data set statistics
- Perform hit analysis using three different methods
- Visualize hit candidates
- Compare hit analysis for most robust hit identification
- Retrieve information about single sgRNAs for hit candidates
- Generate standardized reports, which can be customized (for advanced users)

1.1 Download caRpools

CaRpools is available as an R package **caRpools** without the scripts and template files.
The complete package with the PERL scripts and all template files can be obtained from Github (https://github.com/boutroslab/carpools) and our website CRISPR-AnalyzeR.de.

We recommend to download the template files and Scripts from Github and install caRpools in R using the package installer 'install.packages("caRpools").

1.2 Quality Control

Available Quality Control plots include (for sgRNAs or summarized for genes):

- Read-count Distribution `carpools.read.distribution`
- Read Depth `carpools.read.depth`
- sgRNAs present per Gene Target `carpools.reads.genedesigns`
- Read-count Overview per sgRNA / Gene `carpools.read.ballmap`
- Read-count comparison of different samples `carpools.read.count.vs`
- Read-count / Fold-change of sgRNAs for single genes `carpools.raw.genes`

1.3 Annotation

Since several gene identifiers are used for generating CRISPR pooled KO libraries, e.g. EnsemblID,
loaded data-sets can automatically be annotated with further gene annotation like official gene symbol or descriptions
using the `biomaRt` interface.
More details can be found below in the section of `get.gene.info` or via `?get.gene.info` / `?biomaRt`.

1.4 Gene Read Count Data or Removing Gene Data

Furthermore, sgRNA read-count data can be aggregated (summed up) to corresponding genes `aggregatetogenes` or gene data can be excluded for the analysis using `gene.remove`.

1.5 Hit Analysis

Moreover, this package can be used to identify screening hits using either

- DESeq2 `stat.DESeq`
- Wilcox-based hit calling `stat.wilcox`
- MAGeCK `stat.mageck`

and hit calling / data analysis can be compared between all the methods with `compare.analysis` or `carpools.hit.overview`. Finally evaluated data analysis can be visualized in different ways with `carpools.hitident` for all methods listed above.

1.6 sgRNA Information

Moreover, you will also be provided with in-depth information about the sgRNAs of your genes.
These can be derived from `carpools.raw.genes` for any gene or `carpools.hit.sgrna` for your hit candidates in an automated way.

1.7 Report

All this data can be used to automatically generate a standardized report using the provided R Markdown template file including all plots and tables for you data analysis.
Two different templates are provides:

- A standard report for generating short, brief data analysis
- An extended report with additional plots and tables

The R Markdown templates provide the user with the ability to generate HTML output as well as PDF output (LATEX installation necessary) including high-quality plots.

Provided PDF templates:

- CaRpools-PDF.rmd
- CaRpools-extended-PDF.rmd

Provided HTML templates:

- CaRpools-HTML.rmd
- CaRpools-extended-HTML.rmd

2 Requirements and Installation

2.1 Virtual Box Image

We also included a Virtual Box Image that already includes all necessary software and package files.
You just need to install Virtual Box 5 from the Website.

You can download the caRpools virtual box image from our website crispr-analyzer.de or Github (https://github.com/boutroslab/carpools).

2.1.1 How to use the Virtual Box caRpools

Please see the VirtualBox tutorial for instructions.

2.2 Download caRpools

CaRpools is available as an R package **caRpools** without the scripts and template files.
The complete package with the PERL scripts and all template files can be obtained from Github (https://github.com/boutroslab/carpools) and our website.

We recommend to download the template files and Scripts from Github and install caRpools in R using the package installer 'install.packages("caRpools").

2.3 Hardware Requirements

For CRISPR-Libraries of 12 K size (12K sgRNAs), caRpools will work on any laptop/PC with at least 4GB of RAM and a modern dual-core CPU.
CRISPR-Libraries with a size of more than 100 K (100 K sgRNAs) run best with at least 8 GB of RAM.

2.4 Software Requirements

CaRpools was tested on MacOSX Yosemite and Ubuntu 14.04 LTS.
However, it should work on any operating system that fulfills the software requirements.

The following software needs to be installed:

- PERL 5
- Bowtie2 2.2.0 or higher Website
- MAGeCK 0.51 (password protected download) Website
- TexLive Website
- pdflatex
- xelatex
- R 3.2.0 or higher Website
- Pandoc 1.15.0.6 Website
- R-Studio Website (GUI)

The following **R packages** need be installed (can be done via `load.packages()`):

- Bioconductor Basics
  - BiocInstaller >= 1.18.3
  - BiocGenerics >= 0.14.0

- **biomaRt >=2.24.0**
- seqinr >= 3.1-3
- xlsx >= 0.5.7
- rJava >= 0.9.6
- xlsxjars >= 0.6.1
- stringi >= 0.5
- scatterplot3d >= 0.3
- MESS >= 0.3
- DESeq2 >= 1.8.1
- rmarkdown >= 0.7
- knitr >= 1.10.5
- VennDiagram >= 1.6.9
- sm >= 2.2

2.5 BiomaRt and Annotation Requirements

Please note that for any annotation, biomaRt needs full access to the internet. In case of incorrect proxy settings, the report generation will fail with a biomaRt error.
This means that if any proxy server is used, this has to be configured before using caRpools as described in the following articles:

- BiomaRt vignette
- Setting up proxy in R-Studio
- Setting up proxy in R, Stackoverflow
- Setting up Proxy for R/R-Studio in Ubuntu
- Configuration of Proxy for R

2.6 Installation Procedure

Install all software listed above according to the installation information stated on the software website.
All necessary R packages can be installed automatically by `load.packages()`.

2.7 Dataset / Screening Requirements

Since CaRpools fosters reproducibility of CRISPR/Cas9 screens, the following requirements for pooled screening data **must be fullfilled to analyze data**:

- Two biological replicates of *untreated* sample data
- Two biological replicates of *treated* sample data

By default, *TWO* replicates per condition are used by the templates provided. If you want to analyse more than two replicates per group, please have a look at *Providing more than two replicates* .

2.8 Files that can be loaded

The following files are required for data analysis:

Either
* NGS FASTQ file for each sample * Library reference in FASTA format

or
* Final read count file for each sample * Library reference in FASTA format

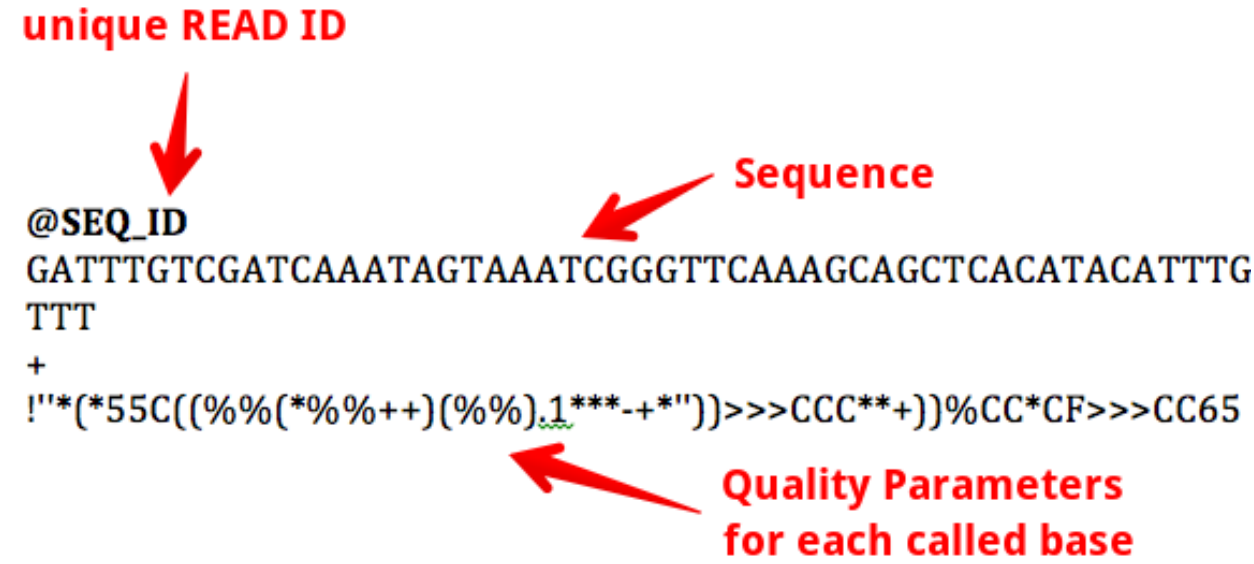
CaRpools accepts either FASTQ files or read count files for each sample. FASTQ file are then extracted and mapped using Bowtie2 against the library reference. Finally, read count files for each sample are generated.

As an alternative, these final read count files can be provided as well, so that no extraction or mapping is necessary.
In addition, a library reference file in FASTA format is necessary, usually this is the file that was used for ordering custom oligo libraries.

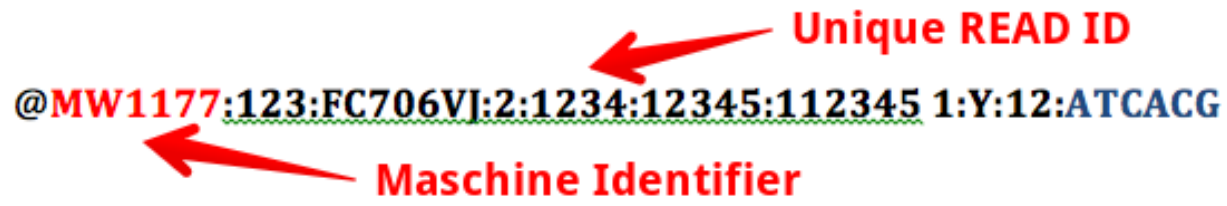
File structures are shown below.

2.8.1 Structure of NGS Readcount Data

General information about the FASTQ file format can be obtained in an easy-to-understand article from Wikipedia.



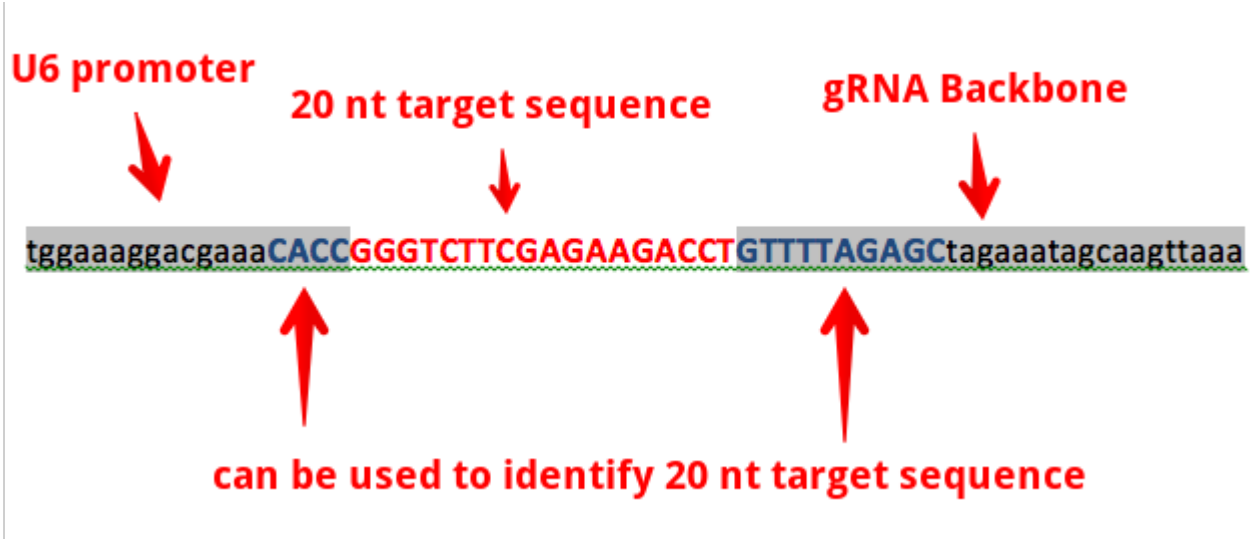
maschine.pattern
The machine pattern used for extracting the sequences is a regular expression to identify the read ID including your sequencing machine.
in the case of the above sample, the PERL regular expression used must be **M01100**.



2.8.2 Extraction Pattern

CaRpools extracts the integrated DNA sequence of your target sequence as a DNA barcode.
In order to extract this sequence, a **PERL regular expression** pattern is used to identify the desired nucleotide sequence, which is called **seq.pattern**.

As an example, part of a U6 promoter-driven sgRNA cassette is given as follows:



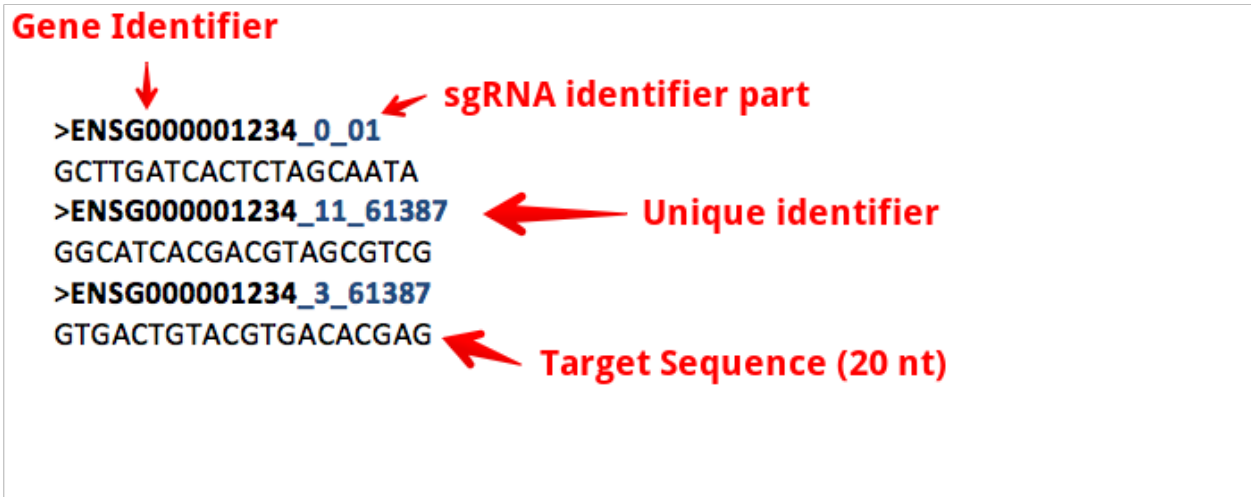
Since we want to extract the target sequence, the regular expression will use a part of the U6 promoter and a part of the sgRNA backbone to identify the target sequence.

CACC **(.{20})** GTTTTAGAGC

The parenthesis are necessary to extract the target sequence, for more information please see RegExR.

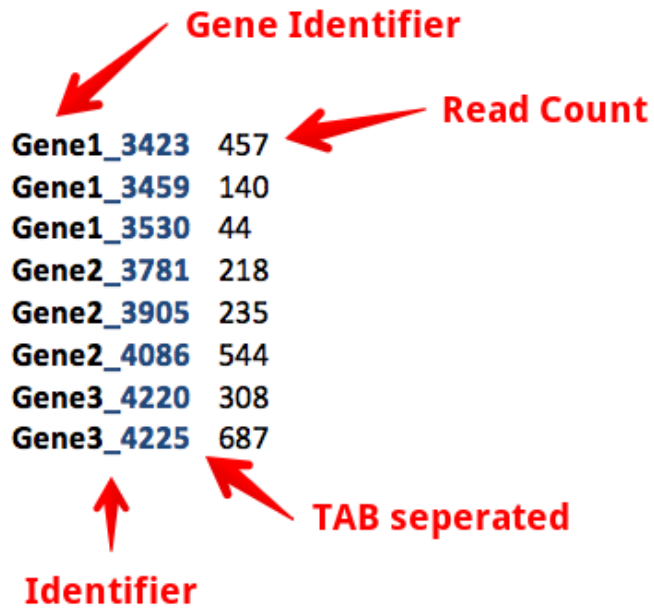
2.8.3 Structure of FASTA Library Reference File

The library reference file must be in FASTA format and include ALL sgRNAs present in the data-set with exactly the same naming.
e.g.



2.8.4 Read-Count Data Files

CaRpools also takes read count files. If FASTQ files are provided and extracted/mapped with CaRpools, read count files will be created for each sample.
Data for each sample must be formatted in a **tab-separated** way as follows:



As shown in the above file, **Gene1** is the Gene identifier, **_3423** is a unique part for identification of this sgRNA for the given gene and *Gene1_3423* is the whole identifier which uniquely identifies this sgRNA within the data-set.

The name of each complete sgRNA **must** contain the gene it refers to either as gene symbol or any other identifier which shares the same separator in addition to a part of identifier that is unique for each sgRNA for that gene.
The name can therefore be anything, as long as the *identifier is the same for each gene* and the *separator is the same for all data.*

In principle a sgRNA identifier must consist of a **gene identifier**, followed by a **seperator** (e.g. _ or -) as well as a *secondary* **sgRNA identifier**.

Please note:
* Read counts MUST be numeric only. * Within the sgRNA/Gene identifer, no special characters *except for* _ *and* - are allowed!

2.8.5 File Loading

Files can be loaded via `load.file(filename, header, sep)` with the following arguments

- filename: the filename as character
- header: TRUE / FALSE , whether a header is used (as it is for the above sample)
- sep: the separator, default is `\t` for tab-separated files.

Please see `?read.table` for a detailed description of the arguments.

2.9 Files and Folder Structure to use CaRpools

Please note: the MAIN FOLDER must be the R working directory!
Data and Script paths can be adjusted in the MIACCS file.

The following files are necessary to use CaRpools for report generation:

MIACCS.xls
Minimum Information About CRISPR/Cas Screens. This file needs to be filled out to provide all necessary information about the screen.

R Markdown Template files
Either CaRpools-extended-PDF.rmd, CaRpools-PDF.rmd or CaRpools-extended-HTML.rmd or CaRpools-HTML.rmd. Is the template for report generation.

Data Files
Two replicates (or more) per Control and Treated. Can be FASTQ files OR already mapped, not normalized read count files.

CRISPR-mapping.pl
PERL script to map your extracted FASTQ files, if desired (as indicated in the MIACCS.xls)

CRISPR-extract.pl
PERL script to extract 20 nt target sequence from FAST files, if desired (as indicated in the MIACCS.xls)

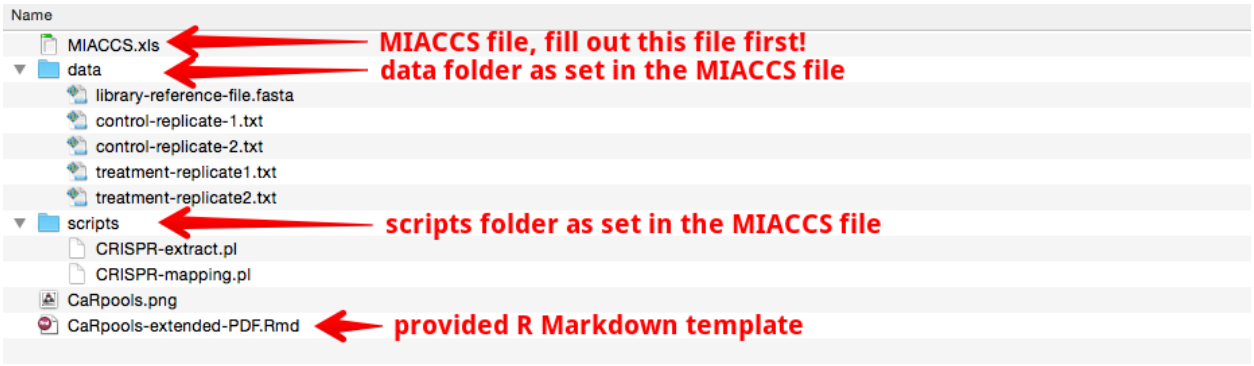
CaRpools.png
The logo file

The following files are necessary to use *single* CaRpools functions:

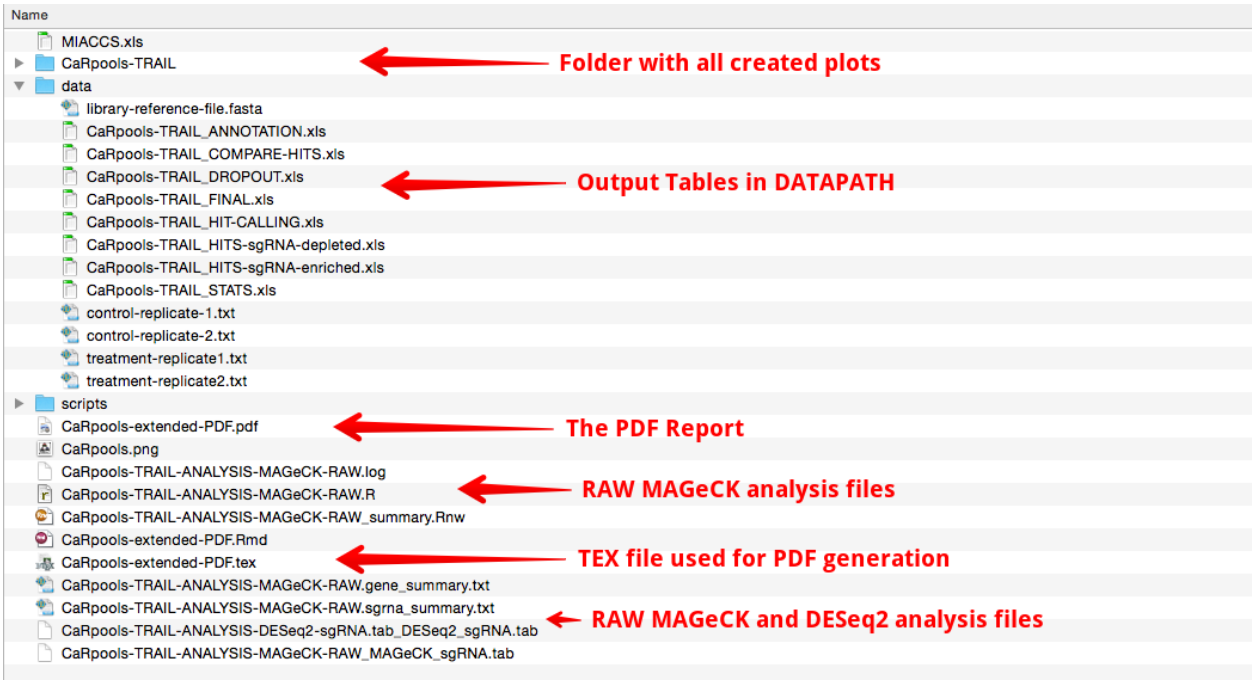
Data Files
Either raw read count files or FASTQ files (that need to be extracted and mapped using CaRpools)

Please note that CaRpools always starts with loading data files. For raw-read-count files, use `load.file`. For FASTQ files, please see the sections below.

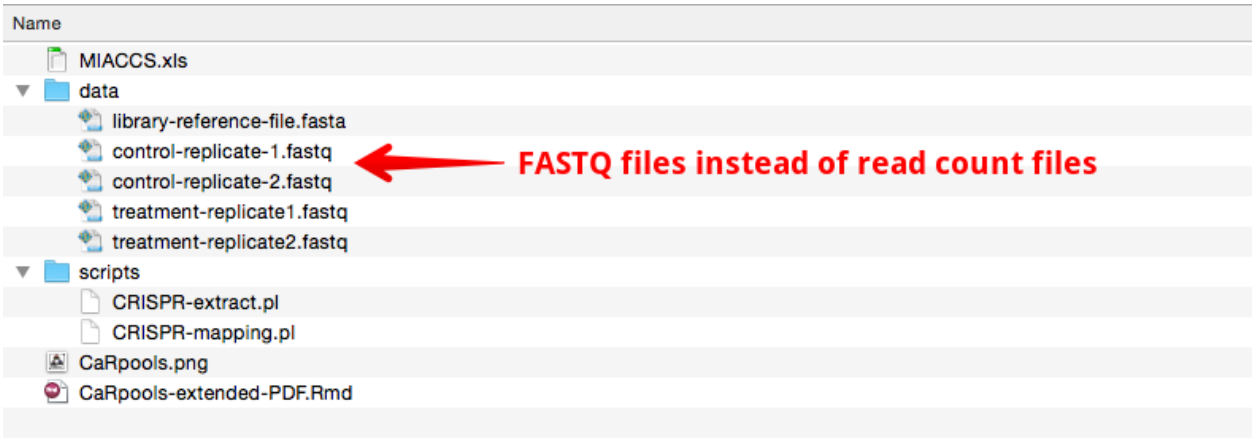
CaRpools folder structure for Report Generation using raw Read Count files:



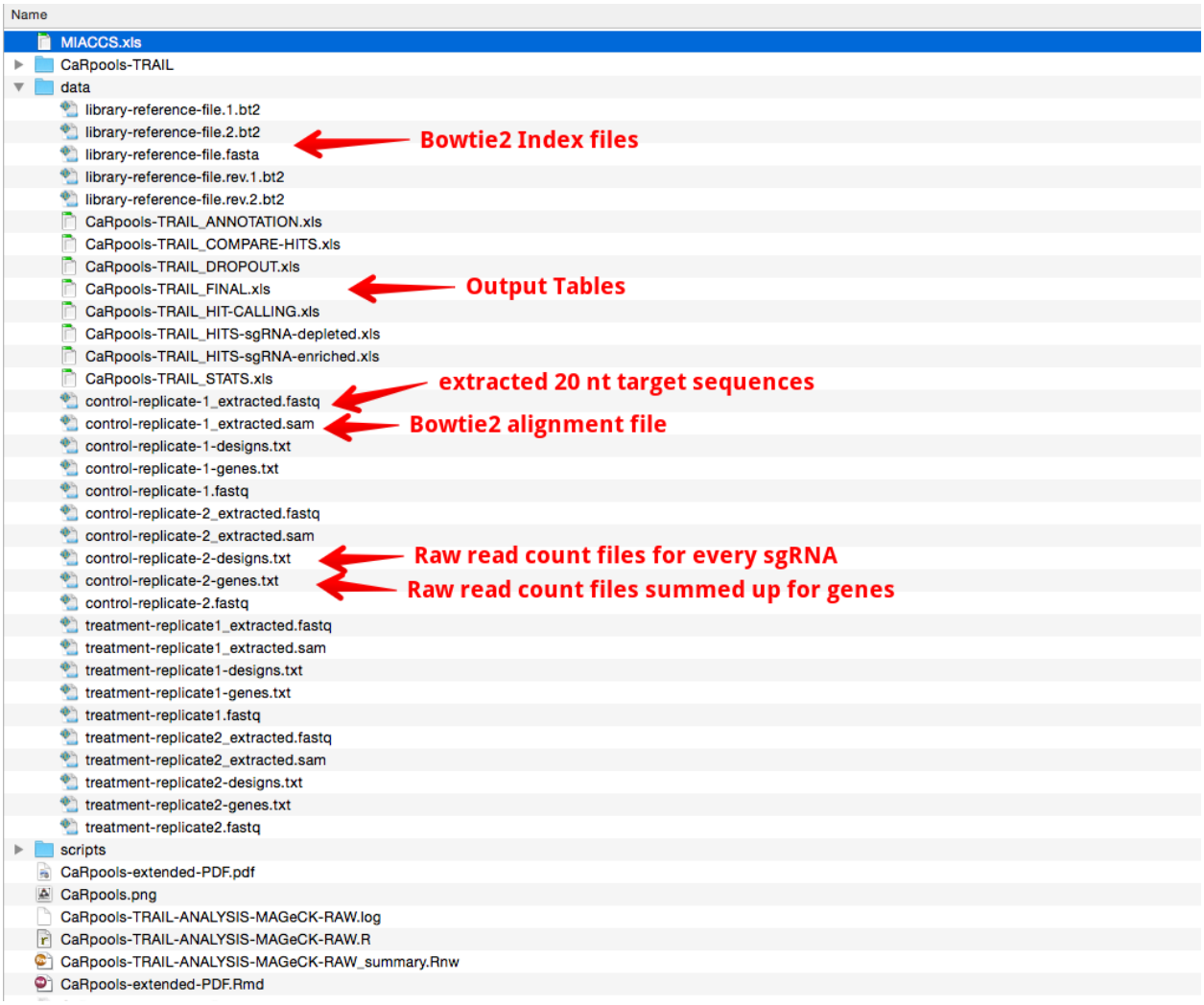
CaRpools folder structure for Report Generation using raw Read Count files AFTER REPORT GENERATION:



CaRpools folder structrue for Report Generation using FASTQ files:



CaRpools folder structure for Report Generation using FASTQ files AFTER REPORT GENERATION:



3 Using CaRpools with Provided R Markdown Templates

By default, CaRpools is meant to be used to generate standardized reports of CRISPR/Cas9 pooled screenings using the provided R markdown templates.
However, these templates can also serve as a basic frame for advanced users to modify them and create customized reports to their needs.

In the following sections, the use of CaRpools with the provided template files is explained in more detail. Please see the above instructions of how to install all necessary tools, packages and components for CaRpools.

For the modification of the templates, please see the manuals for R Markdown and knitr. Single parameters and detailed instructions of the single function within CaRpools are given in the section **Using CaRpools as a stand-alone R Package**.

3.1 Setup Files and R-Studio

All packages and software tools need be installed correctly as shown before.

1. Copy all files in the designated folders as shown above.

- **Please note: the MAIN FOLDER must be R working directory!**
- The MIACCS.xls as well the R markdown template and CaRpools.png must be in the same folder as the R working dir.

2. Adjust the path to the data and scripts folder if necessary in the MIACCS.xls . Use the absolute path. If the folder structure is as shown above, you do not need to make any adjustments.
3. Adjust and fill out the **MIACCS.xls** file.
4. You can use `CarPools(type="check")` to check for the correct folder structure and data file presence as it is indicated in the MIACCS.xls file.
5. You can check for your R working directory by `getwd()` and set it to any directory you want by `setwd("/PATH")`.

3.1.1 Check Setup

You can verify that the MIACCS.xls file as well as the used template file and all necessary scripts are found by calling `check.caRpools()`.
See below for more information about the arguments.
By default, it **requires a correct MIACCS file + the script files + all packages installed + MAGeCK + Bowtie2 + Pandoc.**

3.2 The MIACCS.xls file

The MIACCS (**M**inimum **I**nformation **A**bout **C**RISPR/**C**as **S**creens) file, provided as Excel sheet, combines all information which is necessary to reproduce the screens. This includes a description of the screen, hypothesis, all materials used as well as the experimental workflow.
Moreover, the MIACCS file is used to setup the analysis using caRpools, thus allowing a reproducible and standardized way of performing and analyzing pooled CRISPR/Cas screens.

The following parameters are found within the MIACCS.xls file:

3.2.1 Using FASTQ Files

Instead of providing already mapped read count files, caRpools provides the functionality to use the generated RAW FASTQ files as they are supplied by the sequencing machine, e.g. Illumina MiSeq/NextSeq/HiSeq systems.
If desired, caRpools can extract your 20 nt target sequence (without filtering for quality) from these files and map it against your library reference using bowtie2. In this case, you need to provide the following files that must be within the datapath set in the MIACCS file:

- Two replicate FASTQ files for the untreated sample
- Two replicate FASTQ files for the treated sample
- Your ordered library reference as FASTA file
- optional bowtie2 index files of your library fasta files (generated via bowtie2-build)

Moreover, the wrapper files CRISPR-mapping.pl and CRISPR-extract.pl need to be in the scriptpath as set in the MIACCS file.

caRpools will then generate the following files during the extraction and mapping:

filename_extracted.fastq FASTQ file with the extracted 20 nt target sequence
multiple library-filename.bt2 bowtie2 index files (if create index files set as TRUE in the MIACCS file)
filename_extracted.sam SAM alignment file of the extracted 20 nt target sequence against the library reference
filename_extracted-designs.txt Read Count files for each sgRNA
filename_extracted-genes.txt Read Count files SUMMED up for each gene (not used during analysis)

The following parameters must be set in the MIACCS.xls file under **Data Extraction and Bowtie2 Mapping/Alignment** to extract file from FASTQ data:

Do you want to extract the sgRNA target sequence from FASTQ? *TRUE or FALSE*
This indicates whether you would like to extract the 20 nt target sequence from the provided FASTQ file (TRUE).

Regular expression to extract target sequence from FASTQ file *PERL regular expression pattern*
This must be a PERL regular expression pattern that is used to identify the 20 nt target sequence. By default, this is *ACC(.{20})GT{2,4}AGAGC* and can be used for the GeckoV2 library or other libraries that are based on U6-driven sgRNA expression . Can be ANY PERL expression pattern. The only importance is, that the part which identifies the 20 nt target sequence is in parenthesis only.

Regular Expression to extract maschine ID from Reads
This must be the unique sequencing machine ID/serial number as indicated in the FASTQ files.

Is the data within the FASTQ file in Reverse Complement? *TRUE or FALSE*
Did you sequence your data in reverse complement order? By default, this is set to FALSE.

Do you want to map the reads to the reference file? *TRUE or FALSE*
If set to TRUE, bowtie2 will be called to map the extracted or not-extracted FASTQ files to your bowtie2 reference library index. For further information about bowtie2 mapping, please see the Bowtie2 Manual.

Do you want to create the Bowtie2 index files? *TRUE or FALSE*
If set to TRUE, bowtie2 indexes will be created for your library-reference.fasta file. If you do not have created this indexes before, set this to TRUE so that they are created before the mapping to your library reference is performed. More information about bowtie2 index files can be found here.

How many threads shall bowtie2 use? *integer*
In the case bowtie2 mapping is set to TRUE, please give the number of cores or threads to use. For most dual-core CPU, the default value 4 is fine. More information about threading can be found here.

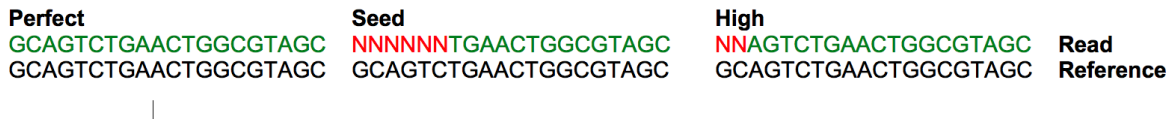
Bowtie2 Sensitivity? *default: very-sensitive-local*
Other options: very-fast, fast, sensitive, very-fast-local, fast-local, sensitive-local

You can adjust the sensitivity of bowtie2 using this parameter. By default, bowtie2 is used in a very-sensitive-local setting. More information about different sensitivity parameters can be found at the bowtie2 options.

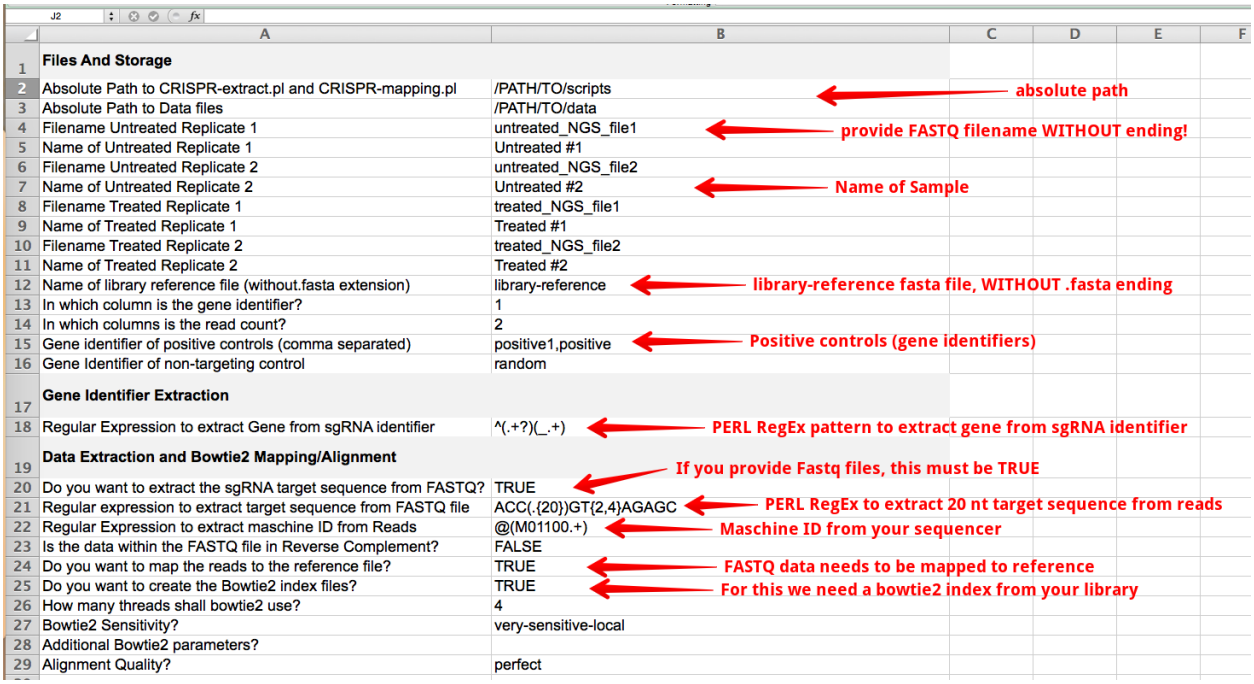
Additional Bowtie2 parameters?
You can pass additional parameters to bowtie2 as they are indicated in the bowtie2 manual. Please leave empty if not used.

Alignment Quality
After bowtie2 mapping, the alignment is converted into read count files *filename_extracted-design.txt* and *filename_extracted-genes.txt*.
You can indicate how well the alignment must be in order to be used for generating the read count for each sgRNA.
By default, this is set to *perfect*, which only employs a mapped read if the full 20 nt from the sequencing match perfectly to the sgRNA found in your library reference. The following options can be used:

- **perfect** - Read is used of all 20 nt from the sequencing are matching the target sequence given in the library reference
- **high** - Read is used if at least 18 nt (starting from the PAM) are matching the target sequence in the reference
- **seed** - Read is used if at least 14 nt (starting from the PAM) are a perfect match against the target sequence in the reference



3.2.1.1 Example of a MIACCS File entry for FASTQ files



	A	B	C	D	E	F
1	**Files And Storage**					
2	Absolute Path to CRISPR-extract.pl and CRISPR-mapping.pl	/PATH/TO/scripts				
3	Absolute Path to Data files	/PATH/TO/data				
4	Filename Untreated Replicate 1	untreated_NGS_file1				
5	Name of Untreated Replicate 1	Untreated #1				
6	Filename Untreated Replicate 2	untreated_NGS_file2				
7	Name of Untreated Replicate 2	Untreated #2				
8	Filename Treated Replicate 1	treated_NGS_file1				
9	Name of Treated Replicate 1	Treated #1				
10	Filename Treated Replicate 2	treated_NGS_file2				
11	Name of Treated Replicate 2	Treated #2				
12	Name of library reference file (without.fasta extension)	library-reference				
13	In which column is the gene identifier?	1				
14	In which columns is the read count?	2				
15	Gene identifier of positive controls (comma separated)	positive1,positive				
16	Gene Identifier of non-targeting control	random				
17	**Gene Identifier Extraction**					
18	Regular Expression to extract Gene from sgRNA identifier	^(.+?)(_.+)				
19	**Data Extraction and Bowtie2 Mapping/Alignment**					
20	Do you want to extract the sgRNA target sequence from FASTQ?	TRUE				
21	Regular expression to extract target sequence from FASTQ file	ACC(.{20})GT{2,4}AGAGC				
22	Regular Expression to extract maschine ID from Reads	@(M01100.+)				
23	Is the data within the FASTQ file in Reverse Complement?	FALSE				
24	Do you want to map the reads to the reference file?	TRUE				
25	Do you want to create the Bowtie2 index files?	TRUE				
26	How many threads shall bowtie2 use?	4				
27	Bowtie2 Sensitivity?	very-sensitive-local				
28	Additional Bowtie2 parameters?					
29	Alignment Quality?	perfect				

3.2.2 Using Read Count Files

Instead of using raw NGS Fastq files, caRpools can also be used to directly analysis read-count files without fastq data extraction and mapping.
In this case, already finalized read-count files are provided that must be in the same format as described in **Read Count Data Files**.

If you provide read-count files, please make sure you fill out the MIACCS file accordingly.
An example of a MIACCS file for read-count files is shown below.

3.2.2.1 Example of a MIACCS File entry for Read-Count files

A	B
Files And Storage	
Absolute Path to CRISPR-extract.pl and CRISPR-mapping.pl	/PATH/TO/scripts
Absolute Path to Data files	/PATH/TO/data
Filename Untreated Replicate 1	**untreated_readcount_file1.txt**
Name of Untreated Replicate 1	Untreated #1
Filename Untreated Replicate 2	**untreated_readcount_file2.txt**
Name of Untreated Replicate 2	Untreated #2
Filename Treated Replicate 1	**treated_readcount_file1.txt**
Name of Treated Replicate 1	Treated #1
Filename Treated Replicate 2	**treated_readcount_file2.txt**
Name of Treated Replicate 2	Treated #2
Name of library reference file (without.fasta extension)	library-reference
In which column is the gene identifier?	1
In which columns is the read count?	2
Gene identifier of positive controls (comma separated)	positive1,positive
Gene Identifier of non-targeting control	random
Gene Identifier Extraction	
Regular Expression to extract Gene from sgRNA identifier	^(.+?)(_.+)
Data Extraction and Bowtie2 Mapping/Alignment	
Do you want to extract the sgRNA target sequence from FASTQ?	**FALSE**
Regular expression to extract target sequence from FASTQ file	ACC(.{20})GT{2,4}AGAGC
Regular Expression to extract maschine ID from Reads	M01100
Is the data within the FASTQ file in Reverse Complement?	**FALSE**
Do you want to map the reads to the reference file?	**FALSE**
Do you want to create the Bowtie2 index files?	**FALSE**
How many threads shall bowtie2 use?	4
Bowtie2 Sensitivity?	very-sensitive-local
Additional Bowtie2 parameters?	
Alignment Quality?	perfect

absoluteb path to script files and data files

Read-Count file, WITH extension

library reference FASTA file, WITHOUT extension

positive control gene identifier

since already mapped read-count files are provided, set this to FALSE

In general:
If read-count files are provided, these settings will not influence data analysis

3.2.3 Data Analysis Options

CaRpools is not only meant as a tool for standardized and fully reproducible data analysis of CRISPR-Cas screens, but also fosters exploratory data analysis.
Therefore caRpools offers the user different options for data analysis as illustrated below. All options can be set within the MIACCS file or directly passed on to the individual functions of caRpools (advanced users).

A	B
General Analysis	
Add statistic to plots?	TRUE
Normalize Read Count of each replicate?	TRUE
Function to normalize replicates	median
Color of highlighted data points	orange
Label names in Hit Calling Plot?	TRUE
Hit Calling	
pvalue threshold wilcox	0.05
random picks for wilcox	150
pvalue threshold DESeq2	0.001
pvalue threshold MAGeCK	0.05
# of top sgRNAs to label for DESeq2	3
# of top sgRNAs to label for MAGeCK	3
Override the p-value cutoff in Hit Analysis Comparison?	FALSE
if Override, how many gene will be compared?	10
Percentage of TOP ENRICHED genes to list	5
Percentage of TOP DEPLETED genes to list	5

TRUE if you want statistics like median, mean etc. added to the plot

if data is not normalized, set TRUE for normalization

the mathematical function used for normalization

color of genes are highlighted within plots

TRUE if you want genes being labeled in the hit calling plot

p-value threshold for Wilcox

if no non-targeting controls are set, wilcox will pick this many sgRNAs

p-value threshold for DESeq2

p-value theshold for MAGeCK

please indicate the number of labeled genes in sgRNA plots (extended only!)

Do you want to override significance levels for final candidate comparison?

If override is set to TRUE, use the top X genes from each method

Moreover, you can indicate a percentage of top genes to be displayed as well

3.3 Providing more than two replicates per Group

CaRpools allows the usage of more than two replicates for both the untreated and treated groups. However, the standard templates we provide accept two biological replicates only.

There are two ways of using the caRpools report generation function with more replicates:

- use the provided templates stating **4-replicates**
- modify the templates

If you like to modify the template of caRpools, you just need to add certain statements for the additional replicates. This will change with the next larger update.

3.3.0.1 Load more files

When loading files (e.g. read-count files or fastq files for extraction), you can just put additional code of the following scheme:

```
## for data extraction of a third replicate or more

# First edit the MIACCS file loading
fileCONTROL3 = as.character(miaccs["carpools.untreated3",3])
d.CONTROL3 = as.character(miaccs["carpools.untreated3.desc",3])

fileTREAT3 = as.character(miaccs["carpools.treated3",3])
d.TREAT3 = as.character(miaccs["carpools.treated3.desc",3])

# Additional CONTROL replicate
fileCONTROL3 = data.extract(scriptpath, datapath, fastqfile=fileCONTROL3,
extract, seq.pattern, maschine.pattern, createindex, referencefile,
mapping, reversecomplement, threads, bowtieparams, sensitivity, match)
# Additional Treatment replicate
fileTREAT3 = data.extract(scriptpath, datapath, fastqfile=fileTREAT3,
extract, seq.pattern, maschine.pattern, createindex, referencefile,
mapping, reversecomplement, threads, bowtieparams, sensitivity, match)

## If you do not want to extract FASTQ files, but just load a read-count file this is the way to go

CONTROL3 = load.file(paste(datapath, fileCONTROL3, sep="/"))
TREAT3 = load.file(paste(datapath, fileTREAT3, sep="/"))
```

3.3.0.2 QC plots and Gene Annotation using several replicates

In principal you can use as many replicates per treatment group as you want, the minimum however is two replicates. Some of the QC plots and other functions require single replicates, others can be used with as many replicates as you want directly within the function call.

This means you need to call the function for each replicate separately for the following functions:

- load.file
- data.extract
- get.gene.info
- aggregatetogenes
- stats.data

- unmapped.genes
- carpools.read.distribution
- carpools.reads.genedesigns

These function accept several replicates directly as a list:

- carpools.read.depth
- carpools.readcount.vs
- stat.wilcox
- stat.deseq
- stat.mageck

All other functions do not depend on read-count data directly so that you do not think about replicates for those, as all handling is done automatically.

3.3.0.3 Provide more replicates to statistical analysis

```
## WILCOX
# Provide more replicates as list
data.wilcox = stat.wilcox(untreated.list = list(CONTROL1, CONTROL2, CONTROL3, CONTROL4),
treated.list = list(TREAT1,TREAT2, TREAT3, TREAT4), namecolumn=namecolumn,
fullmatchcolumn=fullmatchcolumn, normalize=normalize, norm.fun=norm.function,
sorting=FALSE, controls=controls.nontarget, control.picks=control.picks)

## DESEQ2
data.deseq = stat.DESeq(untreated.list =  list(CONTROL1, CONTROL2, CONTROL3, CONTROL4),
treated.list = list(TREAT1,TREAT2, TREAT3, TREAT4), namecolumn=namecolumn,
fullmatchcolumn=fullmatchcolumn, extractpattern=g.extractpattern, sorting=FALSE,
filename.deseq = paste(analysis.name, "-ANALYSIS-DESeq2-sgRNA.tab", sep=""), sgRNA.pval = sig.pval.dese

## MAGECK
data.mageck = stat.mageck(untreated.list = list(CONTROL1, CONTROL2, CONTROL3, CONTROL4),
treated.list = list(TREAT1,TREAT2, TREAT3, TREAT4), namecolumn=namecolumn,
fullmatchcolumn=fullmatchcolumn, norm.fun="median", extractpattern=g.extractpattern,
mageckfolder=NULL, sort.criteria="neg", adjust.method="fdr",
filename = paste(analysis.name, "-ANALYSIS-MAGeCK-RAW", sep=""), fdr.pval=sig.pval.mageck)
```

4 Start CaRpools Report Generation

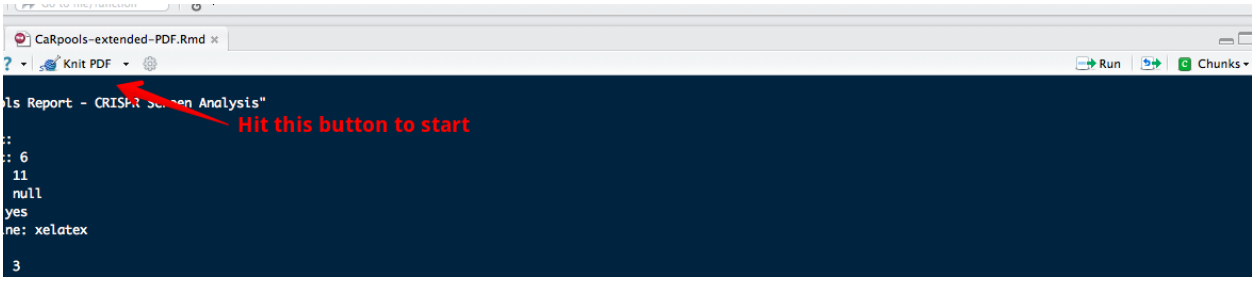
You can start caRpools Report Generation after you did the following steps:

- Installed all required software and R packages (use *check.caRpools(files=FALSE)* to verify)
- Put every file in the correct folder (MIACCS, data files, script files, Rmd templates)
- Put everything in the R working directory or set the working directory to the folder of your files
- Filled out the MIACCS file with all information, e.g. correct filenames, reference, data analysis options

You can check for all requirements by calling `check.caRpools`.

4.1 Start CaRpools using R-Studio

In the case you use R-Studio, you can start caRpools by just opening the corresponding Rmd template file. At the top, you will find the **Knit PDF** or **Knit HTML** button, so you just need to press that and caRpools will generate the report.



As an alternative, you can start caRpools via `use.caRpools` and provide additional parameters (see below).

4.2 Start CaRpools using R console

Moreover, caRpools report generation can also be initiated without R-studio installation, so that this can be done via R command line even on remote computers.
In this case, caRpools report generation can be started via `use.caRpools` with additional parameters, which are described below.

4.2.0.4 use.caRpools()

Usage:
`use.caRpools(type=NULL, file="CaRpools-extended-PDF.Rmd", miaccs="MIACCS.xls", check=TRUE, work.dir=NULL)`

type
Description If you provide a custom Rmd template that can generate both, PDF and HTML reports you can indicate which version you want to generate.
Default NULL
Values "PDF", "HTML"

file
Description The file name of your custom Rmd template file (with extension).
Default "CaRpools-extended-PDF.Rmd"
Values filename as character

miaccs
Description The filename of your MIACCS file.
Default "MIACCS.xls"
Values filename as character

check
Description Indicates whether caRpools will check for correct installation and file access.
Default TRUE
Values TRUE or FALSE (boolean)

work.dir
Description You can provide the absolute path to the working directory in which all files are placed (e.g. the MIACCS.xls and Rmd template).
Default NULL *Values* absolute path (character) or NULL if standard R working directory is used

5 Using CaRpools as a stand-alone R Package

CaRpools can be used without the provided R markdown templates to generate own reports or individualized analysis of CRISPR/Cas9 screens.
The following sections will explain every function included in CaRpools and provide hints, tips and a documentation of how to use CaRpools for data analysis. This will help you in customizing the generated reports to you needs, create new reports or just use single functions of CaRpools within your data analysis workflow.

5.1 Loading Data

CaRpools offers two ways of providing CRISPR/Cas9 screening data.
Either raw **read-count files** are directly used as described before, or read-count files are generated from NGS FASTQ files by extracting the 20 nt target sequence, mapping it against a reference library and extracting the read-count information for each sgRNA identifier.

```
CONTROL1 = load.file("./data/untreated1.txt", header= TRUE, sep="\t")
```

Usage
`load.file(filename, header = TRUE, sep="\t", comment.char="", type = NULL)`

filename Filename to be loaded. If read-count files are loaded, please include the extension. For FASTQ files, please pass on the output of `data.extract` as this is the filename of the extracted file.
Default empty
Values filename (character) or output from `data.extract`

header
Is the first line just a header?
Default TRUE
Values TRUE, FALSE (boolean)

sep
Separator of data, usually tab-separated.
Default "" for TAB-separated data
Values Any separator

comment.char
Are character files within any quote etc? *Default* "" *Values* Any character

type
Indicates which files are being loaded. Three file-types are possible. Either NULL for tab-separated files, fastalib for library reference files or xlsx for loading the MIACCS file.
Default NULL *Values* NULL, "fastalib", "xlsx"

5.1.1 Loading Read-Count files Directly

Read-Count files (normalized or not normalized) can be loaded with `load.file` into a data frame that is used for further usage.
For each untreated and treated sample two replicates are used:

```
CONTROL1 = load.file("./data/untreated1.txt", header= TRUE, sep="\t")
CONTROL = load.file("./data/untreated2.txt", header= TRUE, sep="\t")
TREAT1 = load.file("./data/treated1.txt", header= TRUE, sep="\t")
TREAT2 = load.file("./data/treated2.txt", header= TRUE, sep="\t")
```

```
# Don't forget the library reference
libFILE = load.file(
  paste(datapath, paste(referencefile,".fasta",sep=""), sep="/"),header = FALSE,
  type="fastalib")
```

5.1.2 Loading FASTQ Files to Generate Read-Count Files

In a first step, NGS FASTQ data is extracted and mapped against a reference library file using bowtie2.

```
fileCONTROL1 = data.extract(
  scriptpath="path.to.scripts", datapath="path.to.FASTQ",fastqfile="filename1",
  extract=TRUE, seq.pattern, maschine.pattern, createindex=TRUE,
  bowtie2file=filename.lib.reference, referencefile="filename.lib.reference",
  mapping=TRUE, reversecomplement=FALSE, threads, bowtieparams,
  sensitivity="very-sensitive-local",match="perfect")
# Now we can load the generated Read-Count file directly!
CONTROL1 = load.file(paste(datapath, fileCONTROL1, sep="/")) # Untreated sample 1 loaded

# Don't forget the library reference
libFILE = load.file(
  paste(datapath, paste(referencefile,".fasta",sep=""), sep="/"),header = FALSE,
  type="fastalib")
```

A closer look to the data frame we just loaded:

	sgRNA	Count
40	ABI1_129_0	12
41	ABI1_12_0	50
42	ABI1_132_0	131
43	ABI1_132_16356	90
44	ABI1_132_65424	28
45	ABI1_133_0	14
46	ABI1_134_0	49
47	ABI1_134_49068	92
48	ABL1_0_148908	146
49	ABL1_108_148908	126
50	ABL1_109_148908	62

5.1.2.1 data.extract()

Usage:

```
data.extract(scriptpath=NULL, datapath=NULL, fastqfile=NULL, extract = FALSE, pattern =
"default", maschinepattern = "default", createindex = FALSE, bowtie2file = NULL, referencefile=
NULL, mapping = FALSE, reversecomplement = FALSE, threads = 1, bowtieparams = "", sensitivity
= "very-sensitive-local", match = "perfect")
```

scriptpath
Absolute path of the folder that contains `CRISPR-extract.pl` and `CRISPR-mapping.pl`
Default NULL
Values absolute path (character)

datapath
Absolute path of the folder that contains the data files (e.g. file.FASTQ)
Default NULL
Values absolute path (character)

fastqfile
Filename of FASTQ file WITHOUT .fastq extension
Default NULL
Values filename (character)

extract
Whether CRISPR-extract.pl is used to extract the 20 nt target sequence from the NGS reads using `pattern`
Default FALSE
Values TRUE, FALSE (boolean)

pattern
PERL regular Expression to extract 20 nt target sequence from NGS reads. Please see *extract pattern* in this manual for more information.
Default Regular Expression (character)

maschinepattern
Machine ID of your Sequencing machine. Used to identify the read id.

createindex
Do you want caRpools to generate a bowtie2 index? Only necessary if `mapping=TRUE`. *Default* FALSE
Values TRUE, FALSE

bowtie2file
Filename of bowtie2 index file, without extension. Is the same as reference file, if `createindex=TRUE`.

referencefile
Filename of the library reference FASTA file, without extension. Is the same as bowtie2 file, if `createindex=TRUE`.

mapping
Indicates whether FASTQ files need to be mapped against `referencefile/bowtie2file`. FALSE by default.
Default FALSE
Values TRUE, FALSE

reversecomplement
Is the NGS sequence in reverse complement order?
Default FALSE
Values TRUE, FALSE

threads
How many threads can bowtie2 use for mapping? Only used if `mapping=TRUE`. Usually cores of CPU.
Default 2
Values any integer

bowtieparams
If you want to pass additional parameters to bowtie2.

sensitivity
You can adjust the sensitivity of bowtie2 using this parameter. By default, bowtie2 is used in a very-sensitive-local setting. More information about different sensitivity parameters can be found at the bowtie2 options.
Default "very-sensitive-local"
Other options: very-fast, fast, sensitive, very-fast-local, fast-local, sensitive-local

match
After bowtie2 mapping, the alignment is converted into read count files *filename_extracted-design.txt* and *filename_extracted-genes.txt*.

You can indicate how well the alignment must be in order to be used for generating the read count for each sgRNA.
By default, this is set to *perfect*, which only employs a mapped read if the full 20 nt from the sequencing match perfectly to the sgRNA found in your library reference. The following options can be used:

- **perfect** - Read is used of all 20 nt from the sequencing are matching the target sequence given in the library reference
- **high** - Read is used if at least 18 nt (starting from the PAM) are matching the target sequence in the reference
- **seed** - Read is used if at least 14 nt (starting from the PAM) are a perfect match against the target sequence in the reference

5.1.3 Aggregating sgRNA read-count to Gene read-count

CaRpools allows you to aggregate sgRNA read-counts to gene read-counts using `aggregatetogenes()`.

```
CONTROL1.g=aggregatetogenes(data.frame = CONTROL1, agg.function=sum,
        extractpattern = expression("^(.+?)(_.+)"),type = "aggregate")
```

5.1.3.1 aggregatetogenes()

Usage
`aggregatetogenes(data.frame, namecolumn=1, countcolumn=2, agg.function=sum, extractpattern=expression("`
`type="aggregate")`

data.frame
A data-frame of sgRNA read-count data as done by `load.file`
Default empty
Values data-frame

namecolumn
In which column are the sgRNA identifiers?
Default 1
Values column number (numeric)

countcolumn
In which column are the read counts?
Default 2
Values column number (numeric)

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier.
e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

type
CaRpools can either aggregate the data frame (`type = "annotate"`) or annotate the gene identifiers only as an additional column (`type = "annotate"`).
Default "aggregate"
Values "aggregate", "annotate"

5.1.3.1.1 Example for data aggregation

```
CONTROL1.g=aggregatetogenes(data.frame = CONTROL1, agg.function=sum,
            extractpattern = expression("^(.+?)(_.+)"), type="aggregate")
knitr::kable(CONTROL1.g[1:10,])
```

	sgRNA	Count	Gene
AAK1	AAK1	881	AAK1
AATK	AATK	2114	AATK
ABI1	ABI1	1591	ABI1
ABL1	ABL1	1015	ABL1
ABL2	ABL2	1257	ABL2
ACAD10	ACAD10	968	ACAD10
ACVR1	ACVR1	1526	ACVR1
ACVR1B	ACVR1B	1109	ACVR1B
ACVR1C	ACVR1C	1248	ACVR1C
ACVR2A	ACVR2A	749	ACVR2A

5.1.3.1.2 Example for data annotation

```
CONTROL1.g.annotate=aggregatetogenes(data.frame = CONTROL1, agg.function=sum,
            extractpattern = expression("^(.+?)(_.+)"), type="annotate")
knitr::kable(CONTROL1.g.annotate[1:10,])
```

sgRNA	Count	Gene
AAK1_104_0	0	AAK1
AAK1_105_0	197	AAK1
AAK1_106_0	271	AAK1
AAK1_107_0	1	AAK1
AAK1_108_0	0	AAK1
AAK1_109_0	89	AAK1
AAK1_111_182532	0	AAK1
AAK1_111_60844	11	AAK1
AAK1_112_60844	55	AAK1
AAK1_113_60844	16	AAK1

5.1.4 Convert Gene Identifier or Annotate Gene Information

It is also possible to either enrich the screening data-set file with additional information provided by the biomaRt interface.
For example, gene identifiers can be changed from EnsemblIDs to official gene symbols are Gene Ontology terms can be added to the data-set.
This can be done using `get.gene.info`, which serves as a wrapper for the **biomaRt** package with its load of options and possibilities (more information see `?biomaRt`).

You can convert any gene identifier which is included in your sgRNA identifier to e.g. EnsemblID or HGNC Gene Symbol using caRpools.
Please note that Internet Access is required for biomaRt.
For further information about biomaRt conversion, please see the biomaRt Manual.

```
# Convert HGNC gene symbol to EnsemblID
CONTROL1 = get.gene.info(
  CONTROL1, namecolumn=1, extractpattern=expression("^(.+?)(_.+)"),
  database="ensembl", dataset="hsapiens_gene_ensembl", filters="hgnc_symbol",
  attributes = "ensembl_gene_id", return.val = "dataset")
# Also convert the library reference
libFILE = get.gene.info(
  libFILE, namecolumn=1, extractpattern=expression("^(.+?)(_.+)"),
  database="ensembl", dataset="hsapiens_gene_ensembl", filters="hgnc_symbol",
  attributes = "ensembl_gene_id", return.val = "dataset")
```

This will convert a gene identifier, which is extracted by a PERL Regular Expression **(.+?)(_.+)** and of type hgnc_symbol (gene symbol), to an Ensembl Gene ID.

5.1.4.1 get.gene.info()

Usage

```
get.gene.info = function(data, namecolumn=1, extractpattern=expression("^(.+?)(_.+)"),
database="ensembl", dataset="hsapiens_gene_ensembl", filters="ensembl_gene_id", attributes
= c("hgnc_symbol"), return.val = "dataset", controls=FALSE)
```

data Data frame that contains read-count data. *Default* none *Values* data.frame containing read-count data (data.frame)

namecolumn
In which column are the sgRNA identifiers?
Default 1
Values column number (numeric)

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier.
e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

database
BiomaRt database to be used. See `?listMarts()` or biomaRt documentation.
Default "ensembl", is using the ensembl database
Values Any biomaRt database (character)

dataset
The biomaRt data-set to be used. For *homo sapiens*, *hsapiens_gene_ensembl* is recommended. See `?listDatasets` or biomaRt documentation.
Default "hsapiens_gene_ensembl"
Values Any biomaRt data-set (character)

filters
The input filter information to retrieve biomaRt annotation, usually is the type of gene identifier used in the read-count files, e.g. "ensemble_gene_id". see `?listFilters`
Default "ensembl_gene_id"
Values Any biomaRt filter (character)

attributes
The output attribute to retrieve from biomaRt, usually the annotations that need to be fetched, e.g. "hgnc_symbol". see `?listAttributes`

Default "hgnc_symbol"
Values Any biomaRt attribute (character)

return.val
The type of object that is returned. For whole data-set, e.g. conversion of gene identifiers, use "dataset".
Default "dataset"
Values "dataset" (will give back the same data frame, but with exchanged gene identifiers), "info" (will return a data frame with all attributes fetched for genes, is used to annotate gene with additional information)

controls
Is set to TRUE if `data` is not a data frame, but a vector.
Default FALSE
Values TRUE, FALSE (boolean)

5.1.4.1.1 Example: Replace Gene Identifier (EnsemblID) by Official Gene Symbol

```
CONTROL1.replaced = get.gene.info(CONTROL1, namecolumn=1,
      extractpattern=expression("^(.+?)(_.+)"), database="ensembl", dataset="hsapiens_gene_ensembl",
      filters="hgnc_symbol", attributes = c("ensembl_gene_id"), return.val = "dataset")
```

```
## Creating a generic function for 'nchar' from package 'base' in package 'S4Vectors'
```

```
knitr::kable(CONTROL1.replaced[1:10,])
```

sgRNA	Count
ENSG00000115977_104_0	0
ENSG00000115977_105_0	197
ENSG00000115977_106_0	271
ENSG00000115977_107_0	1
ENSG00000115977_108_0	0
ENSG00000115977_109_0	89
ENSG00000115977_111_182532	0
ENSG00000115977_111_60844	11
ENSG00000115977_112_60844	55
ENSG00000115977_113_60844	16

5.1.4.1.2 Example: Enrich Dataset by Gene Description

```
CONTROL1.replaced.info = get.gene.info(CONTROL1, namecolumn=1,
      extractpattern=expression("^(.+?)(_.+)"), database="ensembl", dataset="hsapiens_gene_ensembl",
      filters="hgnc_symbol", attributes = c("ensembl_gene_id","description"), return.val = "info")

knitr::kable(CONTROL1.replaced.info[1:10,])
```

hgnc_symbol	ensembl_gene_id	description
AAK1	ENSG00000115977	AP2 associated kinase 1 [Source:HGNC Symbol;Acc:HGNC:19679]
AATK	ENSG00000181409	apoptosis-associated tyrosine kinase [Source:HGNC Symbol;Acc:HGNC:21]
ABI1	ENSG00000136754	abl-interactor 1 [Source:HGNC Symbol;Acc:HGNC:11320]
ABL1	ENSG00000097007	ABL proto-oncogene 1, non-receptor tyrosine kinase [Source:HGNC Symbol;Acc:HGN
ABL2	ENSG00000143322	ABL proto-oncogene 2, non-receptor tyrosine kinase [Source:HGNC Symbol;Acc:HGN

hgnc_symbol	ensembl_gene_id	description
ACAD10	ENSG00000111271	acyl-CoA dehydrogenase family, member 10 [Source:HGNC Symbol;Acc:HGNC:21597]
ACVR1	ENSG00000115170	activin A receptor, type I [Source:HGNC Symbol;Acc:HGNC:171]
ACVR1B	ENSG00000135503	activin A receptor, type IB [Source:HGNC Symbol;Acc:HGNC:172]
ACVR1C	ENSG00000123612	activin A receptor, type IC [Source:HGNC Symbol;Acc:HGNC:18123]
ACVR2A	ENSG00000121989	activin A receptor, type IIA [Source:HGNC Symbol;Acc:HGNC:173]

5.2 Quality Control of pooled CRISPR Screening data

There are several different functions available to determine the quality of your data:

- General Data-set Statistics
- Specific Statistics of Included Controls
- Read-count Distribution `plot.read.distribution`
- Read Depth `plot.read.depth`
- sgRNAs present per Gene Target `plot.reads.genedesigns`
- Read-count comparison of different samples `plot.read.count.vs`
- Information about the phenotype of sgRNAs `plot.raw.genes`

5.2.1 Dataset Statistics

General statistics for a given data-set can be obtained by `stats.data`.
For further information about the possibilities, see `?stats.data`.

A typical table would look like this (shorted):

Readcount	Control #1	Control #2	TRAIL #1	TRAIL #2
Mean	79	108	109	109
Median	51	74	41	39
SD	90	115	244	260
Min	0	0	0	0
Max	1141	1304	6819	8257
# sgRNA not present	934	817	1563	1573

```
# General
U1.stats = stats.data(dataset=CONTROL1, namecolumn = 1, fullmatchcolumn = 2,
                      extractpattern=expression("^(.+?)_.+"), type="stats")
U2.stats = stats.data(dataset=CONTROL2, namecolumn = 1, fullmatchcolumn = 2,
                      extractpattern=expression("^(.+?)_.+"), type="stats")
T1.stats =stats.data(dataset=TREAT1, namecolumn = 1, fullmatchcolumn = 2,
                     extractpattern=expression("^(.+?)_.+"), type="stats")
T2.stats =stats.data(dataset=TREAT2, namecolumn = 1, fullmatchcolumn = 2,
                     extractpattern=expression("^(.+?)_.+"), type="stats")
# Combine Stats
combined.stats = cbind.data.frame(U1.stats[,1:2], U2.stats[,2], T1.stats[,2], T2.stats[,2])
colnames(combined.stats) = c("Readcount", d.CONTROL1, d.CONTROL2, d.TREAT1, d.TREAT2)

# output all to a file
xlsx::write.xlsx(combined.stats, file="General-stats.xls", sheetName="Combined Stats", row.names=FALSE)

# Create stats for whole data set
U1.allstats = stats.data(dataset=CONTROL1, namecolumn = namecolumn,
    fullmatchcolumn = fullmatchcolumn, extractpattern=g.extractpattern, type="dataset")
```

5.2.1.1 stats.data()

Usage

```
stats.data = (dataset, namecolumn = 1, fullmatchcolumn = 2, extractpattern=expression("^(.+?)_.+"),
```

`readcount.unmapped.total = NA, controls.target = NULL, controls.nontarget = "random", type="stats")`

dataset
Data frame of read-count object.
Default none
Values data frame as created by `load.file()`

namecolumn
In which column are the sgRNA identifiers?
Default 1
Values column number (numeric)

fullmatchcolumn
In which column are the read counts?
Default 2
Values column number (numeric)

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier. e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

readcount.unmapped.total
Number of raw NGS reads, only used if `type="mapping`.
Default NA
Values Number of raw reads (integer)

controls.target
If `type="controls"`, this is the gene identifier of the positive control.
Default NULL
Value Gene Identifier (character)

controls.nontarget
If `type="controls"`, this is the gene identifier of the non-targeting control.
Default "random"
Value Gene Identifier (character)

type
Which type of statistic will be generated.
Default "stats"
Values "stats" will generate short statistics like median and mean for the data set, "mapping" will generate an overview of how many reads are present, "dataset" is used to generate in-depth statistics for each gene of a data-set, "controls" is used for in-depth statistics of the controls.

5.2.1.1.1 Example of Mapping Statistic

If `readcount.unmapped.total` is provided, which is the number of raw read counts for a given data-set before mapping back to a reference, the percentage of mapped data can be obtained.

```
knitr::kable(stats.data(dataset=CONTROL1, namecolumn = 1, fullmatchcolumn = 2,
  extractpattern=expression("^(.+?)_.+"), readcount.unmapped.total = 1786217, type="mapping"))
```

type	readcount
Mapped Read Counts	990775

type	readcount
Total Read Counts	1786217
Percentage	55.468 %

5.2.1.1.2 Example of Dataset Statistics

In addition to mapping information, whole data-set statistics can be plotted.

```
knitr::kable(stats.data(dataset=CONTROL1, namecolumn = 1, fullmatchcolumn = 2,
  extractpattern=expression("^(.+?)_.+"), readcount.unmapped.total = 1786217,
  type="stats"))
```

readcount	sgRNA	gene
Mean	79	1219
Median	51	1140
SD	90	1363
Min	0	2
Max	1141	37864
# sgRNA not present	934	0

Furthermore, detailed lists can be created for each data-set.

```
knitr::kable(stats.data(dataset=CONTROL1, namecolumn = 1, fullmatchcolumn = 2,
  extractpattern=expression("^(.+?)_.+"), readcount.unmapped.total = 1786217,
  type="dataset")[1:10,1:5])
```

	readcount.mean	readcount.median	readcount.sd	readcount.min	readcount.max
AAK1	55	24	79	0	271
AATK	132	100	111	7	417
ABI1	106	81	118	12	455
ABL1	68	62	55	0	190
ABL2	79	44	70	0	185
ACAD10	60	17	125	0	505
ACVR1	95	60	93	4	353
ACVR1B	74	39	87	0	305
ACVR1C	78	78	71	0	213
ACVR2A	47	43	41	0	159

5.2.2 Missing sgRNAs

CaRpools also provides you with the number of missing sgRNA, that means sgRNAs without a single read during NGS.

```
# Calculate the number
U1.unmapped = unmapped.genes(data=CONTROL1, namecolumn=1, fullmatchcolumn=2, genes=NULL, extractpattern=

# output to a file
xlsx::write.xlsx(U1.unmapped, file="DROPOUT.xls", sheetName=d.CONTROL1, row.names=FALSE)
```

The output is sorted by Gene identifiers:

	name	sgRNA
AAK1	AAK1	3
ABL1	ABL1	1
ABL2	ABL2	2
ACAD10	ACAD10	1
ACVR1B	ACVR1B	2
ACVR1C	ACVR1C	1
ACVR2A	ACVR2A	3
ACVR2B	ACVR2B	1
ACVRL1	ACVRL1	1
ADAM9	ADAM9	4

If you want to know WHICH sgRNAs dropped out for a given gene, please consider using `genes` as an optional argument with the gene identifier of interest:

```
# Calculate the number
U1.unmapped = unmapped.genes(data=CONTROL1, namecolumn=1, fullmatchcolumn=2, genes="random", extractpat
```

sgRNA	gene
random_104	random
random_109	random
random_148	random
random_152	random
random_175	random
random_185	random
random_209	random
random_231	random
random_26	random
random_263	random

5.2.2.1 unmapped.genes()

Usage

```
unmapped.genes(data, namecolumn=1, fullmatchcolumn=2, genes=NULL, extractpattern=expression("^(.+?)_.+"
```

data
A data.frame as created by `load.file`.
Default empty
Values read-count data.frame

namecolumn
In which column are the sgRNA identifiers?
Default 1
Values column number (numeric)

fullmatchcolumn
In which column are the read counts?

Default 2
Values column number (numeric)

genes
If you want to know how many sgRNAs are not present for a single gene, set `genes` to your gene identifier of interest.
Default NULL
Values gene identifier (character)

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier.
e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

5.2.3 Read-Count Distribution

A distribution for NGS data read-count can be created by `carpools.read.distribution` to visualize how the data set is distributed.
This allows to check for data skewness and to estimate the overall assay quality.
For further details see `?carpools.read.distribution`.

```
carpools.read.distribution(CONTROL1, fullmatchcolumn=2, breaks=200,
  title=d.CONTROL1, xlab="log2 Readcount", ylab="# sgRNAs",statistics=TRUE)
carpools.read.distribution(CONTROL1, fullmatchcolumn=2, breaks=200,
  title=d.CONTROL1, xlab="log2 Readcount", ylab="# sgRNAs",statistics=TRUE, type="whisker")
```

5.2.3.1 carpools.read.distribution()

Usage
```
carpools.read.distribution=(dataset,namecolumn=1, fullmatchcolumn=2,breaks="NULL", title="Title",
xlab="X-Axis", ylab="Y-Axis",statistics=TRUE, col=rgb(0, 0, 0, alpha = 0.65), extractpattern=expression
plotgene=NULL, type="distribution", logscale=TRUE)
```

dataset Data frame of read-count data as created by load.file().
Default none
Values A data frame

namecolumn
In which column are the sgRNA identifiers?
Default 1
Values column number (numeric)

fullmatchcolumn
In which column are the read counts?
Default 2
Values column number (numeric)

breaks
Histogram breaks see `?hist`. By default, will be calculated according to the data-set length.
Default NULL *Values* (numeric)

title
Main title of plot
Default "Title"
Values "The title you want" (character)

xlab
Label of X-Axis
Default "X-Axis"
Values "Label of X-Axis" (character)

ylab
Label of Y-Axis
Default "Y-Axis"
Values "Label of Y-Axis" (character)

statistics
Whether basic statistics will be shown in the plot.
Default TRUE
Values TRUE, FALSE (boolean)

col
The color of the plotted data. Can be any R color or RGB object. See ?rgb() for further information.
Default rgb(0, 0, 0, alpha = 0.65)
Values Any R color name or RGB color object (character OR color object)

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier.
e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

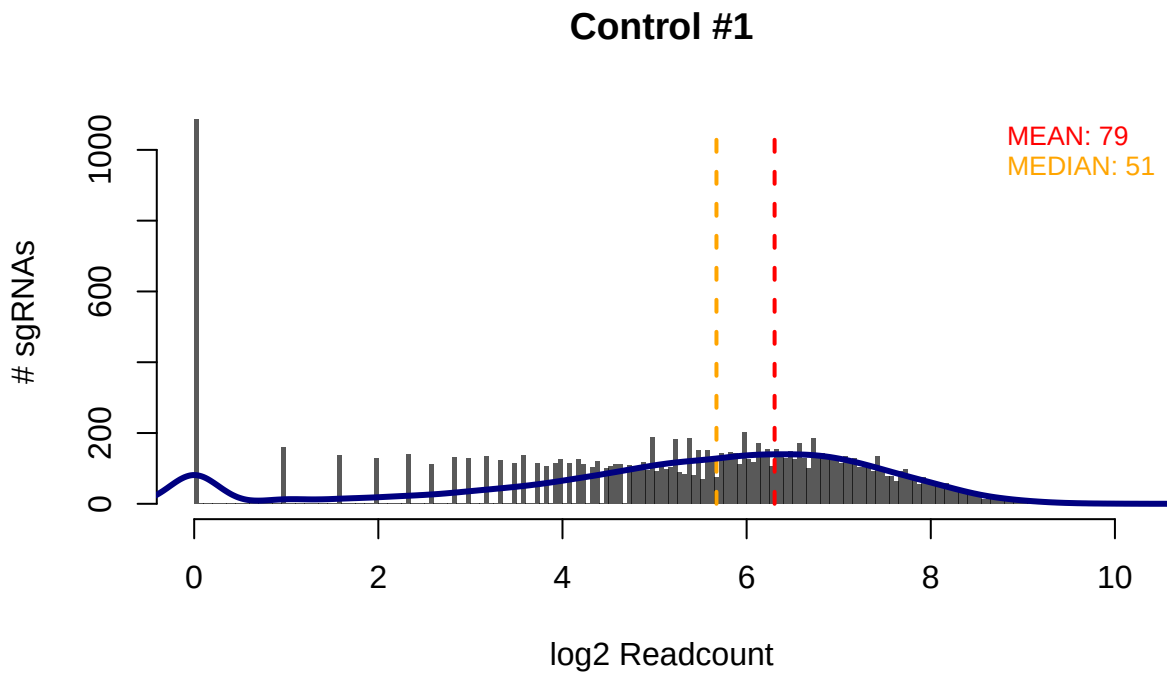
plotgene
You can only plot the read count distribution of sgRNAs belonging to a certain gene, which is given to the function via plotgene.
Default NULL
Value NULL or gene identifier (character)

type
You can plot either the read count distribution either as a normal histogram, or a box-and-whisker plot.
Default "distribution"
Values "distribution" to plot a histogram, or "whisker" to plot a whisker plot (character)

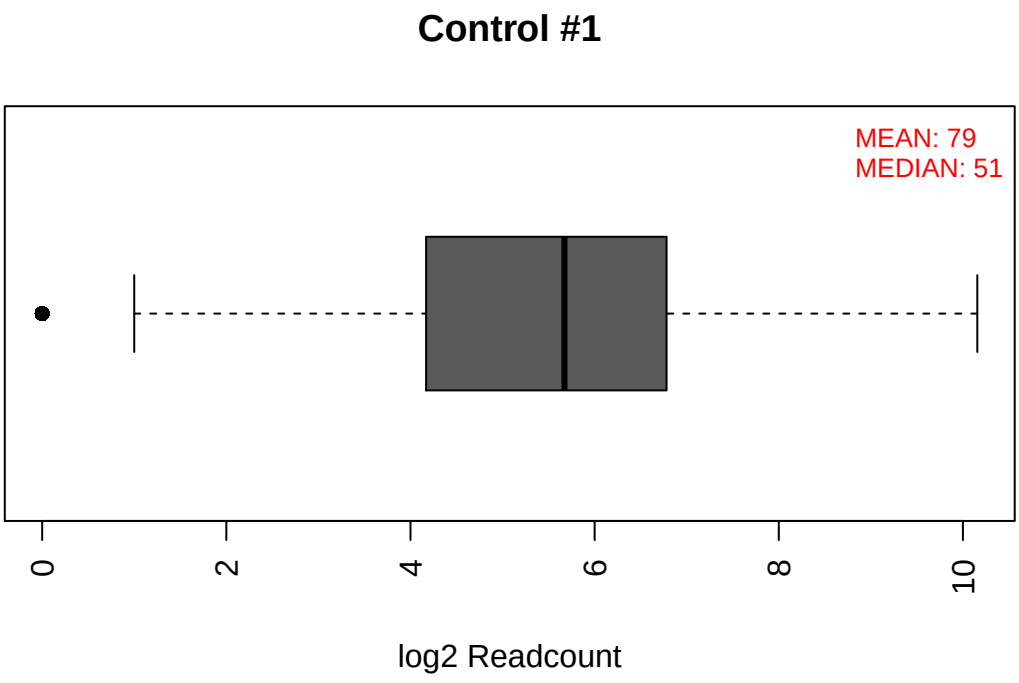
logscale
Indicates whether the read-count is plotted in a logarithmic scale.
Default TRUE
Values TRUE, FALSE (boolean)

5.2.3.1.1 Example for Read-Count Distribution

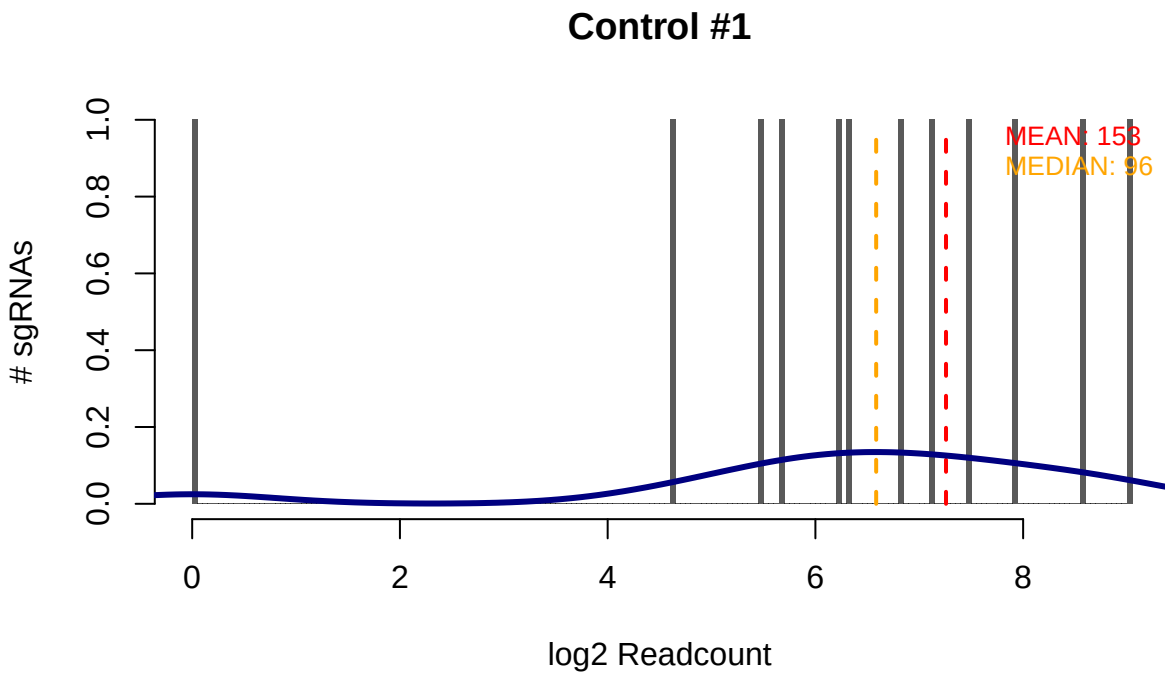
```
# Histogram
carpools.read.distribution(CONTROL1, fullmatchcolumn=2,breaks=200,
  title=d.CONTROL1, xlab="log2 Readcount", ylab="# sgRNAs",statistics=TRUE)
```




```
# Whisker
carpools.read.distribution(CONTROL1, fullmatchcolumn=2,breaks=200,
  title=d.CONTROL1, xlab="log2 Readcount", ylab="# sgRNAs",statistics=TRUE,
  type="whisker")
```




```
# Histogram
carpools.read.distribution(CONTROL1, fullmatchcolumn=2,breaks=200,
  title=d.CONTROL1, xlab="log2 Readcount", ylab="# sgRNAs",statistics=TRUE, plotgene="CASP8")
```



5.2.4 Sequencing Read Depth per sgRNA

You can also visualize the read depth of genes per sgRNA in order to check for sufficient sequencing depth using `carpools.read.depth`.
For further details see `?carpools.read.depth`.
You can either plot single data samples or all four data samples at once.

```
# Plot for a singe sample
carpools.read.depth(datasets = list(CONTROL1), namecolumn=1 ,fullmatchcolumn=2,
  dataset.names=list(d.CONTROL1), extractpattern=expression("^(.+?)_.+"),
  xlab="Genes", ylab="Read Count per sgRNA",statistics=TRUE, labelgenes = NULL,
  controls.target = NULL, controls.nontarget="random")

# Plot for 4 samples at once
carpools.read.depth(datasets = list(CONTROL1,CONTROL2,TREAT1,TREAT2), namecolumn=1,
  fullmatchcolumn=2, dataset.names=list(d.CONTROL1,d.CONTROL2,d.TREAT1,d.TREAT2),
  extractpattern=expression("^(.+?)_.+"), xlab="Genes", ylab="Read Count per sgRNA",
  statistics=TRUE, labelgenes = NULL, controls.target = NULL, controls.nontarget="random")
```

5.2.4.1 carpools.read.depth()

Usage

```
carpools.read.depth(datasets, namecolumn=1, fullmatchcolumn=2, dataset.names=NULL, extractpattern=expres
col=rgb(0, 0, 0, alpha = 0.65), xlab="Genes", ylab="Read Count per sgRNA", statistics=TRUE,
labelgenes = NULL, controls.target = controls.target, controls.nontarget=controls.nontarget,
labelcolor="orange", waterfall=FALSE)
```

dataset A list of data frames of read-count data as created by load.file().
Default none
Values A list of data frames

namecolumn
In which column are the sgRNA identifiers?

Default 1
Values column number (numeric)

fullmatchcolumn
In which column are the read counts?
Default 2
Values column number (numeric)

dataset.names
A list of names that must be according to the list of data sets given in *dataset*.
Default NULL
Value NULL or list of data names (list)

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier.
e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

col
The color of the plotted data. Can be any R color or RGB object. See ?rgb() for further information.
Default rgb(0, 0, 0, alpha = 0.65)
Values Any R color name or RGB color object (character OR color object)

xlab
Label of X-Axis
Default "X-Axis"
Values "Label of X-Axis" (character)

ylab
Label of Y-Axis
Default "Y-Axis"
Values "Label of Y-Axis" (character)

statistics
Whether basic statistics will be shown in the plot.
Default TRUE
Values TRUE, FALSE (boolean)

labelgenes
You can highlight certain genes within the plot. This expects a gene identifier or a vector of gene identifiers.
Default NULL
Values A gene identifier or vector of gene identifiers (character)

labelcolor
Color to highlight genes stated in `labelgenes`.
Default "orange"
Values Any R color or RGB color object.

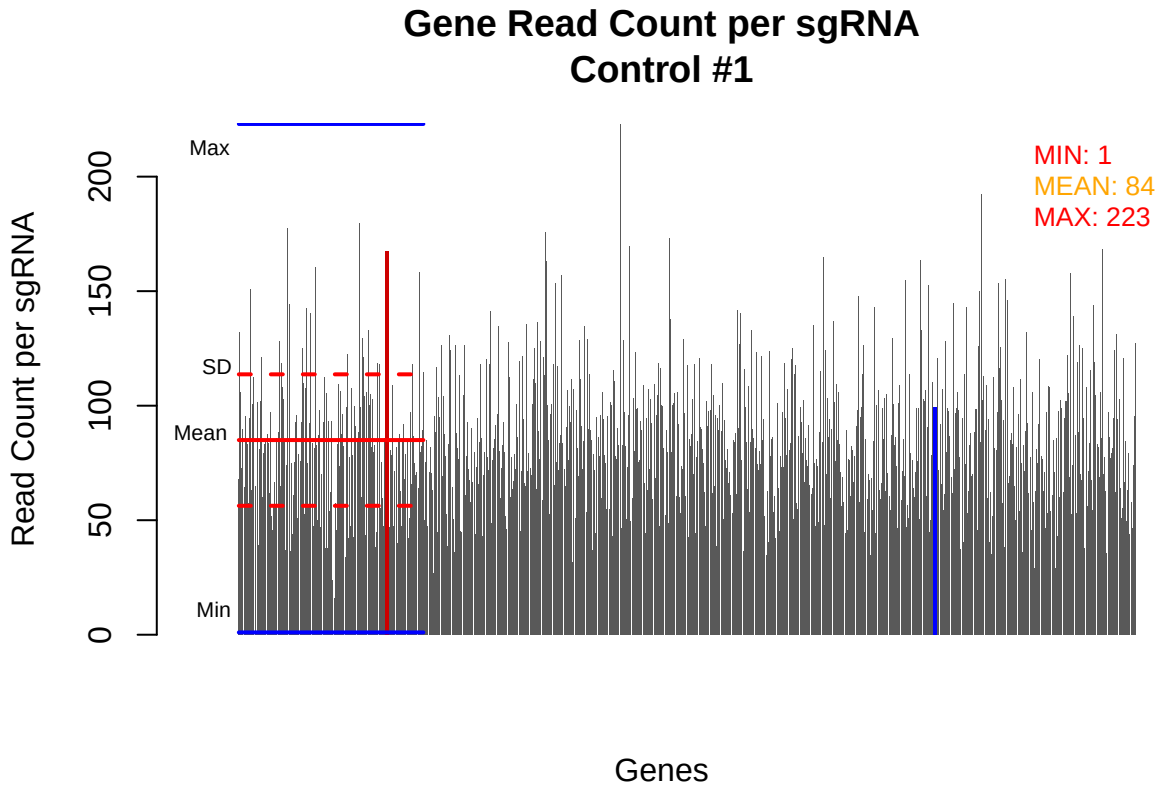
controls.target
Highlights the positive control in red color.
Default NULL
Value Gene Identifier (character)

controls.nontarget
Highlights the non-targeting control in blue color.
Default "random"
Value Gene Identifier (character)

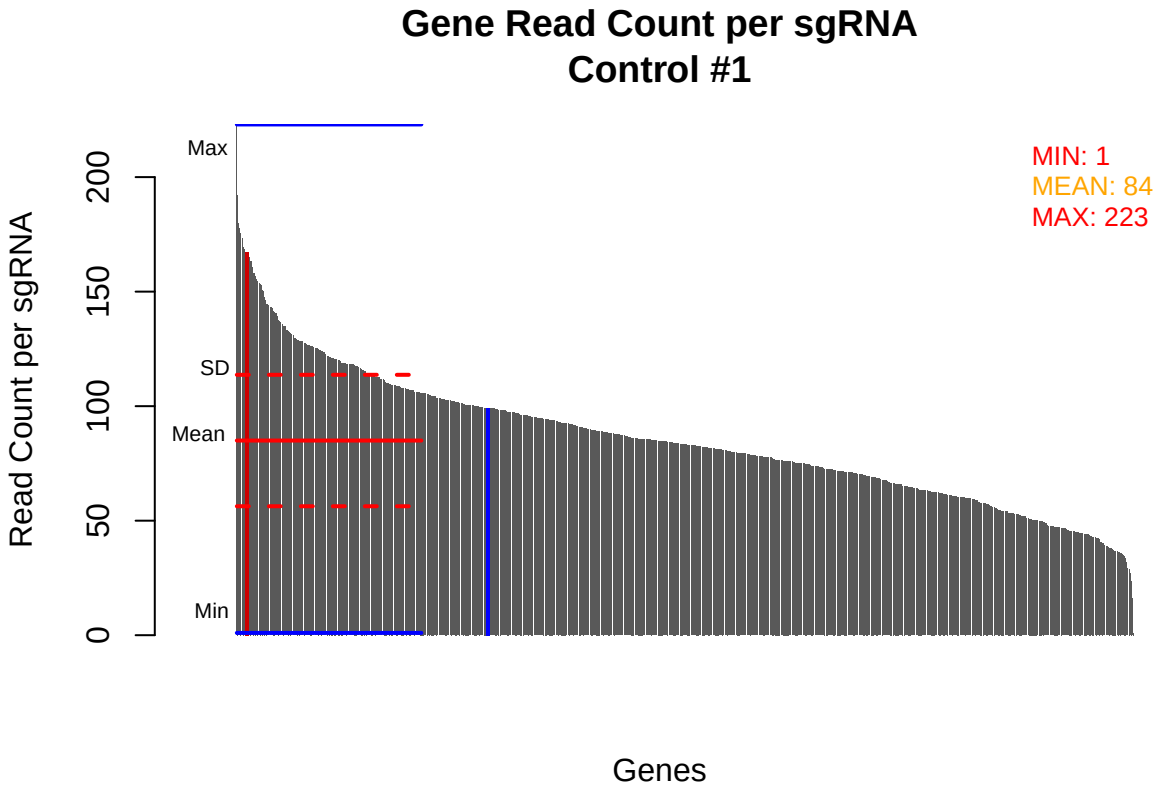
waterfall
You can either plot the read depth sorted by gene identifier (FALSE, default) or according to the read depth.
Default FALSE
Values TRUE, FALSE (boolean)

5.2.4.1.1 Example for Read Depth

```
# Plot for a singe sample
carpools.read.depth(datasets = list(CONTROL1), namecolumn=1 ,fullmatchcolumn=2,
  dataset.names=list(d.CONTROL1), extractpattern=expression("^(.+?)_.+"),
  xlab="Genes", ylab="Read Count per sgRNA",statistics=TRUE, labelgenes = NULL,
  controls.target = "CASP8", controls.nontarget="random", waterfall=FALSE)
```




```
# Plot for a singe sample sorted to read depth
carpools.read.depth(datasets = list(CONTROL1), namecolumn=1 ,fullmatchcolumn=2,
  dataset.names=list(d.CONTROL1), extractpattern=expression("^(.+?)_.+"),
  xlab="Genes", ylab="Read Count per sgRNA",statistics=TRUE, labelgenes = NULL,
  controls.target = "CASP8", controls.nontarget="random", waterfall=TRUE)
```



5.2.5 sgRNAs per Gene Target

Since in most cases several sgRNAs are used to target a gene, the information how many sgRNAs are present in the data for each gene is of interest to make sure the number of sgRNAs present is still sufficient. Typically, only few sgRNAs should get "lost" during the screening procedure, so that the full sgRNA coverage is maintained throughout the assay. The only exception would be drop-out screens with a stringent setup.
The representation of sgRNAs per gene can be plotted using `carpools.reads.genedesigns`.
For further details see `?carpools.reads.genedesigns`.

```
# Plot sgRNA presence per gene for control 1
control1.readspergene = carpools.reads.genedesigns(CONTROL1, namecolumn=1, fullmatchcolumn=2, title=pas
```

5.2.5.1 carpools.reads.genedesigns()

Usage

```
carpools.reads.genedesigns(dataset, namecolumn=1, fullmatchcolumn=2, title="Read Count",
xlab="% of sgRNAs present", ylab="# of Genes", agg.function=sum, extractpattern=expression("^(.+?)_.+")
col = rgb(0, 0, 0, alpha = 0.65))
```

dataset A data frame of read-count data as created by load.file().
Default none
Values a data frame

namecolumn
In which column are the sgRNA identifiers?
Default 1
Values column number (numeric)

fullmatchcolumn
In which column are the read counts?

Default 2
Values column number (numeric)

title
The title of the plot.
Default "Read Count"
Values "Any title" (character)

xlab
Label of X-Axis
Default "X-Axis"
Values "Label of X-Axis" (character)

ylab
Label of Y-Axis
Default "Y-Axis"
Values "Label of Y-Axis" (character)

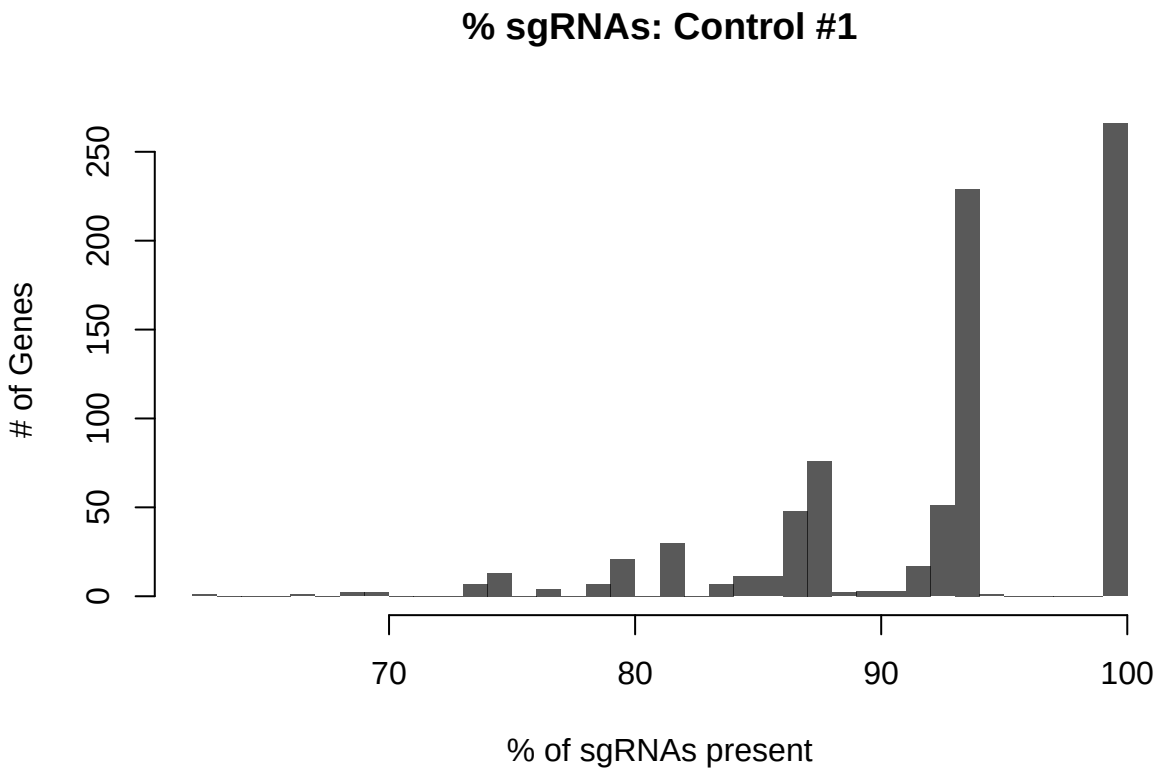
agg.function
The function to aggregate sgRNA read-count. *Default* sum
Values any mathematical function (function)

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier.
e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

col
The color of the plotted data. Can be any R color or RGB object. See ?rgb() for further information.
Default rgb(0, 0, 0, alpha = 0.65)
Values Any R color name or RGB color object (character OR color object)

5.2.5.1.1 Example for Representation of sgRNAs per Gene

```
control1.readspergene = carpools.reads.genedesigns(CONTROL1, namecolumn=1, fullmatchcolumn=2, title=pas
```



5.2.6 Plotting Read-Counts

CaRpools also allows you to compare the read-count for different samples using `carpools.read.count.vs`. By this, you can easily compare the screen and replicate performance as well as highlighting your non-targeting or positive controls. Moreover, you can highlight any gene as well.
For details regarding all arguments and option see `?carpools.read.count.vs`.

```
# Highlight the non-targeting control in tweo dataset
carpools.read.count.vs(dataset=list(TREAT1,CONTROL1), dataset.names = c(d.TREAT1, d.CONTROL1),
  pairs=FALSE, namecolumn=1, fullmatchcolumn=2, title="", pch=16,
  normalize=TRUE, norm.function="median", labelgenes="random", labelcolor="blue",
  center=FALSE, aggregated=FALSE)
# Highlight non-targeting control in all datasets
carpools.read.count.vs(dataset=list(TREAT1, TREAT2, CONTROL1, CONTROL2),
  dataset.names = c(d.TREAT1, d.TREAT2, d.CONTROL1, d.CONTROL2), pairs=TRUE, namecolumn=1,
  fullmatchcolumn=2, title="", pch=16, normalize=TRUE, norm.function="median",
  labelgenes="random", labelcolor="blue", center=FALSE, aggregated=FALSE)
```

5.2.6.1 carpools.read.count.vs()

Usage

```
carpools.read.count.vs(dataset, namecolumn=1, fullmatchcolumn=2, title="Read Count",
dataset.names = NULL, xlab="Readcount Dataset1", ylab="Readcount Dataset2", xlim=NULL,
ylim=NULL, pch=16, col = rgb(0, 0, 0, alpha = 0.65), labelgenes=NULL, labelcolor="red",
extractpattern=expression("^(.+?)_.+"), plotline=TRUE, normalize=TRUE, norm.function=median,
offsetplot=1.2, center=FALSE, aggregated=FALSE, pairs=FALSE, type=NULL, identify=FALSE)
```

dataset A list of data frames of read-count data as created by load.file().
Default none

Values A list of data frames

namecolumn
In which column are the sgRNA identifiers?
Default 1
Values column number (numeric)

fullmatchcolumn
In which column are the read counts?
Default 2
Values column number (numeric)

title
The title of the plot.
Default "Read Count"
Values "Any title" (character)

dataset.names
A list of names that must be according to the list of data sets given in *dataset*.
Default NULL
Value NULL or list of data names (list)

xlab
Label of X-Axis, only if `pairs=FALSE`
Default "X-Axis"
Values "Label of X-Axis" (character)

ylab
Label of Y-Axis only if `pairs=FALSE`
Default "Y-Axis"
Values "Label of Y-Axis" (character)

xlim
You can define the x-axis range being plotted, e.g. `c(0,1)`.
Default empty
Values empty or a vector with the lower and upper limit.

ylim
You can define the y-axis range being plotted, e.g. `c(0,1)`.
Default empty
Values empty or a vector with the lower and upper limit.

pch
The type of point used in the plot. See `?par()`.
Default 16
Values Any number describing the point, e.g. 16 (numeric)

col
The color of the plotted data. Can be any R color or RGB object. See ?rgb() for further information.
Default rgb(0, 0, 0, alpha = 0.65)
Values Any R color name or RGB color object (character OR color object)

labelgenes
You can highlight certain genes within the plot. This expects a gene identifier or a vector of gene identifiers.
Default NULL
Values A gene identifier or vector of gene identifiers (character)

labelcolor
Color to highlight genes stated in `labelgenes`.
Default "orange"
Values Any R color or RGB color object.

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier.
e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

plotline
You can draw additional lines indicating a fold change of 0, 2, 4.
Default TRUE
*Values** TRUE, FALSE (boolean)

normalize
Whether you would like to normalize read-counts first. Recommended if not done already.
Default TRUE
Values TRUE, FALSE (boolean)

norm.function
The mathematical function to normalize data if `normalize=TRUE`. By default, the median is used.
Default median
Values Any mathematical function of R (function)

offsetplot
Offetplot is used to stretch the x- and y-axis for nicer graphs. This will extend plotting area by offsetplot.
Default 1.2 (Plotting area is stretched to 1.2 times)
Values any number (numeric)

center
If you like you can center your data within the plot.
Default FALSE
Values TRUE, FALSE (boolean)

aggregated
If you want to highlight genes, set this to true if you provide already aggregated gene read count instead of sgRNA read counts.
Default FALSE
Values TRUE, FALSE (boolean)

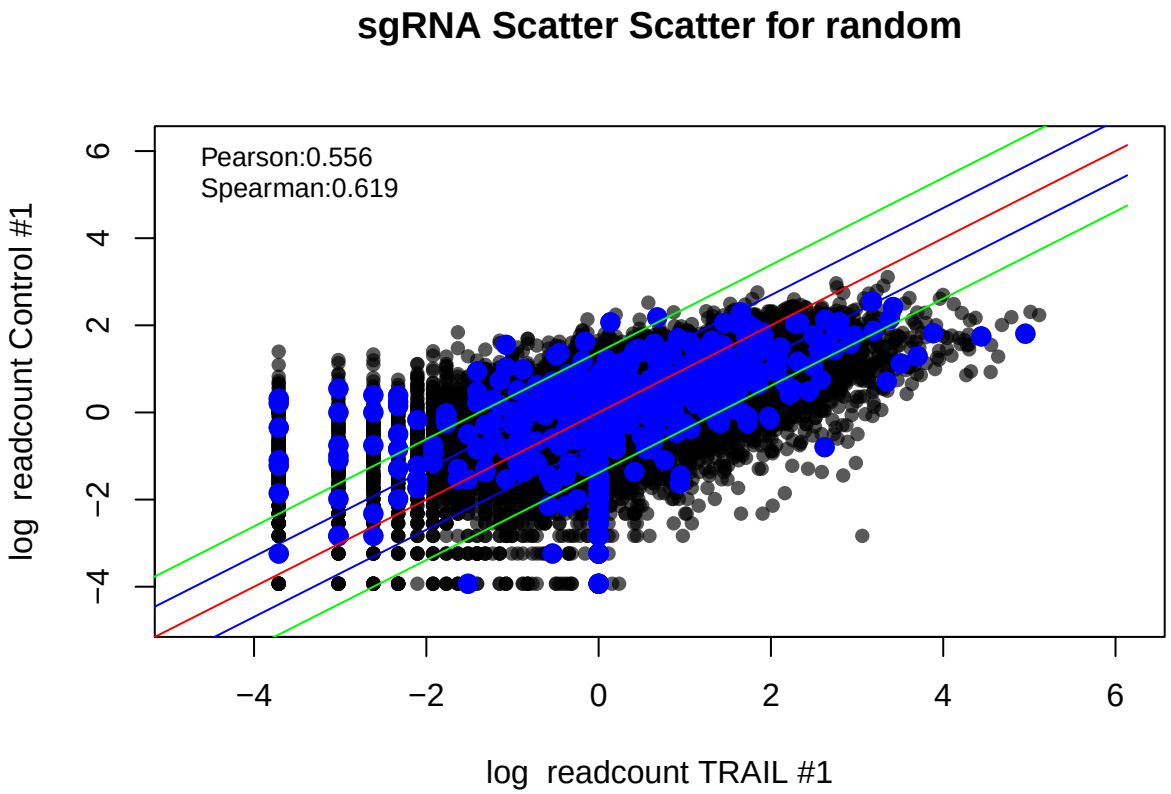
pairs
In the case of plotting all four data sets at once, you can use a pairs plot for easier overview (see `?pairs()`).
Default FALSE
Values TRUE, FALSE (boolean)

type
This indicates whether you would like to color all highlighted genes in either red ("enriched") or blue ("depleted") color according to the standards in caRpools for plotting enriched or depleted genes after analysis.
Default NULL
Values NULL, "enriched", "depleted"

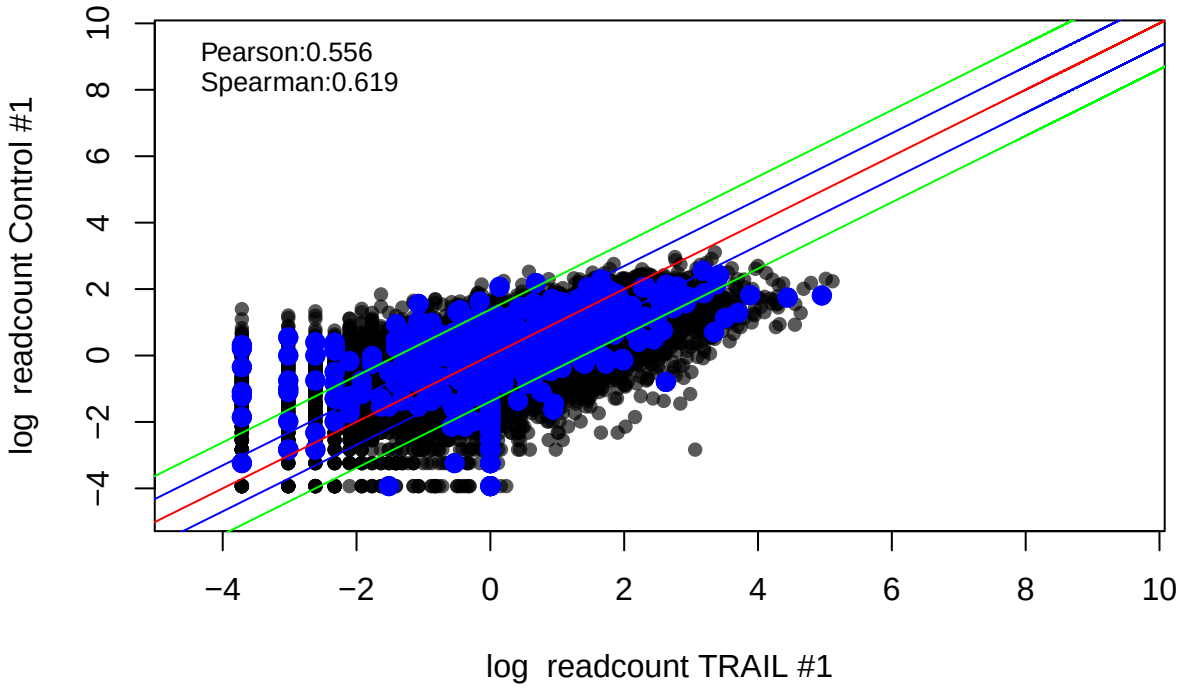
identify
You can ask R to let you identify genes by clicking on the dots in the graph. This only works if `pairs=FALSE`.
Default FALSE
Values TRUE, FALSE (boolean)

5.2.6.1.1 Example Read-Count Control vs. Treated

```
# Highlight the non-targeting control in two datasets
carpools.read.count.vs(dataset=list(TREAT1,CONTROL1), dataset.names = c(d.TREAT1, d.CONTROL1),
  pairs=FALSE, namecolumn=1, fullmatchcolumn=2, title="", pch=16,
  normalize=TRUE, norm.function=median, labelgenes="random", labelcolor="blue",
  center=FALSE, aggregated=FALSE)
```



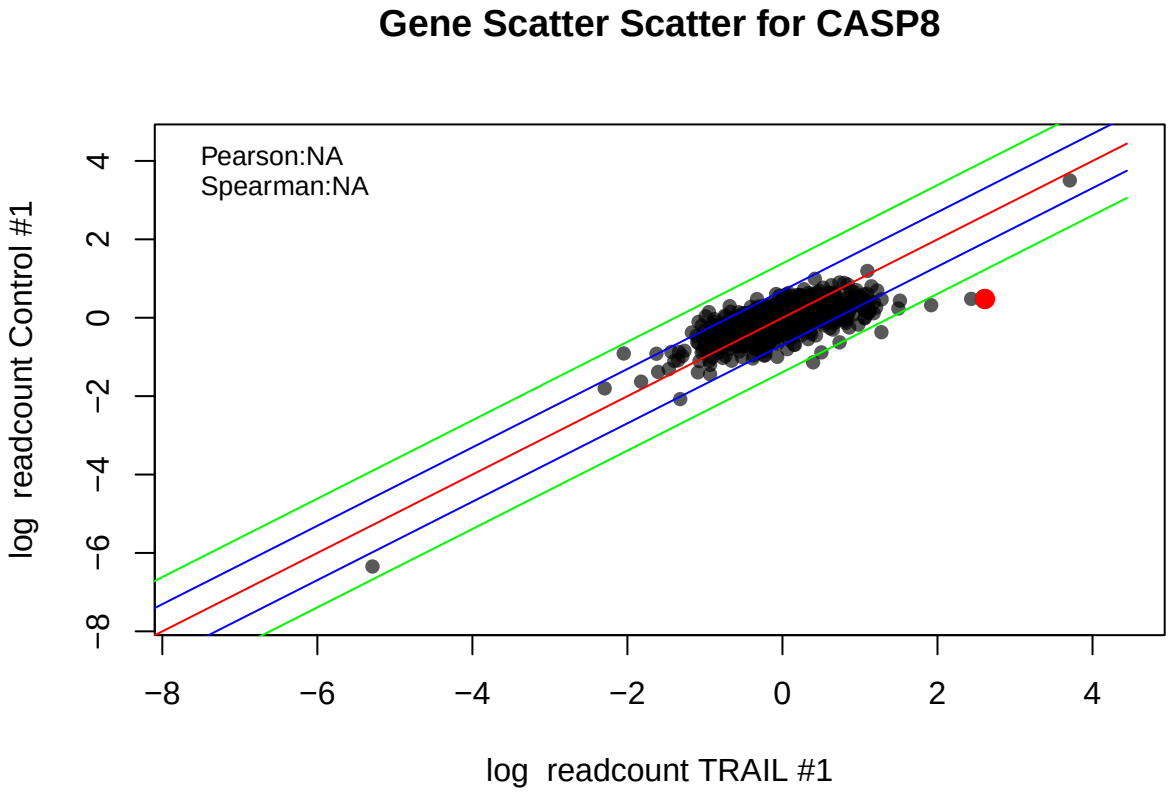

```
# Highlight the non-targeting control in two datasets with CENTERING
carpools.read.count.vs(dataset=list(TREAT1,CONTROL1), dataset.names = c(d.TREAT1, d.CONTROL1),
  pairs=FALSE, namecolumn=1, fullmatchcolumn=2, title="", pch=16,
  normalize=TRUE, norm.function=median, labelgenes="random", labelcolor="blue",
  center=TRUE, aggregated=FALSE)
```




```
# Highlight the targeting control in two datasets on GENE level
carpools.read.count.vs(dataset=list(TREAT1.g,CONTROL1.g), dataset.names = c(d.TREAT1, d.CONTROL1),
  pairs=FALSE, namecolumn=1, fullmatchcolumn=2, title="", pch=16,
  normalize=TRUE, norm.function=median, labelgenes="CASP8", labelcolor="red",
  center=FALSE, aggregated=TRUE)
```

```
## Warning in cor(plot.dataset[plot.dataset[, "designs"] %in% labelgenes,
## "dataset1"], : Standardabweichung ist Null
```

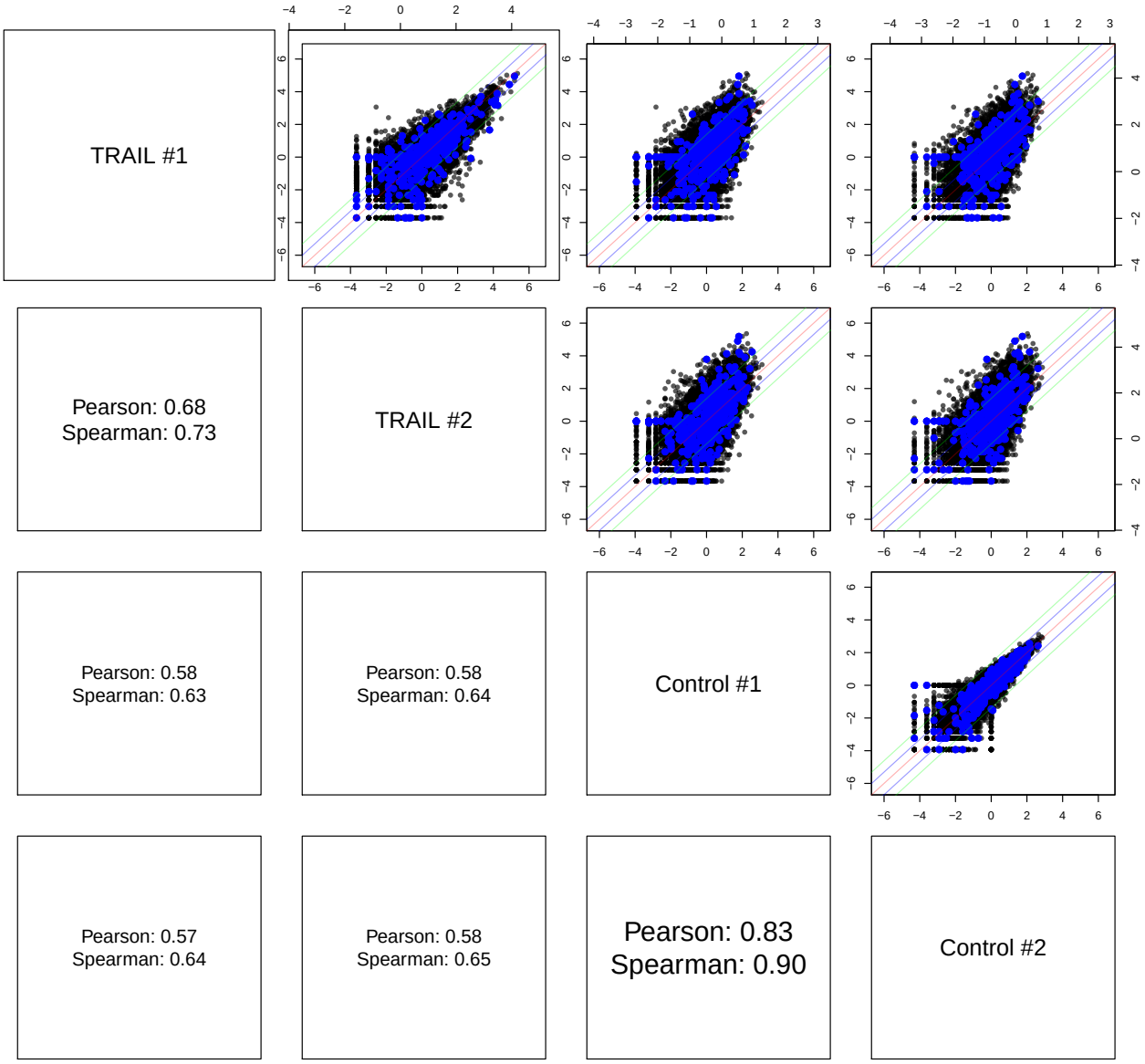
```
## Warning in cor(plot.dataset[plot.dataset[, "designs"] %in% labelgenes,
## "dataset1"], : Standardabweichung ist Null
```



5.2.6.1.2 Example Read-Count Using ALL Data Sets

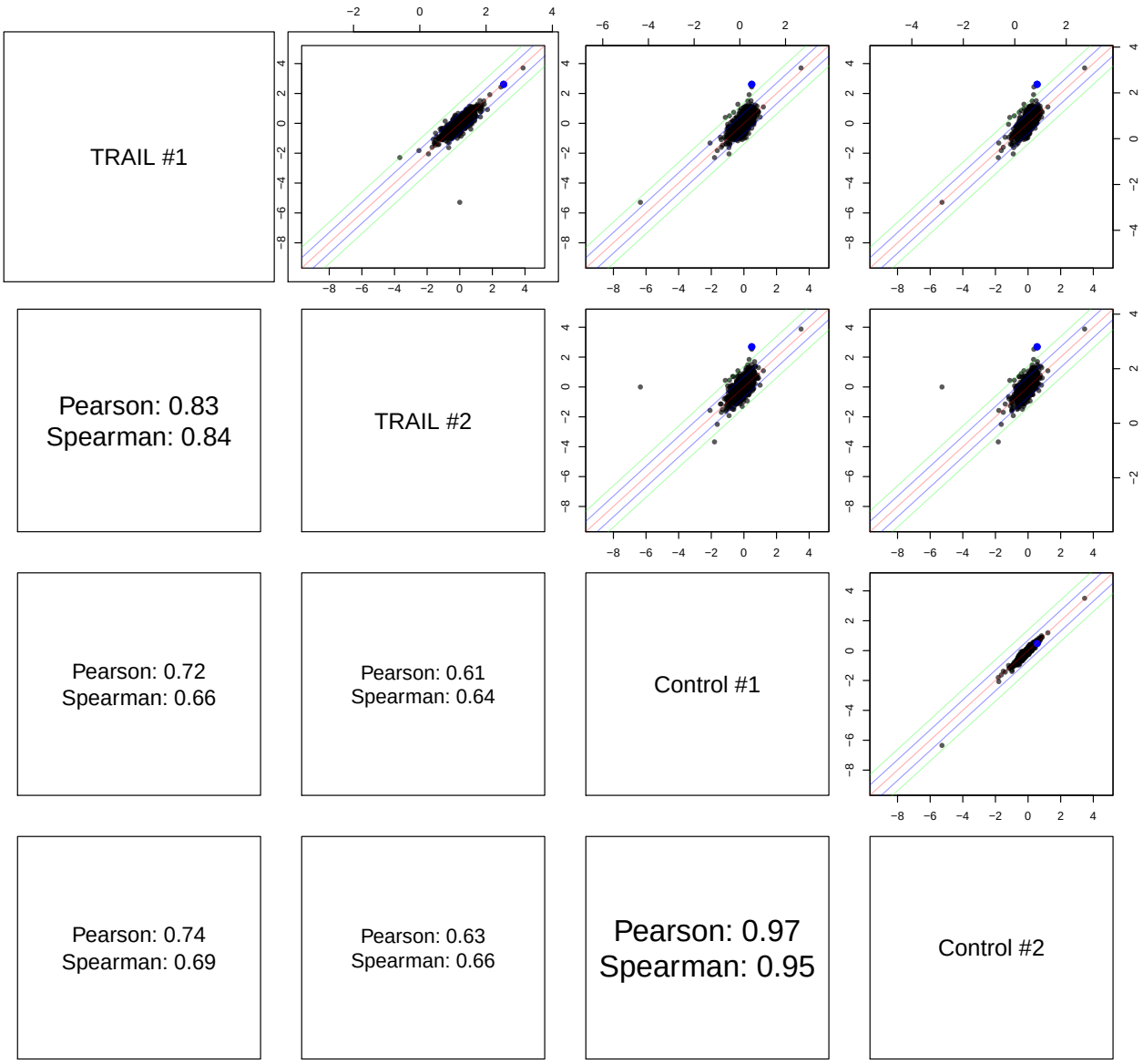
```
# Highlight non-targeting control in all datasets
carpools.read.count.vs(dataset=list(TREAT1, TREAT2, CONTROL1, CONTROL2),
  dataset.names = c(d.TREAT1, d.TREAT2, d.CONTROL1, d.CONTROL2), pairs=TRUE, namecolumn=1,
  fullmatchcolumn=2, title="", pch=16, normalize=TRUE, norm.function=median,
  labelgenes="random", labelcolor="blue", center=FALSE, aggregated=FALSE)
```

sgRNA Scatter for random




```
# Highlight targeting control in all datasets on Gene Level
carpools.read.count.vs(dataset=list(TREAT1.g, TREAT2.g, CONTROL1.g, CONTROL2.g),
  dataset.names = c(d.TREAT1, d.TREAT2, d.CONTROL1, d.CONTROL2), pairs=TRUE, namecolumn=1,
  fullmatchcolumn=2, title="", pch=16, normalize=TRUE, norm.function=median,
  labelgenes="CASP8", labelcolor="blue", center=FALSE, aggregated=TRUE)
```

Gene Scatter for CASP8



5.2.7 Visualize sgRNA Phenotype

CaRpools also allows you to visualize the phenotypic effects of sgRNA belonging to the same gene via `carpools.raw.genes`.
This includes plotting of sgRNA fold-changes, z-score, z-ratios or read-counts. Moreover, `type="vioplot"` will present fold change data in comparison to the whole data-set and controls.

```
carpools.raw.genes(untreated.list = list(CONTROL1, CONTROL2),
  treated.list = list(TREAT1, TREAT2), genes="CASP8", namecolumn=1,
  fullmatchcolumn=2, norm.function=median, extractpattern=expression("^(.+?)_.+"),
  do.plot=TRUE, log=FALSE, put.names=TRUE, type="foldchange" )
```

5.2.7.1 carpools.raw.genes()

Usage `carpools.raw.genes(untreated.list,treated.list, genes=NULL, namecolumn=1, fullmatchcolumn=2, norm.function=median, extractpattern=expression("^(.+?)_.+"), do.plot=TRUE, log=FALSE, put.names=FALSE, type="foldchange", controls.target= NULL, controls.nontarget=NULL, sort=TRUE, ...)`

untreated.list A list of untreated sample data frames of read-count data as created by load.file().
Default none
Values A list of data frames of the untreated samples

treated.list A list of treated sample data frames of read-count data as created by load.file().
Default none
Values A list of data frames of the treated samples

namecolumn
In which column are the sgRNA identifiers?
Default 1
Values column number (numeric)

fullmatchcolumn
In which column are the read counts?
Default 2
Values column number (numeric)

norm.function
The mathematical function to normalize data if `normalize=TRUE`. By default, the median is used.
Default median
Values Any mathematical function of R (function)

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier.
e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

do.plot
Whether a plot is drawn or only tabular output is returned.
Default TRUE
Values TRUE, FALSE (boolean)

log
Plot in log-scale?
Default FALSE
Values TRUE, FALSE (boolean)

put.names
Do you want the sgRNA identifiers to be plotted?
Default FALSE
Values TRUE, FALSE

type
Provides different types. "foldchange" for log2 foldchange, "readcount" for read-count, "z-score" for Z-scores, "z-ratio" for a Z-ratio or "vioplot" for a log2 FC of sgRNA effects.
Default "foldchange"
Values "foldchange", "readcount", "z-score", "z-ratio", "vioplot"

controls.target
Highlights the positive control in red color.
Default NULL
Value Gene Identifier (character)

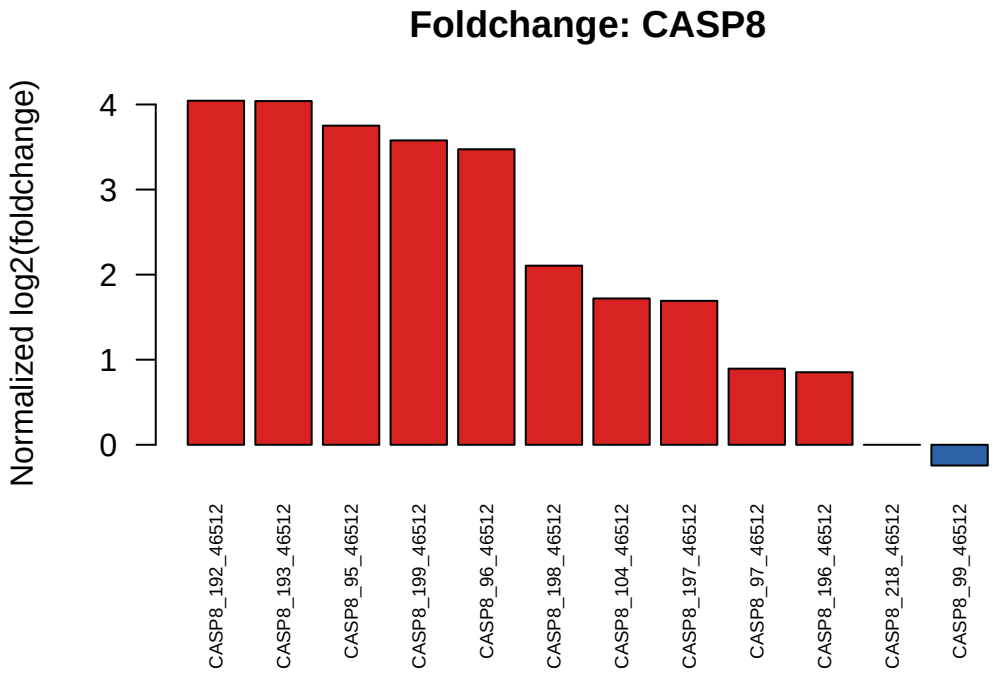
controls.nontarget
Highlights the non-targeting control in blue color.
Default "random"
Value Gene Identifier (character)

sort
This leads to output sorted by fold-change or z-ratio instead of names.
Default TRUE
Values TRUE, FALSE

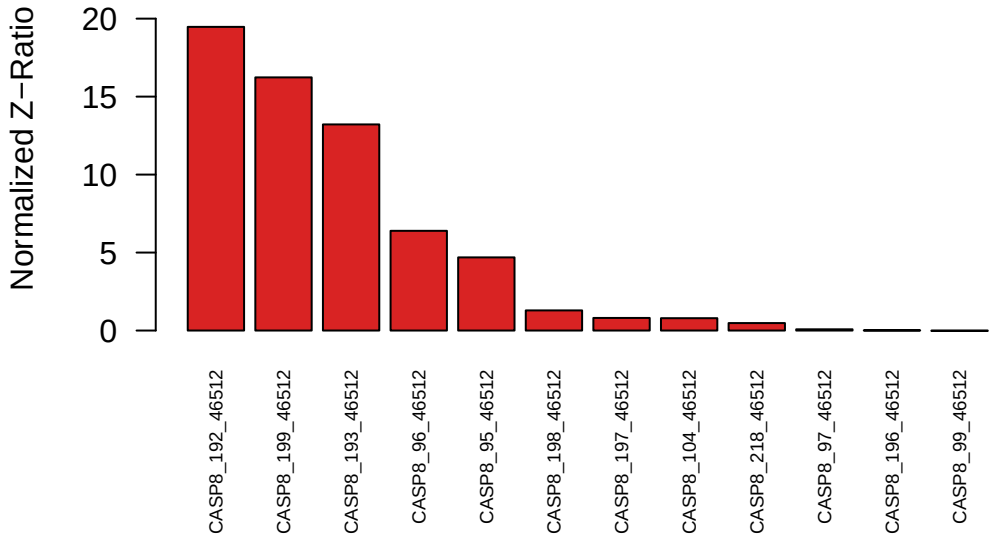
genes
For which gene shall sgRNA effects being plotted?
Default NULL
Values Gene identifier

5.2.7.1.1 Example of sgRNA Phenotype for CASP8

```
# Foldchange
p1 = carpools.raw.genes(untreated.list = list(CONTROL1, CONTROL2),
  treated.list = list(TREAT1, TREAT2), genes="CASP8", namecolumn=1,
  fullmatchcolumn=2, norm.function=median, extractpattern=expression("^(.+?)_.+"),
  do.plot=TRUE, log=FALSE, put.names=TRUE, type="foldchange" )
```

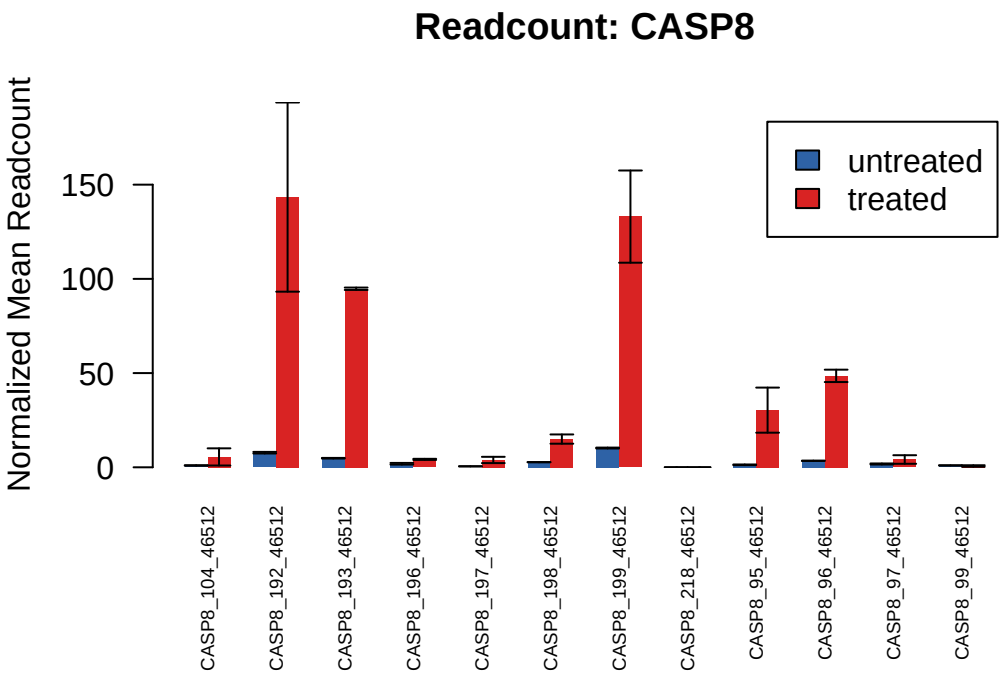



```
# Z-Ratio
p2 = carpools.raw.genes(untreated.list = list(CONTROL1, CONTROL2),
  treated.list = list(TREAT1, TREAT2), genes="CASP8", namecolumn=1,
  fullmatchcolumn=2, norm.function=median, extractpattern=expression("^(.+?)_.+"),
  do.plot=TRUE, log=FALSE, put.names=TRUE, type="z-ratio" )
```




```
# Read Count
p3 = carpools.raw.genes(untreated.list = list(CONTROL1, CONTROL2),
  treated.list = list(TREAT1, TREAT2), genes="CASP8", namecolumn=1,
  fullmatchcolumn=2, norm.function=median, extractpattern=expression("^(.+?)_.+"),
```

```
  do.plot=TRUE, log=FALSE, put.names=TRUE, type="readcount" )
```




```
# Violine plot
p4 = carpools.raw.genes(untreated.list = list(CONTROL1, CONTROL2),
  treated.list = list(TREAT1, TREAT2), genes="CASP8", namecolumn=1,
  fullmatchcolumn=2, norm.function=median, extractpattern=expression("^(.+?)_.+"),
  do.plot=TRUE, log=FALSE, put.names=TRUE, type="vioplot" )
```

5.3 Hit Calling / Candidate Analysis

Hit analysis is performed using three different methods:

- Wilcox
- DESeq2
- MAGeCK

For each analysis method, separate plots will be created and analysis files will be written.

Please note:

All three methods use a different, orthogonal, approach to analyse your data, as it is explained in more detail below.
It is important to mention that no individual analysis method will perform well in all datasets and CRISPR pooled screens, thus it is important to check the result of each method as it is provided by caRpools.
Take your time and have a deep look at the results of each method for your screen.

Moreover, we highly recommend to iteratively change parameters, which are highlighted in the MIACCS file, so that novice users get a feeling of how the analysis methods perform on their screening data.

As a default, caRpools will provide you with **hit candidates called in all three methods as the most robust ones**. However, **having a closer look at candidates of each individual analysis method is highly recommended**.

Wilcox

The Wilcox approach is thought to do a sgRNA population vs. sgRNA population comparison using a Wilcoxon Test.

Within this approach, the read counts of all sgRNAs in one data-set are first normalized by the function set in the MIACCS file. By default, normalization is done by read count division with the data-set median.
Then, the fold change of each population of sgRNAs for a gene is tested against the population of either the non-targeting controls or randomly picked sgRNAs, as defined by the random picks option within the MIACCS file, using a two-sided Mann-Whitney-U test. P-values are finally corrected for multiple testing using FDR.

DESeq2

In this test, DESeq2 is applied on gene-level to perform analysis on gene-level read-count instead of individual sgRNA read-counts.

For the DESeq2 analysis implementation, the read counts of all sgRNAs for a given gene is aggregated.
Since all read-count files consist of the same number of sgRNAs per gene, a gene-level read-count is created for hit analysis.
Then, DESeq2 analysis is performed, which includes the estimation of size-factors, the variance stabilization using a parametric fit and a Wald-Test for difference in log2 fold changes between the untreated and treated data.
More information about this can be found in *Love et al.*
Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2
Genome Biology 2014

MAGeCK

MAGeCK analysis uses a rank-based model to test for a change in abundance of sgRNAs after median normalization of the data-set.
Further information can be found at the MAGeCK Homepage.

All analysis methods require, if they are to be used with caRpools to generate standardized reports, two biological replicates of the untreated and the treated sample.
By default, *two replicates per condition* are used by the templates provided. If you want to analyse more than two replicates per group, please have a look at *Providing more than two replicates* .

Please note that hit candidate comparison strictly requires the output of all three analysis methods.

All output from `stat.wilcox`, `stat.DEseq` and `stat.mageck` can be used for further analysis, e.g. visualization or hit candidate comparison (see below).

5.3.1 Candidate Calling with Wilcox

Wilcox compare the group of sgRNAs for a given gene to either the non-targeting control population or the population of randomly picked sgRNAs from the data-set.

```
data.wilcox = stat.wilcox(untreated.list = list(CONTROL1, CONTROL2),
  treated.list = list(TREAT1,TREAT2), namecolumn=1, fullmatchcolumn=2,
  normalize=TRUE, norm.fun=median, sorting=FALSE, controls="random",
  control.picks=NULL)
```

The generated data includes log2 fold-change as well as a FDR corrected p-value for each gene:

	untreated	treated	foldchange	p.value
AAK1	2.082629	3.067542	1.364821	0.9802939
AATK	3.461728	5.239486	1.370082	0.9802939
ABI1	2.999779	4.458453	1.463144	0.9671744
ABL1	2.296034	2.928893	1.221199	0.9802939
ABL2	2.548473	4.635808	1.749940	1.0000000
ACAD10	2.239865	2.413090	1.141630	0.9802939
ACVR1	2.791890	3.321041	1.222589	1.0000000
ACVR1B	2.425693	2.495935	1.020091	0.9820183
ACVR1C	2.448173	2.830851	1.161288	1.0000000
ACVR2A	1.919399	2.594297	1.312544	1.0000000

5.3.1.1 stat.wilcox()

Usage

```
stat.wilcox(untreated.list=list(dataset1.untreated, dataset2.untreated),treated.list=list(dataset1.trea
dataset2.treated), namecolumn=1, fullmatchcolumn=2,normalize=TRUE,norm.fun=median, extractpattern=expres
controls=NULL, control.picks=150, sorting=TRUE)
```

untreated.list A list of untreated sample data frames of read-count data as created by load.file().
Default none
Values A list of data frames of the untreated samples

treated.list A list of treated sample data frames of read-count data as created by load.file().
Default none
Values A list of data frames of the treated samples

namecolumn
In which column are the sgRNA identifiers?
Default 1
Values column number (numeric)

fullmatchcolumn
In which column are the read counts?
Default 2
Values column number (numeric)

normalize
Whether you would like to normalize read-counts first. Recommended if not done already.
Default TRUE
Values TRUE, FALSE (boolean)

norm.function
The mathematical function to normalize data if `normalize=TRUE`. By default, the median is used.

Default median
Values Any mathematical function of R (function)

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier.
e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

controls
Please give the gene identifier of the non-targeting control. If you do not have non-targeting controls, leave NULL.
Default NULL
Values NULL or gene identifier (character)

control.picks
If no non-targeting controls are present or set, wilcox will pick a random number of sgRNAs from the data set as the alternative population. This is only used if `controls=NULL`.
Default 300
Values numeric

5.3.2 Candidate Calling with DESeq2

DESeq2 can also be used to check for enriched or depleted candidates within the screen.
Hit calling using the DESeq2 methods can be performed by passing on the data in replicates as a list to `stat.DESeq`.
Normalization and Variance estimation is performed by DESeq2 automatically, for detailed information please see `?DESeq2`.

```
# DESeq2
data.deseq = stat.DESeq(untreated.list = list(CONTROL1, CONTROL2),
  treated.list = list(TREAT1,TREAT2), namecolumn=1,
  fullmatchcolumn=2, extractpattern=expression("^(.+?)(_.+)"),
  sorting=FALSE, filename.deseq = "ANALYSIS-DESeq2-sgRNA.tab",
  fitType="parametric")
```

The output of `stat.DESeq` is a list with two data frames. One data frame containing the sgRNA calculations (`data.deseq$sgRNA`) and one list (`data.deseq$genes`) with gene calculations.

This output includes log2 fold-changes as well as corrected p-values (padj) for all genes:

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	genes	sgRNA
AAK1	1189.4567	0.2542426	0.1911656	1.3299594	0.1835316	1	AAK1	6
AATK	2562.0309	0.1095068	0.2325419	0.4709120	0.6377036	1	AATK	3
ABI1	1956.9917	0.1185700	0.2093745	0.5663057	0.5711860	1	ABI1	1
ABL1	1173.9579	-0.0806785	0.2741598	-0.2942756	0.7685473	1	ABL1	3
ABL2	1916.5212	0.5124450	0.2352881	2.1779466	0.0294100	1	ABL2	4
ACAD10	1064.3651	-0.4456368	0.2010605	-2.2164317	0.0266620	1	ACAD10	3
ACVR1	1626.3164	-0.2674216	0.1976850	-1.3527662	0.1761303	1	ACVR1	1
ACVR1B	1107.0705	-0.5577862	0.1919997	-2.9051410	0.0036709	1	ACVR1B	3
ACVR1C	1298.2673	-0.3036710	0.1866838	-1.6266599	0.1038093	1	ACVR1C	1
ACVR2A	959.2813	0.1229099	0.1990204	0.6175743	0.5368560	1	ACVR2A	4

5.3.2.1 stat.DEseq()

Usage

```
stat.DESeq(untreated.list,treated.list,namecolumn=1, fullmatchcolumn=2, agg.function=sum,
extractpattern=expression("^(.+?)_.+"), sorting=FALSE, sgRNA.pval = 0.01, filename.deseq="data",
fitType="parametric", p.adjust="holm")
```

untreated.list A list of untreated sample data frames of read-count data as created by load.file().
Default none
Values A list of data frames of the untreated samples

treated.list A list of treated sample data frames of read-count data as created by load.file().
Default none
Values A list of data frames of the treated samples

namecolumn
In which column are the sgRNA identifiers?
Default 1
Values column number (numeric)

fullmatchcolumn
In which column are the read counts?
Default 2
Values column number (numeric)

agg.function
The function to aggregate sgRNA read-count. *Default* sum
Values any mathematical function (function)

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier.
e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

sorting
Sort output by log2 fold change? *Default* FALSE
Values TRUE, FALSE (boolean)

sgRNA.pval
p-value threshold to count significant sgRNAs for each gene.
Default 0.001
Value (numeric)

filename.deseq
Filename of raw DESeq2 data output.
Default "data"
Values (character)

fit.Type
See `?DESeq2`.
Default "parametric"
Values "parametric", "local" "mean"

p.adjust
Method to adjust p-value for multiple testing. See `?DEseq2`.
Default "holm" *Values* see `?DESeq2`

5.3.3 Candidate Calling with MAGeCK

CaRpools also uses MAGeCK to look for enriched or depleted genes within your screening data. Please note that MAGeCK needs to be installed correctly, this can be tested by `check.caRpools`.

```
# MAGeCK
data.mageck = stat.mageck(untreated.list = list(CONTROL1, CONTROL2), treated.list = list(TREAT1,TREAT2)
```

```
## 'data.frame':    12478 obs. of  6 variables:
##  $ designs: chr  "AAK1_104_0" "AAK1_105_0" "AAK1_106_0" "AAK1_107_0" ...
##  $ genes  : chr  "AAK1" "AAK1" "AAK1" "AAK1" ...
##  $ V3     : num  0 395 274 18 0 250 0 183 10 31 ...
##  $ V4     : num  0 597 145 0 0 142 0 60 68 9 ...
##  $ V5     : num  0 197 271 1 0 89 0 11 55 16 ...
##  $ V6     : num  0 309 250 11 0 93 0 27 113 37 ...
```

The output of `stat.mageck` is a list of two data frames.
`data.mageck$sgRNA` contains sgRNA information and `data.mageck$genes` contain ranks and corrected p-values for genes:

	genes	pos	rank.pos	neg	rank.neg	sgrna.neg.good	sgrna.pos.good
TNFRSF13C	TNFRSF13C	0.996031	555	0.701894	1	4	1
DYRK2	DYRK2	0.987129	373	0.701894	2	9	4
CDC42BPG	CDC42BPG	0.996304	803	0.701894	3	9	2
CHEK1	CHEK1	0.996304	777	0.701894	4	6	2
RAPGEF3	RAPGEF3	0.987129	78	0.701894	5	5	6
TNFRSF11A	TNFRSF11A	0.987129	96	0.701894	6	9	1
CSF1R	CSF1R	0.996304	784	0.701894	7	8	1
PTK6	PTK6	0.996304	743	0.701894	8	9	3
BMF	BMF	0.996304	585	0.701894	9	2	4
TESK2	TESK2	0.987129	478	0.701894	10	6	4

5.3.3.1 stat.mageck()

Usage

```
stat.mageck(untreated.list, treated.list, namecolumn=1, fullmatchcolumn=2, norm.fun=median,
extractpattern=expression("^(.+?)_.+"), mageckfolder=NULL, sort.criteria="neg", adjust.method="fdr",
filename=NULL, fdr.pval=0.05)
```

untreated.list A list of untreated sample data frames of read-count data as created by load.file().
Default none
Values A list of data frames of the untreated samples

treated.list A list of treated sample data frames of read-count data as created by load.file().
Default none
Values A list of data frames of the treated samples

namecolumn
In which column are the sgRNA identifiers?
Default 1
Values column number (numeric)

fullmatchcolumn
In which column are the read counts?

Default 2
Values column number (numeric)

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier. e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

sort.criteria
MAGeCK argument *–sort-criteria*
Default "neg"
Values see MAGeCK documentation

mageckfolder
Folder for MAGeCK raw data output (internally used).
Default NULL
Value (character)

filename
Filename of raw MAGeCK data output.
Default "data"
Values (character)

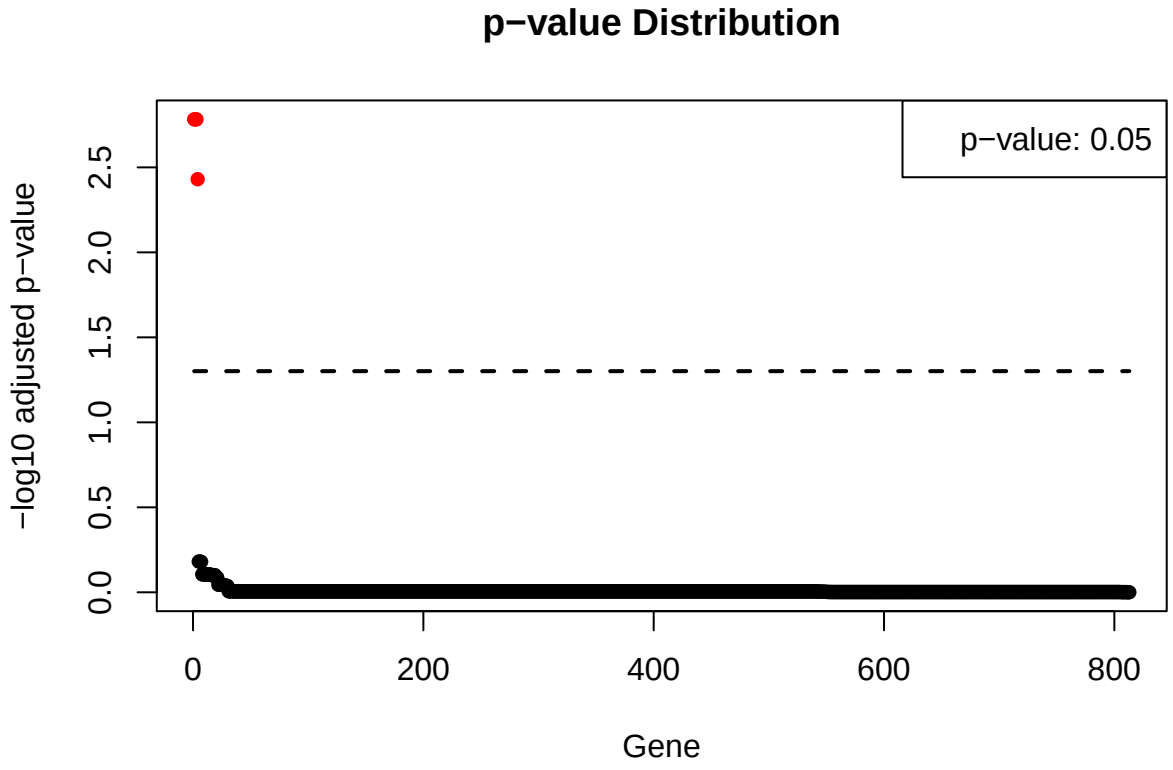
adjust.method
Method to adjust p-value for multiple testing. See MAGeCK documentation.
Default "fdr" *Values* see MAGeCK documentation

fdr.pval
FDR used for correction. *Default* 0.05
Values (numeric)

5.3.4 Visualizing P-Values

Each of the analysis methods returns an adjusted p-value (corrected for multiple testing) as well as a fold change (Wilcox, DESeq2) or gene rank (MAGeCK).
Therefore the -log10 p-value can be plotted against the gene names with `carpools.waterfall.pval`:

```
carpools.waterfall.pval(type="mageck", dataset=data.mageck, pval=0.05, log=TRUE)
```



5.3.4.1 carpools.waterfall.pval()

Usage

```
carpools.waterfall.pval (type=NULL,dataset=NULL, pval=0.05, mageck.type="pos", log=TRUE)
```

type
This indicates which kind of analysis method was used for p-value calculation.
Default NULL
Values “mageck”, “deseq2”, “wilcox”

dataset
Result from either `stat.wilcox`, `stat.DEseq` or `stat.mageck`.
Default NULL
Values Result from either `stat.wilcox`, `stat.DEseq` or `stat.mageck`

pval
The significance value set for the analysis which is to be plotted.
Default 0.05
Values numeric

mageck.type
Only for plotting p-value calculate by MAGeCK. Indicates whether enriched (“pos”) or depleted (“neg”) genes are used.
Default “pos”
Values “pos”, “neg”

log
-log10 of the p-values is plotted if set to TRUE.
Default TRUE
Values TRRUE, FALSE (boolean)

5.3.5 Visualization of Hit Analysis Output

The output from `stat.wilcox`, `stat.DEseq` and `stat.mageck` can be visualized with `carpools.hitident`. In this case, log2 fold changes are plotted against the gene names for all methods as well as the number of significant sgRNAs for data analyzed with DESeq2 or MAGeCK.

```
mageck.result = carpools.hitident(data.mageck, type="mageck",
      title="MAgECK P-values", xlab="Genes", ylab="log10(p-value)",
      print.names=TRUE, offsetplot=1.2, plot.p=0.05, separate=TRUE)
```

Usage
`carpools.hitident(data, type="deseq2", title="DESeq2 plot", print.names=FALSE, cutoff=c(0,0,0,0), inches=0.1, offsetplot=1.2, plot.p=0.01, sgRNA.top=50, separate=FALSE)`

data
Output data from either `stat.wilcox`, `stat.DEseq` or `stat.mageck`.
Default empty
Values Output from either `stat.wilcox`, `stat.DEseq` or `stat.mageck`.

type
Which type of analysis method was used?
Default deseq2
Values "wilcox", "deseq2", "mageck"

title Title of the plot.
Default "DESeq2 plot"
Values (character)

print.names
Shall the names of significant or top candidates being plotted?
Default FALSE
Values TRUE, FALSE (boolean)

cutoff
A vector containing plotting cutoffs if `print.names=TRUE`. c("top enriched", "top depleted", "most sgRNA enriched", "most sgRNA depleted").
Default c(0,0,0,0)
Values Vector of length 4 (numeric)

inches
see `?par`.
Default 0.1
Values (numeric)

offsetplot
Multiplication factor for stretching the plotting area to get a better plot experience.
Default 1.2
Values > 1 (numeric)

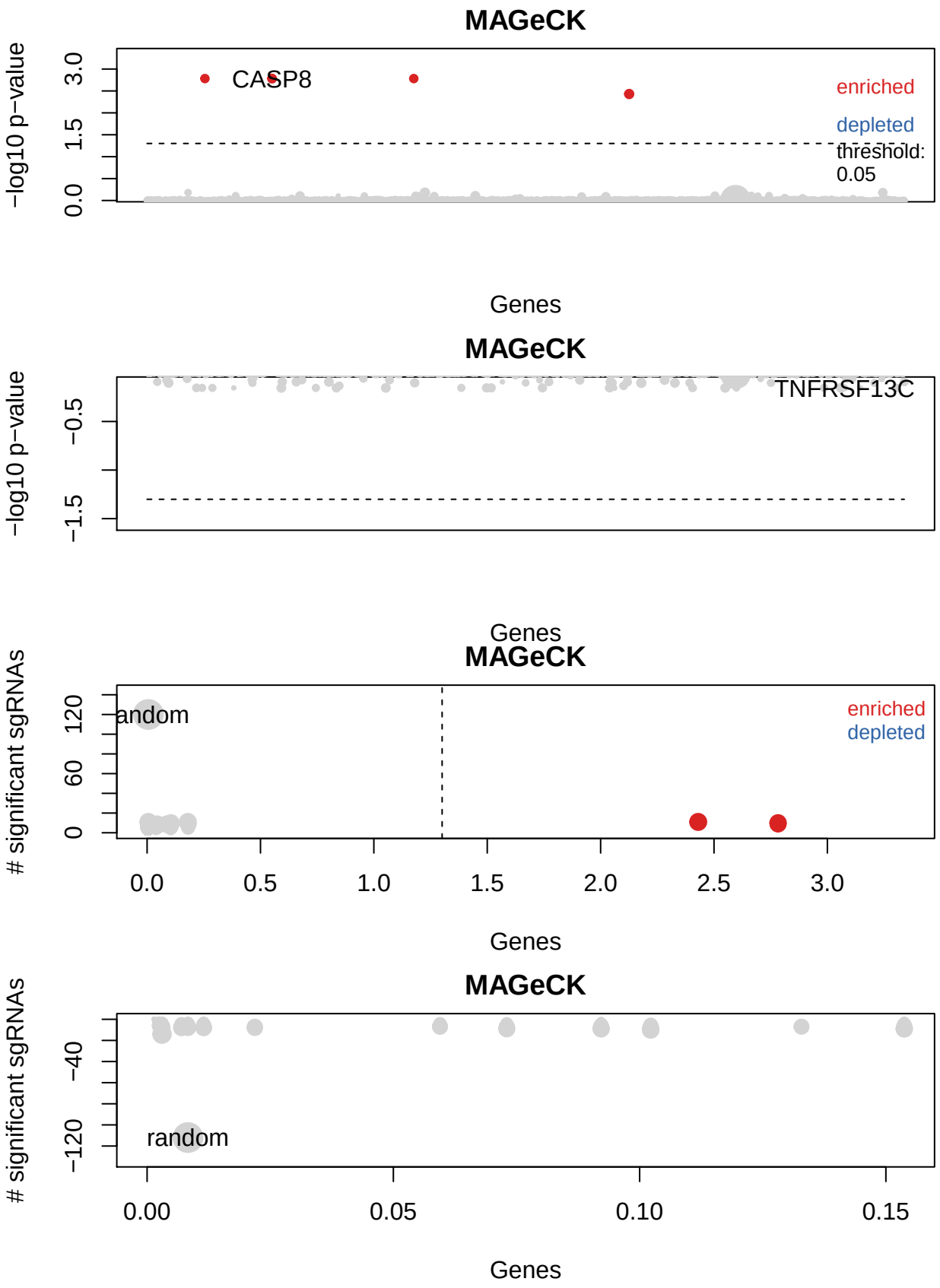
plot.p
Which p-value shall be plotted and used for visualization?
Default 0.05
Values (numeric)

sgRNA.top
For sgRNA plots, this indicates how many genes will be labeled (the top X genes).
Default 1
Values (numeric, integer)

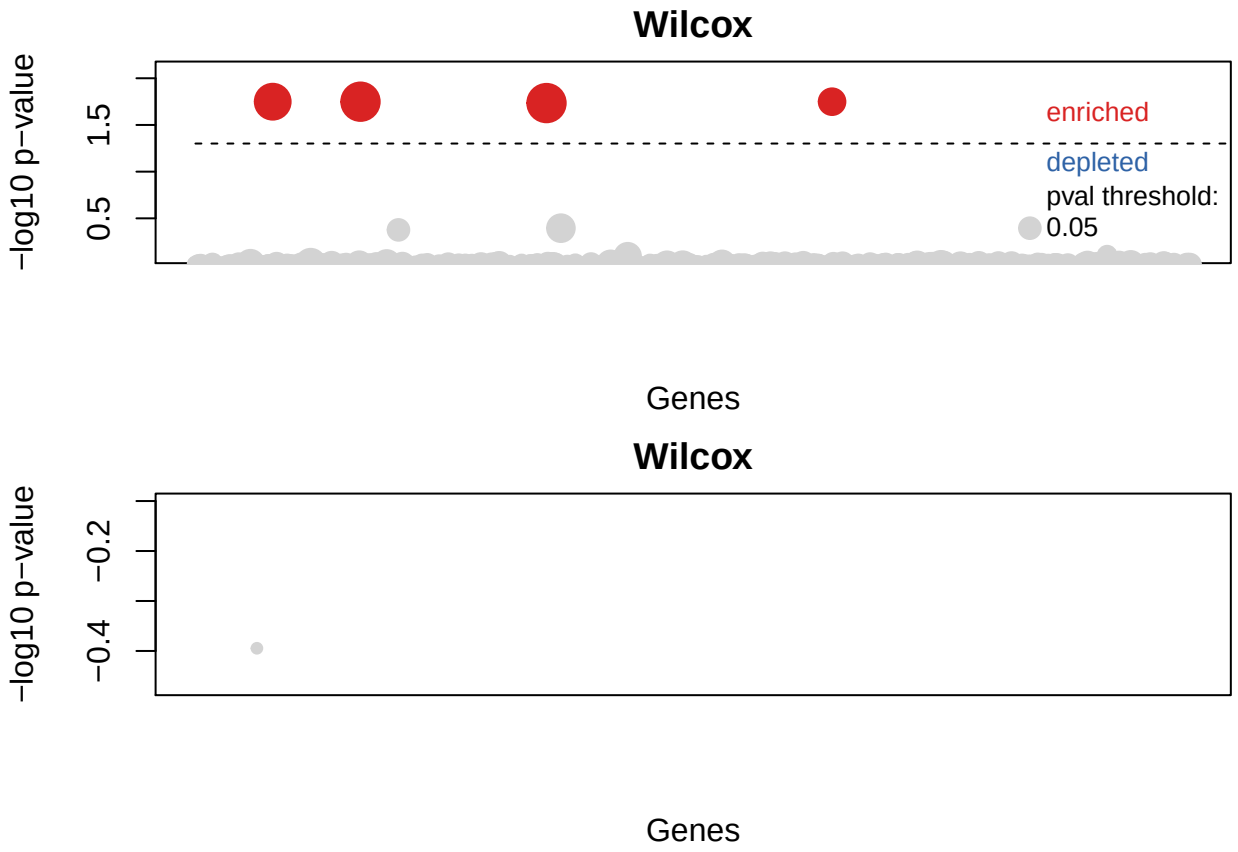
separate
Gene that showed enrichment can be plotted separately from those that have shown a depletion for better overview, works only for wilcox.
Default FALSE
Values TRUE, FALSE

5.3.5.0.1 Examples

```
mageck.result = carpools.hitident(data.mageck, type="mageck", title="MAGeCK", inches=0.1, print.names=T
```


MAGeCK
−log10 p−value
3.0
1.5
0.0
CASP8
enriched
depleted
threshold:
0.05
Genes
MAGeCK
−log10 p−value
−0.5
−1.5
TNFRSF13C
Genes
MAGeCK
significant sgRNAs
120
60
0
andom
enriched
depleted
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Genes
MAGeCK
significant sgRNAs
−40
−120
random
0.00
0.05
0.10
0.15
Genes

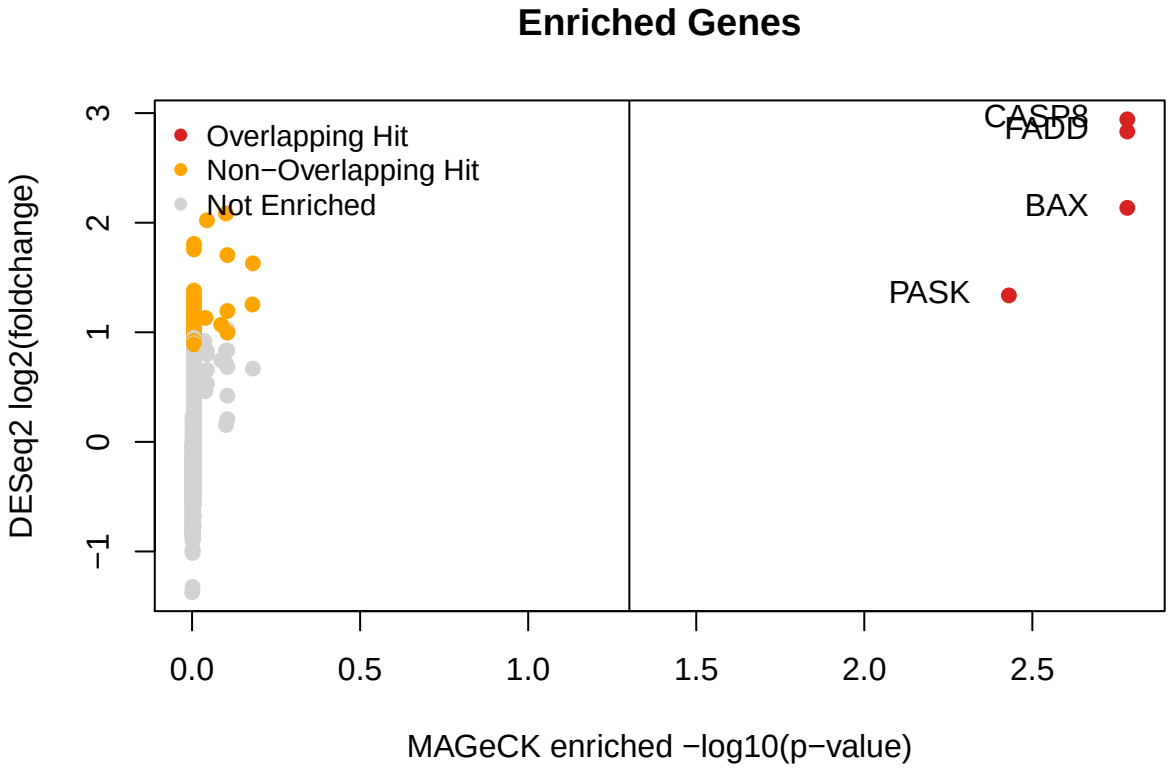
```
wilcox.result = carpools.hitident(data.wilcox, type="wilcox", title="Wilcox", inches=0.1, print.names=T
```



5.3.6 Visualization of Candidate Genes

Candidate genes from all methods can be plotted in an overview to identify overlapping significant candidate genes using `carpools.hit.overview`.

```
carpools.hit.overview(wilcox=data.wilcox, deseq=data.deseq, mageck=data.mageck,
    cutoff.deseq = 0.001, cutoff.wilcox = 0.05, cutoff.mageck = 0.05,
    cutoff.override=FALSE, cutoff.hits=NULL, plot.genes="overlapping", type="enriched")
```



5.3.6.1 carpools.hit.overview()

Usage

```
carpools.hit.overview(wilcox=NULL, deseq=NULL, mageck=NULL, cutoff.deseq = 0.001, cutoff.wilcox
= 0.05, cutoff.mageck = 0.05, cutoff.override=FALSE, cutoff.hits=NULL, plot.genes="overlapping",
type="all")
```

wilcox
Data output from `stat.wilcox`.
Default NULL
Values Data output from `stat.wilcox`.

deseq
Data output from `stat.deseq`.
Default NULL
Values Data output from `stat.deseq`.

mageck
Data output from `stat.mageck`.
Default NULL
Values Data output from `stat.mageck`.

cutoff.deseq
P-Value threshold used to determine significance.
Default 0.001
Values numeric

cutoff.wilcox
P-Value threshold used to determine significance.
Default 0.001
Values numeric

cutoff.mageck
P-Value threshold used to determine significance.
Default 0.001
Values numeric

cutoff.override
Shall the p-value threshold be ignored? If this is TRUE, the top percentage gene of `cutoff.hits` is used instead.
Default FALSE
Values TRUE, FALSE

cutoff.hits
The percentage of top genes being used if `cutoff.override=TRUE`.
*Default** NULL
Values numeric

plot.genes
Defines what kind of data is used. By default, overlapping genes are highlighted in red color.
Default "overlapping"
Values "overlapping"

type
Defines whether all genes are plotted or only those being enriched or depleted.
Default "all"
Values "all", "enriched", "depleted"

5.3.7 Creating VENN Diagrams

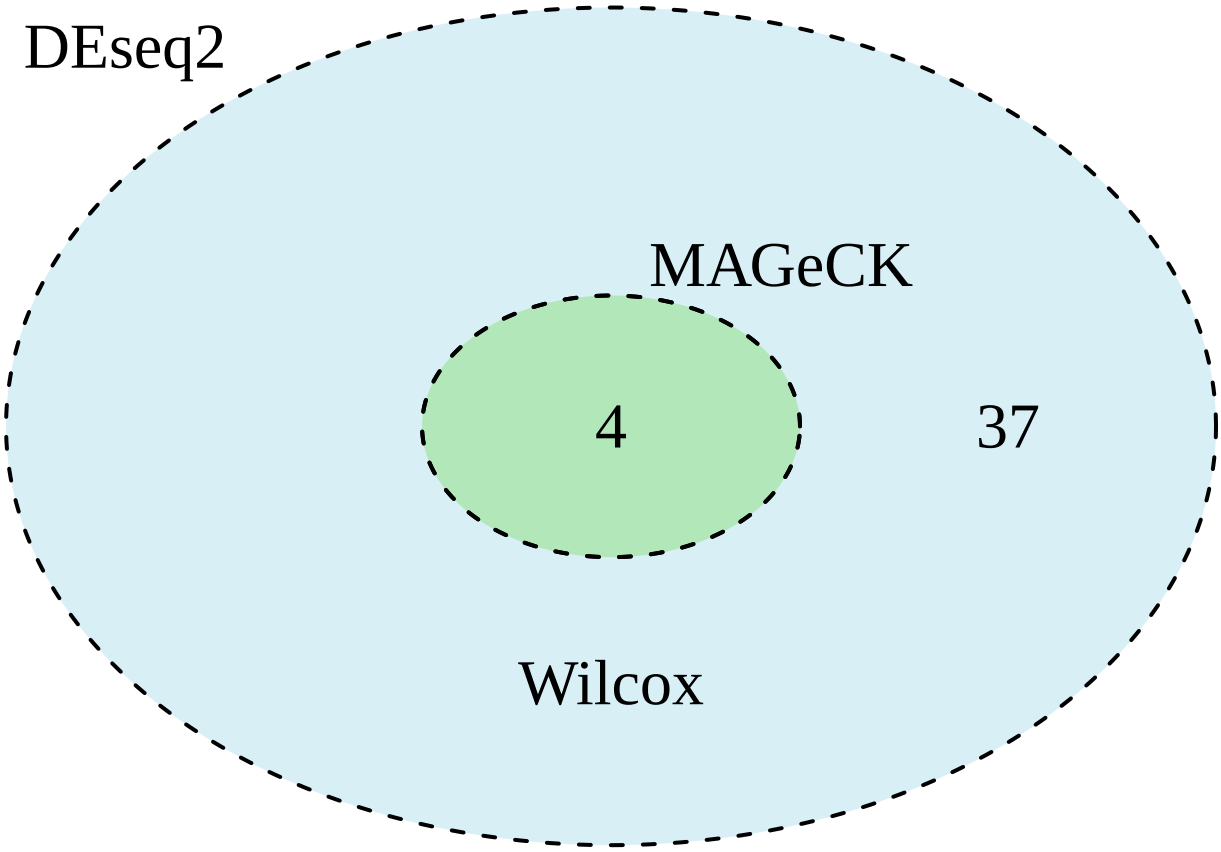
CaRpools can also be used to create Venn Diagrams of the top candidates, thus allowing to see the performance of the analysis methods on your data set using `compare.analysis`.
Therefore, the package `VennDiagram` must be imported.

You can plot a Venn Diagram of all enriched hits like this:

```
venn.enriched = compare.analysis(wilcox=data.wilcox, deseq=data.deseq, mageck=data.mageck, type="enrich
require(VennDiagram)
```

```
## Loading required package: VennDiagram
## Loading required package: grid
## Loading required package: futile.logger
```

```
plot.new()
grid::grid.draw(VennDiagram::venn.diagram(venn.enriched, file=NULL, fill=c("lightgreen","lightblue2","li
```



This diagram shows you, that DESeq2 found 41 significantly enriched genes, out of which 4 where identified both by Wilcox and MAGeCK. Moreover, all found candidates by MAGeCK and Wilcox are the same and included within the DESEq2 ones.

5.4 Export Hit Candidate Lists

Although the candidate lists of each analysis method can be saved separately, caRpools offer a comparative approach, which creates tables that include the information from all analysis methods at once for a faster overview.

This is done using the function `compare.analysis`, which offers not only output for Venn Diagrams, but also for tables.

5.4.0.1 Tabular Output with all information

You can store the information of all enriched genes with the fold change and p-value information at once. Moreover, you can select how it is sorted:

```
# Perform the comparison
data.analysis.enriched = compare.analysis(wilcox=data.wilcox,
    deseq=data.deseq, mageck=data.mageck, type="enriched",
    cutoff.override = FALSE, cutoff.hits=NULL, output="list",
    sort.by=c("mageck","fdr","rank"))
## Write to a file
xlsx::write.xlsx(data.analysis.enriched,
    file="COMPARE-HITS.xls",
    sheetName="Enriched")
# Print to console
knitr::kable(data.analysis.enriched[1:10,c(2:7)])
```

	wilcox.log2fc	wilcox.pval	deseq.log2fc	deseq.pval	mageck.fdr	mageck.rank
CASP8	2.8382733	0.0178241	2.9429395	0.0000000	0.001650	1
BAX	2.4706303	0.0178241	2.1350701	0.0000000	0.001650	2
FADD	2.8332735	0.0184341	2.8317727	0.0000000	0.001650	3
PASK	1.3650980	0.0178241	1.3366397	0.0000000	0.003713	4
TYRO3	1.0116807	0.9671744	0.6686189	0.3923318	0.659241	5
AMHR2	1.4654422	0.9671744	1.6291193	0.0000000	0.659241	6
FN3K	1.4685534	0.4030853	1.2532816	0.0000000	0.661245	7
PRKY	1.2424191	0.9671744	1.1942455	0.0000001	0.785149	8
CDK5RAP3	0.8994121	0.4202992	0.6831604	0.0938068	0.785149	9
ROR2	0.6903593	0.9671744	0.4210292	1.0000000	0.785149	10

By default, data is sorted by the gene rank MAGeCK uses.

5.4.0.2 Rank-based Output

Moreover, you can create a table which gives you the rank of a gene within each analysis method rather than p-value or fold changes. You can do this by choosing `output="rank"`.

```
# Perform the comparison
data.analysis.enriched = compare.analysis(wilcox=data.wilcox,
    deseq=data.deseq, mageck=data.mageck, type="enriched",
    cutoff.override = FALSE, cutoff.hits=NULL, output="rank",
    sort.by=c("mageck","fdr","rank"))
## Write to a file
xlsx::write.xlsx(data.analysis.enriched,
```

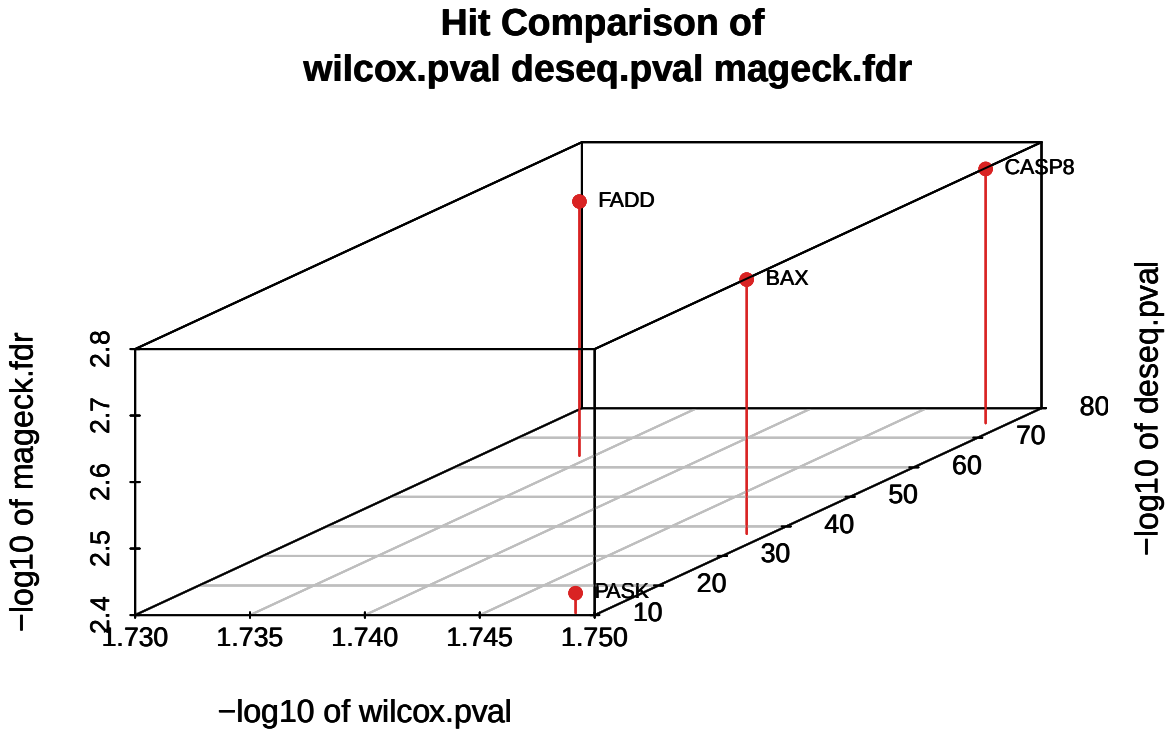
```
    file="COMPARE-RANKs.xls",
    sheetName="Enriched")
# Print to console
knitr::kable(data.analysis.enriched[1:10,])
```

	wilcox	deseq	mageck
CASP8	1	1	1
BAX	3	3	2
FADD	2	2	3
PASK	11	12	4
TYRO3	36	85	5
AMHR2	6	9	6
FN3K	5	16	7
PRKY	16	19	8
CDK5RAP3	60	83	9
ROR2	131	170	10

5.4.0.3 3D Plot Output

You can also create 3D scatterplot view of the overlapping hits by choosing `output="3dplot"`. Please not the additional arguments `plot.method` and `plot.feature`, which define what to plot on each axis.

```
# Perform the comparison and create 3D plot only for enriched
compared = compare.analysis( wilcox=data.wilcox, deseq=data.deseq, mageck=data.mageck, type="enriched",
```



5.4.0.4 compare.analysis()

Usage `compare.analysis(wilcox=NULL, deseq=NULL, mageck=NULL, type="enriched", cutoff.deseq = NULL, cutoff.wilcox = NULL, cutoff.mageck = NULL, cutoff.override=TRUE, cutoff.hits=5, output="list", sort.by=c("mageck","fdr","fdr"), plot.method=c("wilcox","mageck", "deseq"), plot.feature=c("pval","fdr","pval"), pch=16)`

wilcox
Data output from `stat.wilcox`.
Default NULL
Values Data output from `stat.wilcox`.

deseq
Data output from `stat.deseq`.
Default NULL
Values Data output from `stat.deseq`.

mageck
Data output from `stat.mageck`.
Default NULL
Values Data output from `stat.mageck`.

cutoff.deseq
P-Value threshold used to determine significance.
Default 0.001
Values numeric

cutoff.wilcox
P-Value threshold used to determine significance.
Default 0.001
Values numeric

cutoff.mageck
P-Value threshold used to determine significance.
Default 0.001
Values numeric

cutoff.override
Shall the p-value threshold be ignored? If this is TRUE, the top percentage gene of `cutoff.hits` is used instead.
Default FALSE
Values TRUE, FALSE

cutoff.hits
The percentage of top genes being used if `cutoff.override=TRUE`.
*Default** NULL
Values numeric

output
Three different types of output can be generated: A list with all genes including the information from `stat.wilcox`, `stat.DEseq` and `stat.mageck`, a sorted ranked output, a venn diagram compatible output and 3D scatterplot.
Default “list”
Values “list”, “rank”, “venn”, “3dplot”

sort.by
This indicates the sorting for `type="rank" and type="list"` and is a vector. By default, data is sorted by the FDR of MAGeCK. needs to be a vector.
Default c(“mageck”,“fdr”,“fdr”)
Values c(“mageck”,“fdr”,“fdr”), c(“mageck”,“fdr”,“rank”), c(“mageck”,“fdr”,“rank”), c(“wilcox”,“pval”,“pval”), c(“wilcox”,“pval”,“genes”), c(“deseq”,“pval”,“pval”), c(“deseq”,“pval”,“genes”)

plot.method
Used only if `type="3dplot"`. This indicates what is plotted at the x, y and z-axis and thus needs to be a vector of length 3.
Default c("wilcox","mageck", "deseq"), plots wilcox on X-axis, mageck on y-axis and deseq on z-axis
Values c("wilcox","mageck", "deseq") or any other combination

plot.feature
If `type="3dplot"`, this indicates the type of data plotted on each axis of the 3d plot. This can only be set according to the features available of the method used to be plotted as indicated in `plot.method`.
Default c("pval","fdr", "pval") which uses the p-value of wilcox, the fdr or MAGeCK and p-value of DESeq2.
Values c("pval","fdr", "pval"), or ANY combination according to `plot.method`

pch
The type of point used in the plot. See `?par()`.
Default 16
Values Any number describing the point, e.g. 16 (numeric)

5.5 Creating Final Hit Candidate Table

CaRpools also provides you with a final gene table, which includes p-values, fold changes and ranks by all methods in a single tabular output.
This output is **unbiased** and can thus be used for further analysis and data visualization.
It takes the output generated by each analysis method, `stat.wilcox`, `stat.DEseq` and `stat.mageck` and combines it into a single tabular representation.

```
# create final table with all genes + hit analysis results
final.tab = final.table(wilcox=data.wilcox, deseq=data.deseq, mageck=data.mageck, dataset=CONTROL1.g, na

# Write this to an EXCEL sheet
xlsx::write.xlsx(final.tab, "FINAL.xls", sheetName="Genelist",
  col.names=TRUE, row.names=FALSE, append=FALSE, showNA=TRUE)
```

An example of the generated output:

name	wilcox.FC	wilcox.pval	deseq2.pval	mageck.rank.pos	mageck.fdr.pos	mageck.rank.neg	mageck.fdr.neg
AAK1	0.4487116	0.9802939	1	354	0.987129	744	0.994232
AATK	0.4542620	0.9802939	1	426	0.987129	88	0.973911
ABI1	0.5490720	0.9671744	1	356	0.987129	764	0.994232
ABL1	0.2882981	0.9802939	1	679	0.996304	320	0.993066
ABL2	0.8073053	1.0000000	1	100	0.987129	302	0.993066
ACAD10	0.1910946	0.9802939	1	641	0.996304	659	0.994232
ACVR1	0.2899392	1.0000000	1	293	0.987129	171	0.993066
ACVR1B	0.0286973	0.9820183	1	805	0.996304	268	0.993066
ACVR1C	0.2157256	1.0000000	1	580	0.996304	541	0.994232
ACVR2A	0.3923660	1.0000000	1	314	0.987129	516	0.994232
ACVR2B	0.4256165	0.9802939	1	618	0.996304	655	0.994232
ACVRL1	0.2157440	0.9802939	1	651	0.996304	40	0.808760
ADAM9	0.2736503	0.9820183	1	481	0.987129	767	0.994232
ADCK1	0.7768593	0.9671744	1	105	0.987129	544	0.994232
ADCK2	0.3940072	0.9820183	1	509	0.987129	64	0.973911
ADCK4	0.1430250	0.9802939	1	593	0.996304	309	0.993066
ADCK5	0.0257506	1.0000000	1	748	0.996304	667	0.994232
ADK	0.4217891	1.0000000	1	317	0.987129	220	0.993066
ADRA1A	0.3570135	1.0000000	1	413	0.987129	437	0.994232
ADRA1B	0.4534786	1.0000000	1	470	0.987129	381	0.994232

5.5.0.5 final.table()

Usage

```
final.table(wilcox=NULL, deseq=NULL, mageck=NULL, dataset, namecolumn=1, norm.function=median,
type="genes", extractpattern = expression("^(.+?)_.+"))
```

wilcox
Data output from `stat.wilcox`.
Default NULL
Values Data output from `stat.wilcox`.

deseq
Data output from `stat.deseq`.
Default NULL

Values Data output from `stat.deseq`.

mageck
Data output from `stat.mageck`.
Default NULL
Values Data output from `stat.mageck`.

dataset
Data.frame as created by `load.file`
Default empty
Values data frame

namecolumn
In which column are the sgRNA identifiers?
Default 1
Values column number (numeric)

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier. e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

norm.function
The mathematical function to normalize data if `normalize=TRUE`. By default, the median is used.
Default median
Values Any mathematical function of R (function)

type
Output generated.
Default "genes"
Values "genes"

5.5.1 Create List of Overlapping Hit Candidates

CaRpools can also calculate which genes overlapped in all hit analysis methods using `generate.hits`.

```
overlap.enriched = generate.hits(wilcox=data.wilcox, deseq=data.deseq, mageck=data.mageck, type="enrich
```

CASP8
BAX
FADD
PASK

5.5.1.1 generate.hits()

Usage

```
generate.hits(wilcox=NULL, deseq=NULL, mageck=NULL,  type="enriched", cutoff.deseq =
0.001, cutoff.wilcox = 0.05, cutoff.mageck = 0.05, cutoff.override=FALSE, cutoff.hits=NULL,
plot.genes="overlapping")
```

wilcox
Data output from `stat.wilcox`.
Default NULL

Values Data output from `stat.wilcox`.

deseq
Data output from `stat.deseq`.
Default NULL
Values Data output from `stat.deseq`.

mageck
Data output from `stat.mageck`.
Default NULL
Values Data output from `stat.mageck`.

cutoff.deseq
P-Value threshold used to determine significance.
Default 0.001
Values numeric

cutoff.wilcox
P-Value threshold used to determine significance.
Default 0.001
Values numeric

cutoff.mageck
P-Value threshold used to determine significance.
Default 0.001
Values numeric

cutoff.override
Shall the p-value threshold be ignored? If this is TRUE, the top percentage gene of `cutoff.hits` is used instead.
Default FALSE
Values TRUE, FALSE

cutoff.hits
The percentage of top genes being used if `cutoff.override=TRUE`.
*Default** NULL
Values numeric

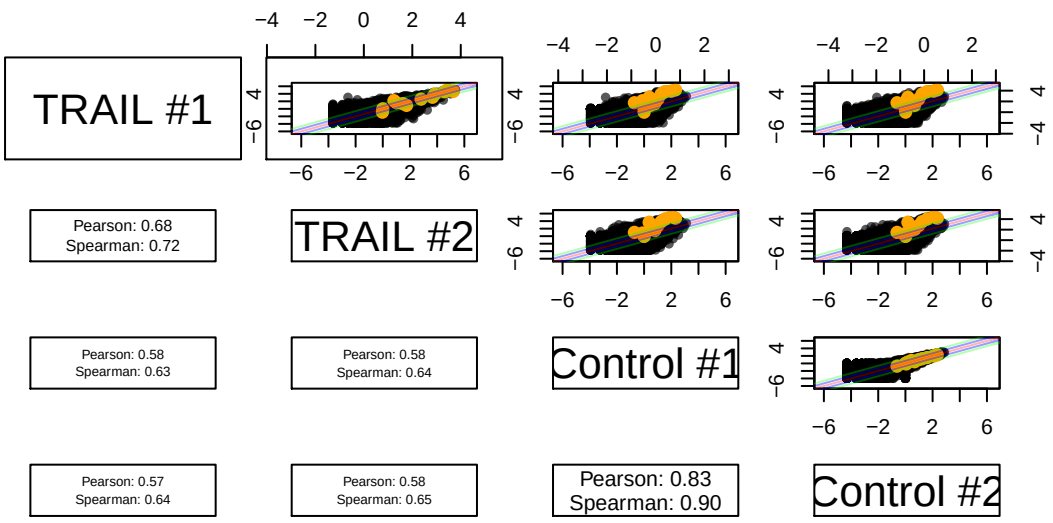
plot.genes
Defines what kind of data is returned, by default only overlapping genes or MAGeCK.
Default "overlapping"
Values "overlapping"

type
Defines whether all genes are plotted or only those being enriched or depleted.
Default "all"
Values "all", "enriched", "depleted"

5.5.2 Read-Count Scatter Plots for Candidate Genes

As described before, scatter plots can be generated for all data-sets. `carpools.hit.scatter` serves as a wrapper for `carpools.read.count.vs` and allows faster plotting for individual candidate genes or all overlapping candidate genes. It generated a pairs plot with the representation of all provided samples and highlights the candidate gene.

```
plothitsscatter.enriched = carpools.hit.scatter(wilcox=data.wilcox, deseq=data.deseq, mageck=data.mageck
```



5.5.2.1 carpools.hit.scatter()

Usage

```
carpools.hit.scatter = function(wilcox=NULL, deseq=NULL, mageck=NULL, dataset, dataset.names
= NULL, namecolumn=1, fullmatchcolumn=2, title="Read Count", xlab="Readcount Dataset1",
ylab="Readcount Dataset2", labelgenes=NULL, labelcolor="orange", extractpattern=expression("^(.+?)_.+")
plotline=TRUE, normalize=TRUE, norm.function=median, offsetplot=1.2, center=FALSE, aggregated=FALSE,
type="enriched", cutoff.deseq = 0.001, cutoff.wilcox = 0.05, cutoff.mageck = 0.05, cutoff.override=FALSE
cutoff.hits=NULL, plot.genes="overlapping", pch=16, col = rgb(0, 0, 0, alpha = 0.65))
```

wilcox
Data output from `stat.wilcox`.
Default NULL
Values Data output from `stat.wilcox`.

deseq
Data output from `stat.deseq`.
Default NULL
Values Data output from `stat.deseq`.

mageck
Data output from `stat.mageck`.
Default NULL
Values Data output from `stat.mageck`.

cutoff.deseq
P-Value threshold used to determine significance.
Default 0.001
Values numeric

cutoff.wilcox
P-Value threshold used to determine significance.
Default 0.001
Values numeric

cutoff.mageck
P-Value threshold used to determine significance.
Default 0.001
Values numeric

dataset A list of data frames of read-count data as created by load.file().
Default none
Values A list of data frames

namecolumn
In which column are the sgRNA identifiers?
Default 1
Values column number (numeric)

fullmatchcolumn
In which column are the read counts?
Default 2
Values column number (numeric)

dataset.names
A list of names that must be according to the list of data sets given in *dataset*.
Default NULL
Value NULL or list of data names (list)

norm.function
The mathematical function to normalize data. By default, the median is used.
Default median
Values Any mathematical function of R (function)

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier.
e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

cutoff.override
Shall the p-value threshold be ignored? If this is TRUE, the top percentage gene of `cutoff.hits` is used instead.
Default FALSE
Values TRUE, FALSE

cutoff.hits
The percentage of top genes being used if `cutoff.override=TRUE`.
*Default** NULL
Values numeric

plot.genes
Defines what kind of data is used. By default, overlapping genes are highlighted in red color.
Default "overlapping"
Values "overlapping"

type
Defines whether all genes are plotted or only those being enriched or depleted.
Default "all"
Values "all", "enriched", "depleted"

labelgenes
For which gene shall the sgRNA effects being plotted? This expects a gene identifier or a vector of gene identifiers.

Default NULL
Values A gene identifier or vector of gene identifiers (character)

xlab
Label of X-Axis, only if `pairs=FALSE`
Default "X-Axis"
Values "Label of X-Axis" (character)

ylab
Label of Y-Axis only if `pairs=FALSE`
Default "Y-Axis"
Values "Label of Y-Axis" (character)

pch
The type of point used in the plot. See `?par()`.
Default 16
Values Any number describing the point, e.g. 16 (numeric)

col
The color of the plotted data. Can be any R color or RGB object. See ?rgb() for further information.
Default rgb(0, 0, 0, alpha = 0.65)
Values Any R color name or RGB color object (character OR color object)

plotline
You can draw additional lines indicating a fold change of 0, 2, 4.
Default TRUE
*Values** TRUE, FALSE (boolean)

normalize
Whether you would like to normalize read-counts first. Recommended if not done already.
Default TRUE
Values TRUE, FALSE (boolean)

offsetplot
Offetplot is used to stretch the x- and y-axis for nicer graphs. This will extend plotting area by offsetplot.
Default 1.2 (Plotting area is stretched to 1.2 times)
Values any number (numeric)

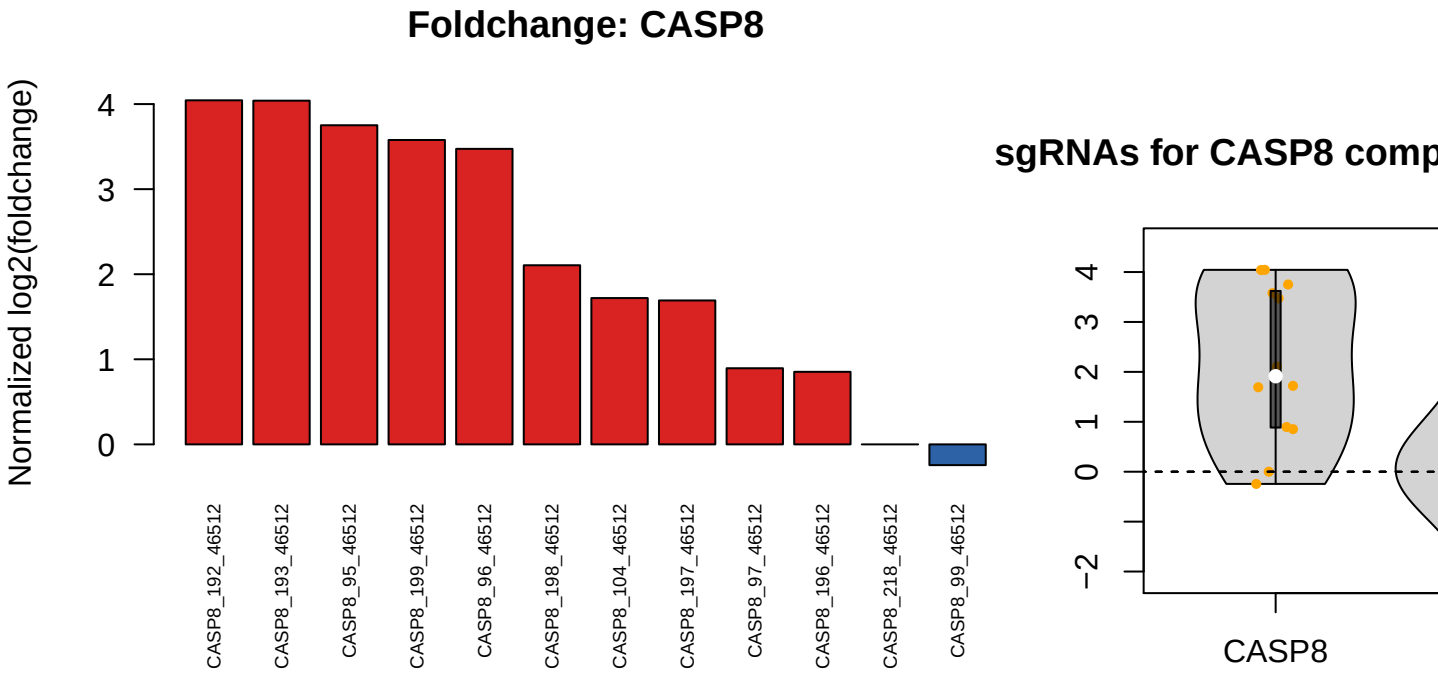
center
If you like you can center your data within the plot.
Default FALSE
Values TRUE, FALSE (boolean)

aggregated
If you want to highlight genes, set this to true if you provide already aggregated gene read count instead of sgRNA read counts.
Default FALSE
Values TRUE, FALSE (boolean)

5.5.3 sgRNA Effect Plots

Since there is more than just one single sgRNA targeting your gene of interest, you can user caRpools to plot different sgRNA phenotype effects, e.g. the fold change or z-ratio, as described before in `carpools.raw.genes`. A set of plots can be generated with `carpools.hit.sgrna`, which serves as a wrapper for `carpools.raw.genes`. By default, a fold-change plot as well as a violine plot are generated.

```
sgrnas.en = carpools.hit.sgrna(wilcox=data.wilcox, deseq=data.deseq,
    mageck=data.mageck, dataset=list(CONTROL1, CONTROL2, TREAT1, TREAT2),
    dataset.names = c(d.CONTROL1, d.CONTROL2, d.TREAT1, d.TREAT2), namecolumn=1,
    fullmatchcolumn=2, norm.function=median, extractpattern=expression("^(.+?)(_.+)"),
    put.names=TRUE, type="enriched", labelgenes="CASP8", plot.type=NULL,
    cutoff.deseq = 0.001, cutoff.wilcox=0.05, cutoff.mageck = 0.05,
    cutoff.override=FALSE, cutoff.hits=NULL, controls.target="CASP8", controls.nontarget="random")
```



5.5.3.1 carpools.hit.sgrna()

Usage

```
carpools.hit.sgrna(wilcox=NULL, deseq=NULL, mageck=NULL, dataset=NULL, dataset.names =
NULL, namecolumn=1, fullmatchcolumn=2, norm.function=median, extractpattern=expression("^(.+?)_.+"),
put.names=TRUE, type="enriched", labelgenes=NULL, cutoff.deseq = 0.05, cutoff.wilcox =
0.05, cutoff.mageck = 0.05, cutoff.override=FALSE, plot.genes="overlapping", cutoff.hits=NULL,
plot.type=NULL, controls.target=NULL, controls.nontarget=NULL)
```

wilcox
Data output from `stat.wilcox`.
Default NULL
Values Data output from `stat.wilcox`.

deseq
Data output from `stat.deseq`.
Default NULL
Values Data output from `stat.deseq`.

mageck
Data output from `stat.mageck`.
Default NULL
Values Data output from `stat.mageck`.

cutoff.deseq
P-Value threshold used to determine significance.

Default 0.001
Values numeric

cutoff.wilcox
P-Value threshold used to determine significance.
Default 0.001
Values numeric

cutoff.mageck
P-Value threshold used to determine significance.
Default 0.001
Values numeric

dataset A list of data frames of read-count data as created by load.file().
Default none
Values A list of data frames

namecolumn
In which column are the sgRNA identifiers?
Default 1
Values column number (numeric)

fullmatchcolumn
In which column are the read counts?
Default 2
Values column number (numeric)

dataset.names
A list of names that must be according to the list of data sets given in *dataset*.
Default NULL
Value NULL or list of data names (list)

norm.function
The mathematical function to normalize data. By default, the median is used.
Default median
Values Any mathematical function of R (function)

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier.
e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

cutoff.override
Shall the p-value threshold be ignored? If this is TRUE, the top percentage gene of `cutoff.hits` is used instead.
Default FALSE
Values TRUE, FALSE

cutoff.hits
The percentage of top genes being used if `cutoff.override=TRUE`.
*Default** NULL
Values numeric

plot.genes
Defines what kind of data is used. By default, overlapping genes are highlighted in red color.
Default "overlapping"
Values "overlapping"

type
Defines whether all genes are plotted or only those being enriched or depleted.
Default "all"
Values "all", "enriched", "depleted"

labelgenes
For which gene shall the sgRNA effects being plotted? This expects a gene identifier or a vector of gene identifiers.
Default NULL
Values A gene identifier or vector of gene identifiers (character)

controls.target
If `type="controls"`, this is the gene identifier of the positive control.
Default NULL
Value Gene Identifier (character)

controls.nontarget
If `type="controls"`, this is the gene identifier of the non-targeting control.
Default "random"
Value Gene Identifier (character)

put.names
Do you want the sgRNA identifiers to be plotted?
Default FALSE
Values TRUE, FALSE

plot.type
Which kind of plot is to be drawn? If NULL, fold-change and violine plots are generated.
Default NULL
Values NULL, "foldchange", "z-score", "z-ratio", "vioplot"

5.5.4 sgRNA Effect Table

Since there is more than just one single sgRNA targeting your gene of interest, you can user caRpools to plot different sgRNA phenotype effects, e.g. the fold change or z-ratio, as described before in `carpools.raw.genes`. In addition to that, caRpools also generated a tabular view which includes the log2 fold change as well as the target sequence, so that the user can directly pick the target sequence of the sgRNA he or shew wants.

This function, `carpools.sgrna.table` is best combined with `carpools.raw.genes` to give a fast overview of the sgRNA performance.

```
sgrnas.en.table = carpools.sgrna.table(wilcox=data.wilcox, deseq=data.deseq,
    mageck=data.mageck, dataset=list(CONTROL1, CONTROL2, TREAT1, TREAT2),
    dataset.names = c(d.CONTROL1, d.CONTROL2, d.TREAT1, d.TREAT2), namecolumn=1,
    fullmatchcolumn=2, norm.function=median, extractpattern=expression("^(.+?)(_.+)"),
    type="enriched", labelgenes="CASP8", cutoff.deseq = 0.001, cutoff.wilcox=0.05,
    cutoff.mageck = 0.05, cutoff.override=FALSE, cutoff.hits=NULL, sgrna.file = libFILE,
    write=FALSE)
```

`Carpools.sgrna.table` return a data.frame, that can be visualized best with knitr `kable` function:

	designs	genes	sequence	log2Foldchange
2	CASP8_192_46512	CASP8	gatgatgcccttgtctccat	4.0442283
3	CASP8_193_46512	CASP8	gagatgtcagctcatagatg	4.0404115
9	CASP8_95_46512	CASP8	ggagacaagggcatcatcta	3.7513973
7	CASP8_199_46512	CASP8	gacatcgctctctcaggctc	3.5778575

	designs	genes	sequence	log2Foldchange
10	CASP8_96_46512	CASP8	ggcatcatctatggcactga	3.4739444
6	CASP8_198_46512	CASP8	gcaaagtgactggatgtacc	2.1055251
1	CASP8_104_46512	CASP8	gcctgagagagcgatgtcct	1.7201422
5	CASP8_197_46512	CASP8	ggaaacacagttattcacag	1.6918053
11	CASP8_97_46512	CASP8	gagctgacatctcagttcac	0.8950126
4	CASP8_196_46512	CASP8	gcagaaagtcagcctcatcc	0.8529520
8	CASP8_218_46512	CASP8	gaaggcataaagcaagtttg	0.0000000
12	CASP8_99_46512	CASP8	ggagcaaccctatttagaaa	-0.2440090

5.5.4.1 carpools.sgrna.table()

Usage

```
carpools.sgrna.table (wilcox=NULL, deseq=NULL, mageck=NULL, dataset=NULL, dataset.names =
NULL, namecolumn=1, fullmatchcolumn=2, norm.function=median, extractpattern=expression("^(.+?)_.+"),
type="enriched", cutoff.deseq = 0.05, cutoff.wilcox = 0.05, cutoff.mageck = 0.05, cutoff.override=FALSE
plot.genes="overlapping", cutoff.hits=NULL, sgrna.file=NULL, labelgenes=NULL, write=FALSE,
datapath=getwd(), analysis.name="Screen")
```

wilcox
Data output from `stat.wilcox`.
Default NULL
Values Data output from `stat.wilcox`.

deseq
Data output from `stat.deseq`.
Default NULL
Values Data output from `stat.deseq`.

mageck
Data output from `stat.mageck`.
Default NULL
Values Data output from `stat.mageck`.

cutoff.deseq
P-Value threshold used to determine significance.
Default 0.001
Values numeric

cutoff.wilcox
P-Value threshold used to determine significance.
Default 0.001
Values numeric

cutoff.mageck
P-Value threshold used to determine significance.
Default 0.001
Values numeric

dataset A list of data frames of read-count data as created by load.file().
Default none
Values A list of data frames

namecolumn
In which column are the sgRNA identifiers?
Default 1
Values column number (numeric)

fullmatchcolumn
In which column are the read counts?
Default 2
Values column number (numeric)

dataset.names
A list of names that must be according to the list of data sets given in *dataset*.
Default NULL
Value NULL or list of data names (list)

norm.function
The mathematical function to normalize data. By default, the median is used.
Default median
Values Any mathematical function of R (function)

extractpattern
PERL regular expression that is used to retrieve the gene identifier from the overall sgRNA identifier.
e.g. in **AAK1_107_0** it will extract **AAK1**, since this is the gene identifier belonging to this sgRNA identifier. **Please see: Read-Count Data Files**
Default expression("^(.+?)(_.+)"), will work for most available libraries.
Values PERL regular expression with parenthesis indicating the gene identifier (expression)

cutoff.override
Shall the p-value threshold be ignored? If this is TRUE, the top percentage gene of `cutoff.hits` is used instead.
Default FALSE
Values TRUE, FALSE

cutoff.hits
The percentage of top genes being used if `cutoff.override=TRUE`.
*Default** NULL
Values numeric

plot.genes
Defines what kind of data is used. By default, overlapping genes are highlighted in red color.
Default "overlapping"
Values "overlapping"

type
Defines whether all genes are plotted or only those being enriched or depleted.
Default "all"
Values "all", "enriched", "depleted"

sgRNA.file
This is the library reference file loaded via `load.file` providing the sgRNA target sequence.
Default NULL
Values object from `load.file`

labelgenes
For which gene shall the sgRNA effects being generated? This expects a gene identifier or a vector of gene identifiers.
Default NULL
Values A gene identifier or vector of gene identifiers (character)

write
If you want to write directly to a file, this must be TRUE. Leave FALSE if you want the function to return a table.
Default FALSE
Values TRUE, FALSE

datapath
If `write=TRUE`, this is the directory the file is written.
Default getwd()
Values absolute path

analysis.name
The name of the file if `write=TRUE`
Default "Screen"
Values any file name (character)

6 Further Information

Please have a look at our website where you can find manuals, pre-defined reports, all scripts mentioned in this manual and a bug-tracker.
If you like to use or modify any of the code of caRpools, please fork or clone the github repository. If you find any bug or if you have a clever idea that you would like to have it included in caRpools, feel free to code or drop us a comment.