02-Hail-rare-variant-analysis

June 16, 2021

1 Institute for Behavioral Genetics International Statistical Genetics 2021 Workshop

2 Rare Variant Analysis of Sequencing Data with Hail

You will learn more details on the analysis of rare variant signals using Hail in the SAIGE session. In this notebook, your learning objective is to :

- Understand basic principles behind simple variant aggregation and burden tests.

GWAS is a great tool for finding associations between **common variants** and disease, but is underpowered to detect rare-variant associations, because rare variants by definition have small sample sizes.

It is possible to find associations between rare variants and disease by **grouping variants of similar effect**, and testing each group.

One possible solution is to sum variant counts according to some genomic interval (for instance, gene), and then association with these intervals. This is often called a gene burden test.

2.1 Setup

Same as in the last practical, these steps initialize our Hail session.

[1]:
```python
import hail as hl
from hail.plot import output_notebook, show
```

[2]:
```python
hl.init()
output_notebook()
```

```
Running on Apache Spark version 3.1.1
SparkUI available at http://10.0.0.72:4040
Welcome to
     __  __     <>__
    / /_/ /__  __/ /
   / __  / _ `/ / /
  /_/ /_/\_,_/_/_/   version 0.2.68-13190f0b6103
LOGGING: writing to /Users/kumar/Dropbox (Partners HealthCare)/HailTeam/Workshop
s/2021_Boulder/2021_IBG_Hail/resources/hail-20210616-1712-0.2.68-13190f0b6103.lo
g
```

3 Step 1: Rapid-fire import, QC, sample annotation

The last notebook covered these steps in detail. We'll do them quickly here:

[3]:
```
# read matrix from disk; this was written from the imported VCF in the common␣
 ↪variant practical
mt = hl.read_matrix_table('resources/hgdp.mt')

# import annotations
sd = hl.import_table('resources/HGDP_sample_data.tsv',
                     key='sample_id',
                     impute=True)

# annotate columns
mt = mt.annotate_cols(sample_data = sd[mt.s])

# remove non-PASS variants
mt = mt.filter_rows(hl.len(mt.filters) == 0)
```

```
2021-06-16 17:13:58 Hail: INFO: Reading table to impute column types
2021-06-16 17:14:53 Hail: INFO: Finished type imputation
  Loading field 'sample_id' as type str (imputed)
  Loading field 'pop' as type str (imputed)
  Loading field 'continental_pop' as type str (imputed)
  Loading field 'sex_karyotype' as type str (imputed)
  Loading field 'sleep_duration' as type int32 (imputed)
  Loading field 'tea_intake_daily' as type int32 (imputed)
  Loading field 'general_happiness' as type float64 (imputed)
  Loading field 'screen_time_per_day' as type int32 (imputed)
```

3.1 Discard common variants

Next, we will keep variants with an allele frequency of under 1%. Including common variants will only reduce the power of a burden test.

We could rerun `hl.variant_qc` here, or use an aggregator designed to compute allele frequencies and counts:

[4]:
```
mt = mt.filter_rows(hl.agg.call_stats(mt.GT, mt.alleles).AF[1] < 0.01)
```

4 Step 2: Group by gene

We have variant annotations in a text file in the `resources/` folder. We will use these to annotate our matrix table with gene and consequence information.

Additionally, you can also use the `VEP` annotation tool which provides a *huge* number of potentially useful variant annotations. If you are running Hail on Google Cloud Platform (GCP), the Hail team has done the work of installing and configuring VEP. The team is also working on a new resource called the "annotation database": see here for more information.

```
[5]: annotation_ht = hl.import_table('resources/hgdp_gene_annotations.tsv',␣
      ↪impute=True)
```

```
2021-06-16 17:14:55 Hail: INFO: Reading table to impute column types
2021-06-16 17:15:01 Hail: INFO: Finished type imputation
  Loading field 'variant' as type str (imputed)
  Loading field 'gene_symbol' as type str (imputed)
  Loading field 'csq' as type str (imputed)
```

```
[6]: annotation_ht.show()
```

```
+---------------------+-----------------+--------------------------------------+
| variant             | gene_symbol     | csq                                  |
+---------------------+-----------------+--------------------------------------+
| str                 | str             | str                                  |
+---------------------+-----------------+--------------------------------------+
| "chr1:17379:G:A"    | "MIR6859-1"     | "mature_miRNA_variant"               |
| "chr1:95068:G:A"    | "AL627309.1"    | "intron_variant"                     |
| "chr1:111735:C:A"   | "AL627309.1"    | "intron_variant"                     |
| "chr1:134610:G:A"   | "CICP27"        | "non_coding_transcript_exon_variant" |
| "chr1:414783:T:C"   | "AL732372.2"    | "intron_variant"                     |
| "chr1:1130877:C:G"  | NA              | NA                                   |
| "chr1:1226707:C:G"  | "SDF4"          | "intron_variant"                     |
| "chr1:1491494:G:A"  | "ATAD3B"        | "intron_variant"                     |
| "chr1:1618118:G:A"  | "MIB2"          | "non_coding_transcript_exon_variant" |
| "chr1:2078529:G:A"  | "PRKCZ"         | "intron_variant"                     |
| "chr1:2512104:G:A"  | "PANK4"         | "intron_variant"                     |
| "chr1:2683695:C:G"  | "TTC34"         | "intron_variant"                     |
| "chr1:2689763:A:C"  | "TTC34"         | "intron_variant"                     |
| "chr1:2758837:C:A"  | "TTC34"         | "intron_variant"                     |
| "chr1:3008858:A:G"  | NA              | NA                                   |
| "chr1:3254545:T:C"  | "PRDM16"        | "intron_variant"                     |
| "chr1:3299310:A:C"  | "PRDM16"        | "intron_variant"                     |
| "chr1:3779436:T:C"  | "LRRC47"        | "3_prime_UTR_variant"                |
| "chr1:3848254:C:T"  | "CEP104"        | "intron_variant"                     |
| "chr1:4156721:C:T"  | NA              | NA                                   |
| "chr1:4510922:C:T"  | NA              | NA                                   |
| "chr1:4748394:G:A"  | "AJAP1"         | "intron_variant"                     |
| "chr1:5008942:A:G"  | "LOC102724429"  | "downstream_gene_variant"            |
| "chr1:5331415:G:T"  | NA              | NA                                   |
+---------------------+-----------------+--------------------------------------+
showing top 24 rows
```

4.0.1 Exercise

Use the `aggregate` method and the familiar `hl.agg.counter` aggregator to compute the number of appearances of each "csq" value.

Which of these would you expect to have no effect on a protein? Which would you expect to have a large effect?

```
[ ]:
```

4.1 Annotate variants with genes

In order join our two tables (the QC-ed data and gene table), we need to create fields of type `locus` and `array<str>` (alleles) to match the row key of our matrix.

We can use the `hl.parse_variant` function to parse the `variant` field of this table of `CHR:POS:REF:ALT` form to a locus and alleles array. Then we assign these new fields to be the key:

```
[7]: parsed = hl.parse_variant(annotation_ht.variant, reference_genome='GRCh38')
annotation_ht = annotation_ht.key_by(locus = parsed.locus, alleles=parsed.
 ↪alleles).drop('variant')
```

Recall how we annotated sample phenotypes earlier in the common variant tutorial – this join looks very similar:

```
[8]: mt = mt.annotate_rows(vep_info = annotation_ht[mt.locus, mt.alleles])
```

Let's `show` the resulting annotations on the matrix table to make sure everything worked:

```
[9]: mt.vep_info.show()
```

```
2021-06-16 17:15:45 Hail: INFO: Coerced sorted dataset

+---------------+------------+----------------------+--------------------------------------+
| locus         | alleles    | vep_info.gene_symbol | vep_info.csq                         |
+---------------+------------+----------------------+--------------------------------------+
| locus<GRCh38> | array<str> | str                  | str                                  |
+---------------+------------+----------------------+--------------------------------------+
| chr1:1226707  | ["C","G"]  | "SDF4"               | "intron_variant"                     |
| chr1:2078529  | ["G","A"]  | "PRKCZ"              | "intron_variant"                     |
| chr1:2512104  | ["G","A"]  | "PANK4"              | "intron_variant"                     |
| chr1:3848254  | ["C","T"]  | "CEP104"             | "intron_variant"                     |
| chr1:5331415  | ["G","T"]  | NA                   | NA                                   |
| chr1:6642670  | ["T","A"]  | "DNAJC11"            | "intron_variant"                     |
| chr1:9228795  | ["G","C"]  | NA                   | NA                                   |
| chr1:10810291 | ["T","G"]  | "LOC105376733"       | "non_coding_transcript_exon_variant" |
| chr1:11830502 | ["A","G"]  | "CLCN6"              | "intron_variant"                     |
| chr1:13832405 | ["G","C"]  | NA                   | NA                                   |
| chr1:16152264 | ["C","A"]  | "EPHA2"              | "intron_variant"                     |
| chr1:17191840 | ["T","G"]  | "LINC02783"          | "intron_variant"                     |
| chr1:17333174 | ["G","A"]  | "PADI4"              | "intron_variant"                     |
| chr1:18777543 | ["C","T"]  | NA                   | NA                                   |
| chr1:21263841 | ["G","A"]  | "ECE1"               | "intron_variant"                     |
| chr1:22255405 | ["C","T"]  | "LOC107985377"       | "intron_variant"                     |
| chr1:22256960 | ["C","G"]  | "LOC107985377"       | "intron_variant"                      |
| chr1:22573653 | ["C","A"]  | "EPHA8"              | "intron_variant"                      |
| chr1:23186871 | ["G","A"]  | NA                   | NA                                    |
| chr1:23228718 | ["C","T"]  | NA                   | NA                                    |
| chr1:24193738 | ["C","T"]  | NA                   | NA                                    |
| chr1:28456724 | ["G","A"]  | "PHACTR4"            | "intron_variant"                      |
| chr1:30088906 | ["C","T"]  | NA                   | NA                                    |
| chr1:30172590 | ["G","A"]  | "LOC105378617"       | "intron_variant"                      |
+---------------+------------+----------------------+---------------------------------------+
showing top 24 rows
```

5 Step 3: Aggregate by gene

Hail's modularity makes it easy to perform non-kernel-based burden tests.

We'll compose two general tools: - group_rows_by / aggregate - hl.linear_regression_rows.

This means that you can flexibly specify the way genotypes are summarized per gene. Using other tools, you may have a few ways to aggregate, but if you want to do something different you are out of luck!

[10]:
```
burden_mt = (
    mt
    .group_rows_by(mt.vep_info.gene_symbol)
    .aggregate(n_variants = hl.agg.count_where(mt.GT.n_alt_alleles() > 0))
)

# filter to genes with at least one rare variant!
burden_mt = burden_mt.filter_rows(hl.agg.sum(burden_mt.n_variants) > 0)
```

[11]:
```
burden_mt.describe(widget=True)
```

```
VBox(children=(HBox(children=(Button(description='globals', layout=Layout(height='30px', width=


Tab(children=(VBox(children=(HTML(value='<p><big>Global fields, with one value in the dataset.
```

[12]:
```
burden_mt.show()
```

```
2021-06-16 17:16:46 Hail: INFO: Coerced sorted dataset
2021-06-16 17:17:16 Hail: INFO: Ordering unsorted dataset with network shuffle

+---------------+--------------------------------+--------------------------------+-----------
| gene_symbol   | 'LP6005441-DNA_F08'.n_variants | 'LP6005441-DNA_C05'.n_variants | 'HGDP00961
+---------------+--------------------------------+--------------------------------+-----------
| str           |                          int64 |                          int64 |
+---------------+--------------------------------+--------------------------------+-----------
| "AAK1"        |                              0 |                              0 |
| "ABHD18"      |                            0 |                            0 |
| "AC000372.1"  |                            0 |                            0 |
| "AC002070.1"  |                            0 |                            0 |
| "AC002996.1"  |                            0 |                            0 |
| "AC003685.1"  |                            0 |                            0 |
| "AC004083.1"  |                            0 |                            0 |
| "AC004951.4"  |                            0 |                            0 |
| "AC005094.1"  |                            0 |                            0 |
| "AC006030.1"  |                            0 |                            0 |
| "AC006041.2"  |                            0 |                            0 |
| "AC006159.1"  |                            0 |                            0 |
| "AC006946.3"  |                            0 |                            0 |
| "AC007402.1"  |                            0 |                            0 |
| "AC007666.2"  |                            0 |                            0 |
| "AC008415.1"  |                            0 |                            0 |
| "AC008507.1"  |                            0 |                            0 |
| "AC008687.1"  |                            0 |                            0 |
| "AC008892.1"  |                            0 |                            0 |
| "AC008966.1"  |                            0 |                            0 |
| "AC009081.1"  |                            0 |                            0 |
| "AC009093.10" |                            0 |                            0 |
| "AC009522.1"  |                            0 |                            0 |
| "AC010183.2"  |                            0 |                            0 |
+---------------+------------------------------+------------------------------+----------

+-----------------------+-----------------------+-----------------------+-----------------
| 'HGDP01269'.n_variants | 'HGDP00241'.n_variants | 'HGDP00110'.n_variants | 'LP6005441-DNA_F0
+-----------------------+-----------------------+-----------------------+-----------------
|                  int64 |                  int64 |                  int64 |
+-----------------------+-----------------------+-----------------------+-----------------
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
|                      0 |                      0 |                      0 |
+------------------------+------------------------+------------------------+------------------
showing top 24 rows
showing the first 10 of 392 columns
```

5.0.1 Exercise

Is this a dense (mostly non-zero) or sparse (mostly zero) matrix? Is this expected? How many variants are in our dataset, and how many genes are there?

6 Step 4: Run linear regression per gene

This should look familiar! We can reuse the same modular components (like `linear_regression_rows`) for many different purposes.

[13]:
```
pca_eigenvalues, pca_scores, pca_loadings = hl.hwe_normalized_pca(mt.GT,␣
↪compute_loadings=True)
```

```
2021-06-16 17:17:55 Hail: INFO: hwe_normalized_pca: running PCA using 1329
variants.
2021-06-16 17:18:16 Hail: INFO: pca: running PCA with 10 components…
2021-06-16 17:18:53 Hail: INFO: Coerced sorted dataset
```

[14]:
```
burden_mt = burden_mt.annotate_cols(pca = pca_scores[burden_mt.s])

burden_results = hl.linear_regression_rows(
    y=burden_mt.sample_data.sleep_duration,
    x=burden_mt.n_variants,
    covariates=[1.0,
                burden_mt.pca.scores[0],
                burden_mt.pca.scores[1],
                burden_mt.pca.scores[2]])
```

```
2021-06-16 17:19:17 Hail: INFO: Coerced sorted dataset
2021-06-16 17:19:28 Hail: INFO: Ordering unsorted dataset with network shuffle
2021-06-16 17:19:36 Hail: INFO: linear_regression_rows: running on 392 samples
for 1 response variable y,
    with input variable x, and 4 additional covariates…
```

6.1 Sorry, no `hl.plot.manhattan` for genes!

Manhattan plots are really only useful for standard GWAS. Instead, we can simply sort by p-value using order_by, and print:

[15]:
```
burden_results.order_by(burden_results.p_value).show()
```

```
+----------------+-------+----------+--------------+-----------+----------------+-----------+-
| gene_symbol    |     n |    sum_x | y_transpose_x |      beta | standard_error |    t_stat |
+----------------+-------+----------+--------------+-----------+----------------+-----------+-
| str            | int32 |  float64 |      float64 |   float64 |        float64 |   float64 |
+----------------+-------+----------+--------------+-----------+----------------+-----------+-
| "ASB5"         |   392 | 4.00e+00 |     3.40e+01 |  2.41e+00 |       7.15e-01 |  3.37e+00 |
| "AC002070.1"   |   392 | 5.00e+00 |     4.10e+01 |  2.10e+00 |       6.41e-01 |  3.27e+00 |
| "AC079313.1"   |   392 | 6.00e+00 |     4.80e+01 |  1.91e+00 |       5.86e-01 |  3.26e+00 |
| "SLC25A28"     |   392 | 6.00e+00 |     4.70e+01 |  1.73e+00 |       5.88e-01 |  2.95e+00 |
| "LOC105378068" |   392 | 2.00e+00 |     1.80e+01 |  2.89e+00 |       1.01e+00 |  2.86e+00 |
| "RBAK-RBAKDN"  |   392 | 6.00e+00 |     4.30e+01 |  1.79e+00 |       6.30e-01 |  2.83e+00 |
| "CRAT37"       |   392 | 5.00e+00 |     2.20e+01 | -1.75e+00 |       6.44e-01 | -2.72e+00 |
| "ANO1"         |   392 | 1.00e+00 |     1.00e+01 |  3.86e+00 |       1.43e+00 |  2.70e+00 |
| "LINC00964"    |   392 | 1.00e+00 |     1.00e+01 |  3.86e+00 |       1.43e+00 |  2.70e+00 |
| "SGCZ"         |   392 | 1.00e+00 |     2.00e+00 | -3.87e+00 |       1.43e+00 | -2.70e+00 |
| "PDE1C"        |   392 | 4.00e+00 |     2.80e+01 |  2.33e+00 |       8.72e-01 |  2.68e+00 |
| "GAREM1"       |   392 | 6.00e+00 |     4.60e+01 |  1.57e+00 |       5.89e-01 |  2.67e+00 |
| "DPP6"         |   392 | 5.00e+00 |     3.90e+01 |  1.70e+00 |       6.44e-01 |  2.64e+00 |
| "AC142381.3"   |   392 | 5.00e+00 |     3.90e+01 |  1.69e+00 |       6.44e-01 |  2.62e+00 |
| "LRBA"         |   392 | 2.00e+00 |     7.00e+00 | -2.61e+00 |       1.01e+00 | -2.58e+00 |
| "SLAMF1"       |   392 | 2.00e+00 |     7.00e+00 | -2.55e+00 |       1.01e+00 | -2.51e+00 |
| "AC008507.1"   |   392 | 4.00e+00 |     1.60e+01 | -1.84e+00 |       7.41e-01 | -2.48e+00 |
| "LOC105370114" |   392 | 3.00e+00 |     2.30e+01 |  2.12e+00 |       8.74e-01 |  2.43e+00 |
| "LINC02268"    |   392 | 1.00e+00 |     9.00e+00 |  3.44e+00 |       1.44e+00 |  2.38e+00 |
| "MORC4"        |   392 | 1.00e+00 |     9.00e+00 |  3.44e+00 |       1.44e+00 |  2.38e+00 |
| "LOC107984976" |   392 | 6.00e+00 |     4.50e+01 |  1.39e+00 |       5.90e-01 |  2.35e+00 |
| "ANK1"         |   392 | 1.00e+00 |     9.00e+00 |  3.45e+00 |       1.47e+00 |  2.35e+00 |
| "MAGI2"        |   392 | 1.00e+00 |     9.00e+00 |  3.45e+00 |       1.47e+00 |  2.35e+00 |
| "SLC35F3"      |   392 | 1.00e+00 |     9.00e+00 |  3.45e+00 |       1.47e+00 |  2.35e+00 |
+----------------+-------+----------+--------------+-----------+----------------+-----------+-
showing top 24 rows
```

A Q-Q plot is still meaningful on genes, though! Let's plot one:

[16]:
```
p = hl.plot.qq(burden_results.p_value)
show(p)
```

```
2021-06-16 17:19:48 Hail: INFO: Ordering unsorted dataset with network shuffle
```

With fewer tests performed (one per gene, instead of one per variant), the X and Y range of the Q-Q plot is much smaller than in the common variant association practical.

This plot is showing us that although our study is relatively well-controlled, it's also very underpowered!

7 The end

You've reached the end of the prepared Hail practical materials! Congratulations – we didn't really expect anyone to make it this far in the allotted time!

If you have questions, please ask the faculty! We are eager to discuss Hail and how it might be of assistance in your science.

8 When the workshop ends and you return to your life

The hosted notebook service that is running this notebook will be turned off in a few hours, but you can continue using Hail!

The Hail website has a page with information about getting started. If you have a MacOS or Linux computer, or have access to a Linux server, you can run Hail.

It is also possible to run Hail on Google Cloud. See the video lectures for guidance on how to do that, or reach out to the team for help!

8.1 The Hail community

Although Hail has a steeper learning curve than many command-line tools, you won't be learning it alone! Hail has a forum and Zulip chatroom full of like-minded users of all experience levels. Please stop by to say hello!