Annotating Genetic Variants - Exercises

Valerie Obenchain

27-28 February 2012

1 Overview

In these excercises we will investigate the TRPV (Transient Receptor Potential Vanilloid) family of transient receptor potential ion channels. These channels are selective for calcium and magnesium over sodium ions. Our goal will be to read in variants that fall in the gene ranges, identify their structural location in the gene and determine the consequece of any amino acid coding changes.

In these exercises we use a VCF file available in the *cgdv17* data package. The package contains Complete Genomics data for chromosome 17 from 11 populations. We will be using one file from population type CEU.

2 Reading VCF files

```
> library(VariantAnnotation)
> library(cgdv17)
> file <- system.file("vcf", "NA06985_17.vcf.gz", package = "cgdv17")
> genefam <- c("TRPV1", "TRPV2", "TRPV3")
> library(org.Hs.eg.db)
> ## get ensembl ids from gene symbols
> geneid <- lapply(genefam, function(gn) get(gn, revmap(org.Hs.egSYMBOL)))
```

Exercise 1
Explore the file header with `scanVcfHeader`*. What elements are in the INFO and FORMAT fields?*

Solution:

```
> hdr <- scanVcfHeader(file)
> hdr[[1]]$Header$INFO
```

```
DataFrame with 3 rows and 3 columns
        Number        Type                Description
   <character> <character>                <character>
NS           1     Integer Number of Samples With Data
DP           1     Integer                Total Depth
DB           0        Flag dbSNP membership, build 131
```

```
> hdr[[1]]$Header$FORMAT

DataFrame with 12 rows and 3 columns
            Number        Type                          Description
       <character> <character>                          <character>
GT                1      String                             Genotype
GQ                1     Integer                    Genotype Quality
DP                1     Integer                          Read Depth
HDP               2     Integer                Haplotype Read Depth
HQ                2     Integer                   Haplotype Quality
PS                2     Integer                           Phase Set
GENE              .      String                    Overlaping Genes
mRNA              .      String                     Overlaping mRNA
rmsk              .      String                   Overlaping Repeats
segDup            .      String Overlaping segmentation duplication
rCov              1       Float                    relative Coverage
cPd               1      String                called Ploidy(level)
```

Exercise 2

- *Extract the ranges for the TRPV family of genes using the TxDb.Hsapiens.UCSC.hg19.knownGene package and Entrez gene ids.*
- *Use the* `transcriptsBy` *function to create a GRangesList of transcripts by gene. Subset this object on chromosome 17 with* `keepSeqlevels`
- *The seqlevels (chromosomes) from the VCF file are named as numbers only; they are not preceded by 'chr'. The ranges we extract from the annotation will be used to retrieve data from the VCF file so the seqlevels must match. Rename the seqlevels (chromosomes) in the GRangesList from 'chr17' to '17'.*
- *Extract the TRPV gene ranges from the GRangesList annotation*

Solution:

```
> library(TxDb.Hsapiens.UCSC.hg19.knownGene)
> txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
> txbygene = transcriptsBy(txdb, "gene")
> tx_chr17 <- keepSeqlevels(txbygene, "chr17")
> tx_17 <- renameSeqlevels(tx_chr17, c(chr17="17"))
> rngs <- unlist(tx_17[names(tx_17) %in% unlist(geneid)], use.names = FALSE)
```

To retrieve a subset of the data from a VCF file we need to create a `ScanVcfParam`. This object can specify genomic coordinates (ranges) or individual VCF elements.

The VCF file must have a tabix index when the data are subset on ranges. An index for this file exists in the data package. In the case where you need to create your own index see ?`indexTabix` for help. Read in the data with `readVcf`.

Exercise 3
Create a `ScanVcfParam` with the ranges extracted from the `TxDb`. The ranges can be collapsed into a single range with `reduce`. If the ranges are not collapsed the `rangesID` column in the `rowData` slot of the result will display which variant came from each range.

Solution:

```
> gnrng <- reduce(rngs)
> param <- ScanVcfParam(which = gnrng, info = "DP", geno = c("GT", "cPd"))
> vcf <- readVcf(file, "hg19", param)
```

Explore the *VCF* object using the `fixed`, `info` and `geno` accessors. Variant ranges are in the *GRanges* found in the `rowData` slot.

3 structural location of variants

When using annotations for overlapping or matching it is important the `seqlevels` match. Here we check the `seqlevels` of the `VCF` file against those of the `txdb`.

```
> seqlevels(vcf)
```

```
[1] "17"
```

```
> head(seqlevels(txdb))
```

```
[1] "chr1" "chr2" "chr3" "chr4" "chr5" "chr6"
```

```
> ## seqlevels do not match
> intersect(seqlevels(vcf), seqlevels(txdb))
```

```
character(0)
```

```
> vcf_mod <- renameSeqlevels(vcf, c("17"="chr17"))
> ## seqlevels now match
> intersect(seqlevels(vcf_mod), seqlevels(txdb))
```

```
[1] "chr17"
```

Exercise 4

- *Call `locateVariants` on the VCF object with the modified seqlevels. Each row of the result represents a variant-transcript match which may result in mulitple rows per variant. Be aware of this 'multiplicity' when intrepreting the reults.*

- *How many variants are in each structural region?*

Solution:

```
> loc <- locateVariants(vcf_mod, txdb)
> ## summarize by gene by region
> table(loc$location, loc$geneID)
```

```
                   162514 23729 51393 7442 84690
  transcript_region      0     0     0    0     0
  intron              1484     0   116  706     0
  5'UTR                 10     0     2    7     0
  3'UTR                 49     2     2   16     6
  coding                52     0     6   56     0
  intergenic             0     0     0    0     0
```

4 Amino acid coding

Load the *BSgenome.Hsapiens.UCSC.hg19* package and call `predictCoding` on the *VCF* object with modified seqlevels.

```
> library(BSgenome.Hsapiens.UCSC.hg19)
> aa <- predictCoding(vcf_mod, txdb, Hsapiens)
```

The *SIFT.Hsapiens.dbSNP132* and *PolyPhen.Hsapiens.dbSNP131* packages provide predictions of how dammaging amino acid coding changes may be on protein structure and function. Both methods use multiple alignment information and PolyPhen also utilizes protein structional databases. Details of the algorithems and outputs offered can be found at ?`SIFT.Hsapiens.dbSNP132` and ?`PolyPhen.Hsapiens.dbSNP131`.

Exercise 5

- *Load PolyPhen.Hsapiens.dbSNP131. Investigate the available columns and keys with the* `keys` *and* `cols` *functions.*
- *The keys in the SIFT and PolyPhen databases are rsid. Obtain the rsid of the nonsynonymous variants identified in the* `predictCoding` *call and query the PolyPhen database. Define a subset of columns to be returned and repeat the call.*

Solution:

```
> library(PolyPhen.Hsapiens.dbSNP131)
> keys <- keys(PolyPhen.Hsapiens.dbSNP131)
> cols <- cols(PolyPhen.Hsapiens.dbSNP131)
> nonsyn <- aa$queryID[aa$consequence == "nonsynonymous"]
```

```
> rsid <- unique(names(rowData(vcf_mod))[nonsyn])
> ## column descriptions found at ?PolyPhenDbColumns
> cols(PolyPhen.Hsapiens.dbSNP131)

 [1] "RSID"        "TRAININGSET" "OSNPID"      "OACC"        "OPOS"
 [6] "OAA1"        "OAA2"        "SNPID"       "ACC"         "POS"
[11] "AA1"         "AA2"         "NT1"         "NT2"         "PREDICTION"
[16] "BASEDON"     "EFFECT"      "PPH2CLASS"   "PPH2PROB"    "PPH2FPR"
[21] "PPH2TPR"     "PPH2FDR"     "SITE"        "REGION"      "PHAT"
[26] "DSCORE"      "SCORE1"      "SCORE2"      "NOBS"        "NSTRUCT"
[31] "NFILT"       "PDBID"       "PDBPOS"      "PDBCH"       "IDENT"
[36] "LENGTH"      "NORMACC"     "SECSTR"      "MAPREG"      "DVOL"
[41] "DPROP"       "BFACT"       "HBONDS"      "AVENHET"     "MINDHET"
[46] "AVENINT"     "MINDINT"     "AVENSIT"     "MINDSIT"     "TRANSV"
[51] "CODPOS"      "CPG"         "MINDJNC"     "PFAMHIT"     "IDPMAX"
[56] "IDPSNP"      "IDQMIN"      "COMMENTS"

> subst <- c("AA1", "AA2", "PREDICTION")
> select(PolyPhen.Hsapiens.dbSNP131, keys=rsid, cols=subst)

       RSID AA1 AA2 PREDICTION
3  rs322965   I   V     benign
31 rs224534   T   I     benign
```