

# YavAch Package Vignette

Olena Yavorska

10 July 2016

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>General Functions</b>	<b>3</b>
2.1	Variation of shRNAs . . . . .	3
2.2	Consistency of phenotypes . . . . .	9
2.3	p53 dependency . . . . .	11
<b>3</b>	<b>Riger Functions</b>	<b>13</b>
3.1	Evaluation of RIGER gene score . . . . .	14
3.2	Evaluation of RIGER gene score across datasets . . . . .	16

## 1 Introduction

This package was designed to analyse the Achilles v2.0 and v2.4 datasets from the Broad Institute, however, any other shRNA screens can also be analysed in a similar fashion through building an “RNAi” object. Namely :

```
library(YavAch)

## Loading required package: ggplot2

## Loading required package: reshape2

## Loading required package: plyr

# The data
data("Achilles2.4_Data") # Achilles v2.4 dataset
data("Achilles2.4_annotation") # The p53 annotation for the Achilles v2.4 dataset

# Read in the shRNA data with a row for each hairpin and columns corresponding to cell
# lines.
head(Achilles2.4_Data[,1:5])
```

```
##           Name      Gene X697_HAEMATOPOIETIC_AND LYMPHOID_TISSUE
## 1 AAAAATGGCATCAACCACCAT  RPS6KA1             -0.04112325
## 2 AAACACATTGGGATGTTCCCT   IGF1R              2.19048750
## 3 AAAGAAGAAAGCTGCAATATCT   TSC1              1.53637450
## 4 AAGCGTGCCGTAGACTGTCCA  CHEK1                NaN
```

```

## 5 AATCTAAGAGAGCTGCCATCG XRCC5 -1.33028525
## 6 AATGAAAGCTCACTCTGGATT PIK3CA 1.09965000
## X7860_KIDNEY A1207_CENTRAL_NERVOUS_SYSTEM
## 1 -0.2428750 -0.7204843
## 2 1.8132215 1.0551227
## 3 1.6716550 1.7499833
## 4 NaN
## 5 -2.9123472 -0.9370367
## 6 0.8221555 0.4313193

```

```

# Create a new RNAi object by filling in the corresponding slots
RNAiObject <- new(Class = "RNAi",
                  genes = Achilles2.4_Data$Gene,
                  sequences = Achilles2.4_Data>Name,
                  cancers = colnames(Achilles2.4_Data) [-(1:2)],
                  cancersP53 = Achilles2.4_annotation,
                  values = Achilles2.4_Data[,-(1:2)])

```

Most of the functions within the package require an “RNAi” object.

## 2 General Functions

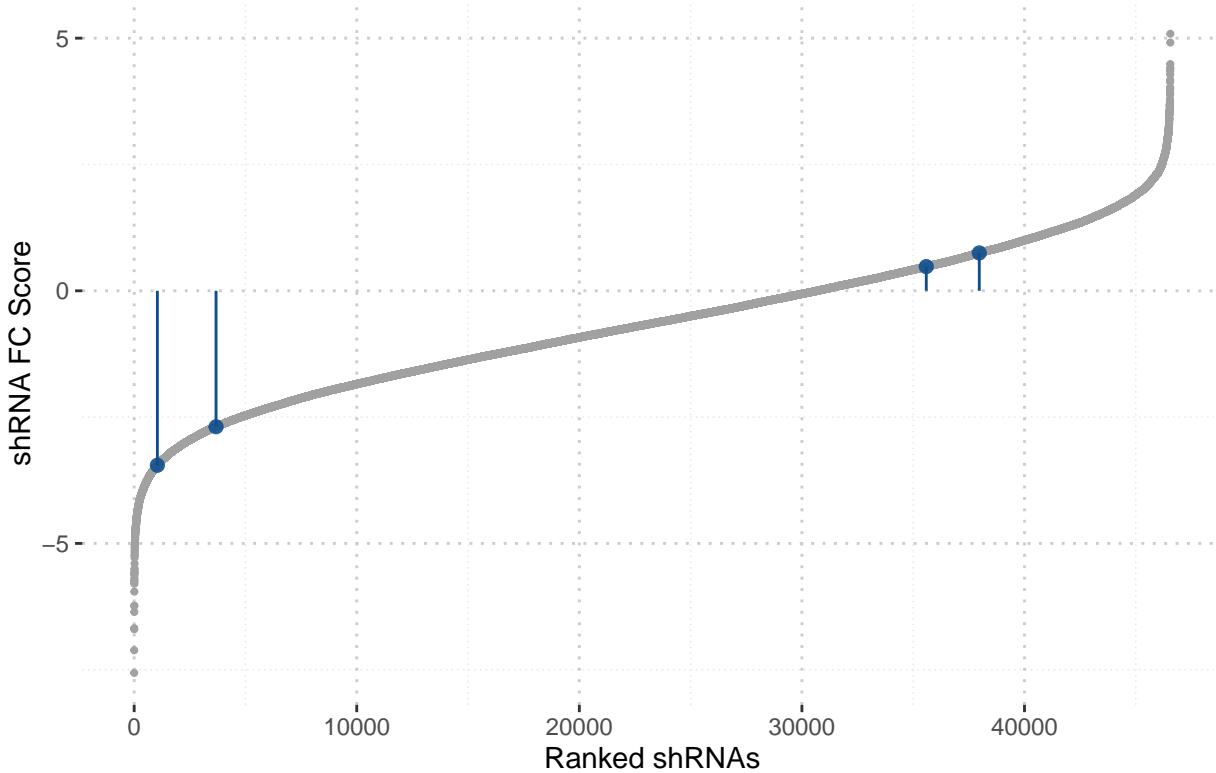
### 2.1 Variation of shRNAs

#### 2.1.1 Variation within cell line

```
# object      an RNAi object
# cancerID   a character string indicating the cell line of interest (partial string
#              matching is sufficient)
# geneName    character string specifying the name of the gene of interest (case
#              sensitive)
# type        can be either "all" or "median"

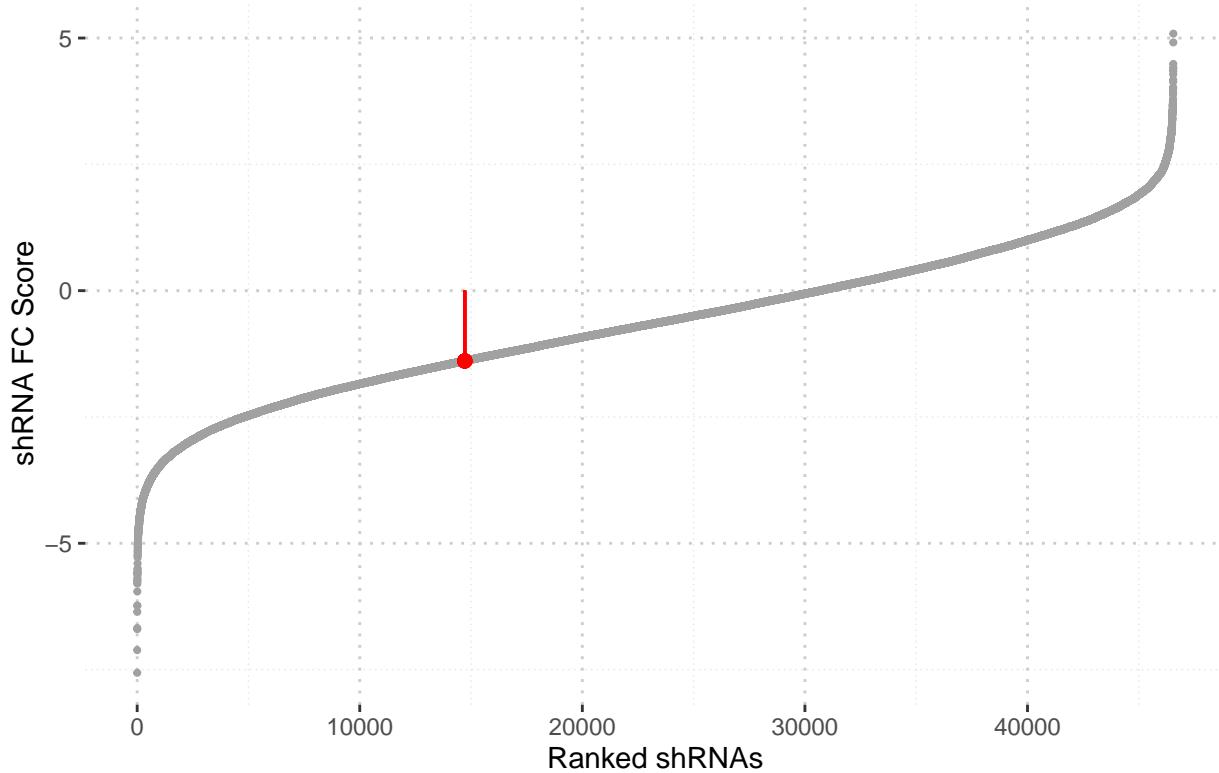
variation.within.cell.line(object = RNAiObject,
                           cancerID = "SLR24",
                           geneName = "MDM4",
                           type = "all")
```

**MDM4 shRNA ranks within the SLR24\_KIDNEY cell line**



```
variation.within.cell.line(object = RNAiObject,
                            cancerID = "SLR24",
                            geneName = "KRAS",
                            type = "median")
```

### KRAS shRNA median rank within the SLR24\_KIDNEY cell line

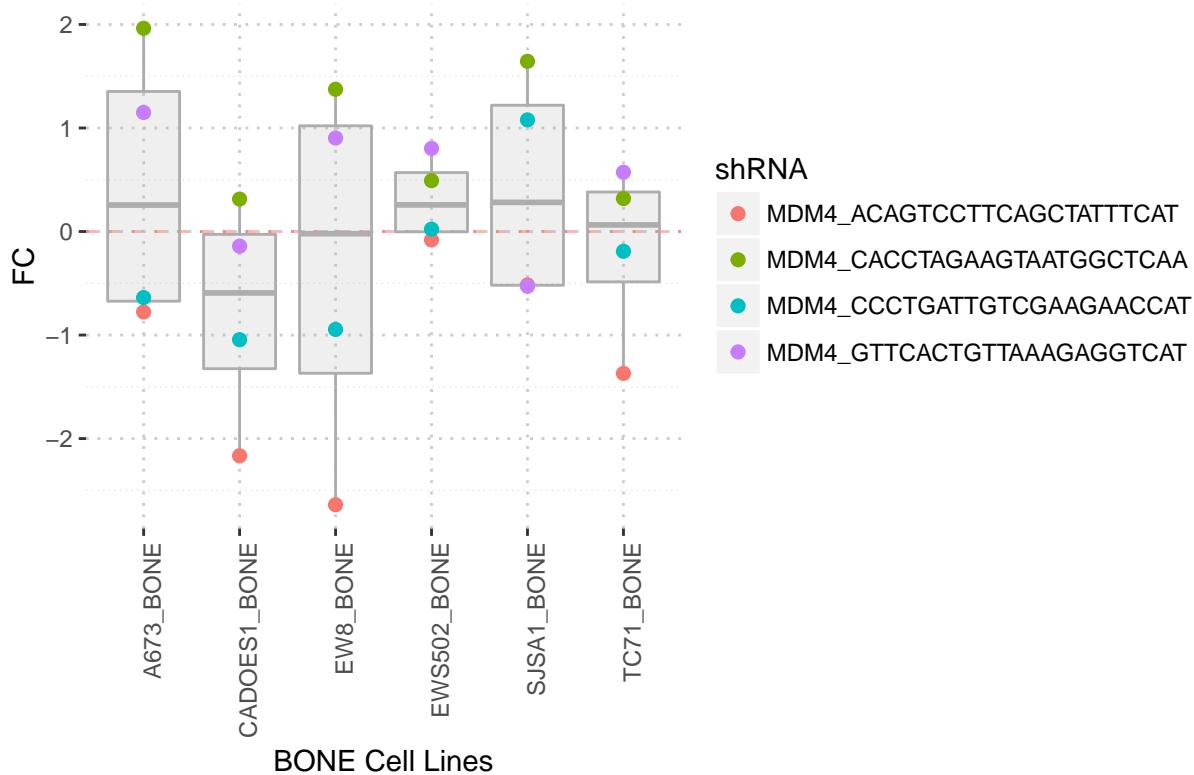


## 2.1.2 Variation within entity

```
# object      an RNAi object
# entityId    the name of one of the entity in question (partial
#               string matching is sufficient)
# geneName    character string specifying the name of the gene of interest (case
#               sensitive)

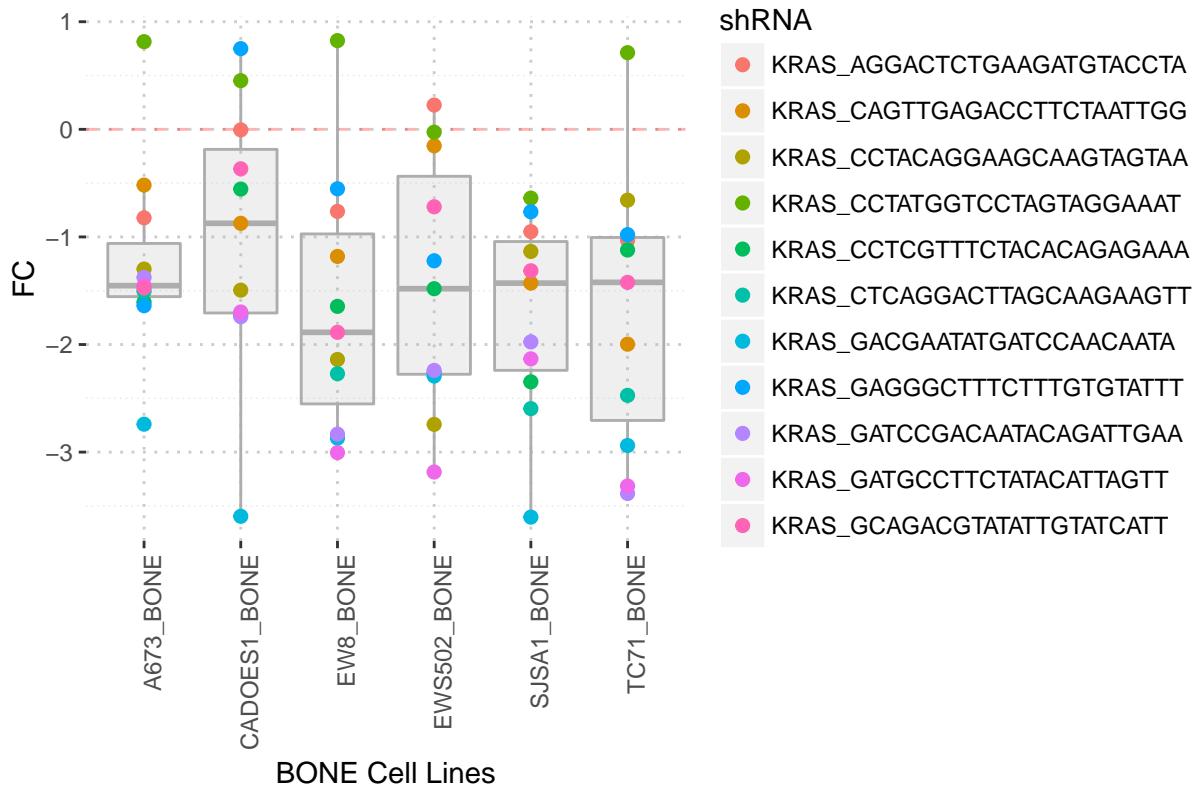
variation.within.entity(object = RNAiObject,
                        entityId = "BONE",
                        geneName = "MDM4")
```

### MDM4 shRNAs grouped by cell line



```
variation.within.entity(object = RNAiObject,
                        entityName = "BONE",
                        geneName = "KRAS")
```

### KRAS shRNAs grouped by cell line



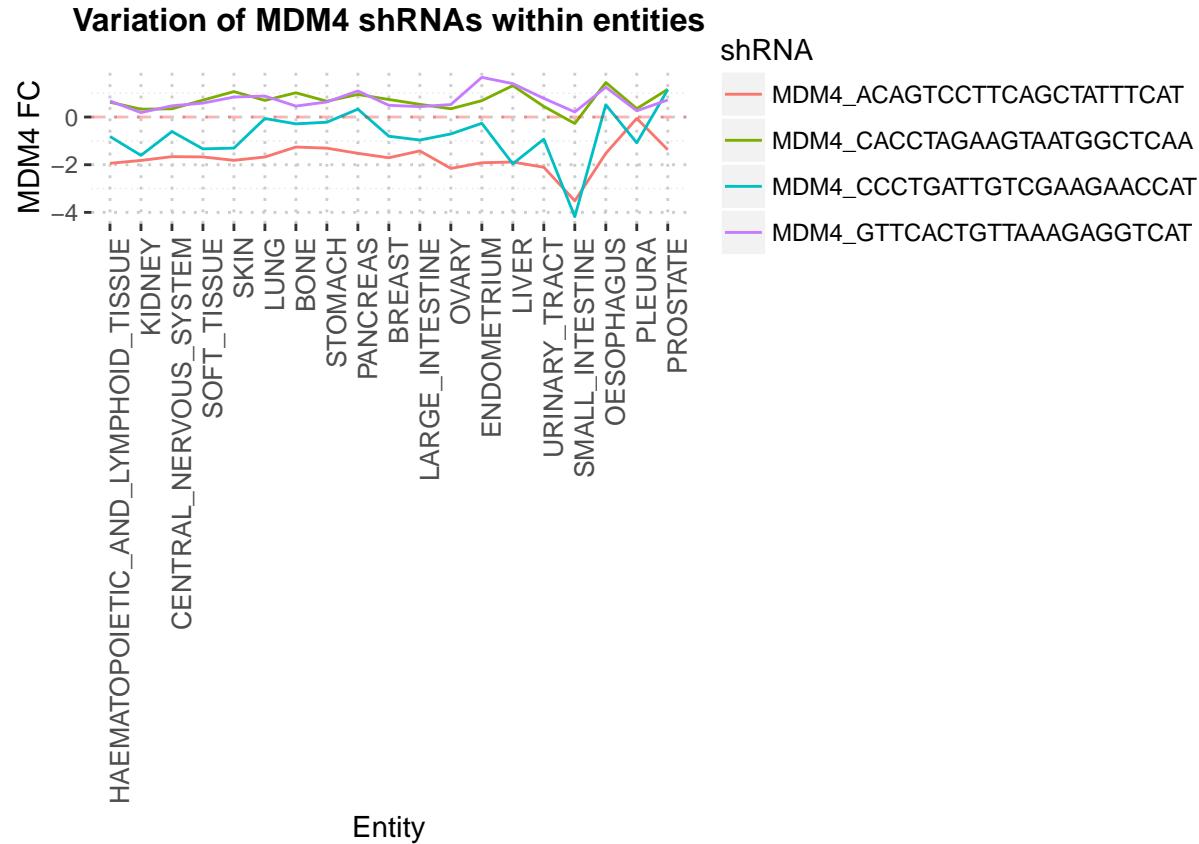
### 2.1.3 Variation within entities

```

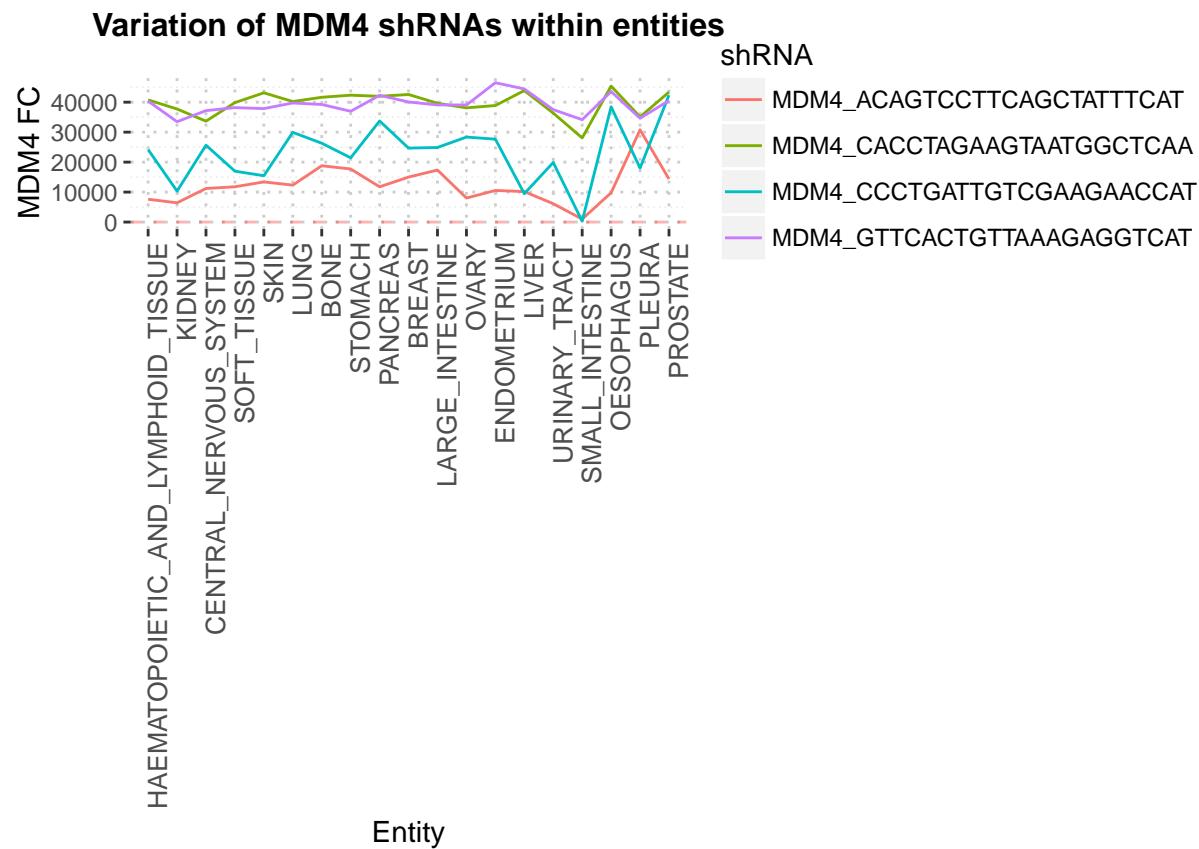
# object      an RNAi object
# geneName   character string specifying the name of the gene of interest (case
#             sensitive)
# type       one of "value" or "rank"
# statistic  one of "mean" or "median"

variation.within.entities(object = RNAiObject,
                           geneName = "MDM4",
                           type = "value",
                           statistic = "mean")

```



```
variation.within.entities(object = RNAiObject,
                           geneName = "MDM4",
                           type = "rank",
                           statistic = "median")
```

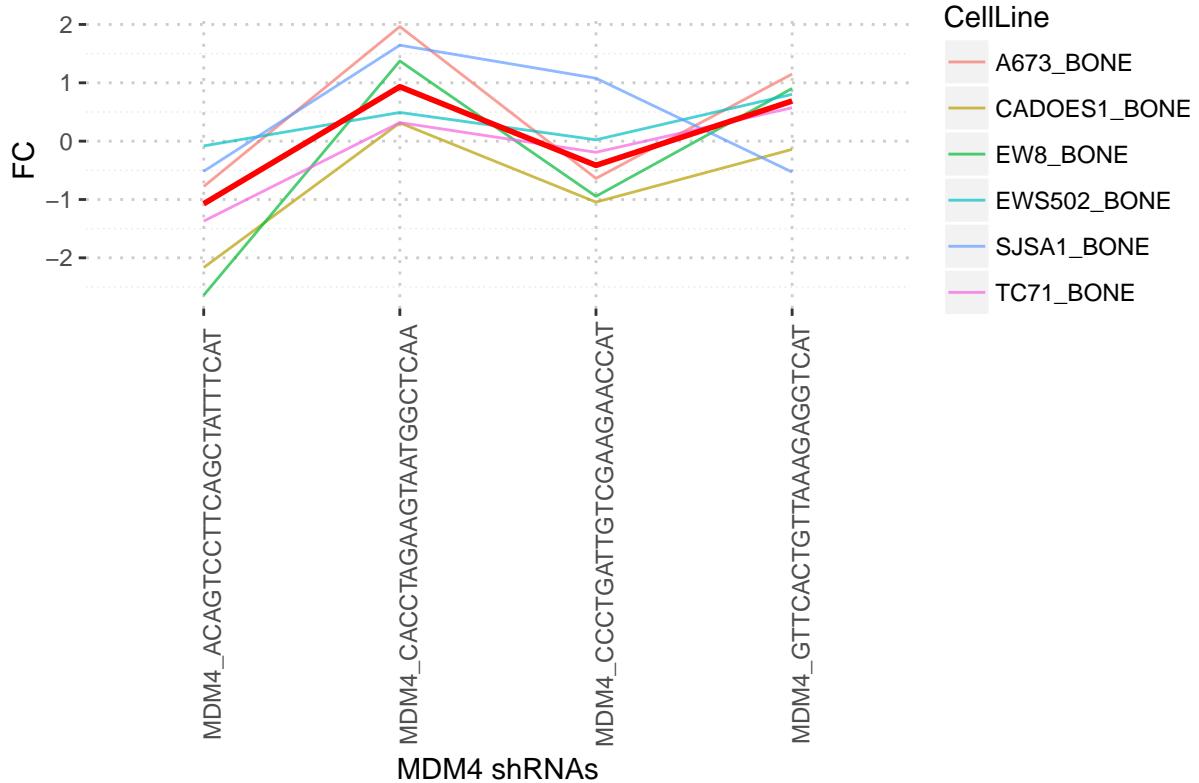


## 2.2 Consistency of phenotypes

```
# object      an RNAi object
# entityName  can either be "all" or the name of one of the entity in question (partial
#               string matching is sufficient)
# geneName    character string specifying the name of the gene of interest (case
#               sensitive)
# type        one of "line" or "box"

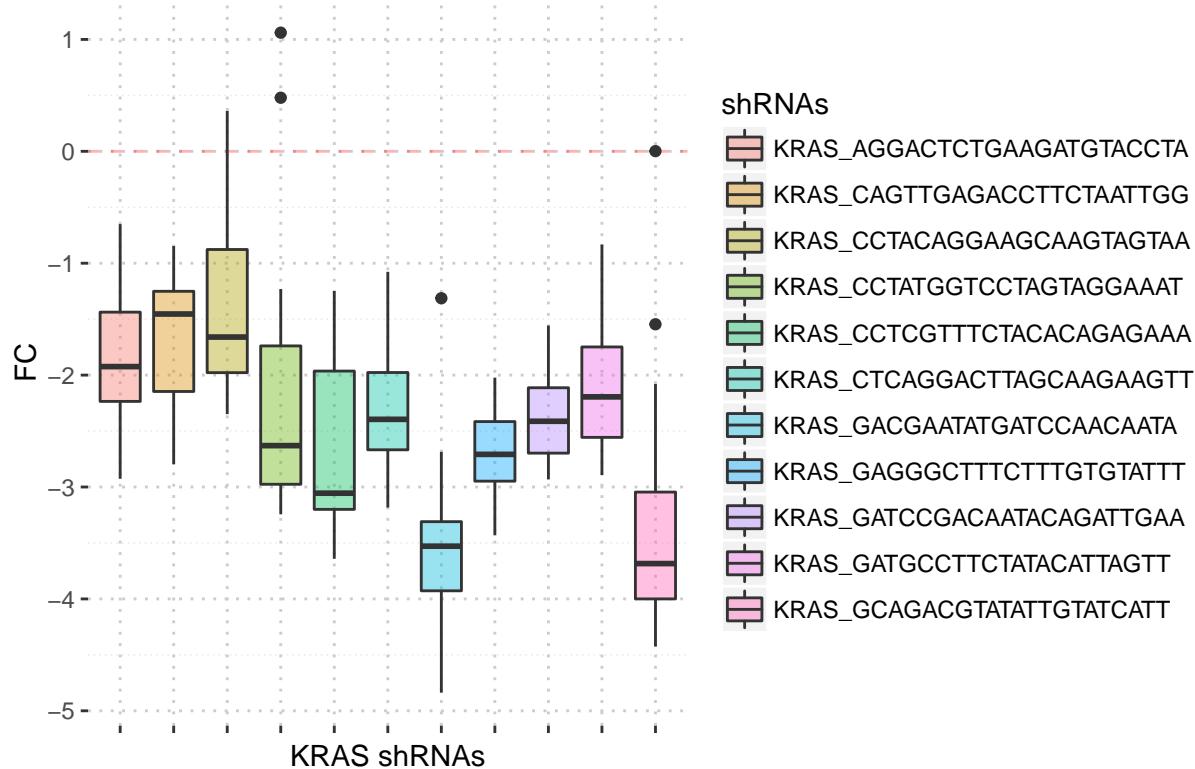
consistency.of.phenotypes(object = RNAiObject,
                           entityName = "BONE",
                           geneName = "MDM4",
                           type = "line")
```

### Consistency of shRNAs across BONE cell lines



```
consistency.of.phenotypes(object = RNAiObject,
                           entityName = "Pancreas",
                           geneName = "KRAS",
                           type = "box")
```

## Consistency of shRNAs across Pancreas cell lines



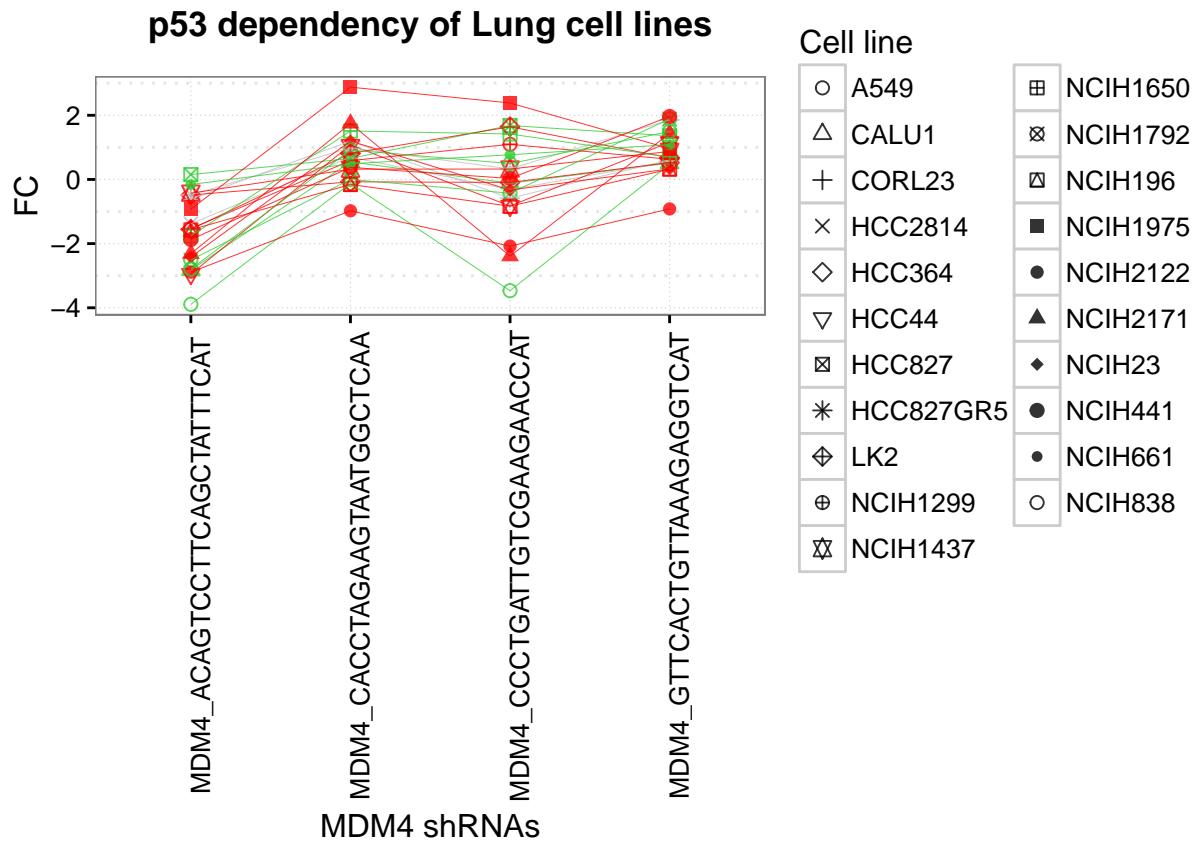
## 2.3 p53 dependency

```

# object      an RNAi object
# entityName  can either be "all" or the name of one of the entity in question (partial
#              string matching is sufficient)
# geneName    character string specifying the name of the gene of interest (case
#              sensitive)
# type        one of "line" or "box"

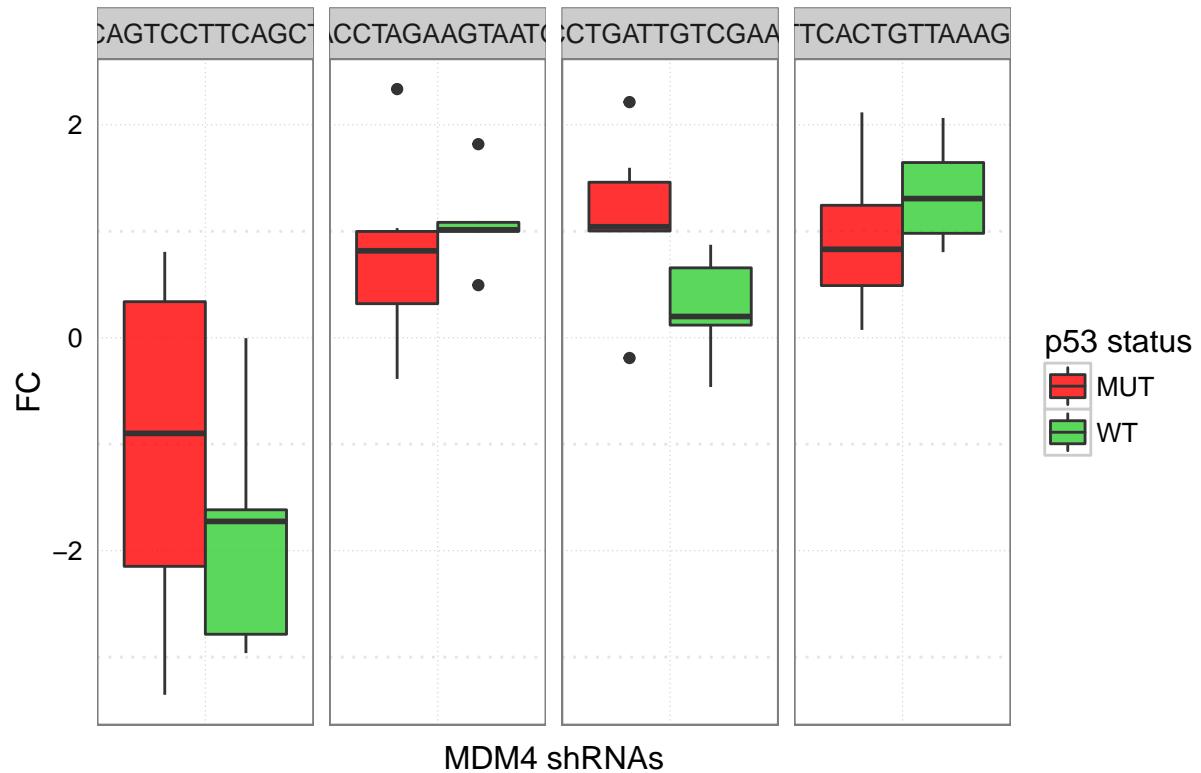
p53.dependency.by.entity(object = RNAiObject,
                           entityName = "Lung",
                           geneName = "MDM4",
                           p53 = "all",
                           type = "line")

```



```
p53.dependency.by.entity(object = RNAiObject,
                           entityName = "Pancreas",
                           geneName = "MDM4",
                           p53 = "all",
                           type = "box")
```

### p53 dependency of Pancreas cell lines



### 3 Riger Functions

The RIGER functions require a separate data structure. In this case a simple data.frame is used (the files used for analysing both Achilles datasets are included in the package). Note that the datasets do not contain all of the entities since some had very few cell lines associated with them.

Achilles2.0 contains :

- “all” (include all cell lines in the dataset)
- “colon”
- “lung”
- “ovary”
- “pancreas”

Achilles2.4 contains :

- “all”
- “breast”
- “cns”
- “colon”
- “haem”
- “lung”
- “ovary”
- “pancreas”
- “skin”

```
data("Achilles2.0") # RIGER data for the Achilles v2.0 dataset  
data("Achilles2.4") # RIGER data for the Achilles v2.4 dataset  
  
head(Achilles2.4[,1:5])
```

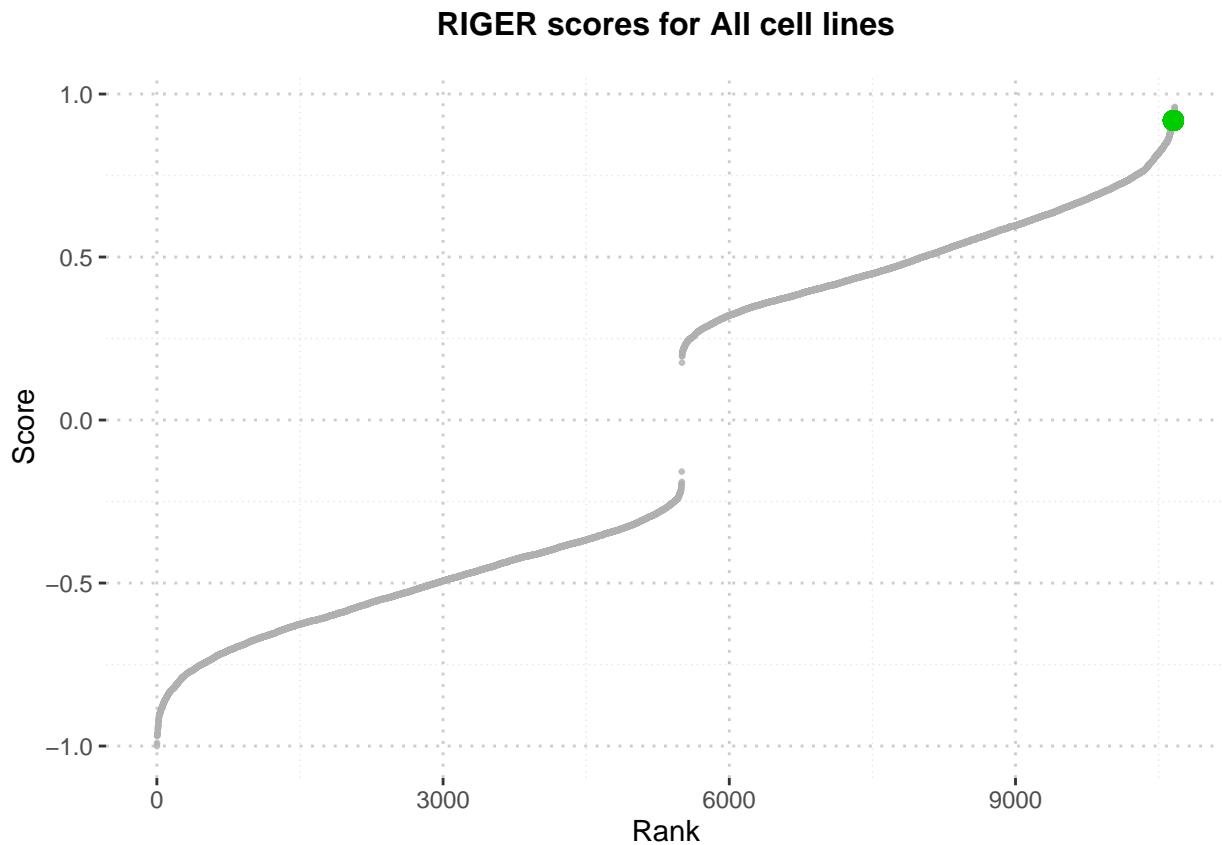
```
##      Gene All_KS_score All_KS_rank All_KS_pvalue All_KS_pvalue_rank  
## 1 1-Dec    -0.6565      9468     0.9291        9842  
## 2 1-Mar    -0.4016      6573     0.6402        6830  
## 3 10-Mar     0.3977      3802     0.3376        3542  
## 4 2-Mar     -0.3035      5551     0.5220        5550  
## 5 3-Mar     -0.6795      9693     0.9103        9635  
## 6 3-Sep     -0.5243      8002     0.7943        8370
```

### 3.1 Evaluation of RIGER gene score

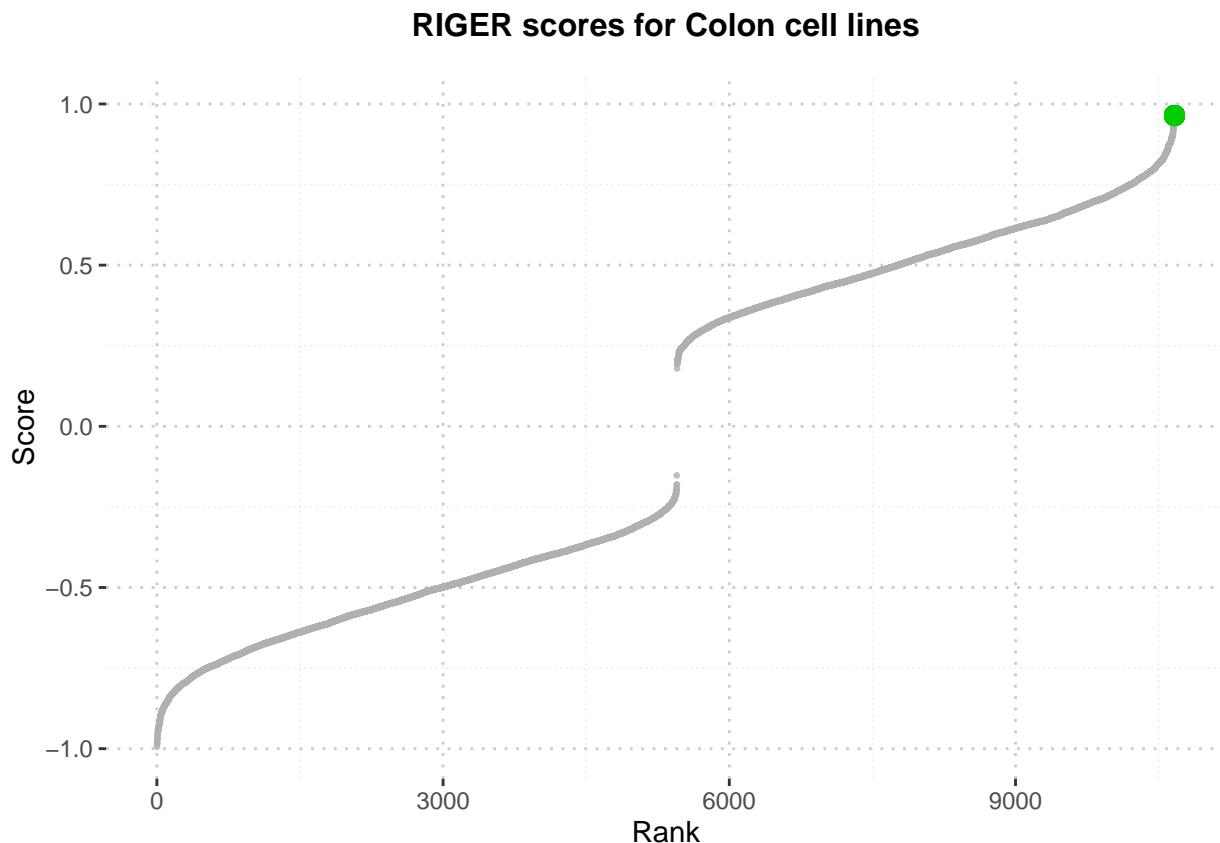
This function allows the user to select the entity (one of those described above) and a gene in order to see that gene's rank when compared to the rest of the dataset.

```
# object      must be an object of the above type
# geneName    character string specifying the name of the gene of interest (case
#              sensitive)
# entityName  can either be "all" or the name of one of the entity in question (partial
#              string matching is sufficient)

riger.gene.comparison(object = Achilles2.4,
                       geneName = "MDM4",
                       entityName = "All")
```



```
riger.gene.comparison(object = Achilles2.4,  
                        geneName = "MDM4",  
                        entityName = "Colon")
```

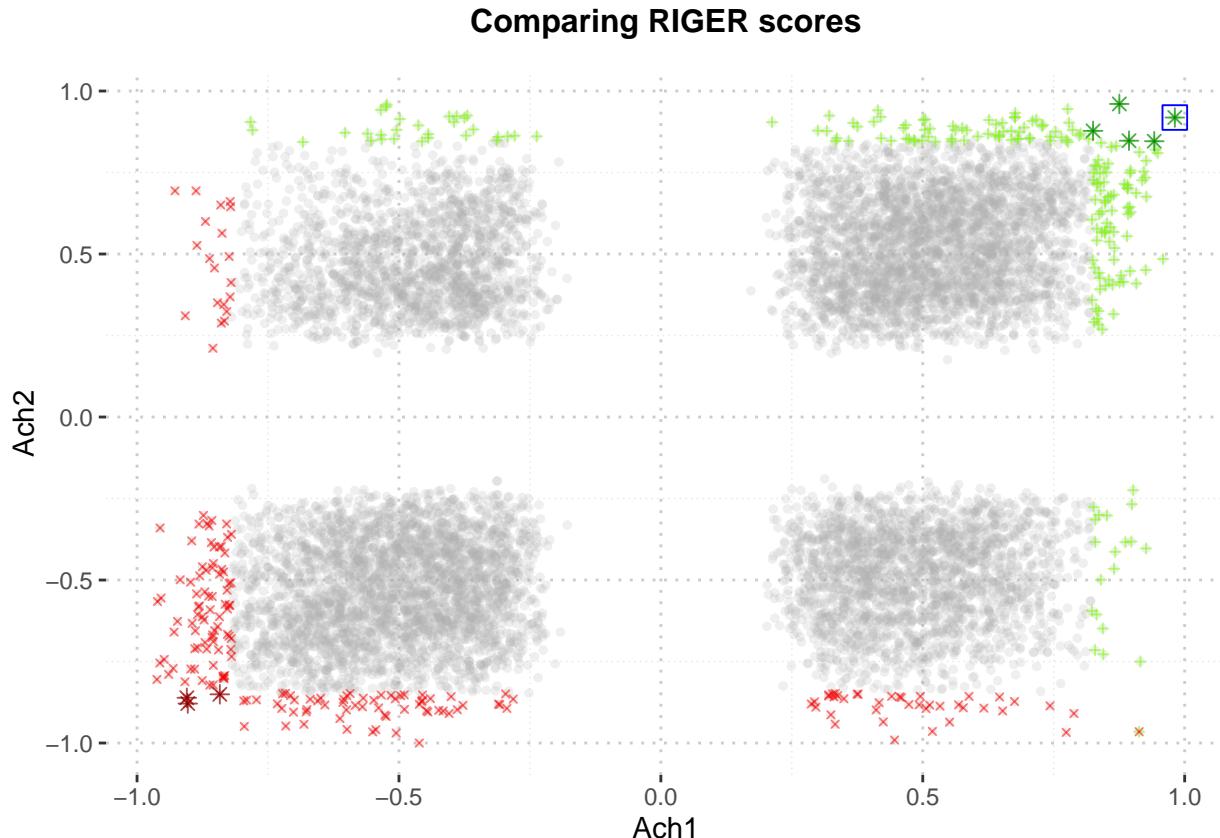


### 3.2 Evaluation of RIGER gene score across datasets

Here one can compare the scores given by the two different datasets. This can be done for the RIGER scores based on “all” data or on “colon”, “lung”, “ovary”, or “pancreas”.

```
# object1      must be an object of the above type
# object2      must be an object of the above type
# object1Name  character vector giving the name of the first dataset used in the plot
# object2Name  character vector giving the name of the second dataset used in the plot
# geneName     character string specifying the name of the gene of interest (case
#               sensitive)
# entityName   can either be "all" or the name of one of the entity in question (partial
#               string matching is sufficient)
# percentage   top and bottom quantiles which should be coloured (number between 1 and
#               100)
# plotName    character string giving the name of the plot

riger.gene.comparison.datasets(object1 = Achilles2.0, object2 = Achilles2.4,
                                object1Name = "Ach1", object2Name = "Ach2",
                                geneName = "MDM4",
                                entityName = "all",
                                percentage = 1,
                                plotName = "Comparing RIGER scores")
```



```
riger.gene.comparison.datasets(object1 = Achilles2.0, object2 = Achilles2.4,
                                 object1Name = "Ach1", object2Name = "Ach2",
                                 geneName = "KRAS",
                                 entityName = "Colon",
                                 percentage = 1,
                                 plotName = "Comparing RIGER scores")
```

