# Hail GWAS analysis notebook

**Apr 26, 2020**

# Contents

1 List of example notebooks 3

A repository for keeping conda env and docker image file for running single cell analysis using [Hail.is](https://hail.is).

# CHAPTER 1

## List of example notebooks

## 1.1 Hail - GWAS Tutorial

### 1.1.1 Load required libraries and initialize

[1]:
```
import hail as hl
from hail.plot import show
from pprint import pprint
import os
from IPython.display import HTML
hl.init()
hl.plot.output_notebook()
```

Data type cannot be displayed: application/javascript, application/vnd.bokehjs_load.v0+json

[2]:
```
hl.spark_context()
```

[2]:
```
<SparkContext master=local[*] appName=pyspark-shell>
```

#### Get SparkUI url for Binder

[59]:
```
spark_ui = '<a>Use SparkUI from sparkContext</a>'
if os.environ.get('JUPYTERHUB_USER') is not None:
  spark_ui = \
    '<a href="https://hub.gke.mybinder.org/user/{0}/proxy/4040/jobs/">Spark UI for␣
↪Binder</a>'.\
      format(os.environ.get('JUPYTERHUB_USER'))
HTML(spark_ui)
```

[59]:
```
<IPython.core.display.HTML object>
```

### 1.1.2 Download public 1000 Genomes data

[4]:
```
hl.utils.get_1kg('/tmp/data/')
```

### 1.1.3 Importing data from VCF

[5]:
```
hl.import_vcf('/tmp/data/1kg.vcf.bgz').write('/tmp/data/1kg.mt', overwrite=True)
```

[8]:
```
mt = hl.read_matrix_table('/tmp/data/1kg.mt')
```

### 1.1.4 Explore data

[9]:
```
mt.rows().select().show(5)
```

```
+---------------+------------+
| locus         | alleles    |
+---------------+------------+
| locus<GRCh37> | array<str> |
+---------------+------------+
| 1:904165      | ["G","A"]  |
| 1:909917      | ["G","A"]  |
| 1:986963      | ["C","T"]  |
| 1:1563691     | ["T","G"]  |
| 1:1707740     | ["T","G"]  |
+---------------+------------+
showing top 5 rows
```

[10]:
```
mt.row_key.show(5)
```

```
+---------------+------------+
| locus         | alleles    |
+---------------+------------+
| locus<GRCh37> | array<str> |
+---------------+------------+
| 1:904165      | ["G","A"]  |
| 1:909917      | ["G","A"]  |
| 1:986963      | ["C","T"]  |
| 1:1563691     | ["T","G"]  |
| 1:1707740     | ["T","G"]  |
+---------------+------------+
showing top 5 rows
```

[11]:
```
mt.s.show(5)
```

```
+-----------+
| s         |
+-----------+
| str       |
+-----------+
```

(continues on next page)

(continued from previous page)

```
| "HG00096" |
| "HG00099" |
| "HG00105" |
| "HG00118" |
| "HG00129" |
+-----------+
showing top 5 rows
```

[12]:
```
mt.entry.take(5)
```

[12]:
```
[Struct(GT=Call(alleles=[0, 0], phased=False), AD=[4, 0], DP=4, GQ=12, PL=[0, 12,␣
↪147]),
 Struct(GT=Call(alleles=[0, 0], phased=False), AD=[8, 0], DP=8, GQ=24, PL=[0, 24,␣
↪315]),
 Struct(GT=Call(alleles=[0, 0], phased=False), AD=[8, 0], DP=8, GQ=23, PL=[0, 23,␣
↪230]),
 Struct(GT=Call(alleles=[0, 0], phased=False), AD=[7, 0], DP=7, GQ=21, PL=[0, 21,␣
↪270]),
 Struct(GT=Call(alleles=[0, 0], phased=False), AD=[5, 0], DP=5, GQ=15, PL=[0, 15,␣
↪205])]
```

### 1.1.5 Adding column fields

[15]:
```
table = (hl.import_table('/tmp/data/1kg_annotations.txt', impute=True)
         .key_by('Sample'))
```

[16]:
```
table.describe()
```

```
----------------------------------------
Global fields:
    None
----------------------------------------
Row fields:
    'Sample': str
    'Population': str
    'SuperPopulation': str
    'isFemale': bool
    'PurpleHair': bool
    'CaffeineConsumption': int32
----------------------------------------
Key: ['Sample']
----------------------------------------
```

[17]:
```
table.show(width=100)
```

```
+-----------+------------+-----------------+----------+------------+------------------
↪---+
| Sample    | Population | SuperPopulation | isFemale | PurpleHair |␣
↪CaffeineConsumption |
+-----------+------------+-----------------+----------+------------+------------------
↪---+
| str       | str        | str             |     bool |       bool |              ␣
↪int32 |
+-----------+------------+-----------------+----------+------------+------------------
↪---+
```

(continues on next page)

(continued from previous page)

```
| "HG00096" | "GBR"      | "EUR"           |    false |      false |                 ␣
↪ 4 |
| "HG00097" | "GBR"      | "EUR"           |     true |       true |                 ␣
↪ 4 |
| "HG00098" | "GBR"      | "EUR"           |    false |      false |                 ␣
↪ 5 |
| "HG00099" | "GBR"      | "EUR"           |     true |      false |                 ␣
↪ 4 |
| "HG00100" | "GBR"      | "EUR"           |     true |      false |                 ␣
↪ 5 |
| "HG00101" | "GBR"      | "EUR"           |    false |       true |                 ␣
↪ 1 |
| "HG00102" | "GBR"      | "EUR"           |     true |       true |                 ␣
↪ 6 |
| "HG00103" | "GBR"      | "EUR"           |    false |       true |                 ␣
↪ 5 |
| "HG00104" | "GBR"      | "EUR"           |     true |      false |                 ␣
↪ 5 |
| "HG00105" | "GBR"      | "EUR"           |    false |      false |                 ␣
↪ 4 |
+-----------+------------+-----------------+----------+------------+-----------------
↪---+
showing top 10 rows
```

[18]:
```
print(mt.col.dtype)
```

```
struct{s: str}
```

[19]:
```
mt = mt.annotate_cols(pheno = table[mt.s])
```

[20]:
```
mt.col.describe()
```

```
--------------------------------------------------------
Type:
        struct {
        s: str,
        pheno: struct {
            Population: str,
            SuperPopulation: str,
            isFemale: bool,
            PurpleHair: bool,
            CaffeineConsumption: int32
        }
    }
--------------------------------------------------------
Source:
    <hail.matrixtable.MatrixTable object at 0x7f0f70afe470>
Index:
    ['column']
--------------------------------------------------------
```

### 1.1.6 Using query functions and the Hail Expression Language

[21]:
```
pprint(table.aggregate(hl.agg.counter(table.SuperPopulation)))
```

```
{'AFR': 1018, 'AMR': 535, 'EAS': 617, 'EUR': 669, 'SAS': 661}
```

[22]:
```
pprint(table.aggregate(hl.agg.stats(table.CaffeineConsumption)))
```

```
{'max': 10.0,
 'mean': 3.9837142857142855,
 'min': -1.0,
 'n': 3500,
 'stdev': 1.7021055628070711,
 'sum': 13943.0}
```

[23]:
```
table.count()
```

[23]:
```
3500
```

[24]:
```
mt.count_cols()
```

[24]:
```
284
```

[25]:
```
mt.aggregate_cols(hl.agg.counter(mt.pheno.SuperPopulation))
```

[25]:
```
{'AFR': 76, 'EAS': 72, 'AMR': 34, 'SAS': 55, 'EUR': 47}
```

[26]:
```
pprint(mt.aggregate_cols(hl.agg.stats(mt.pheno.CaffeineConsumption)))
```

```
{'max': 9.0,
 'mean': 4.415492957746479,
 'min': 0.0,
 'n': 284,
 'stdev': 1.577763427465917,
 'sum': 1254.0}
```

[27]:
```
snp_counts = mt.aggregate_rows(hl.agg.counter(hl.Struct(ref=mt.alleles[0], alt=mt.
↪alleles[1])))
pprint(snp_counts)
```

```
{Struct(ref='C', alt='T'): 2436,
 Struct(ref='T', alt='G'): 468,
 Struct(ref='A', alt='T'): 76,
 Struct(ref='A', alt='C'): 454,
 Struct(ref='A', alt='G'): 1944,
 Struct(ref='G', alt='A'): 2387,
 Struct(ref='T', alt='A'): 79,
 Struct(ref='T', alt='C'): 1879,
 Struct(ref='G', alt='T'): 480,
 Struct(ref='C', alt='G'): 150,
 Struct(ref='C', alt='A'): 496,
 Struct(ref='G', alt='C'): 112}
```

[28]:
```
from collections import Counter
counts = Counter(snp_counts)
counts.most_common()
```

[28]:
```
[(Struct(ref='C', alt='T'), 2436),
 (Struct(ref='G', alt='A'), 2387),
 (Struct(ref='A', alt='G'), 1944),
 (Struct(ref='T', alt='C'), 1879),
 (Struct(ref='C', alt='A'), 496),
```

(continues on next page)

(continued from previous page)

```
 (Struct(ref='G', alt='T'), 480),
 (Struct(ref='T', alt='G'), 468),
 (Struct(ref='A', alt='C'), 454),
 (Struct(ref='C', alt='G'), 150),
 (Struct(ref='G', alt='C'), 112),
 (Struct(ref='T', alt='A'), 79),
 (Struct(ref='A', alt='T'), 76)]
```

[29]:
```
p = hl.plot.histogram(mt.DP, range=(0,30), bins=30, title='DP Histogram', legend='DP')
show(p)
```

Data type cannot be displayed: application/javascript, application/vnd.bokehjs_exec.v0+json

### 1.1.7 Quality Control

[30]:
```
mt.col.describe()
```

```
--------------------------------------------------------
Type:
        struct {
        s: str,
        pheno: struct {
            Population: str,
            SuperPopulation: str,
            isFemale: bool,
            PurpleHair: bool,
            CaffeineConsumption: int32
        }
    }
--------------------------------------------------------
Source:
    <hail.matrixtable.MatrixTable object at 0x7f0f70afe470>
Index:
    ['column']
--------------------------------------------------------
```

[31]:
```
mt = hl.sample_qc(mt)
```

[32]:
```
mt.col.describe()
```

```
--------------------------------------------------------
Type:
        struct {
        s: str,
        pheno: struct {
            Population: str,
            SuperPopulation: str,
            isFemale: bool,
            PurpleHair: bool,
            CaffeineConsumption: int32
        },
        sample_qc: struct {
```

(continues on next page)

(continued from previous page)

```
            dp_stats: struct {
                mean: float64,
                stdev: float64,
                min: float64,
                max: float64
            },
            gq_stats: struct {
                mean: float64,
                stdev: float64,
                min: float64,
                max: float64
            },
            call_rate: float64,
            n_called: int64,
            n_not_called: int64,
            n_filtered: int64,
            n_hom_ref: int64,
            n_het: int64,
            n_hom_var: int64,
            n_non_ref: int64,
            n_singleton: int64,
            n_snp: int64,
            n_insertion: int64,
            n_deletion: int64,
            n_transition: int64,
            n_transversion: int64,
            n_star: int64,
            r_ti_tv: float64,
            r_het_hom_var: float64,
            r_insertion_deletion: float64
        }
    }
----------------------------------------------------------
Source:
    <hail.matrixtable.MatrixTable object at 0x7f0f709cfb00>
Index:
    ['column']
----------------------------------------------------------
```

### Plotting the QC metrics

[33]:
```
p = hl.plot.histogram(mt.sample_qc.call_rate, range=(.88,1), legend='Call Rate')
show(p)
```

Data type cannot be displayed: application/javascript, application/vnd.bokehjs_exec.v0+json

[34]:
```
p = hl.plot.histogram(mt.sample_qc.gq_stats.mean, range=(10,70), legend='Mean Sample␣
↪GQ')
show(p)
```

(continued from previous page)

Data type cannot be displayed: application/javascript, application/vnd.bokehjs_exec.v0+json

[35]:
```
p = hl.plot.scatter(mt.sample_qc.dp_stats.mean, mt.sample_qc.call_rate, xlabel='Mean␣
↪DP', ylabel='Call Rate')
show(p)
```

Data type cannot be displayed: application/javascript, application/vnd.bokehjs_exec.v0+json

[36]:
```
mt = mt.filter_cols((mt.sample_qc.dp_stats.mean >= 4) & (mt.sample_qc.call_rate >= 0.
↪97))
print('After filter, %d/284 samples remain.' % mt.count_cols())
```

```
After filter, 250/284 samples remain.
```

[37]:
```
ab = mt.AD[1] / hl.sum(mt.AD)

filter_condition_ab = ((mt.GT.is_hom_ref() & (ab <= 0.1)) |
                       (mt.GT.is_het() & (ab >= 0.25) & (ab <= 0.75)) |
                       (mt.GT.is_hom_var() & (ab >= 0.9)))

fraction_filtered = mt.aggregate_entries(hl.agg.fraction(~filter_condition_ab))
print(f'Filtering {fraction_filtered * 100:.2f}% entries out of downstream analysis.')
mt = mt.filter_entries(filter_condition_ab)
```

```
Filtering 3.64% entries out of downstream analysis.
```

[38]:
```
mt = hl.variant_qc(mt)
```

[39]:
```
mt.row.describe()
```

```
--------------------------------------------------------
Type:
        struct {
        locus: locus<GRCh37>,
        alleles: array<str>,
        rsid: str,
        qual: float64,
        filters: set<str>,
        info: struct {
            AC: array<int32>,
            AF: array<float64>,
            AN: int32,
            BaseQRankSum: float64,
            ClippingRankSum: float64,
            DP: int32,
            DS: bool,
            FS: float64,
            HaplotypeScore: float64,
            InbreedingCoeff: float64,
            MLEAC: array<int32>,
            MLEAF: array<float64>,
            MQ: float64,
            MQ0: int32,
```

(continues on next page)

(continued from previous page)

```
            MQRankSum: float64,
            QD: float64,
            ReadPosRankSum: float64,
            set: str
        },
        variant_qc: struct {
            dp_stats: struct {
                mean: float64,
                stdev: float64,
                min: float64,
                max: float64
            },
            gq_stats: struct {
                mean: float64,
                stdev: float64,
                min: float64,
                max: float64
            },
            AC: array<int32>,
            AF: array<float64>,
            AN: int32,
            homozygote_count: array<int32>,
            call_rate: float64,
            n_called: int64,
            n_not_called: int64,
            n_filtered: int64,
            n_het: int64,
            n_non_ref: int64,
            het_freq_hwe: float64,
            p_value_hwe: float64
        }
    }
----------------------------------------------------------
Source:
    <hail.matrixtable.MatrixTable object at 0x7f0f7097f390>
Index:
    ['row']
----------------------------------------------------------
```

### 1.1.8 GWAS Analysis

[40]:
```
mt = mt.filter_rows(mt.variant_qc.AF[1] > 0.01)
```

[41]:
```
mt = mt.filter_rows(mt.variant_qc.p_value_hwe > 1e-6)
```

[42]:
```
print('Samples: %d  Variants: %d' % (mt.count_cols(), mt.count_rows()))
```

```
Samples: 250  Variants: 7849
```

[43]:
```
gwas = hl.linear_regression_rows(y=mt.pheno.CaffeineConsumption,
                                 x=mt.GT.n_alt_alleles(),
                                 covariates=[1.0])
gwas.row.describe()
```

```
----------------------------------------------------------
Type:
        struct {
        locus: locus<GRCh37>,
        alleles: array<str>,
        n: int32,
        sum_x: float64,
        y_transpose_x: float64,
        beta: float64,
        standard_error: float64,
        t_stat: float64,
        p_value: float64
    }
----------------------------------------------------------
Source:
    <hail.table.Table object at 0x7f0f7092e358>
Index:
    ['row']
----------------------------------------------------------
```

[44]:
```
p = hl.plot.manhattan(gwas.p_value)
show(p)
```

Data type cannot be displayed: application/javascript, application/vnd.bokehjs_exec.v0+json

[45]:
```
p = hl.plot.qq(gwas.p_value)
show(p)
```

Data type cannot be displayed: application/javascript, application/vnd.bokehjs_exec.v0+json

### Confounded

[46]:
```
eigenvalues, pcs, _ = hl.hwe_normalized_pca(mt.GT)
```

[47]:
```
pprint(eigenvalues)
```

```
[18.02377947184681,
 9.98894555036326,
 3.538312262917122,
 2.6577590783730023,
 1.5966032147658327,
 1.5416611649602379,
 1.5029872248781773,
 1.4720816378531194,
 1.4678188487330746,
 1.4477835201334939]
```

[48]:
```
pcs.show(5, width=100)
```

```
+-----------+
| s         |
+-----------+
| str       |
+-----------+
| "HG00096" |
| "HG00099" |
| "HG00105" |
| "HG00118" |
| "HG00129" |
+-----------+

+-------------------------------------------------------------------------------------
↪-------------+
| scores
↪             |
+-------------------------------------------------------------------------------------
↪-------------+
| array<float64>
↪             |
+-------------------------------------------------------------------------------------
↪-------------+
| [-1.22e-01,-2.81e-01,1.11e-01,-1.28e-01,6.81e-02,-3.72e-03,-2.66e-02,4.99e-03,-9.
↪33e-02,-1.48... |
| [-1.13e-01,-2.90e-01,1.08e-01,-7.04e-02,4.20e-02,3.33e-02,1.61e-02,-1.15e-03,3.29e-
↪02,2.33e-02]  |
| [-1.08e-01,-2.80e-01,1.03e-01,-1.05e-01,9.40e-02,1.27e-02,3.14e-02,3.08e-02,1.06e-
↪02,-1.93e-02]  |
| [-1.25e-01,-2.98e-01,7.21e-02,-1.07e-01,2.89e-02,8.09e-03,-4.70e-02,-3.32e-02,-2.
↪59e-04,8.49e... |
| [-1.07e-01,-2.87e-01,9.72e-02,-1.16e-01,1.38e-02,1.87e-02,-8.37e-02,-4.87e-02,3.73e-
↪02,2.11e-02] |
+-------------------------------------------------------------------------------------
↪-------------+
showing top 5 rows
```

[49]:
```
mt = mt.annotate_cols(scores = pcs[mt.s].scores)
```

[50]:
```
p = hl.plot.scatter(mt.scores[0],
                    mt.scores[1],
                    label=mt.pheno.SuperPopulation,
                    title='PCA', xlabel='PC1', ylabel='PC2')
show(p)
```

Data type cannot be displayed: application/javascript, application/vnd.bokehjs_exec.v0+json

[51]:
```
gwas = hl.linear_regression_rows(
    y=mt.pheno.CaffeineConsumption,
    x=mt.GT.n_alt_alleles(),
    covariates=[1.0, mt.pheno.isFemale, mt.scores[0], mt.scores[1], mt.scores[2]])
```

[52]:
```
p = hl.plot.qq(gwas.p_value)
show(p)
```

Data type cannot be displayed: application/javascript, application/vnd.bokehjs_exec.v0+json

[53]:
```
p = hl.plot.manhattan(gwas.p_value)
show(p)
```

Data type cannot be displayed: application/javascript, application/vnd.bokehjs_exec.v0+json

### Rare variant analysis

[54]:
```
entries = mt.entries()
results = (entries.group_by(pop = entries.pheno.SuperPopulation, chromosome = entries.
↪locus.contig)
      .aggregate(n_het = hl.agg.count_where(entries.GT.is_het())))
```

[55]:
```
results.show()
```

```
+-------+------------+-------+
| pop   | chromosome | n_het |
+-------+------------+-------+
| str   | str        | int64 |
+-------+------------+-------+
| "AFR" | "1"        | 11276 |
| "AFR" | "10"       |  7160 |
| "AFR" | "11"       |  6875 |
| "AFR" | "12"       |  7048 |
| "AFR" | "13"       |  4678 |
| "AFR" | "14"       |  4313 |
| "AFR" | "15"       |  3904 |
| "AFR" | "16"       |  4593 |
| "AFR" | "17"       |  3718 |
| "AFR" | "18"       |  4171 |
+-------+------------+-------+
showing top 10 rows
```

[56]:
```
entries = entries.annotate(maf_bin = hl.cond(entries.info.AF[0]<0.01, "< 1%",
                           hl.cond(entries.info.AF[0]<0.05, "1%-5%", ">5%")))

results2 = (entries.group_by(af_bin = entries.maf_bin, purple_hair = entries.pheno.
↪PurpleHair)
      .aggregate(mean_gq = hl.agg.stats(entries.GQ).mean,
                 mean_dp = hl.agg.stats(entries.DP).mean))
```

[57]:
```
results2.show()
```

```
+---------+-------------+----------+----------+
| af_bin  | purple_hair |  mean_gq |  mean_dp |
+---------+-------------+----------+----------+
| str     |        bool |  float64 |  float64 |
+---------+-------------+----------+----------+
| "1%-5%" |       false | 2.48e+01 | 7.43e+00 |
```

(continues on next page)

(continued from previous page)

```
| "1%-5%" |        true | 2.46e+01 | 7.47e+00 |
| "< 1%"  |       false | 2.35e+01 | 7.55e+00 |
| "< 1%"  |        true | 2.35e+01 | 7.53e+00 |
| ">5%"   |       false | 3.70e+01 | 7.65e+00 |
| ">5%"   |        true | 3.73e+01 | 7.70e+00 |
+---------+-------------+----------+----------+
```

### 1.1.9 Citation

[58]:
```
hl.citation()
```

[58]:
```
'Hail Team. Hail 0.2.38-9d0be8b84f35. https://github.com/hail-is/hail/commit/
↪9d0be8b84f35.'
```

[ ]: