Using the FunciSNP package 'Functional Identification of SNPs with Phenotype by Coincidence with Chromatin Biofeatures'

Simon G. Coetzee[‡,*], Suhn K. Rhie[‡], Benjamin P. Berman[‡], Gerhard A. Coetzee[‡] and Houtan Noushmehr[‡,†]

March 7, 2012

[‡]Norris Cancer Center
Keck School of Medicine
University of Southern California
Los Angeles, CA, USA

Contents

[*]scoetzee NEAR gmail POINT com
[†]houtana NEAR gmail POINT com

1 Introduction

FunciSNP assist in identifying putative functional SNP in LD to previously annotated GWAS SNPs (tagSNP). Extracting information from the 1000 genomes database (1000GP) by relative genomic position of GWAS tagSNP currated for a particular trait or disease, FunciSNP aims to integrate the two information with sequence information provided by peaks identified from high-throughput sequencing. FunciSNP assumes user will provide peaks identified using any available ChIP peak algorithm, such as FindPeaks, HOMER, or SICER. *FunciSNP* will currate all 1000GP SNPs which are in linkage disequilibrium (LD) to a known disease associated tagSNP and more importantly determine if the 1000GP SNP in LD to the tagSNP overlaps a genomic biological feature.

Correlated SNPs are directly imported from the current public release of the 1000 genomes database. 1000 genomes ftp servers available for the 1000 genomes public data:

- National Center for Biotechnology Information (NCBI)[1]
- European Bioinformatics Institute (EBI)[2]

Correlated SNPs in LD to a tagSNP and overlapping genomic biological features are known as putative functional SNPs.

This vignette provides a 'HOW-TO' guide in setting up and running *FunciSNP* on your machine. FunciSNP was developed with the idea that a user will have uninterupted high-speed internet access as well as a desktop machine with more than 4 multiple cores. If

[1] `ftp://ftp-trace.ncbi.nih.gov/1000genomes/`
[2] `ftp://ftp.1000genomes.ebi.ac.uk/vol1/`

user is using a windows machine, multiple cores options will not work and thus total time to complete initial FunciSNP analysis will take longer than expected. Be sure you have uninterupted computing power when using a windows machine. If using a linux machine, please use 'screen' (see 'man screen' for more information).

1.1 Benchmark

Using a 64bit Linux machine running 11.04 Ubuntu OS with 24G RAM and 8 cores connected to a academic high-speed internet port, the amount of time to complete 99 tagSNP across 20 different biofeatures took less than 30 min to complete. We anticipate about 2 hours to complete the same analysis using one core.

1.2 Genome-Wide Association Studies SNP (GWAS SNP)

Genome-wide association studies (GWASs) have yielded numerous single nucleotide polymorphisms (SNPs) associated with many phenotypes. In some cases tens of SNPs, called tagSNPs, mark many loci of single complex diseases such as prostate (> 50 loci), breast (> 20 loci), ovarian (>10 loci), colorectal (>20 loci) and brain cancer (>5 loci) for which functionality remains unknown. Since most of the tagSNPs (>80%) are found in non-protein coding regions, finding direct information on the functional and/or causal variant has been an important limitation of GWAS data interpretation.

1.3 1000 genomes project (1000GP)

The 1000 genomes project recently released a catalog of most human genomic variants (minor allele frequency of >0.1%) across many different ethnic populations. Initially, the 1000 genomes project goal was to sequence up to 1000 individuals, but has since sequenced more than 2000 individuals, thereby increasing our current knowledge of known genomic variations which currently sits at just over 50 million SNPs genome wide (approx. 2% of the entire genome and on average 1 SNP every 60 base pairs)

1.4 Genomic features (Biofeatures)

With the advent of advanced sequencing technologies (next-generation sequencing, NGS), genomic regulatory areas in non-coding regions have been well characterized and annotated. Coupled with chromatin immuno-precipitation for a protein (e.g. transcription factor of histone) of interest, also known as ChIPseq, the technology have provided us with a unique view of the genomic landscape, thereby providing a wealth of new knowledge for genomics research. Work by large consortia groups such as the Encyclopedia of DNA Elements (ENCODE), the Roadmap Epigenomics Mapping Consortium and The Cancer Genome Atlas (TCGA), have made publicly available a growing catalog of many different histone marks, transcription factors and genome-wide sequencing datasets for a variety of different diseases

and cell lines, including well characterized cancer cell lines such as MCF7 (breast cancer), HCT116 (colon cancer), U87 (brain cancer) and LNCaP (prostate cancer).

2 Installing and Loading FunciSNP

Currently, there are two options to obtain a copy of *FunciSNP*:

- Download current source code from Coetzee's lab[3]
- Download and install from Bioconductor[4]

If you download the source code from either method above, you can install *FunciSNP* by following the instructions described in R CRAN. By installing *FunciSNP* from source, the package assumes you have all the required libraries installed.

- Rsamtools (>= 1.6.1)
- rtracklayer(>= 1.14.1)
- GGtools (>= 4.0.0)
- methods
- ChIPpeakAnno (>= 2.2.0)
- GenomicRanges
- TxDb.Hsapiens.UCSC.hg19.knownGene
- VariantAnnotation
- plyr
- org.Hs.eg.db
- snpStats

The following loads the *FunciSNP* library in R.

```
> options(width=80);
> library(FunciSNP);
> package.version("FunciSNP");

[1] "0.1.8"
```

[3] http://coetzeeseq.usc.edu/publication/Coetzee_SG_et_al_2012/
[4] http://www.bioconductor.org

3 Running getFSNPs to identify putative functional SNPs

Before running *getFSNPs*, two input files are required. A list of tagSNPs and a folder with all available biological features (peak files in BED format).

3.1 Create a GWAS SNP file

GWAS SNPs (tagSNP) should be listed in a tab or whitespace separated file. Three columns are required for each tagSNP:

- Position (chrom:position)
- rsID (rsXXXXXXXX)
- population (EUR, AFR, AMR, ASN, or ALL)

'Positon' should be the exact postion for each rsID as reported by human genome build hg19 (chrom:postion). 'rsID' should contain a unique rsID as determined by the 1000 genomes database (1000GP)[5] for each identified 'tagSNP'. Population should be a three letter code to determine original ethnic population for which the associated 'tagSNP' was identified. The three letter code should be either European (EUR), Asian (ASN), African (AFR), American (AMR), or All (ALL). List each tagSNP per ethnic population. If similar rsID was identified in multiple ethnic population, list each duplicate tagSNP separately with the appropriate ethnic pouplation.

Several GWAS SNPs significantly associated with Glioblastoma multiforme (GBM)[6] were collected for this example. GBM is a brain cancer with median survival at less than 12 months, making this form of cancer one of the most aggressive of all cancer types. Currently, there is no known function of any of these associated tagSNPs. In this example, GBM includes lower grade glioma, therefore we use the 'glioma' to label all objects.

```
> ## Full path to the example GWAS SNP regions file for Glioblastoma
> #  (collected from SNPedia on Jan 2012)
> glioma.snp <- file.path(system.file('extdata', package='FunciSNP'),
+ dir(system.file('extdata',package='FunciSNP'), pattern='.snp$'));
> gsnp <- read.delim(file=glioma.snp,sep=" ",header=FALSE);
> gsnp;
```

```
           V1        V2  V3
1 11:118477367  rs498872 EUR
2    5:1286516 rs2736100 ASN
3   9:22068652 rs4977756 EUR
4  20:62309839 rs6010620 EUR
```

[5] Be sure the rsID is located in this browser: `http://browser.1000genomes.org/`

[6] See `http://www.snpedia.com/index.php/Glioma`

Now, `glioma.snp` contains the full path to the GWAS tagSNP.

3.2 Biofeatures in BED format

Each biofeature used to identify correlated SNP should be in standard BED format[7]. Each biofeature should be stored in one folder and should have file extension '*.bed'.

Here is an example of three different biofeatures used for this glioma example. NRSF and PolII (both transcription factors) where extracted from a recent release of ENCODE, as well as promoters of approximately 38,000 gene transcription start sites (TSS). Promoters are identified as +1000 to -100 base pair of each annotated TSS.

```
> ## Full path to the example biological features BED files
> #  derived from the ENCODE project for Glioblastoma U-87 cell lines.
> glioma.bio <- system.file('extdata',package='FunciSNP');
> as.matrix(list.files(glioma.bio, pattern='.bed$'));

     [,1]
[1,] "CTCF_only.bed"
[2,] "EncodeDnaseI_only.bed"
[3,] "EncodeDnaseI_withCTCF.bed"
[4,] "knownGene.Promoters.bed"
[5,] "TFBS_Nrsf_U87.bed"
[6,] "TFBS_Pol2_U87.bed"

> nrsf.filename <- list.files(glioma.bio, pattern='.bed$')[2];
> pol2.filename <- list.files(glioma.bio, pattern='.bed$')[3];
> prom.filename <- list.files(glioma.bio, pattern='.bed$')[1];
> Nrsf <- read.delim(file=paste(glioma.bio, nrsf.filename,sep="/"), sep="\t",
+ header=FALSE);
> PolII <- read.delim(file=paste(glioma.bio, pol2.filename,sep="/"), sep="\t",
+ header=FALSE);
> Promoters <- read.delim(file=paste(glioma.bio, prom.filename,sep="/"), sep="\t",
+ header=FALSE);
> dim(Nrsf);

[1] 175508      3

> dim(PolII);

[1] 4978    3

> dim(Promoters);
```

[7] See UCSC FAQ: `http://genome.ucsc.edu/FAQ/FAQformat`

```
[1] 9917    3

> ## Example of what the BED format looks like:
> head(Nrsf);

    V1     V2     V3
1 chr1 564460 564670
2 chr1 565260 565710
3 chr1 565840 566070
4 chr1 566700 567310
5 chr1 567440 567910
6 chr1 568060 568710
```

As an example, `Nrsf` was created to illustrate the format needed for each biofeatures. To run getFSNPs, only the path to the folder to each biofeature is required (`glioma.bio`).

3.3 getFSNPs analysis using two inputs

To run the example data could take more than 5 minutes, thus the R code is commented out for this tutorial. If you are interested in running the glioma example from scratch, please uncomment the following and rerun in your R session. NOTE: The main method to run FunciSNP is *getFSNPs*.

```
> ## FunciSNP analysis, extracts correlated SNPs from the
> ## 1000 genomes db ("ncbi" or "ebi") and finds overlaps between
> ## correlated SNP and biological features and then
> ## calculates LD (Rsquare, Dprime, distance, p-value).
> ## Depending on number of CPUs and internet connection, this step may take
> ## some time. Please consider using a unix machine to access multiple cores.
>
> # glioma <- getFSNPs(snp.regions.file=glioma.snp, bio.features.loc = glioma.bio,
> # bio.features.TSS=FALSE);
```

As an alternative, `glioma` was pre-run and stored in the package as an *R* object. To call this data object, simily run the following commands.

```
> data(glioma);
> class(glioma);

[1] "TSList"
attr(,"package")
[1] "FunciSNP"
```

Now, `glioma` contains the R data structure that holds all the results for this particular analysis. Each tagSNP is stored as a slot which contains associated correlated SNP and overlapping biofeature. It also contains a number of different annotations (see below for more details). To see a brief summary of the results (*summary*), type the following commands:

```
> glioma;

TagSNP List with  4  Tag SNPs and
 1205 nearby,  potentially correlated SNPs, that overlap at least one biofeature
$`R squared: 0.1`
            Total R.sq>=0.1 Percent
tagSNPs         4         4   100.0
1K SNPs      1205        88     7.3
Biofeatures     4         4   100.0

$`R squared: 0.5`
            Total R.sq>=0.5 Percent
tagSNPs         4         3   75.00
1K SNPs      1205        48    3.98
Biofeatures     4         3   75.00

$`R squared: 0.9`
            Total R.sq>=0.9 Percent
tagSNPs         4         1   25.00
1K SNPs      1205        13    1.08
Biofeatures     4         2   50.00
```

As you can quickly observe from the above analysis, using 4 tagSNPs position and 3 different biological features (ChIPseq for 'NRSF', 'PolII', promoters of approx. 38,000 genes) as two types of input, FunciSNP identified 778 1000GP SNPs that overlap at least one biofeature. Each 1000GP SNP contains an Rsquare value to the associated tagSNP. As a result, the first output (`glioma`), summarizes the analysis subsetted in three different Rsquare values (0.1, 0.5 and 0.9). If we consider Rsquare cutoff at 0.9 (Rsquare $\geq$ 0.9), 13 1000GP SNPs overlapping at least one biofeature. This value represents 1.67% of the total (778). In addition, at this Rsquare cutoff, 2 biological features are represented among the 13 1000GP SNPs.

```
> summary(glioma);

TagSNP List with  4  Tag SNPs and
 1205 nearby,  potentially correlated SNPs, that overlap at least one biofeature
Number of potentially correlated SNPs
overlapping at least x biofeatures, per Tag SNP at a specified R squared
$`R squared: 0.1 in 4 Tag SNPs with a total of `
```

```
                    bio.1 bio.2 bio.3
rs2736100               1     0     0
rs4977756              11     0     0
rs498872               18     2     0
rs6010620              58    10     4
TOTAL # 1000GP SNPs    88    12     4

$`R squared: 0.5 in 3 Tag SNPs with a total of `
                    bio.1 bio.2 bio.3
rs4977756               5     0     0
rs498872                3     0     0
rs6010620              40     6     3
TOTAL # 1000GP SNPs    48     6     3

$`R squared: 0.9 in 1 Tag SNPs with a total of `
                    bio.1
rs6010620              13
TOTAL # 1000GP SNPs    13
```

Running *summary* however will output a slightly different report yet just as informative. At three different Rsquare cutoffs (0.1, 0.5, 0.9), the summary output illustrates the tagSNP with the total number of 1000GP SNPs overlapping a total number of biofeatures. For example, at Rsquare ≥ 0.5, tagSNP 'rs6010620' is assocated with 40 different 1000GP SNPs which overlap at least one biofeature, and 6 of them overlap at least two biofeatures.

Each newly identified 1000GP SNP is now defined as putative functional SNP since they are in LD to an associated tagSNP and they overlap at least one interesting biological feature. Thus, each 1000GP SNP can now be defined as '**Func-y-SNP**' or 'putative functional SNP.'

4 Annotating newly identified putative functional SNPs

All known genomic features (exon, intron, 5'UTR, 3'UTR, promoter, lincRNA or in gene desert (intergentic)) are used to annotate each newly identified Func-y-SNP as described above. Information stored in this `glioma.anno` is used for all summary plots, table, and to output results in BED format (see following sections for more details). The following step will output the data.frame.

```
> glioma.anno <- FunciSNPAnnotateSummary(glioma);
> class(glioma.anno);

[1] "data.frame"

> gl.anno <- glioma.anno;
> ## remove rownames for this example section.
```

```
> rownames(gl.anno) <- c(1:length(rownames(gl.anno)))
> dim(gl.anno);

[1] 1371   28

> names(gl.anno);

 [1] "chromosome"                          "bio.feature.start"
 [3] "bio.feature.end"                     "bio.feature"
 [5] "corr.snp.id"                         "corr.snp.position"
 [7] "tag.snp.id"                          "tag.snp.position"
 [9] "D.prime"                             "R.squared"
[11] "p.value"                             "distance.from.tag"
[13] "population.count"                    "population"
[15] "nearest.lincRNA.ID"                  "nearest.lincRNA.distancetoFeature"
[17] "nearest.lincRNA.coverage"            "nearest.TSS.GeneSymbol"
[19] "nearest.TSS.refseq"                  "nearest.TSS.ensembl"
[21] "nearest.TSS.coverage"                "nearest.TSS.distancetoFeature"
[23] "Promoter"                            "utr5"
[25] "Exon"                                "Intron"
[27] "utr3"                                "Intergenic"

> head(gl.anno[, c(1:18,20:28)]);

  chromosome bio.feature.start bio.feature.end         bio.feature  corr.snp.id
1          5           1200710         1201809 knownGene.Promoters chr5:1200720
2          5           1200710         1201809 knownGene.Promoters chr5:1200766
3          5           1200710         1201809 knownGene.Promoters chr5:1200817
4          5           1200710         1201809 knownGene.Promoters chr5:1200946
5          5           1200710         1201809 knownGene.Promoters chr5:1200976
6          5           1200710         1201809 knownGene.Promoters chr5:1201033
  corr.snp.position tag.snp.id tag.snp.position   D.prime    R.squared p.value
1           1200720  rs2736100          1286516        NA           NA       1
2           1200766  rs2736100          1286516        NA           NA       1
3           1200817  rs2736100          1286516        NA           NA       1
4           1200946  rs2736100          1286516        NA           NA       1
5           1200976  rs2736100          1286516 1.0000000 0.0022585199       1
6           1201033  rs2736100          1286516 0.1795671 0.0004069606       1
  distance.from.tag population.count population nearest.lincRNA.ID
1            -85796              286        ASN     TCONS_00010241
2            -85750              286        ASN     TCONS_00010241
3            -85699              286        ASN     TCONS_00010241
4            -85570              286        ASN     TCONS_00010241
5            -85540              286        ASN     TCONS_00010241
```

```
6            -85483              286        ASN     TCONS_00010241
  nearest.lincRNA.distancetoFeature nearest.lincRNA.coverage
1                            -39302                upstream
2                            -39348                upstream
3                            -39399                upstream
4                            -39528                upstream
5                            -39558                upstream
6                            -39615                upstream
  nearest.TSS.GeneSymbol nearest.TSS.ensembl nearest.TSS.coverage
1                SLC6A19     ENSG00000174358             upstream
2                SLC6A19     ENSG00000174358             upstream
3                SLC6A19     ENSG00000174358             upstream
4                SLC6A19     ENSG00000174358             upstream
5                SLC6A19     ENSG00000174358             upstream
6                SLC6A19     ENSG00000174358             upstream
  nearest.TSS.distancetoFeature Promoter utr5 Exon Intron utr3 Intergenic
1                          -990      YES   NO   NO     NO   NO         NO
2                          -944      YES   NO   NO     NO   NO         NO
3                          -893      YES   NO   NO     NO   NO         NO
4                          -764      YES   NO   NO     NO   NO         NO
5                          -734      YES   NO   NO     NO   NO         NO
6                          -677      YES   NO   NO     NO   NO         NO
```

```
> summary(gl.anno[, c(1:18,20:28)]);
```

```
  chromosome        bio.feature.start   bio.feature.end
 Length:1371        Min.   :  1200710   Min.   :  1201809
 Class :character   1st Qu.: 22079881   1st Qu.: 22080370
 Mode  :character   Median : 62315624   Median : 62324005
                    Mean   : 56394593   Mean   : 56397247
                    3rd Qu.: 62369936   3rd Qu.: 62371110
                    Max.   :118574381   Max.   :118574590

             bio.feature           corr.snp.id   corr.snp.position
 Encode_DnaseCluster:509   chr11:118442863:   3   Min.   :  1200720
 knownGene.Promoters:372   chr11:118443036:   3   1st Qu.: 22080066
 TFBS_Nrsf_U87      : 22   chr11:118443046:   3   Median : 62318595
 TFBS_Pol2_U87      :468   chr20:62289690 :   3   Mean   : 56395952
                           chr20:62289873 :   3   3rd Qu.: 62370349
                           chr20:62290057 :   3   Max.   :118574557
                           (Other)        :1353
    tag.snp.id  tag.snp.position       D.prime            R.squared
 rs2736100:232   Min.   :  1286516   Min.   :7.835e-04   Min.   :9.520e-08
```

```
rs4977756:203   1st Qu.: 22068652   1st Qu.:9.204e-01   1st Qu.:1.127e-03
rs498872 :253   Median : 62309839   Median :1.000e+00   Median :4.501e-03
rs6010620:683   Mean   : 56390082   Mean   :8.717e-01   Mean   :1.026e-01
                3rd Qu.: 62309839   3rd Qu.:1.000e+00   3rd Qu.:2.632e-02
                Max.   :118477367   Max.   :1.000e+00   Max.   :9.776e-01
                                    NA's   :7.670e+02   NA's   :7.670e+02
   p.value           distance.from.tag population.count population
Min.   :9.021e-175   Min.   :-100000   Min.   :286.0    ASN: 232
1st Qu.: 1.000e+00   1st Qu.: -26778   1st Qu.:379.0    EUR:1139
Median : 1.000e+00   Median :  14153   Median :379.0
Mean   : 8.084e-01   Mean   :   5870   Mean   :363.3
3rd Qu.: 1.000e+00   3rd Qu.:  37188   3rd Qu.:379.0
Max.   : 1.000e+00   Max.   :  97190   Max.   :379.0

     nearest.lincRNA.ID nearest.lincRNA.distancetoFeature
TCONS_00010241:232      Min.   :-266862
TCONS_00015797:203      1st Qu.:-133416
TCONS_00020001:253      Median :  55916
TCONS_00027984: 44      Mean   :  -1509
TCONS_00028269:639      3rd Qu.:  77246
                        Max.   : 246019

nearest.lincRNA.coverage    nearest.TSS.GeneSymbol      nearest.TSS.ensembl
downstream:830           TNFRSF6B      :327        ENSG00000243509:327
inside    : 12           PHLDB1        :134        ENSG00000019144:134
upstream  :529           ZGPAT         : 68        ENSG00000244977:116
                         SLC6A18       : 61        ENSG00000215221: 85
                         RTEL1;TNFRSF6B: 43        ENSG00000197114: 68
                         (Other)       :304        ENSG00000164363: 61
                         NA's          :434        (Other)        :580
nearest.TSS.coverage nearest.TSS.distancetoFeature Promoter    utr5
downstream:350       Min.   :-20118                NO :1177   NO :1318
inside    :437       1st Qu.: -2049                YES: 194   YES:  53
upstream  :584       Median :   641
                     Mean   :  7950
                     3rd Qu.: 10782
                     Max.   : 99932

 Exon      Intron     utr3      Intergenic
NO :1272   NO :575   NO :1199   NO :1209
YES:  99   YES:796   YES: 172   YES: 162
```

```
> rm(gl.anno);
```

As you can see, each tagSNP ('tag.snp.id') is associated with an identifiable Func-y-SNP ('corr.snp.id') and each are associated with a biological feature ('bio.feature'). Additional columns are included which assist in summarizing the final results.

Now, if you prefer, you can use several functions to help summarize and plot the final analysis or you can use your own set of scripts to further summarize the results. Either case, the final results are stored in `glioma.anno`.

5 Summarize FunciSNP results

The following sections describe methods to summarize and plot the newly identified Func-y-SNPs.

5.1 Summary table used to describe newly identified Func-y-SNPs

Using a specified Rsquare value (0-1) to subset the data, a table is generated which summarizes the total number of Func-y-SNPs, associated tagSNPs, and number of overlapping biofeatures. This will provide user a first look at the total number of available Func-y-SNP at a particular Rsquare cutoff.

The output is very similar to the output generated by calling `glioma`. But instead of getting a summary report three distinct Rsquare cutoffs, you can now specify the Rsquare cutoffs. In this case, we used rsq = 0.44 (to get a more objective rsq value, see figure 1 on page 16.

```
> FunciSNPtable(glioma.anno, rsq=0.44);
```

```
             Total R.sq>=0.44 Percent
tagSNPs          4          4   100.0
1K SNPs       1205         53     4.4
Biofeatures      4          3    75.0
```

If 'geneSum' argument is set to 'TRUE', a list of gene names is reported instead which informs on the nearest gene symbols to the set of Func-y-SNPs. Only unique gene symbols are reported since multiple distinct Func-y-SNP can be near the same gene.

```
> FunciSNPtable(glioma.anno, rsq=0.44, geneSum=TRUE);
```

```
      Gene_Names
1          CDKN2B
2           LIME1
3          PHLDB1
4        SLC2A4RG
5            TERT
6        TNFRSF6B
```

```
7           TREH
8          ZGPAT
9 RTEL1;TNFRSF6B
```

5.2 Summary of correlated SNPs overlapping biofeatures

FunciSNPsummaryOverlaps function helps to determine the total number of Func-y-SNPs overlapping a number of different biofeatures. This is similar to running *summary* on `glioma` above, except now you can specifically call the function and set a pre-determined 'rsq' value to subset the data and thereby obtain a more objective and informative result.

```
> FunciSNPsummaryOverlaps(glioma.anno)
```

```
                    bio.1 bio.2 bio.3
rs2736100              94     0     0
rs4977756              84     1     0
rs498872               87     8     1
rs6010620             266    52    11
TOTAL # 1000GP SNPs   531    61    12
```

Using a 'rsq' value, the output is subsetted to summarize the results with Rsquare values $\geq$ 'rsq'.

```
> FunciSNPsummaryOverlaps(glioma.anno, rsq=0.44)
```

```
                    bio.1 bio.2 bio.3
rs2736100               1     0     0
rs4977756               6     0     0
rs498872                4     0     0
rs6010620              42     8     3
TOTAL # 1000GP SNPs    53     8     3
```

5.3 Summary of correlated SNPs for a number of different tagSNPs

After running *FunciSNPsummaryOverlaps*, the next question one would like to know is which correlated SNPs overlapping a number of different biofeatures for a number of associated tagSNP. Thus, in the example above, we have determined that we are interested in learning more about the Func-y-SNPs associated with 'rs6010620' and which overlap at least 2 different biofeatures.

```
> rs6010620 <- FunciSNPidsFromSummary(glioma.anno, tagsnpid="rs6010620",
+ num.features=2, rsq=0.44);
> #summary(rs6010620);
> dim(rs6010620);
```

```
[1] 19 28
```

```
> class(rs6010620);
```

```
[1] "data.frame"
```

```
> ## See FunciSNPbed to visualize this data in a genome browser.
```

6 Plot FunciSNP results

6.1 Default plot

FunciSNPplot is a function developed to plot various types of plots to summarize and assist end-user in making informed discoveries of FunciSNP results. Plots can be stored in a folder for future reference. Most plots were created in with the idea that they can be directly outputted in presentations or publication formats.

The following example plots the distribution of the Rsquare values for each Func-y-SNP (Figure 1, page 16). We recommend attempting this plot before subsetting any data by a specified rsq value. The distribution helps to identify a specific Rsquare value that will provide the most informative discovery.

```
> pdf("glioma_dist.pdf")
> FunciSNPplot(glioma.anno)
> dev.off()
```

```
null device
          1
```

Figure 1 (page 16) illustrates the total number of Func-y-SNPsbinned at different Rsquare cutoffs. As you can see in this figure (1, page 16), there are a total of 11 Func-y-SNP with an Rsquare ≥ 0.9. Since this plot does not take into consideration unique Func-y-SNP the number may represent duplicate Func-y-SNP since they may overlap more than one biological feature.

6.2 Split by tagSNP

Using 'splitbysnp' argument, the same type of plot as above (Figure 1, page 16) is generated, however the total number of Func-y-SNPs are now divided by the associated tagSNP (Figure 2, page 17). It should be clear from this plot that 3 of the 4 tagSNP have a number of Func-y-SNP with Rsquares ≥ 0.5. And one tagSNP contains many more Func-y-SNP ('rs6010620').

```
> FunciSNPplot(glioma.anno, splitbysnp=TRUE)
> ggsave("glioma_dist_bysnp.pdf")
```

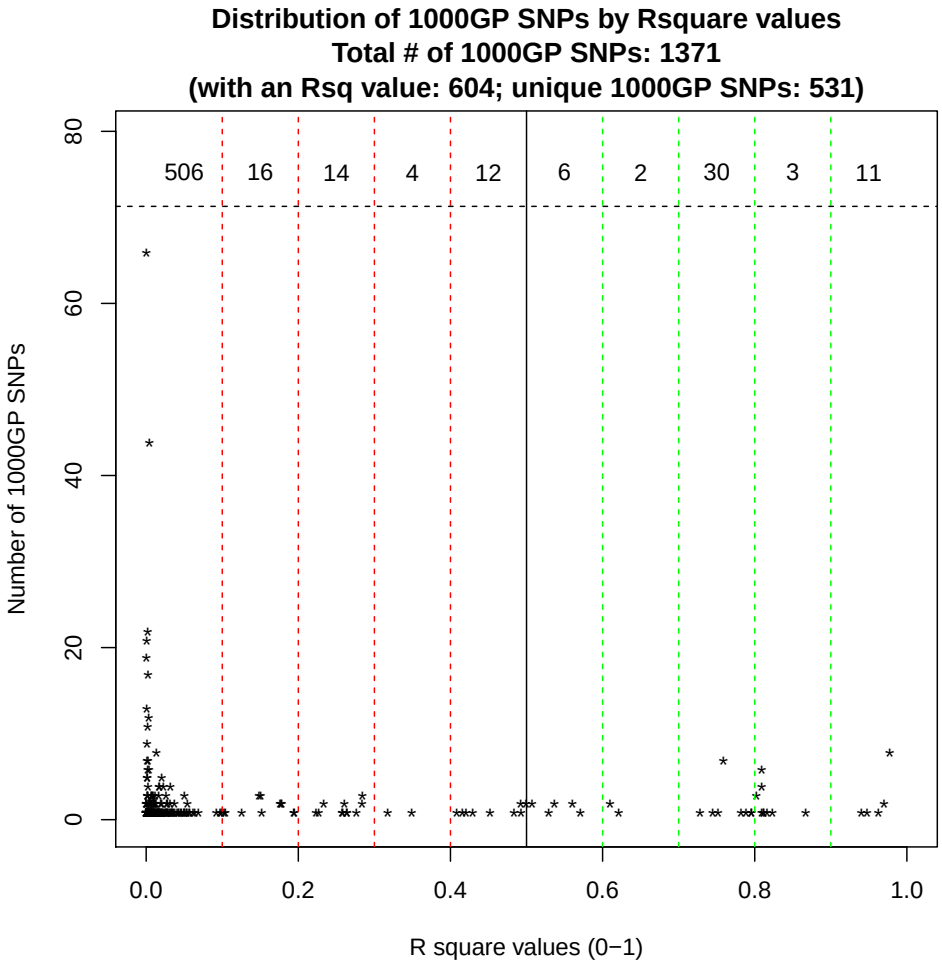

Figure 1: Distribution of Rsquare values of all Func-y-SNPs. Each marked bin contains the total number of Func-y-SNPs (correlated SNPs). The sum of all the counts would total the number of correlated SNPs.

6.3 Heatmap of 1000GP SNPs by tagSNP vs Biofeature

Now, if you are interested in knowing which biofeature and associated tagSNP contains the most number of 1000GP SNPs, run the following.

```
> pdf("glioma_heatmap.pdf")
> FunciSNPplot(glioma.anno, heatmap=TRUE, rsq = 0.1)
> dev.off()

X11cairo
       2
```

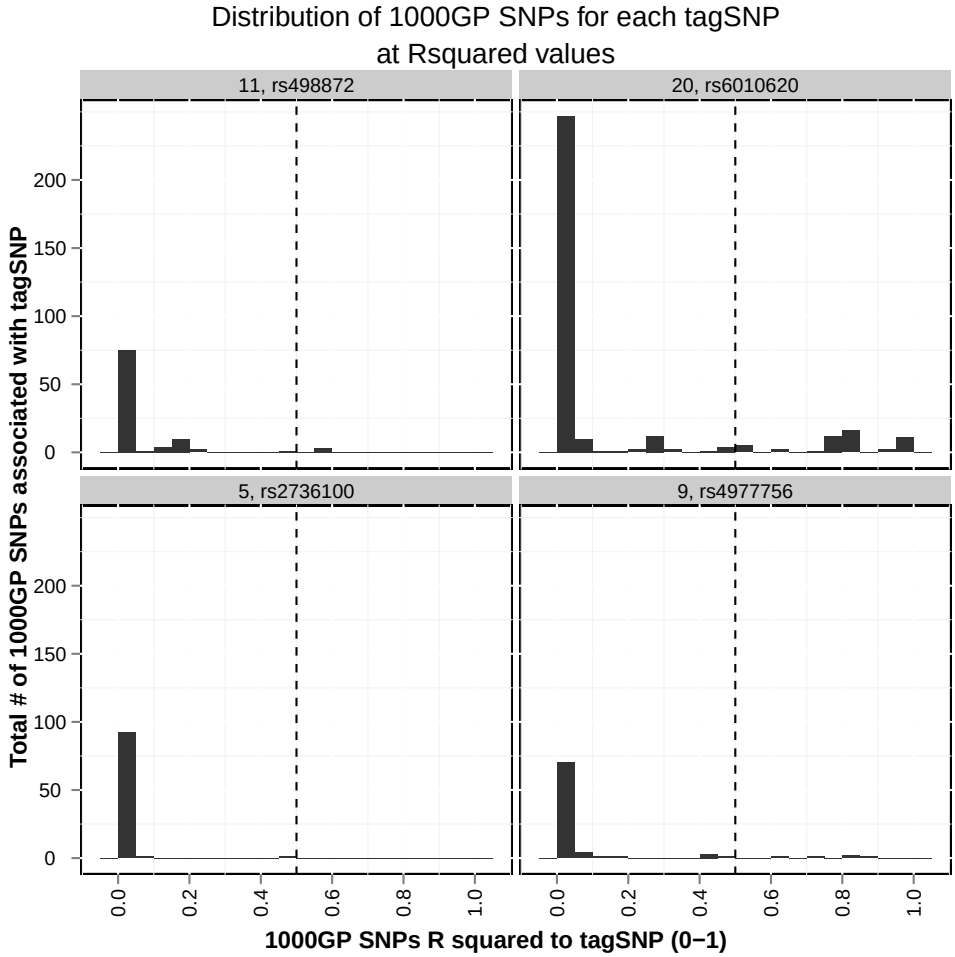

Figure 2: Distribution of Rsquare values of all Func-y-SNPs divided by the tagSNP and by its genomic location.

6.4 TagSNP and Biofeature Summary

Using 'tagSummary' argument will automatically save all plots in a specific folder. This is done because this function will generate a summary plot for each biofeature. The first plot (Figure 4, page 19) is a scatter plot showing the relationship between Rsquare and Distance to tagSNP for each Func-y-SNP.

```
> ## Following will output a series of plots for each biofeature at rsq=0.5
> FunciSNPplot(glioma.anno, tagSummary=TRUE, rsq=0.5)

Finished plotting  1 / 4
Finished plotting  2 / 4
```

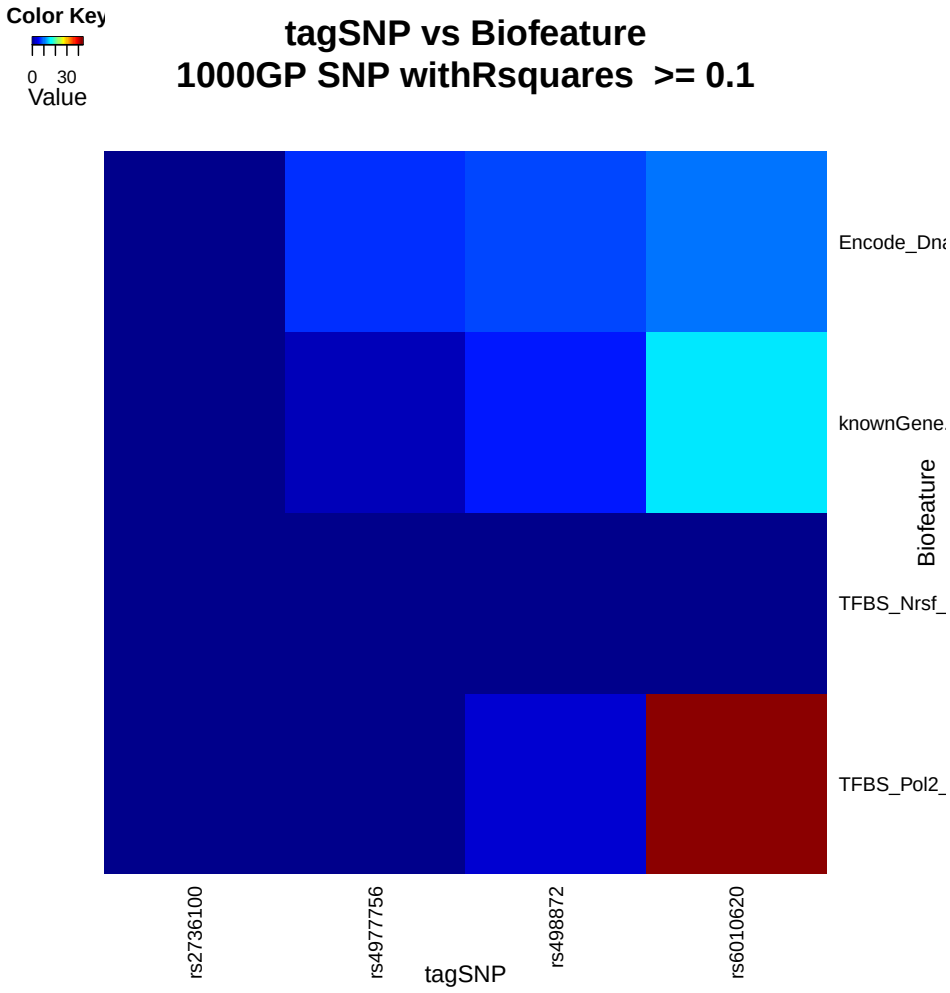

Figure 3: Heatmap of the number of 1000GP SNPs by relationship between tagSNP and biofeature.

```
Finished plotting  3 / 4
Finished plotting  4 / 4
```

Figure 4 on page 19 helps identify the relative postion of all newly identified Func-y-SNP to the associated tagSNP. As highlighted in figure 4, it is clear that tagSNP 'rs6010620' contains many more Func-y-SNP with Rsquares ≥ 0.5, and the majority of them are within 40,000 base pairs of the tagSNP. There are a few Func-y-SNP which are more than 50,000 base pairs away while some are within 5,000 base pairs.

The second plot (Figure 5, page 20) is a histogram distribution of total number of Func-y-SNPs at each Rsquare value. This plot is similar to Figure 2 on page 17, except it is further divided by biofeature. Each set of plot is further divided by tagSNP to help identify locus with the most identifiable Func-y-SNP. This argument is best used in conjunction with a

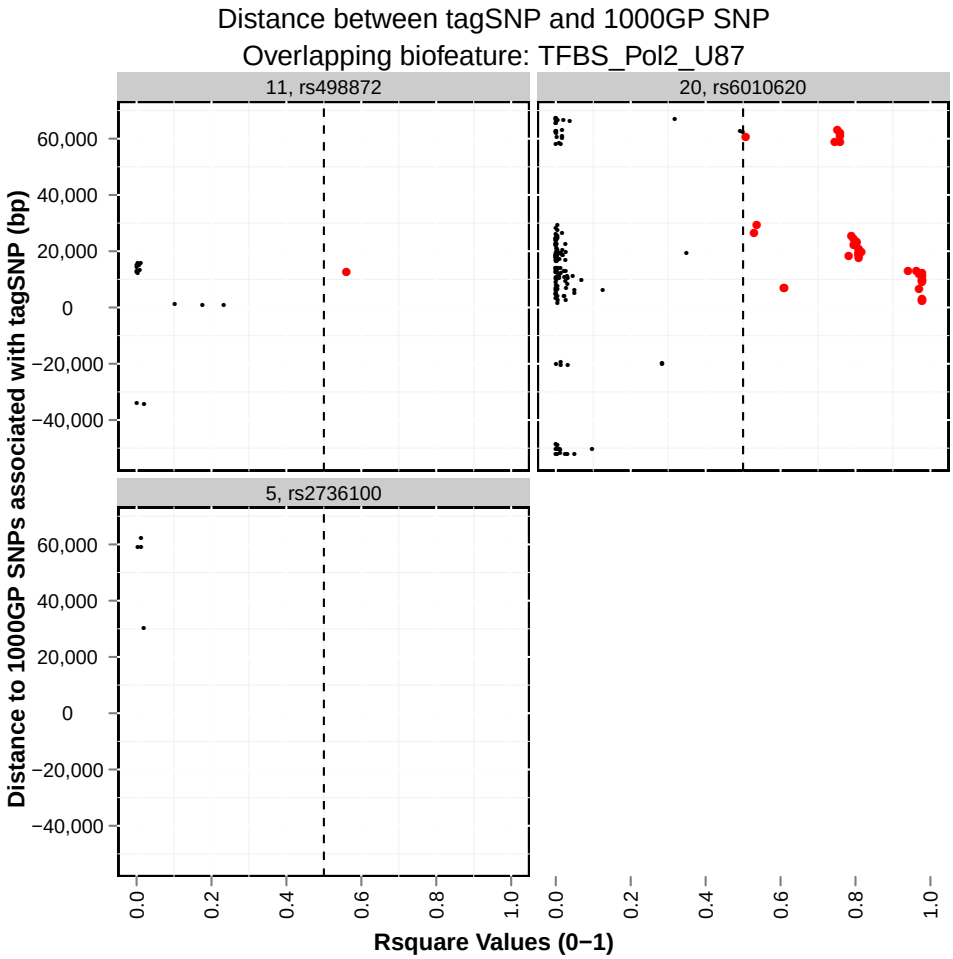

Figure 4: Scatter plot showing the relationship between Rsquare and Distance to tagSNP for each getFSNPs

'rsq' value.

6.5 Genomic Feature Summary

Using 'genomicSum' argument set to 'TRUE' will output the overall genomic distribution of the newly identified Func-y-SNPs (Figure 6, page 21). Using 'rsq' value, the plot is divided into all Func-y-SNPs vs subset. This type of plot informs the relative enrichment for genomic features.

```
> pdf("glioma_genomic_sum_rcut.pdf")
> FunciSNPplot(glioma.anno, rsq=0.5, genomicSum=TRUE, save=FALSE)
> dev.off()
```

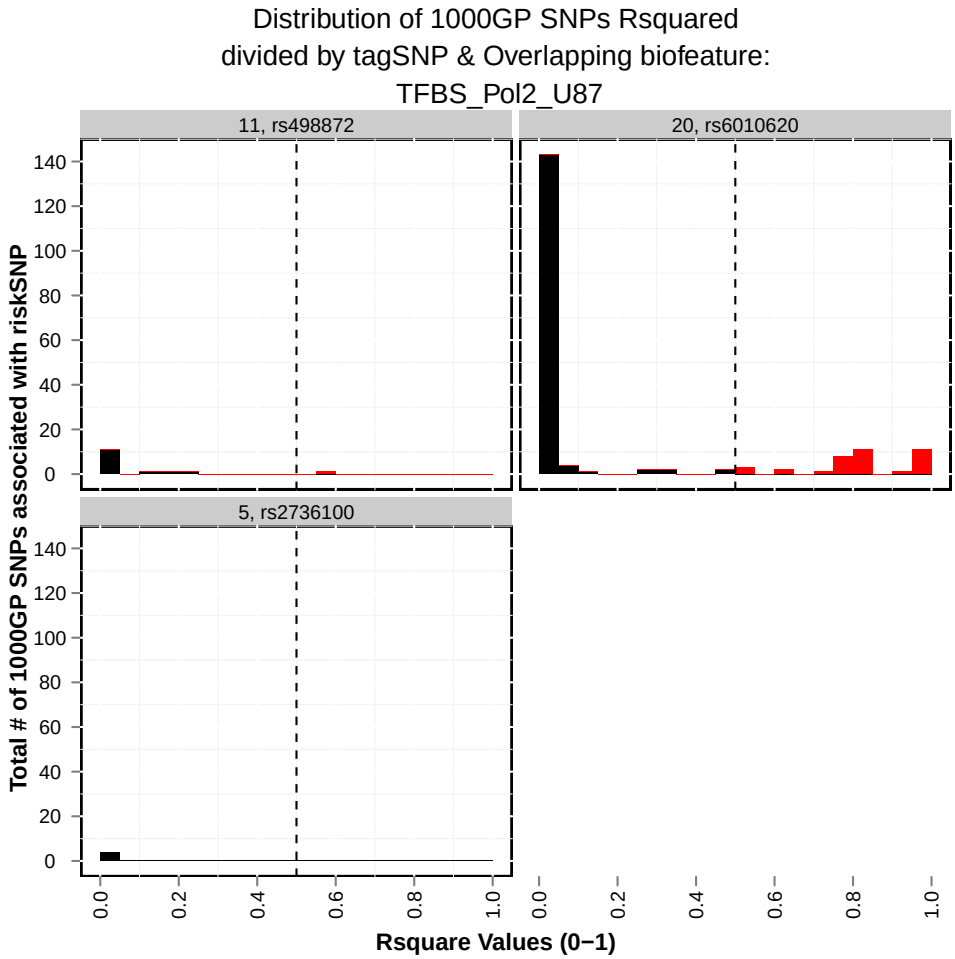

Figure 5: Histogram distribution of number of correlated SNPs at each Rsquare value

```
X11cairo
       2
```

Figure 6 on page 21 illustrates the distribution of the Func-y-SNP by genomic features. It is clear by using an Rsquare cutoff of 0.5, there is a slight enrichment of Func-y-SNP in introns and exonds and a depletion at promoters and other coding regions as well as intergentic regions.

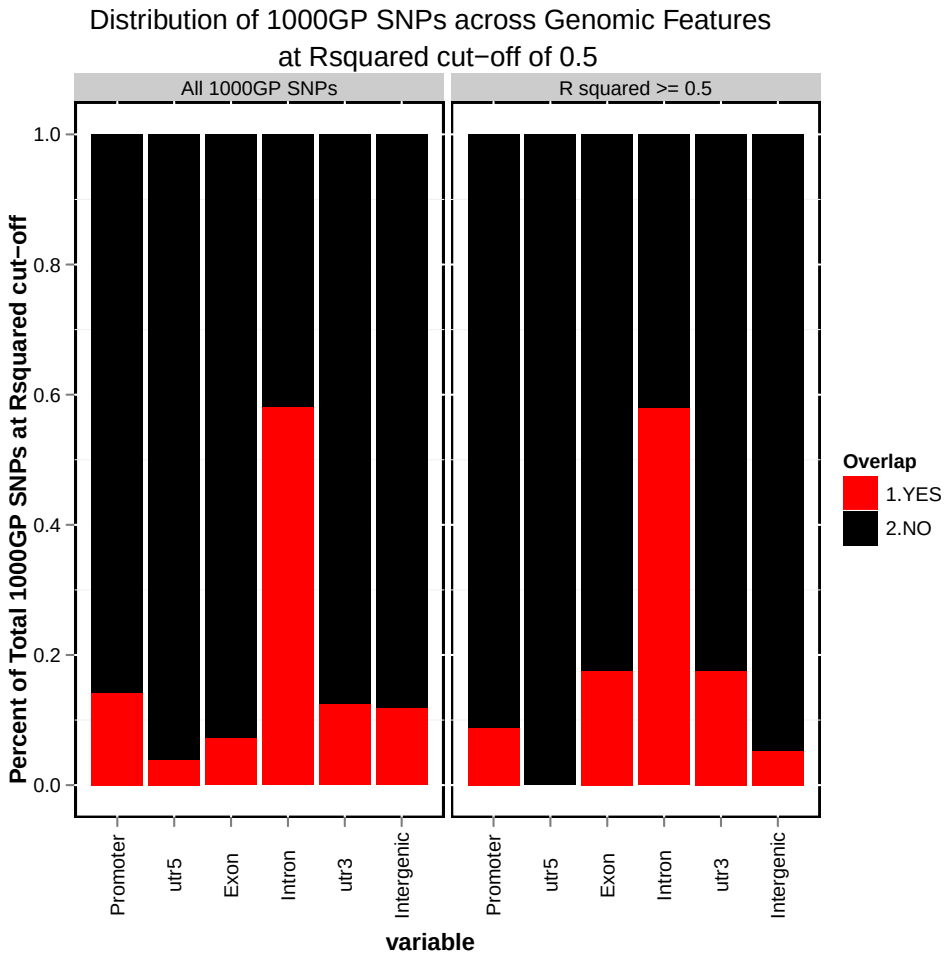

Figure 6: Stacked bar chart summarizing all correlated SNPs for each of the identified genomie features: exon, intron, 5UTR, 3UTR, promoter, lincRNA or in gene desert. Rsquare cutoff at 0.5. This plot is most informative if used with a rsq value.

7 Visualize FunciSNP results in a genomic browser (outputs BED format)

Finally, after evaluating all results using the above tables and plots functions, a unique pattern emerges that helps identify a unique cluster of tagSNP and biofeature that can identify a set of Func-y-SNPs. To better visualize and to get a better perspective of the location of each newly identified Func-y-SNP, the results can be outputted using *FunciSNPbed*.

FunciSNPbed outputs a unique BED file which can be used to view in any genomic browser which supports BED formats. To learn more about BED formats, see UCSC Genome Browser FAQ (`http://genome.ucsc.edu/FAQ/FAQformat`).

```
> ## will output to current working directory.
> FunciSNPbed(glioma.anno, rsq=0.22);

Total corSNP (RED):  71
Total tagSNP (BLK):  4

> # FunciSNPbed(rs6010620, rsq=0.5);
```

Each tagSNP which is in LD to a corresponding Func-y-SNP overlapping at least one biofeature is colored black, while the Func-y-SNP is colored red. The initial position is provided by the first tagSNP and the first linked Func-y-SNP. We recommend using UCSC genome browser to view your BED files. This is useful so you can view all public and private tracks in relation to FunciSNP results. As an example, see Figure 7 on page 22 or visit this saved UCSC Genome Browser session: `http://goo.gl/xrZPD`.

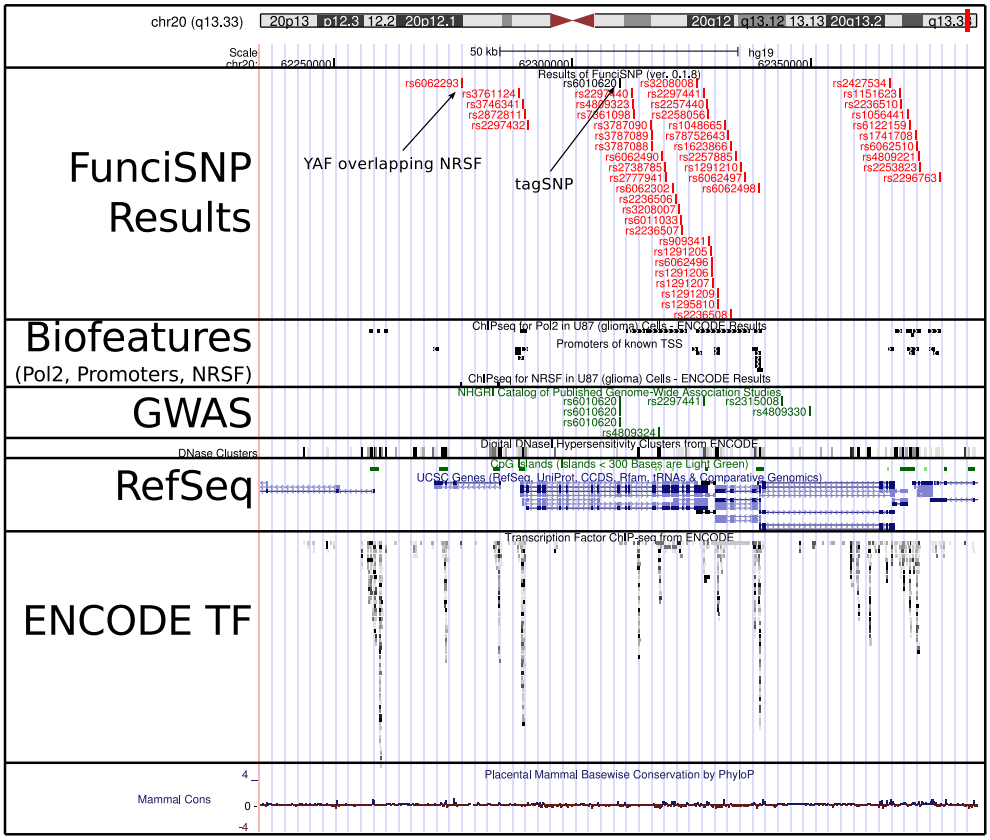

Figure 7: FunciSNP results viewed in UCSC genome browser. Top track represents FunciSNP results, second track is the known GWAS hits.

8 Contact information

Questions or comments, please contact Simon G. Coetzee (scoetzee NEAR gmail POINT com) or Houtan Noushmehr, PhD (houtana NEAR gmail POINT com).

9 sessionInfo

- R version 2.14.2 (2012-02-29), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=en_US.UTF-8`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=C`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, graphics, grDevices, grid, methods, splines, stats, tools, utils
- Other packages: annotate 1.32.1, AnnotationDbi 1.16.11, Biobase 2.14.0, biomaRt 2.10.0, Biostrings 2.22.0, bit 1.1-8, bitops 1.0-4.1, BSgenome 1.22.0, BSgenome.Ecoli.NCBI.20080805 1.3.17, caTools 1.12, ChIPpeakAnno 2.2.0, DBI 0.2-5, ff 2.2-4, FunciSNP 0.1.8, gdata 2.8.2, genefilter 1.36.0, GenomicFeatures 1.6.7, GenomicRanges 1.6.4, GGBase 3.14.0, ggplot2 0.8.9, GGtools 4.0.0, GO.db 2.6.1, gplots 2.10.1, gtools 2.6.2, IRanges 1.12.5, KernSmooth 2.23-7, lattice 0.20-0, limma 3.10.2, matlab 0.8.9, Matrix 1.0-4, multtest 2.10.0, org.Hs.eg.db 2.6.4, plyr 1.7.1, proto 0.3-9.2, RCurl 1.9-5, reshape 0.8.4, Rsamtools 1.6.3, RSQLite 0.11.1, rtracklayer 1.14.4, snpStats 1.4.1, survival 2.36-12, TxDb.Hsapiens.UCSC.hg19.knownGene 2.6.2, VariantAnnotation 1.0.5
- Loaded via a namespace (and not attached): digest 0.5.1, MASS 7.3-16, parallel 2.14.2, XML 3.9-2, xtable 1.6-0, zlibbioc 1.0.0

hn