All by All

Common and rare variant association testing in 250,000 whole genomes across diverse ancestry groups from *All of Us*

Wenhan Lu
Neale and Karczewski Labs
Broad Institute
May 31st, 2024

Introduction to *All of Us*

All of Us – 2023 update

ENROLLMENT

>750k **Consented** Participants

and **>515k** **Core** Participants

Unprecedented diversity, with

81% UBR

and **46% by RE**

>530k Participant **biosamples** suitable for DNA sequencing

Enrolled **FIRST pediatric participant!**

RESEARCH

>8,400

Total Researchers (exceeding our goal toward 10,000 by 2026)

~250k

WGS made widely available for research

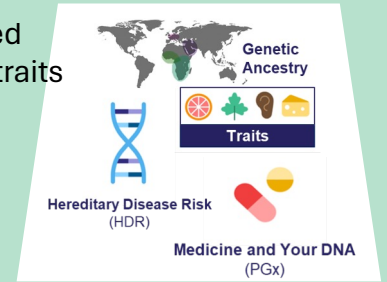
Signed agreements w/

31 **International Institutions** on **6 continents** to broaden researcher access, including **3** from LMICs

VALUE TO PARTICIPANTS

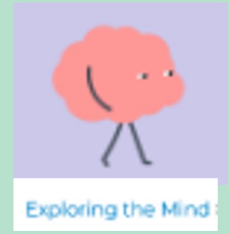
>170k participants received genetic ancestry & traits results

>90k participants received health-related genetic results



Launched **2 Ancillary Studies:**

Nutrition for Precision Health



...and celebrated our 5-year anniversary as a program!

All of Us – data types and sources

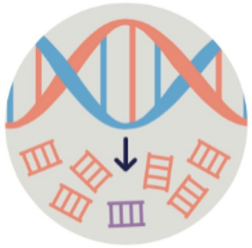
The *All of Us* Research Program's Data and Research Center (DRC) curates a range of different data types as part of the data collection process. As of April 2023, the *All of Us* Researcher Workbench contains the largest set of whole genome sequences widely available for research.

413,350+
Survey Responses

337,500+
Physical Measurements

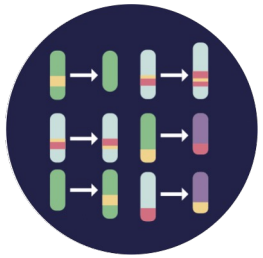
312,900+
Genotyping Arrays

287,000+
Electronic Health Records



245,350+
Whole Genome Sequences

15,600+
Fitbit Records
NEW! Sleep Data



11,350+
Structural Variants
NEW! In 2023

1,000+
Long-Read Sequences
NEW! In 2023

Data as of April 2023

All by All

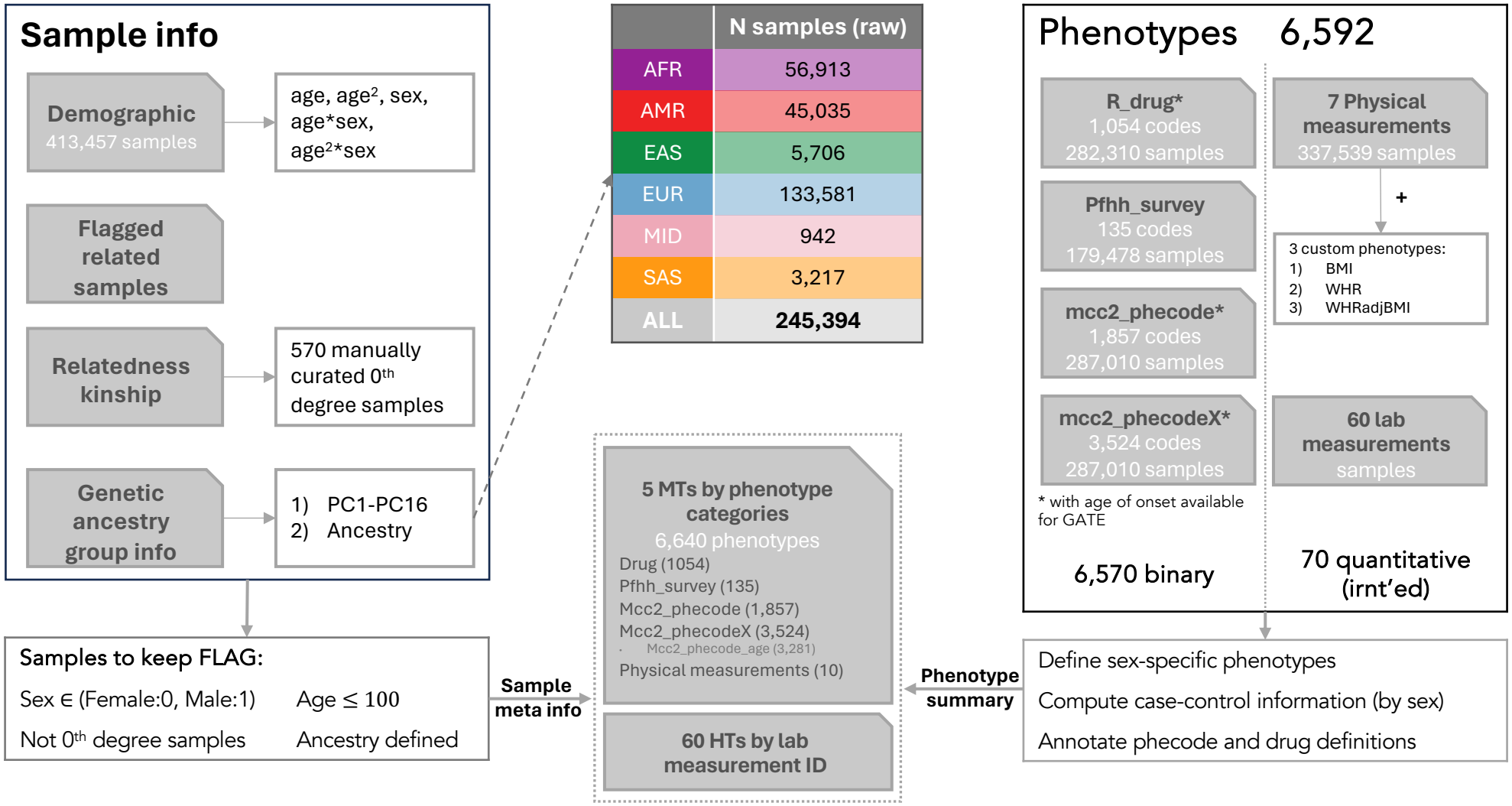
- *All of Us* cohort diversity enables:
  - Additional ancestries
  - Additional phenotypes
- Whole genomes enable:
  - Genome-wide association studies (GWAS), and
  - Rare-variant association studies (RVAS)

Genetic ancestry group	Num. individuals	Num. variants
AFR	56,913	383,702,267
AMR	45,035	334,390,971
EAS	5,706	122,729,124
EUR	133,581	628,935,579
MID	942	41,842,694
SAS	3,217	83,584,317
Total	245,394	1,116,593,592

AFR (African), AMR (Admixed American),
EAS (East Asian), EUR (European),
MID (Middle Eastern), and SAS (South Asian)

All by All pipeline

Phenotype & Sample initial processing



Robert Carroll

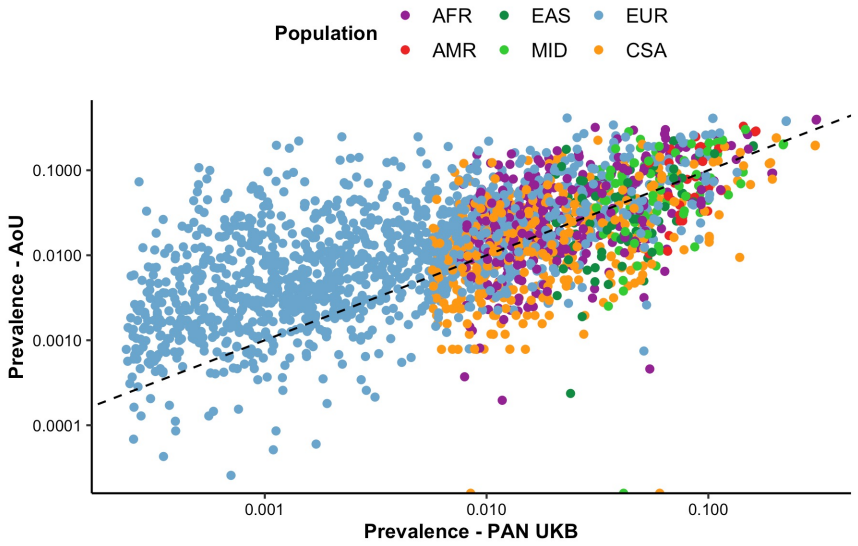
Phenotypes with Num. cases ≥ 200 per ancestry group

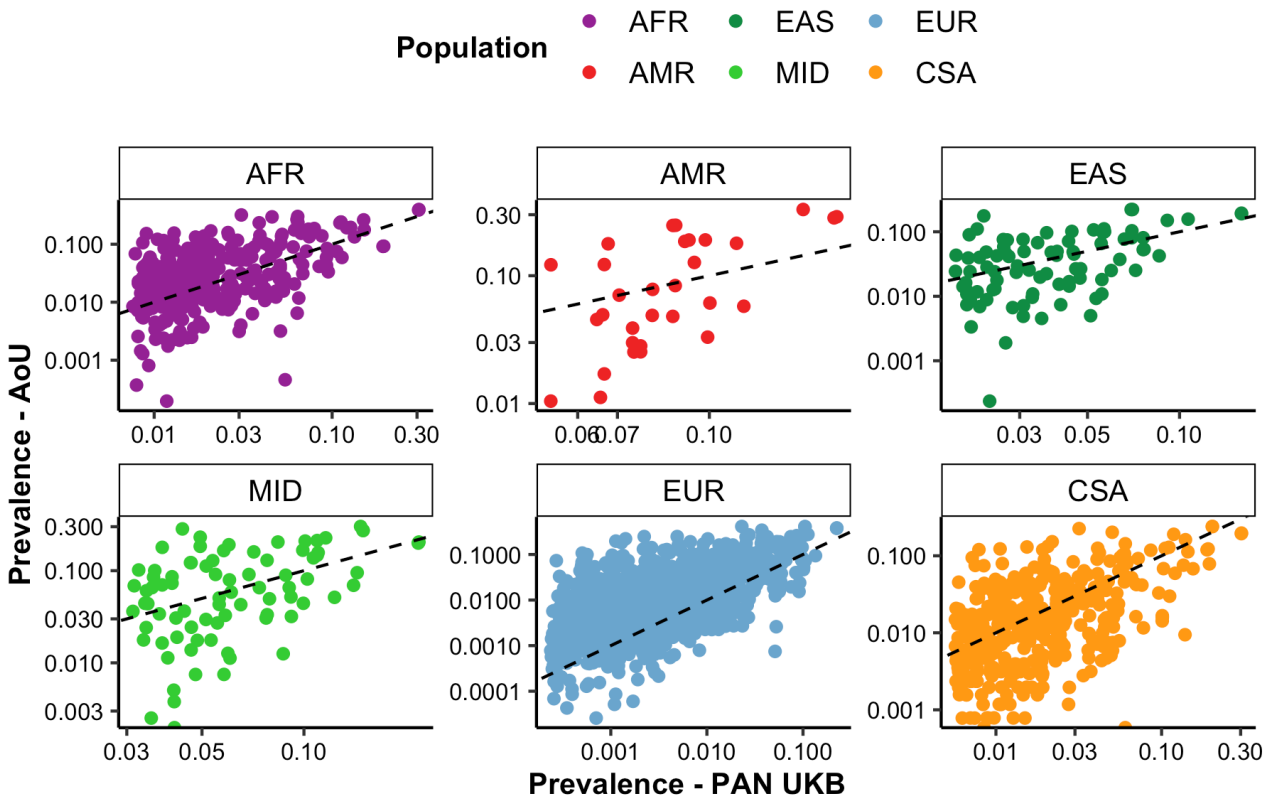
Category	AFR	AMR	EAS	EUR	MID	SAS
Lab measurements	55	53	45	58	23	45
Random phenotypes	30	30	30	30	30	30
mcc2 phecode	573	477	45	1,000	1	20
mcc2 phecodeX	869	776	80	1,361	0	42
r drug	758	715	357	857	42	288
pfhh survey	20	17	0	78	0	0
physical measurements	10	10	10	10	10	10
Total: 8,895 (3,417 unique)	**2,315**	**2,078**	**567**	**3,394**	**106**	**435**

Category	Number of genetic ancestry groups defined					
	6	5	4	3	2	1
Lab measurements	23	22		7	4	2
Random phenotypes	30					
mcc2 phecode	1	19	25	412	131	417
mcc2 phecodeX		41	40	651	167	476
r drug	42	246	69	349	58	96
pfhh survey				15	7	56
physical measurements	10					
Total (3,416)	106	328	134	1,434	367	1,047

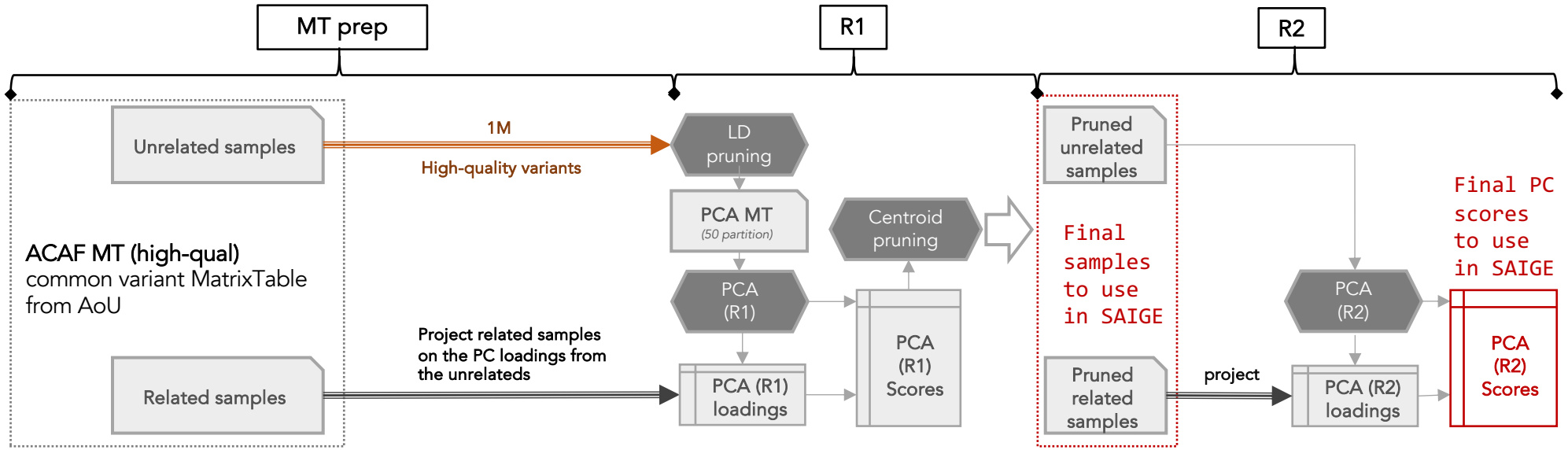
Robert Carroll

Phenotype prevalences are concordant between UK Biobank and *All of Us*



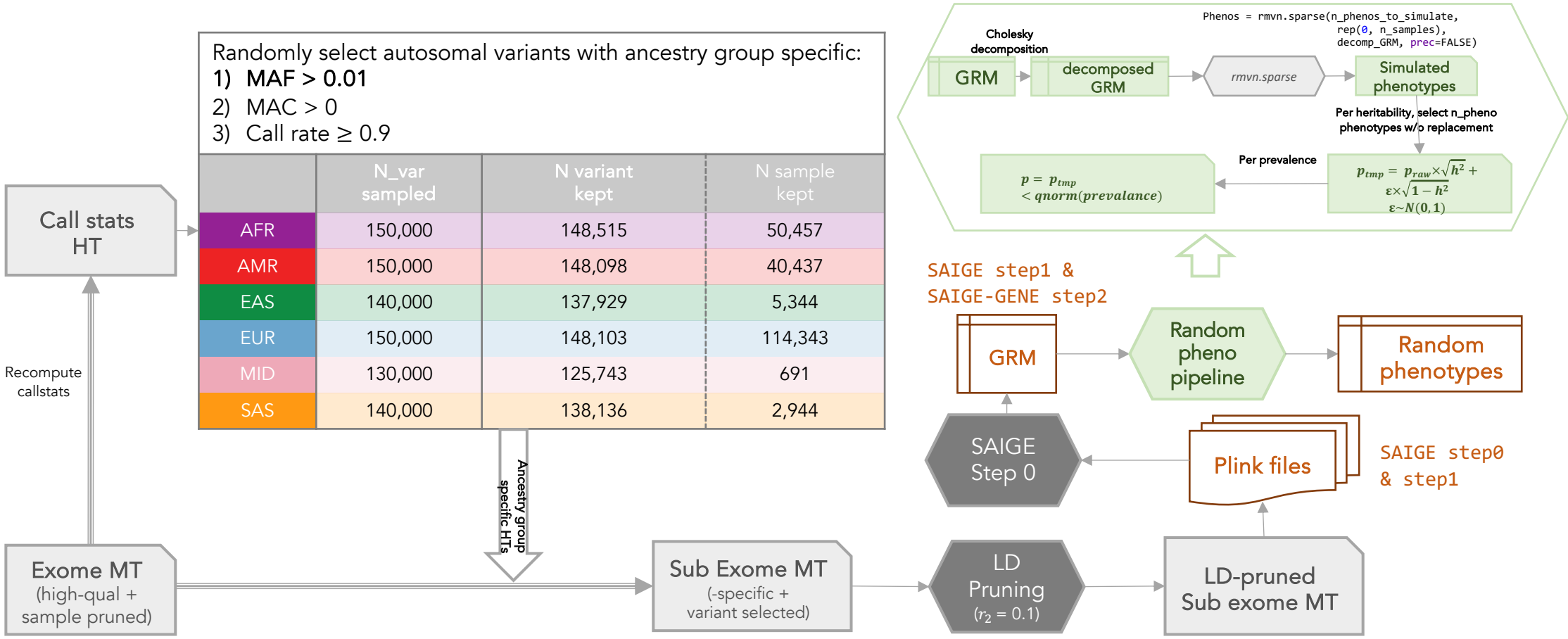


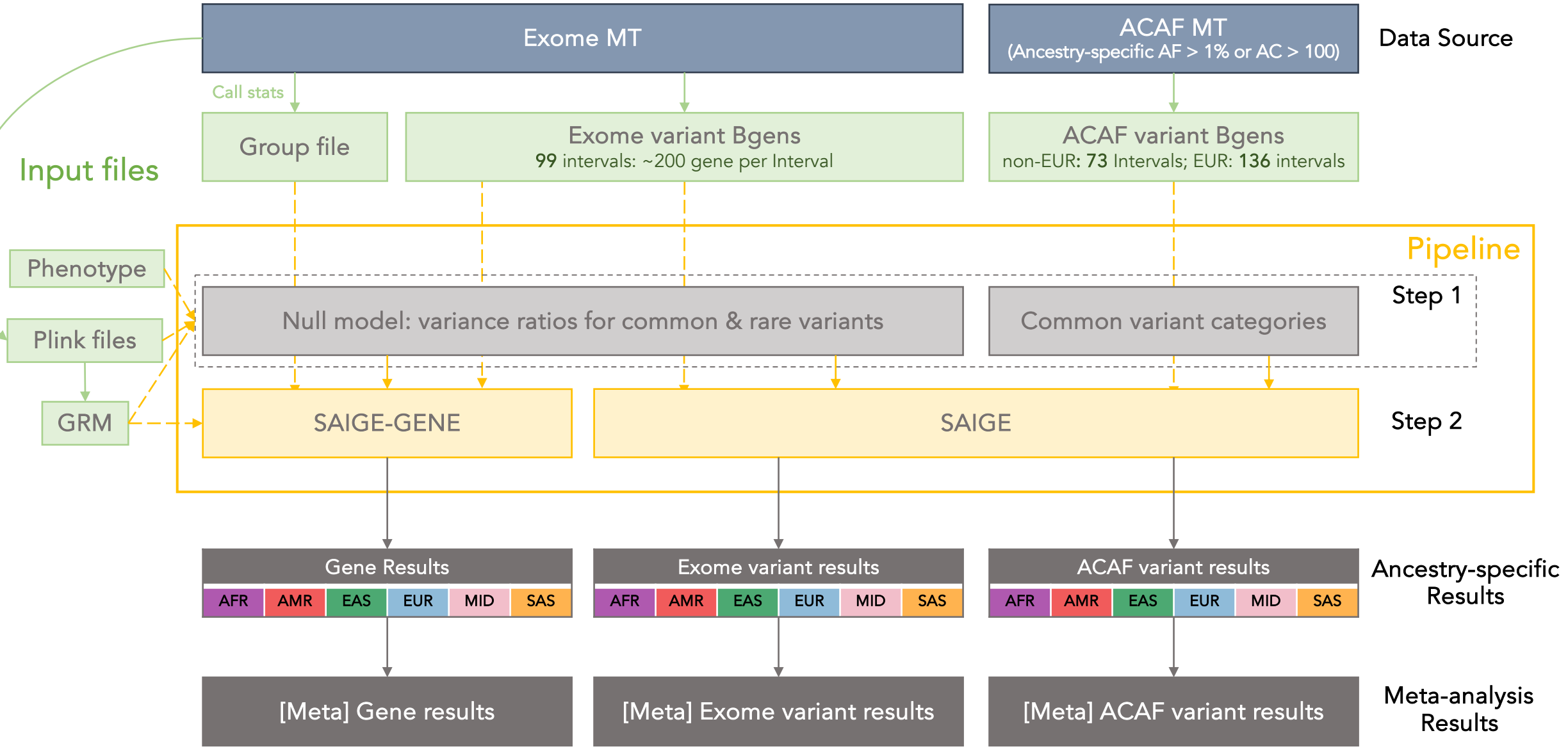
Sample selection to reduce stratification



Genetic ancestry group	N samples (raw)	N samples (high-qual)			N samples (pruned)		
		Unrelated	Related	Total	Unrelated	Related	Total
AFR	56,913	49,893	5,273	55,166	45,518	4,939	50,457 (91.5%)
AMR	45,035	40,327	3,816	44,143	36,887	3,550	40,437 (91.6%)
EAS	5,706	5,462	176	5,638	5,191	153	5,344 (94.8 %)
EUR	133,581	125,627	5,070	130,697	109,782	4,561	114,343 (87.5%)
MID	942	881	33	914	664	27	691 (75.6%)
SAS	3,217	3,080	96	3,176	2,856	88	2,944 (92.7%)

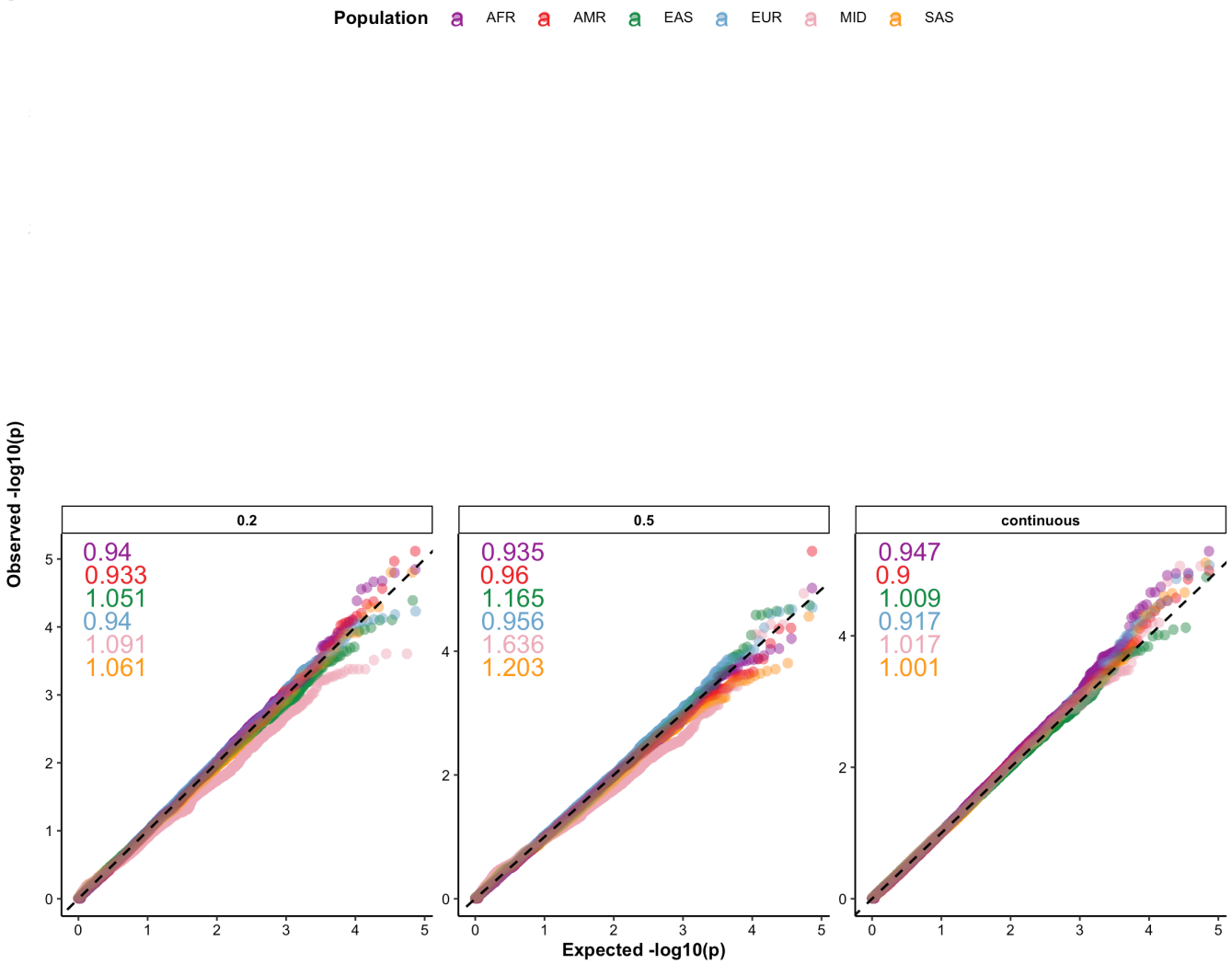
Random phenotypes & GRM




Exome MT
ACAF MT
(Ancestry-specific AF > 1% or AC > 100)
Data Source
Call stats
Input files
Group file
Exome variant Bgens
99 intervals: ~200 gene per Interval
ACAF variant Bgens
non-EUR: 73 Intervals; EUR: 136 intervals
Pipeline
Phenotype
Plink files
GRM
Null model: variance ratios for common & rare variants
Common variant categories
Step 1
SAIGE-GENE
SAIGE
Step 2
Gene Results
Exome variant results
ACAF variant results
AFR AMR EAS EUR MID SAS
Ancestry-specific Results
[Meta] Gene results
[Meta] Exome variant results
[Meta] ACAF variant results
Meta-analysis Results

Random phenotypes

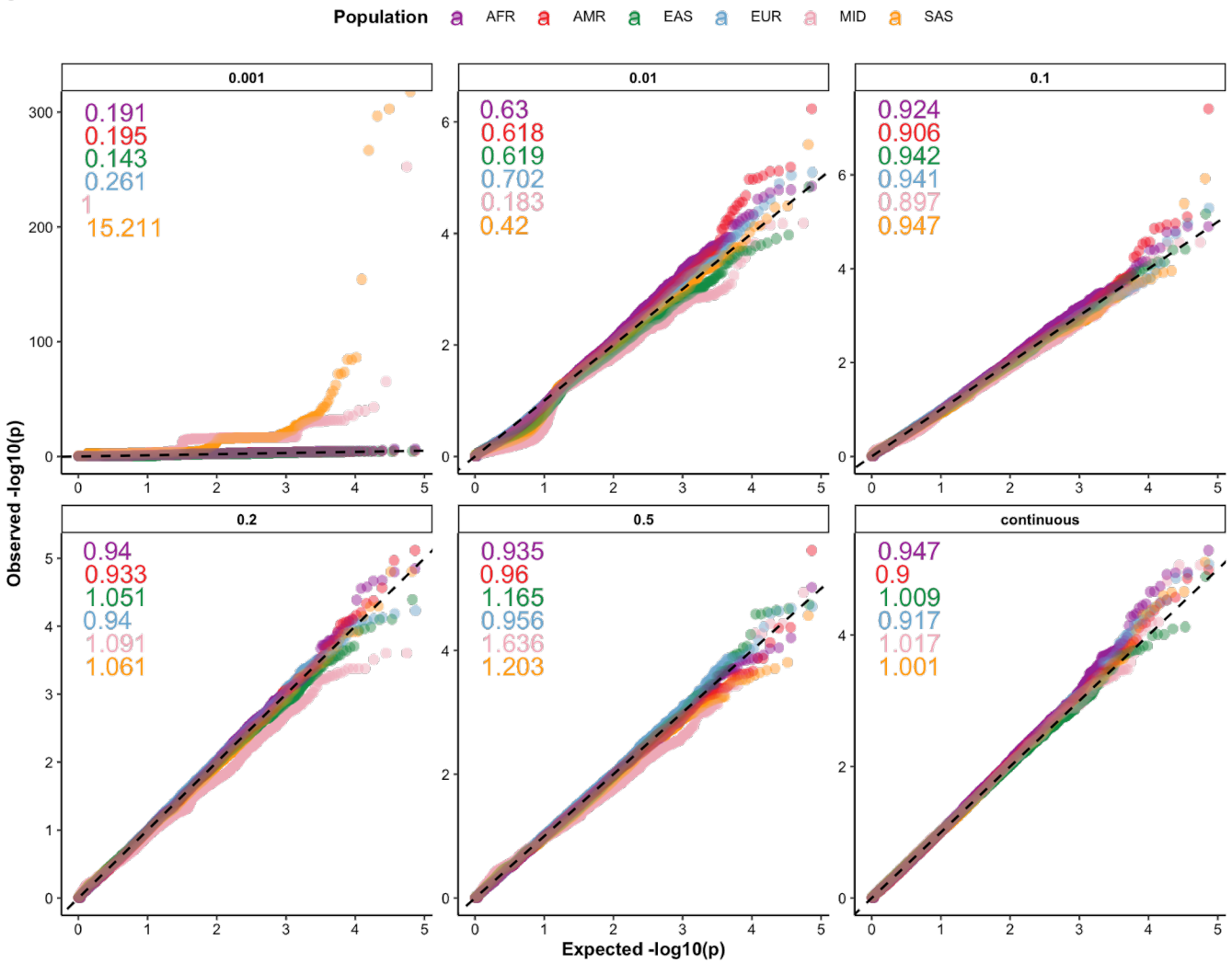
- Well-calibrated summary statistics for common binary traits (20-50%), and quantitative traits (right)



Numbers indicate genomic control (λ_{gc}) – closer to 1 indicates well-calibrated association tests

Random phenotypes

- Well-calibrated summary statistics for common binary traits (20-50%), and quantitative traits (right)
- Rare outcomes (<1%) deflated/underpowered for smaller ancestry groups



Numbers indicate genomic control (λ_{gc}) – closer to 1 indicates well-calibrated association tests

Pipeline stats

- **8,895** GWAS + RVAS (burden and SKAT testing with mixed models)
  - **7,787** in 3 largest genetic ancestry groups: EUR + AFR + AMR

$$\sim (\mathbf{99} + \mathbf{99} + \mathbf{73}) \text{ chunks}$$
$$\times$$
$$\mathbf{8,895} \text{ phenotypes}$$
$$\Downarrow$$
$$\sim \mathbf{2.7 \text{ million}} \text{ jobs}$$

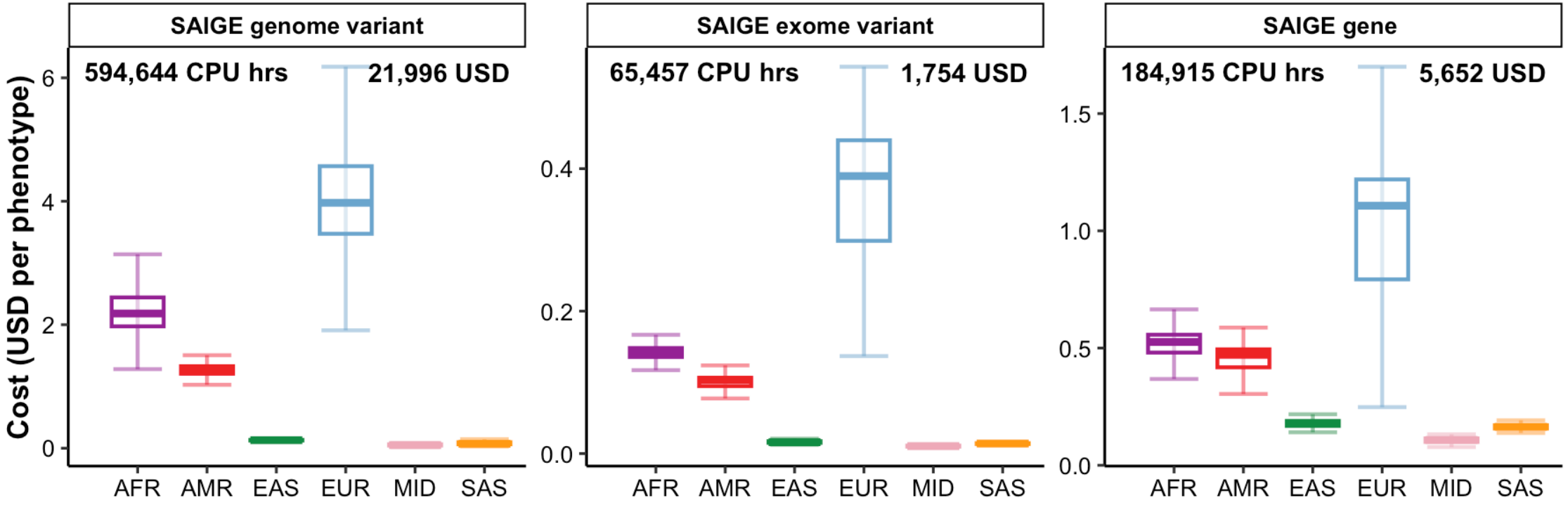
- **845K** CPU-hours
  - Implemented in Hail Batch: hail.is
  - Scalable cloud framework to run tens of thousands of CPUs at once


Daniel Goldstein


Jackie Goldstein

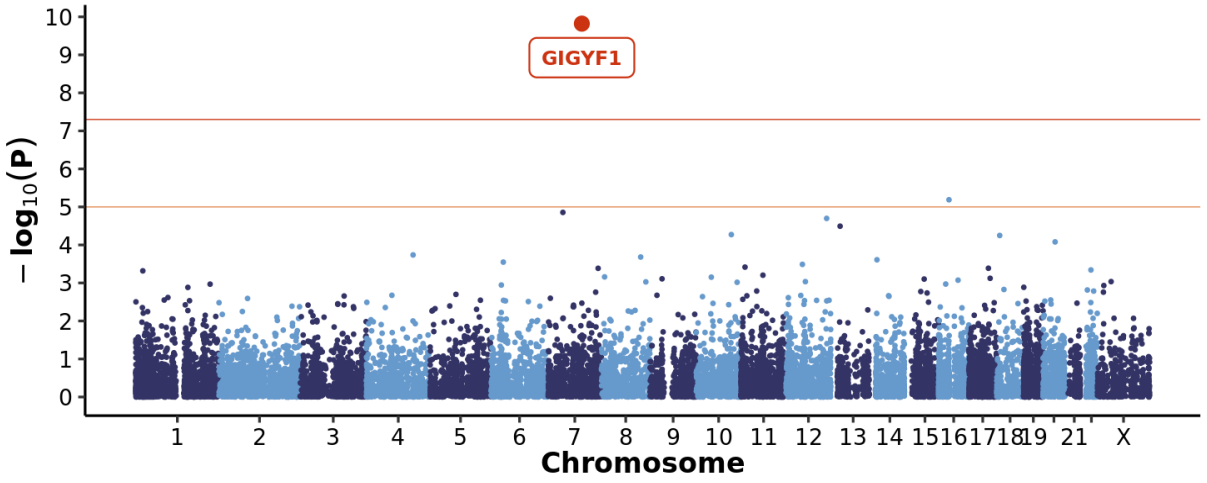

Daniel King

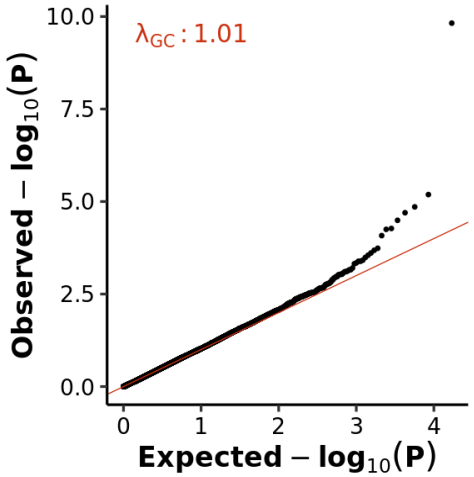
Cost per phenotype



Validate recent observation of rare pLoFs in GIGYF1 for T2D

Here, rare pLoF SKAT-O association with glucose levels





Article | Open access | Published: 03 November 2021

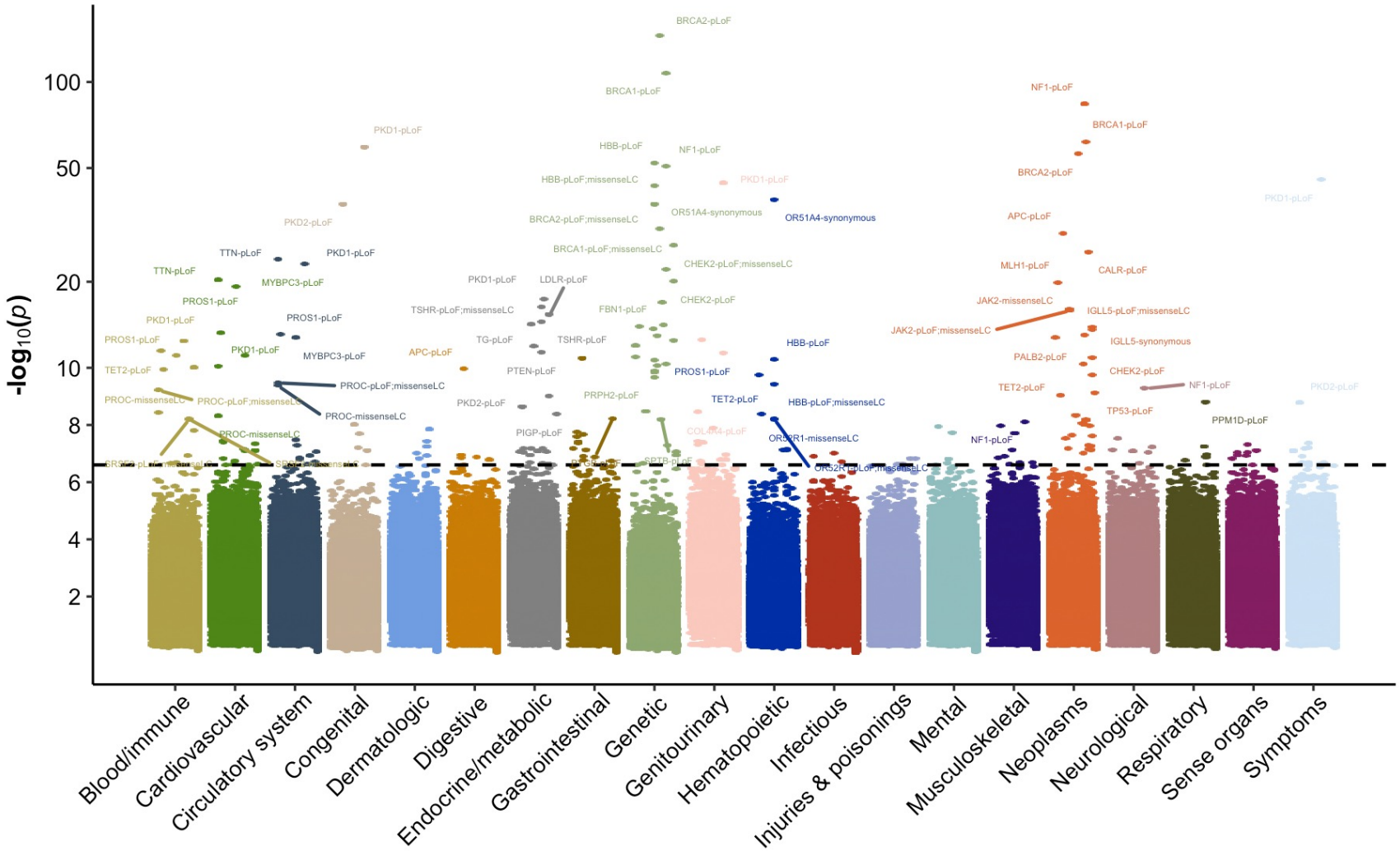
Gene-level analysis of rare variants in 379,066 whole exome sequences identifies an association of *GIGYF1* loss of function with type 2 diabetes

Aimee M. Deaton, Margaret M. Parker, Lucas D. Ward, Alexander O. Flynn-Carroll, Lucas BonDurant, Gregory Hinkle, Parsa Akbari, Luca A. Lotta, Regeneron Genetics Center, DiscovEHR Collaboration, Aris Baras & Paul Nioi

Scientific Reports **11**, Article number: 21565 (2021) | Cite this article

Lab measurements	Phecodes		Prescriptions
	Endocrine/metabolic	Dermatologic	A10A: Insulins and analogues A10B: Blood glucose lowering drugs, excluding insulins
Glucose (3004501)	• Diabetes mellitus (250 & EM_202) • T2D (250.2 & EM_202.2)	• Superficial cellulitis and abscess (681) • Cellulitis and abscess (DE_660.6)	A10, A10A, A10AB, A10AC, A10AD, A10AE, A10B, A10BA, A10BJ

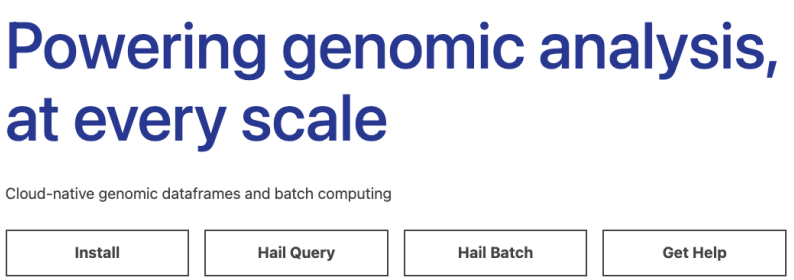
Associations across > 2K phecode + phecodeX



Burden test
Meta-analysis
Max MAF == 0.01

All by All data description

Hail - https://hail.is/



Hail query

- Simplified Analysis
- Genomic Dataframes

Hail batch

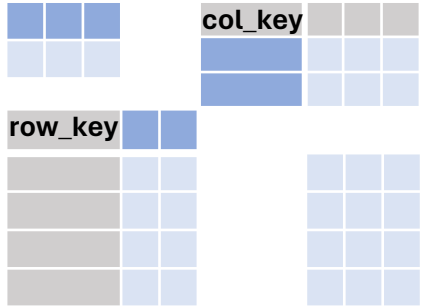
- Arbitrary tools
- Cost efficient and ease of use
- Scalability and Cost control

Hail Table (HT)

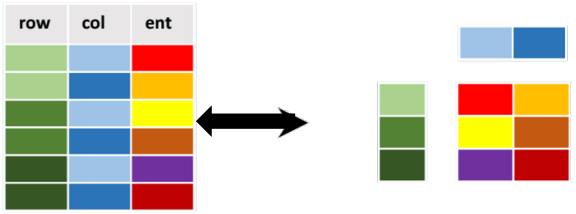
Hail MatrixTable (MT)

⭐https://hail.is/docs/0.2/cheatsheets.html

A MatrixTable is a Table with an extra dimension.



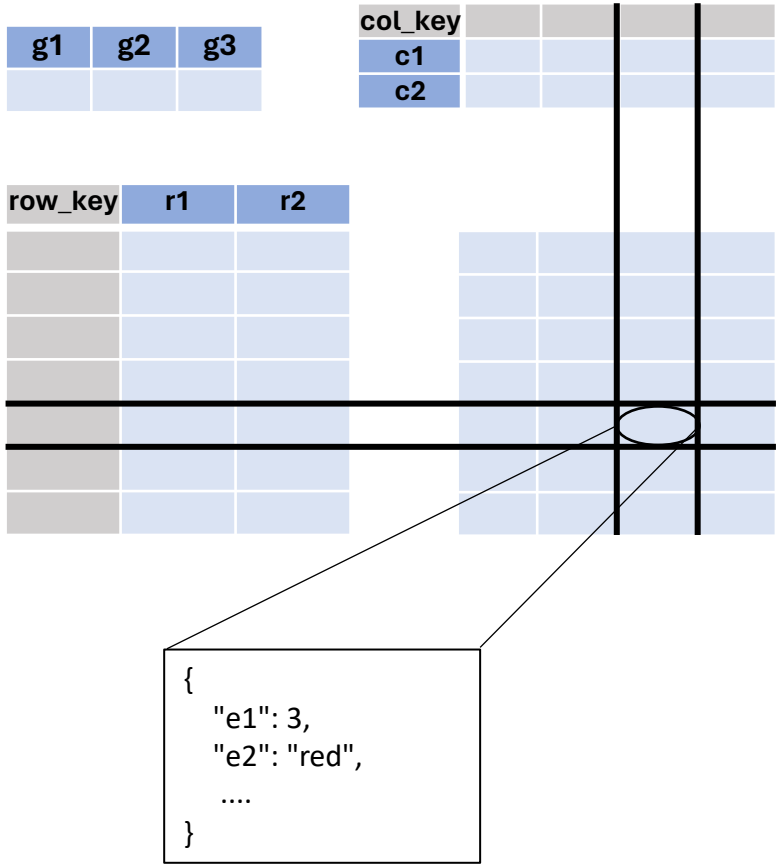
Hail Table ⇔ Hail Matrix Table



MatrixTable

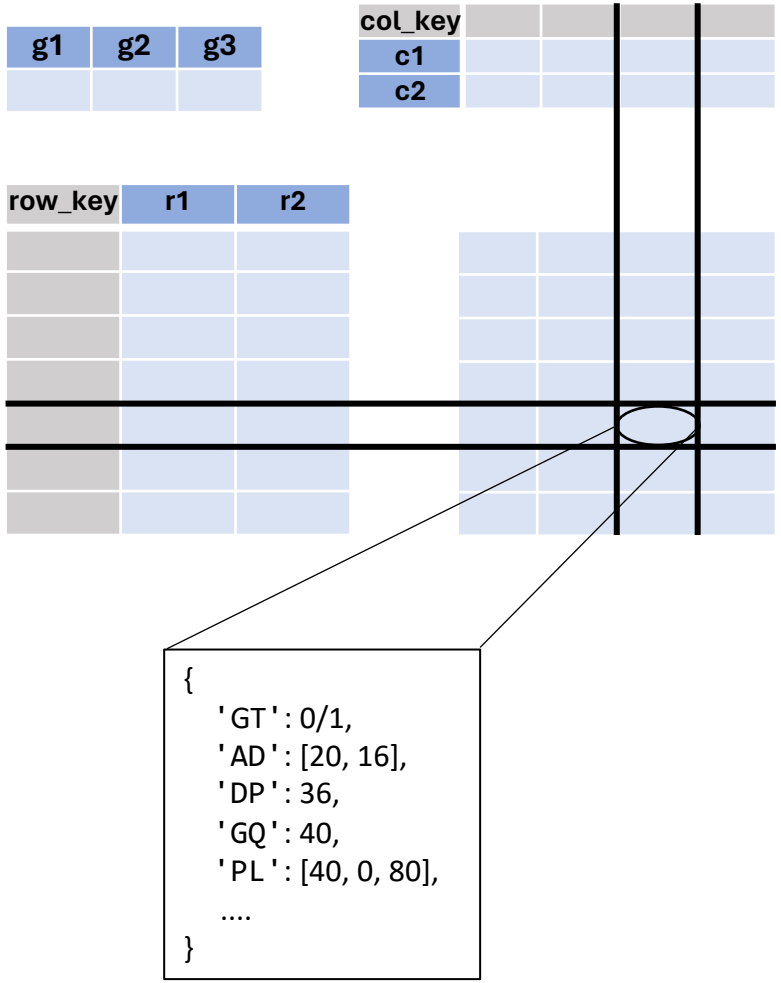
- Rows and columns, which each have their own set of fields
- Entries are structured fields indexed by these two dimensions

- For genetics applications:
  - Mimics VCF structure
  - Rows are variants
  - Columns are samples
  - Entries are genotypes



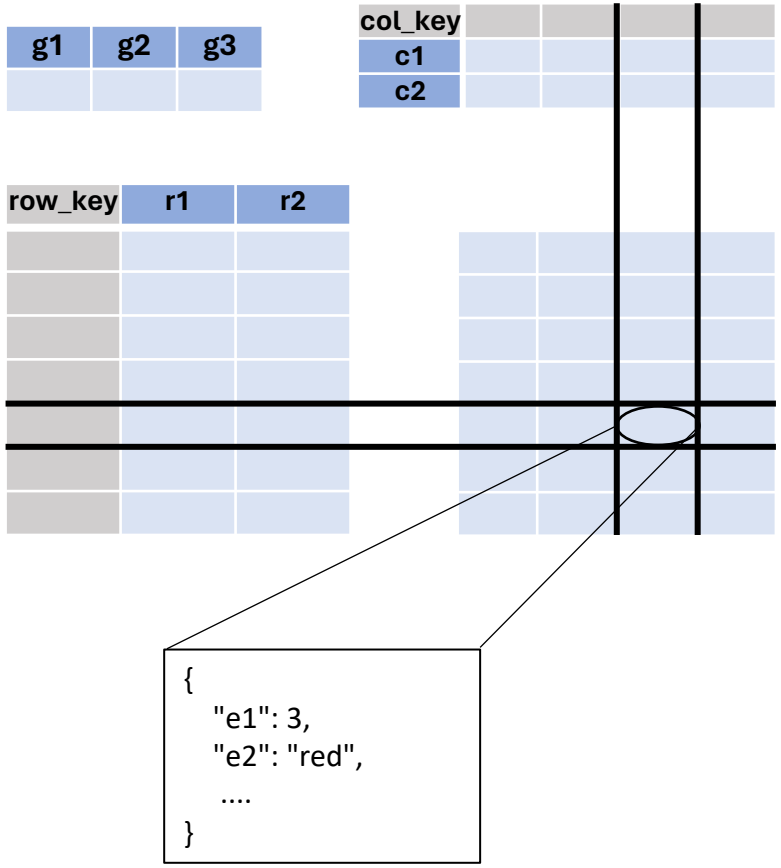
MatrixTable: genotype data

```
Row = {
  ‘locus’  : chr1:1000,
  ‘alleles’: [‘A’, ‘C’],
  ‘filters’: {},
  ‘vep’    : ...
}
Column = {
  ’s’: ‘NA12878’     # sample name
}
Entry = {
  ‘GT’: 0/1,         # genotype
  ‘AD’: [20, 16],    # allelic depth
  ‘DP’: 36,          # depth
  ‘GQ’: 40,          # genotype quality
  ‘PL’: [40, 0, 80]  # genotype likelihoods
}
```



MatrixTable: beyond genotype data

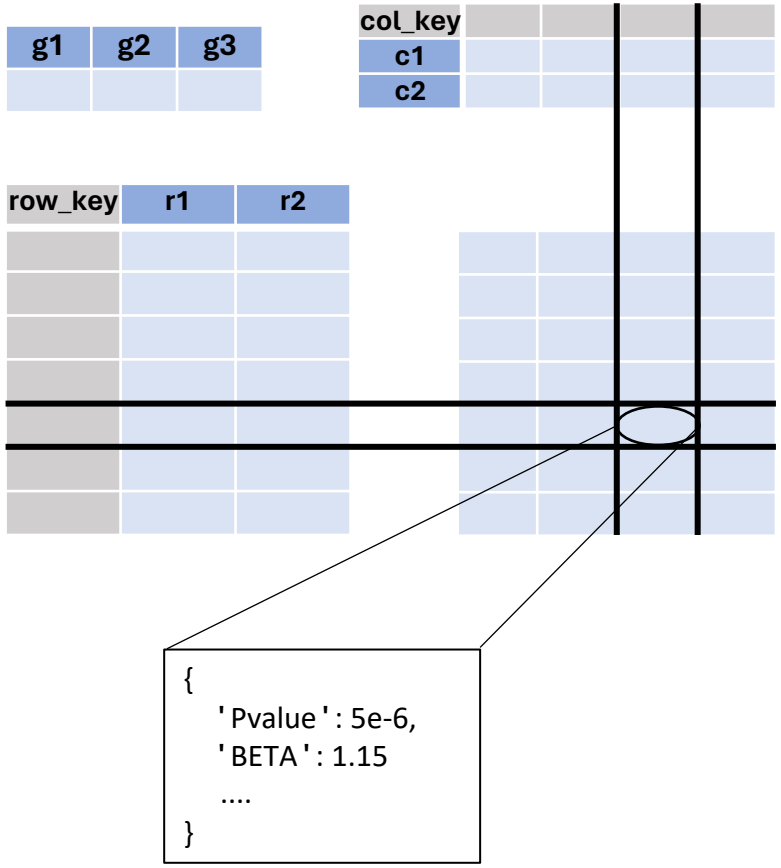
- Summary statistics can lend themselves to the MatrixTable structure as well
  - Rows are still variants
  - Columns are phenotypes
  - Entries are betas, SEs, p-values, etc.

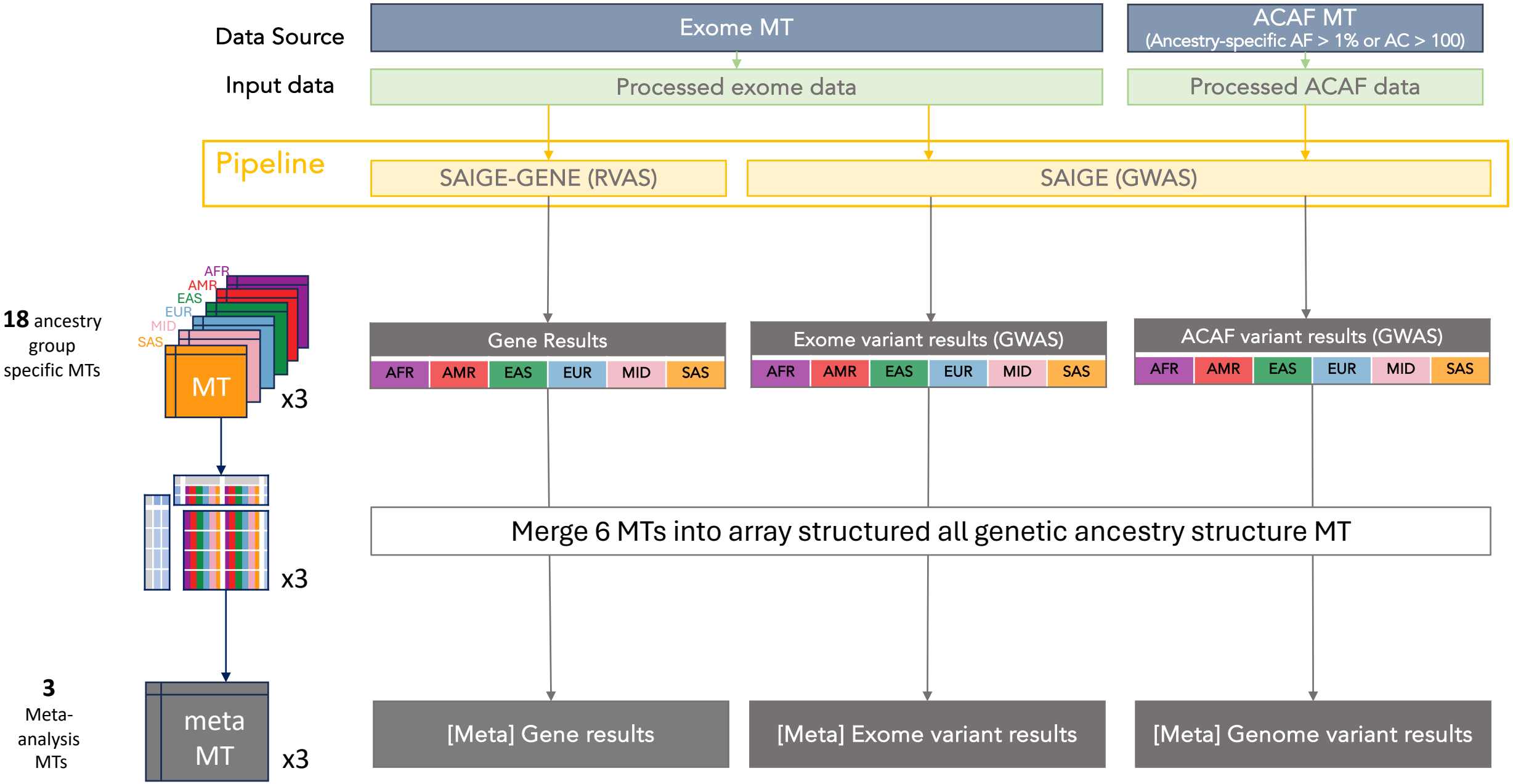


MatrixTable: beyond genotype data

```
Row = {
  ‘locus’  : chr1:1000,
  ‘alleles’: [‘A’, ‘C’],
  ‘MarkerID’: ...
}
Column = {
  ’phenoname’: ‘250.2’
  ’n_cases’: 5,004
  ‘n_controls’: 25,304
  ‘pheno_sex’: ‘Both’
  ‘trait_type’: ‘Binary’
  ...
}
```

```
Entry = {
    'AC_Allele2': int32
    'AF_Allele2': float64
    'MissingRate': float64
    'BETA': float64
    'SE': float64
    'Pvalue': float64
    ...
}
```




Data Source
Exome MT
ACAF MT
(Ancestry-specific AF > 1% or AC > 100)
Input data
Processed exome data
Processed ACAF data
Pipeline
SAIGE-GENE (RVAS)
SAIGE (GWAS)
18 ancestry group specific MTs
AFR
AMR
EAS
EUR
MID
SAS
MT
x3
Gene Results
AFR AMR EAS EUR MID SAS
Exome variant results (GWAS)
AFR AMR EAS EUR MID SAS
ACAF variant results (GWAS)
AFR AMR EAS EUR MID SAS
x3
Merge 6 MTs into array structured all genetic ancestry structure MT
3 Meta-analysis MTs
meta MT
x3
[Meta] Gene results
[Meta] Exome variant results
[Meta] Genome variant results

Genetic ancestry group specific **GWAS** MT

pop	…	…
‘afr’	…	…

phenoname	‘height’	‘174’	‘3028288’	‘D03BA’	…
n_cases	49,984	701	12,558	286	…
n_controls	NA	22,895	NA	40,019	…
pheno_sex	‘Both’	‘Female’	‘Both’	‘Both’	…
trait_type	‘continuous’	‘binary’	‘continuous’	‘continuous’	…
category	‘physical_measurement’	‘phecode’	‘lab_measurement’	‘r_drug’	…
description	‘height’	‘Breast cancer’	‘LDL’	‘Proteolytic enzymes’	…
…	…	…	…	…	…

locus	alleles	MarkerID	vep
chr1:13273	["G","C"]	‘chr1:13273_G/C’	struct{…}
chr1:13289	["CCT","C"]	‘chr1:13289_CCT/C’	struct{…}
chr1:13417	["C","CGAGA"]	‘chr1:13417_C/CGAGA’	struct{…}
chr1:14487	["G","A"]	chr1:14487_G/A	struct{…}
chr1:14488	["T","TC"]	chr1:14488_T/TC	struct{…}
chr1:14671	["G","C"]	chr1:14671_G/C	struct{…}
…	…	…	…

```
{
 'AC_Allele2': int32
 'AF_Allele2': float64
 'MissingRate': float64
 'BETA': float64
 'SE': float64
 'var': float64
 'p.value.NA': float64
 'Is.SPA': bool
 'AF_case': float64
 'AF_ctrl': float64
 'Pvalue': float64
}
```

Each entry:
Association test results for
1 phenotype x 1 variant
E.g.
chr1:14671:G:C x breast cancer

```
----------------------------------------
Global fields:
    None
----------------------------------------
Column fields:
    'phenoname': str
    'n_cases': int32
    'n_controls': int64
    'pheno_sex': str
    'trait_type': str
    'category': str
    'description': str
----------------------------------------
Row fields:
    'locus': locus<GRCh38>
    'alleles': array<str>
    'MarkerID': str
    'annotation': ...
    'quality_flags': struct {
        hq_exp_AC_variant: bool
    }
    'quality_flags_lambda': struct {
        hq_AF_variant: bool
    }
    'hq_variant': bool
    'hq_variant_lambda': bool
----------------------------------------
Entry fields:
    'AC_Allele2': int32
    'AF_Allele2': float64
    'MissingRate': float64
    'BETA': float64
    'SE': float64
    'var': float64
    'p.value.NA': float64
    'Is.SPA': bool
    'AF_case': float64
    'AF_ctrl': float64
    'Pvalue': float64
----------------------------------------
Column key: ['phenoname']
Row key: ['locus', 'alleles']
----------------------------------------
```

MT

Genetic ancestry group specific **Gene** MT

pop	…	…
‘afr’	…	…

phenoname	‘height’	‘174’	‘3028288’	‘D03BA’	…
n_cases	49,984	701	12,558	286	…
n_controls	NA	22,895	NA	40,019	…
pheno_sex	‘Both’	‘Female’	‘Both’	‘Both’	…
trait_type	‘continuous’	‘binary’	‘continuous’	‘continuous’	…
category	‘physical_measurement’	‘phecode’	‘lab_measurement’	‘r_drug’	…
description	‘height’	‘Breast cancer’	‘LDL’	‘Proteolytic enzymes’	…
…	…	…	…	…	…

gene_id	gene_symbol	annotation	max_MAF	…
ENSG00000000003	TSPAN6	pLoF	0.01	…
ENSG00000000003	TSPAN6	missenseLC	0.01	…
ENSG00000000003	TSPAN6	synonymous	0.01	…
ENSG00000000003	TSPAN6	pLoF;missenseLC	0.01	…
ENSG00000000003	TSPAN6	pLoF	0.001	…
ENSG00000000003	TSPAN6	pLoF	0.0001	…
…	…	…	…	…

```
['pLoF',
'missenseLC',
'synonymous',
'pLoF;missenseLC']
```

4 x annotations

```
[0.01,
0.001,
0.0001]
```

3 x max_MAF cutoffs

```
[SKATO,
 Burden,
 SKAT]
```

3 x Tests

```
{
  'Pvalue': float64
  'Pvalue_Burden': float64
  'Pvalue_SKAT': float64
  'Pvalue_log10': float64
  'Pvalue_Burden_log10': float64
  'Pvalue_SKAT_log10': float64
  'BETA_Burden': float64
  'SE_Burden': float64
  'MAC_case': int32
  'MAC_control': int32
  'total_variants_pheno': int32
  'hq_exp_CAC': bool}
```

Each entry:
Association test results for
1 phenotype x 1 gene-annotation-max_MAF
E.g.
pLoF variant with max_MAF 0.01% in gene TSPAN6 x breast cancer

```
----------------------------------------
Column fields:
    'n_cases': int64
    'n_controls': int64
    'heritability': float64
    'saige_version': str
    'inv_normalized': str
    'phenoname': str
    'pheno_sex': str
    'trait_type': str
    'category': str
    'description': str
    'phecode_category': str
    'description_more': str
    'lambda_gc': float64
----------------------------------------
Row fields:
    'gene_id': str
    'gene_symbol': str
    'annotation': str
    'max_MAF': float64
    'MAC': int32
    'Number_rare': int32
    'Number_ultra_rare': int32
    'total_variants': int32
    'interval': interval<locus<GRCh38>>
    'mean_coverage_raw': float64
    'mean_coverage_max_MAF': float64
    'CAF_raw': float64
    'CAF_max_MAF': float64
    'quality_flags': struct {
        hq_coverage: bool,
        hq_n_var_5: bool,
        hq_exp_CAC_gene: bool
    }
    'quality_flags_lambda': struct {
        hq_n_var_10: bool,
        hq_CAF: bool
    }
    'hq_gene': bool
    'hq_gene_lambda': bool
----------------------------------------
Column key: ['phenoname']
Row key: ['gene_id', 'gene_symbol',
'annotation', 'max_MAF']
----------------------------------------
```

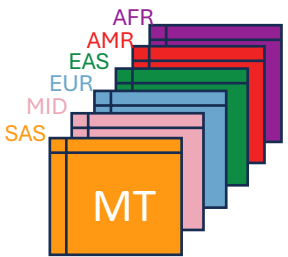
MT

Genetic ancestry group merged **GWAS** MT

global	…	…
…	…	…

phenoname		'height'						'174'			…
trait_type		'continuous'						'binary'			…
category		'physical_measurement'						'phecode'			…
description		'height'						'Breast cancer'			…
pheno_data	n_cases	49,984	38,971	5,245	111,755	680	2,897	701	535	3,622	…
	n_controls	NA	NA	NA	NA	NA	NA	22,895	20,983	55,967	…
	pheno_sex	'Both'	'Both'	'Both'	'Both'	'Both'	'Both'	'Female'	'Female'	'Female'	…
	pop	AFR	AMR	EAS	EUR	MID	SAS	AFR	AMR	EUR	…
…	…	…	…	…	…	…	…	…	…	…	…

locus	alleles		
chr1:13273	["G","C"]		
chr1:13289	["CCT","C"]		
chr1:13417	["C","CGAGA"]		
chr1:14487	["G","A"]		
chr1:14488	["T","TC"]		
chr1:14671	["G","C"]		
…	…	…	…




```
'summary_stats': array<struct {
       AC_Allele2: int32,
       AF_Allele2: float64,
       MissingRate: float64,
       BETA: float64,
       SE: float64,
       var: float64,
       `p.value.NA`: float64,
       `Is.SPA`: bool,
       AF_case: float64,
       AF_ctrl: float64,
       Pvalue: float64
   }>
```

Each entry:
Association test results for
1 phenotype x 1 variant x 6 ancesties in arrays
E.g.
chr1:14671:G:C x breast cancer

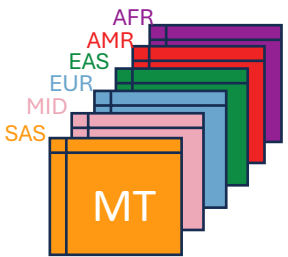
```
----------------------------------------
Column fields:
    'phenoname': str
    'pheno_data': array<struct {
         n_cases: int32
         n_controls: int64
         pheno_sex: str
         pop: str
    }>
    'description': str
    'trait_type': str
    'category': str
    'category_num': int32
----------------------------------------
Row fields:
    'locus': locus<GRCh38>
    'alleles': array<str>
    'MarkerID': str
    'annotation': ...
    'quality_flags': struct {
        hq_exp_AC_variant: bool
    }
    'quality_flags_lambda': struct {
        hq_AF_variant: bool
    }
    'hq_variant': bool
    'hq_variant_lambda': bool
----------------------------------------
Entry fields:
    'summary_stats': array<struct {
        AC_Allele2: int32,
        AF_Allele2: float64,
        MissingRate: float64,
        BETA: float64,
        SE: float64,
        var: float64,
        `p.value.NA`: float64,
        `Is.SPA`: bool,
        AF_case: float64,
        AF_ctrl: float64,
        Pvalue: float64
    }>
----------------------------------------
Column key: ['phenoname']
Row key: ['locus', 'alleles']
----------------------------------------
```

Merged FULL **Gene** MT

global	…	…
…	…	…

phenoname		'height'						'174'			…
trait_type		'continuous'						'binary'			…
category		'physical_measurement'						'phecode'			…
description		'height'						'Breast cancer'			…
pheno_data	n_cases	49,984	38,971	5,245	111,755	680	2,897	701	535	3,622	…
	n_controls	NA	NA	NA	NA	NA	NA	22,895	20,983	55,967	…
	pheno_sex	'Both'	'Both'	'Both'	'Both'	'Both'	'Both'	'Female'	'Female'	'Female'	…
	pop	AFR	AMR	EAS	EUR	MID	SAS	AFR	AMR	EUR	…
…	…	…	…	…	…	…	…	…	…	…	…

gene_id	gene_symbol	annotation	max_MAF	…
ENSG00000000003	TSPAN6	pLoF	0.01	…
ENSG00000000003	TSPAN6	missenseLC	0.01	…
ENSG00000000003	TSPAN6	synonymous	0.01	…
ENSG00000000003	TSPAN6	pLoF;missenseLC	0.01	…
ENSG00000000003	TSPAN6	pLoF	0.001	…
ENSG00000000003	TSPAN6	pLoF	0.0001	…
…	…	…	…	…




```
'summary_stats': array<struct {
        Pvalue: float64,
        Pvalue_Burden: float64,
        Pvalue_SKAT: float64,
        Pvalue_log10: float64,
        Pvalue_Burden_log10: float64,
        Pvalue_SKAT_log10: float64,
        BETA_Burden: float64,
        SE_Burden: float64,
        MAC_case: int32,
        MAC_control: int32,
        total_variants_pheno: int32
    }>
```

Each entry:

Association test results for

1 phenotype x 1 gene-annotation-max_MAF x 6 pops in arrays

E.g.

pLoF variant with max_MAF 0.01% in gene TSPAN6 x breast cancer

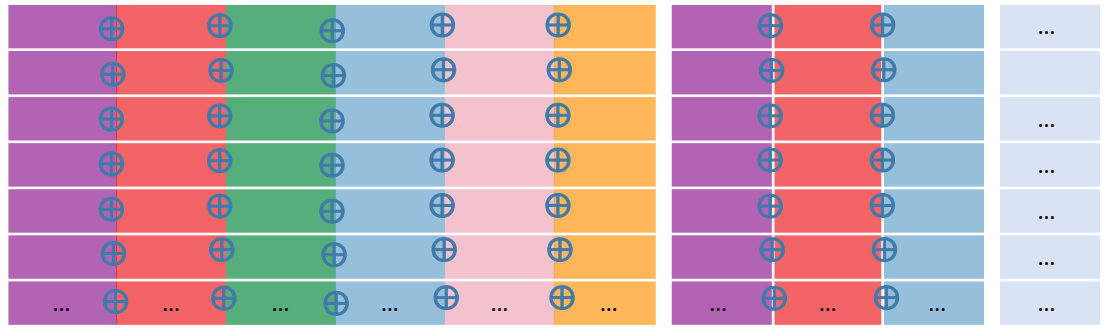
```
----------------------------------------
Global fields:
    None
----------------------------------------
Column fields:
    'phenoname': str
    'pheno_data': array<struct {
        n_cases: int64,
        n_controls: int64,
        pop: str
    }>
    'description': str
    'category': str
    'category_num': int32
    'sex': str
    'pheno_sex': str
    'trait_type': str
    'n_cases': int64
    'n_controls': int64
    'phecode_category': str
    'description_more': str
    'lambda_gc': float64
----------------------------------------
Row fields:
    'gene_id': str
    'gene_symbol': str
    'annotation': str
    'max_MAF': float64
    'MAC': int32
    'Number_rare': int32
    'Number_ultra_rare': int32
    'total_variants': int32
    'interval': interval<locus<GRCh38>>
    'mean_coverage_raw': float64
    'mean_coverage_max_MAF': float64
    'CAF_raw': float64
    'CAF_max_MAF': float64
    'quality_flags': struct {
        hq_coverage: bool,
        hq_n_var_5: bool,
        hq_exp_CAC_gene: bool
    }
    'quality_flags_lambda': struct {
        hq_n_var_10: bool,
        hq_CAF: bool
    }
    'hq_gene': bool
    'hq_gene_lambda': bool
----------------------------------------
Column key: ['phenoname']
Row key: ['gene_id', 'gene_symbol',
'annotation', 'max_MAF']
----------------------------------------
```

Meta-analysis **Gene** MT

global	…	…
…	…	…

phenoname		'height'						'174'			…
trait_type		'continuous'						'binary'			…
category		'physical_measurement'						'phecode'			…
description		'height'						'Breast cancer'			…
pheno_data	n_cases	49,984	38,971	5,245	111,755	680	2,897	701	535	3,622	…
	n_controls	NA	NA	NA	NA	NA	NA	22,895	20,983	55,967	…
	pheno_sex	'Both'	'Both'	'Both'	'Both'	'Both'	'Both'	'Female'	'Female'	'Female'	…
	pop	AFR	AMR	EAS	EUR	MID	SAS	AFR	AMR	EUR	…
…	…	…	…	…	…	…	…	…	…	…	…

gene_id	gene_symbol	annotation	max_MAF	…
ENSG00000000003	TSPAN6	pLoF	0.01	…
ENSG00000000003	TSPAN6	missenseLC	0.01	…
ENSG00000000003	TSPAN6	synonymous	0.01	…
ENSG00000000003	TSPAN6	pLoF;missenseLC	0.01	…
ENSG00000000003	TSPAN6	pLoF	0.001	…
ENSG00000000003	TSPAN6	pLoF	0.0001	…
…	…	…	…	…



```
'summary_stats': array<struct {
      Pvalue: float64,
      Pvalue_Burden: float64,
      Pvalue_SKAT: float64,
      Pvalue_log10: float64,
      Pvalue_Burden_log10: float64,
      Pvalue_SKAT_log10: float64,
      BETA_Burden: float64,
      SE_Burden: float64,
      MAC_case: int32,
      MAC_control: int32,
      total_variants_pheno: int32
   }>
```

Meta analysis

```
'weighted_Z_numerator_SKATO': array<float64>
'weighted_Z_numerator_Burden': array<float64>
'weighted_Z_numerator_SKAT': array<float64>
'META_Stats_SKATO': float64
'META_Pvalue_SKATO': float64
'META_Stats_Burden': float64
'META_Pvalue_Burden': float64
'META_Stats_SKAT': float64
'META_Pvalue_SKAT': float64
```

meta MT

```
----------------------------------------
Global fields:
    None
----------------------------------------
Column fields:
    'phenoname': str
    'pheno_data': array<struct {
        n_cases: int64,
        n_controls: int64,
        pop: str
    }>
    'description': str
    'category': str
    'category_num': int32
    'sex': str
    'pheno_sex': str
    'trait_type': str
    'n_cases': int64
    'n_controls': int64
    'phecode_category': str
    'description_more': str
    'lambda_gc': float64
----------------------------------------
Row fields:
    'gene_id': str
    'gene_symbol': str
    'annotation': str
    'max_MAF': float64
    'MAC': int32
    'Number_rare': int32
    'Number_ultra_rare': int32
    'total_variants': int32
    'interval': interval<locus<GRCh38>>
    'mean_coverage_raw': float64
    'mean_coverage_max_MAF': float64
    'CAF_raw': float64
    'CAF_max_MAF': float64
    'quality_flags': struct {
        hq_coverage: bool,
        hq_n_var_5: bool,
        hq_exp_CAC_gene: bool
    }
    'quality_flags_lambda': struct {
        hq_n_var_10: bool,
        hq_CAF: bool
    }
    'hq_gene': bool
    'hq_gene_lambda': bool
----------------------------------------
Column key: ['phenoname']
Row key: ['gene_id', 'gene_symbol',
'annotation', 'max_MAF']
----------------------------------------
```

Meta-analysis **Gene** MT

global	…	…
…	…	…

phenoname		'height'						'174'			…
trait_type		'continuous'						'binary'			…
category		'physical_measurement'						'phecode'			…
description		'height'						'Breast cancer'			…
pheno_data	n_cases	49,984	38,971	5,245	111,755	680	2,897	701	535	3,622	…
	n_controls	NA	NA	NA	NA	NA	NA	22,895	20,983	55,967	…
	pheno_sex	'Both'	'Both'	'Both'	'Both'	'Both'	'Both'	'Female'	'Female'	'Female'	…
	pop	AFR	AMR	EAS	EUR	MID	SAS	AFR	AMR	EUR	…
…	…	…	…	…	…	…	…	…	…	…	…

gene_id	gene_symbol	annotation	max_MAF	…									
ENSG00000000003	TSPAN6	pLoF	0.01	…									…
ENSG00000000003	TSPAN6	missenseLC	0.01	…									
ENSG00000000003	TSPAN6	synonymous	0.01	…									…
ENSG00000000003	TSPAN6	pLoF;missenseLC	0.01	…									…
ENSG00000000003	TSPAN6	pLoF	0.001	…									…
ENSG00000000003	TSPAN6	pLoF	0.0001	…									…
…	…	…	…	…	…								…

```
'summary_stats':array<struct{…}>
'META_Stats_SKAT0': float64
'META_Pvalue_SKAT0': float64
'META_Stats_Burden': float64
'META_Pvalue_Burden': float64
'META_Stats_SKAT': float64
'META_Pvalue_SKAT': float64
'hq_exp_CAC': bool
```

Each entry:

Meta –analysis results for

1 phenotype x 1 gene-annotation-max_MAF

E.g.

pLoF variant with max_MAF 0.01% in gene TSPAN6 x breast cancer

meta MT

```
----------------------------------------
Global fields:
    None
----------------------------------------
Column fields:
    'phenoname': str
    'pheno_data': array<struct {
        n_cases: int64,
        n_controls: int64,
        pop: str
    }>
    'description': str
    'category': str
    'category_num': int32
    'sex': str
    'pheno_sex': str
    'trait_type': str
    'n_cases': int64
    'n_controls': int64
    'phecode_category': str
    'description_more': str
    'lambda_gc': float64
----------------------------------------
Row fields:
    'gene_id': str
    'gene_symbol': str
    'annotation': str
    'max_MAF': float64
    'MAC': int32
    'Number_rare': int32
    'Number_ultra_rare': int32
    'total_variants': int32
    'interval': interval<locus<GRCh38>>
    'mean_coverage_raw': float64
    'mean_coverage_max_MAF': float64
    'CAF_raw': float64
    'CAF_max_MAF': float64
    'quality_flags': struct {
        hq_coverage: bool,
        hq_n_var_5: bool,
        hq_exp_CAC_gene: bool
    }
    'quality_flags_lambda': struct {
        hq_n_var_10: bool,
        hq_CAF: bool
    }
    'hq_gene': bool
    'hq_gene_lambda': bool
----------------------------------------
Column key: ['phenoname']
Row key: ['gene_id', 'gene_symbol',
'annotation', 'max_MAF']
----------------------------------------
```

Per phenotype HTs

```
----------------------------------------
Global fields:
    'n_cases': int32
    'n_controls': int32
    'heritability': float64
    'saige_version': str
    'inv_normalized': str
    'log_pvalue': bool
    'ranks': struct {
        values: array<float64>,
        ranks: array<int32>,
        _compaction_counts: array<int32>
    }
    'lambda_gc': float64
----------------------------------------
Row fields:
    'CHR': str
    'POS': int32
    'MarkerID': str
    'Allele1': str
    'Allele2': str
    'AC_Allele2': int32
    'AF_Allele2': float64
    'MissingRate': float64
    'BETA': float64
    'SE': float64
    'var': float64
    'p.value.NA': float64
    'Is.SPA': bool
    'AF_case': float64
    'AF_ctrl': float64
    'locus': locus<GRCh38>
    'alleles': array<str>
    'phenoname': str
    'Pvalue': float64
    'Pvalue_log10': float64
    'rank': int64
    'Pvalue_expected': float64
    'Pvalue_expected_log10': float64
----------------------------------------
Key: ['locus', 'alleles', 'phenoname']
----------------------------------------
```

locus	alleles	Pvalue	BETA	SE	…
chr1:13273	["G","C"]				…
chr1:13289	["CCT","C"]				…
chr1:13417	["C","CGAGA"]				…
chr1:14487	["G","A"]				…
chr1:14488	["T","TC"]				…
chr1:14671	["G","C"]				…
…	…	…	…	…	…

Single-variant HT

```
----------------------------------------
Global fields:
    'n_cases': int32
    'n_controls': int32
    'heritability': float64
    'saige_version': str
    'inv_normalized': str
    'lambda_gc_maxmaf_0.01': struct {…}
    'lambda_gc_maxmaf_0.001': struct {…}
    'lambda_gc_maxmaf_0.0001': struct {…}
    'lambda_gc_maxmaf_Cauchy': struct {…}
----------------------------------------
Row fields:
    'max_MAF': float64
    'Pvalue': float64
    'Pvalue_Burden': float64
    'Pvalue_SKAT': float64
    'BETA_Burden': float64
    'SE_Burden': float64
    'MAC': int32
    'Number_rare': int32
    'Number_ultra_rare': int32
    'Pvalue_log10': float64
    'Pvalue_Burden_log10': float64
    'Pvalue_SKAT_log10': float64
    'gene_id': str
    'gene_symbol': str
    'annotation': str
    'phenoname': str
    'total_variants': int32
    'interval': interval<locus<GRCh38>>
    'CHR': str
    'POS': int32
    'Pvalue_expected': float64
    'Pvalue_expected_log10': float64
    'Pvalue_Burden_expected': float64
    'Pvalue_Burden_expected_log10': float64
    'Pvalue_SKAT_expected': float64
    'Pvalue_SKAT_expected_log10': float64
----------------------------------------
Key: ['gene_id', 'gene_symbol', 'annotation', 'max_MAF']
----------------------------------------
```

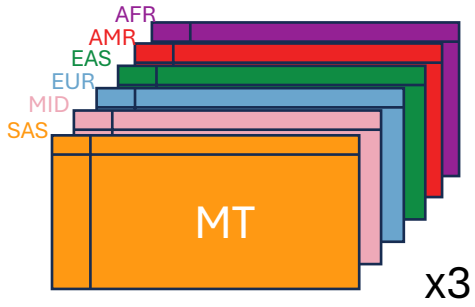
gene_id	gene_symbol	annotation	max_MAF	Pvalue_Burden	BETA_Burden	…
ENSG00000000003	TSPAN6	pLoF	0.01			…
ENSG00000000003	TSPAN6	missenseLC	0.01			…
ENSG00000000003	TSPAN6	synonymous	0.01			…
ENSG00000000003	TSPAN6	pLoF;missenseLC	0.01			…
ENSG00000000003	TSPAN6	pLoF	0.001			…
ENSG00000000003	TSPAN6	pLoF	0.0001			…
…	…	…	…			…

Gene-based HT

	AFR	AMR	EAS	EUR	MID	SAS	META	Total
N phenotypes	2315	2078	567	3394	106	435	3,417	12,312

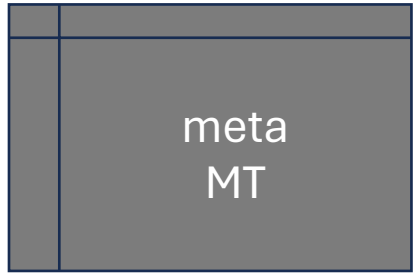
x3

36,936
phenotype result HTs



x3

18 ancestry specific MTs



x3

3 Meta-analysis MTs

21
result MTs

MT sizes

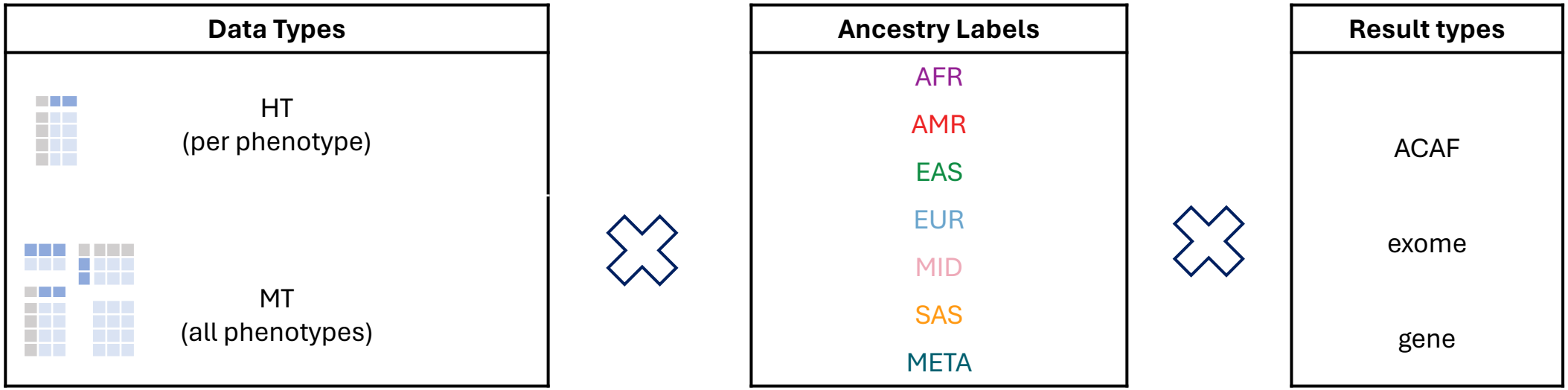
	AFR	AMR	EAS	EUR	MID	SAS	META	Total
ACAF variants	5.77 TiB	5.14 TiB	471.74 GiB	7.96 TiB	70.5 GiB	419.87 GiB	17.3 TiB	57.64 TiB
Exome variants	866.54 GiB	682.88 GiB	49.49 GiB	2.09 TiB	2.49 GiB	31.81 GiB	3.41 TiB	11.01 TiB
Gene	24.76 GiB	21.96 GiB	2.83 GiB	42.31 GiB	273.24 MiB	1.93 GiB	174.07 GiB	0.35 TiB
Total	6.64 TiB	5.83 TiB	0.51 TiB	10.09 TiB	0.07 TiB	0.44 TiB	22.88 TiB	69.00 TiB

Number of Statistics

	AFR	AMR	EAS	EUR	MID	SAS	Total
ACAF variants	117,869,385,057	105,686,088,355	11,973,000,433	162,836,771,233	2,260,313,022	11,223,305,413	411,848,863,513
Exome variants	18,995,833,334	15,033,381,624	1,340,301,444	45,459,117,460	82,063,134	914,822,029	81,825,519,025
Gene	552,050,991	493,406,568	86,426,673	823,662,984	12,205,120	62,649,994	2,030,402,330
Total	137,417,269,382	121,212,876,547	13,399,728,550	209,119,551,677	2,354,581,276	12,200,777,436	495,704,784,868

Demo: Querying All by All Data

Load Data



Load Hail Table (HT)

In [27]:
```
## ACAF HT: AFR height
ht1 = hl.read_table(get_ht_path(ancestry = 'afr', pheno = 'height', test_type = 'ACAF'))

# Exome HT: EUR height
ht2 = hl.read_table(get_ht_path(ancestry = 'eur', pheno = 'height', test_type = 'exome'))

# gene HT: AMR height
ht3 = hl.read_table(get_ht_path(ancestry = 'amr', pheno = 'height', test_type = 'gene'))
```

Load Hail MatrixTable (MT)

In [28]:
```
## ACAF MT: AFR
mt1 = hl.read_matrix_table(get_mt_path(ancestry = 'afr', test_type = 'ACAF'))

# Exome MT: EUR
mt2 = hl.read_matrix_table(get_mt_path(ancestry = 'eur', test_type = 'exome'))

# gene MT: AMR
mt3 = hl.read_matrix_table(get_mt_path(ancestry = 'amr', test_type = 'gene'))
```

Query per phenotype HTs (ACAF)

Exome HT should be similar

Query per phenotype HT

In [29]:
```
## ACAF HT: AFR height
ht1 = hl.read_table(get_ht_path(ancestry = 'afr', pheno = 'height', test_type = 'ACAF'))
ht1.describe()
```

```
----------------------------------------
Global fields:
    'n_cases': int32
    'n_controls': int32
    'heritability': float64
    'saige_version': str
    'inv_normalized': str
    'log_pvalue': bool
    'ranks': struct {
        values: array<float64>,
        ranks: array<int32>,
        _compaction_counts: array<int32>
    }
    'lambda_gc': float64
----------------------------------------
Row fields:
    'CHR': str
    'POS': int32
    'MarkerID': str
    'Allele1': str
    'Allele2': str
    'AC_Allele2': int32
    'AF_Allele2': float64
    'MissingRate': float64
    'BETA': float64
    'SE': float64
    'var': float64
    'p.value.NA': float64
    'Is.SPA': bool
    'AF_case': float64
    'AF_ctrl': float64
    'locus': locus<GRCh38>
    'alleles': array<str>
    'phenoname': str
    'Pvalue': float64
    'Pvalue_log10': float64
    'rank': int64
    'Pvalue_expected': float64
    'Pvalue_expected_log10': float64
----------------------------------------
Key: ['locus', 'alleles', 'phenoname']
----------------------------------------
```

In [40]:
```
ht1 = ht1.filter((ht1.AC_Allele2 > 100) & hl.is_snp(ht1.alleles[0], ht1.alleles[1])) # Filter rows
ht1.select('AC_Allele2', 'AF_Allele2', 'BETA', 'Pvalue', 'SE').show(20) # Select fields of interest and print
```

Hail | 1 EXECUTORS | 4 CORES | Jobs: | 2 COMPLETED

locus	alleles	phenoname	AC_Allele2	AF_Allele2	BETA	Pvalue	SE
locus<GRCh38>	array<str>	str	int32	float64	float64	float64	float64
chr1:13273	["G","C"]	"height"	468	4.68e-03	3.81e-02	1.23e-01	2.47e-02
chr1:51479	["T","A"]	"height"	2815	2.82e-02	-6.94e-03	5.79e-01	1.25e-02
chr1:54421	["A","G"]	"height"	2767	2.77e-02	1.70e-02	2.24e-01	1.40e-02
chr1:54625	["G","A"]	"height"	127	1.27e-03	-2.22e-02	7.28e-01	6.40e-02
chr1:54652	["A","G"]	"height"	114	1.14e-03	-6.83e-03	9.21e-01	6.89e-02
chr1:54815	["T","C"]	"height"	488	4.88e-03	-7.69e-02	1.94e-02	3.29e-02
chr1:54850	["G","C"]	"height"	124	1.24e-03	-5.99e-02	3.62e-01	6.57e-02
chr1:55298	["G","A"]	"height"	220	2.20e-03	-7.31e-02	1.03e-01	4.48e-02
chr1:55330	["G","A"]	"height"	1033	1.03e-02	-5.43e-03	7.97e-01	2.11e-02
chr1:55385	["A","G"]	"height"	343	3.43e-03	-3.75e-02	2.98e-01	3.60e-02
chr1:55416	["G","A"]	"height"	1651	1.65e-02	-7.48e-03	6.53e-01	1.67e-02
chr1:55427	["T","C"]	"height"	430	4.30e-03	-1.34e-02	6.74e-01	3.18e-02
chr1:55565	["G","A"]	"height"	342	3.42e-03	4.90e-02	1.81e-01	3.66e-02
chr1:56644	["A","C"]	"height"	411	4.11e-03	-1.12e-03	9.74e-01	3.37e-02
chr1:57095	["T","C"]	"height"	819	8.19e-03	1.42e-02	5.72e-01	2.51e-02
chr1:57292	["C","T"]	"height"	196	1.96e-03	3.61e-02	4.46e-01	4.74e-02
chr1:61397	["G","A"]	"height"	434	4.34e-03	6.91e-02	4.34e-02	3.42e-02
chr1:61442	["A","G"]	"height"	94778	9.48e-01	-1.24e-02	2.38e-01	1.05e-02
chr1:61898	["T","C"]	"height"	108	1.08e-03	-4.77e-02	4.96e-01	7.01e-02
chr1:61920	["G","A"]	"height"	188	1.88e-03	-2.54e-02	6.40e-01	5.43e-02

showing top 20 rows

Query per phenotype HTs (Gene)

```
In [43]: # gene HT: AMR height
         ht3 = hl.read_table(get_ht_path(ancestry = 'amr', pheno = 'height', test_type = 'gene'))
         ht3.describe()
```

```
----------------------------------------
Global fields:
    'n_cases': int32
    'n_controls': int32
    'heritability': float64
    'saige_version': str
    'inv_normalized': str
    'lambda_gc_maxmaf_0.01': struct {
        lambda_gc_Pvalue: float64,
        lambda_gc_Pvalue_Burden: float64,
        lambda_gc_Pvalue_SKAT: float64
    }
    'lambda_gc_maxmaf_0.001': struct {
        lambda_gc_Pvalue: float64,
        lambda_gc_Pvalue_Burden: float64,
        lambda_gc_Pvalue_SKAT: float64
    }
    'lambda_gc_maxmaf_0.0001': struct {
        lambda_gc_Pvalue: float64,
        lambda_gc_Pvalue_Burden: float64,
        lambda_gc_Pvalue_SKAT: float64
    }
    'lambda_gc_maxmaf_Cauchy': struct {
        lambda_gc_Pvalue: float64,
        lambda_gc_Pvalue_Burden: float64,
        lambda_gc_Pvalue_SKAT: float64
    }
----------------------------------------
Row fields:
    'max_MAF': float64
    'Pvalue': float64
    'Pvalue_Burden': float64
    'Pvalue_SKAT': float64
    'BETA_Burden': float64
    'SE_Burden': float64
    'MAC': int32
    'Number_rare': int32
    'Number_ultra_rare': int32
    'Pvalue_log10': float64
    'Pvalue_Burden_log10': float64
    'Pvalue_SKAT_log10': float64
    'gene_id': str
    'gene_symbol': str
    'annotation': str
    'phenoname': str
    'total_variants': int32
    'interval': interval<locus<GRCh38>>
    'CHR': str
    'POS': int32
    'Pvalue_expected': float64
    'Pvalue_expected_log10': float64
    'Pvalue_Burden_expected': float64
    'Pvalue_Burden_expected_log10': float64
    'Pvalue_SKAT_expected': float64
    'Pvalue_SKAT_expected_log10': float64
----------------------------------------
Key: ['gene_id', 'gene_symbol', 'annotation', 'max_MAF']
----------------------------------------
```

[0.01,
0.001,
0.0001]

3 x max_MAF cutoffs

[SKATO,
Burden,
SKAT]

3 x Tests

[‘pLoF’,
‘missenseLC’,
‘synonymous’,
‘pLoF;missenseLC’]

4 x annotations

```
In [51]: # Check number of records
         ht3.count()
```

Hail | 1 EXECUTORS | 4 CORES | Jobs: | 1 COMPLETED

```
[Stage 18:=============================================================> (38 + 1) / 39]
Out[51]: 239638
```

```
In [52]: # Count number of pLoF results at max_MAF == 0.001
         ht3.aggregate(hl.agg.count_where((ht3.annotation == 'pLoF') & (ht3.max_MAF == 0.001)))
```

Hail | 1 EXECUTORS | 4 CORES | Jobs: | 1 COMPLETED

```
[Stage 19:=============================================================> (38 + 1) / 39]
Out[52]: 16676
```

```
In [53]: # Count number of each annotations
         ht3.aggregate(hl.agg.counter(ht3.annotation))
```

Hail | 1 EXECUTORS | 4 CORES | Jobs: | 1 COMPLETED

```
[Stage 20:=======================================>                 (29 + 4) / 39]
Out[53]: {'Cauchy': 19010,
          'missenseLC': 56934,
          'pLoF': 49927,
          'pLoF;missenseLC': 56970,
          'synonymous': 56797}
```

```
In [54]: # Check the distribution of Pvalue
         ht3.Pvalue.summarize()
```

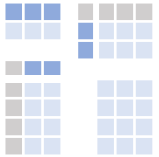
Hail | 1 EXECUTORS | 4 CORES | Jobs: | 1 COMPLETED

```
[Stage 21:=============================================>                 (30 + 4) / 39]
```

239638 records.
Pvalue (*float64*):

Non-missing	239638 (100.00%)
Missing	0
Minimum	0.00
Maximum	1.00
Mean	0.49
Std Dev	0.30

Query MT results



Note:

- for results on one or very few phenotype, it is always more efficient to look at the HTs instead of MTs
- MTs are mostly helpful when:
    1. Looking for association of one or a few gene/variant across MANY phenotypes
    2. Comparing associations of multiple phenotypes

Query MT results

In [58]:
```
# Exome MT: EUR
mt2 = hl.read_matrix_table(get_mt_path(ancestry = 'eur', test_type = 'exome'))
mt2.describe()
```

```
----------------------------------------
Global fields:
    None
----------------------------------------
Column fields:
    'phenoname': str
    'n_cases': int64
    'n_controls': int64
    'heritability': float64
    'saige_version': str
    'inv_normalized': str
    'pheno_sex': str
    'trait_type': str
    'category': str
    'pop': str
    'description': str
    'phecode_category': str
    'description_more': str
    'lambda_gc': float64
----------------------------------------
Row fields:
    'locus': locus<GRCh38>
    'alleles': array<str>
    'MarkerID': str
    'annotation': str
    'variant_id': str
    'gene_id': str
    'gene_symbol': str
    'transcript_id': str
    'revel': float64
    'splice_ai_acceptor_gain_score': float64
    'splice_ai_acceptor_gain_distance': int32
    'splice_ai_acceptor_loss_score': float64
    'splice_ai_acceptor_loss_distance': int32
    'splice_ai_donor_gain_score': float64
    'splice_ai_donor_gain_distance': int32
    'splice_ai_donor_loss_score': float64
    'splice_ai_donor_loss_distance': int32
    'hgvsp': str
    'splice_ai_ds': float64
    'AF_raw': array<float64>
    'AC_raw': array<int32>
    'AN_raw': int32
    'homozygote_count_raw': array<int32>
    'AF': array<float64>
    'AC': array<int32>
    'AN': int32
    'homozygote_count': array<int32>
    'quality_flags': struct {
        hq_exp_AC_variant: bool
    }
    'quality_flags_lambda': struct {
        hq_AF_variant: bool
    }
    'hq_variant': bool
    'hq_variant_lambda': bool
----------------------------------------
Entry fields:
    'AC_Allele2': int32
    'AF_Allele2': float64
    'MissingRate': float64
    'BETA': float64
    'SE': float64
    'var': float64
    'p.value.NA': float64
    'Is.SPA': bool
    'AF_case': float64
    'AF_ctrl': float64
    'Pvalue': float64
    'hq_exp_AC': bool
----------------------------------------
Column key: ['phenoname']
Row key: ['locus', 'alleles']
----------------------------------------
```

In [68]:
```
mt2.cols().show()
```

Hail | 1 EXECUTORS | 4 CORES | Jobs: 1 COMPLETED

phenoname	n_cases	n_controls	heritability	saige_version	inv_normalized	pheno_sex	trait_type	category	pop	description	phec
str	int64	int64	float64	str	str	str	str	str	str	str	str
"008"	923	95256	0.00e+00	"SAIGE_1.3.0"	"True"	"Both"	"binary"	"phecode"	"eur"	"Intestinal infection"	
"038"	1817	94461	2.06e-02	"SAIGE_1.3.0"	"True"	"Both"	"binary"	"phecode"	"eur"	"Septicemia"	
"041"	3506	89625	2.19e-02	"SAIGE_1.3.0"	"True"	"Both"	"binary"	"phecode"	"eur"	"Bacterial infection NOS"	
"041.1"	1095	96625	3.36e-02	"SAIGE_1.3.0"	"True"	"Both"	"binary"	"phecode"	"eur"	"Staphylococcus infections"	
"041.12"	642	97813	4.40e-02	"SAIGE_1.3.0"	"True"	"Both"	"binary"	"phecode"	"eur"	"Methicillin resistant Staphylococcus aureus"	
"041.2"	260	98019	0.00e+00	"SAIGE_1.3.0"	"True"	"Both"	"binary"	"phecode"	"eur"	"Streptococcus infection"	
"053"	1269	95069	1.92e-02	"SAIGE_1.3.0"	"True"	"Both"	"binary"	"phecode"	"eur"	"Herpes zoster"	
"054"	1739	94949	1.03e-02	"SAIGE_1.3.0"	"True"	"Both"	"binary"	"phecode"	"eur"	"Herpes simplex"	
"070"	1573	96629	2.52e-02	"SAIGE_1.3.0"	"True"	"Both"	"binary"	"phecode"	"eur"	"Viral hepatitis"	
"070.3"	1330	97289	3.87e-02	"SAIGE_1.3.0"	"True"	"Both"	"binary"	"phecode"	"eur"	"Viral hepatitis C"	

showing top 10 rows

In [70]:
```
mt2 = mt2.filter_rows((mt2.hq_variant_lambda) & (mt2.annotation != 'non-coding'))
mt2.rows().select('gene_id', 'gene_symbol', 'annotation', 'AF', 'AC', 'AN', 'hq_variant', 'hq_variant_lambda').show(
```

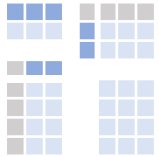
Hail | 1 EXECUTORS | 4 CORES | Jobs: 3 COMPLETED

[Stage 37:==========> (14 + 2) / 16]

locus	alleles	gene_id	gene_symbol	annotation	AF	AC	AN	hq_variant	hq_variant_lambda
locus<GRCh38>	array<str>	str	str	str	array<float64>	array<int32>	int32	bool	bool
chr1:924450	["G","A"]	"ENSG00000187634"	"SAMD11"	"missense"	[1.00e+00,4.77e-04]	[228281,109]	228390	True	True
chr1:924533	["A","G"]	"ENSG00000187634"	"SAMD11"	"synonymous"	[1.90e-02,9.81e-01]	[4334,224238]	228572	True	True
chr1:930248	["G","A"]	"ENSG00000187634"	"SAMD11"	"missense"	[9.94e-01,5.80e-03]	[227351,1327]	228678	True	True
chr1:930285	["G","A"]	"ENSG00000187634"	"SAMD11"	"missense"	[9.99e-01,5.60e-04]	[228544,128]	228672	True	True
chr1:930314	["C","T"]	"ENSG00000187634"	"SAMD11"	"missense"	[9.99e-01,7.35e-04]	[228508,168]	228676	True	True
chr1:935779	["G","A"]	"ENSG00000187634"	"SAMD11"	"missense"	[9.99e-01,1.02e-03]	[228448,234]	228682	True	True
chr1:935835	["C","G"]	"ENSG00000187634"	"SAMD11"	"synonymous"	[9.98e-01,1.53e-03]	[228327,351]	228678	True	True
chr1:939121	["C","T"]	"ENSG00000187634"	"SAMD11"	"missense"	[9.99e-01,5.77e-04]	[228546,132]	228678	True	True
chr1:939285	["G","A"]	"ENSG00000187634"	"SAMD11"	"synonymous"	[9.99e-01,6.08e-04]	[228533,139]	228672	True	True
chr1:939354	["C","T"]	"ENSG00000187634"	"SAMD11"	"synonymous"	[9.98e-01,1.60e-03]	[228302,366]	228668	True	True

showing top 10 rows

Query MT results



Compare several phenotypes

In [73]:
```
pheno_lst = ['height', 'weight', '008']
sub_mt = mt2.filter_cols(hl.literal(pheno_lst).contains(mt2.phenoname))
sub_ht = sub_mt.entries()
sub_ht = sub_ht.select('Pvalue', 'BETA') # select fields of interest
sub_ht.describe()
# sub_ht.export(YOUR_OUTPUT_NAME.txt.bgz)
```

```
----------------------------------------
Global fields:
    None
----------------------------------------
Row fields:
    'locus': locus<GRCh38>
    'alleles': array<str>
    'phenoname': str
    'Pvalue': float64
    'BETA': float64
----------------------------------------
Key: ['locus', 'alleles', 'phenoname']
----------------------------------------
```

Check results of one variant across all phenotypes

In [7]:
```
# Look up results for one variant chr1:930248:G:A and export to flat text file
sub_mt = mt2.filter_rows((mt2.locus == hl.locus('chr1', 930248, reference_genome ='GRCh38')) &
                         (mt2.alleles ==['G', 'A']))
sub_mt = sub_mt.select_rows('annotation', 'gene_id', 'gene_symbol', 'AF', 'AC', 'AN')
sub_mt = sub_mt.select_cols()
sub_mt = sub_mt.select_entries('Pvalue', 'BETA')
sub_ht = sub_mt.entries()
sub_ht.describe()
# sub_ht.export(YOUR_OUTPUT_NAME.txt.bgz)
```

```
----------------------------------------
Global fields:
    None
----------------------------------------
Row fields:
    'locus': locus<GRCh38>
    'alleles': array<str>
    'annotation': str
    'gene_id': str
    'gene_symbol': str
    'AF': array<float64>
    'AC': array<int32>
    'AN': int32
    'phenoname': str
    'Pvalue': float64
    'BETA': float64
----------------------------------------
Key: ['locus', 'alleles', 'phenoname']
----------------------------------------
```

Export your data to text file

HT

```
ht = ht.filter(…)
ht = ht.select(…)
ht.export(f'{MY_PATH}/{FILEMANE}.tsv')
```

MT

- Convert to HT first, e.g.

```
1)  ht = mt.cols()
2)  ht = mt.rows()
3)  mt = mt.filter_rows(…)
    mt = mt.filter_cols(…)
    ht = mt.entries(…)
    ht = ht.select(…)
```

- `ht.export(f'{MY_PATH}/{FILEMANE}.tsv')`

Thanks!

- Konrad Karczewski
- Matt Solomonson
- Riley Grant
- Robert Carroll
- Wei Zhou
- Alicia Martin
- Ying Wang
- Dan Roden
- Ben Neale

- *All of Us*
  - Anji Musick
  - Namrata Gupta
  - DRC
    - Lee Lichtenstein
    - Wail Baalawi
  - VUMC
    - Megan He
    - Michael Lyons
  - Project staff
  - Participants!

- hail team
  - Daniel King
  - Jackie Goldstein
  - Daniel Goldstein
  - Cotton Seed

http://pan.ukbb.broadinstitute.org
https://genebass.org

The new interface.

Hail Table

Hail Table

idx: int
0 1 2 3 4 5 6 7 8 9

A table with ten rows containing the integers from 0 to 9.

Slide credit to: Daniel Goldstein

Hail Table

idx: int
0
1
2
3
4
5
6
7
8
9

A table with ten rows containing the integers from 0 to 9.

```
import hail as hl
ht = hl.utils.range_table(10)
```

Slide credit to: Daniel Goldstein

Hail Table

idx: int	silly: str
0	""
1	"a"
2	"aa"
3	"aaa"
4	"aaaa"
5	"aaaaa"
6	"aaaaaa"
7	"aaaaaaa"
8	"aaaaaaaa"
9	"aaaaaaaaa"

Add a new column.

```
ht = hl.utils.range_table(10)
ht = ht.annotate(
    silly = hl.str("a") * ht.idx
)
```

Slide credit to: Daniel Goldstein

Hail Table

4.5

Aggregate a table to a single value.

This gives us a plain old number, not a table.

```
ht = hl.utils.range_table(10)
mean_idx = ht.aggregate(
    hl.agg.mean(ht.idx)
)
```

Slide credit to: Daniel Goldstein

Hail Table

ht1

idx: int
0
1
2
3
4
5
6
7
8
9

ht2

is_odd: bool	mean_idx: float
False	4.0
True	5.0

Aggregate within groups.

This gives us a new table with a row per group

```
ht1 = hl.utils.range_table(10)
ht2 = ht1.group_by(
    is_odd = ht1.idx % 2 == 1
).aggregate(
    mean_idx = hl.agg.mean(ht1.idx)
)
```

Slide credit to: Daniel Goldstein

Hail Table

ht1

idx: int	silly: str
0	""
1	"a"
2	"aa"
3	"aaa"
4	"aaaa"

ht2

key: int	animal: str
1	"dog"
3	"cat"
4	"fish"
6	"bird"
7	"squirrel"
10	"tiger"

ht3

idx: int	silly: str	animal: str
0	""	NA
1	"a"	"dog"
2	"aa"	NA
3	"aaa"	"cat"
4	"aaaa"	"fish"

We can annotate one table from another

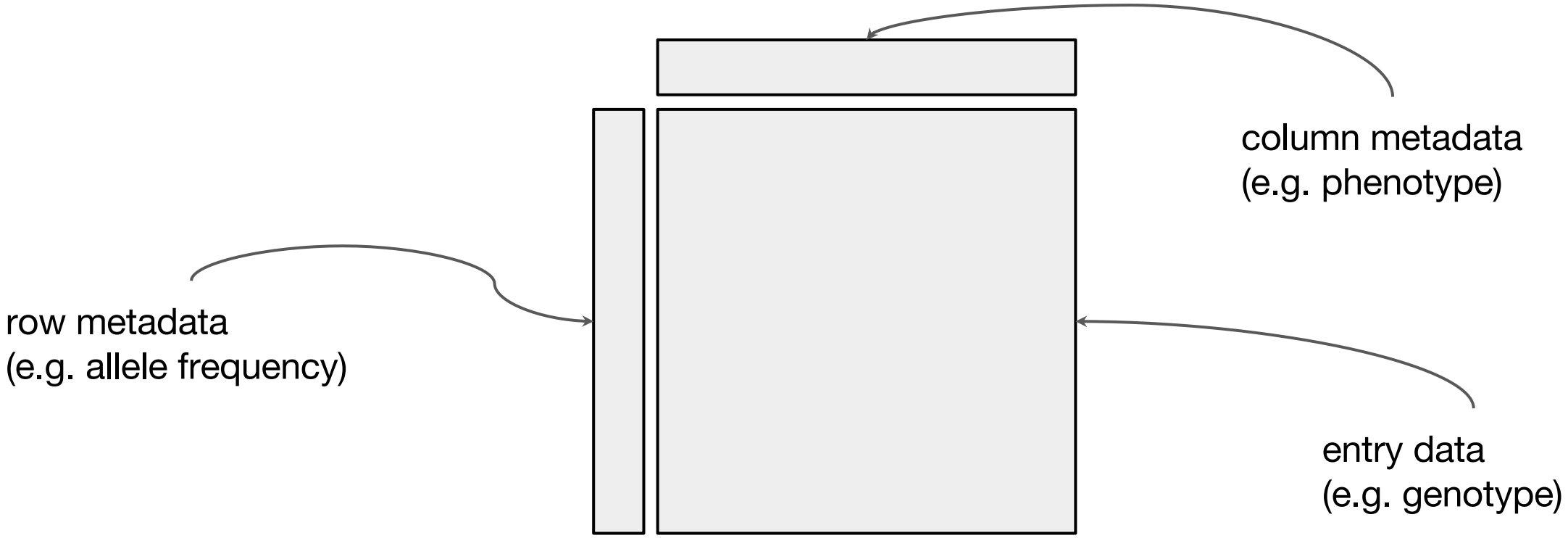
```
ht1 = hl.utils.range_table(5)
ht1 = ht1.annotate(
    silly = hl.str("a") * ht1.idx
)

ht2 = hl.import_table("animals.tsv")

ht3 = ht1.annotate(
    animal = ht2[ht1.idx].animal
)
```

Slide credit to: Daniel Goldstein

The new interface.

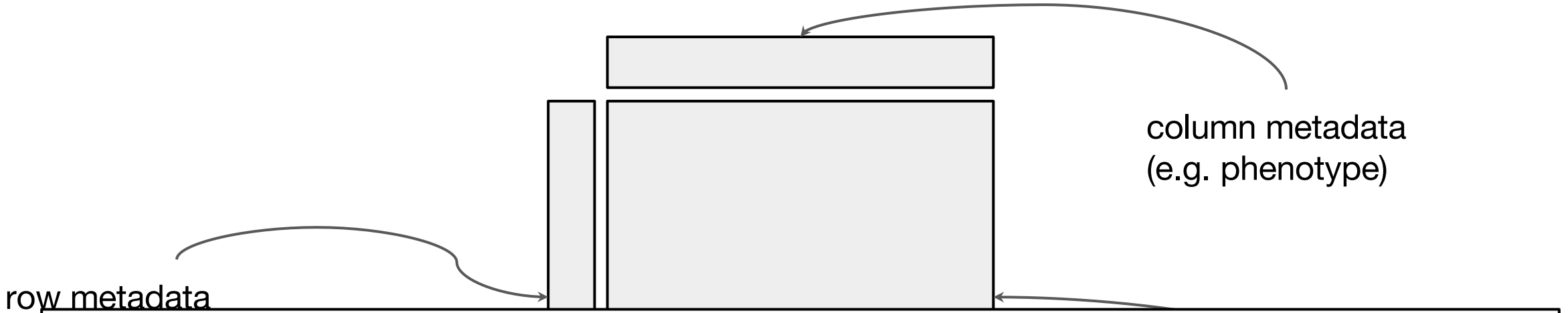
Hail Matrix Table

Hail Matrix Table



Slide credit to: Daniel Goldstein

Hail Matrix Table



In Hail, samples are columns!
This is transposed from traditional statistics.
This matches the orientation of VCF files.

Slide credit to: Daniel Goldstein

Hail Matrix Table

locus:locus<GRCh37>	alleles:array<str>	gene: str	…
chr1:949490	["G","A"]	"NOC2L"	…
chr1:950316	["G","A"]	"NOC2L"	…
chr1:950751	["C","T"]	"NOC2L"	…
chr1:951750	["G","A"]	"NOC2L"	…
chr1:952412	["C","T"]	"NOC2L"	…
chr1:953543	["G","A"]	"NOC2L"	…
chr1:953730	["G","A"]	"NOC2L"	…
chr1:954116	["C","G"]	"NOC2L"	…
chr1:954333	["C","A"]	"NOC2L"	…
chr1:954815	["C","T"]	"NOC2L"	…
⋮	⋮	⋮	⋱

Slide credit to: Daniel Goldstein

Hail Matrix Table

Sample_ID: str	sample1	sample2	…
height_in: float	64.3	69.1	…
pop: str	"AFR"	"AMR"	…
⋮	⋮	⋮	⋱

locus: locus<GRCh37>	alleles: array<str>	gene: str	…
chr1:949490	["G","A"]	"NOC2L"	…
chr1:950316	["G","A"]	"NOC2L"	…
chr1:950751	["C","T"]	"NOC2L"	…
chr1:951750	["G","A"]	"NOC2L"	…
chr1:952412	["C","T"]	"NOC2L"	…
chr1:953543	["G","A"]	"NOC2L"	…
chr1:953730	["G","A"]	"NOC2L"	…
chr1:954116	["C","G"]	"NOC2L"	…
chr1:954333	["C","A"]	"NOC2L"	…
chr1:954815	["C","T"]	"NOC2L"	…
⋮	⋮	⋮	⋱

Slide credit to: Daniel Goldstein

Hail Matrix Table

Sample_ID: str	sample1	sample2	…
height_in: float	64.3	69.1	…
pop: str	"AFR"	"AMR"	…
⋮	⋮	⋮	⋱

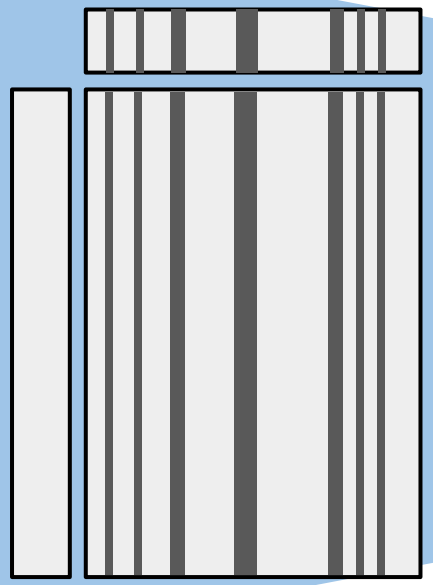
locus: locus<GRCh37>	alleles: array<str>	gene: str	…
chr1:949490	["G","A"]	"NOC2L"	…
chr1:950316	["G","A"]	"NOC2L"	…
chr1:950751	["C","T"]	"NOC2L"	…
chr1:951750	["G","A"]	"NOC2L"	…
chr1:952412	["C","T"]	"NOC2L"	…
chr1:953543	["G","A"]	"NOC2L"	…
chr1:953730	["G","A"]	"NOC2L"	…
chr1:954116	["C","G"]	"NOC2L"	…
chr1:954333	["C","A"]	"NOC2L"	…
chr1:954815	["C","T"]	"NOC2L"	…
⋮	⋮	⋮	⋱

GT	DP	PL	GT	DP	PL	
1/1	51	[1597,142,0	0/0	36	[0,88,1478]	⋱
1/1	33	]	0/1	42	[0,1453,120	
0/0	64	[1136,99,0]	0/0	64	]	
⋮	⋮	[0, 99,	⋮	⋮	[0,99,1336]	
		1336]			⋮	
		⋮				

Slide credit to: Daniel Goldstein

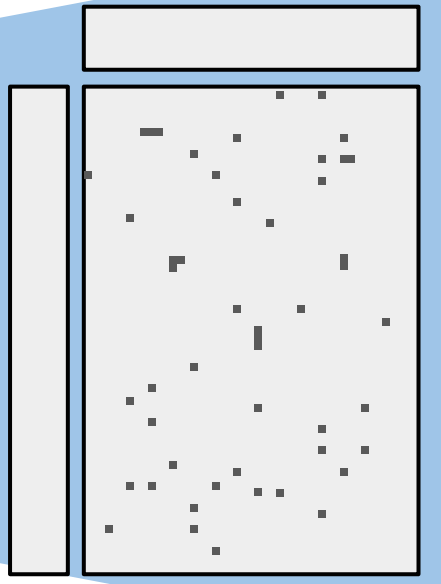
Hail Matrix Table: Filtration

```
mt = mt.filter_rows(mt.af < 0.03)
```



```
mt = mt.filter_cols(mt.age > 45)
```

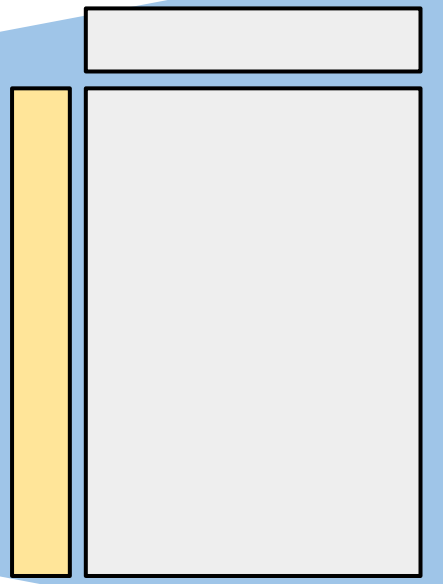
```
mt = mt.filter_entries(mt.GQ > 10)
```

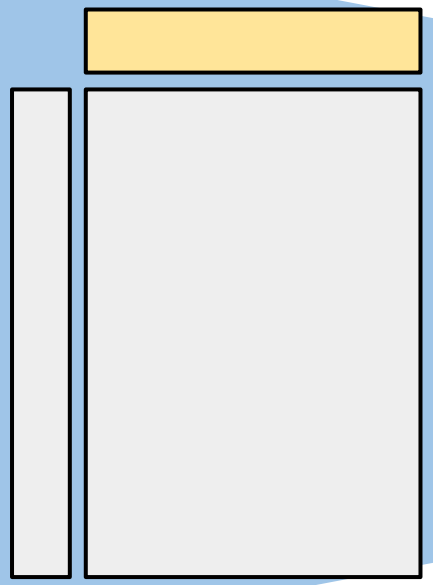


Slide credit to: Daniel Goldstein

Hail Matrix Table: Annotation

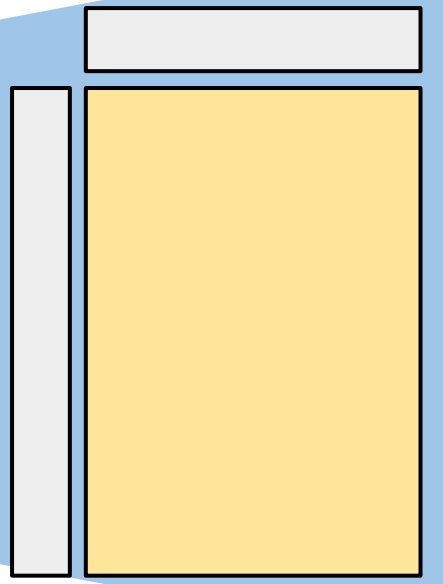
```
mt = mt.annotate_rows(
  gene = genes[mt.locus]
)
```





```
mt = mt.annotate_cols(
  has_disease = mt.value > 10
)
```

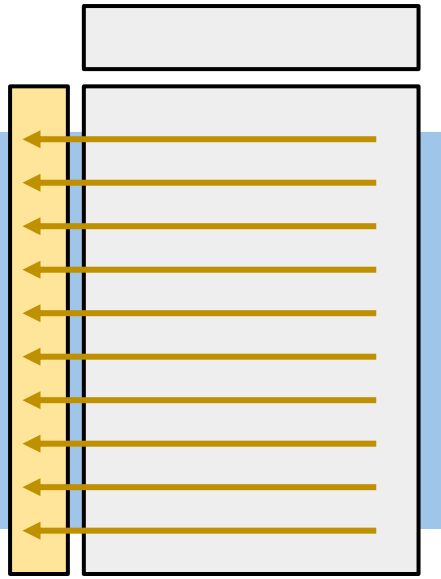
```
mt = mt.annotate_entries(
  x = mt.GT.n_alt_alleles()
)
```

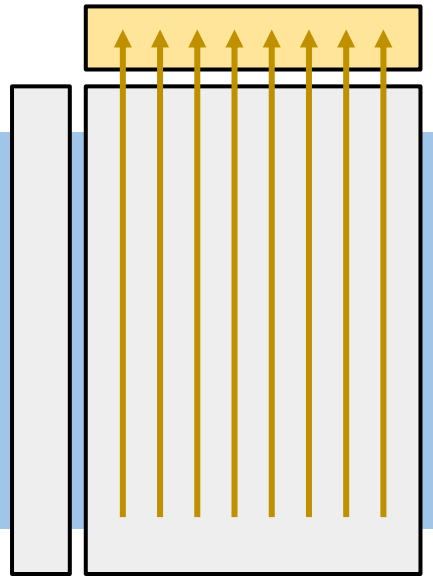


Slide credit to: Daniel Goldstein

Hail Matrix Table: Annotation by Aggregation

```
mt = mt.annotate_rows(
  af = hl.agg.mean(
    mt.GT.n_alt_alleles()
  )
)
```



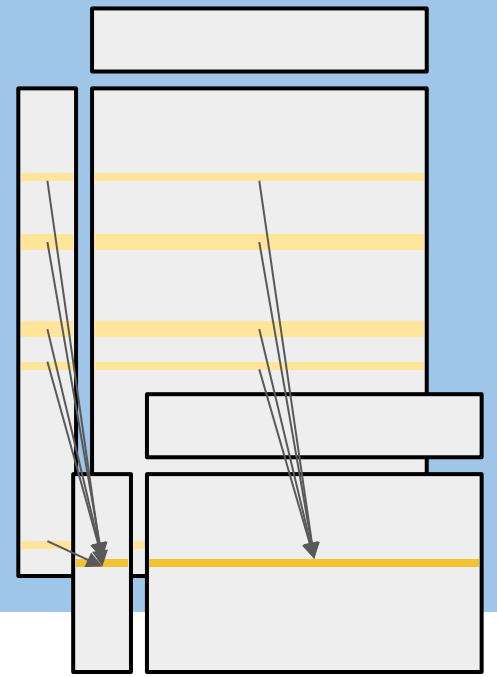


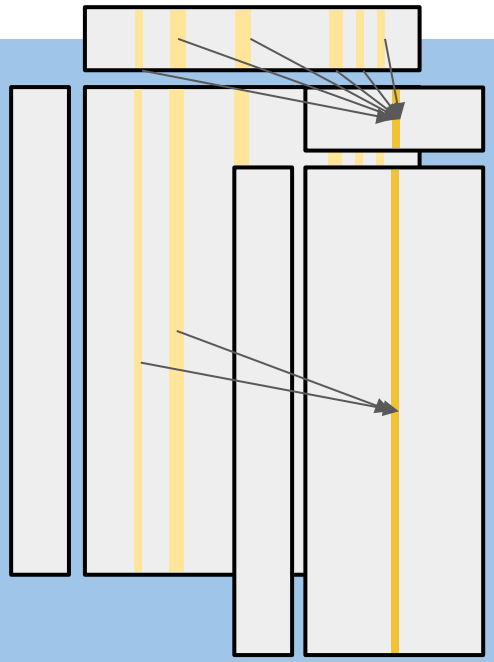
```
mt = mt.annotate_cols(
  heteroyzgosity = hl.agg.fraction(
    mt.GT.is_het()
  )
)
```

Slide credit to: Daniel Goldstein

Hail Matrix Table: Grouped Aggregation

```
mt = mt.group_rows_by(
    mt.gene
).aggregate(
    n_muts = hl.agg.sum(
        mt.GT.n_alt_alleles()
    )
)
```



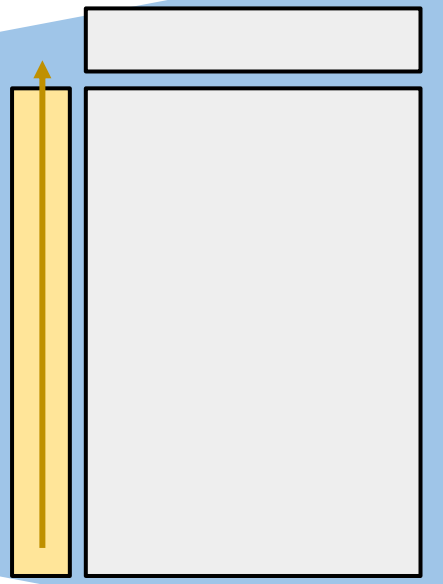


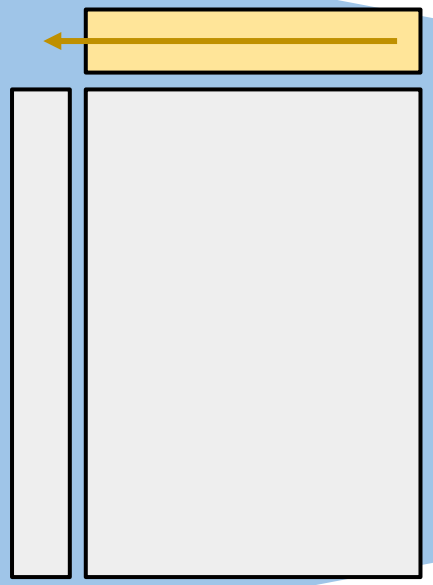
```
mt = mt.group_cols_by(
    mt.ancestral_pop
).aggregate(
    n_hets = hl.agg.count(mt.GT.is_het())
)
```

Slide credit to: Daniel Goldstein

Hail Matrix Table: Total Aggregation

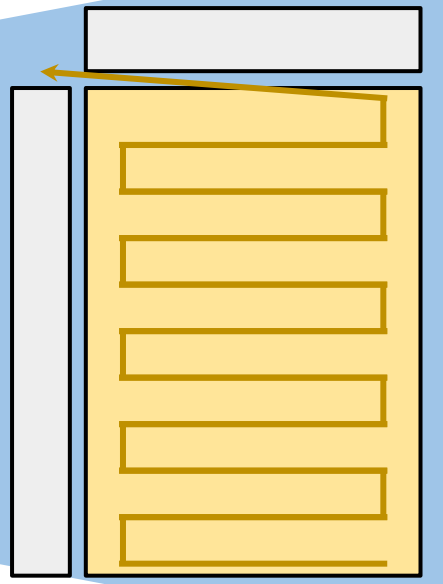
```
af_hist = mt.aggregate_rows(
  hl.agg.approx_cdf(mt.af)
)
```





```
fraction_is_case = mt.aggregate_cols(
  hl.agg.fraction(mt.is_case)
)
```

```
fraction_het = mt.aggregate_entries(
  hl.agg.fraction(mt.is_het())
)
```



Slide credit to: Daniel Goldstein