FunctSNP – A practice session

Contents

Introduction

The following sample session introduces a new user to some of the functions within the *FunctSNP* R Package. Step by step instructions are given and it is expected that *FunctSNP* is executed whilst reading the instructions.

Input to most of the *FunctSNP* functions can be either by SNP ID (using the National Center of Biotechnology Information [NCBI] rs# cluster ID) or gene ID (using NCBI's gene ID). Also, some *FunctSNP* functions accept SNP locations as input. Therefore, the practice session is divided into 3 sections: (1) using SNP IDs as input; (2) using Gene IDs as input; and (3) using SNP locations as input.

Prerequisites prior to starting practice session

FunctSNP MUST be installed and loaded.

To list the packages currently installed:

```
library ()
```

To install *FunctSNP:*

```
install.packages("FunctSNP",dependencies=TRUE)
```

[Note that connection to the internet is required]

To load *FunctSNP:*

```
library (FunctSNP)
```

The practice session uses the *Bos taurus* database (btaSNP.db). To download btaSNP.db:

```
downloadDB (“bta”)
```

Practice Session

Start R as appropriate for your platform.

```
installedDBs ()
```

Lists the databases installed

```
setSpecies (“bta”)
```

Sets the default species code to “bta”. The data for each species is contained in its own database and the species code determines which database is accessed by the *FunctSNP* functions. The data for the *Bos taurus* species is contained in a local database called btaSNP.db and the species code “bta” is used to access this database via the *FunctSNP* functions. All the *FunctSNP* functions allow the default species code to be overridden by using a species option. For example, `getGeneID (snp_ids,species="gga")` and `getGO (snp_ids,"snp","oar").` The *species=”gga”* and *“oar”* is the species code that determines which database is accessed by the executed *FunctSNP* function. The default species code is not changed.

[1] Using SNP IDs as input

```
snp_ids <- read.table ("snps.txt",header=TRUE)
```

Reads a list of SNP IDs from a file called “snps.txt”. The SNP IDs are held in the snp_ids object. The content of “snps.txt” is shown in Appendix A. There are 100 SNP IDs that are true SNP IDS derived from NCBI’s dbSNP database; however, the SNP IDs chosen are from a fictitious whole genome association study. The *header=TRUE* option indicates that the first line is a heading and not data.

```
gene_ids <- getGenes (snp_ids)
```

Extracts the following gene information for the genes on which the SNPs reside: gene ID, gene symbol, chromosome number, chromosome arm location, start and end location, and gene name. The gene information is stored in the object “gene_ids”.

```
gene_ids
```

Displays the gene information to the screen

To display any object to the screen the object name is simply entered. For the rest of the practice session, the instruction for displaying objects will no longer be given and whether the object is displayed during the session is now at the discretion of the user.

```
snps_on_gene <- gene_ids[,c("SNP_ID")]
```

Creates a vector of SNP IDs located on genes. “SNP_ID” is the column header for the column from which to extract the data stored in the gene_ids object

```
hs_snps <- getSNPs (snps_on_gene)
```

Extracts the following SNP information for only the SNPs located on genes: chromosome number, chromosomal location (base pairs), protein coding status, exon/intron status, description of function, and score

```
row.idx <- which(hs_snps [,"Score"]> 20)
```

Retrieves the row index for each row with a value greater than 20 in the “Score” column

During the creation of the *FunctSNP database*, each SNP is given a score. There are two elements to the score that are summed: (1) a score according to the SNP location with respect to a gene. For example, the SNP is given a maximum score if it resides on an exon region with a non-synonymous effect; (2) a score according to how much supporting information is found. For example, if the SNP is linked to proteins, GO terms, KEGG pathways, QTL regions, and homologous genes, the score is incremented for each linkage. It is expected that SNPs that have a high score (greater than 20) are more likely to have a greater functional role.

```
hs_snps <- hs_snps [row.idx,c("SNP_ID")]
```

Creates a vector of SNP IDs extracted from the rows indentified by the indexes stored in “row.idx”. The “SNP_ID” is the column header for the column from which to extract the data stored in the hs_snps object.

```
valid_snps <- getSNPs (snp_ids)
```

Creates a data frame for only SNPs that exist in the database.

```
valid_snps <- valid_snps [,c("snp_ids")]
```

Creates a vector of all the valid SNP IDs

```
snps_not_on_gene <- setdiff(valid_snps, snps_on_gene)
```

Creates a vector of SNP IDs not located on genes. The setdiff R function determines the difference between two vectors, “snp_ids” and “snps_on_gene”.

```
nr_gene_ids <- getGenesByDist (snps_not_on_gene,dist=5000)
```

Generates a data frame of gene IDs for genes found 5000 base pairs (bp) up or down stream from the SNPs not located on genes (stored in the “snps_not_on_gene” object)

```
nr_snp_ids <- getSNPID(nr_gene_ids,loc.keep=FALSE,
 geneid.keep=FALSE)
```

Creates a data frame of SNP IDs located on the genes found within 5000 bp of the initial imported SNPs that are not located on genes. The default for getSNPID is to return SNP locations and gene IDs. The loc.Keep=FALSE and geneid.keep=FALSE means that only SNP IDs are returned.

```
hs_nr_snps <- getSNPs (nr_snp_ids)
```

Extracts the SNP information for SNPs located on the genes found within 5000 bp of the initial imported SNPs

```
row.idx <- which(hs_nr_snps [,"Score"]> 20)
```

Retrieves the row index for each row with a value greater than 20 in the “Score” column

```
hs_nr_snps <- hs_nr_snps [row.idx,c("SNP_ID")]
```

Creates a vector of SNP IDs extracted from the rows indentified by the indexes stored in “row.idx”. The “SNP_ID” is the column header for the column from which to extract the data stored in the hs_nr_snps object.

```
hs_snps <- union(hs_snps,hs_nr_snps)
```

Finds the union between the hs_snps and hs_nr_snps vectors

In effect, the union function concatenates the vector hs_snps (containing the high scoring SNPs on genes) and the vector hs_nr_snps (containing the high scoring SNPs found within 5000 bp of the initial imported SNPs that are not located on genes), and removes duplicate SNP IDs.

```
traits <- getTraits (hs_snps)
```

Extracts the trait description associated with a quantitative trait loci (QTL) region, and the start and end chromosomal location of the QTL region, for the high scoring SNPs that are located within these QTL regions

The source of the QTL data is from Animal Quantitative Trait Locus database (QTLdb) - (http://www.animalgenome.org/QTLdb/)

```
go_terms <- getGO (hs_snps)
```

Extracts the gene ID, gene ontology (GO) ID, term type, and term name for the genes on which the high scoring SNPs reside

For more detailed gene ontology information for the GO IDs listed, a GO ID e.g., GO:0001875, can be used as input under “Search the Gene Ontology Database” at http://www.geneontology.org/

```
pathways <- getKEGG (hs_snps)
```

Extracts the gene ID and KEGG pathway name (from Kyoto Encyclopaedia of Genes and Genomes) for the genes on which the high scoring SNPs reside

For more detailed information, a KEGG pathway can be viewed on the KEGG website at: http://www.genome.jp/kegg/pathway.html. Under “Select Prefix” replace “map” in the text box with the 3 letter organism abbreviation e.g., bta, or select the [organism] button. Then under “Enter Keywords” enter pathway identifier e.g. bta04620 or 04620

```
proteins <- getProteins (hs_snps)
```

Extracts protein ID, UniProt ID, and protein name linked to genes on which the high scoring SNPs reside

For more detailed protein information, a UniProt ID from the list e.g., Q4GZT3, can be used as input under UniProt homepage at http://www.uniprot.org/

```
getHomolo(taxon.list=TRUE)
```

Displays the taxonomy IDs and taxonomy names for all the available species that have homologous genes

```
taxons <- c (9606)
```

Creates an object containing 9606 that represents the taxonomy ID for *Homo sapiens*

```
homolo <-getHomolo (hs_snps,taxon.ids=taxons)
```

Extracts gene ID, gene set ID, gene symbol, protein ID, protein GI, taxonomy ID, and taxonomy name, of the human genes that are homologous to the cattle genes on which the high scoring SNP IDs reside. The *taxon.ids=taxons* option indicates to the function to expect as input, a vector of taxonomy IDs. If the *taxon.ids* option is not used, all taxonomy IDs for all the available species that have homologous genes are used as the default.

[2] Using Gene IDs as input

```
genes <- read.table ("genes.txt",header=TRUE)
```

Reads a list of gene IDs from a file called "genes.txt". The gene IDs are held in the gene_ids object. The content of "genes.txt" is shown in Appendix B. The 5 gene IDs are true gene IDS derived from NCBI's dbSNP database, however, the gene IDs chosen are from a fictitious candidate gene study. The *header=TRUE* option indicates that the first line is a heading and not data.

```
snps <- getSNPs (genes,"gene")
```

Extracts chromosome number, chromosomal location (bp), protein coding status, exon/intron status, function description, and score for ALL SNPs residing on the genes. The *"genes"* option indicates to the function that the input are gene IDS. If "genes" is not entered the function expects SNP IDs as the default input.

```
row.idx <- which(snps [,"Score"]> 20)
```

Retrieves the row index for each row with a value greater than 20 in the "Score" column

During the creation of the *FunctSNP database*, each SNP is given a score. There are two elements to the score that are summed: (1) a score according to the SNP location with respect to a gene. For example, the SNP is given a maximum score if it resides on an exon region with a non-synonymous effect, (2) a score according to how much supporting information is found. For example, if the SNP is linked to proteins, GO terms, KEGG pathways, QTL regions, and homologous genes, the score is incremented for each linkage. It is expected that SNPs that have a high score (greater than 20) are more likely to have a greater functional role.

```
snps_with_hs <- snps [row.idx,c("SNP_ID")]
```

Creates a vector of SNP IDs extracted from the rows indentified by the indexes stored in “row.idx”. The “SNP_ID” is the column header for the column from which to extract the data stored in the `snps_with_hs` object.

```
gene_ids <-getGeneID(snps_with_hs,loc.keep=FALSE,
snpid.keep=FALSE)
```

Extracts gene ID for each SNP ID stored in “snps_with_hs” (i.e. SNPs with a high score > 20). The default for getGeneID is to return SNP locations and SNP IDS. The loc.Keep=FALSE and snpid.keep=FALSE means that only gene IDs are returned.

```
traits <- getTraits (gene_ids,"gene")
```

Extracts the trait description associated with a quantitative trait loci (QTL) region, and the start and end chromosomal location of the QTL region, for all SNPs on the genes that are located within these QTL regions

The source of the QTL data is from Animal Quantitative Trait Locus database (QTLdb) - (http://www.animalgenome.org/QTLdb/)

```
go_terms <- getGO (gene_ids,"gene")
```

Extracts the gene ontology (GO) ID, term type, and term name for the genes

For more detailed gene ontology information for the GO IDs listed, a GO ID e.g., GO:0001875, can be used as input under “Search the Gene Ontology Database” at http://www.geneontology.org/

```
pathways <- getKEGG (gene_ids,"gene")
```

Extracts the KEGG pathway name (from Kyoto Encyclopaedia of Genes and Genomes) for the genes

For more detailed information, a KEGG pathway can be viewed on the KEGG website at: http://www.genome.jp/kegg/pathway.html. Under “Select Prefix” replace “map” in the text box with the 3 letter organism abbreviation e.g., bta, or select the [organism] button. Then under “Enter Keywords” enter pathway identifier e.g. bta04620 or 04620

```
proteins <- getProteins (gene_ids,"gene")
```

Extracts protein ID, UniProt ID, and protein name linked to genes

For more detailed protein information, a UniProt ID from the data frame e.g., Q4GZT3, can be used as input under UniProt homepage at http://www.uniprot.org/

```
getHomolo(taxon.list=TRUE)
```

Displays the taxonomy IDs and taxonomy names for all the available species that have homologous genes

```
taxons <- c (9606,10090)
```

Creates a vector containing 9606 and 10090 that represents the taxonomy IDs for *Homo sapiens* and *Mus musculus* respectively

```
homolo <-getHomolo (gene_ids,id.type="gene")
```

Extracts gene ID, gene set ID, gene symbol, protein ID, protein GI, taxonomy ID, and taxonomy name for all genes that are homologous to the cattle genes.

```
homolo <-getHomolo (gene_ids,id.type="gene",taxon.ids=taxons)
```

Extracts homologous information for the human and mouse genes that are homologous to the cattle genes

[3] Using SNP locations as input

```
locs <- read.table ("locations.txt",header=TRUE)
```

Reads a list of locations from a file called “locations.txt”. The locations are held in the locs object. The content of “locations.txt” is shown in Appendix C. The locations are true SNP locations derived from NCBI’s dbSNP database; however, the locations chosen are from a fictitious study. The *header=TRUE* option indicates that the first line is a heading and not data.

```
snp_loc <- getSNPID (locs,id.type="loc")
```

Extracts SNP ID, gene ID, and location for each SNP chromosomal base pair location in the “locs” object. The *id.type=“loc”* option indicates to the function that the input are locations. If *“id.type”* is not entered the function expects gene IDs as the default input.

```
snp_loc <- getSNPID (locs,id.type="loc",loc.keep=FALSE,
geneid.keep=FALSE)
```

Extracts only SNP ID for each SNP location

```
geneids_1 <- getGeneID  (locs,id.type="loc")
```

Extracts gene ID for each SNP location stored in “locs”. The *id.type=“loc”* option indicates to the function that the input are locations. If *“id.type”* is not entered the function expects SNP IDs as the default input.

```
nr_genes <- getNearGenes (locs,id.type="loc")
```

In relation to the SNP locations, extracts the nearest upstream and downstream gene ID, chromosomal distance in base pairs (bp) to downstream and upstream gene, and the gene ID on which the SNP locations reside

```
row.idx <- which(nr_genes[,"On_Gene_ID"]==0)
```

Retrieves the row index for each row with 0 in the “On_Gene_ID” column

```
ds_gene_id <- nr_genes[row.idx,c("DS_Gene_ID")]
us_gene_id <- nr_genes[row.idx,c("US_Gene_ID")]
```

Extracts the up and down stream gene IDs for the indexes stored in “row.idx”. In other words, extracts the up and down stream gene IDs in relation to SNP locations that are not located on genes.

```
geneids_2 <- union(ds_gene_id,us_gene_id)
```

Finds the union between the ds_gene_id and us_gene_id vectors (i.e. concatenate the two vectors and remove duplicate gene IDs).

```
snps_with_hs <- getHighScoreSNP(locs,"loc",dist=100000)
```

Extracts SNP ID with the highest score within 100,000 base pairs (bp) from each SNP SNP location stored in “locs”. The *“loc”* option indicates to the function that the input is locations. If *“loc”* is not entered the function expects SNP IDs as the default input.

During the creation of the *FunctSNP* database, each SNP is given a score. There are two elements to the score that are summed: (1) a score according to the SNP location with respect to a gene. For example, the SNP is given a maximum score if it resides on an exon region with a non-synonymous effect, (2) a score according to how much supporting information is found. For example, if the SNP is linked to proteins, GO terms, KEGG pathways, QTL regions, and homologous genes, the score is incremented for each linkage

```
snps_with_hs <- getSNPs (snps_with_hs)
```

Extracts chromosome number, chromosomal location (bp), protein coding status, exon/intron status, description of function, and score for the highest score SNPs

```
row.idx <- which(snps_with_hs[,"Score"]> 20)
```

Retrieves the row index for each row with a value greater than 20 in the “Score” column.

```
snps_with_hs <- snps_with_hs [row.idx,c("SNP_ID")]
```

Creates a vector of SNP IDs extracted from the rows indentified by the indexes stored in “row.idx”. The “SNP_ID” is the column header for the column from which to extract the data stored in the `snps_with_hs` object.

```
geneids_3 <- getGeneID  (snps_with_hs)
```

Extracts gene ID for each SNP ID stored in “snps_with_hs” (i.e. SNPs with a high score > 20).

```
geneids_1 <- geneids_1 [,c("Gene_ID")]
geneids_3 <- geneids_3 [,c("Gene_ID")]
```

Creates vectors of the gene IDs.

```
all_gene_ids <- union(geneids_1,geneids_3)
all_gene_ids <- union(all_gene_ids,geneids_2)
```

Finds the union between vectors (i.e. concatenate all vectors and removes duplicate gene IDs).

```
traits <- getTraits (all_gene_ids,"gene")
```

Extracts the trait description associated with a quantitative trait loci (QTL) region, and the start and end chromosomal location of the QTL region for all SNPs on the genes that are located within these QTL regions.

The source of the QTL data is from Animal Quantitative Trait Locus database (QTLdb) - (http://www.animalgenome.org/QTLdb/)

```
go_terms <- getGO (all_gene_ids,"gene")
```

Extracts the gene ontology (GO) ID, term type, and term name for the genes

For more detailed gene ontology information for the GO IDs listed, a GO ID e.g., GO:0001875, can be used as input under “Search the Gene Ontology Database” at http://www.geneontology.org/

```
pathways <- getKEGG (all_gene_ids,"gene")
```

Extracts the KEGG pathway name (from Kyoto Encyclopaedia of Genes and Genomes) for the genes

For more detailed information, a KEGG pathway can be viewed on the KEGG website at: http://www.genome.jp/kegg/pathway.html. Under “Select Prefix” replace “map” in the text box with the 3 letter organism abbreviation e.g., bta, or select the [organism] button. Then under “Enter Keywords” enter pathway identifier e.g. bta04620 or 04620

```
proteins <- getProteins (all_gene_ids,"gene")
```

Extracts protein ID, UniProt ID, and protein name linked to genes

For more detailed protein information, a UniProt ID from the data frame e.g., Q4GZT3, can be used as input under UniProt homepage at http://www.uniprot.org/

```
getHomolo(taxon.list=TRUE)
```

Displays the taxonomy IDs and taxonomy names for all the available species that have homologous genes

```
taxons <- c (9606,10090)
```

Creates a vector containing 9606 and 10090 that represents the taxonomy IDs for *Homo sapiens* and *Mus musculus* respectively

```
homolo <-getHomolo (all_gene_ids,id.type="gene")
```

Extracts gene ID, gene set ID, gene symbol, protein ID, protein GI, taxonomy ID, and taxonomy name for all genes that are homologous to the cattle genes.

```
homolo <-getHomolo (all_gene_ids,id.type="gene",taxon.ids=taxons)
```

Extracts homologous information for the human and mouse genes that are homologous to the cattle genes

APPENDIX A

The following is a list of NCBI SNP IDS from a fictitious whole genome association study:

```
snp_ids
8393057
8203068
17871291
56617464
57617461
55617448
56617446
43022989
42023990
55617462
42022893
42022794
42022996
55617544
55617841
55617241
17872295
13871296
42705722
44704823
42704724
42705725
44703726
43702727
29022435
29022936
29022737
17871897
17871298
17871799
17971308
17871409
17971310
8493069
8593070
8693071
8793072
17870281
17870282
17870283
42391650
17860284
17870285
17880286
17870287
8199041
8183042
8199043
```

```
8183044
8199045
8143046
17871256
17872257
19871258
17891259
18871260
55617457
55697456
55687455
58617451
55677449
19871261
17881263
8293047
8293049
8123050
17870277
18870279
19870280
8793048
17870289
18870290
19870291
10870292
11870293
8193051
8183055
8173058
8163059
8153067
8293056
8143060
8153061
8173065
17871254
18871255
19871265
11871266
12871268
13871269
42392651
52390651
42670913
45670914
42680916
43670917
44670919
41911811
42390651
43702346
```

APPENDIX B

The following is a list of NCBI gene IDS from a fictitious candidate gene study:

gene_ids
508343
512719
530393
509142
614025

APPENDIX C

The following is a list of SNP locations:

snp_locations
265772
37439119
10352855
31091453
735588
35368057
53951943
43643807
112435386
58181139
18320443

APPENDIX D

The following is a list of the R commands for the entire practice session:

```
library (FunctSNP)
downloadDB (“bta”)
installedDBs ()
setSpecies("bta")

#**** SNP IDs as INPUT *****

snp_ids <- read.table ("snps.txt",header=TRUE)
gene_ids <- getGenes (snp_ids)
snps_on_gene <- gene_ids[,c("SNP_ID")]
hs_snps <- getSNPs (snps_on_gene)
row.idx <- which(hs_snps [,"Score"]> 20)
hs_snps <- hs_snps [row.idx,c("SNP_ID")]
valid_snps <- getSNPs (snp_ids)
valid_snps <- valid_snps[,c("SNP_ID")]
snps_not_on_gene <- setdiff(valid_snps, snps_on_gene)
nr_gene_ids <- getGenesByDist (snps_not_on_gene,dist=5000)
nr_snp_ids <-getSNPID(nr_gene_ids,loc.keep=FALSE,geneid.keep=FALSE)
hs_nr_snps <- getSNPs (nr_snp_ids)
```

```
row.idx <- which(hs_nr_snps [,"Score"]> 20)
hs_nr_snps <- hs_nr_snps [row.idx,c("SNP_ID")]
hs_snps <- union(hs_snps,hs_nr_snps)
traits <- getTraits (hs_snps)
go_terms <- getGO (hs_snps)
pathways <- getKEGG (hs_snps)
proteins <- getProteins (hs_snps)
getHomolo(taxon.list=TRUE)
taxons <- c (9606)
homolo <-getHomolo (hs_snps,taxon.ids=taxons)



#**** Gene IDs as INPUT *****

genes <- read.table ("genes.txt",header=TRUE)
snps <- getSNPs (genes,"gene")
row.idx <- which(snps [,"Score"]> 20)
snps_with_hs <- snps [row.idx,c("SNP_ID")]
gene_ids <- getGeneID(snps_with_hs,loc.keep=FALSE,snpid.keep=FALSE)
traits <- getTraits (gene_ids,"gene")
go_terms <- getGO (gene_ids,"gene")
pathways <- getKEGG (gene_ids,"gene")
proteins <- getProteins (gene_ids,"gene")
getHomolo(taxon.list=TRUE)
taxons <- c (9606,10090)
homolo <-getHomolo (gene_ids,id.type="gene")
homolo <-getHomolo (gene_ids,id.type="gene",taxon.ids=taxons)

#**** SNP Locations as INPUT *****

locs <- read.table ("locations.txt",header=TRUE)
snp_loc <- getSNPID (locs,id.type="loc”)
snp_loc <-getSNPID(locs,id.type="loc",loc.keep=FALSE,
geneid.keep=FALSE)
geneids_1 <- getGeneID (locs,id.type="loc")
nr_genes <- getNearGenes (locs,id.type="loc")
row.idx <- which (nr_genes[,"On_Gene_ID"]==0)
ds_gene_id <- nr_genes[row.idx,c("DS_Gene_ID")]
us_gene_id <- nr_genes[row.idx,c("US_Gene_ID")]
geneids_2 <- union (ds_gene_id,us_gene_id)
snps_with_hs <- getHighScoreSNP(locs,"loc",dist=100000)
snps_with_hs <- getSNPs (snps_with_hs)
row.idx <- which (snps_with_hs[,"Score"]> 20)
snps_with_hs <- snps_with_hs [row.idx,c("SNP_ID")]
geneids_3 <- getGeneID (snps_with_hs)
geneids_1 <- geneids_1 [,c("Gene_ID")]
geneids_3 <- geneids_3 [,c("Gene_ID")]
all_gene_ids <- union(geneids_1,geneids_3)
all_gene_ids <- union(all_gene_ids,geneids_2)
traits <- getTraits (all_gene_ids,"gene")
go_terms <- getGO (all_gene_ids,"gene")
pathways <- getKEGG (all_gene_ids,"gene")
proteins <- getProteins (all_gene_ids,"gene")
getHomolo(taxon.list=TRUE)
taxons <- c (9606,10090)
homolo <-getHomolo (all_gene_ids,id.type="gene")
homolo <-getHomolo (all_gene_ids,id.type="gene",taxon.ids=taxons)
```