CRISPRcleanR: An R package for unsupervised identification and correction of gene independent cell responses to CRISPR-cas9 targeting

Francesco Iorio, fi1@sanger.ac.uk

December 9, 2017

1 Quick start

1.1 Installation

First, you need to install and load the devtools package. You can do this from CRAN. Invoke R and then type.

```
install.packages("devtools")
library(devtools)
```

Secondly, install the CRISPRcleanR with the following command:

```
install_github("francescojm/CRISPRcleanR")
```

1.2 Raw sgRNA count median-ratio normalisation and computation of sgRNAs' log fold-changes

Load the package.

```
    library(CRISPRcleanR)
```

```
## Loading required package:  stringr
## Loading required package:  DNAcopy
## Loading required package:  pROC
## Type 'citation("pROC")' for a citation.
##
## Attaching package:  'pROC'
## The following objects are masked from 'package:stats':
##
##    cov, smooth, var
## Loading required package:  pracma
```

Step 1: Load your sgRNA library annotation. In this example we will use a built in data frame containing the annotation of the SANGER v1.0 library [1]:

```
data(KY_Library_v1.0)
```

To use your own library annotation you will have to put it in a data frame with the same format of the `KY_Library_v1.0` data frame (detailed in the corresponding entry of the reference manual of the CRISPRcleanR package).

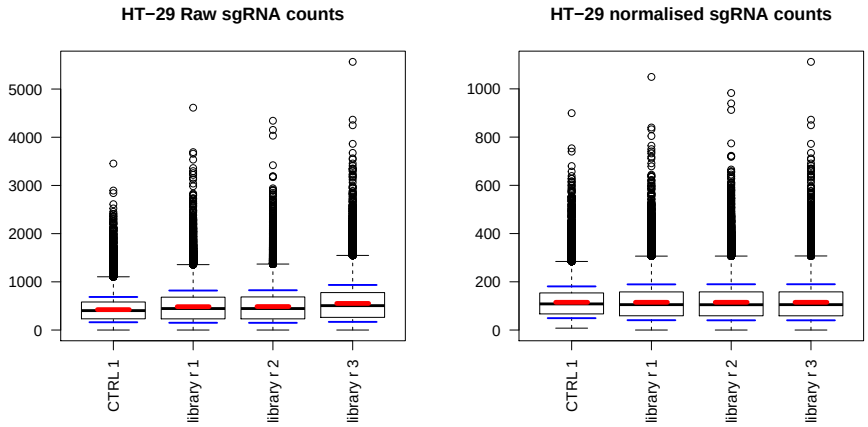
Step 2: Store the path of the tsv file containing your sgRNAs' raw counts in a temporare variable. In this example we will use counts generated upon a CRISPR-Cas9 pooled drop-out screen (described in [2]) built in this package.

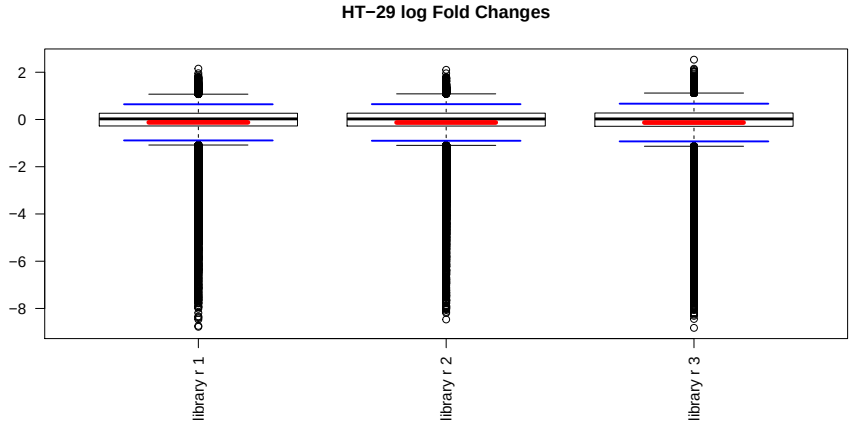
```
fn<-paste(system.file('extdata',package = 'CRISPRcleanR'),
          '/HT-29_counts.tsv',sep='')
```

The tsv with the sgRNAs' raw counts must be formatted as specified in the reference manual entry for function `ccr.NormfoldChanges`.

Step 3: Performing median-ratio normalisation of raw counts and computing sgRNAs' log fold-changes. In this example we will exclude sgRNAs with less than 30 reads in the plasmid sample.

```
normANDfcs<-ccr.NormfoldChanges(fn,
                                min_reads=30,
                                EXPname='HT-29',
                                libraryAnnotation=KY_Library_v1.0)
```





This function returns a list of two data frames, respectively with normalised counts and log fold-changes, and it saves the as Robject in the directory whose path is specified with the parameter `outdir` (set to `'./'` by default).

```
head(normANDfcs$norm_counts)

##                                                sgRNA gene ERS717283.plasmid
## 1 A1BG_CCDS12976.1_ex3_19:58862927-58862950:-_5-1 A1BG         292.14621
## 2 A1BG_CCDS12976.1_ex4_19:58863655-58863678:+_5-2 A1BG         151.02032
## 3 A1BG_CCDS12976.1_ex4_19:58863697-58863720:-_5-3 A1BG         209.08503
## 4 A1BG_CCDS12976.1_ex4_19:58863866-58863889:+_5-4 A1BG         110.40106
## 5 A1BG_CCDS12976.1_ex5_19:58864367-58864390:-_5-5 A1BG          95.81979
## 6  A1CF_CCDS7241.1_ex6_10:52588014-52588037:-_5-1 A1CF          60.92889
##   HT29_c904R1 HT29_c904R2 HT29_c904R3
## 1   308.05192   354.89835   305.56806
## 2   145.38048   113.16912   166.47364
## 3   280.97793   203.70441   223.03071
## 4    80.31191    64.05372    78.74023
## 5    78.71932   123.80701   103.32157
## 6    47.77762    76.50232    59.75464

head(normANDfcs$logFCs)

##                                                sgRNA gene HT29_c904R1
## 1 A1BG_CCDS12976.1_ex3_19:58862927-58862950:-_5-1 A1BG  0.07635566
## 2 A1BG_CCDS12976.1_ex4_19:58863655-58863678:+_5-2 A1BG -0.05472442
## 3 A1BG_CCDS12976.1_ex4_19:58863697-58863720:-_5-3 A1BG  0.42548611
## 4 A1BG_CCDS12976.1_ex4_19:58863866-58863889:+_5-4 A1BG -0.45663336
## 5 A1BG_CCDS12976.1_ex5_19:58864367-58864390:-_5-5 A1BG -0.28197989
## 6  A1CF_CCDS7241.1_ex6_10:52588014-52588037:-_5-1 A1CF -0.34756262
##   HT29_c904R2 HT29_c904R3
## 1  0.28027938  0.06469488
## 2 -0.41467098  0.14010902
## 3 -0.03752168  0.09293733
```

```
## 4 -0.78070106 -0.48496821
## 5  0.36800353  0.10820208
## 6  0.32598467 -0.02784489
```

IMPORTANT: if there are control replicates in your sgRNAs count file their number must be specified by in the parameter `ncontrols` (equal to 1 by default) of the `ccr.NormfoldChanges` function.

1.3 Genome sorting of sgRNAs' log fold-changes and their correction for gene independent responses to CRISPR-Cas9 targeting

Step 1: Map genome-wide sgRNAs' log fold changes (averaged across replicates) on the genome, sorted according to their positions of the targeted region on the chromosomes.

```
gwSortedFCs<-
    ccr.logFCs2chromPos(normANDfcs$logFCs,KY_Library_v1.0)
```

```
head(gwSortedFCs)

##                                          CHR startp   endp  genes
## SAMD11_CCDS2.2_ex3_1:871254-871277:+_5-1   1 871254 871277 SAMD11
## SAMD11_CCDS2.2_ex4_1:874451-874474:-_5-2   1 874451 874474 SAMD11
## SAMD11_CCDS2.2_ex4_1:874487-874510:+_5-3   1 874487 874510 SAMD11
## SAMD11_CCDS2.2_ex5_1:874693-874716:+_5-4   1 874693 874716 SAMD11
## SAMD11_CCDS2.2_ex6_1:876601-876624:-_5-5   1 876601 876624 SAMD11
## NOC2L_CCDS3.1_ex8_1:887388-887411:+_5-1    1 887388 887411  NOC2L
##                                                 avgFC       BP
## SAMD11_CCDS2.2_ex3_1:871254-871277:+_5-1 -0.12965287 871265.5
## SAMD11_CCDS2.2_ex4_1:874451-874474:-_5-2  0.09329615 874462.5
## SAMD11_CCDS2.2_ex4_1:874487-874510:+_5-3  0.25286616 874498.5
## SAMD11_CCDS2.2_ex5_1:874693-874716:+_5-4 -0.05128489 874704.5
## SAMD11_CCDS2.2_ex6_1:876601-876624:-_5-5 -0.02110076 876612.5
## NOC2L_CCDS3.1_ex8_1:887388-887411:+_5-1  -1.27571756 887399.5
```

Step 2: Identify and correct biased sgRNAs' log fold-changes putatively due to gene independent responses to CRISPR-Cas9 targeting (this function calls iteratively the `ccr.cleanChrm` function, which performs the correction in each chromosome individually). In this example we are using a completely unsuerpvised approach and correcting chromosomal segments of equal sgRNA log fold-changes if they include sgRNAs targeting at least 3 different genes, and without making any assumption on gene essentiality nor knowing *a priori* the copy number status of the included genes [2].

```
correctedFCs<-ccr.GWclean(gwSortedFCs,display=TRUE,label='HT-29')
```

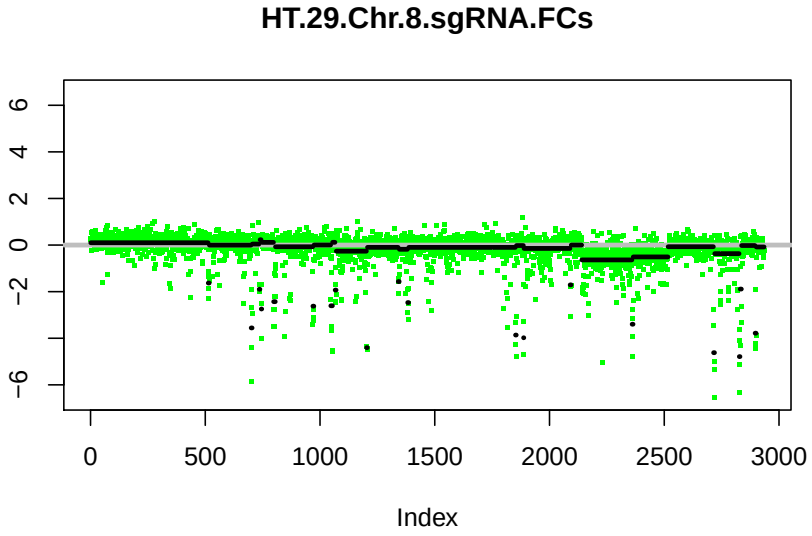
The corrected sgRNAs fold-changes are returned in a list (as a data frame), together with another data frame with annotation of the identified segments and a vector of strings containing all the sgRNAs identifier genome-sorted.

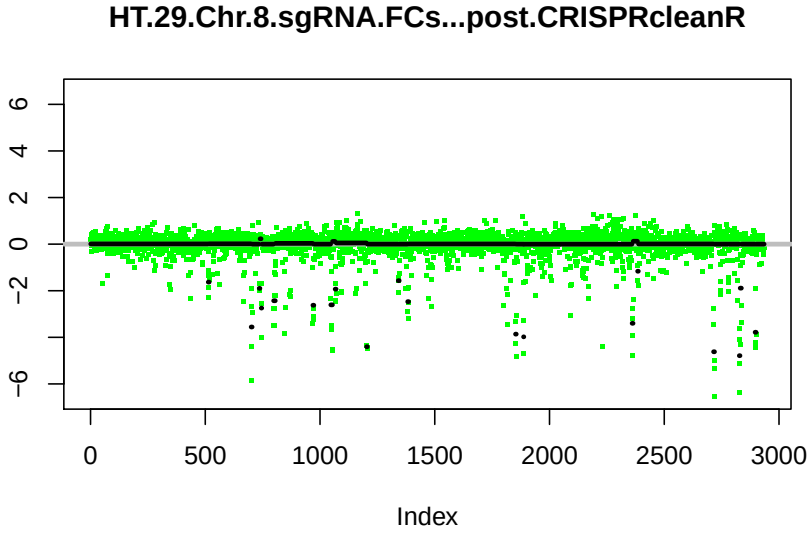
```
    head(correctedFCs$corrected_logFCs)
```

```
##                                          CHR startp   endp  genes
## SAMD11_CCDS2.2_ex3_1:871254-871277:+_5-1   1 871254 871277 SAMD11
## SAMD11_CCDS2.2_ex4_1:874451-874474:-_5-2   1 874451 874474 SAMD11
## SAMD11_CCDS2.2_ex4_1:874487-874510:+_5-3   1 874487 874510 SAMD11
## SAMD11_CCDS2.2_ex5_1:874693-874716:+_5-4   1 874693 874716 SAMD11
## SAMD11_CCDS2.2_ex6_1:876601-876624:-_5-5   1 876601 876624 SAMD11
## NOC2L_CCDS3.1_ex8_1:887388-887411:+_5-1    1 887388 887411  NOC2L
##                                                avgFC       BP correction
## SAMD11_CCDS2.2_ex3_1:871254-871277:+_5-1 -0.12965287 871265.5          0
## SAMD11_CCDS2.2_ex4_1:874451-874474:-_5-2  0.09329615 874462.5          0
## SAMD11_CCDS2.2_ex4_1:874487-874510:+_5-3  0.25286616 874498.5          0
## SAMD11_CCDS2.2_ex5_1:874693-874716:+_5-4 -0.05128489 874704.5          0
## SAMD11_CCDS2.2_ex6_1:876601-876624:-_5-5 -0.02110076 876612.5          0
## NOC2L_CCDS3.1_ex8_1:887388-887411:+_5-1  -1.27571756 887399.5          0
##                                          correctedFC
## SAMD11_CCDS2.2_ex3_1:871254-871277:+_5-1 -0.12965287
## SAMD11_CCDS2.2_ex4_1:874451-874474:-_5-2  0.09329615
## SAMD11_CCDS2.2_ex4_1:874487-874510:+_5-3  0.25286616
## SAMD11_CCDS2.2_ex5_1:874693-874716:+_5-4 -0.05128489
## SAMD11_CCDS2.2_ex6_1:876601-876624:-_5-5 -0.02110076
## NOC2L_CCDS3.1_ex8_1:887388-887411:+_5-1  -1.27571756
```

Details on how the data frame with the corrected sgRNAs fold-changes should be interpreted can be found in the entry of the `ccr.GWclean` function in the package reference manual.

This function also produces one plot per chromosome, with segments of sgRNAs' equal log fold-changes before and after the correction. An example of these plot is reported below (chromosome 8, in HT-29 with the region containing *MYC* highly biased toward consistent negative fold-changes)





1.4 Correcting sgRNAs' treatment counts for mean-variance modeling

In order to apply the inverse transformation described in [2], thus to derive corrected normalised sgRNAs' treatment counts from corrected log fold-changes, it is sufficient to run the function `ccr.correctCounts` as follows:

```
correctedCounts<-ccr.correctCounts('HT-29',
                                   normANDfcs$norm_counts,
                                   correctedFCs,
                                   KY_Library_v1.0,
                                   minTargetedGenes=3,
                                   OutDir='./')
```

With the plasmid counts, are suitable for mean-variance modeling approach (such that implemented in MAGeCK[3]).

```
head(correctedCounts)
```

```
##                                                sgRNA gene ERS717283.plasmid
## 1 A1BG_CCDS12976.1_ex3_19:58862927-58862950:-_5-1 A1BG         292.14621
## 2 A1BG_CCDS12976.1_ex4_19:58863655-58863678:+_5-2 A1BG         151.02032
## 3 A1BG_CCDS12976.1_ex4_19:58863697-58863720:-_5-3 A1BG         209.08503
## 4 A1BG_CCDS12976.1_ex4_19:58863866-58863889:+_5-4 A1BG         110.40106
## 5 A1BG_CCDS12976.1_ex5_19:58864367-58864390:-_5-5 A1BG          95.81979
## 6  A1CF_CCDS7241.1_ex6_10:52588014-52588037:-_5-1 A1CF          60.92889
##   HT29_c904R1 HT29_c904R2 HT29_c904R3
## 1   309.77863   356.73522   307.28892
## 2   144.74469   112.89328   165.60212
## 3   280.34458   203.51866   222.73301
## 4    80.64680    64.52159    79.08797
## 5    78.22697   122.47062   102.36866
## 6    45.21299    71.83850    56.31473
```

This function also saves the correctedCounts as Rdata object at the location specified by the parameter `OutDir`. To run MAGeCK, using these corrected sgRNAs' counts you will need to save them as a tsv file first:

```
write.table(correctedCounts,
            quote=FALSE,
            row.names = FALSE,
            sep='\t',
            file='./HT-29_mgk_input_corrected.tsv')
```

then use this file as input for MAGeCK.

IMPORTANT: the corrected sgRNAs' count are already median-normalised therefore, when executing MAGeCK, the parameter `--norm-method` should be set to `none`.

2 Visualisation and assessment of Results

2.1 Classification performances of reference sets of genes (or sgRNAs) based on depletion log fold-changes

To perform a basic quality control assessment of your data it is possible to test the genome-wide profile of sgRNAs' depletion logFCs (or gene depletion logFCs averaged across targeting sgRNAs) as a classifier of reference sets of core-fitness essential (CFE) and non-essential genes. What you need for this is a named vector of sgRNAs (or gene) log fold changes and two reference gene sets. In this example we make use of a precomputed essentiality profile from the builtin data object `EPLC.272HcorrectedFCs`. This is a list containing corrected sgRNAs log fold-changes and segment annotations for an example cell line (EPLC-272H), obtained using the `ccr.GWclean` function, as detailed in its reference manual entry. However the data frame containing the corrected log fold-changes, included in this list, reports also the original sgRNAs logFC (column `avgFC` which will be used in this example).

```
data(EPLC.272HcorrectedFCs)
```

```
head(EPLC.272HcorrectedFCs$corrected_logFCs)
```

```
##                                          CHR startp   endp  genes
## SAMD11_CCDS2.2_ex3_1:871254-871277:+_5-1   1 871254 871277 SAMD11
## SAMD11_CCDS2.2_ex4_1:874451-874474:-_5-2   1 874451 874474 SAMD11
## SAMD11_CCDS2.2_ex4_1:874487-874510:+_5-3   1 874487 874510 SAMD11
## SAMD11_CCDS2.2_ex5_1:874693-874716:+_5-4   1 874693 874716 SAMD11
## SAMD11_CCDS2.2_ex6_1:876601-876624:-_5-5   1 876601 876624 SAMD11
## NOC2L_CCDS3.1_ex8_1:887388-887411:+_5-1    1 887388 887411  NOC2L
##                                                 avgFC       BP correction
## SAMD11_CCDS2.2_ex3_1:871254-871277:+_5-1 -0.20295496 871265.5          0
## SAMD11_CCDS2.2_ex4_1:874451-874474:-_5-2 -0.08917153 874462.5          0
## SAMD11_CCDS2.2_ex4_1:874487-874510:+_5-3 -0.04417670 874498.5          0
## SAMD11_CCDS2.2_ex5_1:874693-874716:+_5-4  0.30441537 874704.5          0
## SAMD11_CCDS2.2_ex6_1:876601-876624:-_5-5 -0.11240079 876612.5          0
## NOC2L_CCDS3.1_ex8_1:887388-887411:+_5-1  -1.61370746 887399.5          0
##                                          correctedFC
## SAMD11_CCDS2.2_ex3_1:871254-871277:+_5-1 -0.20295496
## SAMD11_CCDS2.2_ex4_1:874451-874474:-_5-2 -0.08917153
## SAMD11_CCDS2.2_ex4_1:874487-874510:+_5-3 -0.04417670
## SAMD11_CCDS2.2_ex5_1:874693-874716:+_5-4  0.30441537
## SAMD11_CCDS2.2_ex6_1:876601-876624:-_5-5 -0.11240079
## NOC2L_CCDS3.1_ex8_1:887388-887411:+_5-1  -1.61370746
```

As reference gene sets we will lists of CFE and non-essential genes assembled from multiple RNAi studies used as classification template by the BAGEL algorithm to call gene depletion significance [4], included in the builtin data objects `BAGEL_essential` and `BAGEL_nonEssential`.

```
data(BAGEL_essential)
data(BAGEL_nonEssential)

head(BAGEL_essential)

## [1] "ACTL6A" "ACTR6"  "ALYREF" "ANAPC4" "ANAPC5" "AP2S1"

head(BAGEL_nonEssential)

## [1] "ABCG8"  "ACCSL"  "ACTL7A" "ACTL7B" "ACTL9"  "ACTRT1"
```

Finally, we will need the sgRNAs library annotation. In this case we will use the builtin object `KY_KY_Library_v1.0` (introduced in the previous section) [1]. To use a different library annotation you will have to put it in a data frame with the same format of the `KY_Library_v1.0` data frame (detailed in the corresponding entry of the reference manual of the CRISPRcleanR package).

```
data(KY_Library_v1.0)
```

We will start with an evualuation at the sgRNA level. As mentioned, the log fold-changes needs to be stored a named vector:

```
FCs<-EPLC.272HcorrectedFCs$corrected_logFCs$avgFC
names(FCs)<-rownames(EPLC.272HcorrectedFCs$corrected_logFCs)
```

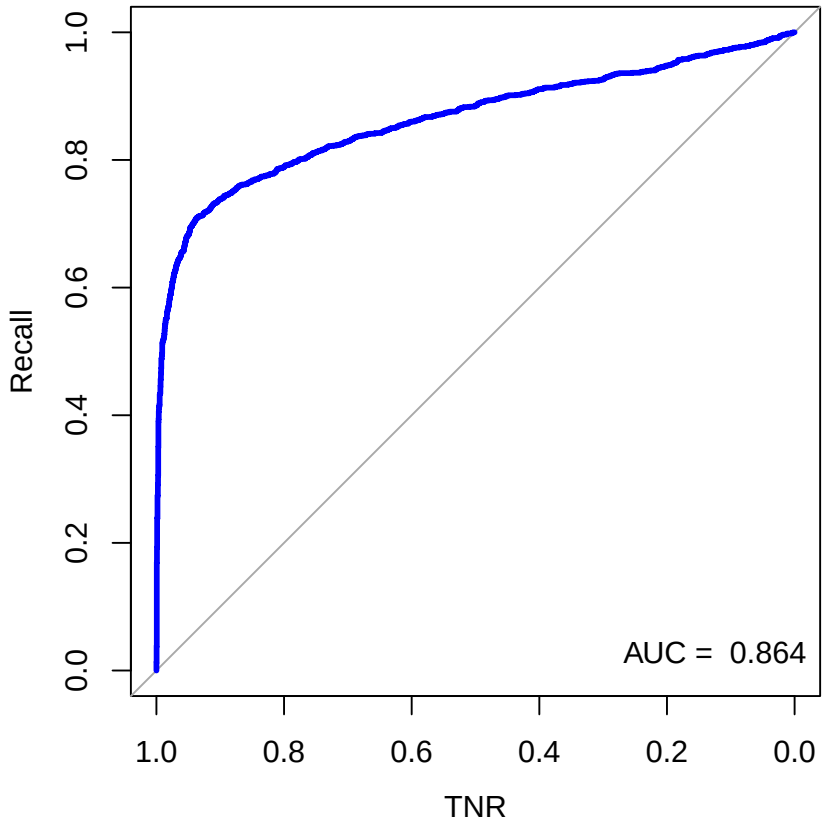
To convert the reference CFE and non-essential gene sets into sets of sgRNAs, the function `ccr.genes2sgRNAs` can be used, as follows:

```
BAGEL_essential_sgRNAs<-
    ccr.genes2sgRNAs(KY_Library_v1.0,BAGEL_essential)
BAGEL_nonEssential_sgRNAs<-
    ccr.genes2sgRNAs(KY_Library_v1.0,BAGEL_nonEssential)
```

Following these calls, possible warning messages could appear informing you that some of the reference genes are not targeted by any sgRNA in the considered library. This has no impact on the following steps and results.

Finally, to visualise the precision-recall curve quantifying the performances in classifying the considered reference sets it is sufficient to call:

```
ccr.PrecisionRecallCurve(FCs,BAGEL_essential_sgRNAs,
                            BAGEL_nonEssential_sgRNAs)
```




```
## $AUC
## Area under the curve: 0.8639
##
## $Recall
## NULL
##
## $sigthreshold
## NULL
```

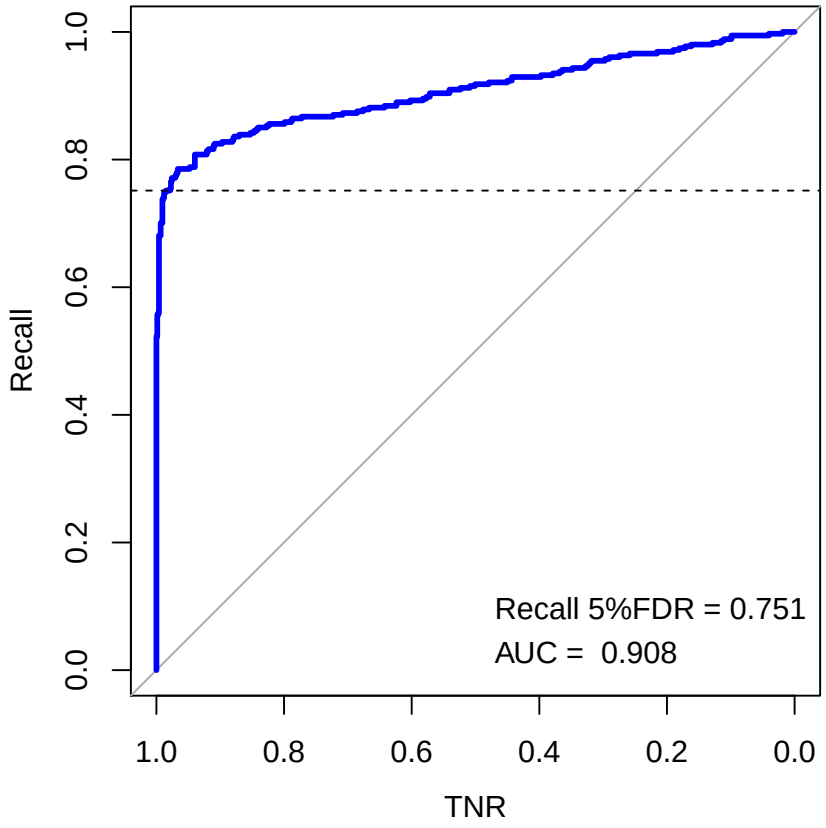
To reperform the analysis at the gene level, first we should convert the profile of sgRNA log fold change into gene level summaries. The function `ccr.geneMeanFCs` performs this conversion by considering for each gene the average logFC across targeting guides.

```
geneFCs<-ccr.geneMeanFCs(FCs,KY_Library_v1.0)
head(geneFCs)
```

```
##       A1BG       A1CF        A2M      A2ML1    A3GALT2     A4GALT
## -0.2474235 -0.1550534  0.2190111  0.3736683 -0.5151889 -0.1698269
```

The following call reperform the analysis at the gene level and it also computes and shows Recall values at a fixed False Discovery Rate (in this case equal to 5%).

```
ccr.PrecisionRecallCurve(geneFCs,
                         BAGEL_essential,
                         BAGEL_nonEssential,
                         FDRth = 0.05)
```




```
## $AUC
## Area under the curve: 0.9078
##
## $Recall
```

```
## [1] 0.7514124
##
## $sigthreshold
## [1] -0.7683409
```

As can be seen above, when setting the parameter `FRDth` to a value different from `NULL` (its default value), this function also return the log fold change threshold at which a classification FDR equal to the inputted value is achieved.

2.2 Depletion profile visualisation with genes signatures superimposed and recall computation

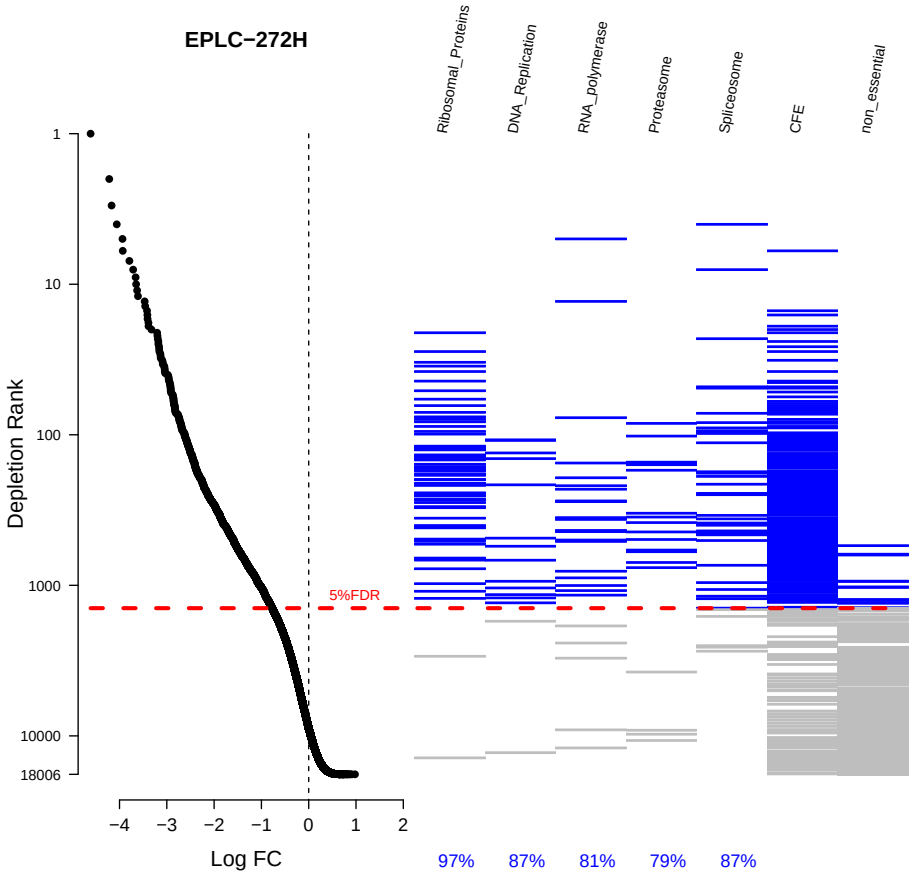
For another quick assessment of your data it is possible to visually inspect enrichments of predefined sets of core-fitness essential genes at the top of the genome wide essentiality profiles (ranked based on depletion logFC in increasing order), and to compute their classification recall at a fixed FDR (determined as deatiled in the previous subsection).

To this aim, in this example we will load additional sets of CFE genes assembled from MsigDB [5] as detailed in [2], and we will assemble them into a named list, as follows:

```
data(EssGenes.ribosomalProteins)
data(EssGenes.DNA_REPLICATION_cons)
data(EssGenes.KEGG_rna_polymerase)
data(EssGenes.PROTEASOME_cons)
data(EssGenes.SPLICEOSOME_cons)

## Assembling a named list with all the considered gene sets
SIGNATURES<-list(Ribosomal_Proteins=EssGenes.ribosomalProteins,
                 DNA_Replication = EssGenes.DNA_REPLICATION_cons,
                 RNA_polymerase = EssGenes.KEGG_rna_polymerase,
                 Proteasome = EssGenes.PROTEASOME_cons,
                 Spliceosome = EssGenes.SPLICEOSOME_cons,
                 CFE = BAGEL_essential,
                 non_essential = BAGEL_nonEssential)
```

Finally we will create a visualisation of the gene essentiality profile with superimposed these signatures, as follows:

```
Recall_scores<-ccr.VisDepAndSig(FCsprofile = geneFCs,
                                SIGNATURES = SIGNATURES,
                                TITLE = 'EPLC-272H',
                                pIs = 6,
                                nIs = 7)
```



IMPORTANT: When calling `ccr.VisDepAndSig` it is important to correctly specify the index position of the reference gene sets that are used as classification template to derive the FDR threshold, within the list of signatures. In this case the template sets are `BAGEL_essential` and `BAGEL_nonEssential`, which in the `SIGNATURE` list are in position 6 and 7, respectively (this must be specified in the `pIs` and `nIs` parameters of the `ccr.VisDepAndSig` function).

This function also returns recall values at 5% FDR for all the inputted signatures.

```
Recall_scores

## Ribosomal_Proteins    DNA_Replication     RNA_polymerase
##          0.96721311         0.86666667         0.80769231
##         Proteasome        Spliceosome                CFE
##          0.78947368         0.86842105         0.75141243
##      non_essential
```

```
##         0.01874163
```

2.3 CRISPRcleanR correction assessment: Statistical tests

To evaluate the effect of the CRISPRcleanR correction on your data it is possible to inspect the logFCs changes of sgRNAs targeting different sets of genes for statistically significant differences with respect to background pre/post CRISPRcleanR correction.

To this aim, in this example we will use the builtin data object `HT.29correctedFCs` containing corrected sgRNAs logFCs and segment annotations for an example cell line (HT-29), obtained using the `ccr.GWclean` function, as detailed in its reference manual entry.

```
data(HT.29correctedFCs)
```

The function `ccr.perf_statTests` performs this analysis, saving pdf figures in a user defined location ('./' by default).

Particularly, this functions assess the statistical difference pre/post CRISPRcleanR correction of log fold changes for sgRNAs targeting respectively:

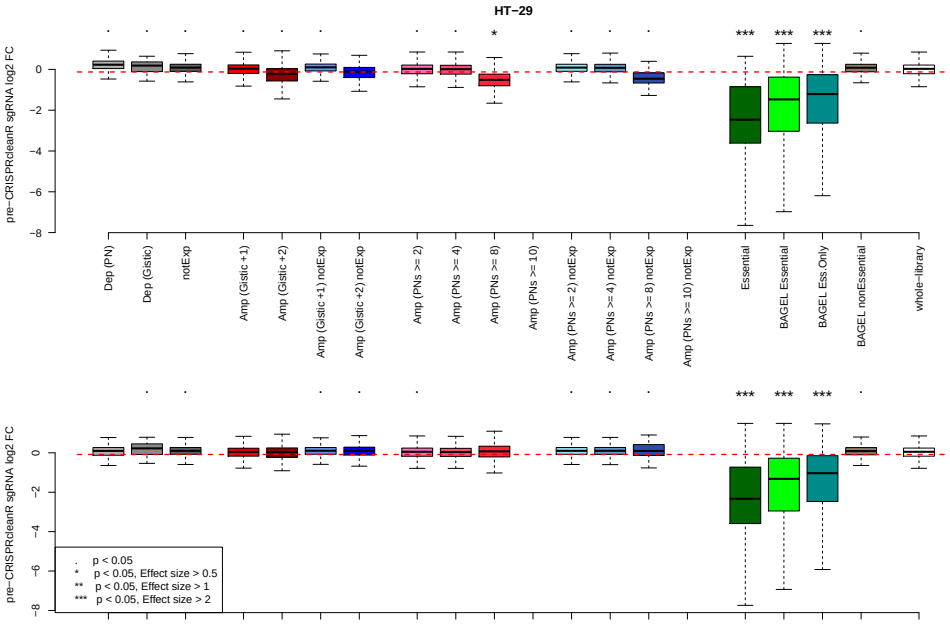
- copy number (CN) deleted genes according to the GDSC1000 repository
- CN deleted genes (gistic score = -2) according to the CCLE repository
- non expressed genes (FPKM lower than 0.05)
- genes with gistic score = 1
- genes with gistic score = 2
- non espressed genes (FPKM lower than 0.05) with gistic score = 1
- non espressed genes (FPKM lower than 0.05) with gistic score = 2
- genes with minimal CN = 2, according to the GDSC1000
- genes with minimal CN = 4, according to the GDSC1000
- genes with minimal CN = 8, according to the GDSC1000
- genes with minimal CN = 10, according to the GDSC1000
- non expressed genes (FPKM lower than 0.05) with minimal CN = 2, according to the GDSC1000
- non expressed genes (FPKM lower than 0.05) with minimal CN = 4, according to the GDSC1000

- non expressed genes (FPKM lower than 0.05) with minimal CN = 8, according to the GDSC1000
- non expressed genes (FPKM lower than 0.05) with minimal CN = 10, according to the GDSC1000

It should be called as follows:

```
RES<-ccr.perf_statTests('HT-29',libraryAnnotation = KY_Library_v1.0,
                    correctedFCs = HT.29correctedFCs$corrected_logFCs,
                    GDSC.geneLevCNA = NULL,
                    CCLE.gisticCNA = NULL,
                    RNAseq.fpkms = NULL)
```

It will save the following figure in the indicated path.



Leaving the parameters `GDSC.geneLevCNA`, `CCLE.gisticCNA`, and `RNAseq.fpkms` to their default NULL value will force this function to use the respective builtin data object containing data only for 15 cell lines used in [2] and in this package documentation to assess the performances of CRISPRcleanR.

IMPORTANT: To analyse data from screening a different cell line ad-hoc `GDSC.geneLevCNA`, `CCLE.gisticCNA`, and `RNAseq.fpkms` data object should be assembled (with the same format of the respective builtin data objects, detailaed in their user reference manual entries, which contain also additional infos on how

to derive data for 1,000 human cance cell lines).

Comprehensive statistical scores (detailed in the user reference manual) resulting from the executio of this function are also returned in output.

Another example, analysing in the same way the essentiality profile of the EPLC-272H cell line is reported below.

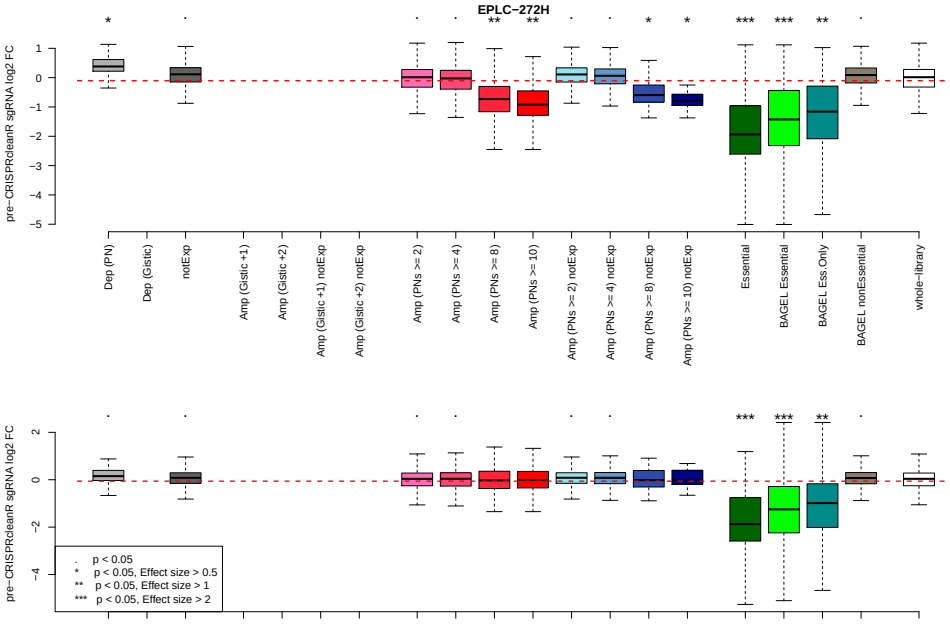
```
RES<-ccr.perf_statTests('EPLC-272H',libraryAnnotation = KY_Library_v1.0,
                   correctedFCs = EPLC.272HcorrectedFCs$corrected_logFCs)
```

```
## [1] "No gistic CNA scores available for this cell line"
## [1] "Testing sgRNAs targeting: Dep (PN) genes"
## [1] "Testing sgRNAs targeting: Dep (Gistic) genes"
## [1] "Testing sgRNAs targeting: notExp genes"
## [1] "Testing sgRNAs targeting: Amp (Gistic +1) genes"
## [1] "Testing sgRNAs targeting: Amp (Gistic +2) genes"
## [1] "Testing sgRNAs targeting: Amp (Gistic +1) notExp genes"
## [1] "Testing sgRNAs targeting: Amp (Gistic +2) notExp genes"
## [1] "Testing sgRNAs targeting: Amp (PNs >= 2) genes"
## [1] "Testing sgRNAs targeting: Amp (PNs >= 4) genes"
## [1] "Testing sgRNAs targeting: Amp (PNs >= 8) genes"
## [1] "Testing sgRNAs targeting: Amp (PNs >= 10) genes"
## [1] "Testing sgRNAs targeting: Amp (PNs >= 2) notExp genes"
## [1] "Testing sgRNAs targeting: Amp (PNs >= 4) notExp genes"
## [1] "Testing sgRNAs targeting: Amp (PNs >= 8) notExp genes"
## [1] "Testing sgRNAs targeting: Amp (PNs >= 10) notExp genes"
## [1] "Testing sgRNAs targeting: Essential genes"
## [1] "Testing sgRNAs targeting: BAGEL Essential genes"
## [1] "Testing sgRNAs targeting: BAGEL Ess.Only genes"
## [1] "Testing sgRNAs targeting: BAGEL nonEssential genes"
```

```
RES$PVALS
```

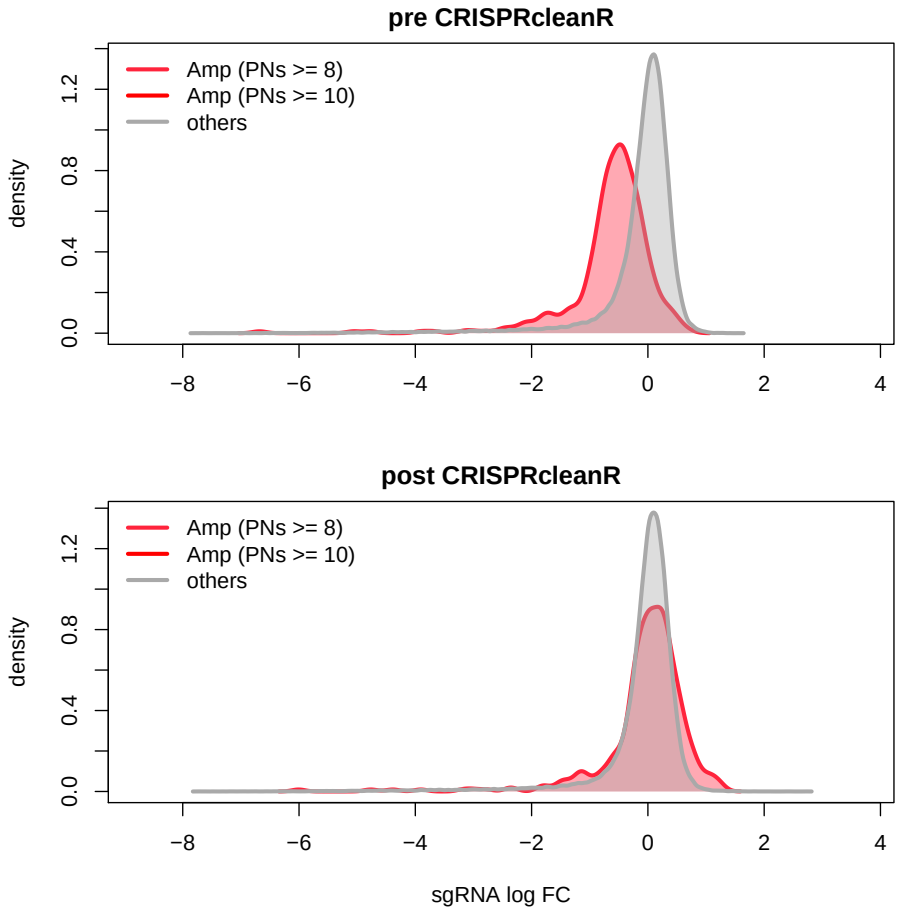
```
##                       Dep (PN) Dep (Gistic)       notExp Amp (Gistic +1)
## pre-CRISPRcleanR  5.145194e-27           NA  0.00000e+00              NA
## post-CRISPRcleanR 3.219730e-06           NA 5.69841e-173              NA
##                   Amp (Gistic +2) Amp (Gistic +1) notExp
## pre-CRISPRcleanR               NA                     NA
## post-CRISPRcleanR              NA                     NA
##                   Amp (Gistic +2) notExp Amp (PNs >= 2) Amp (PNs >= 4)
## pre-CRISPRcleanR                      NA   1.580669e-18   1.780173e-57
## post-CRISPRcleanR                     NA   4.951065e-05   1.593962e-02
##                   Amp (PNs >= 8) Amp (PNs >= 10) Amp (PNs >= 2) notExp
## pre-CRISPRcleanR   1.991668e-137    1.476429e-42         2.268570e-311
## post-CRISPRcleanR   2.036767e-01    9.122396e-02         7.683083e-168
```

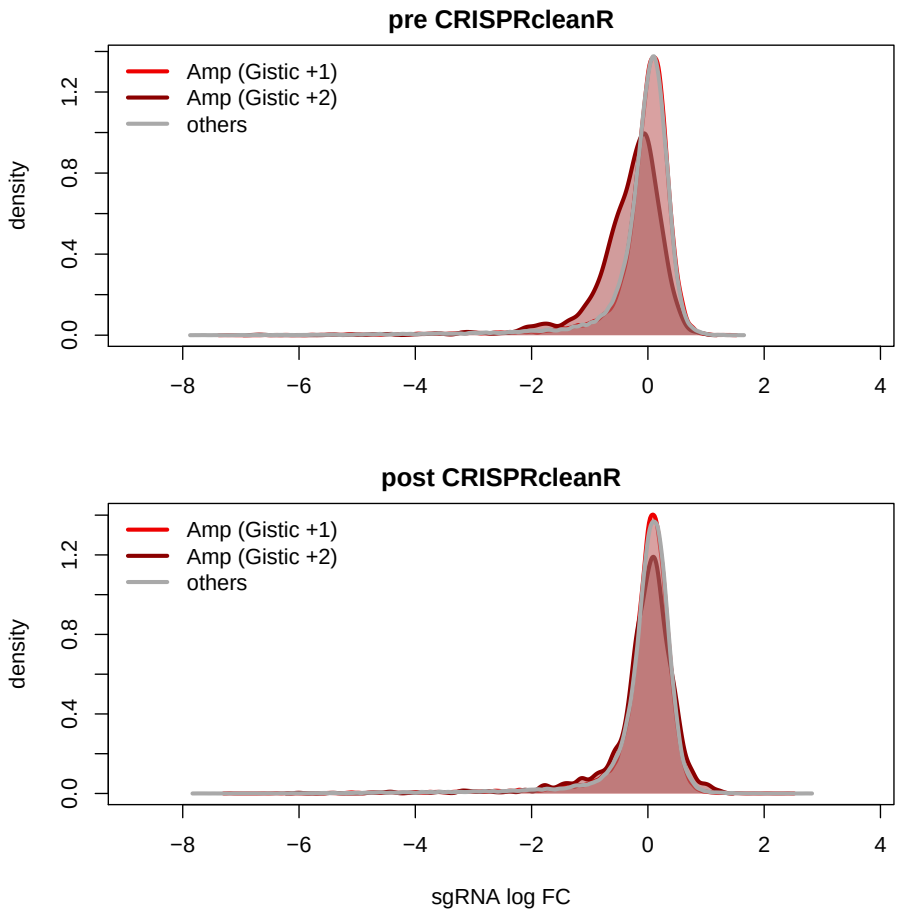
```
##                    Amp (PNs >= 4) notExp Amp (PNs >= 8) notExp
## pre-CRISPRcleanR           2.387775e-52          3.575550e-07
## post-CRISPRcleanR          4.710481e-49          1.871438e-01
##                    Amp (PNs >= 10) notExp     Essential BAGEL Essential
## pre-CRISPRcleanR            4.600584e-05 2.736956e-175   2.898286e-297
## post-CRISPRcleanR           4.089818e-01 3.550035e-159   5.759565e-270
##                    BAGEL Ess.Only BAGEL nonEssential
## pre-CRISPRcleanR    1.591525e-205       8.174798e-81
## post-CRISPRcleanR   2.745117e-183       7.856856e-48

RES$EFFsizes

##                     Dep (PN) Dep (Gistic)    notExp Amp (Gistic +1)
## pre-CRISPRcleanR  0.7269806           NA 0.3141696              NA
## post-CRISPRcleanR 0.2874528           NA 0.2168474              NA
##                    Amp (Gistic +2) Amp (Gistic +1) notExp
## pre-CRISPRcleanR                NA                    NA
## post-CRISPRcleanR               NA                    NA
##                    Amp (Gistic +2) notExp Amp (PNs >= 2) Amp (PNs >= 4)
## pre-CRISPRcleanR                      NA     0.10987879      0.1142627
## post-CRISPRcleanR                     NA     0.05019186      0.0170022
##                    Amp (PNs >= 8) Amp (PNs >= 10) Amp (PNs >= 2) notExp
## pre-CRISPRcleanR       1.0609873       1.34598769             0.3054594
## post-CRISPRcleanR      0.0426818       0.09939445             0.2183217
##                    Amp (PNs >= 4) notExp Amp (PNs >= 8) notExp
## pre-CRISPRcleanR              0.2039852             0.6678038
## post-CRISPRcleanR             0.1916572             0.1506673
##                    Amp (PNs >= 10) notExp Essential BAGEL Essential
## pre-CRISPRcleanR                0.9848744  2.859216        2.223645
## post-CRISPRcleanR               0.1617632  2.945718        2.244647
##                    BAGEL Ess.Only BAGEL nonEssential
## pre-CRISPRcleanR        1.915641          0.2455203
## post-CRISPRcleanR       1.900852          0.1799946
```

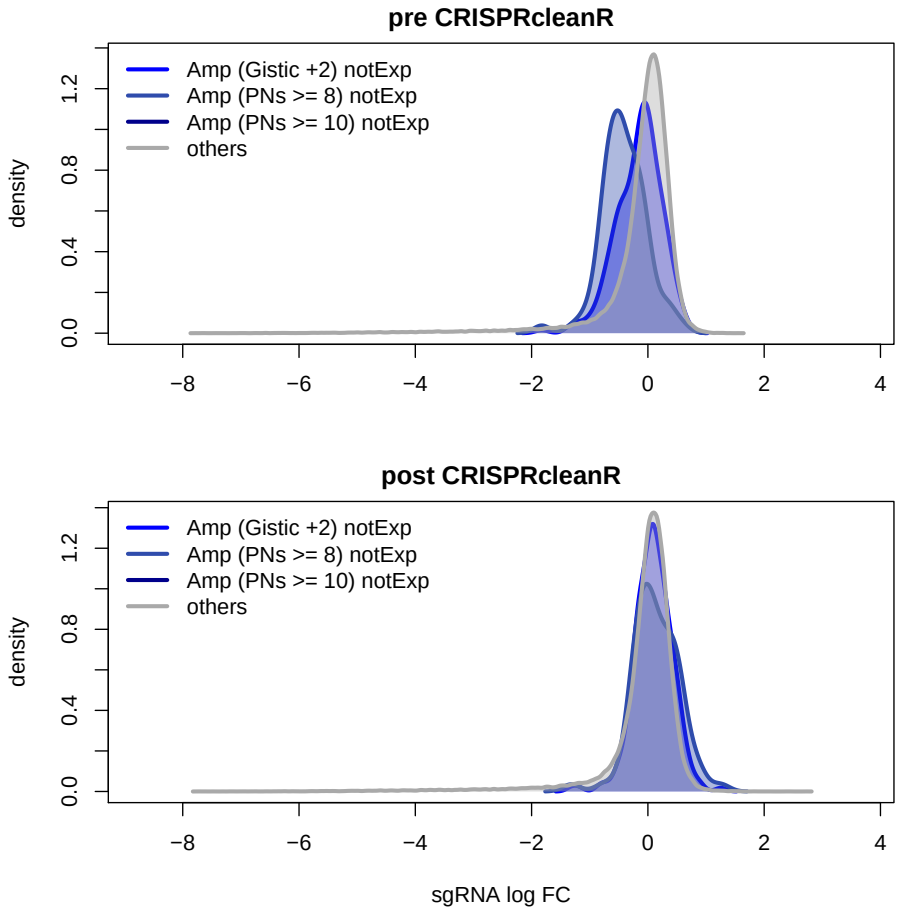


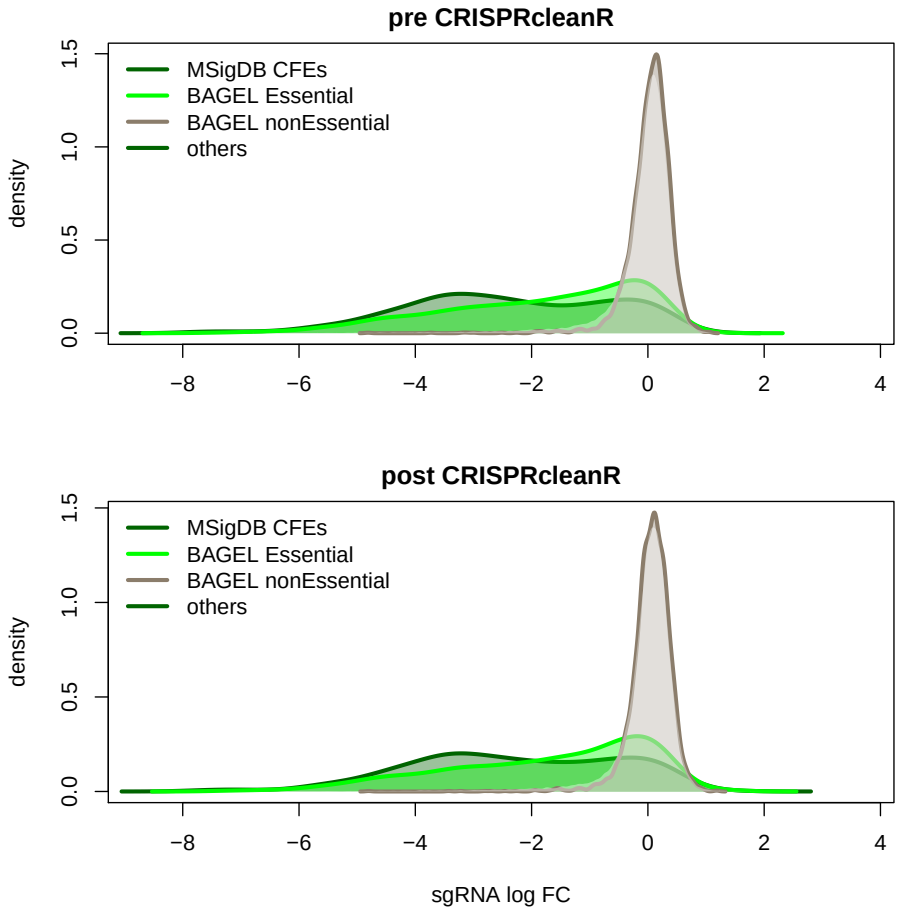
To inspect the variation induced by the CRISPRcleanR correction on distribution densities of sgRNA log fold changes for defined sets of targeted genes prior/post CRISPRcleanR correction, the following function can be also used:

```
ccr.perf_distributions('HT-29',HT.29correctedFCs$corrected_logFCs,
                       libraryAnnotation = KY_Library_v1.0)
```


pre CRISPRcleanR
Amp (PNs >= 8)
Amp (PNs >= 10)
others
density
0.0
0.4
0.8
1.2
−8
−6
−4
−2
0
2
4
post CRISPRcleanR
Amp (PNs >= 8)
Amp (PNs >= 10)
others
density
0.0
0.4
0.8
1.2
−8
−6
−4
−2
0
2
4
sgRNA log FC


pre CRISPRcleanR
Amp (Gistic +1)
Amp (Gistic +2)
others
density
0.0
0.4
0.8
1.2
−8
−6
−4
−2
0
2
4
post CRISPRcleanR
Amp (Gistic +1)
Amp (Gistic +2)
others
density
0.0
0.4
0.8
1.2
−8
−6
−4
−2
0
2
4
sgRNA log FC


pre CRISPRcleanR
Amp (Gistic +2) notExp
Amp (PNs >= 8) notExp
Amp (PNs >= 10) notExp
others
density
0.0
0.4
0.8
1.2
−8
−6
−4
−2
0
2
4
post CRISPRcleanR
Amp (Gistic +2) notExp
Amp (PNs >= 8) notExp
Amp (PNs >= 10) notExp
others
density
0.0
0.4
0.8
1.2
−8
−6
−4
−2
0
2
4
sgRNA log FC



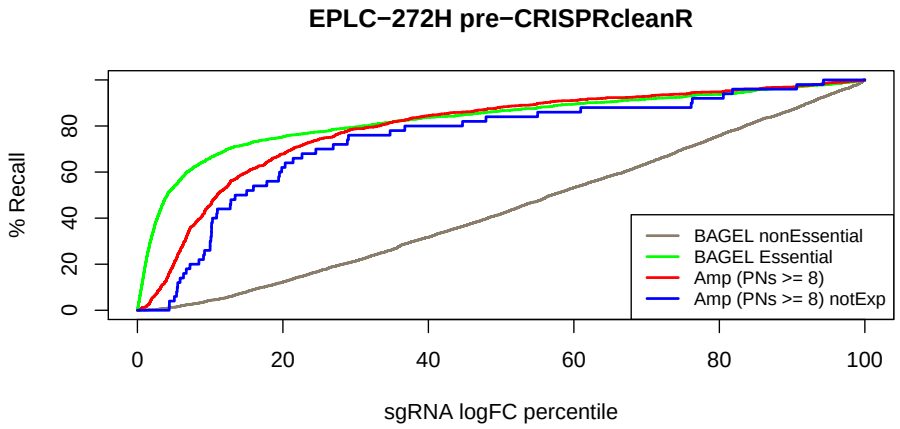
IMPORTANT: The instructions provided regarding what CN/transcriptional data object to pass to the `ccr.perf_statTests` apply also to this function.

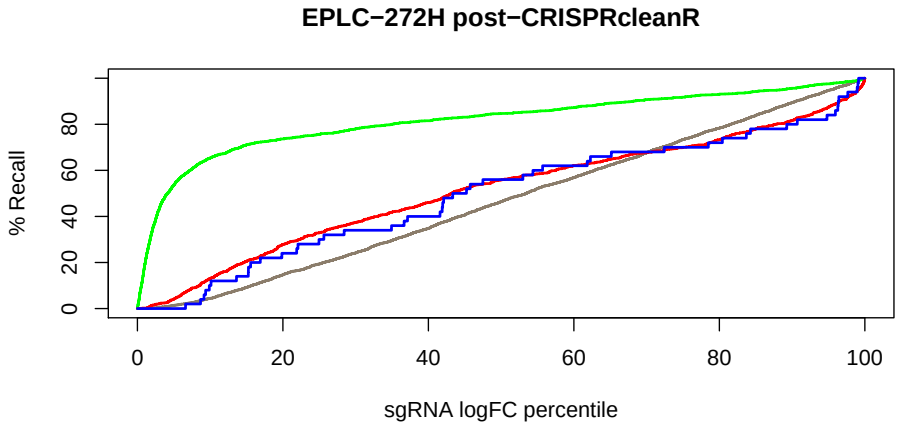
Additional infos on how to use this function can be found in the user reference manual.

2.4 Recall variations following CRISPRcleanR correction for reference, copy number amplified, and non expressed genes

A final analysis that can be done with the CRISPRcleanR package in order to evalute the effect of its correction on the classfication recall of predefined gene sets can be performed with the following call, which can perform the analysis at the sgRNA level:

```
ccr.RecallCurves('EPLC-272H',EPLC.272HcorrectedFCs$corrected_logFCs,
                 libraryAnnotation=KY_Library_v1.0)

## [1] "No gistic CNA scores available for this cell line"
```

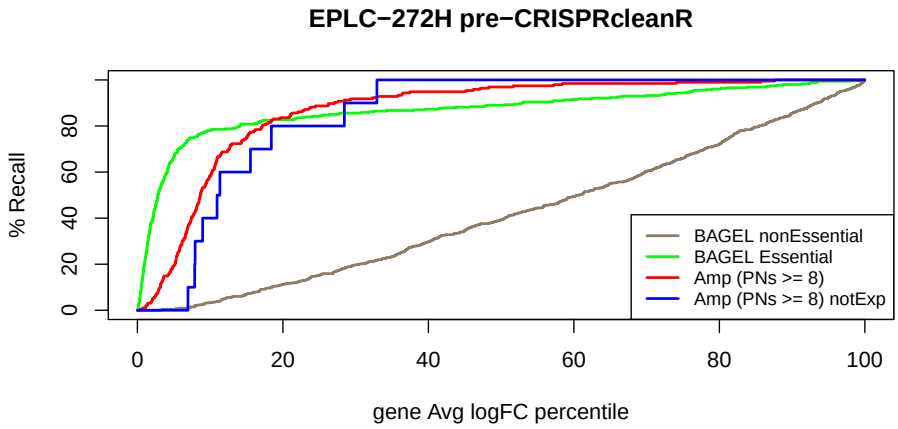


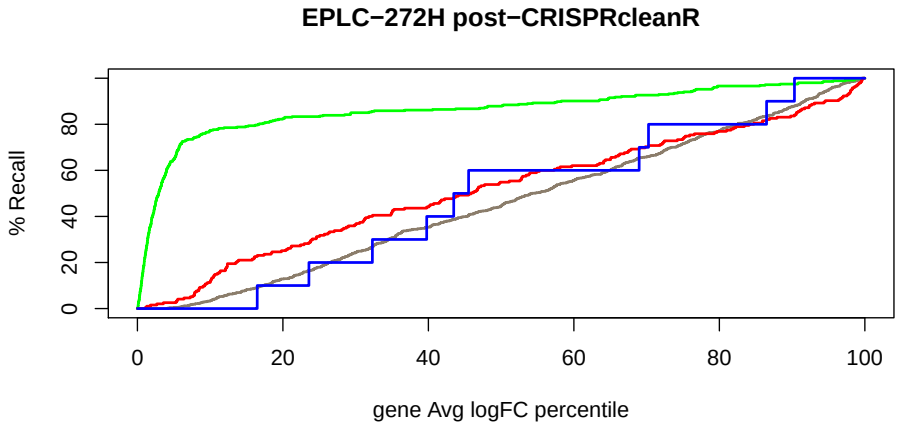



```
##                   BAGEL nonEssential BAGEL Essential Amp (PNs >= 8)
## pre-CRISPRcleanR           0.4405845       0.8282859      0.7921870
## post-CRISPRcleanR          0.4663982       0.8149893      0.5109027
##                   Amp (PNs >= 8) notExp
## pre-CRISPRcleanR                0.7447993
## post-CRISPRcleanR               0.4924973
```

as well as the gene level:

```
ccr.RecallCurves('EPLC-272H',EPLC.272HcorrectedFCs$corrected_logFCs,
                 libraryAnnotation=KY_Library_v1.0,GeneLev = TRUE)

## [1] "No gistic CNA scores available for this cell line"
```






```
##                   BAGEL nonEssential BAGEL Essential Amp (PNs >= 8)
## pre-CRISPRcleanR           0.4188711       0.8704539      0.8697322
## post-CRISPRcleanR          0.4570109       0.8627785      0.5156109
##                   Amp (PNs >= 8) notExp
## pre-CRISPRcleanR                0.8506553
## post-CRISPRcleanR               0.4828279
```

IMPORTANT: The instructions provided regarding what CN/transcriptional data object to pass to the `ccr.perf_statTests` apply also to this function.

Additional infos on how to use this function can be found in the user reference manual.

References

[1] Konstantinos Tzelepis et al. "A CRISPR Dropout Screen Identifies Genetic Vulnerabilities and Therapeutic Targets in Acute Myeloid Leukemia." In: *Cell reports* 17.4 (Oct. 2016), pp. 1193–1205.

[2] Francesco Iorio et al. "Unsupervised correction of gene-independent cell responses to CRISPR-Cas9 targeting". In: *revision* 0.0 (), pp. 0–0.

[3] Wei Li et al. "MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens." In: *Genome Biology* 15.12 (2014), p. 554.

[4] Traver Hart and Jason Moffat. "BAGEL: a computational framework for identifying essential genes from pooled library screens." In: *BMC bioinformatics* 17 (Apr. 2016), p. 164.

[5] A Subramanian et al. "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles". In: *Proceedings of the National Academy of Sciences of the United States of America* 102.43 (2005), p. 15545.