SUPPLEMENTAL MATERIAL

SEQMINER: An R-package to Facilitate the Functional Interpretation of Sequence-based Associations

Xiaowei Zhan[1,4,*] Dajiang J. Liu[2,3,4,*]

1. Quantitative Biomedical Research Center, Department of Clinical Sciences, Center for the Genetics of Host Defense, University of Texas Southwestern Medical Center, Dallas, TX, 75235

2. Institute for Personalized Medicine, College of Medicine, Pennsylvania State University, Hershey, PA, 17033

3. Division of Biostatistics, Department of Public Health Sciences, College of Medicine, Pennsylvania State University, Hershey, PA, 17033

4. Manuscript correspondence should be addressed to:

X.Z. (Email: zhanxw@gmail.com, Tel: 1-214-648-5194)

D.J.L (Email: dajiang.liu@psu.edu, Tel: 1-717-531-4178)

*: These authors contributed equally to the work.

1. EXEMPLAR WORKFLOW AND USAGE FOR SEQMINER FOR ANNOTATING VCF FILES

In this section, we provide an exemplar workflow and introduce available features in *SEQMINER* for annotating and retrieving VCF files. The example is available in the demo of our R-package, and can be reproduced by the command

```
> demo(vcf.workflow, package=”seqminer”)
```

To annotate sequence variants and integrate genomic data, reference genomes and gene transcript definitions must be specified.

```
> param <- list(reference = system.file("tabanno/test.fa", package = "seqminer"), geneFile
=system.file("tabanno/test.gene.txt", package = "seqminer"))
> param <- makeAnnotationParameter(param)
```

Next, gene-based annotation may be performed, where each mutation is compared with the reference genome and gene definitions. The genetic mutations are annotated by the changes on the protein sequences they induce.

```
> input  <- "input.demo.vcf"
> output <- "out.vcf.gz"
> annotateVcf (input, output, param)
```

In the output VCF, the annotation information is written to the ANNO field within the INFO tab. Therefore, the output still conforms to standard VCF specifications.

Using the functionalities of region-based annotation, *SEQMINER* is also capable of integrating bioinformatics databases. Specifically, bioinformatics databases of interest can be formatted as BED files (Kent, et al., 2002). To annotate genetic variants, *SEQMINER* links records in the BED files to corresponding variants in the VCF files or generic TSV files. In the example below, we illustrate how *SEQMINER* can be used to integrate SIFT scores (Ng and Henikoff, 2001) for coding variants in the genome.

```
> param  <- list(reference = "test.fa",
+              geneFile = "test.gene.txt",
+              bed = "REGION=test.bed",
+              tabix = "test.dbNSFP.gz(SIFT=9,PolyPhen=10)",
+              indexOutput = TRUE)

> param  <- makeAnnotationParameter(param)
> input  <- "input.demo.vcf"
> output <- "out.vcf.gz"
```

Leveraging annotated VCF files, complex queries can be performed with built-in functions for *SEQMINER*. Examples include extracting genetic variants by *tabix*-range. Extracted information is automatically parsed and stored in standard R-objects (e.g., matrix, data frame or list). These R-objects can be directly analyzed in downstream statistical analysis and used in data quality control, visualization, etc.

```
> genotypeList <- readVCFToListByRange(fileName = output,
+                              range = "1:1-10",
+                              annoType = "",
+                              vcfColumn = c("CHROM", "POS"),
+                              vcfInfo = c("ANNO", "SIFT"),
+                              vcfIndv = "GT")

> print(genotypeList)
$CHROM
[1] "1" "1" "1"

$POS
[1] 3 5 7

$ANNO
[1] "Normal_Splice_Site:GENE1|GENE3" "Nonsynonymous:GENE1|GENE3"
[3] "Monomorphic"

$SIFT
[1] "0.0" "NA"  "NA"

$GT
    [,1]  [,2]  [,3]
[1,] "1/0" "1/0" "1/0"
[2,] "0/0" "0/0" "0/0"
$sampleId
[1] "NA12891" "NA12892"
```

Similarly, queries to VCF files can also be performed by specifying the gene region of interest.

```
> genotypeList <- readVCFToListByGene(fileName = output,
+                              geneFile = geneFile,
+                              geneName = "GENE1",
+                              annoType = "",
+                              vcfColumn = c("CHROM", "POS"),
+                              vcfInfo = "ANNO",
+                              vcfIndv = "GT")
range of [ GENE1 ] is [ 1:1-67 ]
range = 1:1-67
```

Subsequent statistical analysis can be performed using extracted information.

2. WORKFLOW FOR ANNOTATING SUMMARY ASSOCIATION STATISTICS

In the following, we present an example of using SEQMINER to annotate files of summary association statistics. First we load the package and download the resource files that are necessary to annotate sequence variants (e.g., reference genomes, transcript definitions).

```
> library(seqminer)
> download.annotation.resource("~/seqminer.annotation")
```

Next, we specify the parameters used to annotate summary association statistics. The parameter *reference* specifies the reference genome; the parameter *geneFile* specifies the gene definition; the parameter *codonFile* specifies the codon definition, i.e., how triples of DNA base pairs translate to amino acids; and the parameter *priority* specifies the priority of annotations. When the genetic variant lies on multiple transcripts, the transcript-specific annotations can differ. Priorities can be assigned to these annotations based upon how deleterious the annotation is.

```
param <- makeAnnotationParameter(list(reference = "~/seqminer.annotation/hs37d5.fa",
                                      geneFile = "~/seqminer.annotation/refFlat_hg19.txt.gz",
                                      codonFile = "~/seqminer.annotation/codon.txt",
                                      priorityFile = "~/seqminer.annotation/priority.txt" ))
```

Finally, we specify the input, output files and annotate the summary association statistics using the function *annotatePlain*.

```
> summary.file <- system.file("rvtests/rvtest.MetaScore.assoc.gz", package = "seqminer")
> out.file <- system.file("rvtests/rvtest.MetaScore.assoc.anno.gz", package = "seqminer")
> annotatePlain(summary.file, out.file, param)
```

3. WORKFLOW FOR RETRIEVING SUMMARY ASSOCIATION STATISTICS

Below, we describe an example of using SEQMINER to retrieve summary association statistics from files of RAREMETAL format. The workflow can be reproduced in R using the following command:

```
> demo('meta.workflow')
```

Summary association statistics can be queried by its genomic ranges, expressed using tabix format. For example, to retrieve a genomic region from chromosome 1 in the interval (196621000,196623000), we use the following command:

```
> stats <- rvmeta.readScoreByRange(score.file,
+                                  "1:196621000-196623000")
> print(names(stats))
 [1] "pos"      "ref"      "alt"      "nSample"  "af"       "ac"
 [7] "callRate" "hwe"      "nref"     "nhet"     "nalt"     "ustat"
[13] "vstat"    "effect"   "pVal"     "anno"     "annoFull"
> print(stats[1:5])
$pos
[1] 196621169 196622041
$ref
[1] "A" "A"
$alt
[1] "G" "G"
$nSample
[1] 1092 1092
$af
[1] 0.0247253 0.0306777
```

Covariance matrices between summary association statistics can also be queried using the function *rvmeta.readCovByRange*. An exemplar usage is below:

```
> cov.file <- system.file("rvtests/rvtest.MetaCov.assoc.gz", package = "seqminer")

> stats <- rvmeta.readCovByRange(cov.file,
+                                "1:196621000-196623000")
Total 2 line loaded, now put them to matrix [ 2 x 2 ] in R ...

> print(stats)
           1:196621169 1:196622041
1:196621169  0.05426640  0.00154286
1:196622041  0.00154286  0.06301010
```

To perform meta-analyses of gene-level tests of rare variant associations, both single variant association statistics and their covariance matrices are needed. We implement functions *rvmeta.readDatabyRange* or *rvmeta.readDatabyGene* to retrieve both single variant score statistics and their covariance matrices simultaneously. Retrieved summary association statistics are collated by the variant position, and made ready for subsequent statistical analysis. An exemplar usage is shown below:

```
> stats <- rvmeta.readDataByGene(score.file, cov.file, geneFile = gene.file,
+                                gene = "CFH")
range of [ CFH ] is [ 1:196621007-196716634 ]
1 gene/region to be extracted.
Read score tests...
In study 0
Done read score file: /net/fantasia/home/dajiang/R/library/seqminer/rvtests/rvtest.MetaScore.assoc.anno.gz
Read cov files ...
Done read cov file: /net/fantasia/home/dajiang/R/library/seqminer/rvtests/rvtest.MetaCov.assoc.gz
Finished calculation.

> print(names(stats))
[1] "CFH"

> print(names(stats[[1]]))
 [1] "ref"          "alt"          "nSample"      "af"           "ac"
 [6] "callrate"     "hwe"          "nref"         "nhet"         "nalt"
[11] "ustat"        "vstat"        "effect"       "pVal"         "cov"
[16] "pos"          "anno"         "covXZ"        "covZZ"        "hweCase"
[21] "hweCtrl"      "afCase"       "afCtrl"       "acCase"       "acCtrl"
[26] "callrateCase" "callrateCtrl" "nrefCase"     "nrefCtrl"     "nhetCase"
[31] "nhetCtrl"     "naltCase"     "naltCtrl"     "nCase"        "nCtrl"

> print(stats[[1]][1:5])
$ref
$ref[[1]]
 [1] "A" "A" "T" "A" "G" "A" "T" "G" "G" "T" "A" "G" "G" "A" "T" "A" "A" "T" "C"
[20] "G" "T" "A" "C" "C" "A" "A" "T" "T" "G" "C" "G" "T" "G" "A" "G" "C" "G" "A"
[39] "G" "C" "C" "C" "G" "T" "G" "C" "C" "G" "T" "G" "G" "C" "G" "C" "A" "T" "A"
$alt
$alt[[1]]
 [1] "G" "G" "C" "G" "T" "G" "G" "A" "A" "G" "G" "A" "A" "G" "C" "C" "C" "G" "T"
[20] "T" "C" "G" "A" "A" "C" "G" "A" "C" "A" "T" "A" "C" "T" "G" "A" "A" "A" "G"
[39] "C" "T" "T" "T" "A" "C" "A" "T" "A" "T" "C" "T" "T" "T" "T" "T" "T" "C" "C"
$nSample
$nSample[[1]]
 [1] 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092
[16] 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092
[31] 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092
[46] 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092 1092
$af
$af[[1]]
 [1] 0.024725300 0.030677700 0.118132000 0.028846200 0.249542000 0.021520100
 [7] 0.000457875 0.434524000 0.010531100 0.733974000 0.765110000 0.000457875
[13] 0.000915751 0.133700000 0.189103000 0.704212000 0.722985000 0.006868130
[19] 0.721612000 0.288004000 0.527473000 0.722527000 0.703755000 0.749542000
[25] 0.437729000 0.017399300 0.000457875 0.016941400 0.187729000 0.241758000
[31] 0.007783880 0.000457875 0.000457875 0.233059000 0.001373630 0.000457875
[37] 0.461081000 0.741758000 0.741758000 0.440934000 0.239469000 0.440934000
[43] 0.311813000 0.242216000 0.003205130 0.004120880 0.000457875 0.051739900
[49] 0.005494510 0.232601000 0.003205130 0.000915751 0.074633700 0.004120880
[55] 0.012820500 0.022893800 0.016483500
$ac
$ac[[1]]
 [1]   54   67  258   63  545   47    1  949   23 1603 1671    1    2  292  413
[16] 1538 1579   15 1576  629 1152 1578 1537 1637  956   38    1   37  410  528
[31]   17    1    1  509    3    1 1007 1620 1620  963  523  963  681  529    7
[46]    9    1  113   12  508    7    2  163    9   28   50   36
```