gwascat: structuring and querying the NHGRI GWAS catalog

VJ Carey*

May 2, 2014

Contents

*Generous support of Robert Gentleman and the Computational Biology Group of Genentech, Inc. is gratefully acknowledged

1 Introduction

NHGRI maintains and routinely updates a database of selected genome-wide association studies. This document describes R/Bioconductor facilities for working with contents of this database.

1.1 Installation

The package can be installed using Bioconductor's *BiocInstaller* package, with the sequence

```
library(BiocInstaller)
biocLite("gwascat")
```

1.2 Attachment and access to documentation

Once the package has been installed, use `library(gwascat)` to obtain interactive access to all the facilities. After executing this command, use `help(package="gwascat")` to obtain an overview. The current version of this vignette can always be accessed at www.bioconductor.org, or by suitably navigating the web pages generated with `help.start()`.

1.3 Illustrations: computing

Available functions are:

```
> library(gwascat)
> objects("package:gwascat")
```

```
 [1] "DataFrame"          "bindcadd_snv"       "chklocs"
 [4] "elementMetadata"    "g17SM"              "getRsids"
 [7] "getTraits"          "gg17N"              "gw6.rs_17"
[10] "gwastagger"         "gwcex2gviz"         "gwdf_2012_03_20"
[13] "gwdf_2012_09_22"    "gwrngs"             "impute.snps"
[16] "locon6"             "locs4trait"         "low17"
[19] "makeCurrentGwascat" "match"              "mcols"
[22] "mcols<-"            "obo2graphNEL"       "overlapsAny"
[25] "ranges"             "riskyAlleleCount"   "rules_6.0_1kg_17"
[28] "seqnames"           "subsetByChromosome" "subsetByTraits"
[31] "topTraits"          "traitsManh"         "values"
```

The extended GRanges instance with all SNP-disease associations is:

```
> gwrngs
```

```
gwasloc instance with 14719 records and 35 attributes per record.
Extracted:  2013-12-03
Excerpt:
GRanges with 5 ranges and 3 metadata columns:
      seqnames                ranges strand |
         <Rle>             <IRanges>  <Rle> |
  [1]    chr10 [ 96405502,  96405502]      * |
  [2]    chr11 [  2024544,   2024544]      * |
  [3]     chr4 [137526584, 137526584]      * |
  [4]    chr22 [ 19691502,  19691502]      * |
  [5]    chr11 [ 92784203,  92784203]      * |
                          Disease.Trait        SNPs   p.Value
                            <character> <character> <numeric>
  [1]          Warfarin maintenance dose  rs12777823     5e-12
  [2] DNA methylation (parent-of-origin)   rs4930103     5e-16
  [3] DNA methylation (parent-of-origin)  rs10012307     2e-08
  [4] DNA methylation (parent-of-origin)   rs4819833     6e-08
  [5] DNA methylation (parent-of-origin)   rs7931462     2e-09
  ---
  seqlengths:
       chr1      chr2      chr3      chr4 ...     chr21     chr22      chrX
   249250621 243199373 198022430 191154276 ...  48129895  51304566 155270560
```

To determine the most frequently occurring traits:

```
> topTraits(gwrngs)
```

```
Obesity-related traits      IgG glycosylation                 Height
                   951                    699                    522
       Crohn's disease        Type 2 diabetes     Multiple sclerosis
                   210                    210                    177
         Breast cancer     Ulcerative colitis        Body mass index
                   156                    147                    141
Coronary heart disease
                   140
```

For a given trait, obtain a GRanges with all recorded associations; here only three associations are shown:

```
> subsetByTraits(gwrngs, tr="LDL cholesterol")[1:3]
```

```
gwasloc instance with 3 records and 35 attributes per record.
Extracted:  2013-12-03
Excerpt:
```

```
GRanges with 3 ranges and 3 metadata columns:
      seqnames               ranges strand |   Disease.Trait        SNPs
         <Rle>            <IRanges>  <Rle> |     <character> <character>
  [1]     chr1 [55625548, 55625548]      * | LDL cholesterol  rs17111684
  [2]     chr2 [21241505, 21241505]      * | LDL cholesterol  rs12713956
  [3]     chr2 [44074431, 44074431]      * | LDL cholesterol   rs4245791
        p.Value
      <numeric>
  [1]     2e-17
  [2]     4e-08
  [3]     1e-09
  ---
  seqlengths:
        chr1      chr2      chr3      chr4 ...     chr21     chr22      chrX
   249250621 243199373 198022430 191154276 ...  48129895  51304566 155270560
```

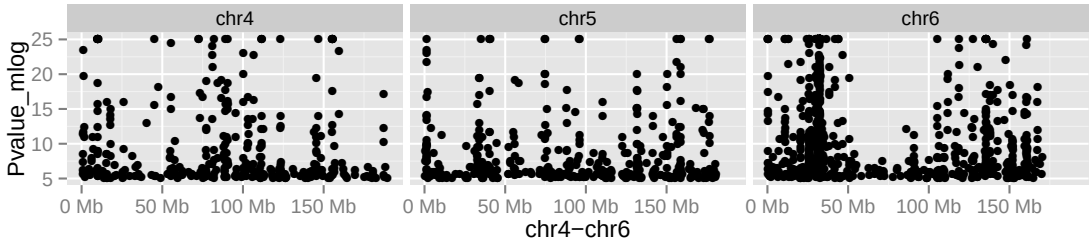
2 Some visualizations

2.1 Basic Manhattan plot

A basic Manhattan plot is easily constructed with the ggbio package facilities. Here we confine attention to chromosomes 4:6. First, we create a version of the catalog with $-log_{10}p$ truncated at a maximum value of 25.

```
> gwtrunc = gwrngs
> mlpv = mcols(gwrngs)$Pvalue_mlog
> mlpv = ifelse(mlpv > 25, 25, mlpv)
> mcols(gwtrunc)$Pvalue_mlog = mlpv
> gwlit = gwtrunc[ which(as.character(seqnames(gwtrunc)) %in% c("chr4", "chr5", "chr6
> library(ggbio)
> mlpv = mcols(gwlit)$Pvalue_mlog
> mlpv = ifelse(mlpv > 25, 25, mlpv)
> mcols(gwlit)$Pvalue_mlog = mlpv

> methods:::bind_activation(FALSE)
> autoplot(gwlit, geom="point", aes(y=Pvalue_mlog), xlab="chr4-chr6")
```



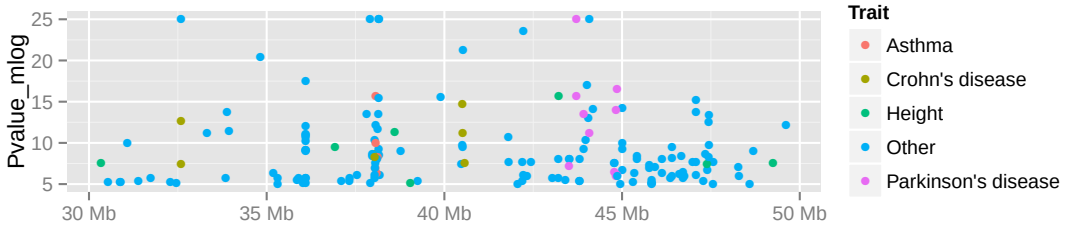
2.2 Annotated Manhattan plot

A simple call permits visualization of GWAS results for a small number of traits. Note the defaults in this call.

```
> args(traitsManh)

function (gwr, selr = GRanges(seqnames = "chr17", IRanges(3e+07,
    5e+07)), traits = c("Asthma", "Parkinson's disease", "Height",
    "Crohn's disease"), truncmlp = 25, ...)
NULL

> traitsManh(gwtrunc)
```



2.3 Integrative view of potential genetic determinants

The following chunk uses GFF3 data on eQTL and related phenomena distributed at the GBrowse instance at eqtl.uchicago.edu. A request for all information at 43-45 Mb was made on 2 June 2012, yielding the GFF3 referenced below. Of interest are locations and scores of genetic associations with DNaseI hypersensitivity (scores identifying dsQTL, see Degner et al 2012).

```
> gffpath = system.file("gff3/chr17_43000000_45000000.gff3", package="gwascat")
> library(rtracklayer)
> c17tg = import(gffpath, asRangedData=FALSE)
```

We make a Gviz DataTrack of the dsQTL scores.

```
> c17td = c17tg[ which(mcols(c17tg)$type == "Degner_dsQTL") ]
> library(Gviz)
> dsqs = DataTrack( c17td, chrom="chr17", genome="hg19", data="score",
+   name="dsQTL")
```

We start the construction of the graph here.

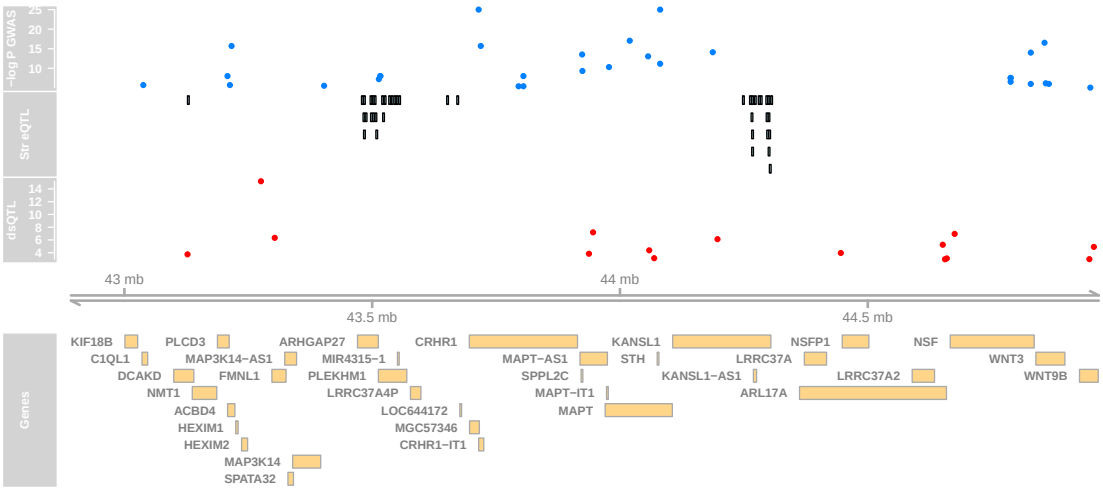
```
> g2 = GRanges(seqnames="chr17", IRanges(start=4.3e7, width=2e6))
> basic = gwcex2gviz(contextGR=g2, plot.it=FALSE)
```

We also collect locations of eQTL in the Stranger 2007 multipopulation eQTL study.

```
> c17ts = c17tg[ which(mcols(c17tg)$type == "Stranger_eqtl") ]
> eqloc = AnnotationTrack(c17ts,  chrom="chr17", genome="hg19", name="Str eQTL")
> displayPars(eqloc)$col = "black"
> displayPars(dsqs)$col = "red"
> integ = list(basic[[1]], eqloc, dsqs, basic[[2]], basic[[3]])
```

Now use Gviz.

```
> plotTracks(integ)
```



3 SNP sets and trait sets

3.1 SNPs by name

We can regard the content of a SNP chip as a set of SNP, referenced by name. The pd.genomewidesnp.6 package describes the Affymetrix SNP 6.0 chip. We can determine which traits are associated with loci interrogated by the chip as follows. We work with a subset of the 1 million loci for illustration.

The `locon6` data frame has information on 10000 probes, acquired through the following code (not executed here to reduce dependence on the pd.genomewidesnp.6 package, which is very large.

```
> library(pd.genomewidesnp.6)
> con = pd.genomewidesnp.6@getdb()
> locon6 = dbGetQuery(con,
+    "select dbsnp_rs_id, chrom, physical_pos from featureSet limit 10000")
```

Instead use the serialized information:

```
> data(locon6)
> rson6 = as.character(locon6[[1]])
> rson6[1:5]

[1] "rs2887286"  "rs1496555"  "rs41477744" "rs3890745"  "rs10492936"
```

We subset the GWAS ranges structure with rsids that are common to both the chip and the GWAS catalog. We then tabulate the diseases associated with the common loci.

```
> intr = gwrngs[ intersect(getRsids(gwrngs), rson6) ]
> sort(table(getTraits(intr)), decreasing=TRUE)[1:10]

                                        Height
                                             3
                          Select biomarker traits
                                             3
                                   Dental caries
                                             2
                                IgG glycosylation
                                             2
Immune reponse to smallpox (secreted IFN-alpha)
                                             2
                                Metabolic traits
                                             2
                               Ulcerative colitis
                                             2
               Age-related macular degeneration
                                             1
                                     Aging traits
                                             1
                               Alcohol dependence
                                             1
```

3.2 Traits by genomic location

We will assemble genomic coordinates for SNP on the Affymetrix 6.0 chip and show the effects of identifying the trait-associated loci with regions of width 1000bp instead of 1bp.

The following code retrieves coordinates for SNP interrogated on 10000 probes (to save time) on the 6.0 chip, and stores the results in a GRanges instance.

```
> gr6.0 = GRanges(seqnames=ifelse(is.na(locon6$chrom),0,locon6$chrom),
+        IRanges(ifelse(is.na(locon6$phys),1,locon6$phys), width=1))
> mcols(gr6.0)$rsid = as.character(locon6$dbsnp_rs_id)
> seqlevels(gr6.0) = paste("chr", seqlevels(gr6.0), sep="")
```

Here we compute overlaps with both the raw disease-associated locus addresses, and with the locus address ± 500bp.

```
> ag = function(x) as(x, "GRanges")
> ovraw = suppressWarnings(subsetByOverlaps(ag(gwrngs), gr6.0))
> length(ovraw)

[1] 89

> ovaug = suppressWarnings(subsetByOverlaps(ag(gwrngs+500), gr6.0))
> length(ovaug)

[1] 131
```

To acquire the subset of the catalog to which 6.0 probes are within 500bp, use:

```
> rawrs = mcols(ovraw)$SNPs
> augrs = mcols(ovaug)$SNPs
> gwrngs[augrs]

gwasloc instance with 131 records and 35 attributes per record.
Extracted:  2013-12-03
Excerpt:
GRanges with 5 ranges and 3 metadata columns:
      seqnames                 ranges strand |
         <Rle>              <IRanges>  <Rle> |
  [1]    chr10 [ 57549296,  57549296]      * |
  [2]     chr1 [109392837, 109392837]      * |
  [3]    chr10 [124214448, 124214448]      * |
  [4]     chr1 [209712889, 209712889]      * |
  [5]     chr1 [150940625, 150940625]      * |
                                                                Disease.Trait
```

```
                                                                    <character>
  [1]               Post-traumatic stress disorder (asjusted for relatedness)
  [2] Adverse response to chemotherapy (neutropenia/leucopenia) (carboplatin)
  [3]                                          Age-related macular degeneration
  [4]                                             Amyotrophic lateral sclerosis
  [5]                                          Rhegmatogenous retinal detachment
            SNPs   p.Value
      <character> <numeric>
  [1]  rs16907840     8e-06
  [2]   rs1277203     9e-06
  [3]  rs10490924     8e-27
  [4]   rs6703183     3e-08
  [5]    rs267738     1e-07
  ---
  seqlengths:
        chr1      chr2      chr3      chr4 ...     chr21     chr22      chrX
   249250621 243199373 198022430 191154276 ...  48129895  51304566 155270560
```

Relaxing the intersection criterion in this limited case leads to a larger set of traits.

```
> setdiff( getTraits(gwrngs[augrs]), getTraits(gwrngs[rawrs]) )

 [1] "Adverse response to chemotherapy (neutropenia/leucopenia) (carboplatin)"
 [2] "Body mass index"
 [3] "Response to taxane treatment (docetaxel)"
 [4] "Neuroblastoma"
 [5] "Response to amphetamines"
 [6] "Lentiform nucleus volume "
 [7] "Venous thromboembolism"
 [8] "Fasting glucose-related traits (interaction with BMI)"
 [9] "Response to angiotensin II receptor blocker therapy"
[10] "Capecitabine sensitivity"
[11] "Response to hepatitis C treatment"
[12] "Response to antidepressant treatment"
[13] "Phospholipid levels (plasma)"
[14] "Endometrial cancer"
[15] "MRI atrophy measures"
[16] "Self-rated health"
[17] "Neonatal lupus"
[18] "Crohn's disease"
[19] "Optic disc size (cup)"
[20] "Response to statin therapy"
[21] "Tanning"
```

```
[22] "Obesity"
[23] "Osteonecrosis of the jaw"
[24] "Hip geometry"
[25] "Parkinson's disease"
```

4 Counting alleles associated with traits

We can use `riskyAlleleCount` to count risky alleles enumerated in the GWAS catalog. This particular function assumes that we have genotyped at the catalogued loci. Below we will discuss how to impute from non-catalogued loci to those enumerated in the catalog.

```
> data(gg17N) # translated from GGdata chr 17 calls using ABmat2nuc
> gg17N[1:5,1:5]

        rs6565733 rs1106175 rs17054921 rs8064924 rs8070440
NA06985 "G/G"     "A/G"     "C/C"      "G/G"     "G/G"
NA06991 "G/G"     "A/A"     "C/C"      "G/G"     "G/G"
NA06993 "G/G"     "A/A"     "C/C"      "G/G"     "G/G"
NA06994 "A/G"     "A/G"     "C/C"      "A/G"     "G/G"
NA07000 "G/G"     "A/A"     "C/C"      "G/G"     "G/G"
```

This function can use genotype information in the A/B format, assuming that B denotes the alphabetically later nucleotide. Because we have direct nucleotide coding in our matrix, we set the `matIsAB` parameter to false in this call.

```
> h17 = riskyAlleleCount(gg17N, matIsAB=FALSE, chr="ch17")
> h17[1:5,1:5]

        rs7217319 rs684232 rs623323 rs2360111 rs11078597
NA06985         0        0        2         0          0
NA06991         0        0        2         1          0
NA06993         0        0        2         2          0
NA06994         0        0        2         2          1
NA07000         0        0        2         0          0

> table(as.numeric(h17))

    0     1     2
16539  7174  5357
```

It is of interest to bind the counts back to the catalog data.

```
> gwr = gwrngs
> gwr = gwr[colnames(h17),]
> mcols(gwr) = cbind(mcols(gwr), DataFrame(t(h17)))
> sn = rownames(h17)
> gwr[,c("Disease.Trait", sn[1:4])]

gwasloc instance with 323 records and 5 attributes per record.
Extracted:  2013-12-03
Excerpt:
GRanges with 5 ranges and 5 metadata columns:
      seqnames             ranges strand |                      Disease.Trait
         <Rle>          <IRanges>  <Rle> |                        <character>
  [1]    chr17 [  38924,   38924]      * |                    AIDS progression
  [2]    chr17 [ 618965,  618965]      * |                     Prostate cancer
  [3]    chr17 [ 700020,  700020]      * |                     Type 2 diabetes
  [4]    chr17 [ 831667,  831667]      * | Economic and political preferences
  [5]    chr17 [1618363, 1618363]      * |                 Serum albumin level
       NA06985   NA06991   NA06993   NA06994
      <integer> <integer> <integer> <integer>
  [1]         0         0         0         0
  [2]         0         0         0         0
  [3]         2         2         2         2
  [4]         0         1         2         2
  [5]         0         0         0         1
  ---
  seqlengths:
       chr1      chr2      chr3      chr4 ...     chr21     chr22      chrX
   249250621 243199373 198022430 191154276 ...  48129895  51304566 155270560
```

Now by programming on the metadata columns, we can identify individuals with particular risk profiles.

5 Imputation to unobserved loci

If we lack information on a specific locus s, but have reasonably dense genotyping on a subject, population genetics may allow a reasonable guess at the genotype at s for this subject. Many algorithms for genotype imputation have been proposed. Here we use a very simple approach due to David Clayton in the *snpStats* package.

We use the "low coverage" 1000 genomes genotypes for the CEU (central European) HapMap cohort as a base for constructing imputation rules. We focus on chromosome 17 for illustration.

The base data are

```
> data(low17)
> low17

A SnpMatrix with  60 rows and  196327 columns
Row names:  NA06985 ... NA12874
Col names:  chr17:1869 ... chr17:78654554
```

A somewhat sparser set of genotypes (HapMap phase II, genomewide 4 million loci) on chromosome 17 is archived as `g17SM`. This has a compact SnpMatrix encoding of genotypes.

```
> data(g17SM)
> g17SM

A SnpMatrix with  90 rows and  89701 columns
Row names:  NA06985 ... NA12892
Col names:  rs6565733 ... rs4986109
```

For a realistic demonstration, we use the subset of these loci that are present on the Affy 6.0 SNP array.

```
> data(gw6.rs_17)
> g17SM = g17SM[, intersect(colnames(g17SM), gw6.rs_17)]
> dim(g17SM)

[1]    90 20359
```

The base data were used to create a set of rules allowing imputation from genotypes in the sparse set to the richer set. Some rules involve only a single locus, some as many as 4. The construction of rules involves tuning of modeling parameters. See snp.imputation in snpStats for details.

```
> if (!exists("rules_6.0_1kg_17")) data(rules_6.0_1kg_17)
> rules_6.0_1kg_17[1:5,]

chr17:1869  ~  rs9915268+rs11247571+rs9895105+rs6598837 (MAF = 0.06666667, R-squared =
chr17:2220  ~  rs4790867+rs10454094+rs2586238+rs7207284 (MAF = 0.125, R-squared = 0.706
chr17:6689  ~  rs4424950+rs4790867+rs7225087+rs11658347 (MAF = 0.125, R-squared = 0.592
rs34663111  ~  rs11658079+rs1609550+rs4985594+rs9788983 (MAF = 0.1166667, R-squared = 0
rs62054999  ~  rs17609440+rs2740351+rs2589492+rs16956017 (MAF = 0.125, R-squared = 0.26
```

The summary of rules shows the degree of association between the predictors and predictands in terms of R^2. Many potential targets are not imputed.

```
> summary(rules_6.0_1kg_17)
```

```
             SNPs used
R-squared     1 tags 2 tags 3 tags 4 tags  <NA>
  [0,0.1)        655    785    276     56     0
  [0.1,0.2)        7    664    926    868     0
  [0.2,0.3)        0    158    916   3054     0
  [0.3,0.4)        0     28    411   5104     0
  [0.4,0.5)        0     20    203   6365     0
  [0.5,0.6)        0     21    121   6052     0
  [0.6,0.7)        0     29    104   5623     0
  [0.7,0.8)        0     54    108   6330     0
  [0.8,0.9)        0    141    225   9506     0
  [0.9,0.95)     652    700    572   8056     0
  [0.95,0.99)   7274   1689   1388   6158     0
  [0.99,1]     33660   1353   2326  10152     0
  <NA>             0      0      0      0 53601
```

The overlap between the 6.0-resident g17SM loci and the catalog is

```
> length(intersect(colnames(g17SM), mcols(gwrngs)$SNPs))
```

```
[1] 132
```

The new expected B allele counts are

```
> exg17 = impute.snps(rules_6.0_1kg_17, g17SM)
```

The number of new loci that coincide with risk loci in the catalog is:

```
> length(intersect(colnames(exg17), mcols(gwrngs)$SNPs))
```

```
[1] 197
```

6 Formal management of trait vocabularies

6.1 Diseases: Disease Ontology

The Disease Ontology project Osborne et al. (2009) formalizes a vocabulary for human diseases. Bioconductor's DO.db package is a curated representation.

```
> library(DO.db)
> DO()
```

```
Quality control information for DO:


This package has the following mappings:

DOANCESTOR has 6343 mapped keys (of 6344 keys)
DOCHILDREN has 1787 mapped keys (of 6344 keys)
DOOBSOLETE has 2383 mapped keys (of 2383 keys)
DOOFFSPRING has 1787 mapped keys (of 6344 keys)
DOPARENTS has 6343 mapped keys (of 6344 keys)
DOTERM has 6344 mapped keys (of 6344 keys)


Additional Information about this package:

DB schema: DO_DB
DB schema version: 1.0
```

All tokens of the ontology are acquired via:

```
> alltob = unlist(mget(mappedkeys(DOTERM), DOTERM))
> allt = sapply(alltob, Term)
> allt[1:5]
```

```
                DOID:0000000                 DOID:0001816
"gallbladder disease"                 "angiosarcoma"
                DOID:0002116                 DOID:0014667
                 "pterygium"          "disease of metabolism"
                DOID:0050004
"seminal vesicle acute gonorrhea"
```

Direct mapping from disease trait tokens in the catalog to this vocabulary succeeds for a modest proportion of records.

```
> cattra = mcols(gwrngs)$Disease.Trait
> mat = match(tolower(cattra), tolower(allt))
> catDO = names(allt)[mat]
> na.omit(catDO)[1:50]
```

```
 [1] "DOID:3347"  "DOID:3347"  "DOID:3347"  "DOID:3347"  "DOID:3347"
 [6] "DOID:3347"  "DOID:1094"  "DOID:1094"  "DOID:1094"  "DOID:1094"
[11] "DOID:2055"  "DOID:2055"  "DOID:1682"  "DOID:12365" "DOID:12365"
[16] "DOID:12365" "DOID:12365" "DOID:12365" "DOID:12365" "DOID:12361"
[21] "DOID:12361" "DOID:12361" "DOID:12361" "DOID:12361" "DOID:12361"
```

```
[26] "DOID:332"   "DOID:332"   "DOID:8689"  "DOID:8689"  "DOID:8689"
[31] "DOID:8689"  "DOID:8689"  "DOID:8689"  "DOID:8689"  "DOID:8689"
[36] "DOID:8689"  "DOID:8689"  "DOID:8689"  "DOID:8689"  "DOID:8689"
[41] "DOID:8689"  "DOID:8689"  "DOID:12129" "DOID:12129" "DOID:12129"
[46] "DOID:12129" "DOID:12129" "DOID:12129" "DOID:12129" "DOID:12129"
```

```
> mean(is.na(catDO))
```

```
[1] 0.7946871
```

Approximate matching of unmatched tokens can proceed by various routes. Some traits are not diseases, and will not be mappable using Disease Ontology. However, consider

```
> unique(cattra[is.na(catDO)])[1:20]
```

```
 [1] "Warfarin maintenance dose"
 [2] "DNA methylation (parent-of-origin)"
 [3] "DNA methylation (variation)"
 [4] "Insomnia"
 [5] "Sleep depth"
 [6] "Sleep duration"
 [7] "Sleep quality"
 [8] "Sleep time"
 [9] "Response to antidepressant treatment (citalopram)"
[10] "Educational attainment"
[11] "LDL cholesterol"
[12] "Triglycerides"
[13] "Blood trace element (Cu levels)"
[14] "Blood trace element (Se levels)"
[15] "Blood trace element (Zn levels)"
[16] "Post-traumatic stress disorder (asjusted for relatedness)"
[17] "Congenital heart malformation"
[18] "Adverse response to radiation therapy"
[19] "Primary tooth development (number of teeth)"
[20] "Primary tooth development (time to first tooth eruption)"
```

```
> nomatch = cattra[is.na(catDO)]
> unique(nomatch)[1:5]
```

```
[1] "Warfarin maintenance dose"          "DNA methylation (parent-of-origin)"
[3] "DNA methylation (variation)"        "Insomnia"
[5] "Sleep depth"
```

Manual searching shows that a number of these have very close matches.

6.2 Other phenotypic traits: Human Phenotype Ontology

Bioconductor does not possess an annotation package for phenotype ontology, but the standardized OBO format can be parsed and modeled into a graph.

```
> hpobo = gzfile(dir(system.file("obo", package="gwascat"), pattern="hpo", full=TRUE)
> HPOgraph = obo2graphNEL(hpobo)
> close(hpobo)
```

The phenotypic terms are obtained via:

```
> hpoterms = unlist(nodeData(HPOgraph, nodes(HPOgraph), "name"))
> hpoterms[1:10]
```

```
                                HP:0000001
                                     "All"
                                HP:0000002
              "Abnormality of body height"
                                HP:0000003
            "Multicystic kidney dysplasia"
                                HP:0000004
               "Onset and clinical course"
                                HP:0000005
                     "Mode of inheritance"
                                HP:0000006
          "Autosomal dominant inheritance"
                                HP:0000007
         "Autosomal recessive inheritance"
                                HP:0000008
"Abnormality of female internal genitalia"
                                HP:0000009
   "Functional abnormality of the bladder"
                                HP:0000010
      "Recurrent urinary tract infections"
```

Exact hits to unmatched GWAS catalog traits exist:

```
> intersect(tolower(nomatch), tolower(hpoterms))
```

```
 [1] "insomnia"                       "scoliosis"
 [3] "eating disorders"               "hypertriglyceridemia"
 [5] "coronary artery calcification" "sagittal craniosynostosis"
 [7] "glioma"                         "autism"
 [9] "atrial fibrillation"            "glioblastoma"
[11] "stroke"                         "iga nephropathy"
[13] "nephropathy"                    "freckling"
[15] "knee osteoarthritis"            "hearing impairment"
```

More work on formalization of trait terms is underway.

7 CADD scores

Kircher et al. (Kircher et al., 2014) define combined annotation-dependent depletion scores measuring variant pathogenicity in an integrative way. Small requests to bind scores for SNV to GRanges can be resolved through HTTP; large requests can be carried out on a local tabix-indexed selection from their archive.

```
> g3 = as(gwrngs, "GRanges")
> bg3 = bindcadd_snv( g3[which(seqnames(g3)=="chr3")][1:20] )
> inds = ncol(mcols(bg3))
> bg3[, (inds-3):inds]
```

This requires cooperation of network interface and server, so we don't evaluate in vignette build but on 1 Apr 2014 the response was:

```
GRanges with 20 ranges and 4 metadata columns:
       seqnames                ranges strand   |         Ref         Alt
          <Rle>             <IRanges>  <Rle>   | <character> <character>
   [1]        3 [109789570, 109789570]      *   |           A           G
   [2]        3 [ 25922285,  25922285]      *   |           G           A
   [3]        3 [109529550, 109529550]      *   |           T           C
   [4]        3 [175055759, 175055759]      *   |           T           G
   [5]        3 [191912870, 191912870]      *   |           C           T
   ...      ...                     ...    ... ...         ...         ...
  [16]        3 [187716886, 187716886]      *   |           A           G
  [17]        3 [160820524, 160820524]      *   |           G           C
  [18]        3 [169518455, 169518455]      *   |           T           C
  [19]        3 [179172979, 179172979]      *   |           G           T
  [20]        3 [171785168, 171785168]      *   |           G           C
         CScore     PHRED
      <numeric> <numeric>
   [1] -0.182763     3.110
   [2] -0.289708     2.616
   [3]  0.225373     5.216
   [4] -0.205689     3.003
   [5] -0.172189     3.161
   ...       ...       ...
  [16] -0.019710     3.913
  [17] -0.375183     2.235
  [18] -0.695270     0.987
```

```
  [19] -0.441673     1.949
  [20]  0.231972     5.252
  ---
  seqlengths:
           1         2         3         4 ...        21        22         X
   249250621 243199373 198022430 191154276 ...  48129895  51304566 155270560
```

8 Appendix: Adequacy of location annotation

A basic question concerning the use of archived SNP identifiers is durability of the association between asserted location and SNP identifier. The `chklocs` function uses a current Bioconductor SNPlocs package to check this.

For example, to verify that locations asserted on chromosome 20 agree between the Bioconductor dbSNP image and the gwas catalog,

```
> if ("SNPlocs.Hsapiens.dbSNP.20120608" %in% installed.packages()[,1]) {
+  library(SNPlocs.Hsapiens.dbSNP.20120608)
+  suppressWarnings(chklocs("20"))
+ }
```

```
[1] TRUE
```

This is not a fast procedure but has succeeded for all chromosomes 1-22 when checked off line.

References

Martin Kircher, Daniela M Witten, Preti Jain, Brian J O'Roak, Gregory M Cooper, and Jay Shendure. A general framework for estimating the relative pathogenicity of human genetic variants. *Nat Genet*, Feb 2014. doi: 10.1038/ng.2892.

John D Osborne, Jared Flatow, Michelle Holko, Simon M Lin, Warren A Kibbe, Lihua Julie Zhu, Maria I Danila, Gang Feng, and Rex L Chisholm. Annotating the human genome with disease ontology. *BMC Genomics*, 10 Suppl 1:S6, Jan 2009. doi: 10.1186/1471-2164-10-S1-S6. URL `http://www.biomedcentral.com/1471-2164/10/S1/S6`.