LowMACA: Low frequency Mutation Analysis via Consensus Alignment

Giorgio Melloni , Stefano de Pretis

October 30, 2017

Contents

1 Introduction

The *LowMACA* package is a simple suite of tools to investigate and analyse the profile of the somatic mutations provided by the cBioPortal (via the *cgdsr*). *LowMACA* evaluates the functional impact of somatic mutations by statistically assessing the number of alterations that accumulates on the same residue (or residue that are conserved in Pfam domains). For example, the known driver mutations G12,G13 and Q61 in KRAS can be found on the corresponding residues of other proteins in the RAS family (PF00071) like NRAS and HRAS, but also in less frequently mutated genes like RRAS and RRAS2. The corresponding residues are identified via multiple sequence alignment. Thanks to this approach the user can identify new driver mutations that occur at low frequency at single protein level but emerge at Pfam level. In addition, the impact of known driver mutations can be transferred to other proteins that share a high degree of sequence similarity (like in the RAS family example).

You can conduct an hypothesis driven exploratory analysis using our package simply providing a set of genes and/or pfam domains of your interest. The user is able to choose the kind of tumor and the type of mutations (like missense, nonsense, frameshift etc.). The data are directly downloaded from the largest cancer sequencing projects and aggregated by LowMACA to evaluate the possible functional impact of somatic mutations by spotting the most conserved variations in the cohort of cancer samples. By connecting several proteins that share sequence similarity via consensus alignment, this package is able to statistically assessing the occurrence of mutations on the same residue and ultimately see:

- where mutations fall and what are the involved domains
- what is the frequency of the aberrations and what is the more represented tumor type
- if and where the mutations tend to clusterize
- what is the degree of conservation of the mutated residues
- if there are new driver genes and in particular, driver mutations

2 System Requirements

LowMACA relies on two external resources to work properly.

- Clustal Omega, our trusted aligner (http://www.clustal.org/omega/)
- Ghostscript, a postscript interpreter needed to draw logo plots (http://www.ghostscript.com/)

Clustal Omega is a fast aligner that can be downloaded from the link above. For both Unix and Windows users, remember to have "clustalo" in your PATH variable. In case you cannot set "clustalo" in the PATH, you can always set the clustalo command from inside R, after creating a LowMACA object:

```
#Given a LowMACA object 'lm'
lm <- newLowMACA(genes=c("TP53" , "TP63" , "TP73"))
lmParams(lm)$clustal_cmd <- "/your/path/to/clustalo"
```

If you cannot install clustalomega, we provide a wrapper around EBI web service (http://www.ebi.ac.uk/Tools/webservices/services/msa/clustalo_soap). You just need to set your email as explained in section setup, but you have a limit of 2000 input sequences and perl must be installed with the modules LWP and XML::Simple.

Ghostscript is an interpreter of postscript language and a pdf reader that is used by the R library grImport.

- For Linux users, simply download the program from http://ghostscript.com/download/gsdnld.html and compile it
- For MacOS users there is a dmg installer at http://pages.uoregon.edu/koch/
- For Windows users, download the program from http://ghostscript.com/download/gsdnld.html and then you have 3 options:
  1. Put C:/Program Files/gs/gs9.05/bin in your PATH once for all (Adjust the path to match your gs installation)
  2. Run the command Sys.setenv(R_GSCMD = '"C:/Program Files/gs/gs9.05/bin/gswin32c.exe"') at every new session of R
  3. Put the command showed above in your .Renviron file

More details can be found here: http://pgfe.umassmed.edu/BioconductorGallery/docs/motifStack/motifStack.htm

LowMACA needs an internet connection to:

- retrieve mutation data from cBioPortal,
- draw the Protter-style plot (http://wlab.ethz.ch/protter/start/) and
- use the web service of clustalomega (http://www.ebi.ac.uk/Tools/webservices/services/msa/clustalo_soap)

3 Create a LowMACA object

First of all, we have to define our target genes or pfam domains that we wish to analyse.

3.1 Find your target family of proteins or pfam you want to know more about

```
library(LowMACA)
#User Input
Genes <- c("ADNP","ALX1","ALX4","ARGFX","CDX4","CRX"
        ,"CUX1","CUX2","DBX2","DLX5","DMBX1","DRGX"
        ,"DUXA","ESX1","EVX2","HDX","HLX","HNF1A"
          ,"HOXA1","HOXA2","HOXA3","HOXA5","HOXB1","HOXB3"
                        ,"HOXD3","ISL1","ISX","LHX8")
Pfam <- "PF00046"
```

```
#Construct the object
lm <- newLowMACA(genes=Genes, pfam=Pfam)

## All Gene Symbols correct!

str(lm , max.level=3)

## Formal class 'LowMACA' [package "LowMACA"] with 4 slots
##   ..@ arguments:List of 7
##   .. ..$ genes       : chr [1:28] "ADNP" "ALX1" "ALX4" "ARGFX" ...
##   .. ..$ pfam        : chr "PF00046"
##   .. ..$ pfamAllGenes:'data.frame': 249 obs. of  7 variables:
##   .. ..$ input       :'data.frame': 28 obs. of  7 variables:
##   .. ..$ mode        : chr "pfam"
##   .. ..$ params      :List of 7
##   .. ..$ parallelize :List of 2
##   ..@ alignment: list()
##   ..@ mutations: list()
##   ..@ entropy  : list()
```

Now we have created a *LowMACA* object. In this case, we want to analyse the homeodomain fold pfam (PF00046), considering 28 genes that belong to this clan. If we don't specify the pfam parameter, *LowMACA* proceeds to analyse the entire proteins passed by the genes parameter (we map only canonical proteins, one per gene). Vice versa, if we don't specify the genes parameter, *LowMACA* looks for all the proteins that contain the specified pfam and analyse just the portion of the protein assigned to the domain.

3.2 Change default parameters

A LowMACA object is composed by four slots. The first slot is `arguments` and is filled at the very creation of the object. It contains information as Uniprot name for the proteins associated to the genes, the amino acid sequences, start and end of the selected domains and the default parameters that can be change to start the analysis.

```
#See default parameters
lmParams(lm)
```

```
## $mutation_type
## [1] "missense"
##
## $tumor_type
## [1] "all_tumors"
##
## $min_mutation_number
## [1] 1
##
## $density_bw
## [1] 0
##
## $clustal_cmd
## [1] "clustalo"
##
## $use_hmm
## [1] FALSE
##
## $datum
## [1] FALSE
```

```
#Change some parameters
#Accept sequences even with no mutations
lmParams(lm)$min_mutation_number <- 0
#Changing selected tumor types
#Check the available tumor types in cBioPortal
available_tumor_types <- showTumorType()
head(available_tumor_types)
```

```
##                                                 Adenoid Cystic Carcinoma of the Breast
##                                                                                 "acbc"
##                                                              Adrenocortical Carcinoma
##                                                                                  "acc"
##                                                                 Adenoid Cystic Carcinoma
##                                                                                 "acyc"
## Hypodiploid Acute Lymphoid Leukemia|Infant MLL-Rearranged Acute Lymphoblastic Leukemia
##                                                                                  "all"
##                                                                       Ampullary Carcinoma
##                                                                                 "ampca"
##       Bladder Cancer|Bladder Cancer, Plasmacytoid Variant|Bladder Urothelial Carcinoma
##                                                                                 "blca"
```

```
#Select melanoma, stomach adenocarcinoma, uterine cancer, lung adenocarcinoma,
#lung squamous cell carcinoma, colon rectum adenocarcinoma and breast cancer
```

```
lmParams(lm)$tumor_type <- c("skcm" , "stad" , "ucec" , "luad"
        , "lusc" , "coadread" , "brca")
```

4 Setup

4.1 Align sequences

```
lm <- alignSequences(lm)

## Aligning sequences...
```

This method is basically self explained. It aligns the sequences in the object. If you didn't install clustalomega yet, you can use the web service of clustalomega that we wrapped in our R package. The limit is set to 2000 sequences and it is obviously slower than a local installation. Rememeber to put your own email in the mail command to activate this option since is required by the EBI server.

```
lm <- alignSequences(lm , mail="lowmaca@gmail.com")
```

```
#Access to the slot alignment
myAlignment <- lmAlignment(lm)
str(myAlignment , max.level=2 , vec.len=2)

## List of 4
##  $ ALIGNMENT:'data.frame': 1708 obs. of  8 variables:
##   ..$ domainID      : Factor w/ 28 levels "ARGFX|PF00046|503582|79;135",..: 1 1 1 1 1 ...
##   ..$ Gene_Symbol   : Factor w/ 27 levels "ADNP","ALX1",..: 4 4 4 4 4 ...
##   ..$ Pfam          : Factor w/ 1 level "PF00046": 1 1 1 1 1 ...
##   ..$ Entrez        : Factor w/ 27 levels "1046","127343",..: 21 21 21 21 21 ...
##   ..$ Envelope_Start: num [1:1708] 79 79 79 79 79 ...
##   ..$ Envelope_End  : num [1:1708] 135 135 135 135 135 ...
##   ..$ Align         : int [1:1708] 1 2 3 4 5 ...
##   ..$ Ref           : num [1:1708] 79 80 81 82 83 ...
##  $ SCORE    :List of 2
##   ..$ DIST_MAT     : num [1:28, 1:28] NA 36.4 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   ..$ SUMMARY_SCORE:'data.frame': 28 obs. of  4 variables:
##  $ CLUSTAL  :Formal class 'AAMultipleAlignment' [package "Biostrings"] with 3 slots
##  $ df       :'data.frame': 61 obs. of  2 variables:
##   ..$ consensus   : Factor w/ 18 levels "A","D","E","F",..: 13 13 1 13 15 ...
##   ..$ conservation: num [1:61] 0.411 0.456 ...
```

- ALIGNMENT: mapping from original position to the position in the consensus

- SCORE: some score of distance between the sequences
- CLUSTAL: an object of class `AAMultipleAlignment` as provided by Biostrings R package
- df: Consesus sequence and conservation Trident Score at every position

4.2 Get Mutations and Map Mutations

```
lm <- getMutations(lm)

## Getting mutations from cancers studies...

lm <- mapMutations(lm)
```

These commands produce a change in the slot mutation and provide the results from R cgdsr package.

```
#Access to the slot mutations
myMutations <- lmMutations(lm)
str(myMutations , vec.len=3 , max.level=1)

## List of 3
##  $ data   :'data.frame': 1762 obs. of  8 variables:
##  $ freq   :'data.frame': 7 obs. of  29 variables:
##  $ aligned: num [1:28, 1:61] 0 0 1 0 0 1 0 0 ...
##   ..- attr(*, "dimnames")=List of 2
```

- data: provide the mutations selected from the cBioPortal divided by gene and patient/tumor type
- freq: a table containing the absolute number of mutated patients by gene and tumor type (this is useful to explore the mutational landscape of your genes in the different tumor types)
- aligned: a matrix of m rows, proteins or pfam, and n columns, consensus positions derived from the mapping of the mutations from the original positions to the new consensus

If we want to check what are the most represented genes in terms of number of mutations divided by tumor type, we can simply run:

```
myMutationFreqs <- myMutations$freq
#Showing the first genes
myMutationFreqs[ , 1:10]

##    StudyID Total_Sequenced_Samples ADNP ALX1 ALX4 ARGFX CDX4 CRX CUX1 CUX2
## 1     brca                   3735   15    4    3     3    2   3    6    8
## 2 coadread                   1053   28    6   15     9    4  13   66   39
```

```
## 3     luad                    610    3    8    7     6    9    8   21   20
## 4     lusc                    179    2    6    2     0    3    3   10    3
## 5     skcm                    686   13    3    9    24   25   14   32   68
## 6     stad                    544   13   10    9     2   10   11   23   24
## 7     ucec                    248   11    8    6     2    9    4   20    7
```

This can be useful for a stratified analysis in the future.

4.3 Whole setup

To simplify this setup process, you can use directly the command setup to launch alignSequences, getMutations and mapMutations at once

```
#Local Installation of clustalo
lm <- setup(lm)
#Web Service
lm <- setup(lm , mail="lowmaca@gmail.com")
```

4.4 Custom Data

If you have your own data and you don't need to rely on the cgdsr package, you can use the getMutations or setup method with the parameter repos, like this:

```
#Reuse the downloaded data as a toy example
myOwnData <- myMutations$data
#How myOwnData should look like for the argument repos
str(myMutations$data , vec.len=1)
```

```
## 'data.frame': 1762 obs. of  8 variables:
##  $ Entrez             : int  3199 3213 ...
##  $ Gene_Symbol        : chr  "HOXA2" ...
##  $ Amino_Acid_Letter  : chr  "L" ...
##  $ Amino_Acid_Position: num  298 123 ...
##  $ Amino_Acid_Change  : chr  "L298M" ...
##  $ Mutation_Type      : chr  "Missense_Mutation" ...
##  $ Sample             : chr  "SA236" ...
##  $ Tumor_Type         : chr  "brca" ...
```

```
#Read the mutation data repository instead of using cgdsr package
#Following the process step by step
lm <- getMutations(lm , repos=myOwnData)
```

```
## Filtering mutations from local repository...
```

```
#Setup in one shot
```

```
lm <- setup(lm , repos=myOwnData)

## Aligning sequences...
## Filtering mutations from local repository...
```

5 Statistics

In this step we calculate the general statistics for the entire consensus profile

```
lm <- entropy(lm)

## Making uniform model...

## Assigned bandwidth:  0

#Global Score
myEntropy <- lmEntropy(lm)
str(myEntropy)

## List of 6
##  $ bw              : num 0
##  $ uniform         :function (nmut)
##  $ absval          : num 3.57
##  $ log10pval       : num -17.2
##  $ pvalue          : num 6.8e-18
##  $ conservation_thr: num 0.1

#Per position score
head(myAlignment$df)

##   consensus conservation
## 1         R    0.4110988
## 2         R    0.4558683
## 3         A    0.1505496
## 4         R    0.9493924
## 5         T    0.6493677
## 6         A    0.3113640
```

With the method entropy, we calculate the entropy score and a pvalue against the null hypothesis that the mutations are distributed randomly accross our consensus protein. In addition, a test is performed for every position of the consensus and the output is reported in the slot `alignment`. The position 4 has a conservation score of 0.88 (highly conserved) and the corrected pvalue is significant (qvalue below 0.01). There are signs of positive selection for the position 4. To retrieve the original mutations that generated that cluster, we can use the function lfm

```
significant_muts <- lfm(lm)
#Display original mutations that formed significant clusters (column Multiple_Aln_pos)
head(significant_muts)
```

```
##   Gene_Symbol Amino_Acid_Position Amino_Acid_Change                 Sample
## 1        ALX1                 184             R184M                 FR9547
## 2        ALX4                 218             R218Q         TCGA-AA-3949-01
## 3        ALX4                 218             R218Q coadread_dfci_2016_2227
## 4        ALX4                 218             R218W coadread_dfci_2016_2354
## 5        ALX4                 265             R265Q         TCGA-D8-A1Y1-01
## 6        ALX4                 265             R265Q coadread_dfci_2016_2944
##   Tumor_Type Envelope_Start Envelope_End Multiple_Aln_pos       metric
## 1       luad            133          189               56 1.485743e-02
## 2   coadread            215          271                4 7.428748e-14
## 3   coadread            215          271                4 7.428748e-14
## 4   coadread            215          271                4 7.428748e-14
## 5       brca            215          271               55 3.570331e-02
## 6   coadread            215          271               55 3.570331e-02
##   Entrez  Entry    UNIPROT Chromosome                      Protein.name
## 1   8092 Q15699 ALX1_HUMAN   12q21.31             ALX homeobox protein 1
## 2  60529 Q9H161 ALX4_HUMAN    11p11.2 Homeobox protein aristaless-like 4
## 3  60529 Q9H161 ALX4_HUMAN    11p11.2 Homeobox protein aristaless-like 4
## 4  60529 Q9H161 ALX4_HUMAN    11p11.2 Homeobox protein aristaless-like 4
## 5  60529 Q9H161 ALX4_HUMAN    11p11.2 Homeobox protein aristaless-like 4
## 6  60529 Q9H161 ALX4_HUMAN    11p11.2 Homeobox protein aristaless-like 4
```

```
#What are the genes mutated in position 4 in the consensus?
genes_mutated_in_pos4 <- significant_muts[ significant_muts$Multiple_Aln_pos==4 , 'Gene_Symbol']
```

```
sort(table(genes_mutated_in_pos4))
```

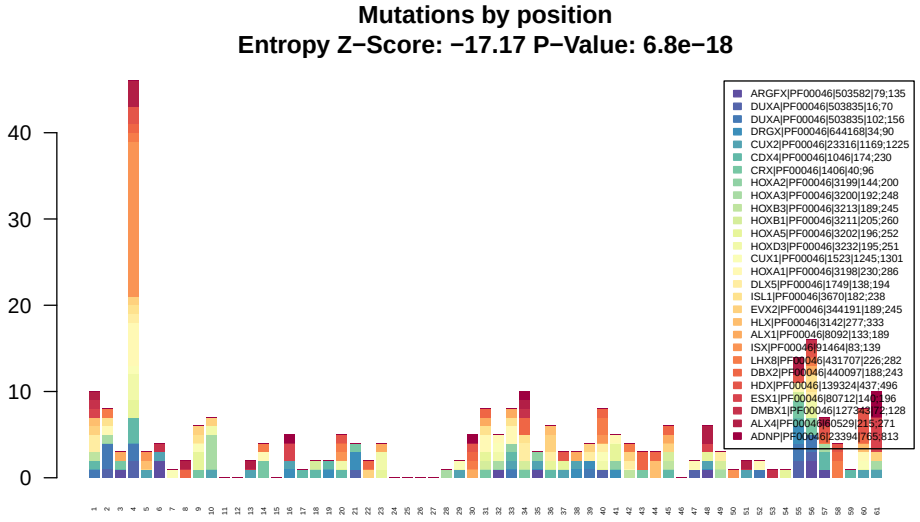
```
## genes_mutated_in_pos4
##  CUX1  DBX2  DLX5  EVX2  ISL1  LHX8   HDX HOXA5  ALX4  CDX4 HOXD3  DUXA
##     1     1     1     1     1     1     2     2     3     3     3     4
## HOXA1   ISX
##     5    18
```

The position 4 accounts for mutations in 13 different genes. The most represented one is ISX (ISX_HUMAN, Intestine-specific homeobox protein).

6 Plot

6.1 Consensus Bar Plot

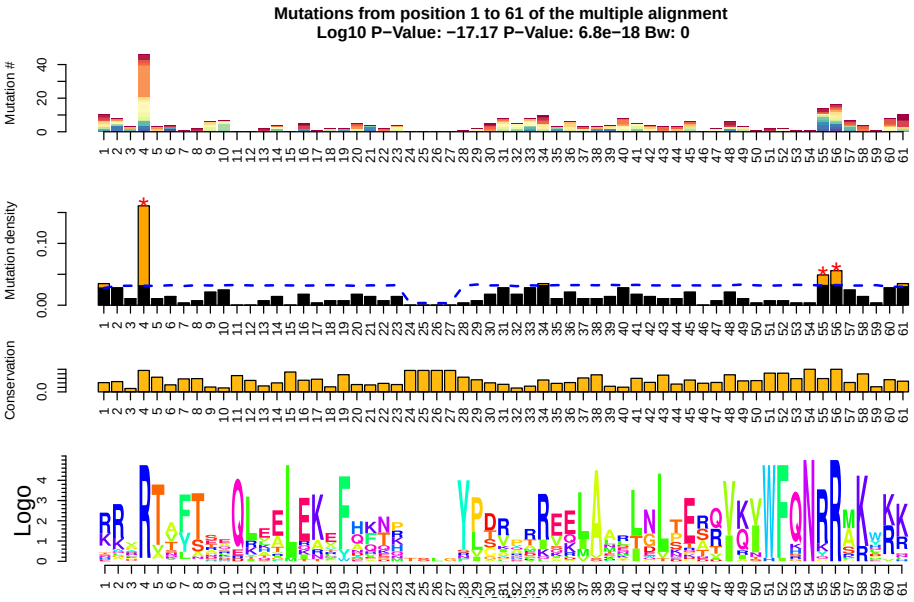
```
bpAll(lm)
```



This barplot shows all the mutations reported on the consensus sequence divided by protein/pfam domain

6.2 LowMACA comprehensive Plot

```
lmPlot(lm)
```



This four layer plot encompasses:

- The bar plot visualize before
- The distribution of mutations against the null hypothesis (blue line) with orange bars representing a pvalue below 0.05 for that position and a red star for qvalue below 0.05
- The Trident score distribution
- The logo plot representing the consensus sequence

6.3 Protter plot

```
#This plot is saved as a png image
protter(lm , filename="homeobox.png")
```

A request to the Protter server is sent and a png file is downloaded with the possible sequence structure of the protein and the significant positions colored in orange and red

7 Data driven workflow

An alternative use of *LowMACA* consists in analysing all the Pfams and single sequences encompassed by a specific set of mutations. For example, it is possible to analyse mutations derived from a cohort of patients to see which Pfams and set of mutations are enriched, following the LowMACA statistics. The function `allPfamAnalysis` takes as input a `data.frame` or the name of a file which contains the set of mutations, analyse all the Pfams that are represented and reports all the significant mutations as output. Moreover, the function `allPfamAnalysis` analyses individually all the mutated genes and reports the significant mutations found by this analysis as part of the output.

```
#Load Homeobox example
data(lmObj)
#Extract the data inside the object as a toy example
myData <- lmMutations(lmObj)$data
#Run allPfamAnalysis on every mutations
significant_muts <- allPfamAnalysis(repos=myData)
```

```
## Warning in mapMutations(object):  We excluded these genes (or domains)
because they have less than 1 mutations
```

```
## NULL
```

```
## Warning in .clustalOAlign(genesData, clustal_cmd, clustalo_filename, mail,
:  There are less than 3 sequences aligned, so no distance matrix can be
calculated
```

```
## Warning in .clustalOAlign(genesData, clustal_cmd, clustalo_filename, mail,
:  There are less than 3 sequences aligned, so no distance matrix can be
calculated
```

```
## Warning in .clustalOAlign(genesData, clustal_cmd, clustalo_filename, mail,
:  There are less than 3 sequences aligned, so no distance matrix can be
calculated
```

```
## Warning in mapMutations(object):  We excluded these genes (or domains)
because they have less than 1 mutations
```

```
## NULL
```

```
#Show the result of alignment based analysis
head(significant_muts$AlignedSequence)
```

```
##   Gene_Symbol Multiple_Aln_pos Pfam_ID binomialPvalue Amino_Acid_Position
## 1        ALX4               4 PF00046      0.8329828                 218
## 2        CDX4               4 PF00046      0.5009311                 177
## 3        CDX4               4 PF00046      0.5009311                 177
## 4        CDX4               4 PF00046      0.5009311                 177
```

```
## 5        CUX1               4 PF00046      0.1908599               1248
## 6        CUX1               4 PF00046      0.1908599               1248
##   Amino_Acid_Change          Sample Tumor_Type Envelope_Start Envelope_End
## 1             R218Q TCGA-AA-3949-01   coadread            215          271
## 2             R177C TCGA-D3-A2JO-06       skcm            174          230
## 3             R177C TCGA-AP-A0LM-01       ucec            174          230
## 4             R177C   MEL-Ma-Mel-85       skcm            174          230
## 5            R1248W TCGA-ER-A193-06       skcm           1245         1301
## 6            R1248W TCGA-BG-A18B-01       ucec           1245         1301
##         metric Entrez  Entry    UNIPROT Chromosome
## 1 6.767185e-12  60529 Q9H161 ALX4_HUMAN    11p11.2
## 2 6.767185e-12   1046 O14627 CDX4_HUMAN     Xq13.2
## 3 6.767185e-12   1046 O14627 CDX4_HUMAN     Xq13.2
## 4 6.767185e-12   1046 O14627 CDX4_HUMAN     Xq13.2
## 5 6.767185e-12   1523 P39880 CUX1_HUMAN     7q22.1
## 6 6.767185e-12   1523 P39880 CUX1_HUMAN     7q22.1
##                         Protein.name
## 1 Homeobox protein aristaless-like 4
## 2             Homeobox protein CDX-4
## 3             Homeobox protein CDX-4
## 4             Homeobox protein CDX-4
## 5        Homeobox protein cut-like 1
## 6        Homeobox protein cut-like 1
```

```
#Show all the genes that harbor significant mutations
unique(significant_muts$AlignedSequence$Gene_Symbol)
```

```
##  [1] ALX4  CDX4  CUX1  DBX2  DUXA  EVX2  HDX   HOXA1 HOXA5 HOXD3 ISL1  ISX
## [13] LHX8
## 13 Levels: ALX4 CDX4 CUX1 DBX2 DUXA EVX2 HDX HOXA1 HOXA5 HOXD3 ISL1 ... LHX8
```

```
#Show the result of the Single Gene based analysis
head(significant_muts$SingleSequence)
```

```
##                Gene_Symbol Amino_Acid_Position Amino_Acid_Change
## PF00046.DUXA.1        DUXA                 103             R103Q
## PF00046.DUXA.2        DUXA                 103             R103L
## PF00046.DUXA.3        DUXA                 103             R103Q
## PF00046.DUXA.4        DUXA                  17              R17H
## PF00046.DUXA.5        DUXA                 105             R105C
## PF00046.DUXA.6        DUXA                 105             R105H
##                          Sample Tumor_Type Envelope_Start Envelope_End
## PF00046.DUXA.1 TCGA-A8-A094-01       brca            102          156
## PF00046.DUXA.2     LUAD-S00488       luad            102          156
## PF00046.DUXA.3 TCGA-B5-A11E-01       ucec            102          156
## PF00046.DUXA.4 TCGA-D1-A0ZS-01       ucec             16           70
## PF00046.DUXA.5 TCGA-60-2722-01       lusc            102          156
```

```
## PF00046.DUXA.6          587284   coadread            102          156
##                Multiple_Aln_pos     metric Entrez  Entry    UNIPROT
## PF00046.DUXA.1                2 0.03360168 503835 A6NLW8 DUXA_HUMAN
## PF00046.DUXA.2                2 0.03360168 503835 A6NLW8 DUXA_HUMAN
## PF00046.DUXA.3                2 0.03360168 503835 A6NLW8 DUXA_HUMAN
## PF00046.DUXA.4                2 0.03360168 503835 A6NLW8 DUXA_HUMAN
## PF00046.DUXA.5                4 0.03360168 503835 A6NLW8 DUXA_HUMAN
## PF00046.DUXA.6                4 0.03360168 503835 A6NLW8 DUXA_HUMAN
##                Chromosome              Protein.name
## PF00046.DUXA.1   19q13.43 Double homeobox protein A
## PF00046.DUXA.2   19q13.43 Double homeobox protein A
## PF00046.DUXA.3   19q13.43 Double homeobox protein A
## PF00046.DUXA.4   19q13.43 Double homeobox protein A
## PF00046.DUXA.5   19q13.43 Double homeobox protein A
## PF00046.DUXA.6   19q13.43 Double homeobox protein A
```

```
#Show all the genes that harbor significant mutations
unique(significant_muts$SingleSequence$Gene_Symbol)
```

```
## [1] "DUXA"
```

The parameter `allLowMACAObjects` can be used to specify the name of the file where all the Pfam analyses will be stored (by default this information is not stored, because the resulting file can be huge, according to the size of the input dataset). In this case, all the analysed Pfams are stored as *LowMACA* objects and they can be loaded and analysed with the usual *LowMACA* workflow.

8 Summary

Copy and paste on your R console and perform the entire analysis by yourself. You need Ghostscript to see all the plots.

```
library(LowMACA)
Genes <- c("ADNP","ALX1","ALX4","ARGFX","CDX4","CRX"
                        ,"CUX1","CUX2","DBX2","DLX5","DMBX1","DRGX"
                        ,"DUXA","ESX1","EVX2","HDX","HLX","HNF1A"
                        ,"HOXA1","HOXA2","HOXA3","HOXA5","HOXB1","HOXB3"
                        ,"HOXD3","ISL1","ISX","LHX8")
Pfam <- "PF00046"
lm <- newLowMACA(genes=Genes , pfam=Pfam)
lmParams(lm)$tumor_type <- c("skcm" , "stad" , "ucec" , "luad"
        , "lusc" , "coadread" , "brca")
lmParams(lm)$min_mutation_number <- 0
lm <- setup(lm , mail="lowmaca@gmail.com")
```

```
        lm <- entropy(lm)
        lfm(lm)
        bpAll(lm)
        lmPlot(lm)
        protter(lm)
```

9 Session Information

```
sessionInfo()
```

```
## R version 3.4.2 Patched (2017-10-07 r73498)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows Server 2012 R2 x64 (build 9600)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=C
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] LowMACA_1.10.0
##
## loaded via a namespace (and not attached):
##  [1] SummarizedExperiment_1.8.0 reshape2_1.4.2
##  [3] lattice_0.20-35            colorspace_1.3-2
##  [5] htmltools_0.3.6            stats4_3.4.2
##  [7] rtracklayer_1.38.0         yaml_2.1.14
##  [9] XML_3.98-1.9               R.oo_1.21.0
## [11] BiocParallel_1.12.0        BiocGenerics_0.24.0
## [13] RColorBrewer_1.1-2         matrixStats_0.52.2
## [15] GenomeInfoDbData_0.99.1    plyr_1.8.4
## [17] stringr_1.2.0              zlibbioc_1.24.0
## [19] Biostrings_2.46.0          munsell_0.4.3
## [21] R.methodsS3_1.7.1          htmlwidgets_0.9
## [23] evaluate_0.10.1            Biobase_2.38.0
## [25] knitr_1.17                 IRanges_2.12.0
## [27] GenomeInfoDb_1.14.0        parallel_3.4.2
## [29] MotIV_1.34.0               highr_0.6
## [31] Rcpp_0.12.13               scales_0.5.0
## [33] backports_1.1.1            rGADEM_2.26.0
## [35] BSgenome_1.46.0            seqLogo_1.44.0
## [37] DelayedArray_0.4.0         S4Vectors_0.16.0
## [39] XVector_0.18.0             cgdsr_1.2.6
## [41] Rsamtools_1.30.0           BiocStyle_2.6.0
```

```
## [43] digest_0.6.12              stringi_1.1.5
## [45] GenomicRanges_1.30.0       grid_3.4.2
## [47] ade4_1.7-8                 rprojroot_1.2
## [49] tools_3.4.2                bitops_1.0-6
## [51] magrittr_1.5               RCurl_1.95-4.8
## [53] Matrix_1.2-11              LowMACAAnnotation_0.99.3
## [55] data.table_1.10.4-3        grImport_0.9-0
## [57] rmarkdown_1.6              GenomicAlignments_1.14.0
## [59] motifStack_1.22.0          compiler_3.4.2
```