MyVariant.info R Client

Adam Mark

June 11, 2015

Contents

1 Overview

MyVariant.Info is a simple-to-use REST web service to query/retrieve genetic variant annotation from an aggregation of variant annotation resources. *myvariant* is an easy-to-use R wrapper to access MyVariant.Info services and explore variant annotions.

2 Variant Annotation Service

2.1 Obtaining HGVS IDs from a VCF file.

- Use `readVcf` from the VariantAnnotation package to read a Vcf file in. The Vcf object can then be passed to `formatHgvs` to retrieve HGVS IDs. HGVS IDs are based on the GRCh38/hg19 reference genome. Support for hg38 is coming soon.

```
> file.path <- system.file("extdata", "dbsnp_mini.vcf", package="myvariant")
> vcf <- readVcf(file.path, genome="hg19")
> rowRanges(vcf)

GRanges object with 240 ranges and 5 metadata columns:
              seqnames           ranges strand | paramRangeID              REF
                 <Rle>        <IRanges>  <Rle> |      <factor>  <DNAStringSet>
   rs376643643        1 [10019, 10020]      * |          <NA>              TA
   rs373328635        1 [10055, 10055]      * |          <NA>               T
    rs62651026        1 [10108, 10108]      * |          <NA>               C
   rs376007522        1 [10109, 10109]      * |          <NA>               A
   rs368469931        1 [10139, 10139]      * |          <NA>               A
           ...      ...              ...    ... ...           ...             ...
   rs544020171        1 [17654, 17654]      * |          <NA>               T
   rs563880190        1 [17694, 17694]      * |          <NA>               C
   rs574335987        1 [17695, 17695]      * |          <NA>               G
   rs374995955        1 [17697, 17697]      * |          <NA>               G
   rs543363182        1 [17709, 17709]      * |          <NA>               T
                             ALT      QUAL      FILTER
              <DNAStringSetList> <numeric> <character>
   rs376643643                T      <NA>           .
   rs373328635               TA      <NA>           .
    rs62651026                T      <NA>           .
   rs376007522                T      <NA>           .
   rs368469931                T      <NA>           .
           ...              ...       ...         ...
   rs544020171                C      <NA>           .
   rs563880190                T      <NA>           .
   rs574335987                A      <NA>           .
   rs374995955                C      <NA>           .
   rs543363182                G      <NA>           .
   -------
   seqinfo: 1 sequence from hg19 genome; no seqlengths
```

- You can then use `formatHgvs` to extract HGVS IDs from the Vcf object.

```
> hgvs <- formatHgvs(vcf, variant_type="snp")
> head(hgvs)

[1] "chr1:g.10108TA>T" "chr1:g.10109T>TA" "chr1:g.10139C>T"  "chr1:g.10150A>T"
[5] "chr1:g.10177A>T"  "chr1:g.10180TA>T"
```

2.2 getVariant

- Use `getVariant`, the wrapper for GET query of "/v1/variant/<hgvsid>" service, to return the variant object for the given HGVS id.

```
> variant <- getVariant("chr1:g.35367G>A")
> variant[[1]]$dbnsfp$genename
```

```
[1] "FAM138A"
```

```
> variant[[1]]$cadd$phred
```

```
[1] 1.493
```

2.3 getVariants

- Use getVariants, the wrapper for POST query of "/v1/variant" service, to return the list of variant objects for the given character vector of HGVS ids.

```
> getVariants(head(hgvs),
+               fields="cadd.consequence")
```

```
DataFrame with 6 rows and 2 columns
    notfound              query
  <logical>        <character>
1      TRUE chr1:g.10108TA>T
2      TRUE chr1:g.10109T>TA
3      TRUE   chr1:g.10139C>T
4      TRUE   chr1:g.10150A>T
5      TRUE   chr1:g.10177A>T
6      TRUE chr1:g.10180TA>T
```

3 Variant Query Service

3.1 queryVariant

- queryVariant is a wrapper for GET query of "/v1/query?q=<query>" service, to return the query result. This function accepts wild card input terms and allows you to query for variants that contain a specific annotation. For example, the following query searches for the CADD phred score and consequence for all variants whose genename (dbNSFP) is MLL2.

```
> queryVariant(q="dbnsfp.genename:MLL2", fields=c("cadd.phred", "cadd.consequence"))
```

```
$hits
                    _id   _score cadd.consequence cadd.phred
1  chr12:g.49418460A>T 9.948488      STOP_GAINED         59
2  chr12:g.49418658C>A 9.948488      STOP_GAINED         57
3  chr12:g.49420281G>T 9.948488      STOP_GAINED         59
4  chr12:g.49420340T>A 9.948488      STOP_GAINED         53
5  chr12:g.49420364T>A 9.948488      STOP_GAINED         53
6  chr12:g.49420844C>A 9.948488      STOP_GAINED         53
```

```
7  chr12:g.49420888G>C 9.948488          STOP_GAINED           53
8  chr12:g.49421602A>T 9.948488          STOP_GAINED           53
9  chr12:g.49421807C>A 9.948488          STOP_GAINED           57
10 chr12:g.49422661T>A 9.948488          STOP_GAINED           45


$max_score
[1] 9.948488


$took
[1] 529


$total
[1] 37953
```

- You can also use `queryVariant` to retrieve all annotations that map to a specific rsID.

```
> queryVariant(q="rs58991260", fields="dbsnp.flags")$hits

                  _id   _score                                            flags
1 chr1:g.218631822G>A 17.48191 ASP, G5, G5A, GNO, KGPhase1, KGPhase3, SLO
```

3.2 `queryVariants`

- `queryVariants` is a wrapper for POST query of "/v1/query?q=<query>" service, to return the query result. Query terms include any available field as long as scopes are defined. The following example reads the dbSNP rsIDs from a VCF and queries for all fields. The returned DataFrame can then be easily subsetted to include, for example, those that have not been documented in the Wellderly study.

```
> rsids <- paste("rs", info(vcf)$RS, sep="")
> res <- queryVariants(q=rsids, scopes="dbsnp.rsid", fields="all")

Finished
Pass returnall=TRUE to return lists of duplicate or missing query terms.

> subset(res, !is.na(wellderly.vartype))$query

 [1] "rs367896724" "rs145427775" "rs147093981" "rs56289060"  "rs55998931"  "rs199606420"
 [7] "rs58108140"  "rs62635284"  "rs62635286"  "rs200579949" "rs531730856" "rs180734498"
[13] "rs527952245" "rs201696125" "rs370886505" "rs546169444" "rs201055865" "rs369473859"
[19] "rs62635298"  "rs79585140"  "rs75454623"  "rs199856693" "rs201855936" "rs71252251"
[25] "rs201045431" "rs71252250"  "rs200030104" "rs78601809"  "rs62636497"  "rs62636497"
[31] "rs201635489" "rs533630043" "rs2691315"   "rs113442401" "rs572465511" "rs372319358"
[37] "rs200358166" "rs11489794"  "rs113141985" "rs62636498"  "rs148220436" "rs141130360"
[43] "rs150723783" "rs201459529" "rs199676946" "rs62636367"  "rs62636368"  "rs200205172"
[49] "rs199745162" "rs200658479" "rs201833382" "rs199740902" "rs200978805" "rs201535981"
[55] "rs192890528" "rs200046632" "rs374995955"
```

4 References

MyVariant.info help@myvariant.info