The American Journal of Human Genetics, Volume *105*

Supplemental Data

A Genocentric Approach

to Discovery of Mendelian Disorders

Adam W. Hansen, Mullai Murugan, He Li, Michael M. Khayat, Liwen Wang, Jill Rosenfeld, B. Kim Andrews, Shalini N. Jhangiani, Zeynep H. Coban Akdemir, Fritz J. Sedlazeck, Allison E. Ashley-Koch, Pengfei Liu, Donna M. Muzny, Task Force for Neonatal Genomics, Erica E. Davis, Nicholas Katsanis, Aniko Sabo, Jennifer E. Posey, Yaping Yang, Michael F. Wangler, Christine M. Eng, V. Reid Sutton, James R. Lupski, Eric Boerwinkle, and Richard A. Gibbs

Supplemental Figures

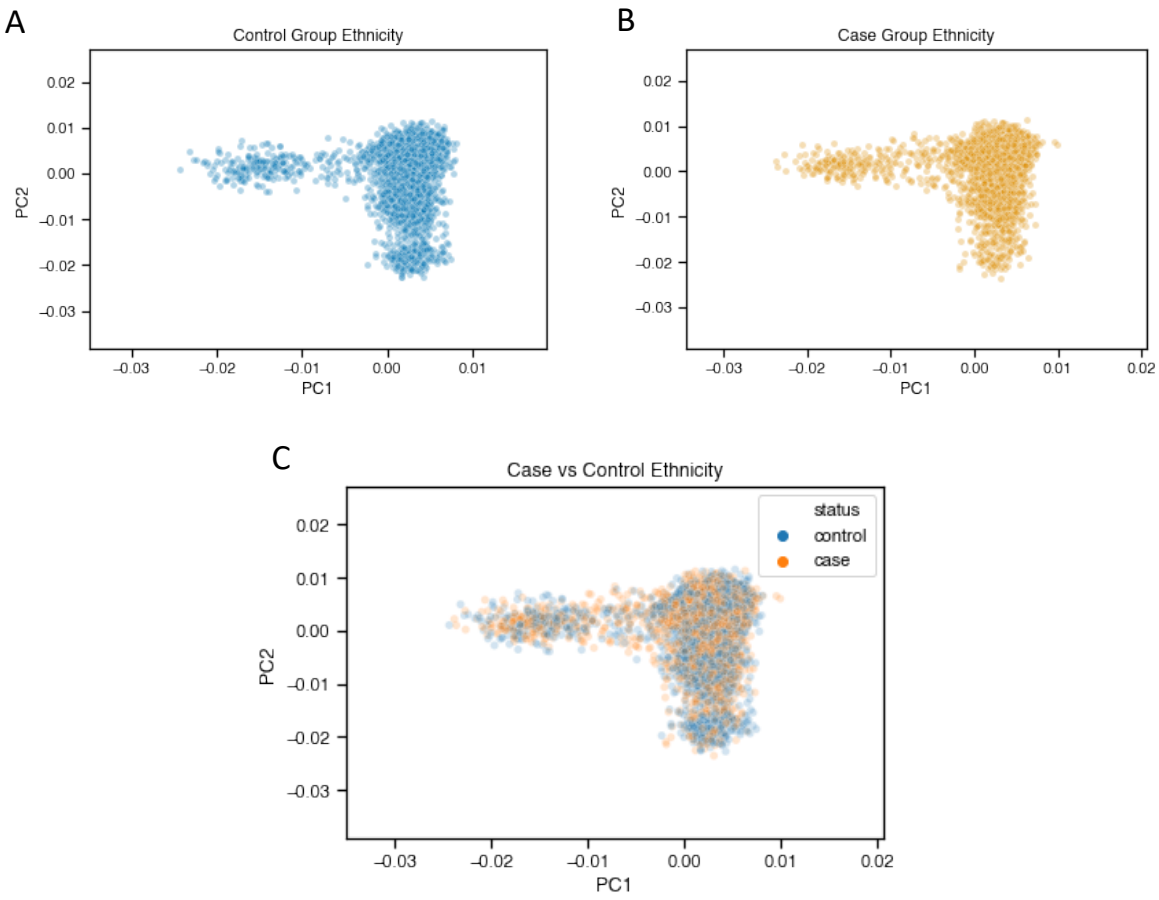


Figure S1 – Case and control ethnicities. Genomic PCA plots showing ethnicity distribution of case vs control samples: A) Control group; B) Random sampling of the case group equal in size to the control group; C) An overlay of plots A and B.

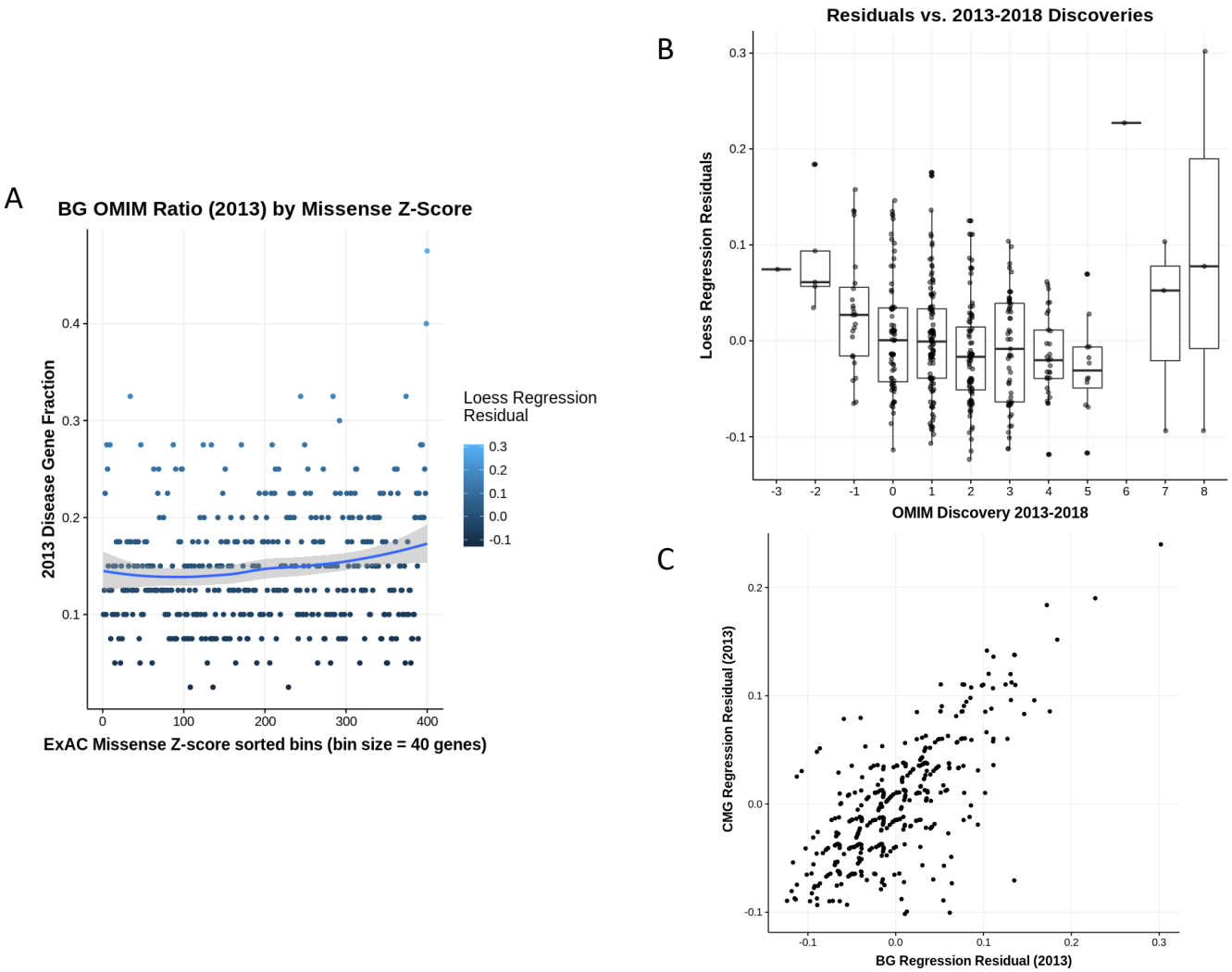


Figure S2 – OMIM ratio outlier regression residual analysis. A) All gene hits are sorted by missense z-score then binned into groups of a consistent, tractable size (40 genes). 2013 OMIM disease gene fraction and a loess regression curve are plotted (95% confidence interval shaded in gray). B) OMIM 2018 vs 2013 disease annotations are compared to quantify disease gene discovery. Gene lists with the higher discovery tend to have the lowest residuals. However, the highest discovery is observed in outliers with the most extreme constraint scores. C) Loess regression residuals of missense z-score bin index vs. 2013 OMIM ratio correlate strongly across independent BG and CMG data sets.

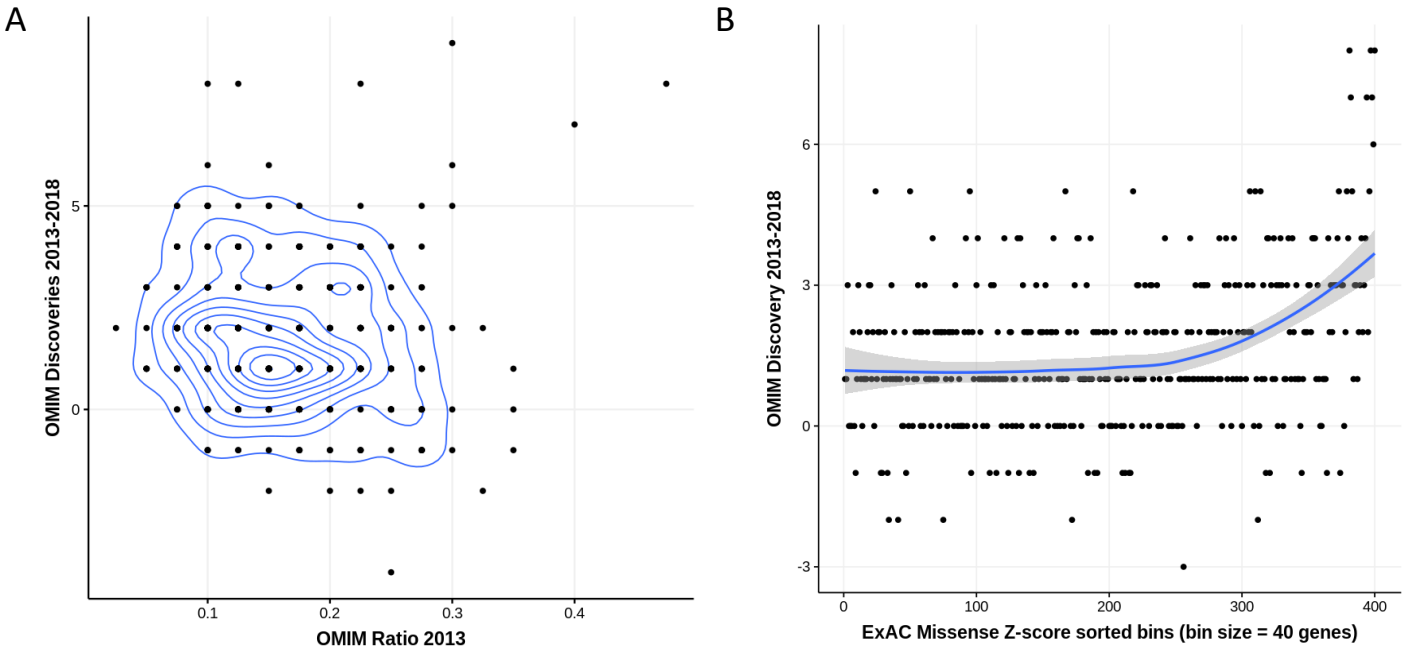


Figure S3 – OMIM 2013-2018 discoveries by other gene set features. In addition to regression residuals (Figure S2), other features of the missense intolerance z-score query series were qualitatively tested for association with discovery through visualization. A) OMIM ratio high outliers (≥ 0.4) had a striking association with discovery. B) Missense intolerance z-score also correlated with discovery, with loess regression plotted in blue, with 95% confidence intervals shaded gray.

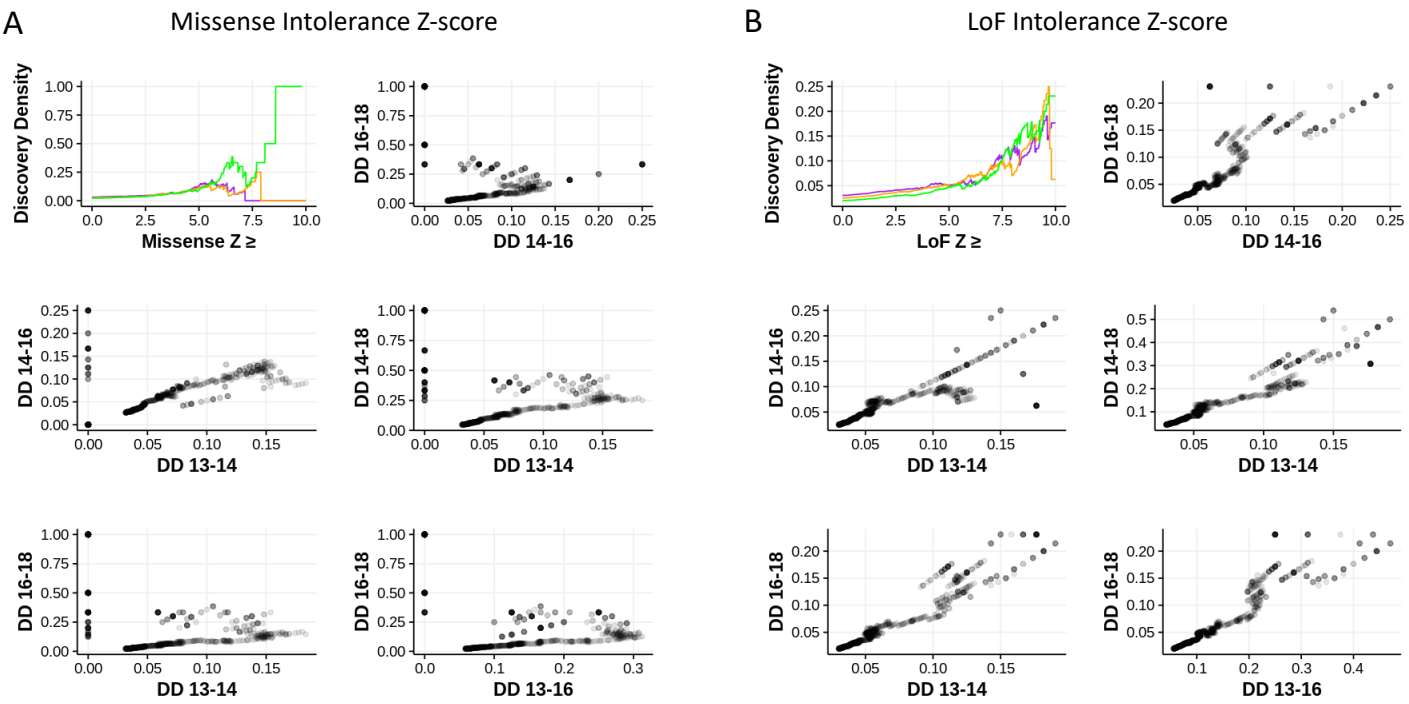


Figure S4 – Past vs. future OMIM discovery density for missense variant parameter sweep query series over time. Past discovery density (DD) consistently correlates with future DD. A) Missense Intolerance Z-score, B) LoF Intolerance Z-score. For the upper-left panels, purple is 2013-2014 discovery density, orange is 2014-2016 discovery density, and green is 2016-2018 discovery density. For the other panels, each point represents a gene query at a fixed z-score cutoff.

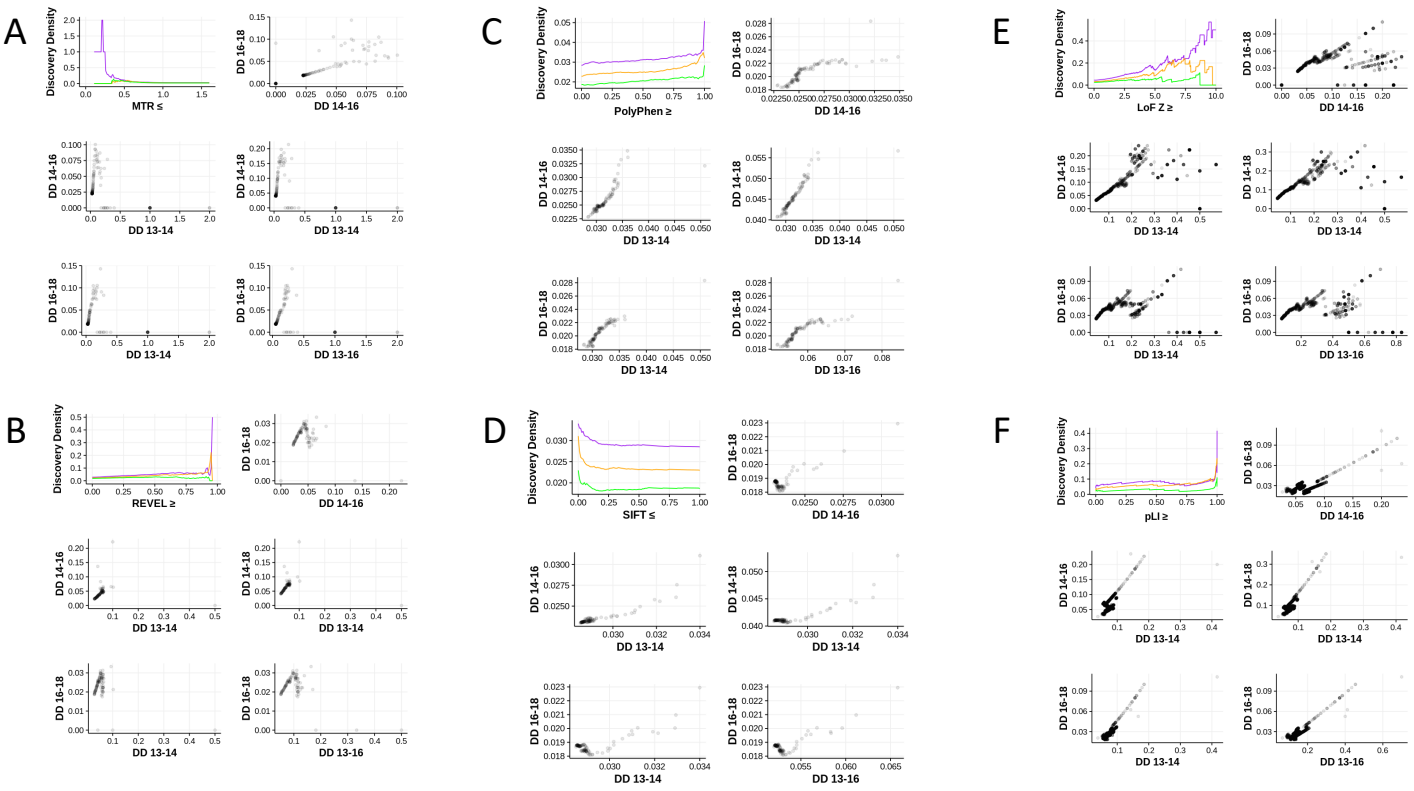
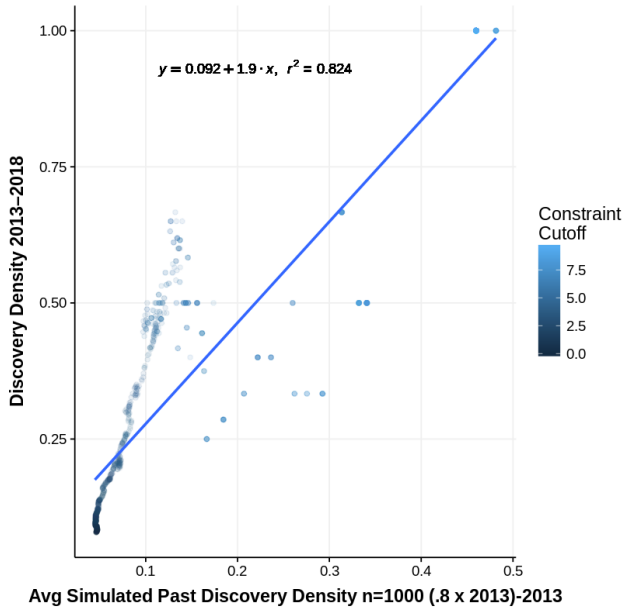


Figure S5 – Past vs. future OMIM discovery density for each annotation parameter sweep query series over time. Past discovery density (DD) correlates with future DD, supporting the strategy of selecting DD-optimizing queries and associated gene lists as candidate disease-associating genes. A-D) Variant-level missense variant parameter sweeps: MTR (A), REVEL (B), PolyPhen (C), and SIFT (D). E-F) LoF variant parameter sweeps: LoF Intolerance Z-score (E) and pLI (F).

A



B

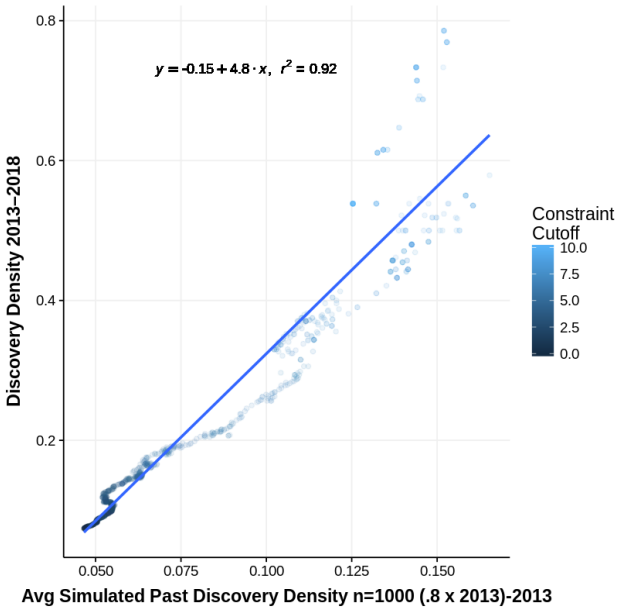


Figure S6 – Simulated (n=1,000) pre-2013 OMIM - 2013 vs. OMIM 2013 - 2018 DD. Simulated past DD correlates with 2013-2018 DD, supporting the strategy of selecting discovery density-optimizing queries and associated gene lists as candidate disease genes. A) Missense Intolerance Z-score, B) LoF Intolerance Z-score (for missense variants).

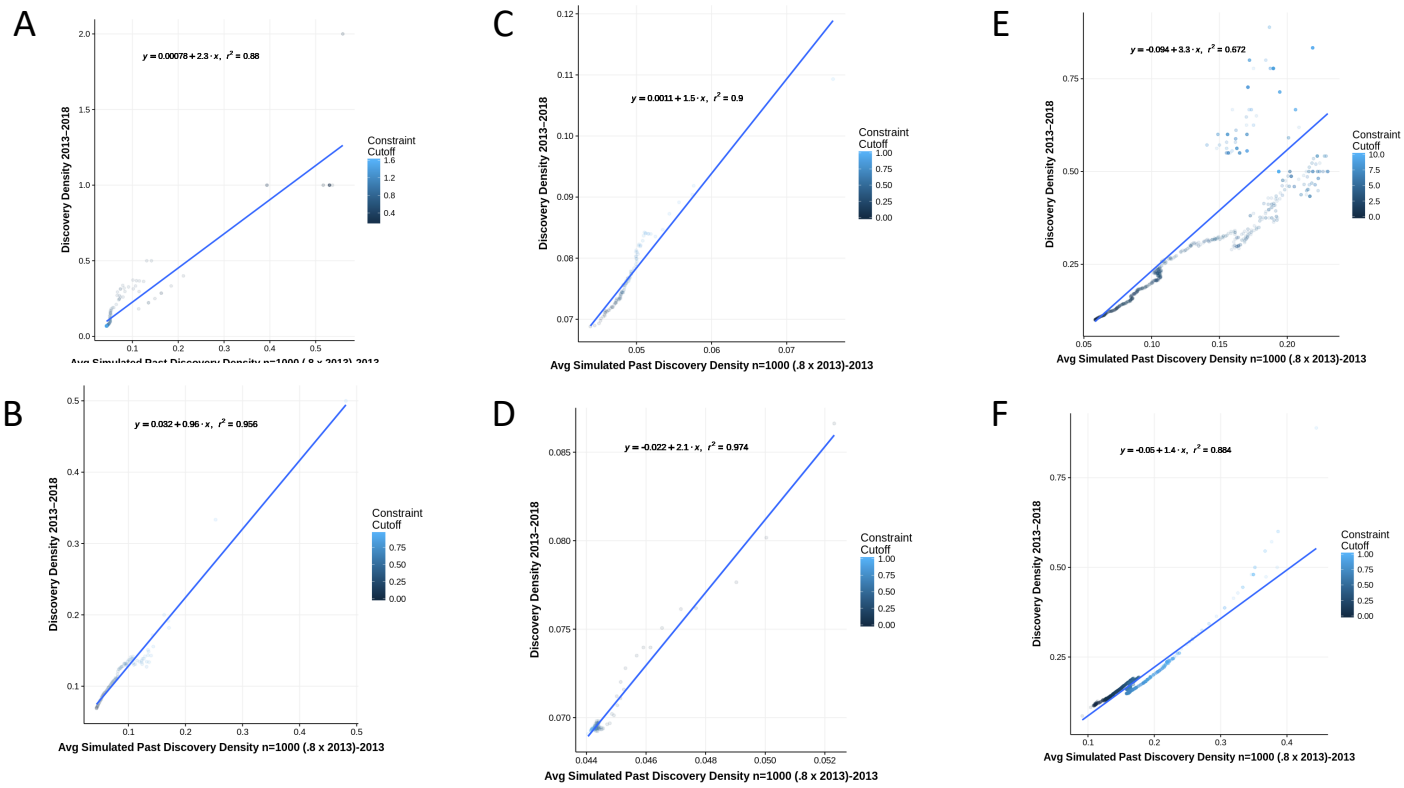


Figure S7 – Simulated (n=1,000) pre-2013 OMIM - 2013 vs. OMIM 2013 - 2018 discovery density. Simulated past DD correlates with 2013-2018 DD, supporting the strategy of selecting discovery density-optimizing queries and associated gene lists as candidate disease genes. A-D) Variant-level missense variant parameter sweeps: MTR (A), REVEL (B), PolyPhen (C), and SIFT (D). E-F) LoF variant parameter sweeps: LoF Intolerance Z-score (E) and pLI (F).

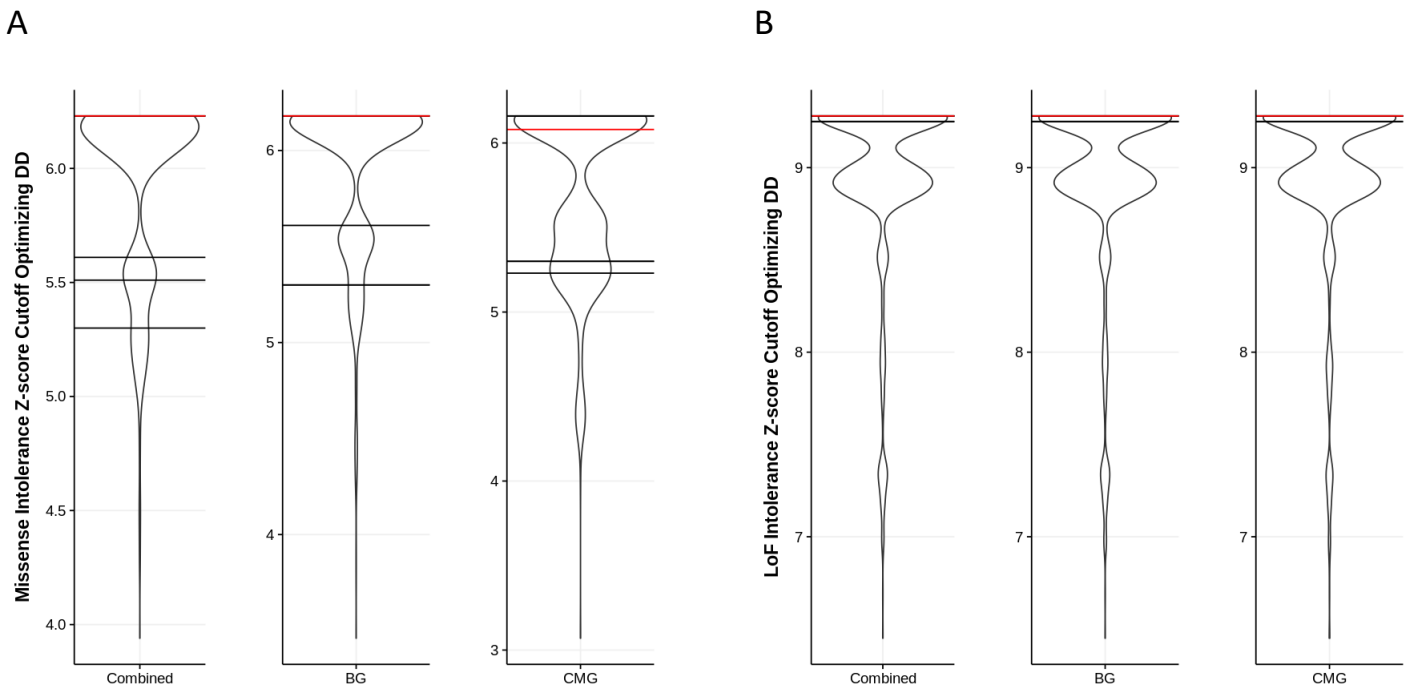


Figure S8 – Discovery-density optimizing filter cutoff variance across OMIM years and input datasets for missense variant parameter sweep query series. A) Missense Intolerance Z-score, B) LoF Intolerance Z-score. Cutoff values for real OMIM data are indicated with horizontal lines. Red horizontal lines indicate 2013-2018 DD-optimizing cutoff values, which for the combined data were used to define candidate novel disease gene lists. For the set of simulated (n=1,000) random downsamplings (.8x) of the 2013 OMIM annotation set (2013_{sim}), the distribution of cutoff values optimizing each 2013_{sim}-2013 DD is indicated by the violin plot width. Real cutoff values optimizing 2013-2018 DD (red) are remarkably stable across independent and aggregate data sets. Real cutoff values for shorter time intervals (black) are less stable across data sets, likely due to noise associated with a smaller change in OMIM annotation volume. For these two query series, many real and simulated DD-optimizing cutoff values are saturated at the minimal constrained candidate disease gene size of 20, evidenced by the appearance of a truncated upper distribution in the simulated data violin plots.

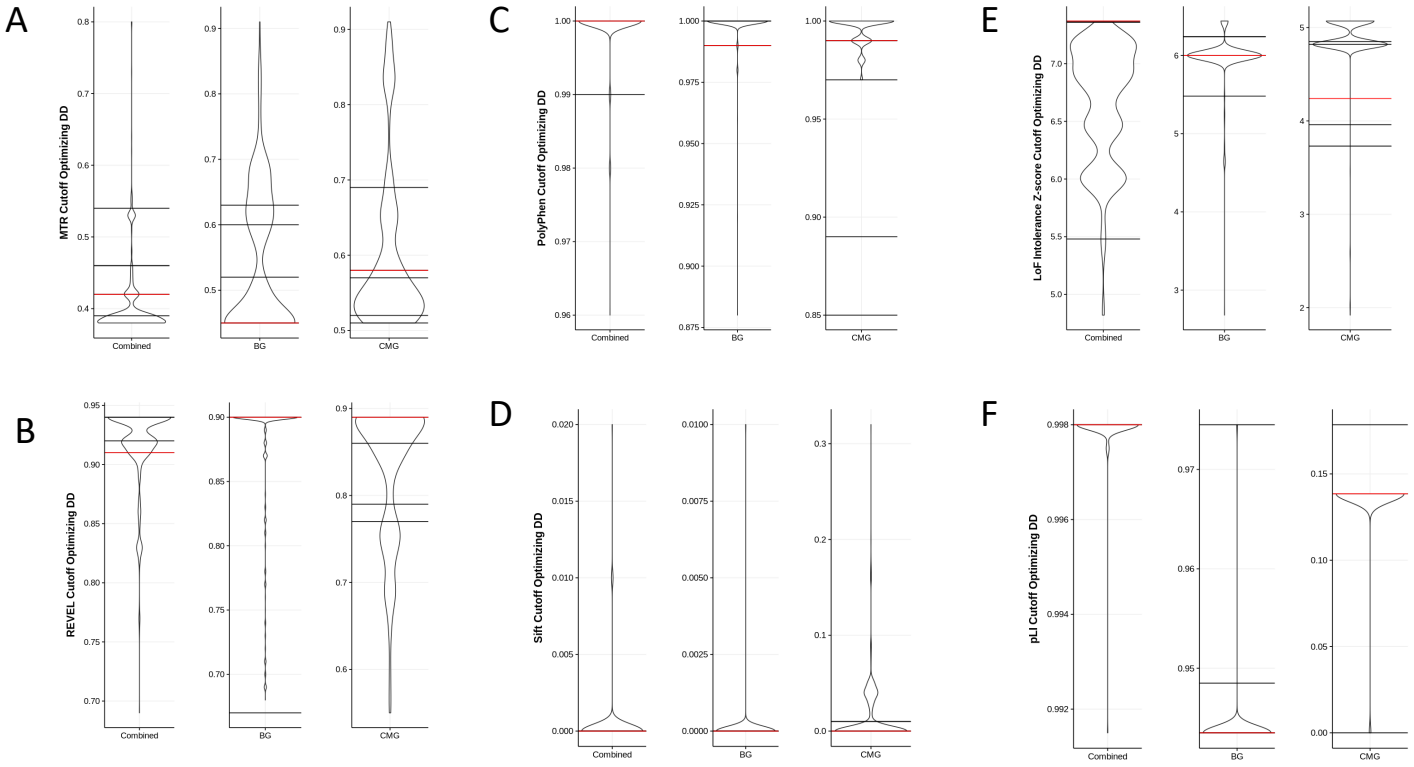


Figure S9 – Discovery-density optimizing filter cutoff variance across OMIM years and input datasets. A-D) Variant-level missense variant parameter sweeps: MTR (A), REVEL (B), PolyPhen (C), and SIFT (D). E-F) LoF variant parameter sweeps: LoF intolerance z-score (E) and pLI (F).

Supplemental Tables

	Avg. Total Time	# of Mappers	# of Reducers	Avg. Map Time	Avg. Shuffle Time	Avg. Merge Time	Avg. Reduce Time
TeraGen	0:17:56	180	NA	0:16:35	NA	NA	NA
TeraSort	0:15:09	7,650	96	0:00:11	0:03:28	0:00:03	0:02:12
TeraValidate	0:01:45	96	1	0:01:30	0:00:02	NA	NA

Table S1 – HARLEE performance benchmarks. Generating, sorting, and validating a terabyte of data with tools TeraGen, TeraSort, and TeraValidate, respectively, *without* encrypting data.

	Avg. Total Time	# of Mappers	# of Reducers	Avg. Map Time	Avg. Shuffle Time	Avg. Merge Time	Avg. Reduce Time
TeraGen	0:22:06	180	NA	0:17:37	NA	NA	NA
TeraSort	0:17:07	7,650	96	0:00:12	0:04:21	0:00:03	0:02:44
TeraValidate	0:01:41	96	1	0:01:15	0:00:02	NA	NA

Table S2 – HARLEE performance benchmarks with data encryption. Generating, sorting, and validating a terabyte of encrypted data with tools TeraGen, TeraSort, and TeraValidate, respectively.

Supplemental Methods

OMIM ratio outlier hypothesis is supported by OMIM ratio regression analysis

When we have finally catalogued all Mendelian disease-gene associations, we can anticipate that associations between OMIM disease enrichment for gene sets filtered against variable bioinformatic filtering parameters across a range of stringencies will reveal generalizable trends (albeit with a certain degree of noise). Until then, at any given point in time, OMIM represents an incomplete set of Mendelian disease-genes associations–an arbitrary subset of the eventual complete set of associations. Thus, as new disease-gene associations are reported to OMIM over time, OMIM annotations for a given parameter value should gradually transition from a stochastic subset into the true complete set of associations. Accordingly, as the curves of plots of OMIM ratio vs. variable bioinformatic filtering parameters gradually take shape, they should reveal an actual relationship between a given parameter and its differential enrichment for proportion of genes associated with Mendelian disorders across a range of filter values applied to a given cohort of affected individuals.

In line with this logic, together with our qualitative observations from the aforementioned preliminary missense intolerance z-score analysis, we hypothesized that, when binning gene lists based on filter parameter value (ie. missense intolerance z-score between 6.5 and 7.0), outlier lists with lower-than-expected disease gene enrichment will be ‘corrected’ as more disease genes are reported. To the extent this hypothesis holds true, we can accordingly prioritize gene lists for discovery efforts as those with lower-than-expected disease gene enrichment.

To test this hypothesis, we sorted all genes harboring recurrent ultra-rare variants across at least five samples in HARLEE by missense intolerance z-score, then grouped them into bins of equal size (40 genes). We then defined an expected degree of enrichment for Mendelian phenotypes by fitting a loess regression curve to the plot of OMIM disease gene fraction by gene-bin missense intolerance z-score rank (Figure S2). We show that the residuals of this plot correlate inversely with disease gene discovery reported in OMIM between 2013-2018. Furthermore, the residuals of loess regression models fitted on corresponding features from independent data sets (BG vs CMG) closely mirror each other, suggesting either robustness of this approach across Mendelian genocentric cohorts or a high degree of similarity of the BG and CMG data sets.

This evidence in support of the OMIM outlier hypothesis–that future discovery can be informed by smoothing a curve of current OMIM ratios and flagging outlier gene sets–supports a generalizable principle that future discovery can be enriched for by identifying gene sets with 'lower-than-expected' disease gene enrichment in OMIM at a given point in time. However, we have not pursued this approach for discovery due to the following rationale: 1) our models lack the degree of refinement and precision that we feel would be ideal for this type of prioritization; 2) regardless, the models tend to predict that genes with extreme filter parameter values are most enriched for discovery; and 3) it follows reason that the most likely of all candidate disease genes should be those with the most extreme constraint–given that we do observe loss-of-function or likely pathogenic missense variants in these genes (Figure S3).

Past discovery density correlates with future discovery density

The strategy of focusing gene discovery efforts on gene sets with a high past discovery density inherently assumes that, within the context of a fixed set of genes resulting from a constant set of query parameters, past discovery density correlates with future discovery density. Thus, in order to evaluate the general correlation between past and future discovery density, discovery density for all five possible nonoverlapping time intervals–based on the OMIM data available–was compared for each parameter sweep query series. Specifically, discovery density over the following combinations was compared: 2013-2014 vs. 2014-16; 2013-2014 vs. 2016-2018; 2013-2014 vs. 2014-2018; 2013-2016 vs. 2016-2018; and 2014-2016 vs 2016-2018. In all instances–except for three of the five REVEL series–a strong positive correlation between past and future discovery density was observed (Figures S4-S5).

We supplemented these data by repeatedly (n=1,000) removing 20% of the 2013 OMIM annotations, effectively 'simulating' a pre-2013 OMIM annotation set. For each query of HARLEE we calculated discovery density for each simulation–as well the mean discovery density across all simulations–as the number of genes with OMIM annotations in 2013, without annotations in a given pre-2013 simulation (Figures S6-S7). For each parameter sweep series, this simulated discovery density was then compared against actual 2013-2018 discovery density, creating a set of discovery density comparisons less prone to the noisy effects of small discovery volume.

For each simulation for a given parameter sweep, we also calculated the discovery density-optimizing filter cutoff value. For each score, the optimal cutoff parameters calculated from the

actual OMIM data points fell within the distribution of the 1,000 simulated optimal cutoff parameters. To test the sensitivity of the optimum discovery density metric to input data, we further calculated these optimum discovery metrics–for both real and simulated OMIM data–separately for independent BG and CMG sample sets. Here we observed more deviation across input data sets than across OMIM annotation period for a fixed input data set (Figures S8-S9).

Significant deviations from the real data correlation trendlines typically occur where discovery density is zero or low for one of the two time periods. We believe this is an artifact of small sample size–or number of discoveries for a given period of time for a given query–as the simulated data contains no instances of a zero-discovery density value. Furthermore, the REVEL simulated data (Figure S5B) does not show the non-linear artifacts seen in three of the real OMIM data REVEL comparisons (Figure S4B upper-right, lower-left, and lower-right panels). Indeed, the average simulated discovery density exhibits a strong positive correlation with actual 2013-2018 discovery density for each of the parameter sweeps tested.

Taken together, we believe these data comparing past vs. future discovery density–with both real and simulated data–validate the assumption that past discovery density correlates with future discovery density. We reasoned that a strategy selecting gene sets with high past discovery density should generally increase the probability that genes in the set without a current reported Mendelian disease association are in fact true, unreported Mendelian disease genes, thus accelerating the overall rate of future gene discovery.

Computer Infrastructure:

Hadoop/Cloudera cluster

HARLEE is a 10-node 280TB Cloudera Hadoop cluster. Each node has a 24-core CPU, 256GB memory and 10x4TB storage drives. Benchmarking and stress testing of the environment was performed with TeraGen, TeraSort, TestDFSIO, NNBench and MRBench. Variant data is stored using two complimentary Hadoop technologies–HBase and Parquet–which provide both rapid sample-level access to the raw variant data and a framework for complex SQL-like querying across the entire data set.

Variant/data ingestion

We have created an extract-transform-load (ETL) process that first ingests raw VCF files into HBase using the HBase Java API. The entire VCF file, including the header, is stored with the body of the VCF being converted to JavaScript Object Notation (JSON) key-value pairs; the INFO field tags are easily encoded in this way as are the sample and format felids. Columns that contain single values such as “filter” and “qual” are given the column name as the key. Unique row keys are created by concatenating a sample unique identifier with the chromosome, start position, end position, ref and alt alleles. Variants are stored in one HBase column and range features such as gVCF blocks, or structural variants are stored in a separate column for convenience when retrieving data. The ingest is secure, pushing VCF data directly into encrypted HBase tables. It is also rapid, a single process easily parsing over 20,000 variants per second, allowing us to store a typical clinical exome in 20-30 seconds; a whole genome gVCF with over 25M lines will be ingested in around 15 minutes. The ingested data is triplicated

within HDFS for redundancy but also compressed using SNAPPY compression–the Hbase footprint is therefore roughly equivalent to the uncompressed VCF. A whole genome gVCF takes up approximately 5Gb per sample, thus giving a theoretical maximum capacity of 56,000 whole genomes with current hardware.

Once the VCF data is encoded and encrypted within HBase, a map-reduce process is used to import the variant data into flat Parquet tables. A simple VCF-like schema is employed, using Apache HIVE user-defined functions (UDFs) to parse JSON data. Non-standard key-value pairs such as INFO field data remain JSON encoded, allowing us to store VCFs from multiple variant callers within the same schema.

Variant annotation

One advantage of the HARLEE is the ability to annotate 'on demand', or as data is queried, by joining annotations to variant data based on a common variant ID. This high degree of modularity is contrasted with a typical bioinformatic variant analysis pipeline, where specific annotation features are more or less hard-coded into an at-scale analysis effort. Modular annotation is ideal for a research environment, where questions to be asked or annotation features to be utilized are not known in advance. In order to streamline this modular annotation, we are creating an "annotation database" within the HGSC Data Lake. We are depositing both gene- and variant-level annotations into this database, including annotations from the following sources: ClinVar, dbSNP, ExAC, gnomAD, MTR (SOURCE), REVEL (SOURCE), 1,000 genomes project, dbNSFP, and other useful annotations downloaded from BioMart and the UCSC Genome Browser.

One difficulty with variant annotation is that many annotation features–such as SIFT and PolyPhen–are transcript-specific. Furthermore, there is no well-established or standardized approach for systematically mapping variants of interest to a single transcript. To resolve this issue for the time being, we have chosen to annotate all distinct HGSC variants with Ensembl Variant Effect Predictor (VEP), storing the output as the "variant" table within the annotation database. As part of its annotation script, VEP has the option of flagging a transcript of interest per variant per gene. Because no well-justified solution to transcript selection been published and extensively adopted to our knowledge, we have chosen to utilize this VEP option, with transcripts flagged by the default multi-tiered VEP logic: 1) canonical status of transcript; 2) APPRIS isoform annotation; 3) transcript support level; 4) biotype of transcript (protein_coding preferred); 5) CCDS status of transcript; 6) consequence rank–according to a table published on the VEP website; 7) translated, transcript or feature length (longer preferred). To maintain the flexibility of this resource, for a given variant we store all possible transcript annotations, flagging selected transcripts as opposed to removing all unselected transcripts. Furthermore, because we are dependent on VEP for transcript flagging, we have decided to obtain several additional annotations from VEP for convenience.

VEP command:

```
perl [VEP_path]/vep -i "$infile" -o "$outfile"
--dir_cache [VEP_path]/cache_files
--dir_plugins [VEP_path]/Plugins
--fork 8
--buffer_size 1000
--merged
--cache
--offline
--force_overwrite
```

```
--stats_text
--json
--assembly GRCh37
--everything
--total_length
--nearest gene
--hgvsg
--check_existing
--flag_pick_allele_gene
--fasta [VEP_path]/fasta_files/
1>process_1.vepRS.log
2>process_1.vepRS.err
```

Thus, following the parquet ingestion, the genotype data is joined with the annotated variant table–which is also mirrored in HBase and Parquet. Any novel variants are first annotated using VEP as described above, then all variants are subsequently queried against other useful annotations from our annotations database. The variant data is organized in this way so as to make obtaining all the relevant information about a given variant across multiple annotation sources a very fast and simple lookup, typically on the order of milliseconds.

Data access

Access to clinical data is strictly controlled. Data is encrypted at rest and in motion and tiered access is provided via Sentry and Kerberos authentication and authorization. Analysts are given access to subsets of the data based on requirements for specific projects. Within their sphere of data access, multiple options are available for analysts to interface with the data. Users with command line computing experience may prefer to utilize the Hadoop File System (HDFS) command line interface, where Apache Hive or Apache Impala shells may be launched. Alternately, the Hadoop User Experience (HUE) provides a user-friendly web app enabling visual

browsing of databases and tables, querying with Hive and Impala, workflow management, and a job browser detailing status of current and past jobs. However, while HUE is a more user-friendly interface than the command line, data querying in HUE still requires experience with SQL-like syntax. Thus, to facilitate data mining for users with limited or no SQL experience, further access is provided through Pentaho web portals which facilitate query building and data visualization without requiring any SQL, command line, or programmatic interface. Furthermore, Pentaho enables a framework for clinical reporting functions and complex analytics/visualization. Finally, data may also be accessed through the Java Database Connectivity (JDBC) API, which in turn allows for direct querying of the data from scripting languages such as R.

Security and compliance

We have implemented a multi-faceted and multi-layered security system to ensure the security and privacy of the data on Hadoop and to provide a compliance ready environment to comply with FISMA, HIPAA, Texas Medical Records Privacy Act and other industry regulations. The five pillars of our security implementation are 1) authentication, 2) authorization, 3) auditing, 4) data protection and 5) perimeter security.

Authentication is implemented by verifying the identity of the entity (user or service) trying to access the data with a strong Kerberos-enabled mechanism, specifically Microsoft Active Directory (AD) with Kerberos authentication. User and service principals are created and authenticated in Active Directory with passwords and keytab files, respectively, before they can

interact with the Hadoop cluster. With Kerberos enablement, users must first authenticate themselves to the Active Directory Kerberos Key Distribution Center (KDC) to obtain a valid Ticket-Granting-Ticket (TGT). The TGT is then used by Hadoop services to verify the user's identity. With Kerberos, a user is not only authenticated on the system they are logged into, but they are also authenticated to the network. Any subsequent interactions with other services that have been configured to allow Kerberos authentication for user access are also secured. (All Hadoop projects we are utilizing i.e. HDFS, MapReduce, HBase, Hive, HUE, Impala, Sentry etc. are all Kerberos enabled.) With this level of Kerberos enabled authentication, we have ensured that only legitimate AD users can authenticate to the system and have virtually eliminated any threat of user impersonation.

The next pillar is authorization. With authorization, we define the access or control an entity has over a given resource. To eliminate the overhead associated with managing access at the user level, we have created groups in Microsoft Active Directory and have assigned users to these groups. These groups are then mapped to roles in Apache Sentry. Sentry implements role-based access control (RBAC) with its roles being mapped to permissions. We have implemented granular permissions at the file-, directory-, database- and table-level in Apache Sentry and have mapped these permissions to the roles we have created. With this level of permission granularity, we can control and manage access to user groups efficiently thereby ensuring while public data is accessible to all groups, sensitive research/clinical data is accessible only to users/groups with access, and PHI data is accessible only to an authorized clinical group with HIPAA training. All the Hadoop projects other than HBase grant access to the users via the groups they are assigned to in Active Directory using Sentry. Sentry plugins are

added to the Hadoop projects during installation (supplementary figure 3); as the entity tries to access a given resource–e.g. Hive–the resource accesses the Sentry service via the plugin and verifies access. HBase utilizes independent Access Control Lists (ACL) for managing access; though access can be configured at the global (all databases), namespace, table or cell level, we have started with granting access at the namespace level to our user groups with the intention of implementing table, column and even cell level security as we start including sensitive data in HBase and need to partition user groups by table or columns. The combination of Cloudera and Hadoop technologies will allow us to manage access as granularly as possible thereby enabling us to house public, private, semi-private and sensitive data together while still providing access to multiple users and groups, both internal and external. We believe this will be greatly beneficial to both researchers and clinicians who hitherto have been only able to access portions of the data with difficulty through different means.

Auditing is a pillar that is critical for managing the compliance and data governance requirements of the Hadoop cluster. Without auditing, all of the other security pillars have limited effect because of a lack of visibility. We have implemented auditing for the Hadoop cluster using Cloudera Navigator and Cloudera Manager. With auditing, we are able to easily keep track of who is doing what on the cluster and when; this includes both positive events–actions that are successful and allowed–and negative events–actions that are unsuccessful and not allowed. In addition to centralized auditing–with Cloudera Navigator and Manager–we are also able to peruse detailed audit reports, giving us a quick and easy overview on who did what and when on the cluster. We are able to view the data lineage, which is helpful in identifying multiple key data attributes: the origin of the data; whether the data can be trusted for the

required analysis; and whether the data is being used by other users. With metadata tagging and indexing we are able to locate and track data easily and subsequently analyze user activity. Ultimately, with the auditing capabilities we have implemented on our Hadoop cluster, we are able to comply relatively easily with compliance and governance rules.

Encryption is a pillar, which generally speaking relates to data protection. With data protection–particularly encryption–becoming mandatory to comply with federal and state regulations, we have implemented both over-the-wire encryption and at-rest encryption. With over-the-wire encryption, we protect data while it is in transit over network channels and with at-rest encryption, we protect data when it is persisted to disk. The data in the Hadoop cluster, stored in HDFS, is protected end-to-end both during transfer and at rest via transparent data encryption (TDE). The performance overhead associated with encrypting/decrypting data is about 10-14%. We decided to encrypt all our data in HDFS as the need to maintain the security and privacy of the data overshadows the performance overhead. We implemented this by establishing encryption zones in HDFS and storing all our data in these zones. The directories and files in these encryption zones will be transparently encrypted upon write and transparently decrypted upon read (TDE). The encryption zone is associated with a key and each file/directory also has its own encrypted key and is managed using the Key Management Service (KMS) and a Key Store. The default KMS implementation combines the KMS and key store functions into a single service. As this implementation should not be used in a production environment with sensitive data, our KMS implementation uses the Cloudera Navigator Key Trustee as the key store. This separates the KMS and key store roles and allows them to be separated on different servers, which in turn provides better key protection. Additionally, we

have also utilized Cloudera Navigator to encrypt areas outside HDFS–i.e. log directories and database storage directories–to protect sensitive data in these locations.

The final security pillar is perimeter security. With perimeter security we provide guarded access to the Hadoop environment. A Cisco ASA firewall with an intrusion detection system tightly controls access to the HGSC network from outside the network. Additionally, the Hadoop cluster and associated ecosystem is designed to run on distinct network VLAN (Virtual LAN) that segregates the Hadoop cluster network traffic from other unencrypted network traffic. The Hadoop cluster and ecosystem reside on an independent, secure demilitarized zone with regulated external access. The security and compliance readiness of the Hadoop environment is ensured through a perimeter fence that houses the servers, network switches and infrastructure rack in a physically secure data center.

Supplemental Note

Task Force for Neonatal Genomics Consortium

Alexander Allori[2], Misha Angrist[3], Patricia Ashley[4], Margarita Bidegain[4], Brita Boyd[5], Eileen Chambers[6], Heidi Cope[1,7], C. Michael Cotten[4], Theresa Curington[1], Erica E. Davis[1], Sarah Ellestad[5], Kimberley Fisher[8], Amanda French[9], William Gallentine[10,11], Ronald Goldberg[4], Kevin Hill[12], Sujay Kansagra[10], Nicholas Katsanis[1], Sara Katsanis[3], Joanne Kurtzberg[13], Jeffrey Marcus[2], Marie McDonald[14], Mohammed Mikati[10], Stephen Miller[12], Amy Murtha[5], Yezmin Perilla[1], Carolyn Pizoli[10], Todd Purves[15], Sherry Ross[15,16], Azita Sadeghpour[1], Edward Smith[10], John Wiener[15]

[1]Center for Human Disease Modeling, Duke University Medical Center, Durham, NC USA
[2]Department of Surgery, Division of Plastic Maxillofacial and Oral Surgery, Duke University Medical Center, Durham, NC USA
[3]Science and Society, Duke University School of Medicine, Durham, NC USA
[4]Department of Pediatrics, Division of Neonatology, Duke University Medical Center, Durham, NC USA
[5]Department of Obstetrics and Gynecology, Division of Maternal-Fetal Medicine, Duke University Medical Center, Durham, NC USA
[6]Department of Pediatrics, Division of Pediatric Nephrology, Duke University Medical Center, Durham, NC USA
[7]Department of Medicine, Duke University Medical Center, Durham, NC, USA
[8]Neonatal Perinatal Research Unit, Duke University Medical Center, Durham, NC USA
[9]Fetal Diagnostic Center, Duke University Medical Center, Durham, NC USA
[10]Department of Pediatrics, Division of Pediatric Neurology, Duke University Medical Center, Durham, NC USA
[11]Present address: Department of Neurology, Division of Pediatric Neurology, Stanford University Lucile Packard Children's Hospital, Palo Alto, CA
[12]Department of Pediatrics, Division of Pediatric Cardiology, Duke University Medical Center, Durham, NC USA
[13]Department of Pediatrics, Division of Pediatric Blood and Marrow Transplantation, Duke University Medical Center, Durham, NC USA
[14]Department of Pediatrics, Division of Medical Genetics, Duke University Medical Center, Durham, NC USA
[15]Department of Surgery, Division of Pediatric Urology, Duke University Medical Center, Durham, NC USA
[16]Present address: Department of Urology, University of North Carolina, Chapel Hill, NC USA