

UKB Exomes Supplement

Data processing and joint calling	4
Data processing	4
Joint calling	4
Representation	5
Fig. S1 SparseMT created from two gVCF inputs.	6
Merge algorithm	7
Phenotype data processing	8
Fig. S2 Phenotype curation pipeline.	9
Genotype, Sample, and Variant Quality Control	11
Concordance between arrays and exomes	11
Interval QC	11
Fig. S3 Histogram showing the percentage of samples meeting 20X mean coverage for each exome capture interval.	12
Sample QC	12
Sex imputation	12
Fig. S4 Normalized chromosome X ploidy plotted against normalized chromosome Y ploidy and colored by sex karyotype.	13
Hard filters	13
Platform inference	14
Fig. S5 Platform inference using missingness PCA.	14
Relatedness inference	15
Ancestry imputation	15
Outlier detection	16
Table S1 Final sample counts passing QC.	16
Variant QC	17
Fig. S6 Variant QC	19
Genotype QC	19
Annotations	19
Scaling association testing using Hail Batch	21
Motivation	21
User interface	22
Fig. S7 Hail Batch schematic for SAIGE association analysis.	23
Implementation	24
Association testing and quality control	27
Association testing framework	27
Random phenotype analysis	28

Fig. S8 QQ-plots of randomly generated heritable (heritability = 100%) phenotypes	29
Fig. S9 Lambda GC by cumulative allele frequency (CAF) by heritability.	30
Fig. S10 Power for rare variant associations.	31
Allele frequency filters	31
Fig. S11 Lambda GC for each phenotype vs case count, split by CAF interval for SKAT-O, SKAT and burden tests.	32
Test calibration filtering	32
Fig. S12 Lambda GC for each phenotype vs case count for SKAT-O, SKAT and burden tests, before and after filtering out genes with CAF < 0.0001 and coverage < 20.	33
Fig. S13 Coverage vs gene-based lambda GC, for SKAT-O (A) and burden tests (B).	34
Fig. S14 Lambda GC for each phenotype.	35
Fig. S15 Lambda GC for each gene.	36
Independent phenotypes	36
Fig. S16 Independence of phenotypes.	37
Table S2 QC of summary statistics.	38
Comparison to known hits	38
Table S3 Comparison to 32 rare (MAF < 1%) variants associated with adult height in GIANT	39
Table S4 Comparison to 59 low-frequency (MAF between 1% and 5%) variants associated with adult height in GIANT	40
Table S5 Comparison to 10 genes associated with adult height in GIANT	44
Fig. S17 Comparison of effect sizes between UK Biobank and GIANT for height.	46
Table S6 Comparison of 20 associations between missense variants and 7 major red blood cell phenotypes discovered at the genome-wide significant loci of the marginal tests in TOPMed	46
Analysis of summary statistics	49
Gene set analyses	49
Developmental delay genes	49
Constrained genes	49
MAF/CAF Matching	50
PolyPhen2 predicted variants	50
Fig. S18 The proportion of variants with at least one association is shown broken down by PolyPhen2 annotation group and allele frequency category.	51
ClinVar variants	52
Data availability and release	53
Data availability	53
Code availability	53
Data browser	53
Navigation and workflow	54

Fig. S19 Overview of the UKBB exome gene browser interface.	55
Exploring associations by gene	55
Exploring associations by phenotype	57
Fig. S20 Results by phenotype.	57
Comparing single variant associations across phenotypes	58
Fig. S21 Multi-phenotype plotting.	58
Fig. S22 Using hover interactions with the multi-phenotype pivot table.	60
Fig. S23 Viewing case-control counts and allele frequencies for pLoF variants across traits in a gene.	61
Fig. S24 Color variants by attribute to uncover patterns in A) consequence, B) p-value, C) beta, D) trait, or E) zygosity.	62
Single variant results	62
Fig. S25 Single variant page.	63
References	64

Data processing and joint calling

Timothy Poterba, Chris Vittal, Duncan Palmer, Cara Mason, Konrad J. Karczewski, Namrata Gupta, Claire Churchhouse, Daniel G. MacArthur, Anthony Philippakis, Cotton Seed, Melissa R. Miller, Benjamin M. Neale

Data processing

All CRAM files were re-processed according to the GATK Best Practices. Briefly, reads were aligned using BWA-MEM 0.7.15.r1140 and processed using Picard and GATK. For each sample, variants were called using the Genome Analysis Toolkit (GATK) 4.0.10.1. Samples were called individually using local realignment by HaplotypeCaller in gVCF mode, such that every position in the genome is assigned likelihoods for discovered variants or for the reference. gVCF outputs were then post-processed, further compressing "blocks" of homozygous reference calls to seven GQ bins: 0, 10, 20, 30, 40, 50 and 60. All analyses were performed on Google Cloud Platform.

Joint calling

Many technologies used to represent, store, and compute on genomic data do not scale efficiently to datasets of hundreds of thousands of samples with whole exome or genome sequence data. The Hail project is invested in addressing these scaling limitations by developing new technology to enable these large callsets. One key innovation that allowed efficient analysis of such a large callset is the Hail SparseMT representation, a storage and analysis target that can serve as an alternative to the project VCF (pVCF) and exhibits strictly linear asymptotic scaling with the number of stored samples. The SparseMT is a conceptual representation for the variant-level data on a sequenced cohort. In the Hail library, we have developed an implementation of an efficient hierarchical merge algorithm to create a Hail SparseMT from single-sample gVCFs, and a rudimentary set of functionality for working with this object as an analysis target.

Representation

We will briefly describe the representation of the SparseMT and where it differs from a pVCF. First, the SparseMT contains a row for each locus defined in any input gVCF. While pVCF only includes a row for each variant site, the SparseMT contains rows for monomorphic sites where reference blocks begin. This leads to significantly more rows in the SparseMT compared to pVCF, though this difference shrinks with increasing sample size when rare variants are discovered at most positions in the genome. The alternate alleles field of the SparseMT at a given locus contains the set of all unique alternate alleles observed in any input gVCF for that locus (Fig. S1).

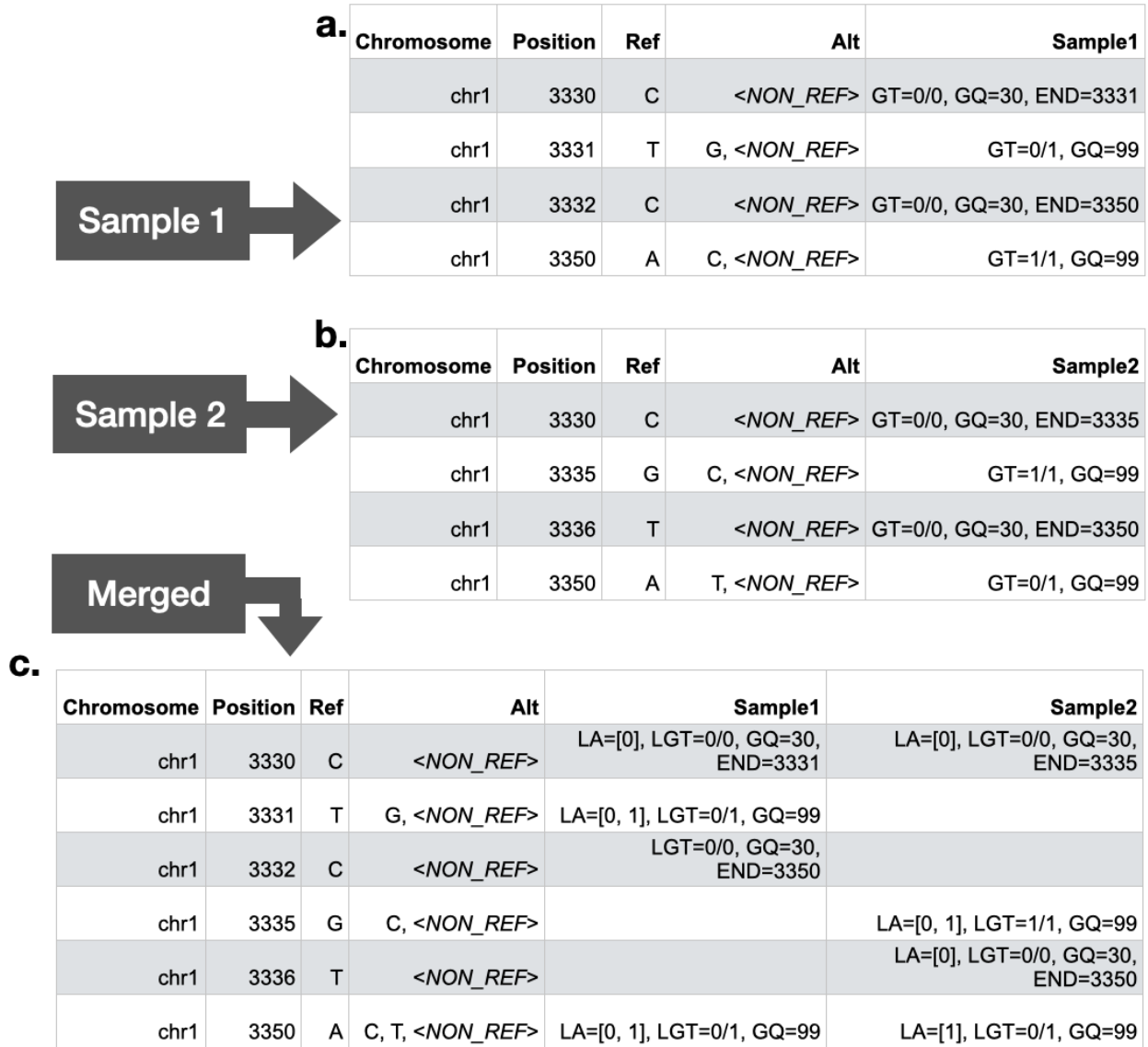


Fig. S1 | SparseMT created from two gVCF inputs. Panels a and b display information contained in gVCFs for two distinct samples in a small genomic window. Panel c represents the merged SparseMT, which contains all loci present in either a or b. There is no entry for Sample2 at chr1:3331 because Sample2's gVCF does not contain the locus chr1:3331. The GT field has been renamed to LGT (local GT), and the LA (local alleles) field has been added to record the original alleles in each gVCF, which is important at chr1:3350, a locus where both input samples have a variant call.

An entry of the SparseMT matrix corresponding to locus L and sample S is dense (defined) if the gVCF for sample S has a line for locus L, and sparse (missing) otherwise. If defined, the entry includes a light transformation of all information from the gVCF for S at locus L, with the two exceptions of the locus information (chromosome and position) and the alleles

(reference and alternates). The chromosome and position are necessarily the same as those constituting L, and instead of including alleles as strings, the SparseMT has an additional field, LA, or “local alleles”, which is an array of integers. The value at each index i of this list is the index in the full list of alternate alleles of the SparseMT row at which the i th alternate allele from sample S’s gVCF can be found. The final transformation is a rename of GT and all fields that are “R-” or “A-” numbered, such as AD and PL, to LGT, LAD, LPL, etc. This rename reflects the change in meaning -- the alleles referred to by these fields are the ones defined in LA, rather than the full list of discovered alleles in the SparseMT row.

It is important to note that the SparseMT representation is a lossless transformation of gVCFs -- the original gVCFs can be recreated by reversing the transformations defined above.

Merge algorithm

We have proposed the SparseMT as an efficient conceptual representation for efficiently storing cohort-level data sequencing data. We now describe a hierarchical merge algorithm that lends itself to scalable construction of a SparseMT.

First, each input gVCF is converted to a single-sample SparseMT. This conversion involves trivial reorganizations and renames of fields (GT to LGT, as described above). The result is a SparseMT with the same number of rows as the gVCF and one column, where every entry is defined.

Second, N SparseMTs are merged together into a new SparseMT. First, an outer join on loci is executed for the N SparseMTs, producing some number of intermediate rows with a locus (the join key), the alleles for each input SparseMT (which may be missing if a SparseMT did not contain that locus), and the entries associated with that locus for each SparseMT. The merged alleles are computed by taking the set of all unique alleles from all input SparseMT matrices, and sorting according to a deterministic string ordering function. The merged entries are

computed by taking the full set of entries across join inputs, and updating one field: the values of each LA (local alleles) field must be recomputed to refer to the new, merged alternate alleles.

The Hail framework features a concrete implementation of this hierarchical merge algorithm. In practice, this algorithm defaults to a value of 100 for the branch factor parameter, N , which performs well empirically and allows for the creation of a million-sample dataset in three rounds of merging.

Dense matrix analysis

The significant scaling advantages of the SparseMT representation are not without cost. The sparsity prevents random access to reference block information at any specific variant, because most of the reference blocks spanning that variant will have been defined at earlier loci. Analyses that require access to reference block metadata (such as GQ or DP) can be implemented following a "densification" pass that carries forward an array of genotype values, one for each sample, corresponding to the most recent reference block entry on the same chromosome. The reference blocks in this array can be used to fill sparse entries for each locus in the SparseMT (as long as a candidate reference block spans the locus of interest). The necessary dense analysis target can thus be realized at low cost on the fly during analysis pipelines, then discarded. It is not necessary to save the dense matrix to durable storage, which would incur prohibitive cost.

Phenotype data processing

In order to automate the curation and harmonisation of the large collection of variable scalings, encodings, and follow-up responses in a coherent manner, we created a modified version of the PHEnome Scan ANalysis Tool (PHESANT), available at <https://github.com/astheeggeeggs/PHESANT>. Unlike the original implementation (1), our version does not perform association analyses, but simply generates a collection of re-coded phenotypes. An outline of the PHESANT pipeline and our chosen filters are displayed in Fig. S2.

We manually curated the collection of phenotypes for study, which we filter to as part of the pipeline. Re-codings of variables, and inherent orderings of ordinal categorical variables, are defined in the data-coding file, which is available in our GitHub repository.

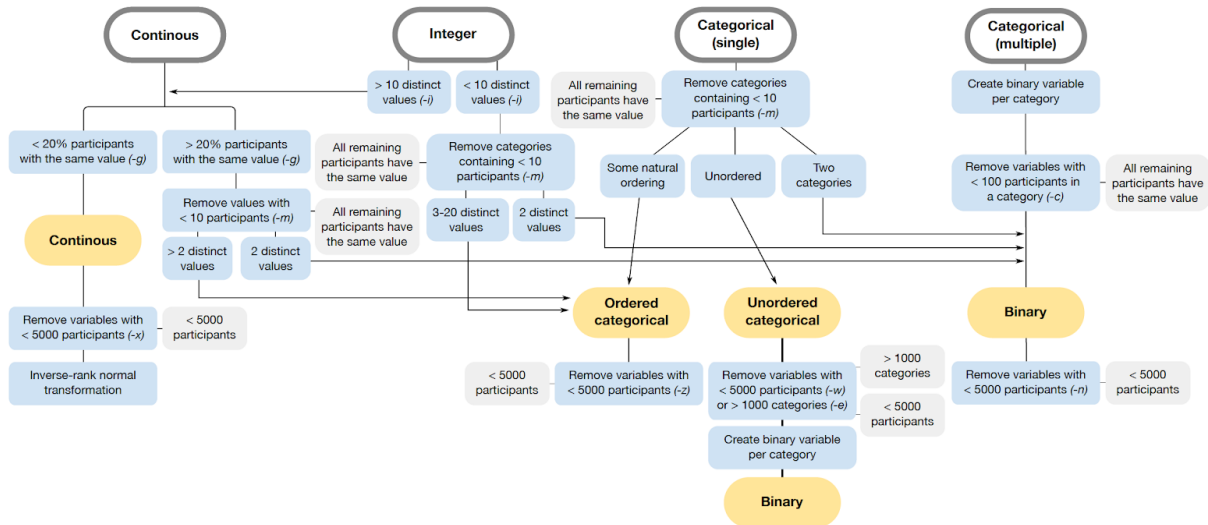


Fig. S2 | Phenotype curation pipeline. Raw phenotype data (gray outlined boxes) are passed to PHESANT, and a collection of filters (blue boxes) are applied. The thresholds shown here are the defaults in our modified version of PHESANT that can be altered in our code as desired using the flags displayed in parentheses. Grey filled boxes display the criteria for removal, and yellow filled boxes show the category of the variable after the rules in the blue boxes have been enforced.

In addition to the inverse-rank normalization applied to the collection of continuous phenotypes, we also retain the raw version of the continuous phenotype, with no transformation applied to the data (though re-codings of the data to guard against spurious results are retained). Following all of these alterations and additions, we run this modified version of PHESANT on the phenotypes in our UK Biobank application using a 200Gb RAM virtual machine on the Google Cloud Platform.

Upon applying the PHESANT pipeline to our selected collection of phenotypes, a small subset of categorical variables remain that should be sex-specific but are not excluded from the both-sexes collection of phenotypes. We manually identified this collection of sex-specific phenotypes and removed them from the both-sex phenotypes before subsequent analysis.

All data is loaded into a Hail MatrixTable using a custom processing script (https://github.com/Nealelab/ukb_exomes/blob/master/hail/load_phenotype_data.py). Briefly, we parsed the PHESANT output, extracted ICD codes from the “first occurrence” data (which were run as binarized outcomes), parsed some custom phenotypes and covariates, and combined these into a phenotype MatrixTable. The MatrixTable is keyed by trait type (continuous, categorical, and ICD code), phenocode, sex, coding (for categorical traits), and modifier (raw vs inverse-rank normalized for continuous traits).

Genotype, Sample, and Variant Quality Control

Julia K. Goodrich, Katherine R. Chao, Laura Gauthier, Konrad J. Karczewski, Benjamin M. Neale, Daniel G. MacArthur, Grace Tiao

We performed quality control in a similar fashion to the approach used for the Genome Aggregation Database (gnomAD) (2). Notably, however, we included a number of additional metrics, including concordance between arrays and exomes and interval QC.

Concordance between arrays and exomes

We confirmed that all 301,495 samples with available array data had a high proportion of variant concordance between their exome and array data. We filtered the UK Biobank array data to autosomal regions and lifted the data to genome build GRCh38 prior to examining concordance. The minimum proportion concordance of non-reference sample genotypes was 0.97 (mean 0.995).

Interval QC

We performed quality controls on intervals targeted by the exome capture before applying sample hard filters. In this interval QC, we investigated how much padding to add around the UK Biobank capture intervals and whether to check coverage across standard Broad exome calling intervals (available on Google Cloud Platform at gs://gcp-public-data--broad-references/hg38/v0/exome_calling_regions.v1.interval_list) as well. We determined that padding more than 50 base pairs into the intron added too much noise and reduced sample call rate and coverage significantly. We also discovered that adding calling intervals unique to the Broad's set of exome targets also reduced call rate and coverage. Therefore, we decided to keep variants only within the 50 base pair padded UK Biobank intervals. For sample QC, we also decided to filter to intervals where 85% of samples had a mean coverage of at least 20X (Fig. S3).

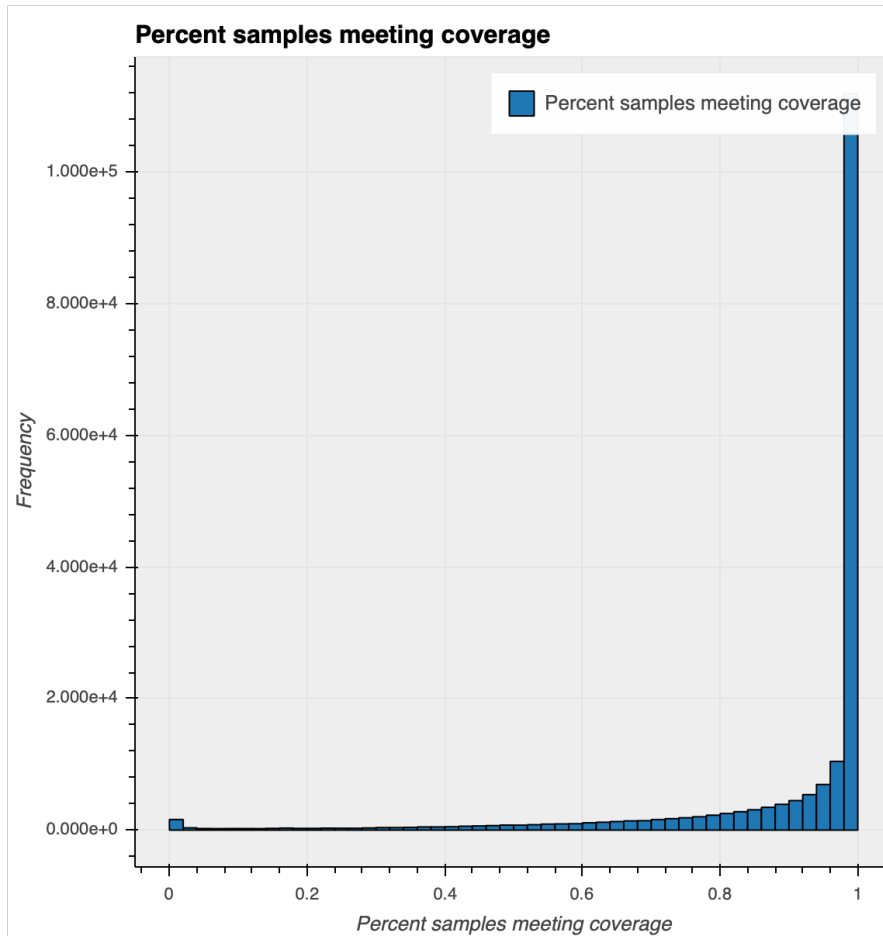


Fig. S3 | Histogram showing the percentage of samples meeting 20X mean coverage for each exome capture interval.

Sample QC

Sex imputation

We used Hail's *impute_sex* method to infer sex using common (allele frequency > 0.1%), non-pseudoautosomal, bi-allelic single nucleotide variants (SNVs) on chromosome X. We then computed non-pseudoautosomal chromosome X and chromosome Y coverage for each sample and normalized these values using sample-specific coverage across chromosome 20. We then checked the distribution of chromosome X and Y ploidies for XX and XY karyotypes to determine each karyotype's ploidy cutoffs (upper cutoff for single X: 0.994, lower cutoff for double X: 1.011, upper cutoff for double X: 1.295, lower cutoff for triple X: 1.321, lower cutoff for

single Y: 0.015, upper cutoff for single Y: 0.290, lower cutoff for double Y: 0.320). The adjusted ploidy cutoffs helped us add additional granularity to the inferred sex, differentiating between X0, XX, XXX, XY, XXY, XYY, and XXYY karyotypes (Fig. S4).

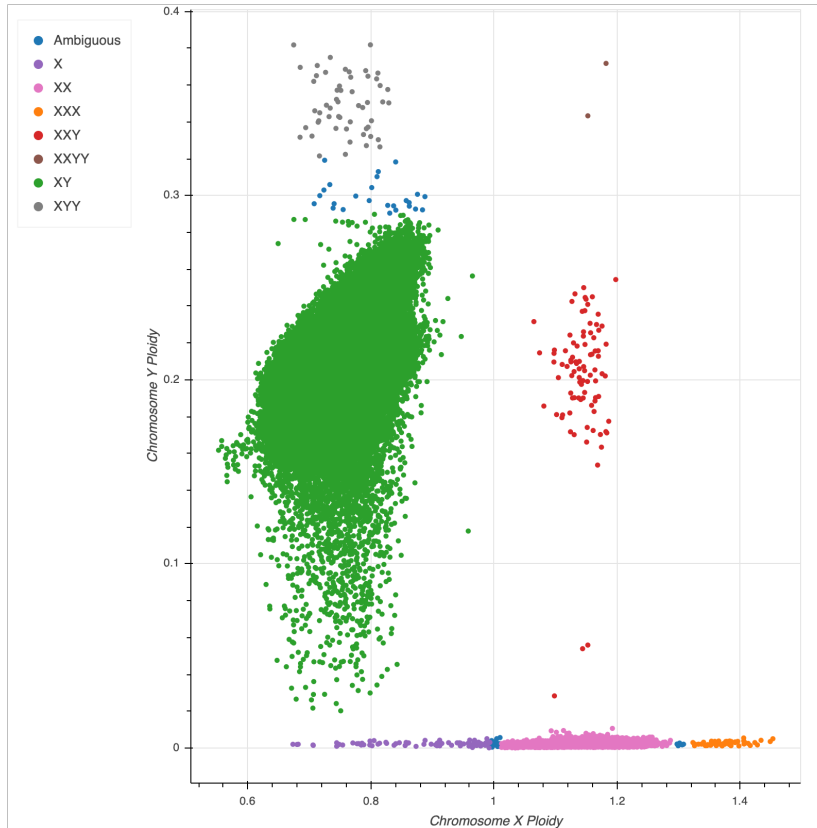


Fig. S4 | Normalized chromosome X ploidy plotted against normalized chromosome Y ploidy and colored by sex karyotype. XY samples are spread out in terms of their normalized chromosome Y coverage. This long tail of samples is likely due to mosaic loss of chromosome Y.

Hard filters

We applied three hard filters to the samples: sex imputation filters (removing any samples that were not inferred as XX or XY), a call rate filter (cutoff: 0.99), and a coverage filter (mean coverage cutoff 20X). We excluded hard-filtered samples from platform PCA, relatedness inference, ancestry imputation, and outlier detection so that these low quality samples would not influence our downstream results.

Platform inference

Although we do not expect there to be noticeable technical artifacts given that the samples were run on the same platform, we ran a principal component analysis (PCA) on the per-individual per-interval call rate matrix, as previously described (2), to make sure there were no significant clusters. Our platform PCA picked up a few clusters and showed some differences between the different sample batches on PCs 3 and 4 (Fig. S5). After some investigation, we discovered that the PC4 separated samples due to a common CNV (chr12:9531541-9531809). As PC3 showed some separation by batch, we decided to use batch status as a proxy for platform in further sample QC and a covariate for association analysis.

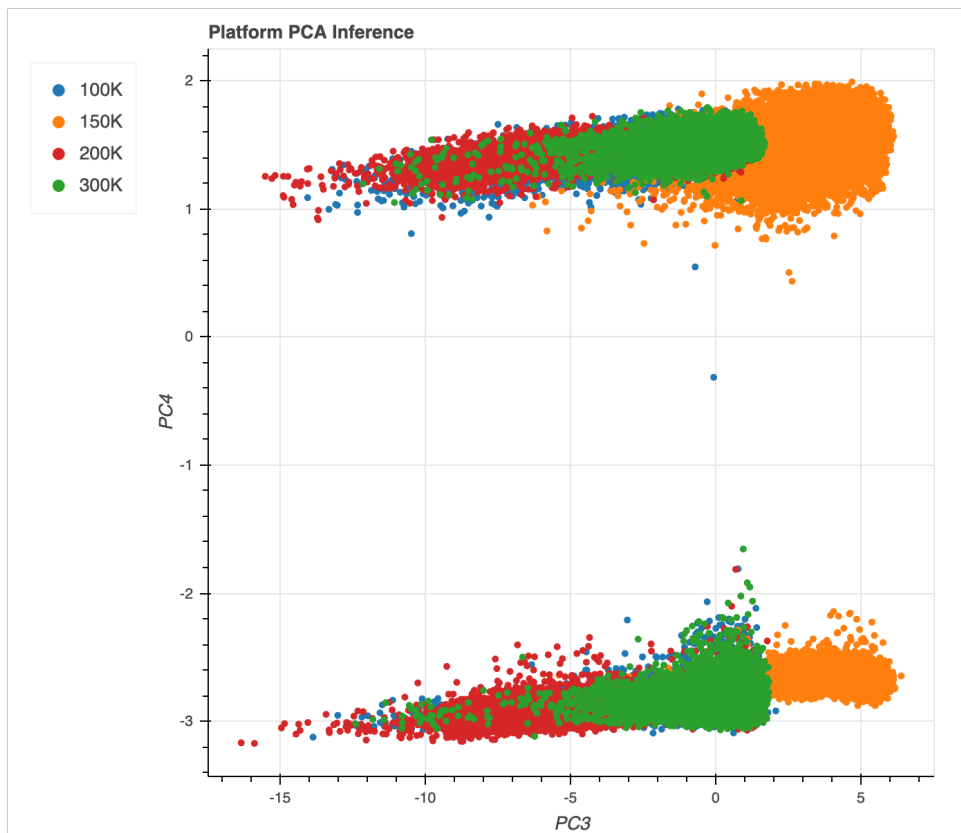


Fig. S5 | Platform inference using missingness PCA. PC3 vs PC4 colored by batch. Note that the batch names indicate the additional samples added from that batch. Thus, '100K' refers to data tranche 1, '150K' refers to samples added in tranche 1.5 (the first 50K samples released to the public), '200K' refers to samples added in tranche 2, and '300K' refers to samples added in tranche 3. The separation in PC4 is driven by a common copy number variant.

Relatedness inference

We used Hail's *pc_relate* method to infer relatedness on SNVs that are autosomal, bi-allelic, common (allele frequency > 0.1%), high call rate (> 99%), and LD-pruned with a cutoff of $r^2 = 0.1$. We then used the *maximal_independent_set* method in Hail to keep the largest set of unrelated samples (samples without second degree or closer relationships; kinship > 0.1), prioritizing samples with greater mean depth. The related samples that were not in the maximal independent set were flagged.

The majority of the samples removed during sample QC were samples inferred to have a second-degree or greater relationship with other samples in the dataset. This finding is not unexpected, as previous studies have shown that the UK Biobank data contains a large number (approximately 30% at third-degree or greater) of related samples (3).

Ancestry imputation

We used a hybrid method to infer ancestry. We projected the UK Biobank data onto the gnomAD population principal components (PCs) and then used a random forest classifier trained on gnomAD ancestry labels to assign ancestry to the UK Biobank samples. We observed that many samples labeled as African using this method were flagged as outliers by our downstream population-stratified outlier detection method. This seemed to be due to the fact that one cluster of samples labeled as African appeared highly admixed. To account for this, we ran a PCA on the UK Biobank samples and applied a clustering method (HDBSCAN). We found that this clustering method split the African labeled samples into additional clusters and reduced the number of samples flagged as outliers while also recapturing most of the same global population clusters observed in gnomAD.

As a result, we chose to assign ancestry using a hybrid of the projection onto gnomAD PCs and the UK Biobank specific PCA clustering: for any sample that was assigned to a cluster using the UK Biobank PCA, the sample was given that cluster as their ancestry assignment in

order to preserve the sub-structure observed using clustering. Any sample that was not assigned to a cluster was given the label from the initial (gnomAD) PCA projection and random forest classification.

Outlier detection

We flagged any sample falling outside 4 median absolute deviations (MADs) from the median of any of the following metrics (stratified by population and tranche as a proxy for platform), which were calculated using hail's *sample_qc* method:

- Number of deletions
- Number of insertions
- Number of SNVs
- Insertion : deletion ratio
- Transition : transversion (TiTv) ratio
- Heterozygous : homozygous ratio

The final counts of samples is shown in Table S1.

Table S1 | Final sample counts passing QC. "nfe" refers to samples inferred as having non-Finnish European ancestry. Note that relatedness was run after hard filtering, so the total number of related and unrelated individuals is equal to the total number of samples less 486.

Category	Related	Unrelated	All
Total	13757	288003	302246
Hard filtered			486
Outlier filtered	63	1693	1756
High quality	13694	286310	300004
High quality (nfe)	13101	268749	281850

Variant QC

Variant filtering consisted of a combination of a random forest (RF) classifier and hard filters. We used the following training sets as true positives for training the random forest model:

- Omni - SNVs present on the Omni 2.5 genotyping array and found in 1000 Genomes data
- Mills - Indels present in the Mills and Devine data (4)
- Transmitted singletons - Variants found in only two individuals, which were a parent-offspring pair
- Sibling singletons - Variants found in only two individuals, which were a sibling pair
- Common (AF > 0.1%) and concordant (>90% non-reference concordance) array variants

For the false positive training set in the random forest model, we used variants that fail traditional GATK hard filters: QUAL by depth (QD) < 2, strand bias estimated using Fisher's exact test (FS) > 60, or root mean square mapping quality (MQ) < 30. We balanced the number of variants in the true positive and false positive training sets by randomly downsampling the false positive training set to the same number of variants found in the true positive training set. RF training was performed on only variants that fall within intervals that pass interval QC (described above; intervals where >85% of samples have a mean coverage >20X).

We used the following allele and site annotations as features in the random forest model (RF feature importance shown in Fig. S6):

- Allele type - SNV, indel
- Number of alleles - Total number of alleles present at the site
- Variant type - SNV, indel, multi-SNV, multi-indel, mixed
- Mixed site - Whether more than one allele type is present at the site
- Spanning deletion - Whether there is a spanning deletion (STAR_AC > 0) that overlaps the site

- Quality by depth - Sum of the non-reference genotype likelihoods divided by the sum of the depth in all carriers of that allele
- Read position RankSum - Rank Sum Test for relative positioning of reference versus alternate alleles within reads
- Mapping quality RankSum - Rank Sum Test for mapping qualities of reference versus alternate reads
- Strand bias odds ratio - Symmetric Odds Ratio test of 2x2 contingency table to detect strand bias
- Max probability of allele balance - Highest p-value for sampling the observed allele balance under a binomial model with $p=0.5$ (maximum across heterozygotes)

RF probability cutoffs for calling a variant PASS were chosen to maximize sensitivity and specificity based on criteria such as the number of de novo mutations found in the 224 trios in the dataset and precision-recall (PR) curves (Fig. S6B) in two truth samples present in our data (NA12878 and a pseudo-diploid sample (syndip); syndip was sequenced at Broad, not with the UK Biobank cohort). Final thresholds were RF true positive probability of 0.4 (approximately 87% of SNVs in well-covered intervals) for single nucleotide variants and 0.2 (approximately 70% of indels in well-covered intervals) for indels. Finally, we also excluded variants with two hard filters:

- Excess heterozygotes defined by an inbreeding coefficient < -0.3
- Variants where no sample had a high quality genotype (see Genotype QC below)

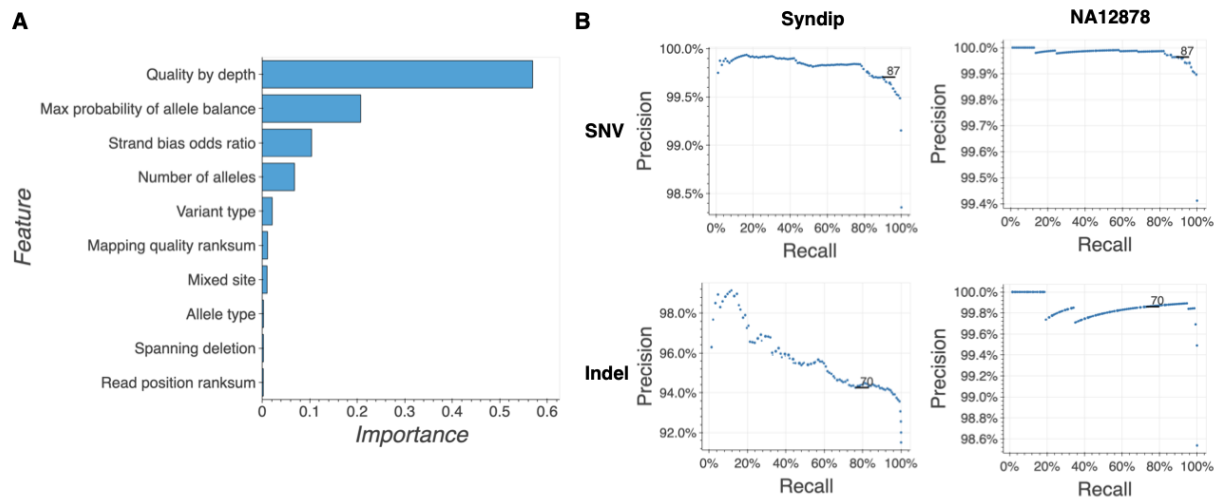


Fig. S6 | Variant QC **(A)**: A summary of the features used in the random forests model and their relative importance in the model generated. **(B)**: Precision and recall curves for the random forest classifier using two truth samples present in our data (NA12878 and syndip). The highlighted points at 87 for SNVs and 70 for indels indicate the cutoffs used for variant filtering.

Genotype QC

We filtered genotypes based on the previously defined “adj” criteria, with a modification for haploid calls on chrX and chrY for XY individuals. Specifically, we filtered to genotypes where depth ≥ 10 (5 for haploid calls), genotype quality ≥ 20 , and minor allele balance > 0.2 for all alternate alleles for heterozygous genotypes.

Annotations

Variants were annotated using VEP v95 as implemented in Hail using the default parameters for GRCh38 (including LOFTEE (2)).

For downstream analyses, variants were grouped by Ensembl Gene ID and functional impact as follows:

- pLoF: High-confidence LoF variants (as indicated by LOFTEE), including stop-gained, essential splice, and frameshift variants, filtered according to a set of first principles as described at <https://github.com/konradjk/loftee>.

- missense|LC: Missense variants are grouped with in-frame insertions and deletions, as well as low-confidence LoF variants (filtered out by LOFTEE). The latter have a frequency spectrum consistent with missense variation (2), and affect a set of amino acids in a similar fashion (e.g. a frameshift in the final exon).
- synonymous: All synonymous variants in the gene.

Scaling association testing using Hail Batch

Jacqueline I. Goldstein, Daniel King, Konrad J. Karczewski, Benjamin M. Neale, Cotton Seed

In order to perform the association analysis using SAIGE-GENE, we developed a new scientific compute scheduler, Hail Batch. Hail Batch is a cloud-based, serverless, multi-tenant platform as a service (PaaS). To use Hail Batch, users construct a computational graph of jobs to be executed, called a batch, using a Python client library (or manually). The batch is then submitted to Hail Batch via a REST API. The Hail Batch scheduler both manages pools of worker virtual machines (VMs) on which to schedule user jobs and schedules jobs on those workers. Hail Batch includes a Web UI for monitoring batches and viewing individual job logs. The documentation can be found here: <https://hail.is/docs/batch/index.html>. Hail Batch is fully open source and is contained in the Hail project monorepo which can be found here:

<https://github.com/hail-is/hail/tree/main/batch>.

Motivation

We built Hail Batch because we wanted a serverless solution with zero operational overhead for users. Based on our experience building Hail Query on Apache Spark, even with cloud platform managed services for running Spark like Google Dataproc, configuring, provisioning and managing compute clusters is a significant operational burden each user must bear (and scales with the number of users). In addition, by operating a multi-tenant compute cluster, we increase the utilization of resources amongst all users. The benefits are especially pronounced when users are working on iterative analyses (as opposed to batch processing) where they might need time to assess a result before moving on to subsequent analyses.

Bioinformatic tools come in a variety of forms, from standalone binaries and command line tools like GATK and SAIGE, to Python and R packages, to cloud-native analytics tools like Hail Query. We wanted a scheduling infrastructure that would support building data processing

pipelines across all these tool modalities. Hail Batch currently supports executing containerized command line tools and serialized Python functions. A Hail Batch based backend for Hail Query which executes JVM bytecode is under development.

Finally, we wanted to support low scheduling overhead at a large scale so that users could decompose pipelines based on natural biological or data considerations rather than computational constraints. For example, running millions of relatively fast statistical tests (seconds or more) for permutation testing requires small scheduling overhead to be effective. The association analysis described here naturally decomposed into a job per megabase per phenotype, for a total of approximately 18 million jobs.

We follow the principle that the systems we build today should themselves become building blocks of the systems we build tomorrow just as the association pipelines described herein build on Hail Batch. Therefore, we wanted a system that had a native programmatic interface so it could be used by other systems. For example, the Hail team's continuous integration (CI) system runs tests and deployments using Hail Batch. A future application of Hail Batch is serving as the underlying execution engine for an incremental joint caller service.

User interface

Hail Batch provides a Python library to construct and execute batches. A simple example is shown in Fig. S7. Rather than focus on the Python interface, which is described in detail in the documentation, we will focus on the main conceptual pieces of a Hail Batch computational graph.

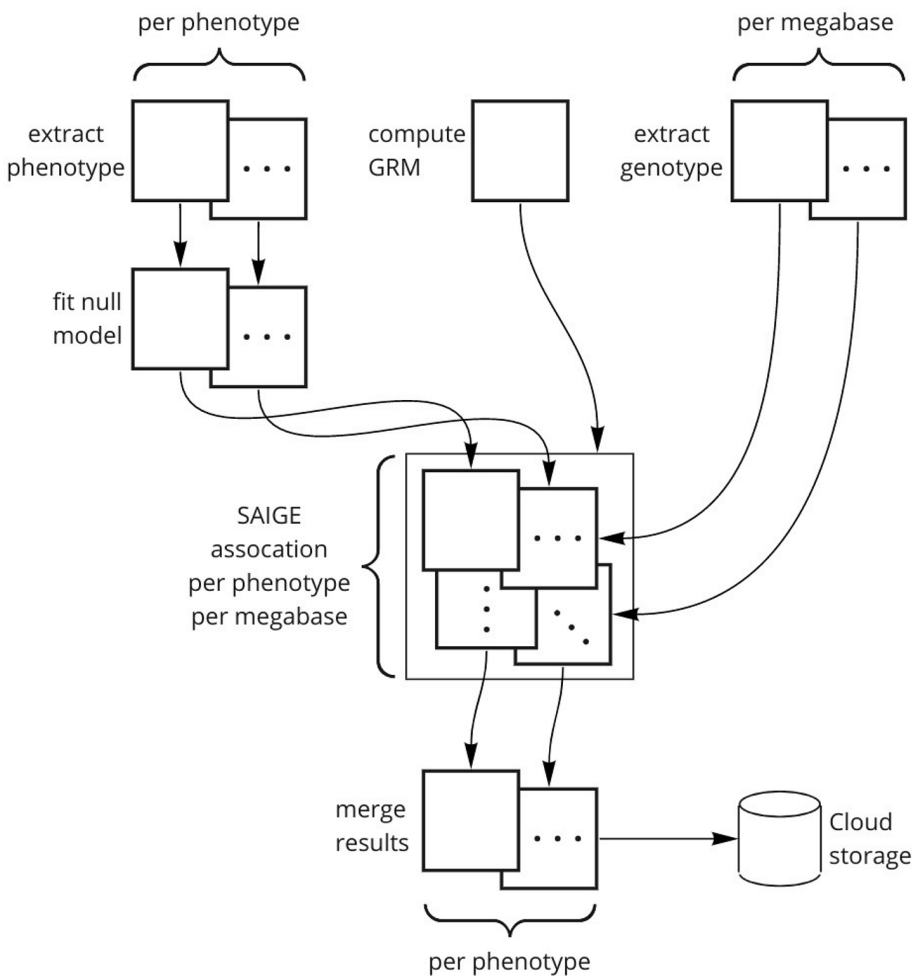


Fig. S7 | Hail Batch schematic for SAIGE association analysis. An example batch (the SAIGE pipeline used in this manuscript) is shown here.

A batch is the unit of submitted work. A batch primarily consists of (1) user-defined attributes, and (2) an ordered list of N jobs. The user-defined attributes are for searching and identifying batches. Jobs are the individual units of scheduling and can depend on previous jobs to form a directed acyclic graph (DAG) representing the full computational pipeline. The main parts of the job description are: (1) user-defined attributes, (2) dependencies, (3) inputs and outputs, (4) compute resources, and (5) executor configuration. Jobs do not run until all of their dependencies have completed. There are mechanisms for cancellation and controlling whether

jobs run if their dependencies failed or were cancelled. Compute resources (CPU, memory, disk and machine type) describe resources needed to schedule the job, which will be provisioned by the autoscaler as needed. We refer the reader to the documentation for more details.

During the submission and execution of a batch, jobs pass through the following states: pending (dependencies are not complete), ready (dependencies are complete and the job is ready to be scheduled), creating (the job is provisioning resources for custom resource configurations), running (the job has been scheduled), and the terminal states: error (the executor failed to run the job), cancelled, failed (the job execution failed) or successful. The REST API and Web UI allow users to monitor running and historical batches and view individual job logs.

There are currently two backends for executing a batch: (1) a local backend that runs jobs locally and (2) a service backend that submits jobs to the Hail Batch service. Currently, the Hail Batch service supports the Google Cloud Platform.

Implementation

Hail Batch is written in Python and makes extensive use of `asyncio`. Its deployment consists of three parts: (1) the front-end `batch` service, (2) `batch-driver` service, and (3) a MySQL database instance. The `batch` service serves the Web UI and user-facing REST API queries. It is stateless and autoscales based on incoming traffic. The `batch-driver` runs the job scheduler, the autoscaler which provisions worker VMs, and the admin Web interface. Job and batch configuration state is stored in MySQL and job logs are stored in Google Storage. Hail Batch relies on a Hail PaaS `auth` service for session ID based authentication. The Hail PaaS services are deployed in Kubernetes and the autoscaler uses the Google Compute Engine (GCE) directly to provision workers. Hail Batch deployment is controlled by the `ci` (continuous integration) service. We also maintain Terraform scripts for bootstrap and disaster recovery of Hail PaaS installations.

The autoscaler in the driver service is organized in terms of instance pools. There are two types of instance pools, shared and private. There are three shared pools for the three main GCP machine types. Shared pools autoscale based on the pool size and the number of ready jobs that run in that pool. Batch pools can scale down to zero, but we also support a minimum size so small jobs can be dispatched immediately without waiting to spin up resources. Shared pool workers schedule jobs across all users and resource utilization is higher because instance startup and shutdown is generally amortized over many jobs. By default, preemptible n1-standard-16 machines are used, but jobs that use special machine types or require non-preemptible instances can run in a private pool which provisions a VM per job.

In addition to autoscaling, the Hail Batch scheduler implements a fair share algorithm across users in order to provide a responsive experience. Jobs are scheduled per user, in reverse order of the amount of resources already allocated. If there are not enough ready jobs to saturate a user's allocation, the unused allocation is made available to the remaining users. This means new jobs submitted by a user to an active cluster will scale up quickly to the user's share and all users enjoy a more responsive experience when the system is actively used.

We make a few remarks on the scheduler performance. The scheduler is able to schedule ~80 jobs/s. The maximum cluster size the scheduler can support is a function of the expected job length. The maximum cluster size is (average job length in second) * 80. Therefore, for the 15-30m SAIGE jobs used in the association analysis here, the scheduler is theoretically (and was practically) able to saturate a ~100K core pool of workers.

Hail Batch currently supports two executors on worker VMs: Docker and the Java Virtual Machine (JVM). The JVM executor is used by Hail Query service. The Docker executor includes the details necessary to run a docker container: image, environment variables, command line, etc. Jobs run on the worker VMs in three steps: input, main and output. The input and output steps are responsible for copying between object storage and the local filesystem. For copying, we developed a pluggable Python asyncio filesystem abstraction and high-performance parallel

copy management engine which supports local files, Google Cloud Storage, S3 and HTTP(S) (read-only). The main step executes the user's code.

Association testing and quality control

Konrad J. Karczewski, Wenhan Lu, Jacqueline I. Goldstein, Daniel King, Wei Zhou, Cotton Seed, Benjamin M. Neale

Association testing framework

We performed association testing on the quality-controlled genotype data using SAIGE-GENE (5), following the recommendations by the authors. The genetic relatedness matrix (GRM) was computed using a dataset sampled from allele frequency categories from the genotype MatrixTable considering only autosomal variants with a minimum call rate of 95%, including approximately 2,000 variants from each of Allele Count (AC) 1-5, AC 6-10, and AC 11-20; and approximately 10,000 variants from each of: AC 20-AF 0.1%, AF 0.1-1%, AF 1-10%, AF > 10%. These variants were LD-pruned to $r^2 = 0.1$, and exported into PLINK format. A sparse GRM was computed using step0 of SAIGE-GENE using the default parameters, with 2,000 markers used for the kinship matrix, and a relatedness cutoff of 0.125. We further created a “gene map” file for each megabase, which included information about the variants to be analyzed together in each group test. We included 3 groups: pLoF variants, including only those annotated as high-confidence by LOFTEE; missense-like variants, including missense variants and variants annotated as low-confidence by LOFTEE; and synonymous variants. The script for pre-processing is available at

https://github.com/Nealelab/ukb_exomes/blob/master/hail/pre_process_saige_data.py.

The remainder of the process was parallelized using Hail Batch (Fig. S7). For each megabase of the genome, we exported a BGEN from the genotype MatrixTable with all variants that lie in genes that have a starting coordinate within that megabase. For each phenotype, we exported a flat file from the phenotype MatrixTable with the covariates used for analysis: age, sex, age², and 20 principal components, as well as interaction terms of age * sex and age² * sex. The phenotype data was combined with the sparse GRM computed above to fit a null model (without genotypes) using step1 of SAIGE-GENE with the default parameters and the

covariates described above. Finally, the genotype tests were run using the BGEN from each megabase, and the null model from each phenotype using step2 of SAIGE-GENE with the default parameters, plus maxMAF=0.5, LOCO=FALSE, and IsSingleVarinGroupTest=TRUE. The results across all megabases were loaded into two Hail Tables for each phenotype, for the group tests and single variant tests. The pipeline is available at https://github.com/Nealelab/ukb_exomes/blob/master/saige_exomes.py with helper scripts that can be found at https://github.com/Nealelab/ukb_common/blob/master/utis/saige_pipeline.py.

We combined the phenotype-level Hail Tables into a Hail MatrixTable using a hierarchical merge, along with phenotype metadata from SAIGE, resulting in one MatrixTable for the group tests and one for the single variant tests. We computed lambda GC values for each phenotype and gene (see below) using the *hl.methods.statgen._lambda_gc_agg* aggregator in Hail. These datasets are publicly released and serve the browser framework described below.

Random phenotype analysis

In order to test the asymptotic properties of our tests, we simulated 314 heritable phenotypes using the sparseMVN package in R (6), using the genetic relatedness matrix generated by SAIGE. From these normal distributions, we simulated continuous phenotypes, as well as binary phenotypes with varying prevalences from 10^{-4} to 50%. We further generated a series of phenotypes of varying heritabilities by introducing an additional noise component (rnorm) and weighting by the square root of the desired heritability. We performed association testing on these phenotypes and computed lambda GC values as above, and here, we show the qq-plots for the single-variant and group (SKAT-O and Burden) tests (Fig. S8).

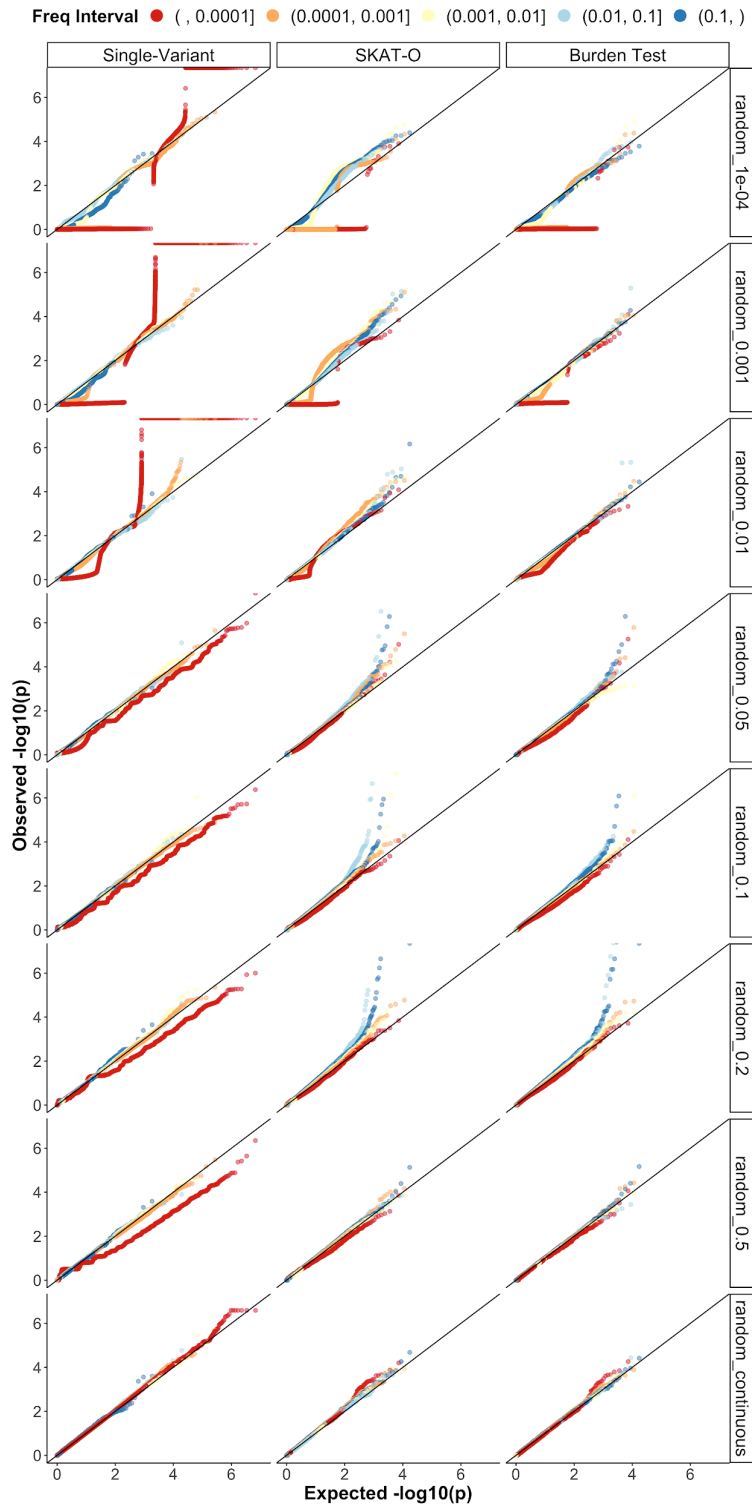


Fig. S8 | QQ-plots of randomly generated heritable (heritability = 100%) phenotypes for single-variant tests (left) and for group tests (SKAT-O, middle; and burden tests, right). The increasing prevalence of each binary phenotype is indicated by the label on the right (1e-4 to 0.5), followed by continuous traits.

We summarize the information in these plots using the lambda GC metric, which is shown in fig. S9. Notably, in Figs. S8-9, we can see increased instability of the QQ-plot and lambda values for rarer variants especially for rarer outcomes, suggesting the need for an allele frequency threshold for large-scale analyses. This is consistent with the minimum frequency and prevalence required to achieve statistical significance at this sample size (Fig. S10).

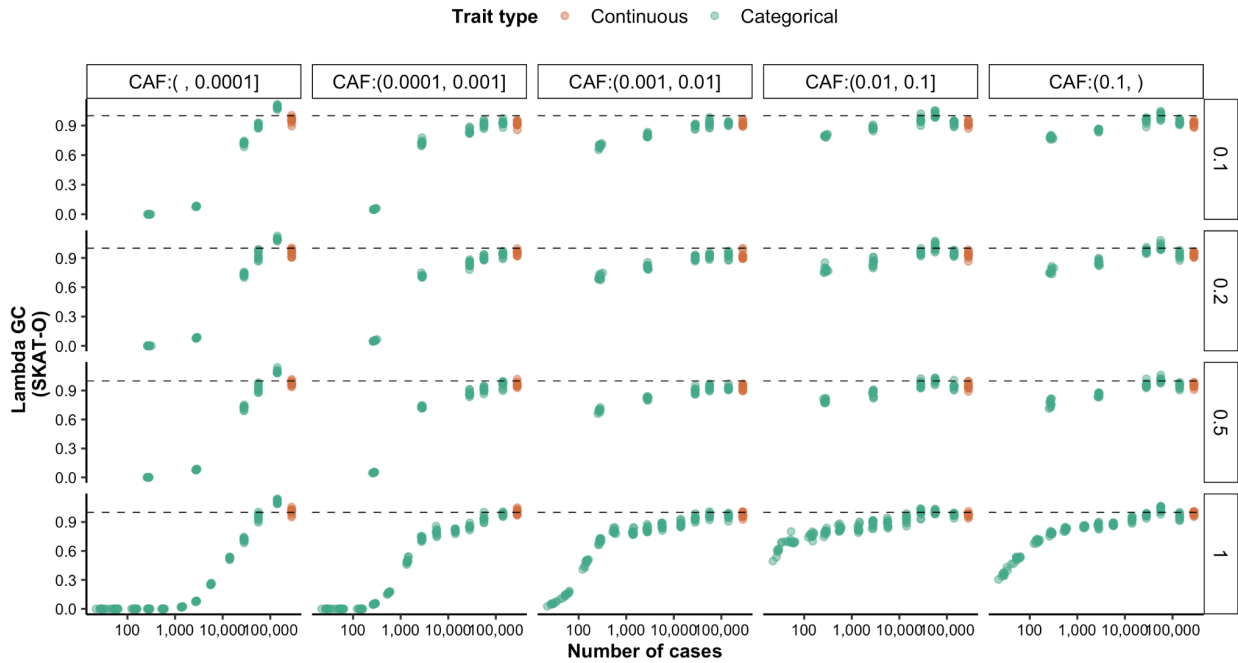


Fig. S9 | Lambda GC by cumulative allele frequency (CAF) by heritability. Additional heritability fractions are shown by the label on the right.

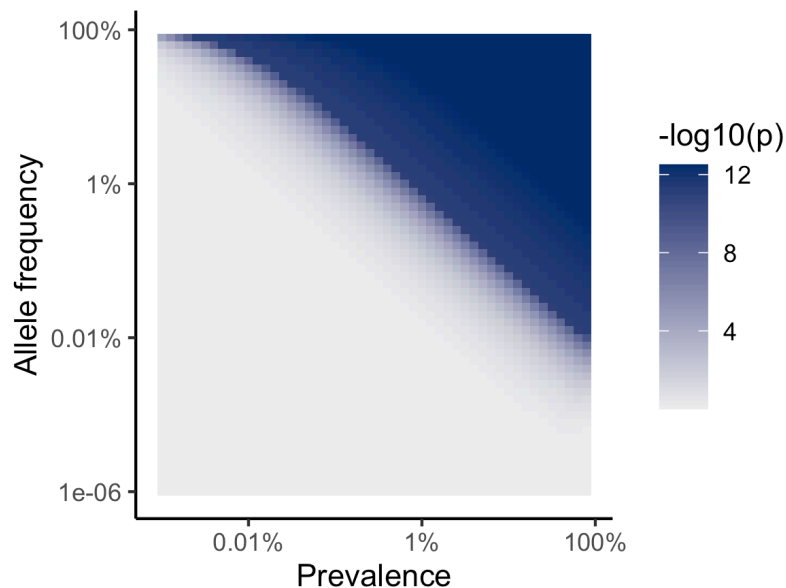


Fig. S10 | Power for rare variant associations. The minimum p-value possible from a protective mechanism with an odds ratio = 0: here, we compute the p-value of a chi-squared test of the case where the variant is absent from cases, while controls have a frequency as plotted. For the color-scale, a second logarithm is applied to p values below 10^{-10} .

To compute an effective number of tests and thus a p-value threshold for each phenotype, we took the minimum p-value for each of the simulated continuous phenotypes: the median of this value across all phenotypes was 5×10^{-6} for SKAT-O tests, 1.3×10^{-5} for burden tests, and 1.6×10^{-7} for single-variant tests. Thus, for downstream analyses, we computed the experiment-wise significance threshold as 2.5×10^{-8} for SKAT-O tests, 6.7×10^{-7} for burden tests, and 8×10^{-9} for single-variant tests.

Allele frequency filters

We computed the cumulative allele frequency (CAF) for each gene by aggregating the allele frequencies of variants of the same annotation group within each gene. For each group test (SKAT-O, SKAT and Burden test), we filtered the test results to genes from each of the

CAF intervals: $[0, 0.0001]$, $(0.0001, 0.001]$, $(0.001, 0.01]$, $(0.01, 0.1]$, and $(0.1, \infty)$ and computed lambda GC values for each phenotype within each frequency bin. Lambda GC values for categorical phenotypes converge to around 1 as CAF increases (Fig. S11). Lambda GC values for continuous phenotypes are not significantly affected by the change of CAF. A smaller number of cases and lower level of CAF results in less stable values of lambda. Because of the highly deflated pattern of lambda GC values observed in genes with CAF 0-0.0001, we filtered out genes with $CAF < 0.0001$.

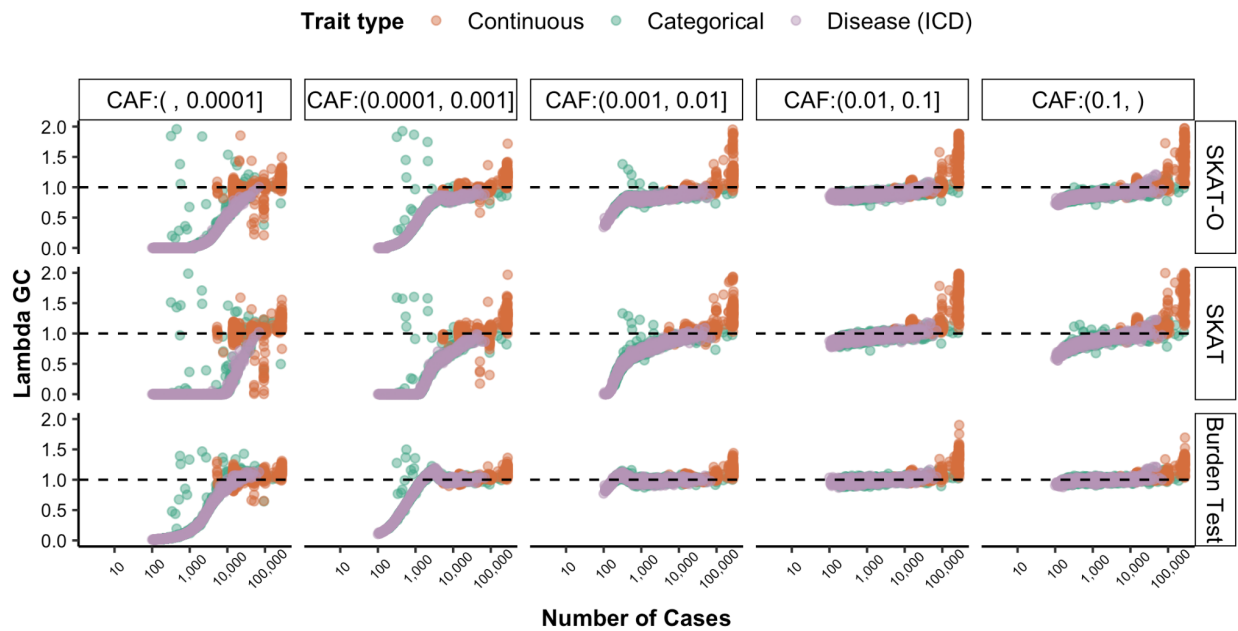
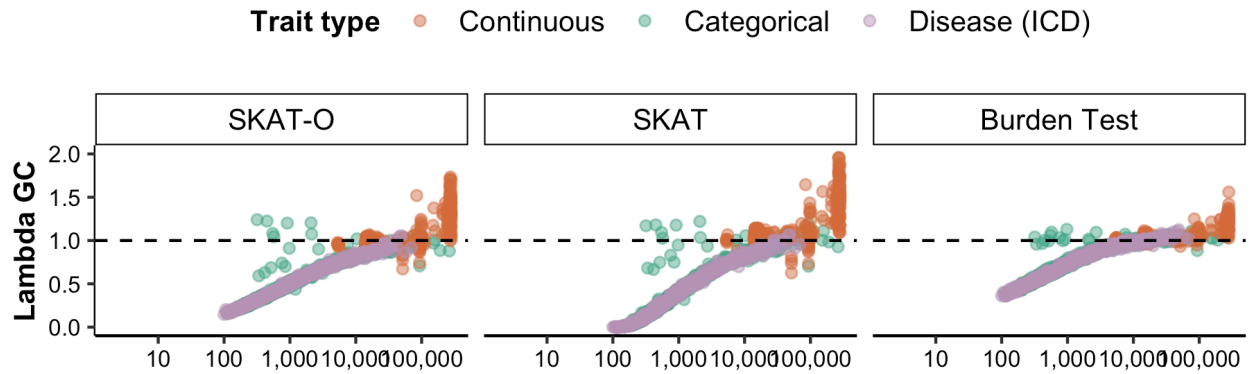


Fig. S11 | Lambda GC for each phenotype vs case count, split by CAF interval for SKAT-O, SKAT and burden tests.

Test calibration filtering

Removing genes with $CAF < 0.0001$ results in improved lambda GC values (fig. S12). Further, we filtered association statistics with standard errors (SE) of 0, as these were the result of a bug in the version of SAIGE-GENE used herein (which has since been corrected).

(A) Before



(B) After

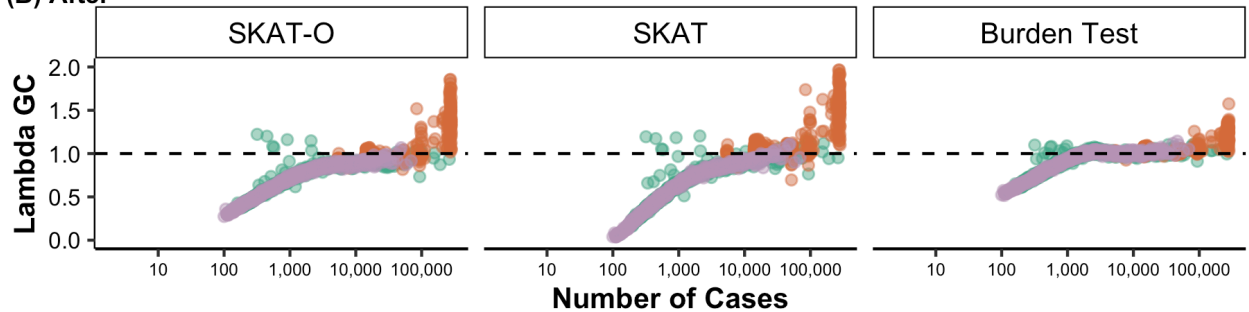
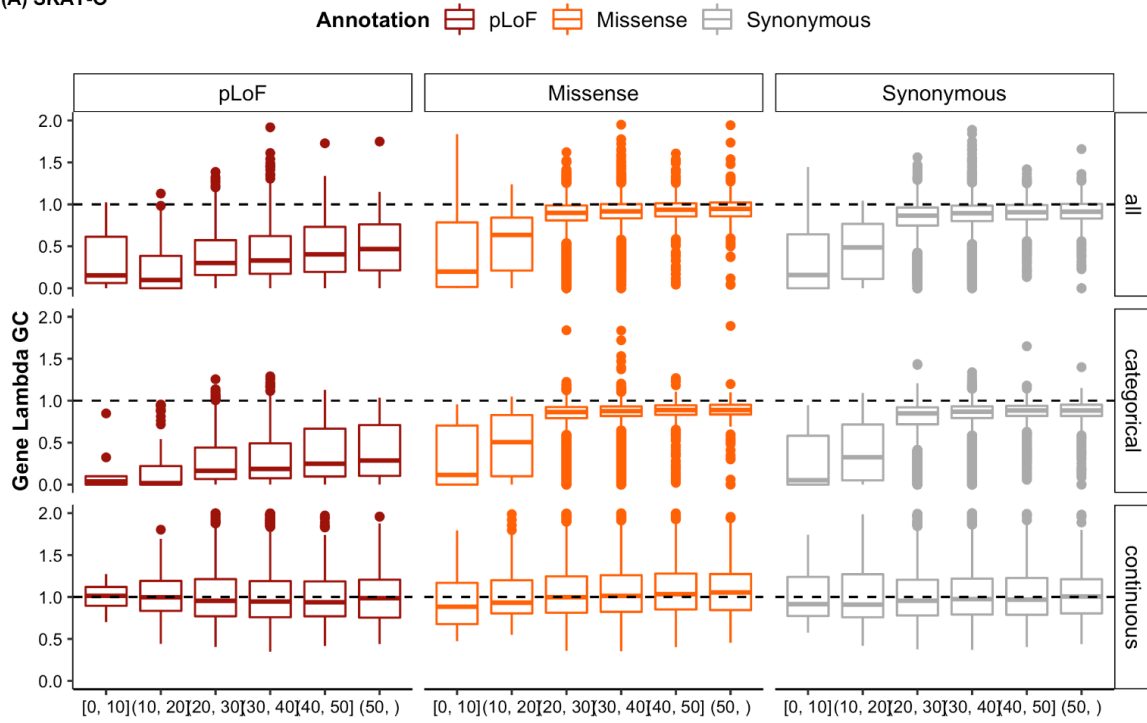


Fig. S12 | Lambda GC for each phenotype vs case count for SKAT-O, SKAT and burden tests, before and after filtering out genes with CAF < 0.0001 and coverage < 20.

Finally, due to the large number of phenotypes available, we devised a metric for the calibration of individual genes, a lambda GC for each gene across phenotypes. For most genes, this metric is well-behaved for synonymous variants (95% range). However, it also marks outliers for removal, and appears to be correlated with mean sequencing coverage (Fig. S13). Thus, we removed genes with coverage < 20 as well as genes that have a synonymous lambda GC < 0.75.

(A) SKAT-O



(B) Burden Test

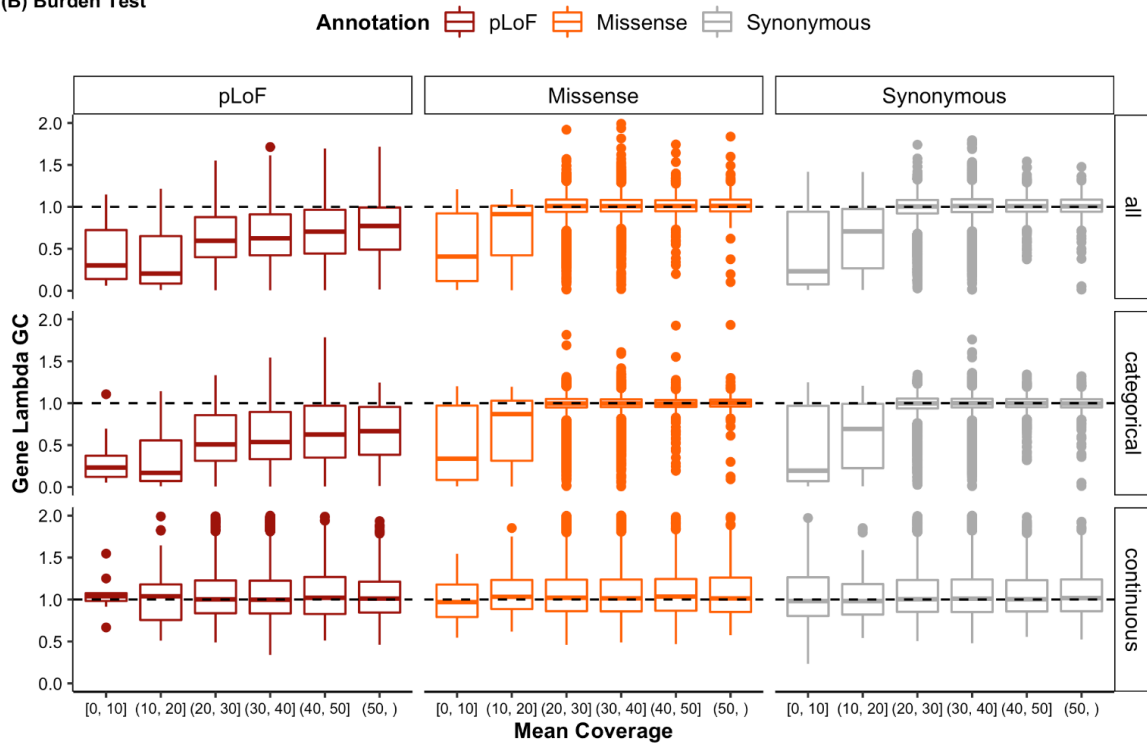


Fig. S13 | Coverage vs gene-based lambda GC, for SKAT-O (A) and burden tests (B).

After applying the above filters, we recompute lambda GC for each phenotype (Fig. S14) and gene (Fig. S15). For downstream analyses, we filter to phenotypes with lambda GC at least 0.75.

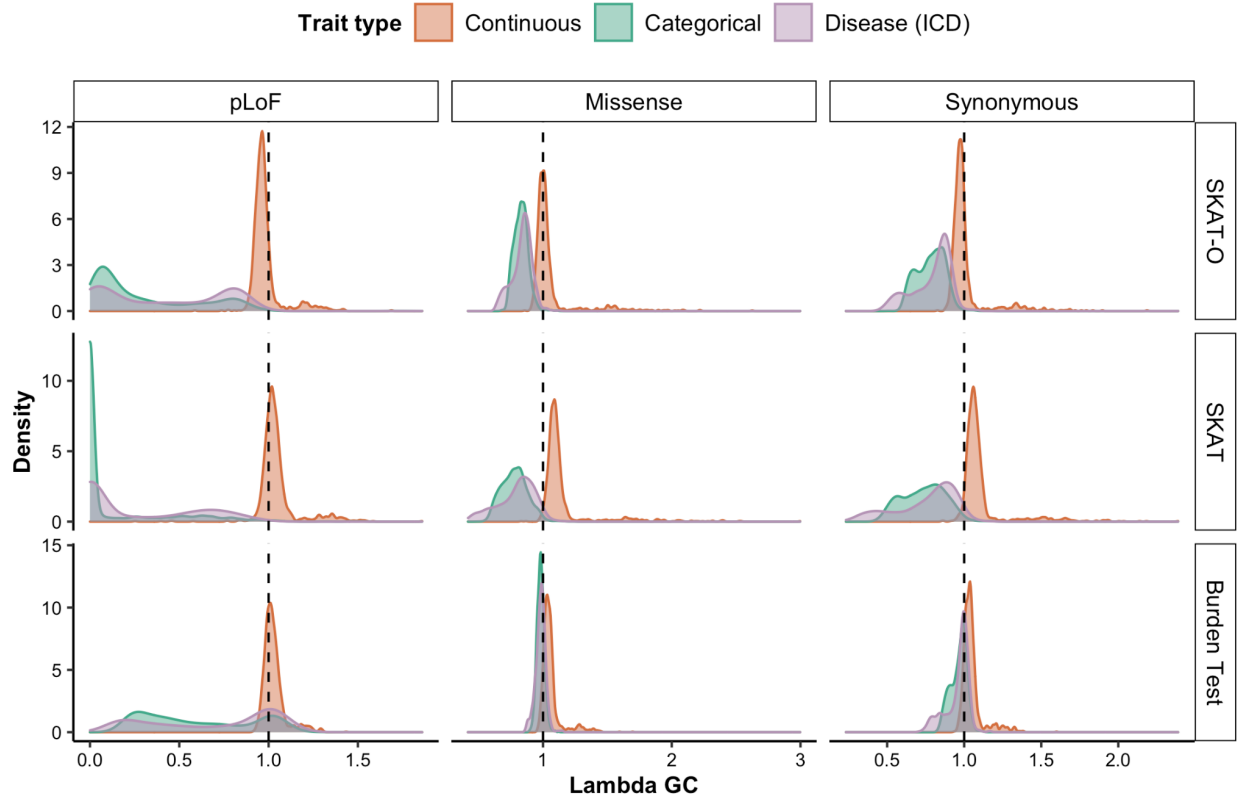


Fig. S14 | Lambda GC for each phenotype. The distribution of lambda GC values for each phenotype is shown, broken down by trait type, test type, and set of variants used in the lambda calculation.

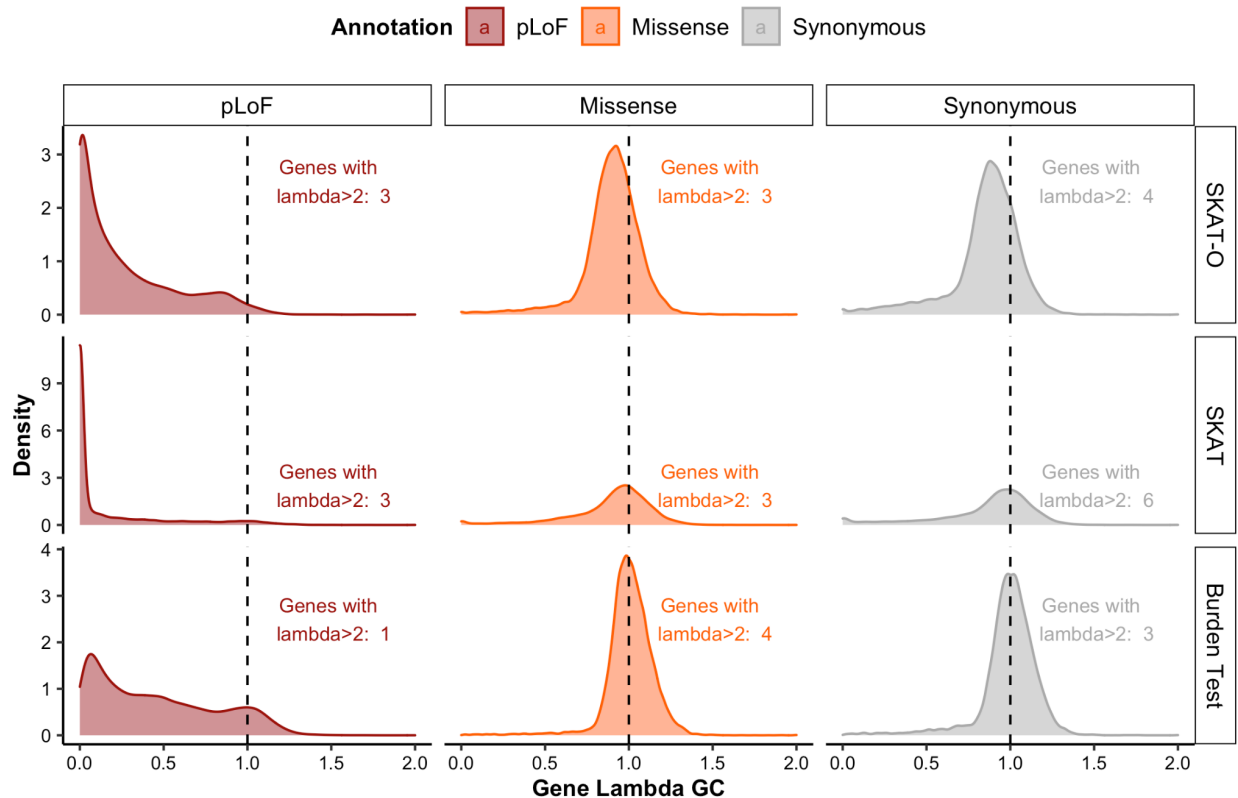


Fig. S15 | Lambda GC for each gene. The distribution of lambda GC values for each gene is shown, broken down by test type and set of variants used in the lambda calculation.

Independent phenotypes

For large-scale analyses, we pruned to a set of relatively uncorrelated phenotypes. Using the UKB phenotype MatrixTable, we generated a pairwise correlation table using a matrix multiplication of the table and its transpose (Fig. S16A), and filtered the table to phenotype pairs with correlations (r^2) over 0.5. We then applied the *maximal_independent_set* function in Hail to the remaining phenotype pairs with a tie-breaker function preferring phenotypes with more cases, resulting in a set of 602 related phenotypes to remove from the dataset (Fig. S16B). A summary of the final QC steps is shown in Table S2.

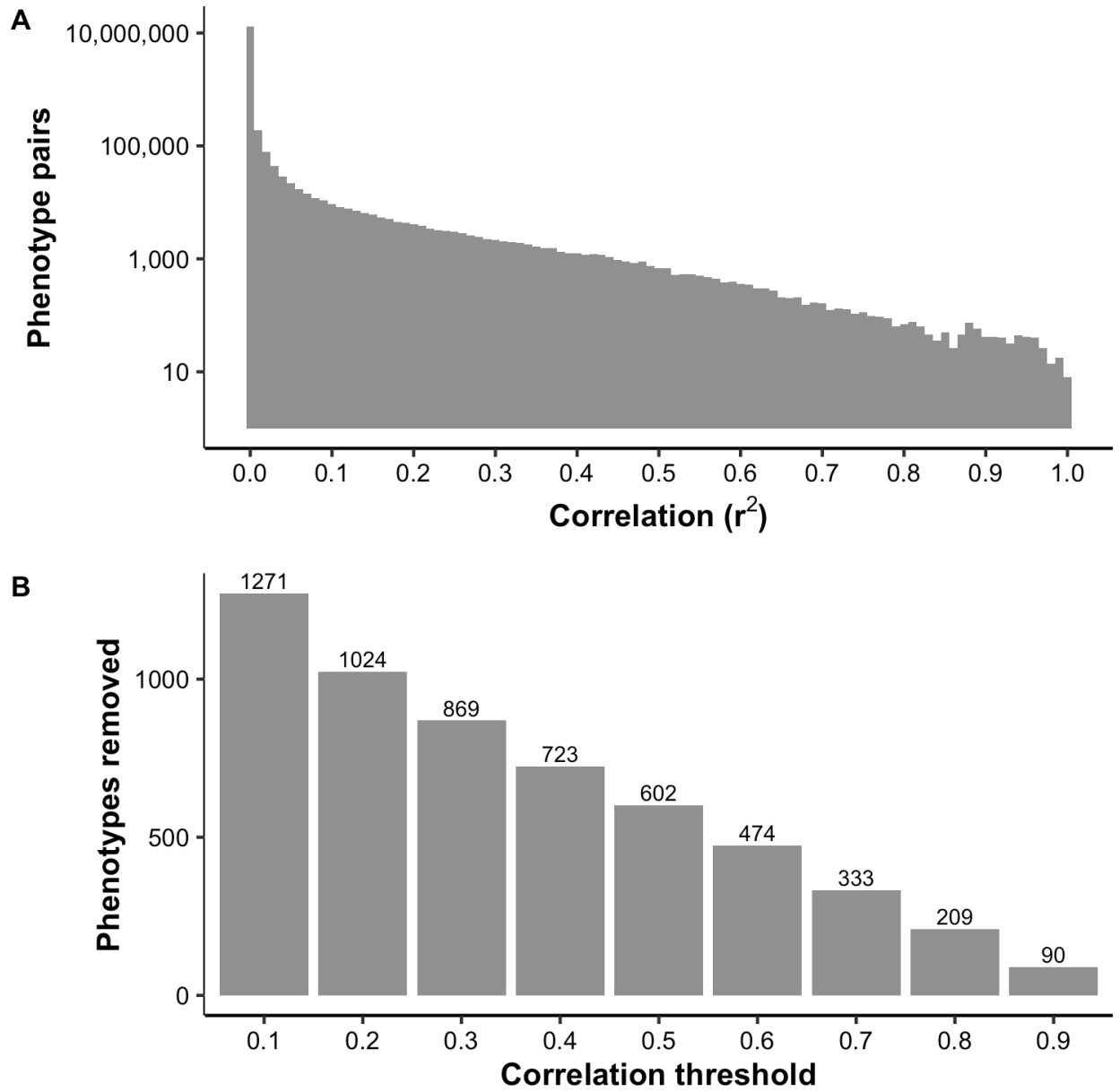


Fig. S16 | Independence of phenotypes. **(A)**: A histogram of the number of phenotype pairs by correlation (r^2). **(B)**: The number of phenotypes that would be removed by the maximum independent set method, by r^2 threshold.

Table S2 | QC of summary statistics. All filters are applied sequentially.

Description		Count (% Percentage Remaining)			
		pLoF	Missense	Synonymous	Total
Group (SKAT-O)	Before filtering	18,527	19,578	19,545	57,650
	Number of variants >= 2	17,930 (96.8%)	19,551 (99.9%)	19,499 (99.8%)	56,980 (98.8%)
	Mean coverage > 20	17,728 (95.7%)	19,236 (98.3%)	19,219 (98.3%)	56,183 (97.5%)
	CAF > 0.01%	11,171 (60.3%)	19,183 (98.0%)	19,109 (97.8%)	49,463 (85.8%)
	Lambda of the synonymous group above 0.75	9,558 (51.6%)	15,457 (79.0%)	15,477 (79.2%)	40,492 (70.2%)
Variant	Before filtering	455,445	4,951,755	2,164,579	7,575,993 (NA: 4,214)
	Annotation defined				7,571,779 (99.9%)
	AF > 0.01%	8,865 (1.9%)	208,723 (4.2%)	132,353 (6.1%)	349,941 (4.6%)
		Continuous	Categorical	Disease (ICD)	Total
Phenotype (SKAT-O)	Before filtering	1,117	1,902	681	3,700
	Lambda above 0.75	1,117 (100%)	523 (27.5%)	310 (45.5%)	1,950 (52.7%)
	Correlation < 0.5	559 (50.0%)	493 (25.9%)	310 (45.5%)	1,362 (36.8%)

Comparison to known hits

We compared the significant associations with height discovered from our results with the 91 height-associated variants ($p < 2 \times 10^{-7}$) and 10 height-associated genes ($p < 5 \times 10^{-7}$) discovered in GIANT (7). Among the 91 rare/low-frequency variants associated with adult height in GIANT, 43 of the variants were found to be associated with height at $p < 8 \times 10^{-9}$, and 81 variants were found to be associated with height at $p < 0.05$ in UK Biobank (Table S3-

Table S4). Among the 10 genes associated with height in GIANT, only UGGT2 is found associated with height at $p < 2.5 \times 10^{-8}$ (SKAT-O), and all but one were found at nominal significance ($p < 0.05$) for either missense or pLoF variants (Table S5).

Table S3 | Comparison to 32 rare (MAF < 1%) variants associated with adult height in GIANT; in UK Biobank, 16 of these variants are found to be associated with height at $p < 8 \times 10^{-9}$ (blue), and 30 are associated with height at $p < 0.05$ (light blue).

Locus	Allele (Ref)	Allele (Alt)	Annotation	Gene	P-value			
					UK Biobank	GIANT		
						Discovery	Validation	Combined
1:32673514	G	C	missense	IQCC	4.47E-09	7.92E-08	3.83E-06	1.34E-12
1:41540902	G	A	missense	SCMH1	6.97E-22	1.58E-25	9.42E-13	1.35E-36
1:41618297	G	A	missense	SCMH1	1.39E-14	1.92E-15	1.32E-08	1.80E-22
1:149902342	C	T	missense	MTMR11	8.91E-11	4.16E-06	7.11E-06	3.03E-10
1:183495812	A	G	missense	SMG7	1.25E-08	4.97E-11	8.94E-05	1.61E-14
1:223178026	T	C	missense	DISP1	3.14E-08	1.11E-09	1.22E-06	1.27E-14
2:219920461	T	A	missense	IHH	4.12E-06	1.09E-15	1.48E-09	1.85E-23
2:220078652	C	T	missense	ABCB6	2.32E-13	3.43E-13	4.40E-04	2.47E-15
3:46939587	C	T	missense	PTH1R	2.14E-06	1.30E-11	5.48E-10	1.14E-19
4:73179445	C	T	missense	ADAMTS3	5.00E-09	1.82E-08	1.32E-04	1.30E-11
4:120422407	T	G	missense	PDE5A	2.77E-08	7.50E-17	1.28E-08	2.65E-23
5:32784907	G	A	missense	NPR3	6.37E-16	1.05E-08	1.78E-06	7.91E-14
5:64766798	G	A	missense	ADAMTS6	1.01E-11	7.82E-09	1.37E-08	4.80E-16
5:127668685	G	T	missense	FBN2	6.44E-22	2.47E-33	5.06E-20	1.47E-52
5:172755066	C	A	missense	STC2	2.96E-29	5.69E-15	1.32E-17	1.15E-30

6:155450779	A	G	missense	TIAM2	NA	1.45E-08	8.50E-01	3.96E-08
7:73482987	G	A	missense	ELN	1.95E-08	2.63E-06	1.51E-03	2.31E-08
8:135614553	G	C	missense	ZFAT	3.86E-37	4.42E-26	1.20E-14	6.12E-38
8:135622851	G	A	missense	ZFAT	5.82E-08	1.54E-12	5.94E-18	2.05E-28
11:27016360	G	A	missense	FIBIN	7.06E-08	5.79E-12	1.56E-03	3.26E-14
11:94533444	G	A	missense	AMOTL1	8.90E-05	9.01E-16	3.84E-07	2.84E-21
12:58138971	G	A	missense	TSPAN31	1.46E-01	8.26E-08	2.85E-03	5.50E-09
12:121756084	G	A	missense	ANAPC5	1.65E-15	1.09E-11	1.44E-11	1.45E-21
15:44153571	C	T	missense	WDR76	5.04E-03	1.56E-06	3.42E-04	2.32E-09
15:89424870	G	T	missense	HAPLN3	1.94E-23	2.84E-13	2.43E-11	1.02E-22
16:31474091	A	G	missense / splice acceptor	ARMC5	5.27E-08	5.88E-12	1.16E-03	1.62E-13
16:47684830	C	A	missense	PHKB	4.97E-08	3.96E-14	1.04E-01	3.43E-12
16:67470505	G	A	missense	HSD11B2	3.78E-04	1.27E-07	3.38E-04	1.97E-10
16:84900645	G	A	missense	CRISPLD2	4.54E-10	9.13E-12	4.34E-09	2.92E-19
16:84902472	G	A	missense	CRISPLD2	6.08E-16	7.75E-14	3.49E-08	2.36E-20
16:88798919	G	T	missense	PIEZO1	5.02E-12	5.27E-12	1.99E-08	8.68E-19
X:66941751	C	G	missense	AR	3.84E-06	7.05E-07	7.12E-09	2.67E-14

Table S4 | Comparison to 59 low-frequency (MAF between 1% and 5%) variants associated with adult height in GIANT; in UK Biobank, 7 of the variants were not tested, 27 of these variants are found to be associated with height at $p < 8 \times 10^{-9}$ (blue), and 51 are associated with height at $p < 0.05$ (light blue).

Locus	Allele (Ref)	Allele (Alt)	Annotation	Gene	P-value	
						GIANT

					UK Biobank	Discovery	Validation	Combine d
1:51873967	G	A	missense	EPS15	3.96E-12	5.07E-08	7.60E-11	2.56E-17
1:119427467	A	C	missense	TBX15	4.21E-20	1.61E-24	4.19E-15	2.79E-36
1:150551327	G	A	missense	MCL1	2.00E-16	2.16E-09	7.86E-12	1.55E-19
1:154987704	C	T	missense	ZBTB7B	4.26E-07	7.30E-17	4.46E-10	3.46E-25
1:180886140	C	T	missense	KIAA1614	1.26E-05	1.41E-06	4.51E-04	2.63E-09
2:20205541	C	T	missense	MATN3	7.28E-29	2.67E-23	6.60E-19	3.74E-41
2:219949184	C	T	intron	NHEJ1	NA	5.96E-21	1.12E-15	8.20E-37
2:179474668	G	A	missense	TTN	NA	1.35E-07	2.15E-01	3.44E-07
2:233077064	A	G	intron	DIS3L2	NA	2.35E-16	2.58E-15	6.46E-31
3:14214524	G	A	missense	XPC	1.05E-06	1.22E-08	1.68E-02	1.29E-08
3:47162886	C	T	missense	SETD2	7.73E-07	2.24E-08	2.22E-07	1.65E-13
3:49162583	C	T	missense	LAMB2	2.63E-28	3.28E-12	1.33E-16	3.49E-27
3:98600385	T	C	missense	DCBLD2	2.46E-04	1.23E-07	5.62E-05	1.68E-12
4:5016883	G	A	missense	CYTL1	3.98E-13	2.01E-17	6.68E-11	1.86E-25
4:87730980	C	T	missense	PTPN13	1.45E-21	1.94E-19	1.38E-15	9.43E-32
4:135121721	T	C	missense	PABPC4L	4.36E-05	1.39E-13	1.33E-04	7.54E-16
4:144359490	C	T	missense	GAB1	3.92E-05	1.04E-08	3.24E-04	4.29E-12
4:154557616	C	T	missense	TMEM131L	8.22E-07	7.75E-08	5.75E-06	2.18E-12
5:102338811	A	G	missense	PAM	1.88E-05	3.76E-06	8.47E-06	1.63E-10
5:126250812	C	T	missense	44258	1.94E-02	4.25E-08	2.45E-03	1.67E-10
5:135288632	A	G	missense	LECT2	8.76E-04	1.02E-07	4.77E-04	1.36E-09
5:172196752	A	G	missense	DUSP1	4.34E-10	4.00E-14	1.26E-06	1.93E-20

5:176637471	G	A	missense	NSD1	8.11E-17	2.38E-17	2.62E-12	4.27E-30
5:176722005	G	A	missense	NSD1	5.41E-26	1.86E-26	8.42E-18	2.32E-41
6:30851933	G	A	intron	DDRI	NA	1.11E-08	1.24E-05	4.64E-13
6:34730395	C	T	synonymous	SNRPC	1.29E-33	9.21E-33	9.59E-31	3.45E-60
6:41903798	C	A	missense	CCND3	2.41E-25	5.51E-17	3.41E-08	1.28E-22
7:99489571	G	A	3'UTR	TRIM4	NA	3.28E-10	2.26E-07	1.40E-17
7:100490077	G	A	synonymous	ACHE	NA	8.59E-10	2.92E-02	2.98E-10
7:135123060	G	C	missense	CNOT4	8.92E-13	2.31E-17	5.04E-10	3.90E-26
8:42226805	C	G	missense	POLB	2.45E-05	1.95E-06	1.30E-02	1.88E-07
9:34660864	C	T	missense	IL11RA	6.11E-08	5.20E-13	4.42E-03	4.01E-13
9:95063947	C	T	missense	NOL8	2.67E-03	2.56E-06	3.45E-02	3.33E-06
10:79580976	G	A	missense	DLG5	1.64E-09	2.72E-11	5.15E-11	7.66E-20
10:97919011	A	G	missense	ZNF518A	5.64E-06	9.94E-08	3.05E-03	3.91E-09
11:65715204	G	A	missense	TSGA10IP	2.34E-30	1.82E-21	1.41E-23	1.52E-43
12:7548996	C	G	missense	CD163L1	2.15E-02	4.11E-08	6.68E-02	1.87E-08
12:69140339	G	C	missense	SLC35E3	1.11E-08	1.13E-09	5.0E-04	1.29E-11
12:104408832	T	C	missense	GLT8D2	1.24E-09	8.72E-10	5.82E-10	1.60E-17
13:50842259	G	A	intron	DLEU1	NA	2.33E-37	7.02E-25	5.66E-57
14:23313633	G	A	missense	MMP14	1.82E-06	1.72E-08	7.81E-09	3.27E-16
14:24707479	G	A	missense	GMPR2	3.18E-10	3.67E-16	1.34E-11	2.13E-29
14:45403699	C	A	missense	KLHL28	2.43E-06	1.55E-06	4.13E-04	3.05E-09
14:70633411	C	T	missense	SLC8A3	1.45E-10	2.49E-11	2.02E-06	2.03E-16
14:94844947	C	T	missense	SERPINA1	5.66E-71	1.39E-45	2.50E-34	1.72E-75

14:101349454	G	T	missense	RTL1	2.04E-08	1.17E-11	2.12E-04	2.50E-15
15:34520687	T	C	missense	EMC4	2.91E-02	1.16E-06	2.19E-02	1.60E-07
15:72462255	C	T	missense	GRAMD2A	9.95E-21	8.72E-17	3.66E-13	1.28E-27
15:89388905	C	T	synonymous	ACAN	1.06E-105	4.30E-72	1.08E-56	3.79E-130
16:4812705	A	G	missense	ZNF500	2.88E-06	8.61E-17	2.34E-07	2.89E-21
16:24804954	A	T	missense	TNRC6A	8.81E-13	1.08E-09	1.65E-07	1.90E-15
16:67409180	G	A	missense	LRRC36	8.12E-11	1.08E-18	3.91E-13	6.40E-31
17:67081278	A	G	missense	ABCA6	1.28E-11	2.17E-06	5.58E-07	5.57E-12
18:74980601	A	T	missense	GALR1	7.29E-05	3.60E-18	3.64E-05	5.11E-19
19:45296806	C	T	missense	CBLC	9.43E-02	1.48E-07	1.19E-02	2.96E-08
19:55879672	C	T	missense	IL11	5.47E-44	1.02E-57	2.28E-23	5.32E-81
19:55993436	G	T	missense	ZNF628	3.50E-33	2.28E-18	1.17E-18	6.33E-34
22:28501414	C	T	missense	TTC28	3.68E-13	9.47E-11	3.24E-09	3.93E-19
22:42095658	T	G	missense	MEI1	9.73E-03	2.25E-08	6.59E-03	3.70E-10

Table S5 | Comparison to 10 genes associated with adult height in GIANT. In UK Biobank, only UGGT2 reaches our genome-wide significance threshold (blue), but all but one are found nominally significant for either pLoF or missense variants (light blue).

Gene	UK Biobank			GIANT P-value			
	Annotation	Burden Test	SKAT-O	SKAT-Broad	VT-Broad	SKAT-Strict	VT-Strict
B4GALNT3	missense LC	1.58E-05	9.15E-06	2.40E-05	1.90E-05	1.80E-05	3.10E-07
	pLoF	2.54E-06	4.42E-06				
	synonymous	3.09E-01	2.81E-01				
CCDC3	missense LC	1.58E-02	7.82E-08	6.30E-04	6.30E-06	3.00E-07	5.40E-09
	pLoF	1.09E-02	1.34E-02				
	synonymous	3.73E-01	5.63E-01				
CRISPLD1	missense LC	7.36E-02	1.07E-01	2.20E-07	6.70E-11	8.50E-06	8.90E-07
	pLoF	2.35E-04	5.43E-04				
	synonymous	5.30E-01	1.81E-01				
CSAD	missense LC	6.98E-02	1.02E-01	2.30E-08	2.40E-09	0.83	0.59
	pLoF	9.46E-01	7.35E-01				
	synonymous	8.64E-01	4.93E-04				
FLNB	missense LC	1.13E-02	2.51E-04	2.20E-06	5.10E-04	2.40E-09	3.20E-06
	pLoF	3.45E-06	8.50E-06				
	synonymous	7.62E-02	4.50E-03				
G6PC	missense LC	2.72E-01	2.10E-01	1.30E-05	3.60E-08	5.50E-06	1.30E-06
	pLoF	7.13E-03	1.23E-02				
	synonymous	7.55E-01	3.02E-01				
NOX4	missense LC	1.13E-04	3.77E-08	5.10E-06	1.40E-07	NA	NA
	pLoF	3.70E-02	6.43E-02				
	synonymous	4.85E-04	3.88E-04				
OSGIN1	missense LC	3.25E-03	3.00E-05	4.30E-11	4.50E-05	0.19	0.18

	pLoF	7.92E-01	3.65E-01				
	synonymous	8.68E-01	8.00E-01				
SNED1	missense LC	1.44E-03	9.77E-04	1.90E-05	4.30E-09	NA	NA
	pLoF	6.61E-01	1.79E-01				
	synonymous	1.74E-02	6.34E-03				
UGGT2	missense LC	2.86E-16	9.24E-18	3.00E-05	2.60E-07	2.30E-05	4.80E-07
	pLoF	2.52E-07	2.58E-07				
	synonymous	7.42E-03	7.03E-05				

We compared the effect sizes from our single-variant test results to those of GIANT.

Among 1,330 ExomeChip variants with p-value $< 2 \times 10^{-7}$ in the GIANT European-ancestry meta-analysis, 566 variants are tested in the UK Biobank data. The effect sizes of this group of shared variants are consistent between the two datasets, with a slight attenuation in results from UK Biobank, consistent with a degree of winner's curse (Fig. S17).

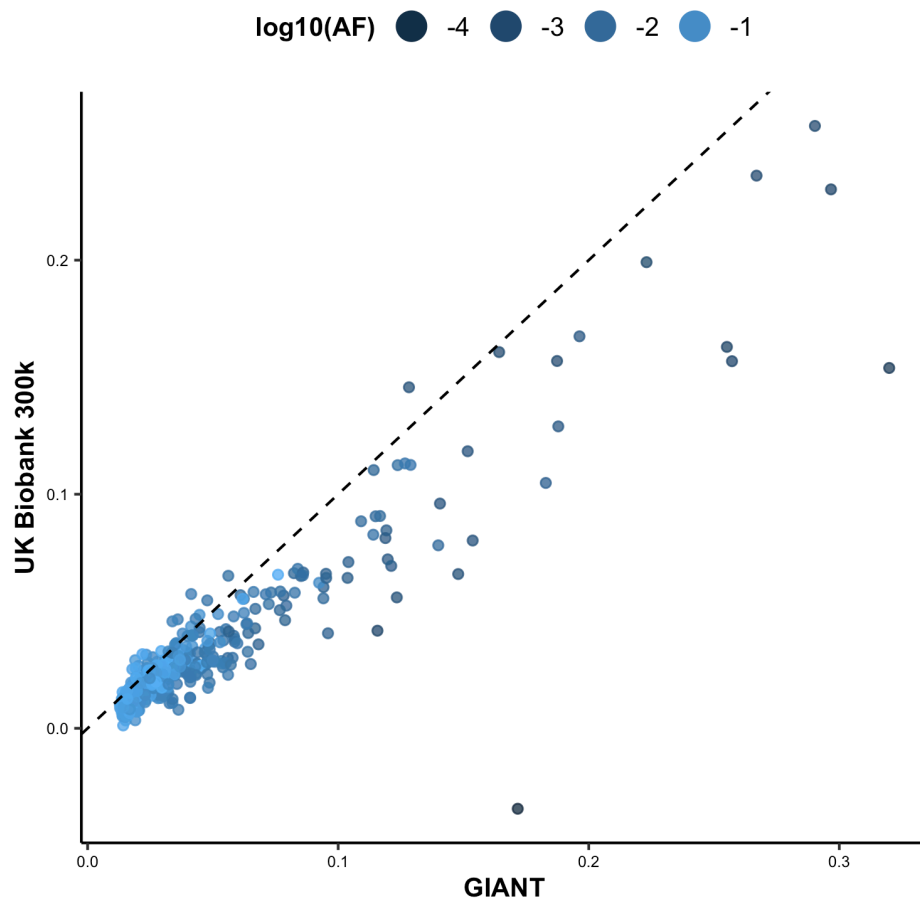


Fig. S17 | Comparison of effect sizes between UK Biobank and GIANT for height. The $y = x$ line is shown for reference.

Finally, we compared associations for 7 red blood cell phenotypes discovered in our results with 20 associations ($p < 5 \times 10^{-9}$) between missense variants and red blood cell phenotypes discovered by TOPMed (8) and find that 18 out of 20 are replicated at $p < 0.05$ with 9 of these associated at $p < 8 \times 10^{-9}$ (Table S6).

Table S6 | Comparison of 20 associations between missense variants and 7 major red blood cell phenotypes discovered at the genome-wide significant loci of the marginal tests in TOPMed; in UK Biobank, 9 of these associations are significant at $p < 8 \times 10^{-9}$ (blue), and 9 are found significant at $p < 0.05$ (light blue).

Phenotype	UKB phenocode	Locus	Allele (Ref)	Allele (Alt)	Gene	Annotation	P-value	
							TOPMed	UK Biobank
hematocrit (HCT)	30030: hematocrit percentage	chr6:26092913	G	A	HFE	missense	6.40E-17	7.70E-132
		chr22:37066896	A	G	TMPRSS6	missense	1.03E-26	7.06E-130
		chrX:154536002	C	T	G6PD	missense	3.36E-22	9.07E-03
hemoglobin (HGB)	30020: hemoglobin concentration	chr6:26092913	G	A	HFE	missense	2.16E-30	3.06E-288
		chr22:37066896	A	G	TMPRSS6	missense	3.16E-51	2.35E-261
		chrX:154536002	C	T	G6PD	missense	1.47E-28	3.23E-03
mean corpuscular hemoglobin (MCH)	30050: Mean corpuscular hemoglobin	chr11:5227003	C	T	HBB	missense	1.24E-23	9.05E-02
		chrX:154536002	C	T	G6PD	missense	2.12E-48	3.20E-02
mean corpuscular hemoglobin concentration (MCHC)	30060: Mean corpuscular hemoglobin concentration	chr6:26092913	G	A	HFE	missense	9.52E-17	3.70E-180
		chr11:5227003	C	T	HBB	missense	4.29E-43	1.32E-01
		chr22:37066896	A	G	TMPRSS6	missense	3.25E-26	9.95E-142
mean corpuscular volume (MCV)	30040: mean corpuscular volume	chr1:247876149	C	T	TRIM58	missense	1.77E-16	1.77E-88
		chr11:5227003	C	T	HBB	missense	1.36E-64	1.46E-02
		chr16:67184472	T	C	EXOC3L1	missense / synonymous	2.13E-09	6.68E-18
		chrX:154536002	C	T	G6PD	missense	3.96E-82	5.54E-03
	30010:	chr11:5227003	C	T	HBB	missense	2.44E-22	8.46E-03

red blood cell count (RBC)	red blood cell (erythrocyte) count	chrX:154536002	C	T	G6PD	missense	3.72E-82	1.74E-04
red blood cell width (RDW)	30070: red blood cell (erythrocyte) distribution width	chr6:26092913	G	A	HFE	missense	5.80E-15	3.90E-320
		chr11:5227003	C	T	HBB	missense	1.59E-10	5.11E-03
		chrX:154536002	C	T	G6PD	missense	8.27E-106	3.23E-03

Analysis of summary statistics

Wenhan Lu, Konrad J. Karczewski, Ellen Tsai, Mark J. Daly, Benjamin M. Neale

We performed all downstream analyses in Hail using the single-variant and group-test MatrixTables as described above.

Gene set analyses

Developmental delay genes

We considered the SKAT-O association results for 470 genes previously implicated in developmental delay (9) and compared the number of associations discovered for these genes with the remaining genes in the dataset. In order to match the background distribution on frequency, we binned genes by their cumulative allele frequency into equal-spaced groups with widths of 0.01, and then matched genes from the remaining set to the distribution of the 470 genes according to their CAF intervals. For each of the three annotation categories (pLoF, missense|LC, and synonymous), we randomly sampled 1,000 matched sets of 470 genes from the remaining set with replacement and computed the mean number of associations and the proportion of genes with at least one association meeting our p-value threshold for each set. By comparing the distribution of the mean and proportion of the 1,000 samples with those of the 470 genes by annotation groups, we found that genes that are implicated in developmental delay are more likely to be associated through a pLoF mechanism with phenotypes in the UK Biobank ($p = 2 \times 10^{-3}$; Fig. 4).

Constrained genes

Similar to the developmental delay, we compared 5,483 (1,848 unique genes) constrained gene-annotation pairs from our dataset with the remaining unconstrained genes on their number of associations discovered from SKAT-O results. We obtained LOEUF values for the genes from gnomAD (v2.1.1) and defined constrained genes as those in the highest decile

of LOEUF (oe_lof_upper_decile=0). We then matched the unconstrained genes to the constrained genes by CAF intervals with widths of 0.01 and randomly sampled 1,000 unconstrained gene-sets that have sample sizes and CAF distributions comparable to the constrained gene-set for each annotation category with replacement. Finally, we compared the mean number of associations and the proportion of genes with at least one association meeting our p-value threshold of the constrained set with the distribution of the 1000 unconstrained samples. We found that constrained genes are more likely to be associated with a phenotype in UK Biobank than the unconstrained genes, for pLoF variants ($p = 2.1 \times 10^{-6}$; Fig. 4).

MAF/CAF Matching

We also compared the number of associations across the three annotation categories for SKAT-O and single-variant tests. We divided CAF of genes and AF of variants into bins with widths of 0.01. For SKAT-O, we matched genes from missense|LC and synonymous categories to the pLoF group by CAF intervals and then sampled a set of missense|LC genes and a set of synonymous genes that have comparable sample sizes and CAF distributions to the 18527 pLoF genes from the dataset with replacement. Similarly, for variants, we matched missense|LC variants and synonymous variants to the 455,445 pLoF variants by their AF Intervals. We compared the mean number of associations and the proportion of genes with at least one association meeting our p-value threshold across the three annotation categories on the matched sets (Fig. 3).

PolyPhen2 predicted variants

We compared the proportion of variants with at least one association meeting our p-value threshold across the three PolyPhen2 prediction groups (probably damaging, possibly damaging, and benign; Fig. S18). We binned variants by their minor allele frequency (MAF) into equal-spaced bins with width of 0.01. Using variants from each one of the three groups as

reference, we matched the remaining two groups to the reference group by their MAF bins. Relative relationships of the proportion among the three groups are similar when using different reference groups. We then split the variants into allele frequency categories and compared the proportion among different PolyPhen2 prediction groups. We conducted a pairwise proportion test between each pair of groups for each allele frequency interval and observed a significant difference between benign and possibly damaging, as well as between benign and probably damaging for both intervals, and possibly damaging and probably damaging for allele frequency between 0.01% and 0.1%. No significant group difference is observed for allele frequencies above 1%.

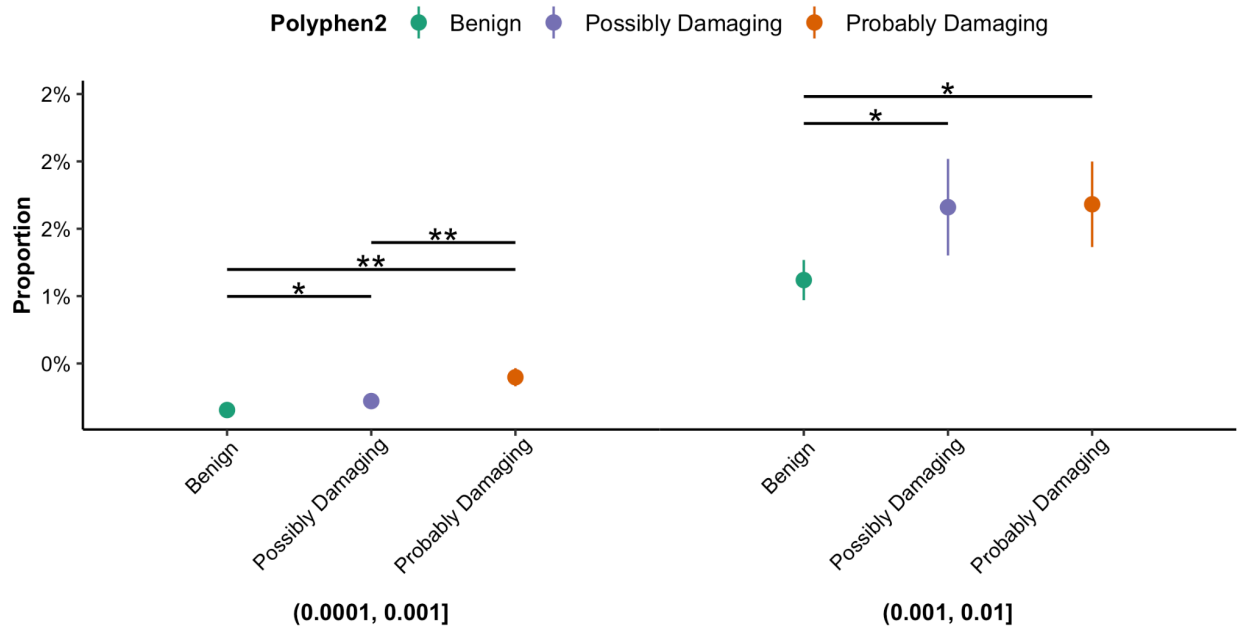


Fig. S18 | The proportion of variants with at least one association is shown broken down by PolyPhen2 annotation group and allele frequency category. * and ** indicate a significant group difference by chi-square test at $p < 0.05$ and $p < 0.001$, respectively. No significant difference is observed for allele frequencies above 1%.

ClinVar variants

We obtained pathogenicity of variants from the ClinVar table in gnomAD reference data and then defined pathogenic and likely pathogenic variants as P/LP, benign and likely benign variants as B/LB. We divided the minor allele frequency (MAF) of variants into equal-spaced bins with widths of 0.01 and then matched variants from B/LB and Uncertain significance group to the 17,487 P/LP variants respectively by their MAF bins. For each of the categories, we randomly sampled 1,000 sets matched to the P/LP variant subset with replacement and computed the mean number of associations and the proportion of variants with at least one association meeting our p-value threshold for each set. By comparing the distributions of the mean and proportion of the 1,000 samples with the P/LP group, we found that pathogenic variants are more likely to be associated with a phenotype in the UK Biobank (Fig. 3).

Data availability and release

Matthew Solomonson, Konrad J. Karczewski, Melissa R. Miller, Ellen Tsai, Nicholas A. Watts, Huy Nguyen, Kevin Nguyen, Cotton Seed, Benjamin M. Neale

Data availability

We make the full dataset available in a browser framework (described below), as well as Hail formats, hosted on the Google Cloud Platform. We provide one MatrixTable for each of the single-variant and gene-based tests, as well as Hail Tables with filtering criteria for variants, genes, and phenotypes.

Code availability

All code to reproduce the analyses herein is available on github:

https://github.com/nealelab/ukb_exomes and https://github.com/nealelab/ukb_common. All

quality control and data analysis tasks were performed using Hail (versions between 0.2.49 and 0.2.62) (10). Unless otherwise indicated, summarized analyses and plotting were performed in R 4.0.2, using tidyverse.

Data browser

Web-based tools like PheWeb have been highly useful in the data processing and dissemination of several recent large-scale biobank genetic studies (11, 12). PheWeb is well-suited for viewing associations from genotyping data along large genomic regions, where the signal is frequently driven by non-coding regulatory variants rather than variation in protein coding sequences. New web-based tools are needed for visualizing association studies in the context of gene-based analyses. Toward this goal, we previously extended on our gnomAD browser toolkit (2) to create a suite of portals for displaying gene analysis results from psychiatric exome association studies for schizophrenia (13), autism (14), bipolar disorder (15), and epilepsy (16). In this study, we extend our exome browser toolkit to support visualization of biobank-scale PheWAS results. We developed new layouts, navigational mechanisms, plots,

and controls that enable users to visualize and compare gene and variant associations across thousands of phenotypes.

Navigation and workflow

The browser interface features a novel split-screen design for rapidly inspecting gene-based PheWAS results (Fig. S19). The left hand side displays the global results index, which displays all hits for a given gene, phenotype, or variant (Fig. S19A). The results index displays PheWAS plots or manhattan plots depending on which navigational button is selected in the top bar (Fig. S19B). The results pane can be condensed, expanded, or hidden entirely by clicking the presets buttons or by dragging the central dotted line left or right. Clicking on one of the arrow buttons in the phenotype table will update the right hand side of the page with a detailed view of the selected gene-phenotype relationship (Fig. S19C). The status bar will update accordingly to reflect the gene, phenotype, variant, or burden dataset that is currently selected (Fig. S19D). Most data is served by a Hail backend, with significant associations cached for speed. Pages load quickly (< 1 second) if the phenotype-gene or phenotype-variant association p-value pair is below the cache threshold (10^{-4} for genes, 10^{-6} for variants) or ~4 seconds if above the cache threshold. Partitioning the page in this way allows users to quickly inspect many associations without losing a sense of context, and either half can be easily hidden to create more screen room for information of interest.

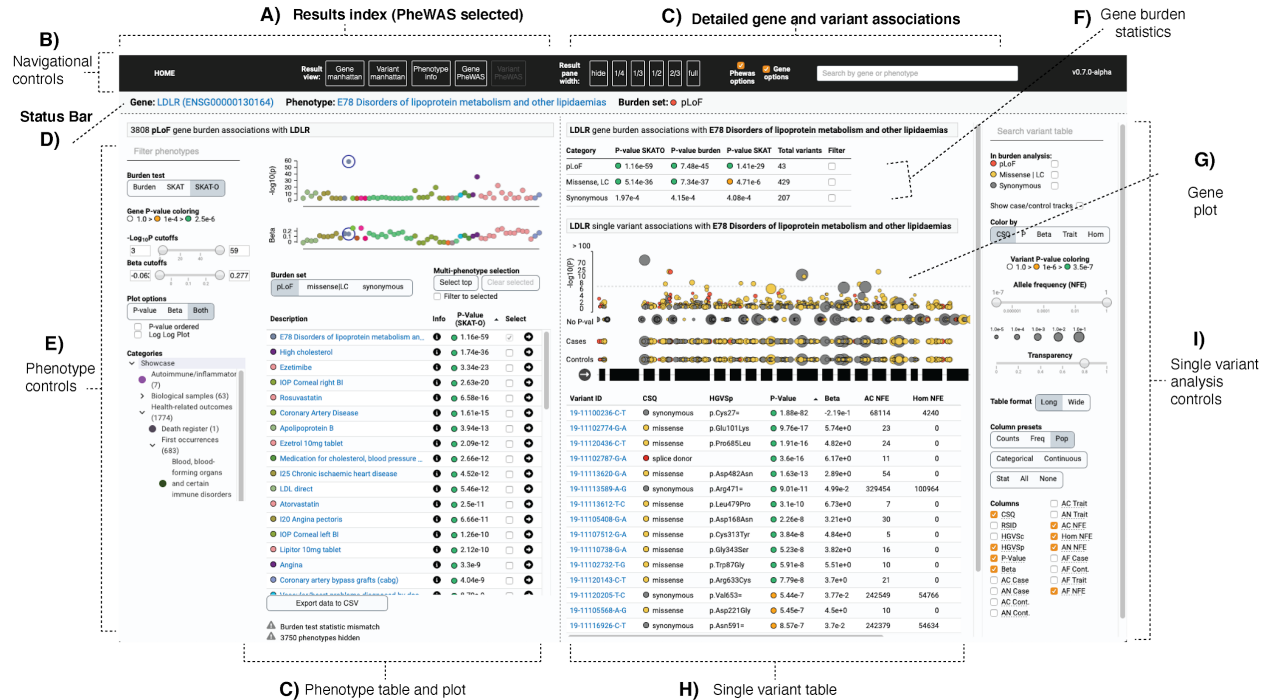


Fig. S19 | Overview of the UKBB exome gene browser interface. The left hand side of the page provides access to all associations with a given gene, variant, or phenotype. The right hand side is for exploring detailed gene test associations (burden, SKAT-O, SKAT) across annotation groups (pLoF, missense and low confidence pLoF, synonymous) in addition to single variants that were included in the burden tests.

Exploring associations by gene

When the “gene PheWAS” results pane is active, the results index displays all phenotypes associated with a particular gene in PheWAS plot and tabular formats (Fig. S19C). The phenotype control panel (Fig. S19E) enables users to specify which of the three burden tests (Burden, SKAT, SKAT-O) or mutational class (pLoF, missense, synonymous) test statistics to display. Phenotypes can be filtered by keywords such as phenotype description or trait type (continuous, categorical, or ICD10). The results can also be filtered by p-value or beta using minimum and/or maximum thresholds. Note that for genes, the beta statistic is always derived from the burden test (SKAT and SKAT-O do not produce beta statistics). The PheWAS plot is colored and grouped by UK biobank showcase category; the category control section can be used to traverse the showcase tree and filter the phenotype list to those belonging to specific categories. The PheWAS plot can be configured to show p-values on either log or double log

scales. Users can expand the plot to focus on p-values only, betas-only, or view p-values and betas simultaneously.

The gene burden statistics table summarizes burden results across all mutational classes and tests (Fig. S19F). The gene plot displays single variants mapped to genomic coordinates along the gene exons. Variant $-\log_{10}p$ values are shown on the Y axis (Fig. S19G). The plot transitions from to a double log scale $\frac{2}{3}$ along the plot height in order to prevent variants with extremely low P-values from dominating the plot, allowing users to focus on novel rare variant associations near the significance threshold. Variants are depicted as circles, with the circle radii log-scaled by allele frequency in the non-Finnish European population. By default, variants are colored by their most severe VEP consequence across transcripts. If the selected phenotype is categorical, two additional case/control variant tracks display variants positions with radii log-scaled to allele frequencies in cases and controls, respectively. If the selected phenotype is continuous, variant radii will be log-scaled by allele frequency among individuals measured for the trait.

The single variant analysis control panel is used to configure data displayed related to single variants (Fig. S19I). Checkboxes enable users to filter variants to those included in the gene burden analysis. Variants are filterable by identifier or annotation using the search box. Users can focus on particular parts of the allele frequency spectrum by dragging the allele frequency filter slider. Detailed summary statistics for all exome variants are available in a table below the plot (Fig. S19H). Users can specify which columns to display using the column selection checkboxes, or they can choose one of the column group presets. Each preset will select a particular set of columns that make sense to compare side-by-side (e.g., allele counts, frequencies, population counts, and columns best suited for categorical or continuous trait types).

Exploring associations by phenotype

When the “gene manhattan” results pane is active, the results index displays all gene associations with a selected phenotype. The results are displayed in manhattan plot, QQ plot, and tabular format (Fig. S20). The three burden test types are displayed as columns, and the burden set (pLoF, missense|LC, or synonymous) can be selected with the “Burden set” segmented control. Clicking on a gene name will navigate to the gene PheWAS view, and clicking on the “details” arrow will update the right hand side without leaving the gene manhattan view. When the “variant manhattan” results pane is active, the left hand side results index takes a similar format as the gene manhattan but displays single variant association p-values instead of gene test statistics. Single variant results can be filtered by consequence category (pLoF, missense, synonymous, and other). Clicking a variant ID will navigate to that variant’s PheWAS view, and clicking the “details” arrow will keep the single variant manhattan view active.

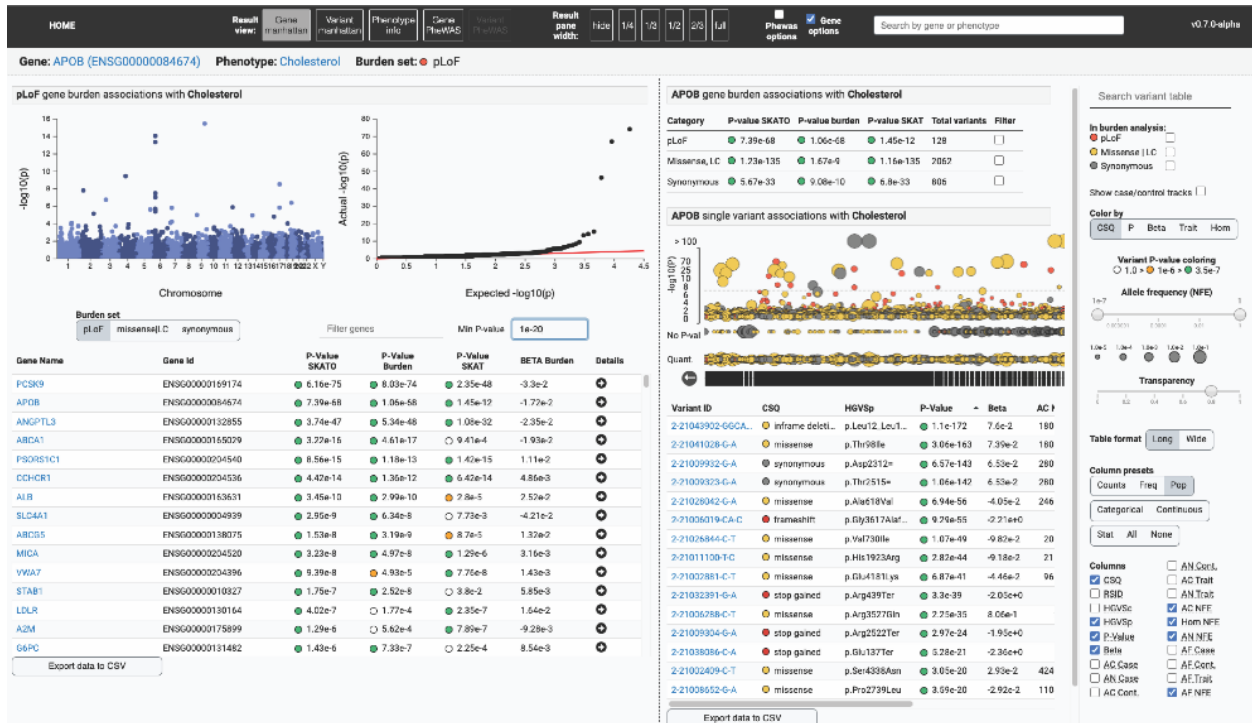


Fig. S20 | Results by phenotype. For a given phenotype, gene or variant association results are displayed in manhattan plot formats in addition to an exportable table. Detailed gene results can be quickly previewed using the arrow button located in each row of the table.

Comparing single variant associations across phenotypes

For a more comprehensive view of all variant-level associations for a gene in a single view, we developed functionality for exploring many phenotypes simultaneously on the gene page (Fig. S21). This feature aims to help users gain insight into pleiotropic patterns of variation across all high-scoring traits for a gene. Each row in the PheWAS table has a checkbox that, when checked, will overlay the phenotype in the gene plot (Fig. S21A). The “select top” button will load all phenotypes below the 10^{-4} p-value threshold; the “clear selected” button will unselect all phenotypes and return to the single phenotype view (Fig. S21B). When selected, phenotypes are assigned randomly generated colors to make them easier to distinguish in the plot and table (Fig. S21C). Many tens or hundreds of phenotypes can be loaded simultaneously; however, an automatic p-value threshold will be applied when there are too many variants to display and the user will be warned in the single variant control panel.

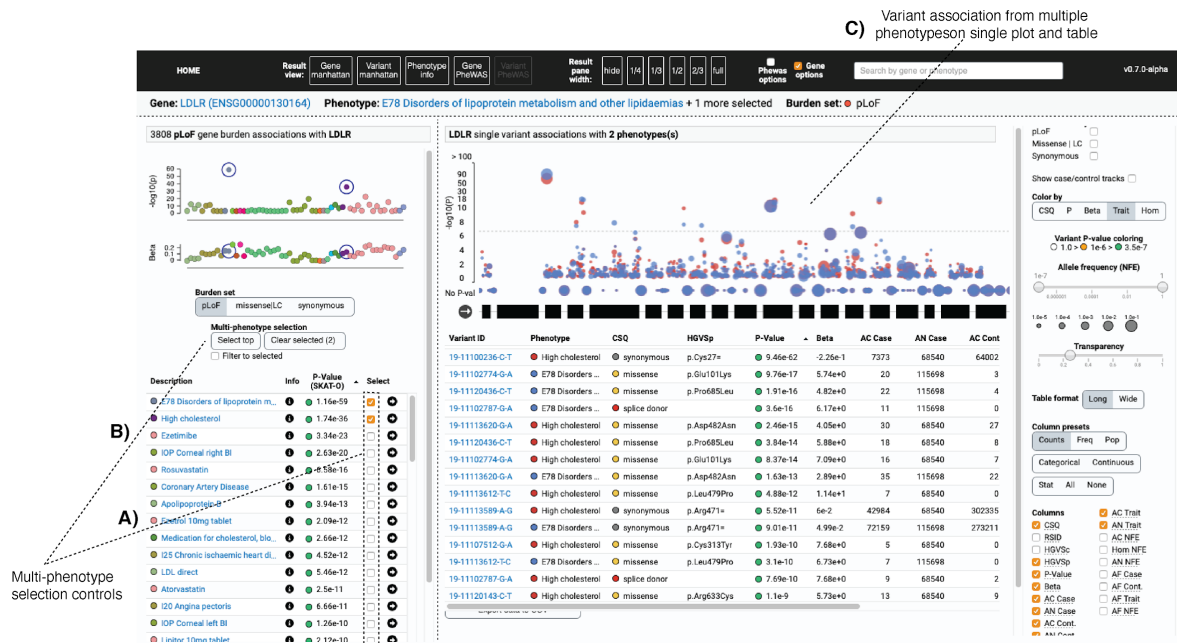
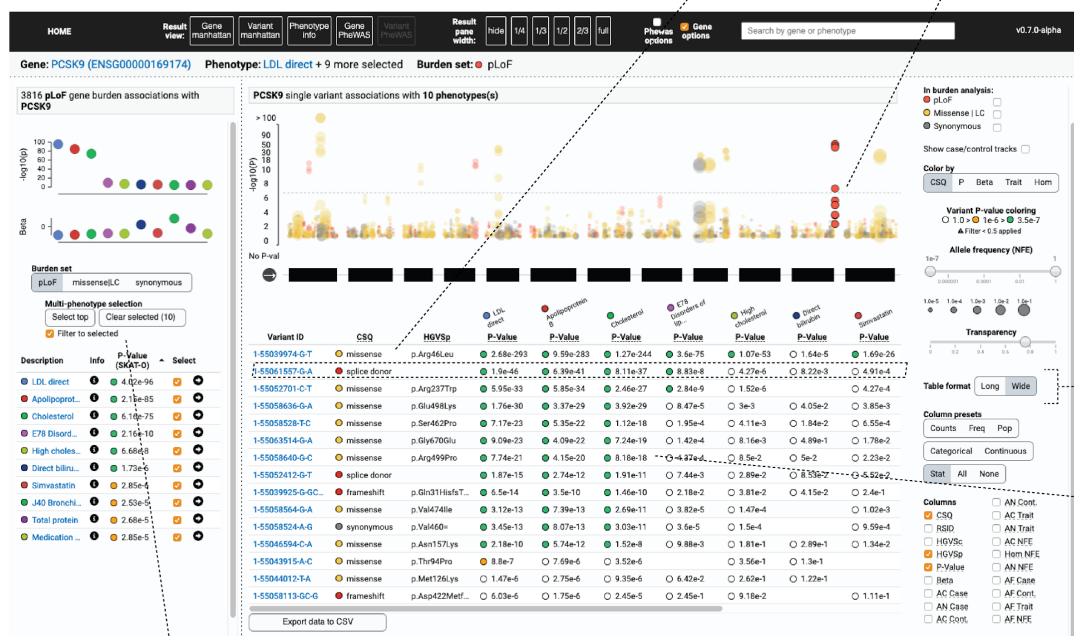


Fig. S21 | Multi-phenotype plotting. Many phenotypes can be selected simultaneously to be overlaid for comparison of single variant analysis associations.

By default, the variant table is configured to the “long” table format; when multiple phenotypes are selected, each variant-phenotype association will appear as a row in the variant table such that the variant table now contains duplicate entries for each variant. To make rows unique and to see comparison of association statistics across traits in a single row, the table can be set to “wide” format (Fig. S22A). The phenotype pivots to the columns, creating a sort of genotype-phenotype matrix (Fig. S22B). The column selection controls will affect both the long and wide table formats. When examining many phenotypes at once, users can click the “filter to selected” button on the phenotype section to simplify the PheWAS plot by only showing the selected phenotypes; this effectively serves as a legend for coloring-by-trait functionality (Fig. S22C).

Hover interactions are especially useful when comparing multiple phenotypes; hovering over variants or phenotypes with the mouse will emphasize the relevant variants and bring them to the foreground (Fig. S22D). The transparency slider sets the opacity level for non-hovered variants, helping the user tune the multi-phenotype plot such that hovered selections can stand out better (Fig. S22E).

D) Hover over variant or phenotype of interest to highlight position in gene and p-value distribution in plot



C) Filter PhenoWAS plot and table to selected phenotypes only using the "Filter to selected" checkbox

E) Set transparency of non-hovered phenotypes

A) Table format switch

B) Genotype/phenotype matrix, displaying P-value as only column

Fig. S22 | Using hover interactions with the multi-phenotype pivot table. Here 10 LDLR associations are compared simultaneously and one splice donor of interest is hovered in the variant table to highlight the plot.

For categorical traits in particular, it is useful to get a visual sense of how case/control variant positions and allele frequencies differ across the gene. The "show case/control tracks" checkbox will fold out case/control tracks for all traits currently selected (Fig. S23A). Continuous traits will be displayed in a single track and the allele frequency for individuals measured for the trait will be displayed. By viewing the case/control counts in the study (Fig. S23B), alongside the case/control allele counts for variants in the variant table (Fig. S23C) and the plots (Fig. S23D) users can very quickly compare burden results across phenotypes, genes, and individual variants to get a sense of which specific variants may be driving gene burden signals. When the per-phenotype tracks are expanded, it can be useful to use the "Color by" switch to look for trends and variants across genomic coordinate, trait, consequences, association p-value, effect size, and zygosity (Fig. S24).



Fig. S23 | Viewing case-control counts and allele frequencies for pLoF variants across traits in a gene.

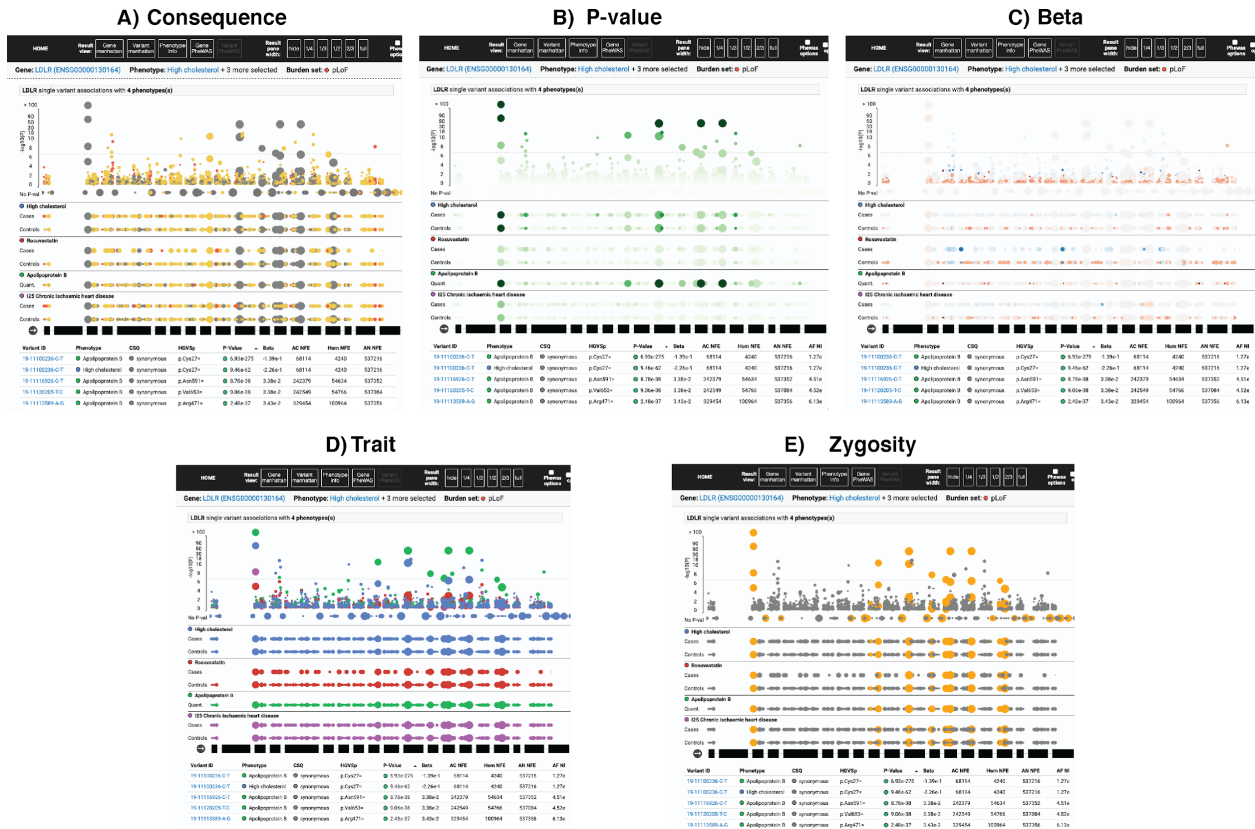


Fig. S24 | Color variants by attribute to uncover patterns in A) consequence, B) p-value, C) beta, D) trait, or E) zygosity.

Single variant results

When a variant is clicked on the single variant manhattan plot or on the gene page, the page will focus on the selected variant, and a PheWAS displays all associations with that variant (Fig. S25). Similar to the gene PheWAS page, multiple phenotypes can be selected and loaded at once. In the variant view, the table rows show statistics for the selected variant, and the columns show values across selected phenotypes. The variant position is displayed along the genomic coordinate. Clicking the “unselect” button will return to the gene page. In this way, users can easily flip back and forth between single variants and the gene context.

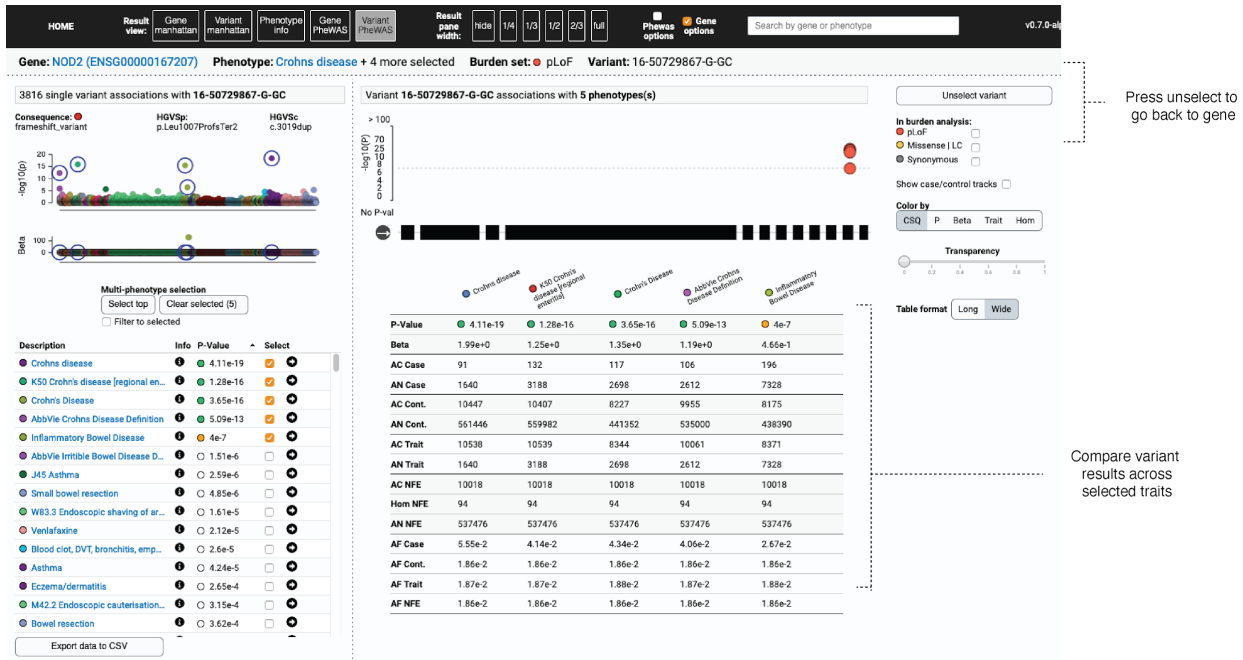


Fig. S25 | Single variant page.

References

1. L. A. C. Millard, N. M. Davies, T. R. Gaunt, G. Davey Smith, K. Tilling, Software Application Profile: PHESANT: a tool for performing automated phenome scans in UK Biobank. *Int. J. Epidemiol.* **47**, 29–35 (2018).
2. K. J. Karczewski, L. C. Francioli, G. Tiao, B. B. Cummings, J. Alföldi, Q. Wang, R. L. Collins, K. M. Laricchia, A. Ganna, D. P. Birnbaum, L. D. Gauthier, H. Brand, M. Solomonson, N. A. Watts, D. Rhodes, M. Singer-Berk, E. M. England, E. G. Seaby, J. A. Kosmicki, R. K. Walters, K. Tashman, Y. Farjoun, E. Banks, T. Poterba, A. Wang, C. Seed, N. Whiffin, J. X. Chong, K. E. Samocha, E. Pierce-Hoffman, Z. Zappala, A. H. O'Donnell-Luria, E. V. Minikel, B. Weisburd, M. Lek, J. S. Ware, C. Vittal, I. M. Armean, L. Bergelson, K. Cibulskis, K. M. Connolly, M. Covarrubias, S. Donnelly, S. Ferreira, S. Gabriel, J. Gentry, N. Gupta, T. Jeandet, D. Kaplan, C. Llanwarne, R. Munshi, S. Novod, N. Petrillo, D. Roazen, V. Ruano-Rubio, A. Saltzman, M. Schleicher, J. Soto, K. Tibbetts, C. Tolonen, G. Wade, M. E. Talkowski, Genome Aggregation Database Consortium, B. M. Neale, M. J. Daly, D. G. MacArthur, The mutational constraint spectrum quantified from variation in 141,456 humans. *Nature.* **581**, 434–443 (2020).
3. C. Bycroft, C. Freeman, D. Petkova, G. Band, L. T. Elliott, K. Sharp, A. Motyer, D. Vukcevic, O. Delaneau, J. O'Connell, A. Cortes, S. Welsh, A. Young, M. Effingham, G. McVean, S. Leslie, N. Allen, P. Donnelly, J. Marchini, The UK Biobank resource with deep phenotyping and genomic data. *Nature.* **562**, 203–209 (2018).
4. R. E. Mills, C. T. Luttig, C. E. Larkins, A. Beauchamp, C. Tsui, W. S. Pittard, S. E. Devine, An initial map of insertion and deletion (INDEL) variation in the human genome. *Genome Res.* **16**, 1182–1190 (2006).
5. W. Zhou, J. B. Nielsen, L. G. Fritsche, J. LeFaive, S. A. Gagliano Taliun, W. Bi, M. E. Gabrielsen, M. J. Daly, B. M. Neale, K. Hveem, G. R. Abecasis, C. J. Willer, S. Lee, Scalable generalized linear mixed model for region-based association tests in large biobanks and cohorts. *bioRxiv* (2019), p. 583278.
6. M. Braun, sparseMVN: an R package for MVN sampling with sparse covariance and precision matrices. *R package version 0. 2. 0*, URL <http://cran.r-project.org/package=sparseMVN> (2015).
7. E. Marouli, M. Graff, C. Medina-Gomez, K. S. Lo, A. R. Wood, T. R. Kjaer, R. S. Fine, Y. Lu, C. Schurmann, H. M. Highland, S. Rieger, G. Thorleifsson, A. E. Justice, D. Lamparter, K. E. Stirrups, V. Turcot, K. L. Young, T. W. Winkler, T. Esko, T. Karaderi, A. E. Locke, N. G. D. Masca, M. C. Y. Ng, P. Mudgal, M. A. Rivas, S. Vedantam, A. Mahajan, X. Guo, G. Abecasis, K. K. Aben, L. S. Adair, D. S. Alam, E. Albrecht, K. H. Allin, M. Allison, P. Amouyel, E. V. Appel, D. Arveiler, F. W. Asselbergs, P. L. Auer, B. Balkau, B. Banas, L. E. Bang, M. Benn, S. Bergmann, L. F. Bielak, M. Blüher, H. Boeing, E. Boerwinkle, C. A. Böger, L. L. Bonnycastle, J. Bork-Jensen, M. L. Bots, E. P. Bottinger, D. W. Bowden, I. Brandslund, G. Breen, M. H. Brilliant, L. Broer, A. A. Burt, A. S. Butterworth, D. J. Carey, M. J. Caulfield, J. C. Chambers, D. I. Chasman, Y.-D. I. Chen, R. Chowdhury, C. Christensen, A. Y. Chu, M. Cocca, F. S. Collins, J. P. Cook, J. Corley, J. C. Galbany, A. J. Cox, G. Cuellar-Partida, J. Danesh, G. Davies, P. I. W. de Bakker, G. J. de Borst, S. de Denus, M. C. H. de Groot, R. de Mutsert, I. J. Deary, G. Dedoussis, E. W. Demerath, A. I. den Hollander, J. G. Dennis, E. Di Angelantonio, F. Drenos, M. Du, A. M. Dunning, D. F.

Easton, T. Ebeling, T. L. Edwards, P. T. Ellinor, P. Elliott, E. Evangelou, A.-E. Farmaki, J. D. Faul, M. F. Feitosa, S. Feng, E. Ferrannini, M. M. Ferrario, J. Ferrieres, J. C. Florez, I. Ford, M. Fornage, P. W. Franks, R. Frikke-Schmidt, T. E. Galesloot, W. Gan, I. Gandin, P. Gasparini, V. Giedraitis, A. Giri, G. Girotto, S. D. Gordon, P. Gordon-Larsen, M. Gorski, N. Grarup, M. L. Grove, V. Gudnason, S. Gustafsson, T. Hansen, K. M. Harris, T. B. Harris, A. T. Hattersley, C. Hayward, L. He, I. M. Heid, K. Heikkilä, Ø. Helgeland, J. Hernesniemi, A. W. Hewitt, L. J. Hocking, M. Hollensted, O. L. Holmen, G. K. Hovingh, J. M. M. Howson, C. B. Hoyng, P. L. Huang, K. Hveem, M. A. Ikram, E. Ingelsson, A. U. Jackson, J.-H. Jansson, G. P. Jarvik, G. B. Jensen, M. A. Jhun, Y. Jia, X. Jiang, S. Johansson, M. E. Jørgensen, T. Jørgensen, P. Jousilahti, J. W. Jukema, B. Kahali, R. S. Kahn, M. Kähönen, P. R. Kamstrup, S. Kanoni, J. Kaprio, M. Karaleftheri, S. L. R. Kardia, F. Karpe, F. Kee, R. Keeman, L. A. Kiemeny, H. Kitajima, K. B. Kluivers, T. Kocher, P. Komulainen, J. Kontto, J. S. Kooner, C. Kooperberg, P. Kovacs, J. Kriebel, H. Kuivaniemi, S. Kury, J. Kuusisto, M. La Bianca, M. Laakso, T. A. Lakka, E. M. Lange, L. A. Lange, C. D. Langefeld, C. Langenberg, E. B. Larson, I.-T. Lee, T. Lehtimäki, C. E. Lewis, H. Li, J. Li, R. Li-Gao, H. Lin, L.-A. Lin, X. Lin, L. Lind, J. Lindström, A. Linneberg, Y. Liu, Y. Liu, A. Lophatananon, J. 'an Luan, S. A. Lubitz, L.-P. Lyytikäinen, D. A. Mackey, P. A. F. Madden, A. K. Manning, S. Männistö, G. Marenne, J. Marten, N. G. Martin, A. L. Mazul, K. Meidtner, A. Metspalu, P. Mitchell, K. L. Mohlke, D. O. Mook-Kanamori, A. Morgan, A. D. Morris, A. P. Morris, M. Müller-Nurasyid, P. B. Munroe, M. A. Nalls, M. Nauck, C. P. Nelson, M. Neville, S. F. Nielsen, K. Nikus, P. R. Njølstad, B. G. Nordestgaard, I. Ntalla, J. R. O'Connell, H. Oksa, L. M. O. Loohuis, R. A. Ophoff, K. R. Owen, C. J. Packard, S. Padmanabhan, C. N. A. Palmer, G. Pasterkamp, A. P. Patel, A. Pattie, O. Pedersen, P. L. Peissig, G. M. Peloso, C. E. Pennell, M. Perola, J. A. Perry, J. R. B. Perry, T. N. Person, A. Pirie, O. Polasek, D. Posthuma, O. T. Raitakari, A. Rasheed, R. Rauramaa, D. F. Reilly, A. P. Reiner, F. Renström, P. M. Ridker, J. D. Rioux, N. Robertson, A. Robino, O. Rolandsson, I. Rudan, K. S. Ruth, D. Saleheen, V. Salomaa, N. J. Samani, K. Sandow, Y. Sapkota, N. Sattar, M. K. Schmidt, P. J. Schreiner, M. B. Schulze, R. A. Scott, M. P. Segura-Lepe, S. Shah, X. Sim, S. Sivapalaratnam, K. S. Small, A. V. Smith, J. A. Smith, L. Southam, T. D. Spector, E. K. Speliotes, J. M. Starr, V. Steinthorsdottir, H. M. Stringham, M. Stumvoll, P. Surendran, L. M. 't Hart, K. E. Tansey, J.-C. Tardif, K. D. Taylor, A. Teumer, D. J. Thompson, U. Thorsteinsdottir, B. H. Thuesen, A. Tönjes, G. Tromp, S. Trompet, E. Tsafantakis, J. Tuomilehto, A. Tybjaerg-Hansen, J. P. Tyrer, R. Uher, A. G. Uitterlinden, S. Ulivi, S. W. van der Laan, A. R. Van Der Leij, C. M. van Duijn, N. M. van Schoor, J. van Setten, A. Varbo, T. V. Varga, R. Varma, D. R. V. Edwards, S. H. Vermeulen, H. Vestergaard, V. Vitart, T. F. Vogt, D. Vozzi, M. Walker, F. Wang, C. A. Wang, S. Wang, Y. Wang, N. J. Wareham, H. R. Warren, J. Wessel, S. M. Willems, J. G. Wilson, D. R. Witte, M. O. Woods, Y. Wu, H. Yaghootkar, J. Yao, P. Yao, L. M. Yerges-Armstrong, R. Young, E. Zeggini, X. Zhan, W. Zhang, J. H. Zhao, W. Zhao, W. Zhao, H. Zheng, W. Zhou, EPIC-InterAct Consortium, CHD Exome+ Consortium, ExomeBP Consortium, T2D-Genes Consortium, GoT2D Genes Consortium, Global Lipids Genetics Consortium, ReproGen Consortium, MAGIC Investigators, J. I. Rotter, M. Boehnke, S. Kathiresan, M. I. McCarthy, C. J. Willer, K. Stefansson, I. B. Borecki, D. J. Liu, K. E. North, N. L. Heard-Costa, T. H. Pers, C. M. Lindgren, C. Oxvig, Z. Kutalik, F. Rivadeneira, R. J. F. Loos, T. M. Frayling, J. N. Hirschhorn, P. Deloukas, G. Lettre, Rare and low-frequency coding variants alter human adult height. *Nature*. **542**, 186–190 (2017).

8. Y. Hu, A. M. Stilp, C. P. McHugh, S. Rao, D. Jain, X. Zheng, J. Lane, S. Méric de Bellefon, L. M. Raffield, M.-H. Chen, L. R. Yanek, M. Wheeler, Y. Yao, C. Ren, J. Broome, J.-Y. Moon, P. S. de Vries, B. D. Hobbs, Q. Sun, P. Surendran, J. A. Brody, T. W. Blackwell, H. Choquet, K. Ryan, R. Duggirala, N. Heard-Costa, Z. Wang, N. Chami, M. H. Preuss, N.

- Min, L. Ekunwe, L. A. Lange, M. Cushman, N. Faraday, J. E. Curran, L. Almasy, K. Kundu, A. V. Smith, S. Gabriel, J. I. Rotter, M. Fornage, D. M. Lloyd-Jones, R. S. Vasan, N. L. Smith, K. E. North, E. Boerwinkle, L. C. Becker, J. P. Lewis, G. R. Abecasis, L. Hou, J. R. O'Connell, A. C. Morrison, T. H. Beaty, R. Kaplan, A. Correa, J. Blangero, E. Jorgenson, B. M. Psaty, C. Kooperberg, R. T. Walton, B. P. Kleinstiver, H. Tang, R. J. F. Loos, N. Soranzo, A. S. Butterworth, D. Nickerson, S. S. Rich, B. D. Mitchell, A. D. Johnson, P. L. Auer, Y. Li, R. A. Mathias, G. Lettre, N. Pankratz, C. C. Laurie, C. A. Laurie, D. E. Bauer, M. P. Conomos, A. P. Reiner, NHLBI Trans-Omics for Precision Medicine (TOPMed) Consortium, Whole-genome sequencing association analysis of quantitative red blood cell phenotypes: The NHLBI TOPMed program. *Am. J. Hum. Genet.* **108**, 874–893 (2021).
9. J. Kaplanis, K. E. Samocha, L. Wiel, Z. Zhang, K. J. Arvai, R. Y. Eberhardt, G. Gallone, S. H. Lelieveld, H. C. Martin, J. F. McRae, P. J. Short, R. I. Torene, E. de Boer, P. Danecek, E. J. Gardner, N. Huang, J. Lord, I. Martincorena, R. Pfundt, M. R. F. Reijnders, A. Yeung, H. G. Yntema, Deciphering Developmental Disorders Study, L. E. L. M. Vissers, J. Juusola, C. F. Wright, H. G. Brunner, H. V. Firth, D. R. FitzPatrick, J. C. Barrett, M. E. Hurles, C. Gilissen, K. Retterer, Evidence for 28 genetic disorders discovered by combining healthcare and research data. *Nature.* **586**, 757–762 (2020).
 10. Hail Team, *Hail 0.2.54*. <https://github.com/hail-is/hail/releases/tag/0.2.54> (2020); <https://github.com/hail-is/hail/releases/tag/0.2.54>).
 11. R. J. Pruim, R. P. Welch, S. Sanna, T. M. Teslovich, P. S. Chines, T. P. Gliedt, M. Boehnke, G. R. Abecasis, C. J. Willer, LocusZoom: regional visualization of genome-wide association scan results. *Bioinformatics.* **26**, 2336–2337 (2010).
 12. S. A. Gagliano Taliun, P. VandeHaar, A. P. Boughton, R. P. Welch, D. Taliun, E. M. Schmidt, W. Zhou, J. B. Nielsen, C. J. Willer, S. Lee, L. G. Fritsche, M. Boehnke, G. R. Abecasis, Exploring and visualizing large-scale genetic associations by using PheWeb. *Nat. Genet.* **52**, 550–552 (2020).
 13. T. Singh, B. M. Neale, M. J. Daly, Exome sequencing identifies rare coding variants in 10 genes which confer substantial risk for schizophrenia. *bioRxiv* (2020), , doi:10.1101/2020.09.18.20192815.
 14. F. K. Satterstrom, J. A. Kosmicki, J. Wang, M. S. Breen, S. De Rubeis, J.-Y. An, M. Peng, R. Collins, J. Grove, L. Klei, C. Stevens, J. Reichert, M. S. Mulhern, M. Artomov, S. Gerges, B. Sheppard, X. Xu, A. Bhaduri, U. Norman, H. Brand, G. Schwartz, R. Nguyen, E. E. Guerrero, C. Dias, Autism Sequencing Consortium, iPSYCH-Broad Consortium, C. Betancur, E. H. Cook, L. Gallagher, M. Gill, J. S. Sutcliffe, A. Thurm, M. E. Zwick, A. D. Børghlum, M. W. State, A. E. Cicek, M. E. Talkowski, D. J. Cutler, B. Devlin, S. J. Sanders, K. Roeder, M. J. Daly, J. D. Buxbaum, Large-Scale Exome Sequencing Study Implicates Both Developmental and Functional Changes in the Neurobiology of Autism. *Cell* (2020), doi:10.1016/j.cell.2019.12.036.
 15. D. S. Palmer, D. P. Howrigan, S. B. Chapman, R. Adolfsson, N. Bass, D. Blackwood, M. P. M. Boks, C.-Y. Chen, C. Churchhouse, A. P. Corvin, N. Craddock, D. Curtis, A. Di Florio, F. Dickerson, F. S. Goes, X. Jia, I. Jones, L. Jones, L. Jonsson, R. S. Kahn, M. Landén, A. Locke, A. McIntosh, A. McQuillin, D. W. Morris, M. C. O'Donovan, R. A. Ophoff, M. J. Owen, N. Pedersen, D. Posthuma, A. Reif, N. Risch, C. Schaefer, L. Scott, T. Singh, J. W. Smoller, M. Solomonson, D. St. Clair, E. A. Stahl, A. Vreeker, J. Walters, W. Wang, N. A. Watts, R. Yolken, P. Zandi, B. M. Neale, Exome sequencing in bipolar disorder reveals

shared risk gene AKAP11 with schizophrenia. *bioRxiv* (2021), ,
doi:10.1101/2021.03.09.21252930.

16. Epi25 Collaborative, Ultra-Rare Genetic Variation in the Epilepsies: A Whole-Exome Sequencing Study of 17,606 Individuals. *Am. J. Hum. Genet.* **105**, 267–282 (2019).