AnnotationHub How-To's

Package: *AnnotationHub* **Authors**: Martin Morgan [cre], Marc Carlson [ctb], Dan Tenenbaum [ctb], Sonali Arora [ctb] **Modified**: Sun Jun 28 10:41:23 2015 **Compiled**: Mon Jul 2 15:23:41 2018

Accessing Genome-Scale Data

Non-model organism gene annotations

Bioconductor offers pre-built `org.*` annotation packages for model organisms, with their use described in the OrgDb section of the Annotation work flow. Here we discover available `OrgDb` objects for less-model organisms

```
library(AnnotationHub)
ah <- AnnotationHub()
```

```
## snapshotDate(): 2018-04-30
```

```
query(ah, "OrgDb")
```

```
## AnnotationHub with 1691 records
## # snapshotDate(): 2018-04-30
## # $dataprovider: ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/
## # $species: Escherichia coli, 'Caballeronia concitans', 'Chlorella vulgaris'_C-169, 'Frankia cas...
## # $rdataclass: OrgDb
## # additional mcols(): taxonomyid, genome, description, coordinate_1_based, maintainer,
## #   rdatadateadded, preparerclass, tags, rdatapath, sourceurl, sourcetype
## # retrieve records with, e.g., 'object[["AH61768"]]'
##
##             title
##   AH61768 | org.Ag.eg.db.sqlite
##   AH61769 | org.At.tair.db.sqlite
##   AH61770 | org.Bt.eg.db.sqlite
##   AH61771 | org.Cf.eg.db.sqlite
##   AH61772 | org.Gg.eg.db.sqlite
##   ...       ...
##   AH63468 | org.Salmonella_typhimurium_LT2.eg.sqlite
##   AH63469 | org.Acinetobacter_baumannii.eg.sqlite
##   AH63470 | org.Acinetobacter_genomosp._2.eg.sqlite
##   AH63471 | org.Acinetobacter_genomospecies_2.eg.sqlite
##   AH63472 | org.Bacterium_anitratum.eg.sqlite
```

```
orgdb <- query(ah, "OrgDb")[[1]]
```

```
## downloading 0 resources
```

```
## loading from cache
##     '/Users/sdavis2//.AnnotationHub/68514'
```

The object returned by AnnotationHub is directly usable with the `select()` interface, e.g., to discover the available keytypes for querying the object, the columns that these keytypes can map to, and finally selecting the SYMBOL and GENENAME corresponding to the first 6 ENTREZIDs

```
keytypes(orgdb)
```

```
##  [1] "ACCNUM"       "ENSEMBL"      "ENSEMBLPROT"  "ENSEMBLTRANS" "ENTREZID"     "ENZYME"
##  [7] "EVIDENCE"     "EVIDENCEALL"  "GENENAME"     "GO"           "GOALL"        "ONTOLOGY"
## [13] "ONTOLOGYALL"  "PATH"         "PMID"         "REFSEQ"       "SYMBOL"       "UNIGENE"
## [19] "UNIPROT"
```

```
columns(orgdb)
```

```
##  [1] "ACCNUM"       "ENSEMBL"      "ENSEMBLPROT"  "ENSEMBLTRANS" "ENTREZID"     "ENZYME"
##  [7] "EVIDENCE"     "EVIDENCEALL"  "GENENAME"     "GO"           "GOALL"        "ONTOLOGY"
## [13] "ONTOLOGYALL"  "PATH"         "PMID"         "REFSEQ"       "SYMBOL"       "UNIGENE"
## [19] "UNIPROT"
```

```
egid <- head(keys(orgdb, "ENTREZID"))
select(orgdb, egid, c("SYMBOL", "GENENAME"), "ENTREZID")
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
##   ENTREZID          SYMBOL      GENENAME
## 1  1267437 AgaP_AGAP012606 AGAP012606-PA
## 2  1267439 AgaP_AGAP012559 AGAP012559-PA
## 3  1267440 AgaP_AGAP012558 AGAP012558-PA
## 4  1267447 AgaP_AGAP012586 AGAP012586-PA
## 5  1267450 AgaP_AGAP012834 AGAP012834-PA
## 6  1267459 AgaP_AGAP012589 AGAP012589-PA
```

Roadmap Epigenomics Project

All Roadmap Epigenomics files are hosted here. If one had to download these files on their own, one would navigate through the web interface to find useful files, then use something like the following *R* code.

```
url <- "http://egg2.wustl.edu/roadmap/data/byFileType/peaks/consolidated/broadPeak/E001-H3K4me1.broadPe
filename <-  basename(url)
download.file(url, destfile=filename)
if (file.exists(filename))
   data <- import(filename, format="bed")
```

This would have to be repeated for all files, and the onus would lie on the user to identify, download, import, and manage the local disk location of these files.

AnnotationHub reduces this task to just a few lines of *R* code

```
library(AnnotationHub)
ah = AnnotationHub()
```

```
## snapshotDate(): 2018-04-30
```

```
epiFiles <- query(ah, "EpigenomeRoadMap")
```

A look at the value returned by `epiFiles` shows us that 18248 roadmap resources are available via *AnnotationHub*. Additional information about the files is also available, e.g., where the files came from (dataprovider), genome, species, sourceurl, sourcetypes.

```
epiFiles
```

```
## AnnotationHub with 18248 records
## # snapshotDate(): 2018-04-30
## # $dataprovider: BroadInstitute
## # $species: Homo sapiens
## # $rdataclass: BigWigFile, GRanges, data.frame
## # additional mcols(): taxonomyid, genome, description, coordinate_1_based, maintainer,
## #   rdatadateadded, preparerclass, tags, rdatapath, sourceurl, sourcetype
## # retrieve records with, e.g., 'object[["AH28856"]]'
##
##             title
##   AH28856 | E001-H3K4me1.broadPeak.gz
##   AH28857 | E001-H3K4me3.broadPeak.gz
##   AH28858 | E001-H3K9ac.broadPeak.gz
##   AH28859 | E001-H3K9me3.broadPeak.gz
##   AH28860 | E001-H3K27me3.broadPeak.gz
##   ...       ...
##   AH49540 | E058_mCRF_FractionalMethylation.bigwig
##   AH49541 | E059_mCRF_FractionalMethylation.bigwig
##   AH49542 | E061_mCRF_FractionalMethylation.bigwig
##   AH49543 | E081_mCRF_FractionalMethylation.bigwig
##   AH49544 | E082_mCRF_FractionalMethylation.bigwig
```

A good sanity check to ensure that we have files only from the Roadmap Epigenomics project is to check that all the files in the returned smaller hub object come from *Homo sapiens* and the hg19 genome

```
unique(epiFiles$species)
```

```
## [1] "Homo sapiens"
```

```
unique(epiFiles$genome)
```

```
## [1] "hg19"
```

Broadly, one can get an idea of the different files from this project looking at the sourcetype

```
table(epiFiles$sourcetype)
```

```
##
##    BED BigWig    GTF    tab    Zip
##   8298   9932      3      1     14
```

To get a more descriptive idea of these different files one can use:

```
sort(table(epiFiles$description), decreasing=TRUE)
```

```
##
##                         Bigwig File containing -log10(p-value) signal tracks from EpigenomeRoadMap Pro
##
##                         Bigwig File containing fold enrichment signal tracks from EpigenomeRoadMap Pro
##
##                            Narrow ChIP-seq peaks for consolidated epigenomes from EpigenomeRoadMap Pro
##
##                             Broad ChIP-seq peaks for consolidated epigenomes from EpigenomeRoadMap Pro
##
##                            Gapped ChIP-seq peaks for consolidated epigenomes from EpigenomeRoadMap Pro
##
##                                Narrow DNasePeaks for consolidated epigenomes from EpigenomeRoadMap Pro
##
##                                             15 state chromatin segmentations from EpigenomeRoadMap Pro
##
```

```
##      Broad domains on enrichment for DNase-seq for consolidated epigenomes from EpigenomeRoadMap Pro
##
##                                            RRBS fractional methylation calls from EpigenomeRoadMap Proj
##
##                              Whole genome bisulphite fractional methylation calls from EpigenomeRoadMap Pro
##
##                                      MeDIP/MRE(mCRF) fractional methylation calls from EpigenomeRoadMap Pro
##
## GencodeV10 gene/transcript coordinates and annotations corresponding to hg19 version of the human ge
##
##                                          RNA-seq read count matrix for intronic protein-coding RNA elem
##
##                                                         RNA-seq read counts matrix for ribosomal gene e
##
##                                                             RPKM expression matrix for ribosomal gene e
##
##                                                                       Metadata for EpigenomeRoadMap Pro
##
##                                                              RNA-seq read counts matrix for non-coding
##
##                                                         RNA-seq read counts matrix for protein coding e
##
##                                                         RNA-seq read counts matrix for protein coding g
##
##                                                              RNA-seq read counts matrix for ribosomal g
##
##                                                                  RPKM expression matrix for non-coding
##
##                                                             RPKM expression matrix for protein coding e
##
##                                                             RPKM expression matrix for protein coding g
##
##                                                                   RPKM expression matrix for ribosomal
##
```

The ‘metadata’ provided by the Roadmap Epigenomics Project is also available. Note that the information displayed about a hub with a single resource is quite different from the information displayed when the hub references more than one resource.

```
metadata.tab <- query(ah , c("EpigenomeRoadMap", "Metadata"))
metadata.tab
```

```
## AnnotationHub with 1 record
## # snapshotDate(): 2018-04-30
## # names(): AH41830
## # $dataprovider: BroadInstitute
## # $species: Homo sapiens
## # $rdataclass: data.frame
## # $rdatadateadded: 2015-05-11
## # $title: EID_metadata.tab
## # $description: Metadata for EpigenomeRoadMap Project
## # $taxonomyid: 9606
## # $genome: hg19
## # $sourcetype: tab
## # $sourceurl: http://egg2.wustl.edu/roadmap/data/byFileType/metadata/EID_metadata.tab
## # $sourcesize: 18035
## # $tags: c("EpigenomeRoadMap", "Metadata")
## # retrieve record with 'object[["AH41830"]]'
```

So far we have been exploring information about resources, without downloading the resource to a local cache and importing it into R. One can retrieve the resource using `[[` as indicated at the end of the show method

```
## downloading 0 resources

## loading from cache
##     '/Users/sdavis2//.AnnotationHub/47270'
```

```
metadata.tab <- ah[["AH41830"]]
```

```
## downloading 0 resources

## loading from cache
##     '/Users/sdavis2//.AnnotationHub/47270'
```

The metadata.tab file is returned as a *data.frame*. The first 6 rows of the first 5 columns are shown here:

```
metadata.tab[1:6, 1:5]
```

```
##    EID    GROUP   COLOR          MNEMONIC                                 STD_NAME
## 1 E001      ESC #924965             ESC.I3                              ES-I3 Cells
## 2 E002      ESC #924965            ESC.WA7                             ES-WA7 Cells
## 3 E003      ESC #924965             ESC.H1                                 H1 Cells
## 4 E004 ES-deriv #4178AE ESDR.H1.BMP4.MESO H1 BMP4 Derived Mesendoderm Cultured Cells
## 5 E005 ES-deriv #4178AE ESDR.H1.BMP4.TROP H1 BMP4 Derived Trophoblast Cultured Cells
## 6 E006 ES-deriv #4178AE       ESDR.H1.MSC          H1 Derived Mesenchymal Stem Cells
```

One can keep constructing different queries using multiple arguments to trim down these 18248 to get the files one wants. For example, to get the ChIP-Seq files for consolidated epigenomes, one could use

```
bpChipEpi <- query(ah , c("EpigenomeRoadMap", "broadPeak", "chip", "consolidated"))
```

To get all the bigWig signal files, one can query the hub using

```
allBigWigFiles <- query(ah, c("EpigenomeRoadMap", "BigWig"))
```

To access the 15 state chromatin segmentations, one can use

```
seg <- query(ah, c("EpigenomeRoadMap", "segmentations"))
```

If one is interested in getting all the files related to one sample

```
E126 <- query(ah , c("EpigenomeRoadMap", "E126", "H3K4ME2"))
E126
```

```
## AnnotationHub with 6 records
## # snapshotDate(): 2018-04-30
## # $dataprovider: BroadInstitute
## # $species: Homo sapiens
## # $rdataclass: BigWigFile, GRanges
## # additional mcols(): taxonomyid, genome, description, coordinate_1_based, maintainer,
## #   rdatadateadded, preparerclass, tags, rdatapath, sourceurl, sourcetype
## # retrieve records with, e.g., 'object[["AH29817"]]'
##
##             title
##   AH29817 | E126-H3K4me2.broadPeak.gz
##   AH30868 | E126-H3K4me2.narrowPeak.gz
##   AH31801 | E126-H3K4me2.gappedPeak.gz
##   AH32990 | E126-H3K4me2.fc.signal.bigwig
##   AH34022 | E126-H3K4me2.pval.signal.bigwig
##   AH40177 | E126-H3K4me2.imputed.pval.signal.bigwig
```

Hub resources can also be selected using `$`, `subset()`, and `display()`; see the main *AnnotationHub* vignette for additional detail.

Hub resources are imported as the appropriate *Bioconductor* object for use in further analysis. For example, peak files are returned as *GRanges* objects.

```
## require("rtracklayer")

## downloading 0 resources

## loading from cache
##     '/Users/sdavis2//.AnnotationHub/35257'
```

```
peaks <- E126[['AH29817']]
```

```
## downloading 0 resources

## loading from cache
##     '/Users/sdavis2//.AnnotationHub/35257'
```

```
seqinfo(peaks)
```

```
## Seqinfo object with 93 sequences (1 circular) from hg19 genome:
##   seqnames       seqlengths isCircular genome
##   chr1            249250621      FALSE   hg19
##   chr2            243199373      FALSE   hg19
##   chr3            198022430      FALSE   hg19
##   chr4            191154276      FALSE   hg19
##   chr5            180915260      FALSE   hg19
##   ...                   ...        ...    ...
##   chrUn_gl000245      36651      FALSE   hg19
##   chrUn_gl000246      38154      FALSE   hg19
##   chrUn_gl000247      36422      FALSE   hg19
##   chrUn_gl000248      39786      FALSE   hg19
##   chrUn_gl000249      38502      FALSE   hg19
```

BigWig files are returned as *BigWigFile* objects. A *BigWigFile* is a reference to a file on disk; the data in the file can be read in using `rtracklayer::import()`, perhaps querying these large files for particular genomic regions of interest as described on the help page `?import.bw`.

Each record inside *AnnotationHub* is associated with a unique identifier. Most *GRanges* objects returned by *AnnotationHub* contain the unique AnnotationHub identifier of the resource from which the *GRanges* is derived. This can come handy when working with the *GRanges* object for a while, and additional information about the object (e.g., the name of the file in the cache, or the original sourceurl for the data underlying the resource) that is being worked with.

```
metadata(peaks)
```

```
## $AnnotationHubName
## [1] "AH29817"
##
## $`File Name`
## [1] "E126-H3K4me2.broadPeak.gz"
##
## $`Data Source`
## [1] "http://egg2.wustl.edu/roadmap/data/byFileType/peaks/consolidated/broadPeak/E126-H3K4me2.broadPe
```

```
##
## $Provider
## [1] "BroadInstitute"
##
## $Organism
## [1] "Homo sapiens"
##
## $`Taxonomy ID`
## [1] 9606
```

```
ah[metadata(peaks)$AnnotationHubName]$sourceurl
```

```
## [1] "http://egg2.wustl.edu/roadmap/data/byFileType/peaks/consolidated/broadPeak/E126-H3K4me2.broadPea
```

Ensembl GTF and FASTA files for TxDb gene models and sequence queries

Bioconductor represents gene models using 'transcript' databases. These are available via packages such as *TxDb.Hsapiens.UCSC.hg38.knownGene* or can be constructed using functions such as *GenomicFeatures*::`makeTxDbFromBiomart()`.

AnnotationHub provides an easy way to work with gene models published by Ensembl. Let's see what Ensembl's Release-80 has in terms of data for pufferfish, *Takifugu rubripes*.

```
query(ah, c("Takifugu", "release-80"))
```

```
## AnnotationHub with 7 records
## # snapshotDate(): 2018-04-30
## # $dataprovider: Ensembl
## # $species: Takifugu rubripes
## # $rdataclass: FaFile, GRanges
## # additional mcols(): taxonomyid, genome, description, coordinate_1_based, maintainer,
## #   rdatadateadded, preparerclass, tags, rdatapath, sourceurl, sourcetype
## # retrieve records with, e.g., 'object[["AH47101"]]'
##
##             title
##   AH47101 | Takifugu_rubripes.FUGU4.80.gtf
##   AH47475 | Takifugu_rubripes.FUGU4.cdna.all.fa
##   AH47476 | Takifugu_rubripes.FUGU4.dna_rm.toplevel.fa
##   AH47477 | Takifugu_rubripes.FUGU4.dna_sm.toplevel.fa
##   AH47478 | Takifugu_rubripes.FUGU4.dna.toplevel.fa
##   AH47479 | Takifugu_rubripes.FUGU4.ncrna.fa
##   AH47480 | Takifugu_rubripes.FUGU4.pep.all.fa
```

We see that there is a GTF file descrbing gene models, as well as various DNA sequences. Let's retrieve the GTF and top-level DNA sequence files. The GTF file is imported as a *GRanges* instance, the DNA sequence as a compressed, indexed Fasta file

```
gtf <- ah[["AH47101"]]
```

```
## downloading 0 resources
```

```
## loading from cache
##     '/Users/sdavis2//.AnnotationHub/52579'
```

```
dna <- ah[["AH47477"]]
```

```
## downloading 0 resources
## loading from cache
##     '/Users/sdavis2//.AnnotationHub/53323'
##     '/Users/sdavis2//.AnnotationHub/53324'
```

```
head(gtf, 3)
```

```
## GRanges object with 3 ranges and 19 metadata columns:
##         seqnames      ranges strand |   source       type     score     phase            gene_id
##            <Rle>   <IRanges>  <Rle> | <factor>   <factor> <numeric> <integer>        <character>
##   [1] scaffold_1 10422-11354      - |  ensembl       gene      <NA>      <NA> ENSTRUG00000003702
##   [2] scaffold_1 10422-11354      - |  ensembl transcript      <NA>      <NA> ENSTRUG00000003702
##   [3] scaffold_1 10422-11354      - |  ensembl       exon      <NA>      <NA> ENSTRUG00000003702
##       gene_version gene_source   gene_biotype      transcript_id transcript_version
##          <numeric> <character>    <character>        <character>          <numeric>
##   [1]            1     ensembl protein_coding               <NA>               <NA>
##   [2]            1     ensembl protein_coding ENSTRUT00000008740                  1
##   [3]            1     ensembl protein_coding ENSTRUT00000008740                  1
##       transcript_source transcript_biotype exon_number            exon_id exon_version  protein_id
##             <character>        <character>   <numeric>        <character>    <numeric> <character>
##   [1]              <NA>               <NA>        <NA>               <NA>         <NA>        <NA>
##   [2]           ensembl     protein_coding        <NA>               <NA>         <NA>        <NA>
##   [3]           ensembl     protein_coding           1 ENSTRUE00000055472            1        <NA>
##       protein_version   gene_name transcript_name
##             <numeric> <character>     <character>
##   [1]            <NA>        <NA>            <NA>
##   [2]            <NA>        <NA>            <NA>
##   [3]            <NA>        <NA>            <NA>
##   -------
##   seqinfo: 2056 sequences (1 circular) from FUGU4 genome; no seqlengths
```

```
dna
```

```
## class: FaFile
## path: /Users/sdavis2//.AnnotationHub/53323
## index: /Users/sdavis2//.AnnotationHub/53324
## isOpen: FALSE
## yieldSize: NA
```

```
head(seqlevels(dna))
```

```
## [1] "scaffold_1" "scaffold_2" "scaffold_3" "scaffold_4" "scaffold_5" "scaffold_6"
```

Let's identify the 25 longest DNA sequences, and keep just the annotations on these scaffolds.

```
keep <- names(tail(sort(seqlengths(dna)), 25))
gtf_subset <- gtf[seqnames(gtf) %in% keep]
```

It is trivial to make a TxDb instance of this subset (or of the entire gtf)

```
library(GenomicFeatures)         # for makeTxDbFromGRanges
txdb <- makeTxDbFromGRanges(gtf_subset)
```

```
## Warning in .get_cds_IDX(type, phase): The "phase" metadata column contains non-NA values for feature
##   information was ignored.
```

and to use that in conjunction with the DNA sequences, e.g., to find exon sequences of all annotated genes.

```
library(Rsamtools)               # for getSeq,FaFile-method
exons <- exons(txdb)
```

```r
length(exons)
```

```
## [1] 66219
```

```r
getSeq(dna, exons)
```

```
##   A DNAStringSet instance of length 66219
##         width seq                                                           names
##     [1]    72 ATGGCCTATCAGTTGTACAGGAATACCACTC...CCTGCAGGAGAGTCTGGACGAGCTTATCCAG scaffold_1
##     [2]   105 ACTCAGCAGATCACCCCTCAGCTGGCTCTCC...TAATCGTGTCCGCAACCGTGTGAACTTCAGG scaffold_1
##     [3]   156 GGTTCTCTCAACACCTACCGGTTCTGTGACA...GAGCGAATCCATGCAAAACAAACTGGATAAA scaffold_1
##     [4]    88 CAAACCAATCTCCTCGCTGTCTCTTCTCGTT...CATCAGCCAGAGGGACGGATCATCTCAGGTT scaffold_1
##     [5]   271 AGACGAGATGAGTGAGGACGCATTCAACGCC...AACACAGTGTGGAGACTTCAGAGGACGCCAC scaffold_1
##     ...   ... ...
## [66215]    67 ACGACTGGATGACAACATCAGGACCGTGGTA...TCAGACCAATGTGGGTCAGGATGGCAGACAG scaffold_9
## [66216]    50 TCTTTGGCTAATATTGACGATGTGGTAAACAAGATTCGTCTGAAGATTCG                scaffold_9
## [66217]    81 GTATTTCCCAGCCAAGACCCGCTGGACAGGG...ATACATCAACACACTGTTTCCCACCGAGCAG scaffold_9
## [66218]    87 ATGATGGAGGATGAAGAATTTGAATTTGCGG...GACCCCAGAGGTGCAGCTAGCAATTGAACAG scaffold_9
## [66219]   213 GACGACATCCTCGTGTGGGGCCGCTCTAGGG...GACAGCTGCTGTTCGCCTGTTTCCCCCCCCC scaffold_9
```

There is a one-to-one mapping between the genomic ranges contained in `exons` and the DNA sequences returned by `getSeq()`.

Some difficulties arise when working with this partly assembled genome that require more advanced GenomicRanges skills, see the *GenomicRanges* vignettes, especially "*GenomicRanges* HOWTOs" and "An Introduction to *GenomicRanges*".

liftOver to map between genome builds

Suppose we wanted to lift features from one genome build to another, e.g., because annotations were generated for hg19 but our experimental analysis used hg18. We know that UCSC provides 'liftover' files for mapping between genome builds.

In this example, we will take our broad Peak *GRanges* from E126 which comes from the 'hg19' genome, and lift over these features to their 'hg38' coordinates.

```r
chainfiles <- query(ah , c("hg38", "hg19", "chainfile"))
chainfiles
```

```
## AnnotationHub with 2 records
## # snapshotDate(): 2018-04-30
## # $dataprovider: UCSC
## # $species: Homo sapiens
## # $rdataclass: ChainFile
## # additional mcols(): taxonomyid, genome, description, coordinate_1_based, maintainer,
## #   rdatadateadded, preparerclass, tags, rdatapath, sourceurl, sourcetype
## # retrieve records with, e.g., 'object[["AH14108"]]'
##
##             title
##   AH14108 | hg38ToHg19.over.chain.gz
##   AH14150 | hg19ToHg38.over.chain.gz
```

We are interested in the file that lifts over features from hg19 to hg38 so lets download that using

```
## downloading 0 resources
```

```
## loading from cache
##     '/Users/sdavis2//.AnnotationHub/18245'
```

```
chain <- chainfiles[['AH14150']]
```

```
## downloading 0 resources
```

```
## loading from cache
##     '/Users/sdavis2//.AnnotationHub/18245'
```

```
chain
```

```
## Chain of length 25
## names(25): chr1 chr2 chr3 chr4 chr5 chr6 chr7 chr8 ... chr18 chr19 chr20 chr21 chr22 chrX chrY chrM
```

Perform the liftOver operation using `rtracklayer::liftOver()`:

```
library(rtracklayer)
gr38 <- liftOver(peaks, chain)
```

This returns a *GRangeslist*; update the genome of the result to get the final result

```
genome(gr38) <- "hg38"
gr38
```

```
## GRangesList object of length 153266:
## [[1]]
## GRanges object with 1 range and 5 metadata columns:
##       seqnames            ranges strand |        name     score signalValue    pValue    qValue
##          <Rle>         <IRanges>  <Rle> | <character> <numeric>   <numeric> <numeric> <numeric>
##   [1]     chr1 28667912-28670147      * |      Rank_1       189    10.55845  22.01316  18.99911
##
## [[2]]
## GRanges object with 1 range and 5 metadata columns:
##       seqnames            ranges strand |   name score signalValue   pValue   qValue
##   [1]     chr4 54090990-54092984      * | Rank_2   188     8.11483 21.80441 18.80662
##
## [[3]]
## GRanges object with 1 range and 5 metadata columns:
##       seqnames            ranges strand |   name score signalValue   pValue   qValue
##   [1]    chr14 75293392-75296621      * | Rank_3   180     8.89834 20.97714 18.02816
##
## ...
## <153263 more elements>
## -------
## seqinfo: 23 sequences from hg38 genome; no seqlengths
```

Working with dbSNP Variants

One may also be interested in working with common germline variants with evidence of medical interest. This information is available at NCBI.

Query the dbDNP files in the hub:

This returns a *VcfFile* which can be read in using `r Biocpkg("VariantAnnotation")`; because VCF files can be large, `readVcf()` supports several strategies for importing only relevant parts of the file (e.g., particular genomic locations, particular features of the variants), see `?readVcf` for additional information.

```
variants <- readVcf(vcf, genome="hg19")
variants
```

```
## class: CollapsedVCF
## dim: 111138 0
## rowRanges(vcf):
##   GRanges with 5 metadata columns: paramRangeID, REF, ALT, QUAL, FILTER
## info(vcf):
##   DataFrame with 58 columns: RS, RSPOS, RV, VP, GENEINFO, dbSNPBuildID, SAO, SSR, WGT, VC, PM, T...
## info(header(vcf)):
##                 Number Type    Description
##    RS           1      Integer dbSNP ID (i.e. rs number)
##    RSPOS        1      Integer Chr position reported in dbSNP
##    RV           0      Flag    RS orientation is reversed
##    VP           1      String  Variation Property.  Documentation is at ftp://ftp.ncbi.nlm.nih.g...
##    GENEINFO     1      String  Pairs each of gene symbol:gene id.  The gene symbol and id are de...
##    dbSNPBuildID 1      Integer First dbSNP Build for RS
##    SAO          1      Integer Variant Allele Origin: 0 - unspecified, 1 - Germline, 2 - Somatic...
##    SSR          1      Integer Variant Suspect Reason Codes (may be more than one value added to...
##    WGT          1      Integer Weight, 00 - unmapped, 1 - weight 1, 2 - weight 2, 3 - weight 3 o...
##    VC           1      String  Variation Class
##    PM           0      Flag    Variant is Precious(Clinical,Pubmed Cited)
##    TPA          0      Flag    Provisional Third Party Annotation(TPA) (currently rs from PHARMG...
##    PMC          0      Flag    Links exist to PubMed Central article
##    S3D          0      Flag    Has 3D structure - SNP3D table
##    SLO          0      Flag    Has SubmitterLinkOut - From SNP->SubSNP->Batch.link_out
##    NSF          0      Flag    Has non-synonymous frameshift A coding region variation where one...
##    NSM          0      Flag    Has non-synonymous missense A coding region variation where one a...
##    NSN          0      Flag    Has non-synonymous nonsense A coding region variation where one a...
##    REF          0      Flag    Has reference A coding region variation where one allele in the s...
##    SYN          0      Flag    Has synonymous A coding region variation where one allele in the ...
##    U3           0      Flag    In 3' UTR Location is in an untranslated region (UTR). FxnCode = 53
##    U5           0      Flag    In 5' UTR Location is in an untranslated region (UTR). FxnCode = 55
##    ASS          0      Flag    In acceptor splice site FxnCode = 73
##    DSS          0      Flag    In donor splice-site FxnCode = 75
##    INT          0      Flag    In Intron FxnCode = 6
##    R3           0      Flag    In 3' gene region FxnCode = 13
##    R5           0      Flag    In 5' gene region FxnCode = 15
##    OTH          0      Flag    Has other variant with exactly the same set of mapped positions o...
##    CFL          0      Flag    Has Assembly conflict. This is for weight 1 and 2 variant that ma...
##    ASP          0      Flag    Is Assembly specific. This is set if the variant only maps to one...
##    MUT          0      Flag    Is mutation (journal citation, explicit fact): a low frequency va...
##    VLD          0      Flag    Is Validated.  This bit is set if the variant has 2+ minor allele...
##    G5A          0      Flag    >5% minor allele frequency in each and all populations
##    G5           0      Flag    >5% minor allele frequency in 1+ populations
##    HD           0      Flag    Marker is on high density genotyping kit (50K density or greater)...
##    GNO          0      Flag    Genotypes available. The variant has individual genotype (in SubI...
##    KGPhase1     0      Flag    1000 Genome phase 1 (incl. June Interim phase 1)
##    KGPhase3     0      Flag    1000 Genome phase 3
##    CDA          0      Flag    Variation is interrogated in a clinical diagnostic assay
##    LSD          0      Flag    Submitted from a locus-specific database
##    MTP          0      Flag    Microattribution/third-party annotation(TPA:GWAS,PAGE)
##    OM           0      Flag    Has OMIM/OMIA
##    NOC          0      Flag    Contig allele not present in variant allele list. The reference s...
##    WTD          0      Flag    Is Withdrawn by submitter If one member ss is withdrawn by submit...
##    NOV          0      Flag    Rs cluster has non-overlapping allele sets. True when rs set has ...
##    CAF          .      String  An ordered, comma delimited list of allele frequencies based on 1...
##    COMMON       1      Integer RS is a common SNP.  A common SNP is one that has at least one 10...
##    CLNHGVS      .      String  Variant names from HGVS.    The order of these variants correspon...
##    CLNALLE      .      Integer Variant alleles from REF or ALT columns.  0 is REF, 1 is the firs...
##    CLNSRC       .      String  Variant Clinical Chanels
##    CLNORIGIN    .      String  Allele Origin. One or more of the following values may be added: ...
##    CLNSRCID     .      String  Variant Clinical Channel IDs
##    CLNSIG       .      String  Variant Clinical Significance, 0 - Uncertain significance, 1 - no...
##    CLNDSDB      .      String  Variant disease database name
##    CLNDSDBID    .      String  Variant disease database ID
##    CLNDBN       .      String  Variant disease name
##    CLNREVSTAT   .      String  no_assertion - No assertion provided, no_criteria - No assertion ...
##    CLNACC       .      String  Variant Accession and Versions
## geno(vcf):
##   SimpleList of length 0:
```

`rowRanges()` returns information from the CHROM, POS and ID fields of the VCF file, represented as a *GRanges* instance

```
rowRanges(variants)
```

```
## GRanges object with 111138 ranges and 5 metadata columns:
##                seqnames    ranges strand | paramRangeID            REF                ALT      QUAL
##                   <Rle> <IRanges>  <Rle> |     <factor> <DNAStringSet> <DNAStringSetList> <numeric>
##   rs786201005         1   1014143      * |         <NA>              C                  T      <NA>
##   rs672601345         1   1014316      * |         <NA>              C                 CG      <NA>
##   rs672601312         1   1014359      * |         <NA>              G                  T      <NA>
##   rs115173026         1   1020217      * |         <NA>              G                  T      <NA>
##   rs201073369         1   1020239      * |         <NA>              G                  C      <NA>
##           ...       ...       ...    ... .          ...            ...                ...       ...
##   rs527236200        MT     15943      * |         <NA>              T                  C      <NA>
##   rs118203890        MT     15950      * |         <NA>              G                  A      <NA>
##   rs199474700        MT     15965      * |         <NA>              A                  G      <NA>
##   rs199474701        MT     15967      * |         <NA>              G                  A      <NA>
##   rs199474699        MT     15990      * |         <NA>              C                  T      <NA>
##                     FILTER
##                <character>
##   rs786201005            .
##   rs672601345            .
##   rs672601312            .
##   rs115173026            .
##   rs201073369            .
##           ...          ...
##   rs527236200            .
##   rs118203890            .
##   rs199474700            .
##   rs199474701            .
##   rs199474699            .
##   -------
##   seqinfo: 25 sequences from hg19 genome; no seqlengths
```

Note that the broadPeaks files follow the UCSC chromosome naming convention, and the vcf data follows the NCBI style of chromosome naming convention. To bring these ranges in the same chromosome naming convention (ie UCSC), we would use

```
seqlevelsStyle(variants) <-seqlevelsStyle(peaks)
```

And then finally to find which variants overlap these broadPeaks we would use:

```
overlap <- findOverlaps(variants, peaks)
overlap
```

```
## Hits object with 10904 hits and 0 metadata columns:
##           queryHits subjectHits
##           <integer>   <integer>
##       [1]        35       20333
##       [2]        36       20333
##       [3]        37       20333
##       [4]        38       20333
##       [5]        41        7733
##       ...       ...         ...
##   [10900]    110761       21565
##   [10901]    110762       21565
##   [10902]    110763       21565
##   [10903]    110764       21565
##   [10904]    110765       21565
##   -------
##   queryLength: 111138 / subjectLength: 153266
```

Some insight into how these results can be interpretted comes from looking a particular peak, e.g., the 3852nd peak

```
idx <- subjectHits(overlap) == 3852
overlap[idx]
```

```
## Hits object with 39 hits and 0 metadata columns:
##        queryHits subjectHits
##        <integer>   <integer>
##    [1]    102896        3852
##    [2]    102897        3852
##    [3]    102898        3852
##    [4]    102899        3852
##    [5]    102900        3852
##    ...       ...         ...
##   [35]    102930        3852
##   [36]    102931        3852
##   [37]    102932        3852
##   [38]    102933        3852
##   [39]    102934        3852
##   -------
##   queryLength: 111138 / subjectLength: 153266
```

There are three variants overlapping this peak; the coordinates of the peak and the overlapping variants are

```
peaks[3852]
```

```
## GRanges object with 1 range and 5 metadata columns:
##       seqnames            ranges strand |        name     score signalValue    pValue    qValue
##          <Rle>         <IRanges>  <Rle> | <character> <numeric>   <numeric> <numeric> <numeric>
##   [1]    chr22 50622494-50626143      * |   Rank_3852        79     6.06768  10.18943   7.99818
##   -------
##   seqinfo: 93 sequences (1 circular) from hg19 genome
```

```r
rowRanges(variants)[queryHits(overlap[idx])]
```

```
## GRanges object with 39 ranges and 5 metadata columns:
##               seqnames    ranges strand | paramRangeID            REF                ALT      QUAL
##                  <Rle> <IRanges>  <Rle> |     <factor> <DNAStringSet> <DNAStringSetList> <numeric>
##     rs6151429    chr22  50625049      * |         <NA>              T                  C      <NA>
##     rs6151428    chr22  50625182      * |         <NA>              C                A,T      <NA>
##   rs774153480    chr22  50625182      * |         <NA>              C           CG,CGGGG      <NA>
##   rs199476388    chr22  50625204      * |         <NA>              A                C,G      <NA>
##    rs74315482    chr22  50625213      * |         <NA>              G                  A      <NA>
##           ...      ...       ...    ... .          ...            ...                ...       ...
##   rs199476369    chr22  50625936      * |         <NA>              C                  G      <NA>
##     rs2071421    chr22  50625988      * |         <NA>              T                  C      <NA>
##    rs74315475    chr22  50626033      * |         <NA>              T                  A      <NA>
##   rs398123419    chr22  50626052      * |         <NA>              C                  A      <NA>
##   rs398123418    chr22  50626057      * |         <NA>              G                  A      <NA>
##                    FILTER
##               <character>
##     rs6151429           .
##     rs6151428           .
##   rs774153480           .
##   rs199476388           .
##    rs74315482           .
##           ...         ...
##   rs199476369           .
##     rs2071421           .
##    rs74315475           .
##   rs398123419           .
##   rs398123418           .
##   -------
##   seqinfo: 25 sequences from hg19 genome; no seqlengths
```

sessionInfo

```r
sessionInfo()
```

```
## R version 3.5.0 RC (2018-04-16 r74624)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Sierra 10.12.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats4    parallel  stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] BSgenome.Hsapiens.UCSC.hg19_1.4.0 BSgenome_1.47.5
##  [3] rtracklayer_1.39.13              BiocStyle_2.7.9
##  [5] VariantAnnotation_1.25.13        SummarizedExperiment_1.9.18
##  [7] DelayedArray_0.5.35              BiocParallel_1.13.3
##  [9] matrixStats_0.53.1               Rsamtools_1.31.3
## [11] Biostrings_2.47.12               XVector_0.19.9
## [13] GenomicFeatures_1.31.10          AnnotationDbi_1.41.6
## [15] Biobase_2.39.2                   GenomicRanges_1.31.23
## [17] GenomeInfoDb_1.15.5              IRanges_2.13.29
## [19] S4Vectors_0.17.43                AnnotationHub_2.12.0
## [21] BiocGenerics_0.25.3
##
## loaded via a namespace (and not attached):
##  [1] progress_1.1.2               lattice_0.20-35              htmltools_0.3.6
##  [4] yaml_2.1.19                  interactiveDisplayBase_1.18.0 blob_1.1.1
##  [7] XML_3.98-1.11                later_0.7.2                  DBI_0.8
## [10] bit64_0.9-7                  GenomeInfoDbData_1.1.0       stringr_1.3.1
## [13] zlibbioc_1.25.0              memoise_1.1.0                evaluate_0.10.1
## [16] knitr_1.20                   biomaRt_2.35.13              httpuv_1.4.3
## [19] BiocInstaller_1.30.0         curl_3.2                     Rcpp_0.12.16
## [22] xtable_1.8-2                 promises_1.0.1               backports_1.1.2
## [25] mime_0.5                     bit_1.1-12                   digest_0.6.15
## [28] stringi_1.2.2                shiny_1.0.5                  rprojroot_1.3-2
## [31] grid_3.5.0                   tools_3.5.0                  bitops_1.0-6
## [34] magrittr_1.5                 RCurl_1.95-4.10              RSQLite_2.1.0
## [37] pkgconfig_2.0.1              Matrix_1.2-14                prettyunits_1.0.2
## [40] assertthat_0.2.0             rmarkdown_1.9                httr_1.3.1
## [43] R6_2.2.2                     GenomicAlignments_1.15.14    compiler_3.5.0
```