MSSNG - Researcher README

The MSSNG project makes data available to trusted researchers with the goal of improving our understanding of Autism Spectrum Disorder (ASD). An associated publication can be found at https://www.ncbi.nlm.nih.gov/pubmed/28263302

The purpose of this document is to provide an overview of the available data, associated tools, and basic examples of using the tools to access the data.

Notable updates to the data or portal can always be found in the CHANGELOG.

Table of Contents

- Data
  - Overview
  - Types of data
    - Sample/subject Data
      - subject
      - measures
      - subject_sample
    - Aligned Reads
    - Called Variants
    - Annotations
    - De-novo Variants
    - Sanger-validated Variants
    - Copy Number Variants (CNVs)
  - Repositories
- Access
- Tools
- Examples
  - Subject/sample data
    - BigQuery web interface
      - Setup
      - Subject/Sample Data Example
      - More Subject/Sample data Query Examples
      - Genomic Variants Examples
    - R interface to BigQuery
      - Setup
      - Query phenotypes

Data

Overview

Data is available for 7187 individuals (7231 genome samples[1]), including:

- 3425 affected individuals (2691 males, 734 females)
- 3762 unaffected individuals (1882 males, 1880 females)

Individuals typically belong to family trios (2 parents, 1 affected child) or quads (2 parents, 2 affected children). A few different family structures are also present. A total of 2756 families are available.

Family members	Families	Individuals
1	815	815
2	111	222
3	1262	3786
4	488	1952
5	70	350
6	8	48
7	2	14

This provides, in summary:

Genome Samples	Individuals	Affected	Unaffected	Sequencing Technology
1652	1646	699	947	Complete Genomics
4987	4985	2544	2441	Illumina HiSeqX
582	582	194	388	Illumina HiSeq
10	10	3	7	Illumina HiSeq2500

A summary of the DNA source of the samples is as follows:

DNA Source	Genome Samples
Blood	6543
Cell line	359
Saliva	1
White blood cell	328

[1] A few individuals were sequenced more than once.

Types of data

The following types of data for these individuals is available:

- Sample/Subject Data
- Aligned Reads
- Called Variants

Sample/subject Data

Sample/subject data are broken out into three separate tables: `subject`, `measures`, and `subject_sample`. These tables are available as BigQuery tables (`idyllic-analyst-574:mssng_20171020a` data-set); see the Repositories sub-section and the Examples section for how to access and query BigQuery tables.

subject

The `subject` table provides basic information about each individual in the database such as sex, date of birth, and whether they are affected:

Field	Description
INDEXID	Unique identifier for the individual
FATHERID	Identifier of the individual's father
MOTHERID	Identifier of the individual's mother
AFFECTION	"1" if unaffected or "2" if affected
SEX	"M" (male), "F" (female)
FAMILYID	Family identifier

FAMILYTYPE	“SPX” (simplex), “MPX” (multiplex)
DOB	Date of birth; yyyy-mm-dd. (if information available). Day set to “01” for anonymization.

measures

Subjects’ psychometric test results using established scales, typically available only for affected subjects. Subjects are identified by `INDEXID`. Test results are linked to the date at which the test(s) were run (TESTDATE). For subjects with single measurements at different dates, the measurement can be usually collapsed together, while ad-hoc rules need to be used for subjects with more than one measurement at different dates. The number of measurements available is too large for a detailed description in this document. Please see this spreadsheet for a more detailed explanation (link).

Test data is available in the measures table.

measures: table with 4 columns organized in tidy format (many records per subject)

measures

Field	Description
INDEXID	Unique identifier for the individual
CODE	Identifier for the type of test
TESTDATE	Date in which the test was administered
MEASURE	Test result

subject_sample

The `subject_sample` table provides metadata about all genome samples available in the MSSNG database. `SUBMITTEDID` is the genome sample identifier that you should use to join subject/sample data to the variant data ‘`call.call_set_name`’ field.

Field	**Description**
SUBMITTEDID	Unique identifier for the genome sample. Note that while this value is usually the same as the INDEXID, that is not always the case. This corresponds to ‘call.call_set_name’ in the variant tables.
INDEXID	Unique identifier of the individual found in the `subject` table

DNASOURCE	biological sample type used as DNA source: "Blood" (fresh blood), “White blood cell” (frozen as opposed to fresh white blood cells), "Cell line" (lymphoblastoid cell line), “Saliva”
PLATFORM	sequencing platform: "Illumina HiSeq" (HiSeq2000), “Illumina HiSeq2500”, “Illumina HiSeqX”, "Complete Genomics" (different pipeline versions)
NIMHID	NIMH identifier
RUDCRID	Rutgers repository identifier
COMMENTS	Any specific comments regarding a subject
SOFTWARE_VERSION	Complete Genomics software version used to sequence sample
PREDICTED_ANCESTRY	Predicted ancestry of sample. Consensus of computationally derived predictions from two tools

Subjects may have multiple samples, and each will be referenced as a separate row in the `subject_sample` table.

Aligned Reads

In the MSSNG database, an individual genome sample's reads are available as a set of aligned BAM files. Aligned Reads are available for all 5,579 Illumina genome samples. The “b37” human genome reference was used. For further information about the alignment pipeline for MSSNG Illumina samples, please read this document.

BAM files are available to researchers by following the Process for Researchers to Access MSSNG BAM and VCF files.

Called Variants

In the MSSNG database, an individual sample's variants can be found in BigQuery tables as well as in VCF files.

variants

The `variants` table contains both true variants as well as reference segments (gVCF data) for all samples. In addition to the description here, a variants table codelab is available.

The `variants` table fields are described here:

Field	Description
reference_name	Chromosome identifier following b37 conventions (e.g. chromosome 1 is represented as “1”, chromosome X as “X”)
start	0-positional variant start, reference: b37
end	0-positional variant end, reference: b37
reference_bases	reference sequence at variant locus
alternate_bases	alternate allele sequence(s) at variant locus
quality	[this field needs to be removed, please ignore]
filter	[this field needs to be removed, please ignore]
names	[no description]
call	the call record contains all information items specific to a given variant call (as opposed to more generally a variant locus, defined as a genome locus subject to variation)
call.call_set_id	genome sample identifier, just for internal use
call.call_set_name	genome sample identifier, corresponding to the SUBMITTEDID
call.genotype	VCF-coded genotype index, typically two values per variant call (e.g. 1,1 corresponds to homozygous, 1,0 or 0,1 corresponds to heterozygous reference+alternate, 1,2 corresponds to heterozygous with two alternate alleles)
call.phaseset	phase identifier (available only for a subset of the data)
call.genotype_likelihood	GATK genotype likelihoods
call.AD	allelic read counts (typically two values per variant call, add up to DP)
call.DP	total read count (GATK: reads with MQ=255 or with bad mates are filtered)
call.EHQ	available only for Complete Genomics, corresponds to EAF allelic quality scores (use for more stringent quality filtering)
call.FILTER	main quality filter (GATK: based on VQSR, Complete Genomics: derived from VarFilter), use = “PASS” to select variants of minimum quality

call.GQ	GATK genotype quality
call.HQ	GATK haplotype quality
call.MIN_DP	GATK minimum DP observed within the GVCF block
call.PGT	GATK phased genotype index
call.PID	[no description]
call.QUAL	GATK variant quality, not commonly used for quality filters
call.BaseQRankSum	Z-score from Wilcoxon rank sum test of Alt Vs. Ref base qualities
call.ClippingRankSum	Z-score From Wilcoxon rank sum test of Alt vs. Ref number of hard clipped bases
call.DP	Approximate read depth; some reads may have been filtered
call.FS	Phred-scaled p-value using Fisher's exact test to detect strand bias
call.GQ_MEAN	Mean of all GQ values
call.MLEAC	Maximum likelihood expectation (MLE) for the allele counts (not necessarily the same as the AC), for each ALT allele, in the same order as listed
call.MLEAF	Maximum likelihood expectation (MLE) for the allele frequency (not necessarily the same as the AF), for each ALT allele, in the same order as listed
call.MQ	RMS Mapping Quality
call.MQ0	Total Mapping Quality Zero Reads
call.MQRankSum	Z-score From Wilcoxon rank sum test of Alt vs. Ref read mapping qualities
call.QD	Variant Confidence/Quality by Depth
call.ReadPosRankSum	Z-score from Wilcoxon rank sum test of Alt vs. Ref read position bias
call.SOR	Symmetric Odds Ratio of 2x2 contingency table to detect strand bias
call.VQSLOD	Log odds ratio of being a true variant versus being false under the trained gaussian mixture model

call.culprit	The annotation which was the worst performing in the Gaussian mixture model, likely the reason why the variant was filtered out
AC, AF, AN, NCC, NS	[these fields are from GATK and they are useful as summary statistics only when jointly calling a larger number of samples; genomes are currently called one at a time, so currently they cannot be meaningfully used]
POSITIVE_TRAIN_SITE	marks variants used in the GATK VQSR positive training set
NEGATIVE_TRAIN_SITE	marks variants used in the GATK VQSR negative training set
DB	dbSNP membership

Note that for insertion and deletion representation we follow the VCF convention of capturing the first reference base before the insertion or deletion within the variant locus (e.g. reference_bases: AT, alternate_bases: A, for a deletion of T).

Called variants are available for all 7,231 Illumina and Complete Genomics genome samples. For further information about the variant calling pipeline for MSSNG Illumina samples, please read this document. Complete Genomics variant calls are processed using a handful of custom steps, which structure the data to be more like the Illumina variant calls. More information can be found here.

passing_variants

When querying variants there are some common repeated patterns:

- Eliminate low-quality variants (FILTER != 'PASS')
- Eliminate reference segments (gVCF non-variant segments)

The passing_variants table was created to facilitate analysis on high quality variants. Also since queries are frequently performed over specific regions of the genome, the `passing_variants` table has been partitioned by chromosome in tables named `passing_variants_<chr>` where chr is in (1..22, X, Y, MT).

The `passing_variants_<chr>` tables support the Advanced Variant Search in the MSSNG Portal.

The `passing_variants` table schema is nearly the same as the `variants` table. The `variant_id` field is removed from the `passing_variants` table.

annotated_variants

Querying variants commonly involves looking at variants in affected individuals, assessing variant damage potential, and looking at variant inheritance. While the MSSNG BigQuery dataset contains the data to do this via joining the `passing_variants`, `annotations`, `subject`, and `subject_sample` tables, a pre-joined, pre-filtered table called `annotated_variants` was created.

The Advanced Variant Search in the MSSNG Portal uses the `passing_variants`, `annotations`, `subject`, `subject_sample`, `variants_denovo` and `variant_sanger` tables.

The OneBox Search and Trio Search in the MSSNG Portal use the `annotated_variants` table.

The `annotated_variants` table contains:

- for affected individuals:
  - variants with frequency less than 10% (see [*] below)
  - variants with damage potential > 0 (see [**] below)
  - variant calls quality filter = `PASS`
- for parents
  - variant calls, independent of quality filter, if any affected child has the variant

The `annotated_variants` table does not contain non-transmitted rare variants from the parents.

[*] Rare variants are defined as occurring at a frequency of < 10% in:

- 1000 genomes (all, eur, amr, eas, afr)
- NHLBI (all, aa, ea)
- ExAC (all, AFR, AMR, EAS, FIN, NFE, OTH, SAS)
- cg1KG436_AllFreq, cg1KG436_CalledFreq, cgW597_AllFreq, cgW597_CalledFreq
- gnomAD_exome (ALL, AFR, AMR, ASJ, EAS, FIN, NFE, OTH, SAS)
- gnomAD_genome (ALL, AFR, AMR, EAS, FIN, NFE, OTH).

[**] The damage potential of variants are scored on Annovar annotations. Variants are classified as having zero or more of eight possible effects (Frameshift, Stop Gain, Splice Site, Missense, Other, Predicted Splicing, UTR, Non-coding RNA gene) depending on the Annovar assigned “typeseq” and “effect”. The effects are further categorized as having “High”, “Medium”, or “Low” impact depending on several other values resulting from Annovar annotation, including: sift_score, polyphen_score, ma_score, phylopPMam_avg, phylopVert100_avg, CADD_phred, mt_score, phastCons_placental, dbsnp, dbsnp_common, dbsnp_region.

- Loss of function (frame-shift, stop gain and splice variants): high impact
- Missense: based on the number of methods tagging variant as damaging are flagged as high (>=4 methods), medium (2-3 methods) or low (1 method) impact variants

- Other coding and non-coding: based on CADD_phred, phylopPMam_avg and phylopVert100_avg conservation scores are tagged high, medium or low impact variants.

The annotated_de_novo_variants table is also available for download. A separate tab includes definitions for the column headers.

Annotations

The variant annotations (BigQuery table `annotations`) are generated using an Annovar-based pipeline, following Annovar priority rules to report variants. For further information on the databases/data sources and versions used to derive annotations, please read this document.

Field	Description
id	variant id (chromosome-start-end-reference-alternate)
typeseq	type of sequence overlapped, with respect to known genes/transcripts and their coding / noncoding status: (a) “exonic” represents coding exons, (b) “exonic:splicing” represents the beginning/end of coding exons which may also affect splicing, (c) “splicing” represents core splicing site (2 bp on the intron side of intron-exon and exon-intron junctions), (d) "ncRNA_exonic" represents exons of non-coding RNA genes, (e) "ncRNA_splicing" represents core splicing sites of non-coding RNA genes , (f) “UTR5” represents 5’ untranslated region, (g) “UTR3” represents 3’ unstranslated region, (h) “upstream” represents 1kb ubstream of TSS, (i) “downstream” represents 1kb downstream of TSS and (j) “intergenic” represents intergenic regions (beyond upstream/downstream threshold(1kb)). For variants with multiple sequence overlaps (eg, exonic for one transcript and intronic for other), all possible typseq values will be listed in semicolon-delimited format (eg: exonic;intronic).
typeseq_priority	Prioritized sequence overlap for multi-sequence overlap variants. Annovar prioritization scheme was used for implementing this (http://annovar.openbioinformatics.org/en/latest/user-guide/gene/).
refseq_id	combined Annovar output on coding sequence mapping and effect, composed of: (a) for coding exonic changes (typeseq “exonic”): gene official symbol, RefSeq transcript isoform ID, position in the coding sequence, amino acid change; (b) for core splice site changes (typeseq "exonic"): gene official symbol, RefSeq transcript isoform ID, exon number, coding

	sequence position and change. In very rare cases, the UCSC RefSeq tables used by Annovar have a coding frame error; Annovar is able to catch these and issue a warning; in these cases, the UCSC known type of sequence overlap, effect and gene mapping replaces RefSeq (this happens very rarely)
effect	type of effect on the coding sequence: (a) “synonymous SNV”, (b) "nonsynonymous SNV", (c) "stopgain SNV", (d) "frameshift deletion", (e) "frameshift insertion", (f) "frameshift substitution", (g) "nonframeshift deletion", (h) "nonframeshift insertion", (i) "nonframeshift substitution" , (j) “stoploss SNV”. For variants with multiple effects, all possible values will be represented in comma-separated fashion.
effect_priority	Prioritized effects for coding variants with multiple effects (http://annovar.openbioinformatics.org/en/latest/user-guide/gene/).
aa_flag	this flag is set to 1 if more than one distinct amino acid change is reported in the “refseq_id” field
leftD, rightD	“left” and “right” distance from the two nearest splice sites
gene_symbol	official gene symbol, extracted from the “refseq_id” field
entrez_id	NCBI entrez-gene id
gene_desc	full gene name
gene_type	protein coding or specific type of ncRNA genes: snRNA (small nuclear) and snoRNA (small nucleolar), antisense, tRNA, rRNA, readthrough, pseudogene, or unknown
Omim_id, omim_phenotype	omim gene accession id, omim disorder/disease description when available for the corresponding omim gene accession
MPO	MPO (Mammalian Phenotype Ontology) top level phenotype(s), imported from MGI and mapped from the human orthologs of the mouse gene (orthology is based on NCBI Homologene). Each top level phenotype associated to the gene is reported as: MPO term ID, MPO term description, type of experiment (het, hom, etc...), using “@” as separator
HPO	(set of) HPO (Human Phenotype Ontology) top level phenotype(s), imported from HPO. Each top level phenotype associated to the gene is reported as: HPO term ID, HPO term description, mode of inheritance (AD: autosomal dominant, XL: X-linked, AR: autosomal recessive), using “@” as separator. Primary HPO annotations are up-propagated to top level phenotypes using the HPO ontology graph

CGD_disease, CGD_inheritance	The Clinical Genomics Database is compiled by curators and maintained by the NHGRI (National Human Genome Research Institute); for every gene in the database, the CGD provides a list of one or more genetic disorders and a mode of inheritance; these field report the genetic disorder(s), and relative mode of inheritance
ExAc_mis_Z, ExAc_lof_Z, ExAc_pLI	Missense Z-score from ExAc, Loss-of-function Z-score from ExAc, Probability of being loss-of-function intolerant
ACMG_disease	Any (exonic, intronic or splice) variants in genes in ACMG published recommendations for reporting incidental findings (https://www.ncbi.nlm.nih.gov/clinvar/docs/acmg/)
dbsnp, dbsnp_common, dbsnp_region, dbsnp_wind	exact match (by coordinates, reference allele and alternate allele) to dbSNP, exact match (by coordinates, reference allele and alternate allele) to common dbSNP track UCSC, Annovar overlap-based match for common dbSNP track (UCSC), window (+/- 7 bp) overlap-based match for dbSNP
cosmic	exact match (position, allele) to the Cosmic database of somatic variants
Clinvar_SIG, Clinvar_CLNREF, Clinvar_CLNACC, Clinvar_SIG_ord, Clinvar_ReviewStatus	Overall ClinVar significance code; “pathogenic” is the code of interest for rare disorders. (https://www.ncbi.nlm.nih.gov/clinvar/docs/clinsig/), clinvar associated disorder/disease, clinvar accession ID, Clinical significance(s) for individual submissions (SCV) in ClinVar, The level of review supporting the assertion of clinical significance (https://www.ncbi.nlm.nih.gov/clinvar/docs/details/#review_status)
A1000g_all, A1000g_eur, A1000g_amr, A1000g_eas, A1000g_afr, A1000g_sas	1000 Genome allele frequencies (global and different ethnic subsets)
NHLBI_all, NHLBI_aa, NHLBI_ea	NHLBI-ESP allele frequencies (global and different ethnic subsets)
ExAC_Freq, ExAC_AFR, ExAC_AMR, ExAC_EAS, ExAC_FIN, ExAC_NFE, ExAC_OTH, ExAC_SAS	ExAC allele frequencies (global and different ethnic subsets)

cg, cg_filtered	Allele frequencies in the unrelated 54 CGI ethnically-diverse controls (no quality filters, quality-filtered)
cgW597_AllFreq, cgW597_CalledFreq, cgW597_11s, cgW597_Hs, cgW597_Ls, cg1KG436_AllFreq, cg1KG436_CalledFreq, cg1KG436_11s, cg1KG436_Hs, cg1KG436_Ls	Allele frequencies in two internal Complete Genomics database of 597 and 436 apparently healthy individuals (frequencies with total allele count as denominator, frequencies with called allele count as denominator, homozygous allele count, high quality allele count, low quality allele count).
gnomAD_exome_ALL, gnomAD_exome_AFR, gnomAD_exome_AMR, gnomAD_exome_ASJ, gnomAD_exome_EAS, gnomAD_exome_FIN, gnomAD_exome_NFE, gnomAD_exome_OTH, gnomAD_exome_SAS, gnomAD_genome_ALL, gnomAD_genome_AFR, gnomAD_genome_AMR, gnomAD_genome_ASJ, gnomAD_genome_EAS, gnomAD_genome_FIN, gnomAD_genome_NFE, gnomAD_genome_OTH	Genome Aggregation Database allele frequencies (global and different ethnic subsets) for exomes and whole genome sequences
sift_score, polyphen_score, PROVEAN score, ma_score, mt_score, CADD_phred	dbNSFP pre-computed missense effect score: SIFT (values <= 0.05 correspond to damaging), Polyphen2 HVAR (values >= 0.90 correspond to damaging), amino acid substitution or indel prediction score from Provean software (values < -2.5 corresponds to damaging, Mutation Assessor (values >= 1.9 correspond to damaging), Mutation Taster (values >= 0.5 correspond to damaging), CADD (values >= 15 correspond to damaging)
phylopPMam, phylopPMam_avg, phylopVert100, phylopVert100_avg	PhyloP conservation values for placental mammals (PMam) and 100 vertebrates (Vert100); for variants spanning more than one position, please refer to [...]_avg for the average value
phastCons_placental	PhastCons element conservation score placental mammals (> 0 corresponds to conservation)
gerp_elem, gerp_wgs	Rejected substitution(RS) score for GERP++ elements, whole-genome GERP++ RS scores greater than 2 (smaller scores indicate less conservation)

pfam_annovar	overlap with PFAM protein domain (coding exons only)
spx_dpsi, spx_dpsi_z, spx_gene, spx_strand, spx_transcript, spx_exonN, spx_seqType, spx_effType, spx_spliceDist	splicing regulatory exon inclusion/exclusion predicted difference in percentage of transcripts with the exon (treat < -3.5 as potentially damaging, and < -5 as damaging), corresponding z-score, gene symbol, strand, transcript id, exon number, sequence type, sequence effect type, distance from splice site
dbscSNV_ADA_SCORE, dbscSNV_RF_SCORE	Splice site prediction scores from dbscSNV. dbscSNV_RF_SCORE or dbscSNV_ADA_SCORE > 0.6, the variant is predicted to impact splicing
per_cds_affected	percentage of coding exonic sequence affected
per_transcripts_affected	percentage of transcripts with variant overlapping them and reported following Annovar prioritization rules
SegDup	overlap with UCSC Segmental Duplications
Repeat	overlap with UCSC RepeatMasker
effect_impact	pre-computed effects and impacts for the MSSNG portal

See the example section for how to join this table to the variant table.

De-novo Variants

De-novo variants are called in probands but not present in the parent's genomes. De-novo calling is currently not part of the automated genome analysis pipeline, and de-novo variants are available only for a subset of the genomes (2281/3425 affected). The `variants_de_novo` table lists all available de-novo variants. The table can also be viewed as a spreadsheet.

Field	Description
id	as in `annotation` table
reference_name	as in `variant` table
start	as in `variant` table
end	as in `variant` table
reference_bases	as in `variant` table
alternate_bases	as in `variant` table

PLATFORM	as in `subject_sample` table
COMMENT	Any comments about this variant
SUBMITTEDID	as in `subject_sample` table

Sanger-validated Variants

For the variants in the `variants_sanger` table, Sanger validation results are available; positive as well as negative results are reported. Only a small minority of variants have undergone Sanger validation.

Field	Description
id	as in `annotation` table
reference_name	as in `variant` table
start	as in `variant` table
end	as in `variant` table
reference_bases	as in `variant` table
alternate_bases	as in `variant` table
Sanger_exists	possible values: YES (variant found), NO (variant not found)
Sanger_inheritance	possible values: de novo, maternal inherited, paternal inherited, NA (for variants not found)
PLATFORM	as in `subject_sample` table
SUBMITTEDID	as in `subject_sample` table

Copy Number Variants (CNVs)

For samples sequenced on Illumina platforms, copy number variants (CNVs) were detected using ERDS and CNVnator (with default parameters) as described by Trost et al. 2018. For CNVnator, calls for which more than 50% of the reads in the CNV region were q0 (zero mapping quality) were removed (q0 filter), except for in homozygous autosomal deletions or X-linked deletions in males (with normalized average read depth, NRD, less than 0.03). Stringent CNVs are defined as those greater than 1kb and detected by both algorithms with minimum 50% reciprocal overlap. For samples sequenced on HiSeqX, duplications less than

50 kb, detected by only ERDS are also considered as high quality, but have a higher false discovery rate than stringent CNVs. For samples sequenced by Complete Genomics, CNV calls were used as provided, with all CNVs being greater than 2 kb. We defined a rare CNV as being detected at a frequency of less than or equal 1% in the parental samples in MSSNG, across all sequencing platforms.

References:

Zhu, M. et al. Using ERDS to infer copy-number variants in high-coverage genomes. American Journal of Human Genetics 91:408–421 (2012).

Abyzov, A. et al. CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. Genome Research 21:974–984 (2011).

Trost, B. et al. A Comprehensive Workflow for Read Depth-Based Identification of Copy-Number Variation from Whole-Genome Sequence Data. American Journal of Human Genetics 4:142-155 (2018).

Field	Description
Sample	SubmittedID
Chromosome	Chromosome
Start	CNV Start
End	CNV End
CNVType	CNV type (DEL, DUP or DUP|DEL)
CopyNumber	Assigned copy number (CG: CN, HISEQ: ERDS CN|CNVnator normalized read depth, HISEQ2000 and HISEQX: ERDS CN)
Size	Size of CNV
Overlap	Overlap between CNV calls when more than one method is used for CNV detection (HISEQX and HISEQ2000: overlap of ERDS call with CNVnator call | overlap of CNVnator call with ERDS calls, HISEQ: fraction of the CNV detected by both ERDS

	and CNVnator)
Putative_Inheritance	Computed inheritance (Maternal, Paternal, Inherited_Ambiguous: both parents have the variant, Ambiguous, no_parent: parents not sequenced, one_parent: only one parent sequenced, p_denovo: putative de novo)
GC_Content_Percent	GC_content
CytobandAnn	Cytoband
Gene_Symbol	official gene symbol, transcript overlap
Gene_egID	Entrez gene IDs, transcript overlap
Exon_Symbol	official gene symbol, exonic overlap
Exon_egID	official gene Entrez gene ID, exonic overlap
CDS_Symbol	official gene symbol with CDS overlap
CDS_egID	official gene Entrez ID with CDS overlap
ISCA_region	Genomic disease region from ISCA db
CNV_ISCA_percOverlap	% length of CNV overlapped by ISCA region
ExAC_pLI	ExAC pLI value
UncleanGenome_percOverlap	overlap with gaps, segmental duplications, centromere, telomere etc
MPO_NervousSystem	MPO terms part of nervous system phenotype
HPO_NervousSystem	HPO terms part of nervous system phenotype
CGD	CGD db
OMIM_MorbidMap	Morbid Map
DECIPHER_region	Genomic disease region from Decipher db
CNV_decipher_percOverlap	% length of CNV overlapped by Decipher region
DGV_N_studies	N studies overlap CNV/DGV restricted to 50% overlap
DGVpercFreq_subjects_allStudies	DGV (50% reciprocal overlap) any study

DGVpercFreq_subjects_coverageStudies	DGV (50% reciprocal overlap), only study with coverage
DGV_percOverlap_any	DGV (no cutoff used)
DGV_50percRecipOverlap	DGV (50% reciprocal overlap)
CGparentalPercFreq_50percRecipOverlap	Internal MSSNG database - MSSNG parents sequenced by Complete Genomics
erdsPercFreq_50percRecipOverlap	Internal MSSNG database - MSSNG parents sequenced by Illumina HiSeqX called by ERDS
cnvnatorPercFreq_50percRecipOverlap	Internal MSSNG database - MSSNG parents sequenced by Illumina HiSeqX called by CNVN
Comment	Tagging high confidence rare variants
Curated	Tagging manually curated de novo CNVs and chromosomal abnormalities
Platform	Platform (CG, HISEQ, HISEQX, HISEQ2000)

Repositories

Data for the MSSNG database is stored in two repositories:

- Google BigQuery
- Google Cloud Storage

Google BigQuery is a service designed for storing generic structured data and allowing for querying over massive datasets in seconds. Google BigQuery supports a SQL-like query language which can be accessed via the BigQuery web-based interface, command line tool, or programmatic API. This allows access to data from any data analysis tool (such as R or Python) that supports the Google BigQuery API.

Google Cloud Storage is a repository for storing and sharing files. Cloud Storage supports storing and retrieving files using a web-based interface, command-line tool, or programmatic API. This allows file-based access to data from any data analysis tool (such as R or Python).

MSSNG data is available in the following repositories:

	Google BigQuery	Google Cloud Storage

Aligned Reads		X
Called Variants	X	X
Sample/subject Data	X	

Access

The following describes ways to access the MSSNG data stored in the Google Cloud:

- To get started, you can access the MSSNG researcher portal at https://research.mss.ng/
- If you would like to issue custom queries against the MSSNG BigQuery tables, then you will need to create a Google Cloud Project. See the instructions below for getting started using BigQuery.
- If you would like to download the BAM and VCF files, please refer to the document on the process for researchers to access MSSNG BAM and VCF files.

When you have created your Genomics-enabled project, you will be ready to use all of the tools discussed in the next section.

Tools

With a Google Cloud project created you will be ready to start accessing data in the MSSNG database. The following examples will demonstrate basic use of:

- BigQuery web interface
- R interface BigQuery

Examples

Subject/sample data

Subject/sample and variant data is stored in Google BigQuery. Genomics data in BigQuery is most commonly accessed through the BigQuery web interface and the BigQuery interface for R.

BigQuery web interface

The BigQuery web interface can be used for issuing ad hoc queries over the genomic variant data and subject/sample data.

Setup

The following steps demonstrate accessing the MSSNG subject/sample data.

1. Go to https://bigquery.cloud.google.com
2. Set the active project
   a. If the project name in the left-hand navigation is **MSSNG Portal**, then it must be changed:
      i. Click on the drop down icon beside MSSNG Portal in the left-hand navigation pane.
      ii. Pick *'Switch to project'* in the menu, and then select your Google Cloud Project from the list.
3. Add the MSSNG project to the list of available datasets
   a. Click on the drop down icon beside your project name in the left-hand navigation pane.
   b. Pick *'Switch to project'* in the menu, and then select *'Display Project'*.
   c. In the Add Project dialog, enter the Project ID "`idyllic-analyst-574`"

In the left-hand navigation pane, you should see listed MSSNG project:

- `idyllic-analyst-574`

If you click on the `idyllic-analyst-574` project, it should expand to show the dataset:

- `mssng_20171020a`

If you click on the `mssng_20171020a` dataset it should expand to show (among many others), the tables:

  - `annotations`
  - `annotations_{1..22,MT,X,Y}`
  - `annotated_variants`
  - `annotated_de_novo_variants`

  - `passing_variants`
  - `passing_variants_{1..22,MT,X,Y}`
  - `subject`
  - `subject_sample`
  - `variants`
  - `variants_de_novo`
  - `variants_sanger`

Subject/Sample Data Example

Your first example query will be on the subject table. Clicking on the `subject` table will open the New Query pane on the right hand side.

In the New Query text area enter the query:

```
#standardSQL
SELECT
  sex,
  COUNT(INDEXID) AS count
FROM
  `idyllic-analyst-574.mssng_20171020a.subject`
GROUP BY
  sex
ORDER BY
  sex
```

Clicking on the Run Query button should generate results in a few seconds which looks like:

Row	SEX	count	
1	F	2614	
2	M	4573	

To see the number of ASD affected individuals, change the query to:

```
#standardSQL
SELECT
  sex,
  COUNT (INDEXID) AS count
FROM
  `idyllic-analyst-574.mssng_20171020a.subject`
WHERE
```

```
  affection = '2'
GROUP BY
  sex
ORDER BY
  sex
```

(note that: `AFFECTION = '2'` means: ASD affected)

Clicking on the Run Query button should generate results in a few seconds which looks like:

Row	gender	count	
1	F	734	
2	M	2691	

More Subject/Sample data Query Examples

For more queries, see the Subjects and Samples codelab.

Genomic Variants Examples

Genomic variants are stored in the `mssng_20171020a.variants` and `passing_variants` tables (described above). This table uses some features of Google BigQuery not commonly seen in relational databases (which you may already be familiar with), namely Array fields.

Each record in the `variants` table describes a variant for which has been called at least once within the set of samples. Within the variant record is `call` field, which contains a reference to all calls of this variant.

The schema for the `variants` table can be found by:

1. Select the `variants` table in the left hand pane of the BigQuery interface
2. The Schema button in the right hand pane should be selected by default and the Table Details should be displayed.

To see a sampling of the data, select the Details button. Below the table Details will be a Preview of the data.

For sample queries, see the Variants codelab.

Many more example queries on the variants table can be found here.

To build your own, more sophisticated queries, see the BigQuery Query Reference.

R interface to BigQuery

Data from Google BigQuery can be queried from R using the bigrquery package.

Setup

To install the bigrquery package, launch R and execute:

```
install.packages("bigrquery")
```

Query phenotypes

Once successfully installed, the following R code can be used to query the phenotype data, as in the BigQuery example above:

```
library(bigrquery)

# Specify the id of the project you created
project <- "<your project id>"

# Define a variable to hold the query
querySql <- "
#standardSQL
SELECT
  sex,
  COUNT(INDEXID) AS count
FROM
  `idyllic-analyst-574.mssng_20171020a.subject`
GROUP BY
  sex
ORDER BY
  sex

"

# Display the updated SQL.
cat(querySql)
```

```
# Dispatch the query to BigQuery for execution.
result <- query_exec(querySql, project)

# Emit query results
result
```

The following results should be displayed:

```
 sex count
1      F   2614
2      M   4573
```

Many more example queries on the variants table via R can be found in the Google Genomics Getting Started with BigQuery repository.