gwascat: structuring and querying the NHGRI GWAS catalog

VJ Carey*

May 15, 2016

Contents

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()`.

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

1.3 Illustrations: computing

Available functions are:

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

```
 [1] "adj"                "bindcadd_snv"       "chklocs"
 [4] "getRsids"           "getTraits"          "gwcex2gviz"
 [7] "impute.snps"        "ldtagr"             "locs4trait"
[10] "makeCurrentGwascat" "metadata"           "node2uri"
[13] "nodeData"           "nodes"              "obo2graphNEL"
[16] "riskyAlleleCount"   "show"               "subGraph"
[19] "subsetByChromosome" "subsetByTraits"     "topTraits"
[22] "traitsManh"         "ugraph"             "uri2node"
```

The extended GRanges instance with all SNP-disease associations is obtained as follows, using the hg19 genome build (GRCh38/hg38-based ranges are also available as ebicat38).

```
> data(ebicat37)
```

To determine the most frequently occurring traits:

```
> topTraits(ebicat37)
```

```
 Obesity-related traits                 Height       IgG glycosylation
                    957                    822                    699
        Type 2 diabetes   Rheumatoid arthritis        Crohn's disease
                    340                    294                    249
          Schizophrenia Blood metabolite levels        HDL cholesterol
                    248                    245                    220
          Breast cancer
                    199
```

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

```
> subsetByTraits(ebicat37, tr="LDL cholesterol")[1:3]
```

```
gwasloc instance with 3 records and 36 attributes per record.
Extracted:
Genome:  GRCh37
Excerpt:
GRanges object with 3 ranges and 3 metadata columns:
```

```
    seqnames                ranges strand |   DISEASE/TRAIT        SNPS
       <Rle>             <IRanges>  <Rle> |     <character> <character>
[1]     chr2 [ 44072576,  44072576]      * | LDL cholesterol   rs4299376
[2]     chr1 [150958836, 150958836]      * | LDL cholesterol    rs267733
[3]     chr2 [ 21263900,  21263900]      * | LDL cholesterol   rs1367117
      P-VALUE
    <numeric>
[1]     4e-72
[2]     5e-09
[3]    1e-182
-------
seqinfo: 23 sequences from GRCh37 genome
```

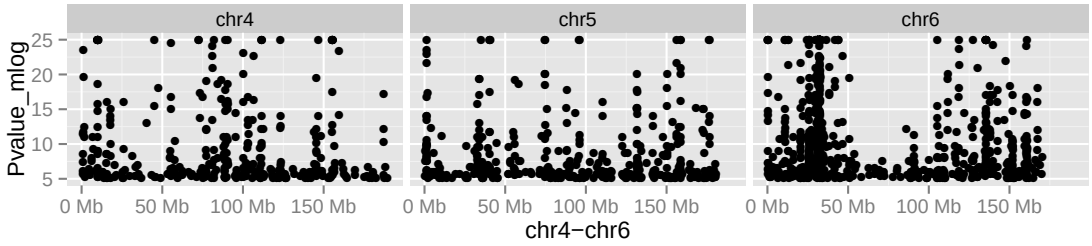
2 Some visualizations

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 = ebicat37
> mlpv = mcols(ebicat37)$PVALUE_MLOG
> mlpv = ifelse(mlpv > 25, 25, mlpv)
> mcols(gwtrunc)$PVALUE_MLOG = mlpv
> library(GenomeInfoDb)
> seqlevelsStyle(gwtrunc) = "UCSC"
> 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")
```



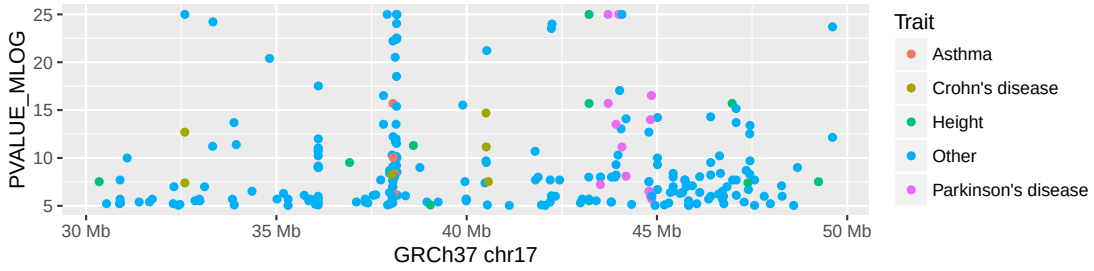
2.2 Annotated Manhattan plot

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)
```

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.

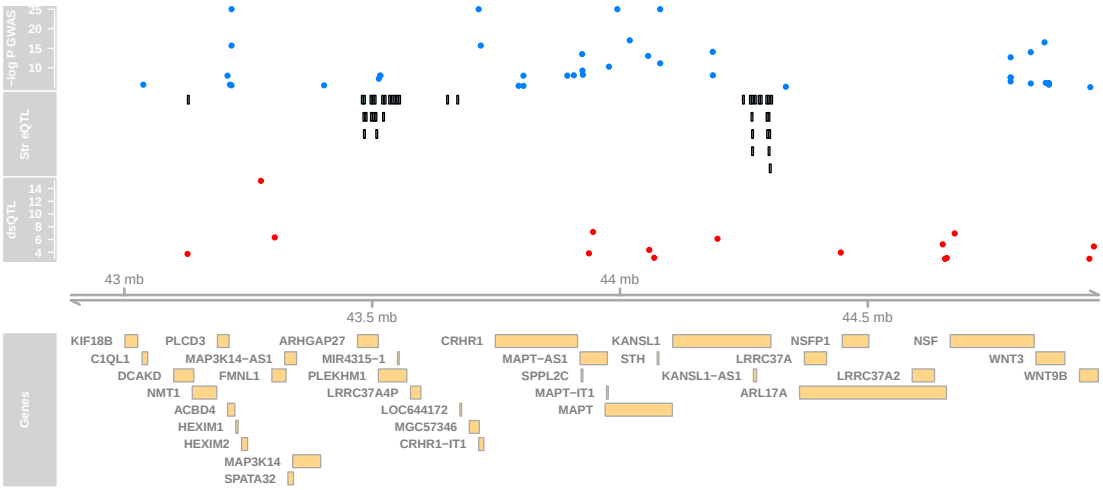
```
> g2 = GRanges(seqnames="chr17", IRanges(start=4.3e7, width=2e6))
> seqlevelsStyle(ebicat37) = "UCSC"
> basic = gwcex2gviz(basegr = ebicat37, contextGR=g2, plot.it=FALSE)
```

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 = ebicat37[ intersect(getRsids(ebicat37), rson6) ]
> sort(table(getTraits(intr)), decreasing=TRUE)[1:10]
```

```
                                                 Height
                                                      6
                                  Select biomarker traits
                                                      3
                                        Bipolar disorder
                                                      2
                                           Dental caries
                                                      2
                                      Homocysteine levels
                                                      2
                                        IgG glycosylation
                                                      2
      Immune reponse to smallpox (secreted IFN-alpha)
                                                      2
                                         Metabolic traits
                                                      2
                                 Mitochondrial DNA levels
                                                      2
Response to cytadine analogues (cytosine arabinoside)
                                                      2
```

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(ebicat37), gr6.0))
> length(ovraw)

[1] 128

> ovaug = suppressWarnings(subsetByOverlaps(ag(ebicat37+500), gr6.0))
> length(ovaug)

[1] 207
```

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

```
> rawrs = mcols(ovraw)$SNPS
> augrs = mcols(ovaug)$SNPS
> ebicat37[augrs]

gwasloc instance with 207 records and 36 attributes per record.
Extracted:
Genome:  GRCh37
Excerpt:
GRanges object with 5 ranges and 3 metadata columns:
      seqnames                ranges strand |
         <Rle>             <IRanges>  <Rle> |
  [1]    chr11 [  4605195,   4605195]      * |
  [2]    chr11 [ 46761055,  46761055]      * |
  [3]     chr1 [169519049, 169519049]      * |
  [4]    chr11 [ 61640379,  61640379]      * |
  [5]     chr1 [  8495945,   8495945]      * |
                                                                DISEASE/TRAIT
                                                                  <character>
  [1] Response to serotonin reuptake inhibitors in major depressive disorder
  [2]                                                  Venous thromboembolism
  [3]                                                  Venous thromboembolism
  [4]                                                 Trans fatty acid levels
  [5]                                                          Optic cup area
            SNPS   P-VALUE
      <character> <numeric>
  [1]   rs2566255     2e-06
  [2]   rs1799963     2e-09
  [3]      rs6025     1e-96
```

```
  [4]    rs174449     8e-07
  [5]    rs301801     3e-08
  -------
  seqinfo: 23 sequences from GRCh37 genome
```

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

```
> setdiff( getTraits(ebicat37[augrs]), getTraits(ebicat37[rawrs]) )

 [1] "Response to serotonin reuptake inhibitors in major depressive disorder"
 [2] "Venous thromboembolism"
 [3] "Hippocampal volume"
 [4] "Red blood cell fatty acid levels"
 [5] "Febrile seizures (MMR vaccine-related)"
 [6] "Eosinophilic esophagitis"
 [7] "QT interval"
 [8] "Smoking cessation"
 [9] "Systolic blood pressure (alcohol consumption interaction)"
[10] "QRS duration in Tripanosoma cruzi seropositivity"
[11] "Functional impairment in major depressive disorder, bipolar disorder and schizoph
[12] "Periodontitis (CDC/AAP)"
[13] "Odorant perception (&beta;-damascenone)"
[14] "Bronchopulmonary dysplasia"
[15] "Self-reported allergy"
[16] "Migraine"
[17] "Reading and spelling"
[18] "Adverse response to chemotherapy (neutropenia/leucopenia) (carboplatin)"
[19] "Obesity"
[20] "Autism spectrum disorder, attention deficit-hyperactivity disorder, bipolar disor
[21] "Response to taxane treatment (docetaxel)"
[22] "Response to amphetamines"
[23] "Neuroblastoma"
[24] "Lentiform nucleus volume"
[25] "Capecitabine sensitivity"
[26] "Fasting glucose-related traits (interaction with BMI)"
[27] "Response to angiotensin II receptor blocker therapy"
[28] "Response to hepatitis C treatment"
[29] "Response to anti-depressant treatment in major depressive disorder"
[30] "Phospholipid levels (plasma)"
[31] "Endometrial cancer"
[32] "MRI atrophy measures"
[33] "Self-rated health"
[34] "Neonatal lupus"
```

```
[35] "Crohn's disease"
[36] "Response to statin therapy"
[37] "Tanning"
[38] "Osteonecrosis of the jaw"
[39] "Hip geometry"
[40] "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",
+  gwwl = ebicat37)
> h17[1:5,1:5]
```

```
        rs7217319 rs684232 rs623323 rs747685 rs747687
NA06985         0        0        2        1        1
NA06991         0        0        2        2        2
NA06993         0        0        2        1        1
NA06994         0        0        2        2        2
NA07000         0        0        2        2        2
```

```
> table(as.numeric(h17))
```

```
    0     1     2
23876  9954  7570
```

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

```
> gwr = ebicat37
> 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 460 records and 5 attributes per record.
Extracted:
Genome:  GRCh37
Excerpt:
GRanges object with 5 ranges and 5 metadata columns:
      seqnames           ranges strand |    DISEASE.TRAIT   NA06985   NA06991
         <Rle>        <IRanges>  <Rle> |      <character> <integer> <integer>
  [1]    chr17 [ 38924,  38924]      * | AIDS progression         0         0
  [2]    chr17 [618965, 618965]      * |  Prostate cancer         0         0
  [3]    chr17 [700020, 700020]      * |  Type 2 diabetes         2         2
  [4]    chr17 [775051, 775051]      * |   Blood pressure         1         2
  [5]    chr17 [775334, 775334]      * |   Blood pressure         1         2
       NA06993   NA06994
      <integer> <integer>
  [1]         0         0
  [2]         0         0
  [3]         2         2
  [4]         1         2
  [5]         1         2
  -------
  seqinfo: 23 sequences from GRCh37 genome
```

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(ebicat37)$SNPS))
```

```
[1] 184
```

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(ebicat37)$SNPS))
```

```
[1] 295
```

6 Formal management of trait vocabularies

6.1 Diseases: Disease Ontology

The Disease Ontology project **?** 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 6569 mapped keys (of 6570 keys)
DOCHILDREN has 1811 mapped keys (of 6570 keys)
DOOBSOLETE has 2374 mapped keys (of 2374 keys)
DOOFFSPRING has 1811 mapped keys (of 6570 keys)
DOPARENTS has 6569 mapped keys (of 6570 keys)
DOTERM has 6570 mapped keys (of 6570 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:0001816                       DOID:0002116
                 "angiosarcoma"                        "pterygium"
                  DOID:0014667                       DOID:0050004
        "disease of metabolism" "seminal vesicle acute gonorrhea"
                  DOID:0050012
                 "chikungunya"
```

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

```
> cattra = mcols(ebicat37)$Disease.Trait
> mat = match(tolower(cattra), tolower(allt))
> catDO = names(allt)[mat]
> na.omit(catDO)[1:50]
```

```
 [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[26] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
```

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

```
[1] NaN
```

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]
```

```
NULL
```

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

```
NULL
```

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))
```

```
character(0)
```

More work on formalization of trait terms is underway.

7 CADD scores

Kircher et al. (**?**) 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(ebicat37, "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.dbSNP144.GRCh37" %in% installed.packages()[,1]) {
+  library(SNPlocs.Hsapiens.dbSNP144.GRCh37)
+  suppressWarnings(chklocs("20", ebicat37))
+ }

[1] TRUE
```

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