Package 'CRAMAC'

June 5, 2012

Type Package

Title CRAMAC

Version 1.0

Date 2012-05-22

Author Marco Fracassetti

Maintainer Marco Fracassetti <marcofraca@gmail.com>

Description Several functions used at CRA-MAC, it works with linux.

Depends simpleaffy, gplots, agricolae, seqinr, csbl.go, Biostrings

License GPL-2

LazyLoad yes

R topics documented:

`align_control` *align_control*

Description

The function check where the primers anneal in the maize genome.

Usage

```
align_control(primer_L, primer_R, bp_max = 10000, e_value = 10,
 lista = "/home/marco/maizeseq/ZmB73_RefGen_v2/lista_chr",
 dir_chr = "/home/marco/maizeseq/ZmB73_RefGen_v2/")
```

Arguments

`primer_L`	Left primer.
`primer_R`	Right primer.
`bp_max`	Max bp between primers, (default 1000).
`e_value`	Score for the blast, (default 10).
`lista`	Filename of chromosome list, (default "/home/marco/maizeseq/ZmB73_RefGen_v2/lista_chr").
`dir_chr`	Directory whit the chromosome data, (default "/home/marco/maizeseq/ZmB73_RefGen_v2/").

Value

print where the primers anneal.

Author(s)

Marco Fracassetti

Examples

```
#primer_L="CAATTAAGCGAGGCGATGAGC"
#primer_R=revcomp("CTCTACATCATCGCGCAGTG")
#align_control(primer_L,primer_R)
```

anva *anva*

Description

The function calculate test of variance and post-hoc test.

Usage

```
anva(dati, gruppi, p_value = 0.05)
```

Arguments

dati	Vector with dependent variables.
gruppi	Vector with indipendent variables (type factor).
p_value	p value for significance (default 0.05)

Details

For normality shapiro test.
For omoscedasticity bartlett and fligner tests.
N v O v ANOVA
N v O n Welch one-way ANOVA
N n O v Kruskal test
N n O n Friedman test

Value

Print in terminal the information about tests.
Return table with means standar deviation and comparison group of post-hoc tests.

Author(s)

Marco Fracassetti

Examples

```
# a <- c(13.47, 10.21, 15.10, 14.65, 9.03, 15.14, 6.09, 3.43, 5.95, 10.72)
# b <- c(4.02, 14.03, 3.09, 10.25, 6.25, 1.00, 7.50, 10.18, 6.07, 2.03)
# c <- c(10.56, 14.61, 11.88, 11.43, 9.73, 12.37, 3.82, 13.04, 13.28, 15.28)
# d <- c(7.74, 2.77, 10.29, 4.03, 10.23, 7.67, 7.93, 6.75, 5.60, 8.19)
# dati <- c(a,b,c,d)
# gruppi = factor(rep(c("uno","tre","bla","fgh"), each = 12))
# ta=anva(dati,gruppi)
```

`array2GRM` *array2GRM*

Description

From array id to maizesequence id (GRM).

Usage

```
array2GRM(id_array, tab_conf, tab_array = NULL, transcript = FALSE,
 score_length = NULL, file_gramene = "/tmp/X_gramene")
```

Arguments

`id_array`	Vector of array id.
`tab_conf`	File with ids table. ID_affy ID_maizeseq score_length pvalue Zm.1000.1.A1_at AC231180.2_FGT006 2.85233918128655 0 Zm.10015.1.A1_at GRMZM2G109680_T01 1.81548599670511 0 Zm.10015.1.A1_at GRMZM2G109680_T04 1.0164744645799 2e-175 Zm.10009.1.A1_at GRMZM2G133629_T01 1.50325097529259 0 Zm.10009.1.A1_at GRMZM2G133629_T02 1.43302990897269 0 Zm.10003.1.A1_at GRMZM2G402977_T01 1.75844594594595 0
`tab_array`	Table with array data, rownames same id_array, (default NULL).
`transcript`	If TRUE the function gives transcript, (default FALSE).
`score_length`	Minimum score blast in tab_conf, (default NULL).
`file_gramene`	Filename input for gramene, (default "/tmp/X_gramene").

Details

The tab_conf have be created by blast the array spots against maizesequence transcript database.
Only results with score greater than 1 and p value less 0.05 have been taken into account.
created tables:
/home/marco/maizeseq/affy_B73_5a_cdna_pul
/home/marco/maizeseq/affy_B73_5b_cdna_pul
/home/marco/maizeseq/maizearray_B73_5a_cdna_pul
/home/marco/maizeseq/maizearray_B73_5b_cdna_pul

Value

If tab_array equal NULL table with array id and GRM id,
else list table with array id and GRM id, table with GRM id and array data.

Author(s)

Marco Fracassetti

Examples

```
#esempio
# page=read.table("/home/marco/Microarray/opaque/prova/array/
# risultati_fin/analisi_fin_11_def/X_PAGE",stringsAsFactors=FALSE,header=TRUE)
# rownames(page)=page[,1]
# page=page[unique(union(which(abs(page[,"d1"])>0.58),
# which(abs(page[,"d2"])>0.58))),c("d1","d2")]
# page2=page[which(abs(page[,"d1"])>0.58),"d1"]
# id_array=rownames(page)
# tab_array=page2
# tab_conf="/home/marco/maizeseq/maizearray_B73_5b_cdna_pul"
# transcript=FALSE
# score_length=NULL
# tab=array2GRM(id_array,tab_conf,tab_array=page)
```

`coo2geni` *coo2geni*

Description

The function give the transcripts present in a genomic region.

Usage

```
coo2geni(coo, db, fout = "/tmp/out_coo")
```

Arguments

`coo`	Coordinates "chr:start-end".
`db`	Database to use.
`fout`	Filename of the transcript sequences file, (default "/tmp/out_coo").

Details

Working set database:
/home/marco/maizeseq/B73_5a_cdna/B73_5a_cdna
Filtered set database:
/home/marco/maizeseq/B73_5b_cdna/B73_5b_cdna

Value

Table with transcripts and coordinates.
File with transcript sequences

Author(s)

Marco Fracassetti

Examples

```
# coo="8:104,811,553-117,198,140" # pro1 +- 5 centimorgan
# coo="8:110811553-111198140"
# coo="8:110709532-111356822" #contiene anche tip4.2
# db="/home/marco/maizeseq/B73_5b_cdna/B73_5b_cdna"
# fout="/home/marco/Microarray/pro1_1/cdna_pro1_B73_5_b.fasta"
# db="/home/marco/maizeseq/B73_5a_cdna/B73_5a_cdna"
# fout="/tmp/proooo"
# tab=coo2geni(coo,db,fout) #####tabella con coordinate
# write.table(tab,quote=FALSE,sep="\t",file="/home/marco/Microarray/GRM_locus_O6_B73_5a")
```

---

csbl1 *csb1*

Description

Agglomerative hierarchical clustering based on the GO similarity matrix.
For each cluster a GO term enrichment analysis was performed,
for each of the three ontology class (BP, CC, MF),
the GO term with the maximum frequency among the spots, a q value < 0.05
and a priori probability value < 0.4 was chosen to represent the cluster.
First run csbl1 after csbl2 in two different R session.

Usage

```
csbl1(id, file_set, nome_GO, nome_ris, percorso,
 cutoff = 0.5, metric = "Lin", onto = NULL)
```

Arguments

id	Vector with the selected ids.
file_set	Filename of enrichment probability table.
nome_GO	File with ids and GO, one id per line.
nome_ris	Filename of output.
percorso	Directory where to put the output.
cutoff	Cutoff value for the clusters, (default 0.5).
metric	The similarity metric. One of "Resnik", "ResnikGraSM", "Lin", "LinGraSM", "JiangConrath", "JiangConrathGraSM", "Relevance", "Kappa", "Cosine", "WeightedJaccard", "CzekanowskiDice". (default "Lin")

onto	Ontologies used: "MF" molecular function "CC" cellular compartiment "BP" biological process NULL all (default NULL)

Details

GENE ONTOLOGY ANALYSIS:
1. obo_obsoleti find obsolete GO term in obo tot file.
2. rem_obo remove obsolete GO term for GO dataset.
3. redGO reduce GO dataset by GOslim.
4. prob_table calculate similarity and enrichment probability tables.
5. enrich enrichment analysis.
6. csbl1 clustering based on the GO similarity matrix
7. csbl2 clustering based on the GO similarity matrix

Value

Two file used by csbl2. For results see csbl2.

Author(s)

Marco Fracassetti

See Also

obo_obsoleti
rem_obo
redGO
prob_table
enrich
csbl1
csbl2

Examples

```
#id=readLines("/home/marco/biosegen_array/definitivo/setdiff_GRM")
#nome_ris="setdiff"
#percorso="/home/marco/biosegen_array/definitivo/csbl_pro/sim/"
#file_set="/home/marco/funzioni_R/agriGO/prob_tab_sim_slimGRM_def"
#nome_GO="/home/marco/funzioni_R/agriGO/slimGO_GRM_csbl_pul"
#csbl1(id,file_set,nome_GO,nome_ris,percorso,onto=NULL)
```

csbl2 *csbl2*

Description

Agglomerative hierarchical clustering based on the GO similarity matrix.
For each cluster a GO term enrichment analysis was performed,
for each of the three ontology class (BP, CC, MF),
the GO term with the maximum frequency among the spots, a q value < 0.05
and a priori probability value < 0.4 was chosen to represent the cluster.
First run csbl1 after csbl2 in two different R session.

Usage

```
csbl2(nome_ris, percorso, n_lim = 5, file_set, nome_GO)
```

Arguments

nome_ris	Filename of output.
percorso	Directory where to put the output.
n_lim	Minimun number of ids that belongs to a cluster, (default 5).
file_set	Filename of enrichment probability table.
nome_GO	File with ids and GO, one id per line.

Details

GENE ONTOLOGY ANALYSIS:
1. obo_obsoleti find obsolete GO term in obo tot file.
2. rem_obo remove obsolete GO term for GO dataset.
3. redGO reduce GO dataset by GOslim.
4. prob_table calculate similarity and enrichment probability tables.
5. enrich enrichment analysis.
6. csbl1 clustering based on the GO similarity matrix
7. csbl2 clustering based on the GO similarity matrix

Value

GO_pul_filename
Ids with GO terms, one id per line.

Filename_idclu
Ids divided into clusters, one cluster per line.

Filenames_tab_go
Table with GO terms of selected clusters, frequencies and q value.

Filenames_tab_DEF
Table with GO terms of selected clusters and percentual.

Directory enr
there are files of enrichment analysis done on clusters data.

Bargraph of selected clusters.

Bargraph of selected clusters with table.

Author(s)

Marco Fracassetti

See Also

obo_obsoleti
rem_obo
redGO
prob_table
enrich
csbl1
csbl2

Examples

```
#id=readLines("/home/marco/biosegen_array/definitivo/setdiff_GRM")
#nome_ris="setdiff"
#percorso="/home/marco/biosegen_array/definitivo/csbl_pro/sim/"
#file_set="/home/marco/funzioni_R/agriGO/prob_tab_sim_slimGRM_def"
#nome_GO="/home/marco/funzioni_R/agriGO/slimGO_GRM_csbl_pul"
#csbl2(nome_ris,percorso,n_lim=5,file_set,nome_GO)
```

---

enrich *enrich*

Description

The function does enrichment analysis through csbl.go packages.

Usage

```
enrich(id, nome_ris, percorso, file_set, nome_GO)
```

Arguments

id	Vector with the selected ids.
nome_ris	Filename of output.
percorso	Directory where to put the output.
file_set	Filename of enrichment probability table.
nome_GO	File with ids and GO, one id per line.

Details

GENE ONTOLOGY ANALYSIS:
1. obo_obsoleti find obsolete GO term in obo tot file.
2. rem_obo remove obsolete GO term for GO dataset.
3. redGO reduce GO dataset by GOslim.
4. prob_table calculate similarity and enrichment probability tables.
5. enrich enrichment analysis.
6. csbl1 clustering based on the GO similarity matrix
7. csbl2 clustering based on the GO similarity matrix

Value

For each ontology:

Table_freq (goid,desc,freq,proportion,p.value,priori,q.value)
table with all GO terms.
goid GO term.
desc description of GO term.
freq number of ids with GO term.
proportion proportion of ids with GO term.
p.value p value using Fisher's Exact Test (from csbl.go)
priori value from enrichment probability table.
q.value false discovery rate (from csbl.go)

Table_sel (goid,query,ref,desc,p_value,q_value,ids)
table with selected GO terms (query>ref, p_value < 0.05, q_value < 0.05).
goid GO term.
query proportion of ids with GO term.
ref proportion of ids with GO term in reference.
desc description of GO term.
p.value p value using Fisher's Exact Test (from csbl.go)
q.value false discovery rate (from csbl.go)
ids ids with the GO term.

Bargraph with selected GO terms.

Author(s)

Marco Fracassetti

See Also

obo_obsoleti
rem_obo
redGO
prob_table
enrich
csbl1
csbl2

Examples

```
# id=readLines("/home/marco/biosegen_array/definitivo/setdiff_GRM")
# nome_ris="setdiff"
# percorso="/home/marco/biosegen_array/definitivo/csbl_pro/"
# file_set="/home/marco/funzioni_R/agriGO/prob_tab_sim_slimGRM_def"
# nome_GO="/home/marco/funzioni_R/agriGO/slimGO_GRM_csbl_pul"
# enrich(id,nome_ris,percorso,file_set,nome_GO)
```

---

find_cit *find_cit*

Description

Search function for citations (text between parenthesis).

Usage

```
find_cit(file_in, file_out)
```

Arguments

file_in	Filename input, in plain text.
file_out	Filename output.

Value

File with citations.

Author(s)

Marco Fracassetti

Examples

```
# file_in="/home/marco/Dropbox/art/maydica/cit"
# file_out="/home/marco/Dropbox/art/maydica/cit_ris"
# find_cit(file_in,file_out)
```

geni2coo *geni2coo*

Description

The function gives the coordinates of GRM ids.

Usage

```
geni2coo(id, db, transcript = FALSE)
```

Arguments

id	GRM ids parent gene or transcript.
db	database to use.
transcript	If TRUE use transcript ids, (default FALSE).

Details

Working set database
/home/marco/maizeseq/B73_5a_cdna/B73_5a_cdna
Filtered set database
/home/marco/maizeseq/B73_5b_cdna/B73_5b_cdna

Value

Table with transcript ids, coordinates and parent genes.

Author(s)

Marco Fracassetti

Examples

```
#db="/home/marco/maizeseq/B73_5b_cdna/B73_5b_cdna"
#id=readLines("/home/marco/biosegen_array/definitivo/setdiff_GRM")
#ris=geni2coo(id,db)
```

GRM2array *GRM2array*

Description

From maizesequence id (GRM) to array id.

Usage

```
GRM2array(id_maize, tab_conf, score_length = NULL)
```

Arguments

id_maize	Vector of GRM id.
tab_conf	File with ids table. ID_affy ID_maizeseq score_length pvalue Zm.1000.1.A1_at AC231180.2_FGT006 2.85233918128655 0 Zm.10015.1.A1_at GRMZM2G109680_T01 1.81548599670511 0 Zm.10015.1.A1_at GRMZM2G109680_T04 1.0164744645799 2e-175 Zm.10009.1.A1_at GRMZM2G133629_T01 1.50325097529259 0 Zm.10009.1.A1_at GRMZM2G133629_T02 1.43302990897269 0 Zm.10003.1.A1_at GRMZM2G402977_T01 1.75844594594595 0
score_length	Minimum score blast in tab_conf, (default NULL).

Details

The tab_conf have be created by blast the array spots against maizesequence transcript database.
Only results with score greater than 1 and p value less 0.05 have been taken into account.
created tables:
/home/marco/maizeseq/affy_B73_5a_cdna_pul
/home/marco/maizeseq/affy_B73_5b_cdna_pul
/home/marco/maizeseq/maizearray_B73_5a_cdna_pul
/home/marco/maizeseq/maizearray_B73_5b_cdna_pul

Value

Table of GRM ids and array ids.

Author(s)

Marco Fracassetti

Examples

```
# tab_conf="/home/marco/maizeseq/maizearray_B73_5b_cdna_pul"
# id_maize=c("GRMZM2G370852","GRMZM2G404855")
# tab_ris=GRM2array(id_maize,tab_conf)
```

hypep *hypep*

Description

The function detects and displays various useful metrics about a protein sequence.
Require EMBOSS installed.

Usage

```
hypep(ids, db)
```

Arguments

ids	Vector of ids to use.
db	Protein database to use.

Value

List:
id ids analyzed.
MW molecular weight.
n_AA number of amino acid.
inc_bodies Improbability of expression in inclusion bodies.
tab_AA Table with percentual of amino acid.
Tiny (A+C+G+S+T)
Small(A+B+C+D+G+N+P+S+T+V)
Aliphatic (A+I+L+V)
Aromatic (F+H+W+Y)
Non-polar (A+C+F+G+I+L+M+P+V+W+Y)
Polar (D+E+H+K+N+Q+R+S+T+Z)
Charged (B+D+E+H+K+R+Z)
Basic (H+K+R)
Acidic (B+D+E+Z)

HY_OHM mean of OHM Hydropathy parameters (Sweet & Eisenberg) of all amino acids.
HY_Doo mean of Kyte & Doolittle hydropathy parameters of all amino acids.

Author(s)

Marco Fracassetti

Examples

```
# ids=c("GRMZM2G129133_T01", "GRMZM2G162450_T01", "GRMZM2G401342_T01", "GRMZM2G163015_T01
# "AC204530.4_FGT003", "AC190636.3_FGT005","GRMZM2G439784_T01", "GRMZM2G006234_T01",
# "GRMZM2G463968_T01", "GRMZM2G319307_T01")
# db="/home/marco/maizeseq/B73_5b_prot/B73_5b_prot"
# ris=hypep(ids,db)
```

`id2desc` *id2desc*

Description

the function gives description from ids.

Usage

```
id2desc(id, tipo, fout = "/tmp/desc_out")
```

Arguments

`id`	Vector of ids.
`tipo`	Type of ids: "GRM" maizesequence "array" maizearray "affy" affymetrix
`fout`	Filename output file, (default "/tmp/desc_out").

Details

Files needed:
array /home/marco/maizeseq/maizearray.v4.annotation.txt
affy /home/marco/maizeseq/Maize.na32.annot.csv GRM /home/marco/maizeseq/display.txt GRM /home/marco/maizeseq/xref.txt

Value

Vector of descriptions and file.

Author(s)

Marco Fracassetti

Examples

```
#id=readLines("/home/marco/Microarray/nuovo_16_02/id_O6locus_B73_5a")
#tipo="GRM"
#ris=id2desc(id,tipo)
```

obo_obsoleti *obo_obsoleti*

Description

Find function for obsolete GO term.

Usage

```
obo_obsoleti(obo_tot, nome)
```

Arguments

obo_tot	File obo to scan.
nome	Filename output.

Details

GENE ONTOLOGY ANALYSIS:
1. obo_obsoleti find obsolete GO term in obo tot file.
2. rem_obo remove obsolete GO term for GO dataset.
3. redGO reduce GO dataset by GOslim.
4. prob_table calculate similarity and enrichment probability tables.
5. enrich enrichment analysis.
6. csbl1 clustering based on the GO similarity matrix
7. csbl2 clustering based on the GO similarity matrix

Value

File with obsolete GO term and valid GO term.

Author(s)

Marco Fracassetti

See Also

obo_obsoleti
rem_obo
redGO
prob_table
enrich
csbl1
csbl2

Examples

```
#obo_tot="/home/marco/funzioni_R/csblGRM/gene_ontology_ext.obo"
#nome="/home/marco/funzioni_R/csblGRM/obsoleti"
#obo_obsoleti(obo_tot,nome)
```

`primer_s` *primer_s*

Description

Find primers in a sequence.

Usage

```
primer_s(seq, temp = 60, win_min = 19, win_max = 21, n_primer = 5, fin = TRUE)
```

Arguments

`seq`	String with the sequence.
`temp`	Melting temperature (default 60).
`win_min`	Minimum length primer, (default 19).
`win_max`	Maximum length primer, (default 21).
`n_primer`	Number of primers displayed.
`fin`	If TRUE the primer have to start and finish with C or G, (default TRUE).

Value

Print the primers with TM.

Author(s)

Marco Fracassetti

Examples

```
#seq_L="ATTTTGTAATCAATTAAGCGAGGCGATGAGCTCGTCCTCTCTCTTGCACGAGCC"
#primer_s(seq_L)
```

prob_table *prob_table*

Description

Create similarity and enrichment probability table for csbl.go.

Usage

```
prob_table(nome, tab_in, dir)
```

Arguments

nome	Filename output.
tab_in	Table with ids and GO. Zm.6307.1.A1_at GO:0009408 Zm.6307.1.A1_at GO:0009651 Zm.6309.1.A1_at GO:0019482 Zm.6309.1.A1_at GO:0046251 Zm.631.1.S1_at GO:0008094 Zm.6310.1.A1_at GO:0016787
dir	Directory where to place files.

Details

GENE ONTOLOGY ANALYSIS:
1. obo_obsoleti find obsolete GO term in obo tot file.
2. rem_obo remove obsolete GO term for GO dataset.
3. redGO reduce GO dataset by GOslim.
4. prob_table calculate similarity and enrichment probability tables.
5. enrich enrichment analysis.
6. csbl1 clustering based on the GO similarity matrix
7. csbl2 clustering based on the GO similarity matrix

Value

File with enrichment probability table.
File with enrichment probability table.
File with ids and GO, one id per line.

Author(s)

Marco Fracassetti

See Also

obo_obsoleti
rem_obo
redGO
prob_table
enrich
csbl1
csbl2

Examples

```
# tab=read.table("/home/marco/Microarray/csbl/Maize_spp_AgriGO",
#header=TRUE,stringsAsFactors=FALSE)
# tab=tab[which(tab[[1]]=="maizeAffy"),]
# tab_in=tab[,c(2,3)]
#
# nome="Affy_AgriGO"
#
# prob_table(nome,tab_in)
```

---

redGO *redGO*

Description

The GO dataset will be reduce by GOslim.

Usage

```
redGO(file_in, nome, file_obo, onto = "")
```

Arguments

file_in	Filename of table with ids and GO. Zm.6307.1.A1_at GO:0009408 Zm.6307.1.A1_at GO:0009651 Zm.6309.1.A1_at GO:0019482 Zm.6309.1.A1_at GO:0046251 Zm.631.1.S1_at GO:0008094 Zm.6310.1.A1_at GO:0016787
nome	Filename output.
file_obo	file obo with GOslim terms.
onto	Ontologies used: "MF" molecular function "CC" cellular compartiment "BP" biological process

"" all
(default "")

Details

GENE ONTOLOGY ANALYSIS:
1. obo_obsoleti find obsolete GO term in obo tot file.
2. rem_obo remove obsolete GO term for GO dataset.
3. redGO reduce GO dataset by GOslim.
4. prob_table calculate similarity and enrichment probability tables.
5. enrich enrichment analysis.
6. csbl1 clustering based on the GO similarity matrix
7. csbl2 clustering based on the GO similarity matrix

Value

File of table with ids and GOslim term.

Author(s)

Marco Fracassetti

See Also

`obo_obsoleti`
`rem_obo`
`redGO`
`prob_table`
`enrich`
`csbl1`
`csbl2`

Examples

```
# file_in="/home/marco/funzioni_R/csblGRM/GO_GRM_pul_BP"
# nome="/home/marco/funzioni_R/csblGRM/slimGO_GRM_pul_BP"
# file_obo="/home/marco/funzioni_R/csblGRM/map_slimplant"
# onto="BP"
#  redGO(file_in,nome,onto="BP")
```

rem_obo *rem_obo*

Description

Removing function for obsolete GO term.

Usage

```
rem_obo(file_in, obs, file_out)
```

Arguments

file_in	Filename of table with ids and GO. Zm.6307.1.A1_at GO:0009408 Zm.6307.1.A1_at GO:0009651 Zm.6309.1.A1_at GO:0019482 Zm.6309.1.A1_at GO:0046251 Zm.631.1.S1_at GO:0008094 Zm.6310.1.A1_at GO:0016787
obs	File whit obsolete GO term, (created with obo_obsoleti).
file_out	Filename output.

Details

GENE ONTOLOGY ANALYSIS:
1. obo_obsoleti find obsolete GO term in obo tot file.
2. rem_obo remove obsolete GO term for GO dataset.
3. redGO reduce GO dataset by GOslim.
4. prob_table calculate similarity and enrichment probability tables.
5. enrich enrichment analysis.
6. csbl1 clustering based on the GO similarity matrix
7. csbl2 clustering based on the GO similarity matrix

Value

File with non obsolete GO terms, ready for prob_table.

Author(s)

Marco Fracassetti

See Also

obo_obsoleti
rem_obo
redGO
prob_table
enrich
csbl1
csbl2

Examples

```
#file_in="/home/marco/funzioni_R/csbl_old/csblGRM/GO_GRM"
#obs="/home/marco/funzioni_R/csbl_old/csblGRM/obsoleti"
#file_out="/home/marco/funzioni_R/csbl_old/csblGRM/GO_GRM_pul"
#rem_obo(file_in, obs, file_out)
```

---

revcomp *revcomp*

Description

Reverse and complement of a sequence.

Usage

```
revcomp(seq, lowercase = FALSE)
```

Arguments

seq	A string of DNA.
lowercase	If TRUE output in lowercase, (default FALSE).

Value

A string with reverse and complement DNA.

Author(s)

Marco Fracassetti

Examples

```
#primer_R=revcomp("CTCTACATCATCGCGCAGTG")
```

simaffy *simaffy*

Description

The function analyzes affymetrix data.

Usage

```
simaffy(dir, conf, fold_change = 0.58, p_value = 0.05,
nome_covdesc = "covdesc.txt", heat_flag = FALSE, graph_flag = FALSE)
```

Arguments

dir	Directory where the CEL files are placed.
conf	List of the comparison that will be done.
fold_change	Fold change for significant spot, (default 0.58).
p_value	p value for significant spot, (default 0.05).
nome_covdesc	Name of the file covdesc, (default "covdesc.txt").
heat_flag	If TRUE heatmap will be done, (default FALSE).
graph_flag	If TRUE heatmap will be done, (default FALSE).

Details

Ue the package simpleaffy.

Value

Table of comparisons.
Pre-processing image if graph_flag = TRUE.
heatmap if heat_flag = TRUE

Author(s)

Marco Fracassetti

Examples

```
# dir="/home/marco/fusarium/array/"
# conf=list(c("fus","open"),c("fus","h2o"),c("h2o","open"))
# simaffy(dir,conf)
# fus_open=read.table("/home/marco/fusarium/array/ris_fus_open.txt")
# fus_open=rownames(fus_open)
# fus_h2o=read.table("/home/marco/fusarium/array/ris_fus_h2o.txt")
# fus_h2o=rownames(fus_h2o)
# h2o_open=read.table("/home/marco/fusarium/array/ris_h2o_open.txt")
# h2o_open=rownames(h2o_open)
# ris=setdiff(intersect(fus_open,fus_h2o),h2o_open)
```

`trascritti` *trascritti*

Description

The function creates a html page with colored transcripts.

Usage

```
trascritti(file_gen, file_html)
```

Arguments

`file_gen`	Filename input, downloaded from maizesequence in genbank format.
`file_html`	Filename output.

Value

A html page with each transcript with a different color.

Author(s)

Marco Fracassetti

References

http://www.maizesequence.org/

Examples

```
# file_gen="/home/marco/RT_PCR/ensembl.txt"
# file_html="/home/marco/RT_PCR/prova.html"
```

`t_mel` *t_mel*

Description

Calculate the melting temperature TM.
formula:
69.3 + (41 X nGC/nTOT) - (650/nTOT)

Usage

```
t_mel(primer)
```

Arguments

primer	String with primer.

Value

Print and return TM.

Author(s)

Marco Fracassetti

Examples

```
#t_mel("TCCTCCGCTTATTGATATGC")
```

---

wil_t *wil_t*

Description

The function calculate test of variance and post-hoc test, for samples with 2 levels of indipendent variable.

Usage

```
wil_t(dati, gruppi)
```

Arguments

dati	Vector with dependent variables.
gruppi	Vector with indipendent variables (type factor).

Details

For normality shapiro test.
T-student test, T-student test Welch approximation,
Wilcox test.

Value

Print in terminal the information about tests.
Return p value.

Author(s)

Marco Fracassetti

Examples

```
#  a <- c(13.47, 10.21, 15.10, 14.65, 9.03, 15.14,
#6.09, 3.43, 5.95, 10.72, 10.01, 8.17)
#  b <- c(4.02, 14.03, 3.09, 10.25, 6.25,
#1.00, 7.50, 10.18, 6.07, 2.03, 4.17, 7.28)
#  dati <- c(a,b)
#  gruppi = factor(rep(c("uno","tre"), each = 12))
#  p=wil_t(dati,gruppi)
```