

LowMACA: Low frequency Mutation Analysis via Consensus Alignment

Giorgio Melloni , Stefano de Pretis

May 15, 2016

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/s>). 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 .Renv file

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

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)

##                                Adrenocortical Carcinoma
##                                "acc"
##                                Adenoid Cystic Carcinoma
##                                "acyc"
##                                Infant MLL-Rearranged Acute Lymphoblastic Leukemia
##                                "all"
##                                Bladder Cancer|Bladder Cancer, Plasmacytoid Variant|Bladder Urothelial Carcinoma
##                                "blca"
##                                Breast Invasive Carcinoma|Breast cancer patient xenografts
##                                "brca"
## Clear Cell Renal Cell Carcinoma|Multiregion Sequencing of Clear Cell Renal Cell Carcinoma
##                                "ccrcc"

#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. Remember 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 ...
## ..$ 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: Consensus 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': 1335 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	1263	14	2	2	3	2	3	5	7
## 2	coadread	434	11	4	4	5	0	4	10	9
## 3	luad	610	3	8	7	6	9	8	20	20
## 4	lusc	178	2	6	2	0	3	3	9	3
## 5	skcm	559	10	3	6	19	24	13	29	59
## 6	stad	438	11	7	9	2	7	9	18	21
## 7	ucec	248	11	8	6	2	9	4	17	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': 1335 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 3.56
## $ log10pval         : num -13.9
## $ pvalue            : num 1.2e-14
## $ 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 across 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	Tumor_Type
## 1	ALX4	218	R218Q	TCGA-AA-3949-01	coadread
## 2	CDX4	177	R177C	TCGA-AP-A0LM-01	ucec
## 3	CDX4	177	R177C	MEL-Ma-Me1-85	skcm
## 4	CDX4	177	R177C	TCGA-D3-A2J0-06	skcm
## 5	CUX1	1248	R1248W	TCGA-BG-A18B-01	ucec
## 6	CUX1	1248	R1248W	TCGA-ER-A193-06	skcm

```
## Envelope_Start Envelope_End Multiple_Aln_pos metric Entrez Entry UNIPROT
## 1 215 271 4 5.256835e-13 60529 Q9H161 ALX4_HUMAN
## 2 174 230 4 5.256835e-13 1046 014627 CDX4_HUMAN
## 3 174 230 4 5.256835e-13 1046 014627 CDX4_HUMAN
## 4 174 230 4 5.256835e-13 1046 014627 CDX4_HUMAN
## 5 1245 1301 4 5.256835e-13 1523 P39880 CUX1_HUMAN
## 6 1245 1301 4 5.256835e-13 1523 P39880 CUX1_HUMAN
## Chromosome Protein.name
## 1 11p11.2 Homeobox protein aristaless-like 4
## 2 Xq13.2 Homeobox protein CDX-4
## 3 Xq13.2 Homeobox protein CDX-4
## 4 Xq13.2 Homeobox protein CDX-4
## 5 7q22.1 Homeobox protein cut-like 1
## 6 7q22.1 Homeobox protein cut-like 1

#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))
```

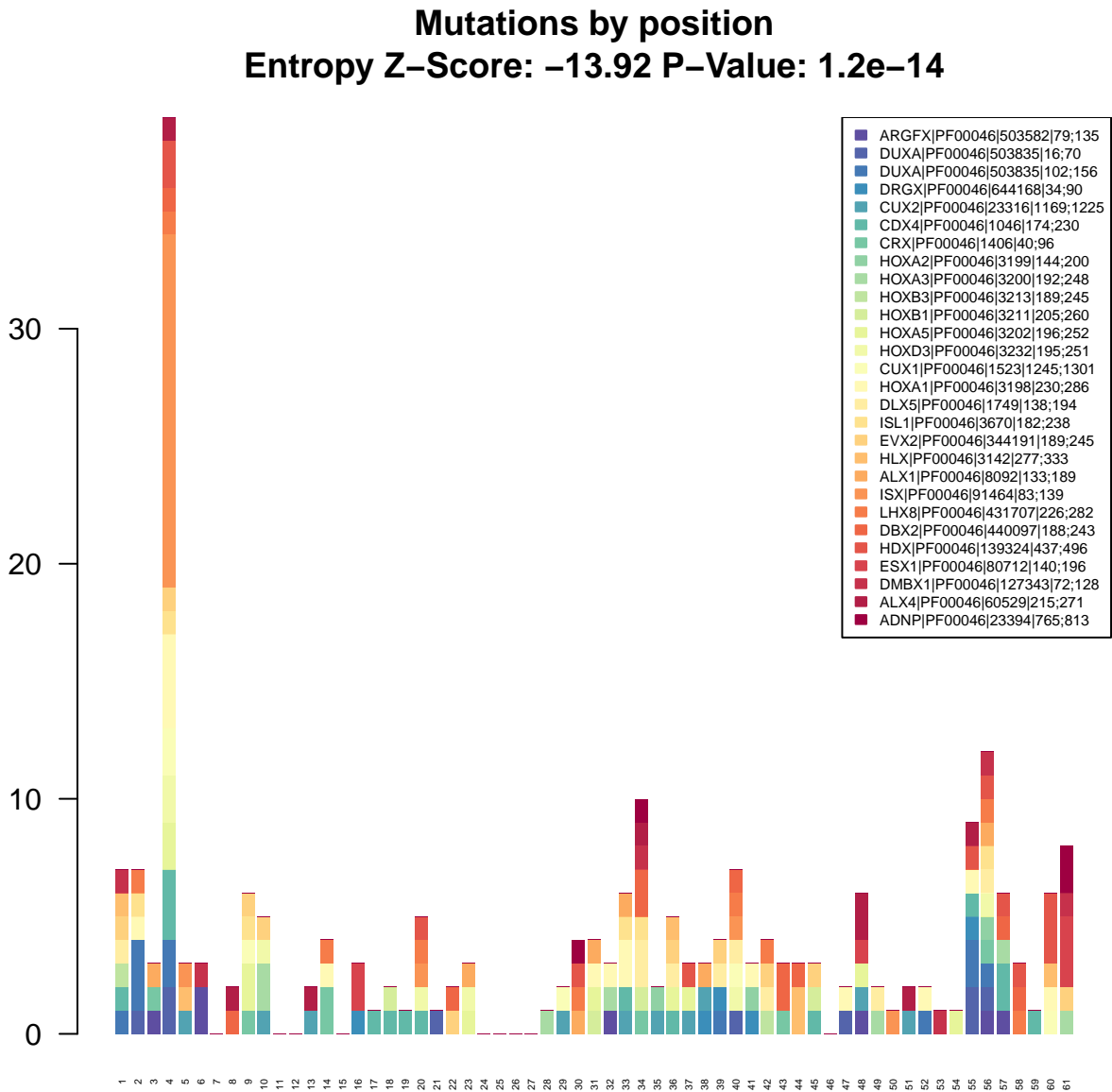
##	genes_mutated_in_pos4
##	ALX4 DBX2 EVX2 ISL1 LHX8 CUX1 HDX HOXA5 HOXD3 CDX4 DUXA HOXA1 ISX
##	1 1 1 1 1 2 2 2 2 3 4 4 15

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

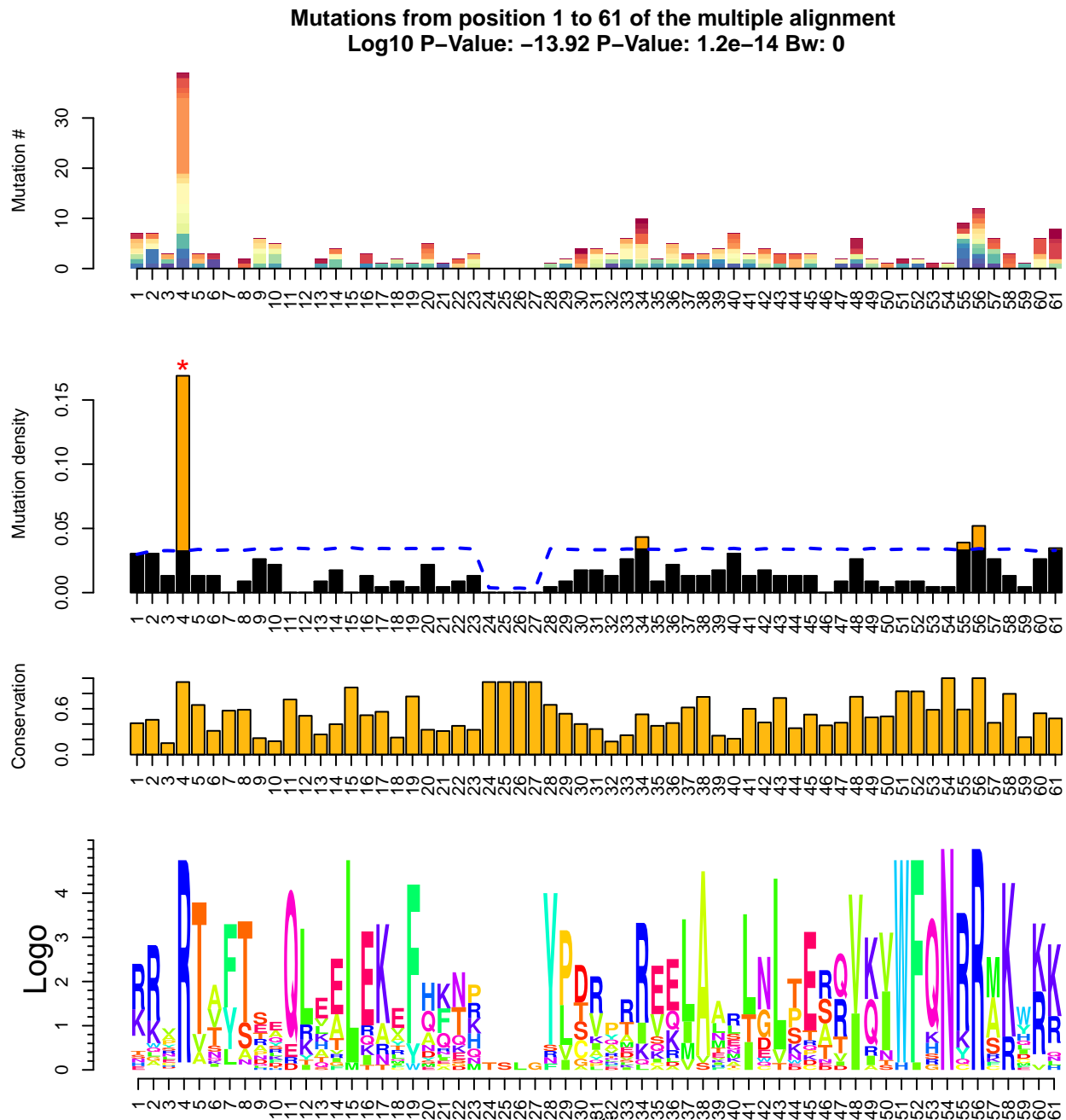
bpAll (1m)



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

6.2 LowMACA comprehensive Plot

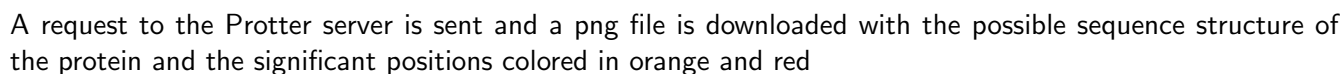
```
lmPlot(lm)
```



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

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



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 .clustal0Align(genesData, clustal_cmd, clustalo_filename, mail, : There are
less than 3 sequences aligned, so no distance matrix can be calculated

## Warning in .clustal0Align(genesData, clustal_cmd, clustalo_filename, mail, : There are
less than 3 sequences aligned, so no distance matrix can be calculated

## Warning in .clustal0Align(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      metric
## 1          R218Q TCGA-AA-3949-01   coadread          215          271 2.294669e-12
## 2          R177C TCGA-D3-A2J0-06      skcm          174          230 2.294669e-12
## 3          R177C TCGA-AP-A0LM-01      ucec          174          230 2.294669e-12
## 4          R177C MEL-Ma-Mel-85      skcm          174          230 2.294669e-12
## 5          R1248W TCGA-ER-A193-06      skcm         1245         1301 2.294669e-12
## 6          R1248W TCGA-BG-A18B-01      ucec         1245         1301 2.294669e-12
##   Entrez  Entry  UNIPROT Chromosome      Protein.name
## 1  60529 Q9H161 ALX4_HUMAN   11p11.2 Homeobox protein aristaless-like 4
## 2   1046 014627 CDX4_HUMAN    Xq13.2   Homeobox protein CDX-4
## 3   1046 014627 CDX4_HUMAN    Xq13.2   Homeobox protein CDX-4
## 4   1046 014627 CDX4_HUMAN    Xq13.2   Homeobox protein CDX-4
## 5   1523 P39880 CUX1_HUMAN    7q22.1   Homeobox protein cut-like 1
## 6   1523 P39880 CUX1_HUMAN    7q22.1   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 LHX8
## Levels: ALX4 CDX4 CUX1 DBX2 DUXA EVX2 HDX HOXA1 HOXA5 HOXD3 ISL1 ISX LHX8

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

##           Gene_Symbol Amino_Acid_Position Amino_Acid_Change      Sample
## PF00046.DUXA.1      DUXA                103              R103Q TCGA-A8-A094-01
## PF00046.DUXA.2      DUXA                103              R103L   LUAD-S00488
## PF00046.DUXA.3      DUXA                103              R103Q TCGA-B5-A11E-01
## PF00046.DUXA.4      DUXA                 17              R17H TCGA-D1-A0ZS-01
## PF00046.DUXA.5      DUXA                105              R105C TCGA-60-2722-01
## PF00046.DUXA.6      DUXA                105              R105H      587284
##           Tumor_Type Envelope_Start Envelope_End Multiple_Aln_pos      metric Entrez
## PF00046.DUXA.1      brca                102              156          2 0.03431009 503835
## PF00046.DUXA.2      luad                102              156          2 0.03431009 503835
## PF00046.DUXA.3      ucec                102              156          2 0.03431009 503835
## PF00046.DUXA.4      ucec                 16               70          2 0.03431009 503835
## PF00046.DUXA.5      lusc                102              156          4 0.03431009 503835
## PF00046.DUXA.6      coadread            102              156          4 0.03431009 503835
##           Entry      UNIPROT Chromosome      Protein.name
## PF00046.DUXA.1 A6NLW8 DUXA_HUMAN   19q13.43 Double homeobox protein A
## PF00046.DUXA.2 A6NLW8 DUXA_HUMAN   19q13.43 Double homeobox protein A
## PF00046.DUXA.3 A6NLW8 DUXA_HUMAN   19q13.43 Double homeobox protein A
## PF00046.DUXA.4 A6NLW8 DUXA_HUMAN   19q13.43 Double homeobox protein A
## PF00046.DUXA.5 A6NLW8 DUXA_HUMAN   19q13.43 Double homeobox protein A
## PF00046.DUXA.6 A6NLW8 DUXA_HUMAN   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 3.3.0 (2016-05-03)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.9.5 (Mavericks)
##
## locale:
## [1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] LowMACA_1.4.2
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.5           GenomeInfoDb_1.8.2     formatR_1.4
## [4] RColorBrewer_1.1-2    plyr_1.8.3             highr_0.6
## [7] XVector_0.12.0        bitops_1.0-6           R.methodsS3_1.7.1
## [10] tools_3.3.0           zlibbioc_1.18.0        evaluate_0.9
## [13] lattice_0.20-33       BSgenome_1.40.0        motifStack_1.16.2
## [16] parallel_3.3.0        seqLogo_1.38.0         rtracklayer_1.32.0
## [19] cgdsr_1.2.5           stringr_1.0.0          knitr_1.13
## [22] Biostrings_2.40.0     S4Vectors_0.10.0       IRanges_2.6.0
## [25] stats4_3.3.0          ade4_1.7-4             grid_3.3.0
## [28] grImport_0.9-0        Biobase_2.32.0         data.table_1.9.6
## [31] MotIV_1.28.0          XML_3.98-1.4           BiocParallel_1.6.2
## [34] rGADEM_2.20.0         LowMACAAnnotation_0.99.3 reshape2_1.4.1
## [37] magrittr_1.5          GenomicAlignments_1.8.0 Rsamtools_1.24.0
## [40] GenomicRanges_1.24.0  scales_0.4.0           BiocGenerics_0.18.0
## [43] SummarizedExperiment_1.2.2 BiocStyle_2.0.2        colorspace_1.2-6
## [46] stringi_1.0-1         RCurl_1.95-4.8         munsell_0.4.3
## [49] chron_2.3-47          R.oo_1.20.0
```