***Variant Workbench Reference Sheet***

Contents

[Kids First Primary Files 3](#_heading=h.17dp8vu)

[variants 3](#_heading=h.z337ya)

[occurrences [kf-strides-variant-parquet-prd] 6](#_heading=h.1y810tw)

[consequences 8](#_heading=h.2xcytpi)

[Public Tables 12](#_heading=h.qsh70q)

[orphanet\_gene\_set 12](#_heading=h.32hioqz)

[omim\_gene\_set 13](#_heading=h.1hmsyys)

[human\_genes 14](#_heading=h.41mghml)

[hpo\_terms 15](#_heading=h.2grqrue)

[hpo\_gene\_set 16](#_heading=h.vx1227)

[gnomad\_genomes\_3\_1\_1 (\_subset) 17](#_heading=h.3fwokq0)

[dbnsfp 18](#_heading=h.1v1yuxt)

[gnomad 29](#_heading=h.4f1mdlm)

[genes 32](#_heading=h.2u6wntf)

[ensembl\_mapping 33](#_heading=h.19c6y18)

[1000\_genomes 34](#_heading=h.3tbugp1)

[annovar 35](#_heading=h.28h4qwu)

[cancer\_hotspots 38](#_heading=h.unwguv2sr1rg)

[clinvar 41](#_heading=h.nmf14n)

[cosmic\_gene\_set 42](#_heading=h.37m2jsg)

[ddd\_gene\_set 43](#_heading=h.1mrcu09)

[Kids First Non-Variant Files 44](#_heading=h.46r0co2)

[biospecimens 44](#_heading=h.2lwamvv)

[biospecimens\_diagnoses 46](#_heading=h.111kx3o)

[diagnoses 47](#_heading=h.3l18frh)

[families 48](#_heading=h.206ipza)

[family\_relationships 49](#_heading=h.4k668n3)

[genomic\_files 50](#_heading=h.2zbgiuw)

[investigators 51](#_heading=h.1egqt2p)

[outcomes 52](#_heading=h.3ygebqi)

[participants 53](#_heading=h.2dlolyb)

[phenotypes 54](#_heading=h.sqyw64)

[sequencing\_experiments 55](#_heading=h.3cqmetx)

[studies 56](#_heading=h.1rvwp1q)

[study\_files 57](#_heading=h.4bvk7pj)

[**Variant Workbench Functions 58**](#_heading=)

[Count 58](#_heading=h.puho9wjp2xbh)

[Select 58](#_heading=h.2r0uhxc)

[Drop 58](#_heading=h.1664s55)

[Filter 58](#_heading=h.3q5sasy)

[Join 59](#_heading=h.25b2l0r)

[Show 59](#_heading=h.kgcv8k)

[printSchema 59](#_heading=h.34g0dwd)

[Convert to Pandas dataframe 59](#_heading=h.1jlao46)

# README

This document is intended to be used as a reference guide for using the Kids First variant workbench on CAVATICA. Each file is listed here along with the column names of data contained within that file. Some of the commonly used files also have a brief description, and additional descriptions will be added in the future. This document is best used with the search function, rather than scrolling through. It also matches up with the [example notebook](https://cavatica.sbgenomics.com/u/jared.rozowsky/ashg-variant-workbench/analysis/cruncher/variant-workbench-starter-kit) on the [CAVATICA project](https://cavatica.sbgenomics.com/u/jared.rozowsky/ashg-variant-workbench). In addition to this reference sheet, there is a useful [table mapping search/filter functions on the portal to filter functions on the CAVATICA variant workbench](https://github.com/kids-first/variant-workbench-migration/blob/main/KFDRP-Portal-to-VWB.xlsx).

# Kids First Primary Files

## variants

[Registered Access]

The **variants** file contains all variants found in Kids First registered studies as well as which studies they are found in. There is some additional, commonly used, information, such as frequencies in Kids First, 1000 Genomes, Topmed, and gnomad, and clinical significance.

Code to read in file:

variants = spark.read.parquet('/sbgenomics/project-files/variants')

#### Columns

* start: long
* reference: string
* alternate: string
* name: string
* end: long
* hgvsg: string
* variant\_class: string
* lower\_bound\_kf\_ac\_by\_study: map
* key: string
* value: long
* lower\_bound\_kf\_an\_by\_study: map
* key: string
* value: long
* lower\_bound\_kf\_af\_by\_study: map
* key: string
* value: double
* lower\_bound\_kf\_homozygotes\_by\_study: map
* key: string
* value: long
* lower\_bound\_kf\_heterozygotes\_by\_study: map
* key: string
* value: long
* upper\_bound\_kf\_ac\_by\_study: map
* key: string
* value: long
* upper\_bound\_kf\_an\_by\_study: map
* key: string
* value: long
* upper\_bound\_kf\_af\_by\_study: map
* key: string
* value: double
* upper\_bound\_kf\_homozygotes\_by\_study: map
* key: string
* value: long
* upper\_bound\_kf\_heterozygotes\_by\_study: map
* key: string
* value: long
* studies: array
* element: string
* consent\_codes: array
* element: string
* consent\_codes\_by\_study: map
* key: string
* value: array
* element: string
* transmissions\_by\_study: map
* key: string
* value: map
* key: string
* value: long
* release\_id: string
* zygosity: array
* element: string
* frequencies: struct
* upper\_bound\_kf: struct
* an: long
* ac: long
* af: double
* homozygotes: long
* heterozygotes: long
* lower\_bound\_kf: struct
* an: long
* ac: long
* af: double
* homozygotes: long
* heterozygotes: long
* transmissions: map
* key: string
* value: long
* 1k\_genomes: struct
* ac: integer
* an: integer
* af: double
* topmed: struct
* ac: integer
* an: integer
* af: double
* homozygotes: integer
* heterozygotes: integer
* gnomad\_genomes\_2\_1: struct
* ac: integer
* an: integer
* af: double
* hom: integer
* gnomad\_exomes\_2\_1: struct
* ac: integer
* an: integer
* af: double
* hom: integer
* gnomad\_genomes\_3\_0: struct
* ac: integer
* an: integer
* af: double
* hom: integer
* gnomad\_genomes\_3\_1\_1: struct
* ac: integer
* an: integer
* af: double
* hom: integer
* dbsnp\_id: string
* clinvar\_id: string
* clin\_sig: array
* element: string
* chromosome: string

## occurrences [kf-strides-variant-parquet-prd]

[Controlled Access – must have dbgap approval]

The **occurrence** files identify the Kids First participant(s) that have specific variants. For a list of all variants, look at the [variants](#_heading=h.30j0zll) file.

Code to read in file for one study:

*occ = spark.read.parquet('/sbgenomics/project-files/kf-strides-variant-parquet-prd/occurrences\_family/occurrences\_family\_sd\_preasa7s\_re\_000017/study\_id=SD\_PREASA7S/dbgap\_consent\_code=phs001138.c1/family\_id=FM\_YZP5K1HF/participant\_id=PT\_Q4NV8YBD')*

#### Columns

* biospecimen\_id: string
* start: long
* end: long
* reference: string
* alternate: string
* name: string
* hgvsg: string
* variant\_class: string
* ad: array
* element: integer
* dp: integer
* gq: integer
* calls: array
* element: integer
* is\_multi\_allelic: boolean
* old\_multi\_allelic: string
* quality: double
* filter: string
* info\_ac: integer
* info\_an: integer
* info\_af: double
* info\_culprit: string
* info\_sor: double
* info\_read\_pos\_rank\_sum: double
* info\_inbreeding\_coeff: double
* info\_pg: array
* element: integer
* info\_fs: double
* info\_dp: integer
* info\_ds: boolean
* info\_info\_negative\_train\_site: boolean
* info\_positive\_train\_site: boolean
* info\_vqslod: double
* info\_clipping\_rank\_sum: double
* info\_raw\_mq: double
* info\_base\_qrank\_sum: double
* info\_mleaf: double
* info\_mleac: integer
* info\_mq: double
* info\_qd: double
* info\_db: boolean
* info\_m\_qrank\_sum: double
* info\_excess\_het: double
* info\_haplotype\_score: double
* file\_name: string
* release\_id: string
* is\_normalized: boolean
* is\_lo\_conf\_denovo: boolean
* is\_hi\_conf\_denovo: boolean
* is\_gru: boolean
* is\_hmb: boolean
* is\_proband: boolean
* affected\_status: boolean
* gender: string
* mother\_id: string
* father\_id: string
* mother\_calls: array
* element: integer
* father\_calls: array
* element: integer
* mother\_affected\_status: boolean
* father\_affected\_status: boolean
* zygosity: string
* mother\_zygosity: string
* father\_zygosity: string
* parental\_origin: string
* transmission: string
* family\_id: string
* participant\_id: string
* study\_id: string
* has\_alt: integer
* dbgap\_consent\_code: string
* chromosome: string

## consequences

[Registered Access]

This table provides additional information about variants. Particularly, this table can be used to link variants to gene name, for gene specific queries. It also contains information from many public references.

Code to read in file:

consequences = spark.read.parquet('/sbgenomics/project-files/consequences')

#### Columns

* start: long
* reference: string
* alternate: string
* ensembl\_transcript\_id: string
* ensembl\_gene\_id: string
* end: long
* ensembl\_regulatory\_id: string
* feature\_type: string
* consequences: array
* element: string
* name: string
* impact: string
* symbol: string
* strand: integer
* biotype: string
* variant\_class: string
* exon: struct
* rank: integer
* total: integer
* intron: struct
* rank: integer
* total: integer
* hgvsc: string
* hgvsp: string
* hgvsg: string
* cds\_position: integer
* cdna\_position: integer
* protein\_position: integer
* amino\_acids: struct
* reference: string
* variant: string
* codons: struct
* reference: string
* variant: string
* original\_canonical: boolean
* study\_ids: array
* element: string
* aa\_change: string
* coding\_dna\_change: string
* release\_id: string
* mane\_plus: boolean
* mane\_select: boolean
* refseq\_mrna\_id: string
* refseq\_protein\_id: string
* canonical: boolean
* SIFT\_score: double
* SIFT\_pred: string
* SIFT\_converted\_rankscore: double
* Polyphen2\_HDIV\_score: double
* Polyphen2\_HDIV\_pred: string
* Polyphen2\_HDIV\_rankscore: double
* Polyphen2\_HVAR\_score: double
* Polyphen2\_HVAR\_pred: string
* Polyphen2\_HVAR\_rankscore: double
* FATHMM\_score: double
* FATHMM\_pred: string
* FATHMM\_converted\_rankscore: double
* CADD\_raw: double
* CADD\_raw\_rankscore: double
* CADD\_phred: double
* DANN\_score: double
* DANN\_rankscore: double
* REVEL\_rankscore: double
* LRT\_converted\_rankscore: double
* LRT\_pred: string
* phyloP100way\_vertebrate: double
* phyloP100way\_vertebrate\_rankscore: double
* phyloP30way\_mammalian: double
* phyloP30way\_mammalian\_rankscore: double
* phyloP17way\_primate: double
* phyloP17way\_primate\_rankscore: double
* phastCons100way\_vertebrate: double
* phastCons100way\_vertebrate\_rankscore: double
* phastcons30way\_mammalian: double
* phastCons30way\_mammalian\_rankscore: double
* phastCons17way\_primate: double
* phastCons17way\_primate\_rankscore: double
* GERP++\_NR: double
* GERP++\_RS: double
* GERP++\_RS\_rankscore: double
* MutPred\_rankscore: double
* MutPred\_score: double
* MutationAssessor\_pred: string
* MutationAssessor\_score: double
* MutationAssessor\_rankscore: double
* MutationTaster\_converted\_rankscore: double
* PROVEAN\_pred: string
* PROVEAN\_score: double
* PROVEAN\_converted\_rankscore: double
* VEST4\_score: double
* VEST4\_rankscore: double
* MetaSVM\_pred: string
* MetaSVM\_rankscore: double
* MetaSVM\_score: double
* MetaLR\_pred: string
* MetaLR\_rankscore: double
* MetaLR\_score: double
* M-CAP\_pred: string
* M-CAP\_score: double
* M-CAP\_rankscore: double
* MPC\_score: double
* MPC\_rankscore: double
* MVP\_score: double
* MVP\_rankscore: double
* PrimateAI\_pred: string
* PrimateAI\_rankscore: double
* PrimateAI\_score: double
* DEOGEN2\_pred: string
* DEOGEN2\_score: double
* DEOGEN2\_rankscore: double
* BayesDel\_addAF\_pred: string
* BayesDel\_addAF\_rankscore: double
* BayesDel\_addAF\_score: double
* BayesDel\_noAF\_pred: string
* BayesDel\_noAF\_rankscore: double
* BayesDel\_noAF\_score: double
* ClinPred\_pred: string
* ClinPred\_rankscore: double
* ClinPred\_score: double
* LIST-S2\_pred: string
* LIST-S2\_score: double
* LIST-S2\_rankscore: double
* fathmm-MKL\_coding\_pred: string
* fathmm-MKL\_coding\_rankscore: double
* fathmm-MKL\_coding\_score: double
* fathmm-MKL\_coding\_group: string
* fathmm-XF\_coding\_pred: string
* fathmm-XF\_coding\_rankscore: double
* fathmm-XF\_coding\_score: double
* Eigen-PC-phred\_coding: double
* Eigen-PC-raw\_coding: double
* Eigen-PC-raw\_coding\_rankscore: double
* Eigen-phred\_coding: double
* Eigen-raw\_coding: double
* Eigen-raw\_coding\_rankscore: double
* GenoCanyon\_rankscore: double
* GenoCanyon\_score: double
* integrated\_confidence\_value: double
* integrated\_fitCons\_rankscore: double
* integrated\_fitCons\_score: double
* GM12878\_confidence\_value: double
* GM12878\_fitCons\_rankscore: double
* GM12878\_fitCons\_score: double
* H1-hESC\_confidence\_value: double
* H1-hESC\_fitCons\_rankscore: double
* H1-hESC\_fitCons\_score: double
* HUVEC\_confidence\_value: double
* HUVEC\_fitCons\_rankscore: double
* HUVEC\_fitCons\_score: double
* LINSIGHT: double
* LINSIGHT\_rankscore: double
* bStatistic: double
* bStatistic\_converted\_rankscore: double
* Interpro\_domain: string
* GTEx\_V8\_gene: array
* element: string
* GTEx\_V8\_tissue: array
* element: string
* chromosome: string

# Public Tables

## orphanet\_gene\_set

[Open Access]

Description to be added…

Code to read in file:

orphanet\_gene\_set = spark.read.parquet('/sbgenomics/project-files/public/orphanet\_gene\_set')

#### Columns

* orpha\_code: long
* disorder\_id: long
* expert\_link: string
* name: string
* disorder\_type\_id: long
* disorder\_type\_name: string
* disorder\_group\_id: long
* disorder\_group\_name: string
* gene\_source\_of\_validation: string
* gene\_id: long
* gene\_symbol: string
* gene\_name: string
* gene\_synonym\_list: array
  + element: string
* ensembl\_gene\_id: string
* genatlas\_gene\_id: string
* HGNC\_gene\_id: string
* omim\_gene\_id: string
* reactome\_gene\_id: string
* swiss\_prot\_gene\_id: string
* association\_type: string
* association\_type\_id: long
* association\_status: string
* gene\_locus\_id: long
* gene\_locus: string
* gene\_locus\_key: long
* average\_age\_of\_onset: array
  + element: string
* average\_age\_of\_death: array
  + element: string
* type\_of\_inheritance: array
  + element: string

## omim\_gene\_set

[Open Access]

Description to be added…

Code to read in file:

omim\_gene\_set = spark.read.parquet('/sbgenomics/project-files/public/omim\_gene\_set')

#### Columns

* chromosome: string
* start: integer
* end: integer
* cypto\_location: string
* computed\_cypto\_location: string
* omim\_gene\_id: integer
* symbols: array
  + element: string
* name: string
* approved\_symbol: string
* entrez\_gene\_id: integer
* ensembl\_gene\_id: string
* documentation: string
* phenotype: struct
  + name: string
  + omim\_id: string
  + inheritance: array
    - element: string
  + inheritance\_code: array
    - element: string

## human\_genes

[Open Access]

Description to be added…

Code to read in file:

human\_genes = spark.read.parquet('/sbgenomics/project-files/public/human\_genes')

#### Columns

* tax\_id: integer
* entrez\_gene\_id: integer
* symbol: string
* locus\_tag: string
* synonyms: array
  + element: string
* external\_references: map
  + key: string
  + value: string
* chromosome: string
* map\_location: string
* description: string
* type\_of\_gene: string
* symbol\_from\_nomenclature\_authority: string
* full\_name\_from\_nomenclature\_authority: string
* nomenclature\_status: string
* other\_designations: array
  + element: string
* feature\_types: map
  + key: string
  + value: string
* ensembl\_gene\_id: string
* omim\_gene\_id: string

## hpo\_terms

[Open Access]

Description to be added…

Code to read in file:

hpo\_terms = spark.read.parquet('/sbgenomics/project-files/public/hpo\_terms')

#### Columns

* ancestors: array
  + element: struct
    - id: string
    - name: string
    - parents: array
    - element: string
* id: string
* is\_leaf: boolean
* name: string
* parents: array
  + element: string

## hpo\_gene\_set

[Open Access]

Description to be added…

Code to read in file:

hpo\_gene\_set = spark.read.parquet('/sbgenomics/project-files/public/hpo\_gene\_set')

#### Columns

* entrez\_gene\_id: integer
* symbol: string
* hpo\_term\_id: string
* hpo\_term\_name: string
* frequency\_raw: string
* frequency\_hpo: string
* source\_info: string
* source: string
* source\_id: string
* ensembl\_gene\_id: string

## gnomad\_genomes\_3\_1\_1 (\_subset)

[Open Access]

Description to be added…

Code to read in file:

gnomad\_genomes\_3\_1\_1\_subset = spark.read.parquet('/sbgenomics/project-files/public/gnomad\_genomes\_3\_1\_1\_subset')

#### Columns

* start: long
* end: long
* reference: string
* alternate: string
* qual: double
* name: string
* ac: integer
* an: integer
* af: double
* nhomalt: integer
* ac\_raw: integer
* an\_raw: integer
* af\_raw: double
* nhomalt\_raw: integer
* chromosome: string

## dbnsfp

[Open Access]

Description to be added…

* Scores
* Variant

Code to read in file:

dbnsfp\_scores = spark.read.parquet('/sbgenomics/project-files/public/dbnsfp/scores')

dbnsfp\_variant = spark.read.parquet('/sbgenomics/project-files/public/dbnsfp/variant')

#### Columns

##### Scores:

* start: string
* reference: string
* alternate: string
* aaref: string
* symbol: string
* ensembl\_gene\_id: string
* ensembl\_protein\_id: string
* VEP\_canonical: string
* ensembl\_transcript\_id: string
* cds\_strand: string
* SIFT\_score: double
* SIFT\_pred: string
* SIFT\_converted\_rankscore: double
* Polyphen2\_HDIV\_score: double
* Polyphen2\_HDIV\_pred: string
* Polyphen2\_HDIV\_rankscore: double
* Polyphen2\_HVAR\_score: double
* Polyphen2\_HVAR\_pred: string
* Polyphen2\_HVAR\_rankscore: double
* FATHMM\_score: double
* FATHMM\_pred: string
* FATHMM\_converted\_rankscore: double
* CADD\_raw: double
* CADD\_raw\_rankscore: double
* CADD\_phred: double
* DANN\_score: double
* DANN\_rankscore: double
* REVEL\_rankscore: double
* LRT\_converted\_rankscore: double
* LRT\_pred: string
* phyloP100way\_vertebrate: double
* phyloP100way\_vertebrate\_rankscore: double
* phyloP30way\_mammalian: double
* phyloP30way\_mammalian\_rankscore: double
* phyloP17way\_primate: double
* phyloP17way\_primate\_rankscore: double
* phastCons100way\_vertebrate: double
* phastCons100way\_vertebrate\_rankscore: double
* phastcons30way\_mammalian: double
* phastCons30way\_mammalian\_rankscore: double
* phastCons17way\_primate: double
* phastCons17way\_primate\_rankscore: double
* GERP++\_NR: double
* GERP++\_RS: double
* GERP++\_RS\_rankscore: double
* MutPred\_rankscore: double
* MutPred\_score: double
* MutationAssessor\_pred: string
* MutationAssessor\_score: double
* MutationAssessor\_rankscore: double
* MutationTaster\_converted\_rankscore: double
* PROVEAN\_pred: string
* PROVEAN\_score: double
* PROVEAN\_converted\_rankscore: double
* VEST4\_score: double
* VEST4\_rankscore: double
* MetaSVM\_pred: string
* MetaSVM\_rankscore: double
* MetaSVM\_score: double
* MetaLR\_pred: string
* MetaLR\_rankscore: double
* MetaLR\_score: double
* M-CAP\_pred: string
* M-CAP\_score: double
* M-CAP\_rankscore: double
* MPC\_score: double
* MPC\_rankscore: double
* MVP\_score: double
* MVP\_rankscore: double
* PrimateAI\_pred: string
* PrimateAI\_rankscore: double
* PrimateAI\_score: double
* DEOGEN2\_pred: string
* DEOGEN2\_score: double
* DEOGEN2\_rankscore: double
* BayesDel\_addAF\_pred: string
* BayesDel\_addAF\_rankscore: double
* BayesDel\_addAF\_score: double
* BayesDel\_noAF\_pred: string
* BayesDel\_noAF\_rankscore: double
* BayesDel\_noAF\_score: double
* ClinPred\_pred: string
* ClinPred\_rankscore: double
* ClinPred\_score: double
* LIST-S2\_pred: string
* LIST-S2\_score: double
* LIST-S2\_rankscore: double
* fathmm-MKL\_coding\_pred: string
* fathmm-MKL\_coding\_rankscore: double
* fathmm-MKL\_coding\_score: double
* fathmm-MKL\_coding\_group: string
* fathmm-XF\_coding\_pred: string
* fathmm-XF\_coding\_rankscore: double
* fathmm-XF\_coding\_score: double
* Eigen-PC-phred\_coding: double
* Eigen-PC-raw\_coding: double
* Eigen-PC-raw\_coding\_rankscore: double
* Eigen-phred\_coding: double
* Eigen-raw\_coding: double
* Eigen-raw\_coding\_rankscore: double
* GenoCanyon\_rankscore: double
* GenoCanyon\_score: double
* integrated\_confidence\_value: double
* integrated\_fitCons\_rankscore: double
* integrated\_fitCons\_score: double
* GM12878\_confidence\_value: double
* GM12878\_fitCons\_rankscore: double
* GM12878\_fitCons\_score: double
* H1-hESC\_confidence\_value: double
* H1-hESC\_fitCons\_rankscore: double
* H1-hESC\_fitCons\_score: double
* HUVEC\_confidence\_value: double
* HUVEC\_fitCons\_rankscore: double
* HUVEC\_fitCons\_score: double
* LINSIGHT: double
* LINSIGHT\_rankscore: double
* bStatistic: double
* bStatistic\_converted\_rankscore: double
* Interpro\_domain: string
* GTEx\_V8\_gene: array
* element: string
* GTEx\_V8\_tissue: array
* element: string
* chromosome: string

##### Variant:

* start: string
* reference: string
* alternate: string
* aaref: string
* aaalt: string
* rs\_dbSNP151: string
* hg19\_chr: string
* hg19\_pos: string
* hg18\_chr: string
* hg18\_pos: string
* aapos: string
* genename: string
* Ensembl\_geneid: string
* Ensembl\_transcriptid: string
* Ensembl\_proteinid: string
* Uniprot\_acc: string
* Uniprot\_entry: string
* HGVSc\_ANNOVAR: string
* HGVSp\_ANNOVAR: string
* HGVSc\_snpEff: string
* HGVSp\_snpEff: string
* HGVSc\_VEP: string
* HGVSp\_VEP: string
* APPRIS: string
* GENCODE\_basic: string
* TSL: string
* VEP\_canonical: string
* cds\_strand: string
* refcodon: string
* codonpos: string
* codon\_degeneracy: string
* Ancestral\_allele: string
* AltaiNeandertal: string
* Denisova: string
* VindijiaNeandertal: string
* SIFT\_score: string
* SIFT\_converted\_rankscore: string
* SIFT\_pred: string
* SIFT4G\_score: string
* SIFT4G\_converted\_rankscore: string
* SIFT4G\_pred: string
* Polyphen2\_HDIV\_score: string
* Polyphen2\_HDIV\_rankscore: string
* Polyphen2\_HDIV\_pred: string
* Polyphen2\_HVAR\_score: string
* Polyphen2\_HVAR\_rankscore: string
* Polyphen2\_HVAR\_pred: string
* LRT\_score: string
* LRT\_converted\_rankscore: string
* LRT\_pred: string
* LRT\_Omega: string
* MutationTaster\_score: string
* MutationTaster\_converted\_rankscore: string
* MutationTaster\_pred: string
* MutationTaster\_model: string
* MutationTaster\_AAE: string
* MutationAssessor\_score: string
* MutationAssessor\_rankscore: string
* MutationAssessor\_pred: string
* FATHMM\_score: string
* FATHMM\_converted\_rankscore: string
* FATHMM\_pred: string
* PROVEAN\_score: string
* PROVEAN\_converted\_rankscore: string
* PROVEAN\_pred: string
* VEST4\_score: string
* VEST4\_rankscore: string
* MetaSVM\_score: string
* MetaSVM\_rankscore: string
* MetaSVM\_pred: string
* MetaLR\_score: string
* MetaLR\_rankscore: string
* MetaLR\_pred: string
* Reliability\_index: string
* M-CAP\_score: string
* M-CAP\_rankscore: string
* M-CAP\_pred: string
* REVEL\_score: string
* REVEL\_rankscore: string
* MutPred\_score: string
* MutPred\_rankscore: string
* MutPred\_protID: string
* MutPred\_AAchange: string
* MutPred\_Top5features: string
* MVP\_score: string
* MVP\_rankscore: string
* MPC\_score: string
* MPC\_rankscore: string
* PrimateAI\_score: string
* PrimateAI\_rankscore: string
* PrimateAI\_pred: string
* DEOGEN2\_score: string
* DEOGEN2\_rankscore: string
* DEOGEN2\_pred: string
* BayesDel\_addAF\_score: string
* BayesDel\_addAF\_rankscore: string
* BayesDel\_addAF\_pred: string
* BayesDel\_noAF\_score: string
* BayesDel\_noAF\_rankscore: string
* BayesDel\_noAF\_pred: string
* ClinPred\_score: string
* ClinPred\_rankscore: string
* ClinPred\_pred: string
* LIST-S2\_score: string
* LIST-S2\_rankscore: string
* LIST-S2\_pred: string
* Aloft\_Fraction\_transcripts\_affected: string
* Aloft\_prob\_Tolerant: string
* Aloft\_prob\_Recessive: string
* Aloft\_prob\_Dominant: string
* Aloft\_pred: string
* Aloft\_Confidence: string
* CADD\_raw: string
* CADD\_raw\_rankscore: string
* CADD\_phred: string
* CADD\_raw\_hg19: string
* CADD\_raw\_rankscore\_hg19: string
* CADD\_phred\_hg19: string
* DANN\_score: string
* DANN\_rankscore: string
* fathmm-MKL\_coding\_score: string
* fathmm-MKL\_coding\_rankscore: string
* fathmm-MKL\_coding\_pred: string
* fathmm-MKL\_coding\_group: string
* fathmm-XF\_coding\_score: string
* fathmm-XF\_coding\_rankscore: string
* fathmm-XF\_coding\_pred: string
* Eigen-raw\_coding: string
* Eigen-raw\_coding\_rankscore: string
* Eigen-phred\_coding: string
* Eigen-PC-raw\_coding: string
* Eigen-PC-raw\_coding\_rankscore: string
* Eigen-PC-phred\_coding: string
* GenoCanyon\_score: string
* GenoCanyon\_rankscore: string
* integrated\_fitCons\_score: string
* integrated\_fitCons\_rankscore: string
* integrated\_confidence\_value: string
* GM12878\_fitCons\_score: string
* GM12878\_fitCons\_rankscore: string
* GM12878\_confidence\_value: string
* H1-hESC\_fitCons\_score: string
* H1-hESC\_fitCons\_rankscore: string
* H1-hESC\_confidence\_value: string
* HUVEC\_fitCons\_score: string
* HUVEC\_fitCons\_rankscore: string
* HUVEC\_confidence\_value: string
* LINSIGHT: string
* LINSIGHT\_rankscore: string
* GERP++\_NR: string
* GERP++\_RS: string
* GERP++\_RS\_rankscore: string
* phyloP100way\_vertebrate: string
* phyloP100way\_vertebrate\_rankscore: string
* phyloP30way\_mammalian: string
* phyloP30way\_mammalian\_rankscore: string
* phyloP17way\_primate: string
* phyloP17way\_primate\_rankscore: string
* phastCons100way\_vertebrate: string
* phastCons100way\_vertebrate\_rankscore: string
* phastCons30way\_mammalian: string
* phastCons30way\_mammalian\_rankscore: string
* phastCons17way\_primate: string
* phastCons17way\_primate\_rankscore: string
* SiPhy\_29way\_pi: string
* SiPhy\_29way\_logOdds: string
* SiPhy\_29way\_logOdds\_rankscore: string
* bStatistic: string
* bStatistic\_converted\_rankscore: string
* 1000Gp3\_AC: string
* 1000Gp3\_AF: string
* 1000Gp3\_AFR\_AC: string
* 1000Gp3\_AFR\_AF: string
* 1000Gp3\_EUR\_AC: string
* 1000Gp3\_EUR\_AF: string
* 1000Gp3\_AMR\_AC: string
* 1000Gp3\_AMR\_AF: string
* 1000Gp3\_EAS\_AC: string
* 1000Gp3\_EAS\_AF: string
* 1000Gp3\_SAS\_AC: string
* 1000Gp3\_SAS\_AF: string
* TWINSUK\_AC: string
* TWINSUK\_AF: string
* ALSPAC\_AC: string
* ALSPAC\_AF: string
* UK10K\_AC: string
* UK10K\_AF: string
* ESP6500\_AA\_AC: string
* ESP6500\_AA\_AF: string
* ESP6500\_EA\_AC: string
* ESP6500\_EA\_AF: string
* ExAC\_AC: string
* ExAC\_AF: string
* ExAC\_Adj\_AC: string
* ExAC\_Adj\_AF: string
* ExAC\_AFR\_AC: string
* ExAC\_AFR\_AF: string
* ExAC\_AMR\_AC: string
* ExAC\_AMR\_AF: string
* ExAC\_EAS\_AC: string
* ExAC\_EAS\_AF: string
* ExAC\_FIN\_AC: string
* ExAC\_FIN\_AF: string
* ExAC\_NFE\_AC: string
* ExAC\_NFE\_AF: string
* ExAC\_SAS\_AC: string
* ExAC\_SAS\_AF: string
* ExAC\_nonTCGA\_AC: string
* ExAC\_nonTCGA\_AF: string
* ExAC\_nonTCGA\_Adj\_AC: string
* ExAC\_nonTCGA\_Adj\_AF: string
* ExAC\_nonTCGA\_AFR\_AC: string
* ExAC\_nonTCGA\_AFR\_AF: string
* ExAC\_nonTCGA\_AMR\_AC: string
* ExAC\_nonTCGA\_AMR\_AF: string
* ExAC\_nonTCGA\_EAS\_AC: string
* ExAC\_nonTCGA\_EAS\_AF: string
* ExAC\_nonTCGA\_FIN\_AC: string
* ExAC\_nonTCGA\_FIN\_AF: string
* ExAC\_nonTCGA\_NFE\_AC: string
* ExAC\_nonTCGA\_NFE\_AF: string
* ExAC\_nonTCGA\_SAS\_AC: string
* ExAC\_nonTCGA\_SAS\_AF: string
* ExAC\_nonpsych\_AC: string
* ExAC\_nonpsych\_AF: string
* ExAC\_nonpsych\_Adj\_AC: string
* ExAC\_nonpsych\_Adj\_AF: string
* ExAC\_nonpsych\_AFR\_AC: string
* ExAC\_nonpsych\_AFR\_AF: string
* ExAC\_nonpsych\_AMR\_AC: string
* ExAC\_nonpsych\_AMR\_AF: string
* ExAC\_nonpsych\_EAS\_AC: string
* ExAC\_nonpsych\_EAS\_AF: string
* ExAC\_nonpsych\_FIN\_AC: string
* ExAC\_nonpsych\_FIN\_AF: string
* ExAC\_nonpsych\_NFE\_AC: string
* ExAC\_nonpsych\_NFE\_AF: string
* ExAC\_nonpsych\_SAS\_AC: string
* ExAC\_nonpsych\_SAS\_AF: string
* gnomAD\_exomes\_flag: string
* gnomAD\_exomes\_AC: string
* gnomAD\_exomes\_AN: string
* gnomAD\_exomes\_AF: string
* gnomAD\_exomes\_nhomalt: string
* gnomAD\_exomes\_AFR\_AC: string
* gnomAD\_exomes\_AFR\_AN: string
* gnomAD\_exomes\_AFR\_AF: string
* gnomAD\_exomes\_AFR\_nhomalt: string
* gnomAD\_exomes\_AMR\_AC: string
* gnomAD\_exomes\_AMR\_AN: string
* gnomAD\_exomes\_AMR\_AF: string
* gnomAD\_exomes\_AMR\_nhomalt: string
* gnomAD\_exomes\_ASJ\_AC: string
* gnomAD\_exomes\_ASJ\_AN: string
* gnomAD\_exomes\_ASJ\_AF: string
* gnomAD\_exomes\_ASJ\_nhomalt: string
* gnomAD\_exomes\_EAS\_AC: string
* gnomAD\_exomes\_EAS\_AN: string
* gnomAD\_exomes\_EAS\_AF: string
* gnomAD\_exomes\_EAS\_nhomalt: string
* gnomAD\_exomes\_FIN\_AC: string
* gnomAD\_exomes\_FIN\_AN: string
* gnomAD\_exomes\_FIN\_AF: string
* gnomAD\_exomes\_FIN\_nhomalt: string
* gnomAD\_exomes\_NFE\_AC: string
* gnomAD\_exomes\_NFE\_AN: string
* gnomAD\_exomes\_NFE\_AF: string
* gnomAD\_exomes\_NFE\_nhomalt: string
* gnomAD\_exomes\_SAS\_AC: string
* gnomAD\_exomes\_SAS\_AN: string
* gnomAD\_exomes\_SAS\_AF: string
* gnomAD\_exomes\_SAS\_nhomalt: string
* gnomAD\_exomes\_POPMAX\_AC: string
* gnomAD\_exomes\_POPMAX\_AN: string
* gnomAD\_exomes\_POPMAX\_AF: string
* gnomAD\_exomes\_POPMAX\_nhomalt: string
* gnomAD\_exomes\_controls\_AC: string
* gnomAD\_exomes\_controls\_AN: string
* gnomAD\_exomes\_controls\_AF: string
* gnomAD\_exomes\_controls\_nhomalt: string
* gnomAD\_exomes\_controls\_AFR\_AC: string
* gnomAD\_exomes\_controls\_AFR\_AN: string
* gnomAD\_exomes\_controls\_AFR\_AF: string
* gnomAD\_exomes\_controls\_AFR\_nhomalt: string
* gnomAD\_exomes\_controls\_AMR\_AC: string
* gnomAD\_exomes\_controls\_AMR\_AN: string
* gnomAD\_exomes\_controls\_AMR\_AF: string
* gnomAD\_exomes\_controls\_AMR\_nhomalt: string
* gnomAD\_exomes\_controls\_ASJ\_AC: string
* gnomAD\_exomes\_controls\_ASJ\_AN: string
* gnomAD\_exomes\_controls\_ASJ\_AF: string
* gnomAD\_exomes\_controls\_ASJ\_nhomalt: string
* gnomAD\_exomes\_controls\_EAS\_AC: string
* gnomAD\_exomes\_controls\_EAS\_AN: string
* gnomAD\_exomes\_controls\_EAS\_AF: string
* gnomAD\_exomes\_controls\_EAS\_nhomalt: string
* gnomAD\_exomes\_controls\_FIN\_AC: string
* gnomAD\_exomes\_controls\_FIN\_AN: string
* gnomAD\_exomes\_controls\_FIN\_AF: string
* gnomAD\_exomes\_controls\_FIN\_nhomalt: string
* gnomAD\_exomes\_controls\_NFE\_AC: string
* gnomAD\_exomes\_controls\_NFE\_AN: string
* gnomAD\_exomes\_controls\_NFE\_AF: string
* gnomAD\_exomes\_controls\_NFE\_nhomalt: string
* gnomAD\_exomes\_controls\_SAS\_AC: string
* gnomAD\_exomes\_controls\_SAS\_AN: string
* gnomAD\_exomes\_controls\_SAS\_AF: string
* gnomAD\_exomes\_controls\_SAS\_nhomalt: string
* gnomAD\_exomes\_controls\_POPMAX\_AC: string
* gnomAD\_exomes\_controls\_POPMAX\_AN: string
* gnomAD\_exomes\_controls\_POPMAX\_AF: string
* gnomAD\_exomes\_controls\_POPMAX\_nhomalt: string
* gnomAD\_genomes\_flag: string
* gnomAD\_genomes\_AC: string
* gnomAD\_genomes\_AN: string
* gnomAD\_genomes\_AF: string
* gnomAD\_genomes\_nhomalt: string
* gnomAD\_genomes\_AFR\_AC: string
* gnomAD\_genomes\_AFR\_AN: string
* gnomAD\_genomes\_AFR\_AF: string
* gnomAD\_genomes\_AFR\_nhomalt: string
* gnomAD\_genomes\_AMR\_AC: string
* gnomAD\_genomes\_AMR\_AN: string
* gnomAD\_genomes\_AMR\_AF: string
* gnomAD\_genomes\_AMR\_nhomalt: string
* gnomAD\_genomes\_ASJ\_AC: string
* gnomAD\_genomes\_ASJ\_AN: string
* gnomAD\_genomes\_ASJ\_AF: string
* gnomAD\_genomes\_ASJ\_nhomalt: string
* gnomAD\_genomes\_EAS\_AC: string
* gnomAD\_genomes\_EAS\_AN: string
* gnomAD\_genomes\_EAS\_AF: string
* gnomAD\_genomes\_EAS\_nhomalt: string
* gnomAD\_genomes\_FIN\_AC: string
* gnomAD\_genomes\_FIN\_AN: string
* gnomAD\_genomes\_FIN\_AF: string
* gnomAD\_genomes\_FIN\_nhomalt: string
* gnomAD\_genomes\_NFE\_AC: string
* gnomAD\_genomes\_NFE\_AN: string
* gnomAD\_genomes\_NFE\_AF: string
* gnomAD\_genomes\_NFE\_nhomalt: string
* gnomAD\_genomes\_AMI\_AC: string
* gnomAD\_genomes\_AMI\_AN: string
* gnomAD\_genomes\_AMI\_AF: string
* gnomAD\_genomes\_AMI\_nhomalt: string
* gnomAD\_genomes\_SAS\_AC: string
* gnomAD\_genomes\_SAS\_AN: string
* gnomAD\_genomes\_SAS\_AF: string
* gnomAD\_genomes\_SAS\_nhomalt: string
* gnomAD\_genomes\_POPMAX\_AC: string
* gnomAD\_genomes\_POPMAX\_AN: string
* gnomAD\_genomes\_POPMAX\_AF: string
* gnomAD\_genomes\_POPMAX\_nhomalt: string
* clinvar\_id: string
* clinvar\_clnsig: string
* clinvar\_trait: string
* clinvar\_review: string
* clinvar\_hgvs: string
* clinvar\_var\_source: string
* clinvar\_MedGen\_id: string
* clinvar\_OMIM\_id: string
* clinvar\_Orphanet\_id: string
* Interpro\_domain: string
* GTEx\_V8\_gene: string
* GTEx\_V8\_tissue: string
* Geuvadis\_eQTL\_target\_gene: string
* chromosome: string

## gnomad

[Open Access]

Description to be added…

Code to read in file:

gnomad = spark.read.parquet('/sbgenomics/project-files/public/gnomad/gnomad\_genomes\_3.0')

#### Columns

* chromosome: string
* start: integer
* reference: string
* alternate: string
* asj\_female\_ac: integer
* asj\_female\_an: integer
* asj\_female\_af: double
* asj\_female\_hom: integer
* eas\_female\_ac: integer
* eas\_female\_an: integer
* eas\_female\_af: double
* eas\_female\_hom: integer
* afr\_male\_ac: integer
* afr\_male\_an: integer
* afr\_male\_af: double
* afr\_male\_hom: integer
* female\_ac: integer
* female\_an: integer
* female\_af: double
* female\_hom: integer
* fin\_male\_ac: integer
* fin\_male\_an: integer
* fin\_male\_af: double
* fin\_male\_hom: integer
* oth\_female\_ac: integer
* oth\_female\_an: integer
* oth\_female\_af: double
* oth\_female\_hom: integer
* ami\_ac: integer
* ami\_an: integer
* ami\_af: double
* ami\_hom: integer
* oth\_ac: integer
* oth\_an: integer
* oth\_af: double
* oth\_hom: integer
* male\_ac: integer
* male\_an: integer
* male\_af: double
* male\_hom: integer
* ami\_female\_ac: integer
* ami\_female\_an: integer
* ami\_female\_af: double
* ami\_female\_hom: integer
* afr\_ac: integer
* afr\_an: integer
* afr\_af: double
* afr\_hom: integer
* eas\_male\_ac: integer
* eas\_male\_an: integer
* eas\_male\_af: double
* eas\_male\_hom: integer
* sas\_ac: integer
* sas\_an: integer
* sas\_af: double
* sas\_hom: integer
* nfe\_female\_ac: integer
* nfe\_female\_an: integer
* nfe\_female\_af: double
* nfe\_female\_hom: integer
* asj\_male\_ac: integer
* asj\_male\_an: integer
* asj\_male\_af: double
* asj\_male\_hom: integer
* raw\_ac: integer
* raw\_an: integer
* raw\_af: double
* raw\_hom: integer
* oth\_male\_ac: integer
* oth\_male\_an: integer
* oth\_male\_af: double
* oth\_male\_hom: integer
* nfe\_male\_ac: integer
* nfe\_male\_an: integer
* nfe\_male\_af: double
* nfe\_male\_hom: integer
* asj\_ac: integer
* asj\_an: integer
* asj\_af: double
* asj\_hom: integer
* amr\_male\_ac: integer
* amr\_male\_an: integer
* amr\_male\_af: double
* amr\_male\_hom: integer
* ac: integer
* an: integer
* af: double
* hom: integer
* amr\_female\_ac: integer
* amr\_female\_an: integer
* amr\_female\_af: double
* amr\_female\_hom: integer
* sas\_female\_ac: integer
* sas\_female\_an: integer
* sas\_female\_af: double
* sas\_female\_hom: integer
* fin\_ac: integer
* fin\_an: integer
* fin\_af: double
* fin\_hom: integer
* afr\_female\_ac: integer
* afr\_female\_an: integer
* afr\_female\_af: double
* afr\_female\_hom: integer
* sas\_male\_ac: integer
* sas\_male\_an: integer
* sas\_male\_af: double
* sas\_male\_hom: integer
* amr\_ac: integer
* amr\_an: integer
* amr\_af: double
* amr\_hom: integer
* nfe\_ac: integer
* nfe\_an: integer
* nfe\_af: double
* nfe\_hom: integer
* eas\_ac: integer
* eas\_an: integer
* eas\_af: double
* eas\_hom: integer
* ami\_male\_ac: integer
* ami\_male\_an: integer
* ami\_male\_af: double
* ami\_male\_hom: integer
* fin\_female\_ac: integer
* fin\_female\_an: integer
* fin\_female\_af: double
* fin\_female\_hom: integer

## genes

[Open Access]

Description to be added…

Code to read in file:

genes = spark.read.parquet('/sbgenomics/project-files/public/genes')

#### Columns

* chromosome: string
* symbol: string
* entrez\_gene\_id: integer
* omim\_gene\_id: string
* hgnc: string
* ensembl\_gene\_id: string
* location: string
* name: string
* alias: array
* element: string
* biotype: string
* orphanet: array
* element: struct
* disorder\_id: long
* panel: string
* inheritance: array
* element: string
* hpo: array
* element: struct
* hpo\_term\_id: string
* hpo\_term\_name: string
* hpo\_term\_label: string
* omim: array
* element: struct
* name: string
* omim\_id: string
* inheritance: array
* element: string
* inheritance\_code: array
* element: string
* ddd: array
* element: struct
* disease\_name: string
* cosmic: array
* element: struct
* tumour\_types\_germline: array
* element: string

## ensembl\_mapping

[Open Access]

Description to be added…

Code to read in file:

ensembl\_mapping = spark.read.parquet('/sbgenomics/project-files/public/ensembl\_mapping')

#### Columns

* ensembl\_gene\_id: string
* ensembl\_transcript\_id: string
* tags: array
* element: string
* refseq: array
* element: struct
* id: string
* database: string
* entrez: array
* element: struct
* id: string
* database: string
* uniprot: array
* element: struct
* id: string
* database: string
* species: string
* tax\_id: string
* primary\_accessions: array
* element: string
* secondary\_accessions: array
* element: string
* refseq\_mrna\_id: string
* refseq\_protein\_id: string
* is\_canonical: boolean
* is\_mane\_select: boolean
* is\_mane\_plus: boolean
* genome\_build: string
* ensembl\_release\_id: integer

## 1000\_genomes

[Open Access]

Description to be added…

Code to read in file:

thousand\_genomes = spark.read.parquet('/sbgenomics/project-files/public/1000\_genomes')

#### Columns

* chromosome: string
* start: long
* end: long
* name: string
* reference: string
* alternate: string
* ac: integer
* af: double
* an: integer
* afr\_af: double
* eur\_af: double
* sas\_af: double
* amr\_af: double
* eas\_af: double
* dp: integer

## annovar

[Open Access]

Description to be added…

Code to read in file:

annovar = spark.read.parquet('/sbgenomics/project-files/public/annovar')

#### Columns

* start: long
* end: long
* reference: string
* alternate: string
* DamagePredCount: string
* SIFT\_score: float
* SIFT\_converted\_rankscore: float
* SIFT\_pred: string
* SIFT4G\_score: float
* SIFT4G\_converted\_rankscore: float
* SIFT4G\_pred: string
* Polyphen2\_HDIV\_score: float
* Polyphen2\_HDIV\_rankscore: float
* Polyphen2\_HDIV\_pred: string
* Polyphen2\_HVAR\_score: float
* Polyphen2\_HVAR\_rankscore: float
* Polyphen2\_HVAR\_pred: string
* LRT\_score: float
* LRT\_converted\_rankscore: float
* LRT\_pred: string
* MutationTaster\_score: float
* MutationTaster\_converted\_rankscore: float
* MutationTaster\_pred: string
* MutationAssessor\_pred: string
* FATHMM\_score: float
* FATHMM\_converted\_rankscore: float
* FATHMM\_pred: string
* PROVEAN\_score: float
* PROVEAN\_converted\_rankscore: float
* PROVEAN\_pred: string
* VEST4\_score: float
* VEST4\_rankscore: float
* MetaSVM\_score: float
* MetaSVM\_rankscore: float
* MetaSVM\_pred: string
* MetaLR\_score: float
* MetaLR\_rankscore: float
* MetaLR\_pred: string
* MetaRNN\_score: float
* MetaRNN\_rankscore: float
* MetaRNN\_pred: string
* M-CAP\_score: float
* M-CAP\_rankscore: float
* M-CAP\_pred: string
* REVEL\_score: float
* REVEL\_rankscore: float
* MutPred\_score: float
* MutPred\_rankscore: float
* MVP\_score: float
* MVP\_rankscore: float
* MPC\_score: float
* MPC\_rankscore: float
* PrimateAI\_score: float
* PrimateAI\_rankscore: float
* PrimateAI\_pred: string
* DEOGEN2\_score: float
* DEOGEN2\_rankscore: float
* DEOGEN2\_pred: string
* BayesDel\_addAF\_score: float
* BayesDel\_addAF\_rankscore: float
* BayesDel\_addAF\_pred: string
* BayesDel\_noAF\_score: float
* BayesDel\_noAF\_rankscore: float
* BayesDel\_noAF\_pred: string
* ClinPred\_score: float
* ClinPred\_rankscore: float
* ClinPred\_pred: string
* LIST-S2\_score: float
* LIST-S2\_rankscore: float
* LIST-S2\_pred: string
* Aloft\_pred: string
* Aloft\_Confidence: string
* CADD\_raw: float
* CADD\_raw\_rankscore: float
* CADD\_phred: string
* DANN\_score: float
* DANN\_rankscore: float
* fathmm-MKL\_coding\_score: float
* fathmm-MKL\_coding\_rankscore: float
* fathmm-MKL\_coding\_pred: string
* fathmm-XF\_coding\_score: float
* fathmm-XF\_coding\_rankscore: float
* fathmm-XF\_coding\_pred: string
* Eigen-raw\_coding: float
* Eigen-raw\_coding\_rankscore: float
* Eigen-PC-raw\_coding: float
* Eigen-PC-raw\_coding\_rankscore: float
* GenoCanyon\_score: float
* GenoCanyon\_rankscore: float
* integrated\_fitCons\_score: float
* integrated\_fitCons\_rankscore: float
* integrated\_confidence\_value: string
* LINSIGHT: float
* LINSIGHT\_rankscore: float
* GERP++\_NR: float
* GERP++\_RS: float
* GERP++\_RS\_rankscore: float
* phyloP100way\_vertebrate: float
* phyloP100way\_vertebrate\_rankscore: float
* phyloP30way\_mammalian: float
* phyloP30way\_mammalian\_rankscore: float
* phastCons100way\_vertebrate: float
* phastCons100way\_vertebrate\_rankscore: float
* phastCons30way\_mammalian: float
* phastCons30way\_mammalian\_rankscore: float
* SiPhy\_29way\_logOdds: float
* SiPhy\_29way\_logOdds\_rankscore: float
* Interpro\_domain: string
* GTEx\_V8\_gene: array
* element: string
* GTEx\_V8\_tissue: array
* element: string
* TWINSUK\_AC: long
* TWINSUK\_AF: float
* ALSPAC\_AC: long
* ALSPAC\_AF: float
* UK10K\_AC: long
* UK10K\_AF: float
* chromosome: string

## cancer\_hotspots

[Open Access]

Description to be added…

Code to read in file:

cancer\_hotspots = spark.read.parquet('/sbgenomics/project-files/public/cancer\_hotspots')

#### Columns

* chromosome: string
* start: long
* end: long
* hugo\_symbol: string
* entrez\_gene\_id: string
* center: string
* ncbi\_build: string
* strand: string
* variant\_classification: string
* variant\_type: string
* reference\_allele: string
* tumor\_seq\_allele1: string
* tumor\_seq\_allele2: string
* dbsnp\_rs: string
* dbsnp\_val\_status: string
* tumor\_sample\_barcode: string
* matched\_norm\_sample\_barcode: string
* match\_norm\_seq\_allele1: string
* match\_norm\_seq\_allele2: string
* tumor\_validation\_allele1: string
* tumor\_validation\_allele2: string
* match\_norm\_validation\_allele1: string
* match\_norm\_validation\_allele2: string
* verification\_status: string
* validation\_status: string
* mutation\_status: string
* sequencing\_phase: string
* sequence\_source: string
* validation\_method: string
* score: string
* bam\_file: string
* sequencer: string
* tumor\_sample\_uuid: string
* matched\_norm\_sample\_uuid: string
* hgvsc: string
* hgvsp: string
* hgvsp\_short: string
* transcript\_id: string
* exon\_number: string
* t\_depth: string
* t\_ref\_count: string
* t\_alt\_count: string
* n\_depth: string
* n\_ref\_count: string
* n\_alt\_count: string
* all\_effects: string
* allele: string
* gene: string
* feature: string
* feature\_type: string
* consequence: string
* cdna\_position: string
* cds\_position: string
* protein\_position: string
* amino\_acids: string
* codons: string
* existing\_variation: string
* allele\_num: string
* distance: string
* strand\_vep: string
* symbol: string
* symbol\_source: string
* hgnc\_id: string
* biotype: string
* canonical: string
* ccds: string
* ensp: string
* swissprot: string
* trembl: string
* uniparc: string
* refseq: string
* sift: string
* polyphen: string
* exon: string
* intron: string
* domains: string
* clin\_sig: string
* somatic: string
* pubmed: string
* motif\_name: string
* motif\_pos: string
* high\_inf\_pos: string
* motif\_score\_change: string
* impact: string
* pick: string
* variant\_class: string
* tsl: string
* hgvs\_offset: string
* pheno: string
* minimised: string
* gene\_pheno: string
* filter: string
* flanking\_bps: string
* variant\_id: string
* variant\_qual: string
* tumortype: string
* platform: string
* judgement: string
* amino\_acid\_change: string
* amino\_acid\_position: string
* protein\_lenght: string
* reference\_amino\_acid: string
* variant\_amino\_acid: string
* allele\_freq: string
* tm: string
* amino\_acid\_length: string
* ref\_tri: string
* oncotree\_organtype: string
* oncotree\_parent: string
* oncotree\_detailed: string
* master\_id: string

## clinvar

[Open Access]

Description to be added…

Code to read in file:

clinvar = spark.read.parquet('/sbgenomics/project-files/public/clinvar')

#### Columns

* column…

## cosmic\_gene\_set

[Open Access]

Description to be added…

Code to read in file:

cosmic\_gene\_set = spark.read.parquet('/sbgenomics/project-files/public/cosmic\_gene\_set')

#### Columns

* chromosome: string
* start: long
* end: long
* symbol: string
* name: string
* entrez\_gene\_id: string
* tier: integer
* genome\_location: string
* hallmark: boolean
* chr\_band: string
* somatic: boolean
* germline: boolean
* tumour\_types\_somatic: array
* element: string
* tumour\_types\_germline: array
* element: string
* cancer\_syndrome: string
* tissue\_type: array
* element: string
* molecular\_genetics: string
* role\_in\_cancer: array
* element: string
* mutation\_types: array
* element: string
* translocation\_partner: array
* element: string
* other\_germline\_mutation: boolean
* other\_syndrome: array
* element: string
* synonyms: array
* element: string

## ddd\_gene\_set

[Open Access]

Description to be added…

Code to read in file:

ddd\_gene\_set = spark.read.parquet('/sbgenomics/project-files/public/ddd\_gene\_set')

#### Columns

* symbol: string
* omim\_gene\_id: string
* disease\_name: string
* disease\_omim\_id: string
* ddd\_category: string
* mutation\_consequence: string
* phenotypes: array
* element: string
* organ\_specificity: array
* element: string
* panel: string
* hgnc\_id: string

# Kids First Non-Variant Files

These are Kids First generated files to use in conjunction with variants, occurrences, and consequences.

## biospecimens

[Open Access]

Description to be added…

Code to read in file:

biospecimens = spark.read.parquet('/sbgenomics/project-files/dataservice/biospecimens/biospecimens\_re\_000019')

#### Columns

* age\_at\_event\_days: integer
* analyte\_type: string
* composition: string
* concentration\_mg\_per\_ml: double
* consent\_type: string
* created\_at: string
* duo\_ids: array
* element: string
* dbgap\_consent\_code: string
* external\_aliquot\_id: string
* external\_sample\_id: string
* kf\_id: string
* method\_of\_sample\_procurement: string
* modified\_at: string
* ncit\_id\_anatomical\_site: string
* ncit\_id\_tissue\_type: string
* shipment\_date: string
* shipment\_origin: string
* genomic\_files: array
* element: string
* participant\_id: string
* source\_text\_tumor\_descriptor: string
* source\_text\_tissue\_type: string
* source\_text\_anatomical\_site: string
* spatial\_descriptor: string
* uberon\_id\_anatomical\_site: string
* volume\_ul: double
* diagnoses: array
* element: struct
* age\_at\_event\_days: integer
* created\_at: string
* diagnosis\_category: string
* external\_id: string
* icd\_id\_diagnosis: string
* kf\_id: string
* modified\_at: string
* mondo\_id\_diagnosis: string
* participant\_id: string
* source\_text\_diagnosis: string
* uberon\_id\_tumor\_location: string
* source\_text\_tumor\_location: string
* ncit\_id\_diagnosis: string
* spatial\_descriptor: string
* diagnosis\_text: string
* biospecimens: array
* element: string
* visible: boolean
* visible: boolean
* sequencing\_center\_id: string
* sequencing\_center: string
* family\_id: string
* biospecimen\_id: string
* study\_id: string
* release\_id: string

## biospecimens\_diagnoses

[Open Access]

Description to be added…

Code to read in file:

biospecimens\_diagnoses = spark.read.parquet('/sbgenomics/project-files/dataservice/biospecimens\_diagnoses/biospecimens\_diagnoses\_re\_000019')

#### Columns

* kf\_id: string
* biospecimen\_id: string
* diagnosis\_id: string
* visible: boolean
* study\_id: string
* release\_id: string

## diagnoses

[Open Access]

Description to be added…

Code to read in file:

diagnoses = spark.read.parquet('/sbgenomics/project-files/dataservice/diagnoses/diagnoses\_re\_000019')

#### Columns

* age\_at\_event\_days: integer
* created\_at: string
* diagnosis\_category: string
* external\_id: string
* icd\_id\_diagnosis: string
* kf\_id: string
* modified\_at: string
* mondo\_id\_diagnosis: string
* participant\_id: string
* source\_text\_diagnosis: string
* uberon\_id\_tumor\_location: string
* source\_text\_tumor\_location: string
* ncit\_id\_diagnosis: string
* spatial\_descriptor: string
* diagnosis\_text: string
* biospecimens: array
* element: string
* visible: boolean
* study\_id: string
* release\_id: string

## families

[Open Access]

Description to be added…

Code to read in file:

families = spark.read.parquet('/sbgenomics/project-files/dataservice/families/families\_re\_000019')

#### Columns

* kf\_id: string
* modified\_at: string
* created\_at: string
* visible: boolean
* family\_id: string
* participants: array
* element: string
* study\_id: string
* release\_id: string

## family\_relationships

[Open Access]

Description to be added…

Code to read in file:

family\_relationships = spark.read.parquet('/sbgenomics/project-files/dataservice/family\_relationships/family\_relationships\_re\_000019')

#### Columns

* kf\_id: string
* modified\_at: string
* created\_at: string
* participant1: string
* participant2: string
* participant1\_to\_participant2\_relation: string
* participant2\_to\_participant1\_relation: string
* visible: boolean
* study\_id: string
* release\_id: string

## genomic\_files

[Open Access]

Description to be added…

Code to read in file:

genomic\_files = spark.read.parquet('/sbgenomics/project-files/dataservice/genomic\_files/genomic\_files\_re\_000019')

#### Columns

* acl: array
* element: string
* access\_urls: array
* element: string
* availability: string
* controlled\_access: boolean
* created\_at: string
* data\_type: string
* external\_id: string
* file\_format: string
* file\_name: string
* hashes: struct
* md5: string
* instrument\_models: array
* element: string
* is\_harmonized: boolean
* is\_paired\_end: boolean
* kf\_id: string
* latest\_did: string
* metadata: struct
* placeholder: string
* modified\_at: string
* platforms: array
* element: string
* reference\_genome: string
* repository: string
* size: long
* urls: array
* element: string
* visible: boolean
* study\_id: string
* release\_id: string

## investigators

[Open Access]

Description to be added…

Code to read in file:

investigators = spark.read.parquet('/sbgenomics/project-files/dataservice/investigators/investigators\_re\_000019')

#### Columns

* kf\_id: string
* modified\_at: string
* created\_at: string
* institution: string
* name: string
* studies: array
* element: string
* visible: boolean
* study\_id: string
* release\_id: string

## outcomes

[Open Access]

Description to be added…

Code to read in file:

outcomes = spark.read.parquet('/sbgenomics/project-files/dataservice/outcomes/outcomes\_re\_000019')

#### Columns

* kf\_id: string
* modified\_at: string
* created\_at: string
* age\_at\_event\_days: integer
* disease\_related: string
* participant\_id: string
* vital\_status: string
* visible: boolean
* study\_id: string
* release\_id: string

## participants

[Open Access]

Description to be added…

Code to read in file:

participants = spark.read.parquet('/sbgenomics/project-files/dataservice/participants/participants\_re\_000019')

#### Columns

* affected\_status: boolean
* alias\_group: string
* biospecimens: array
* element: string
* created\_at: string
* diagnoses: array
* element: string
* diagnosis\_category: string
* ethnicity: string
* external\_id: string
* family\_id: string
* gender: string
* is\_proband: boolean
* kf\_id: string
* modified\_at: string
* outcomes: array
* element: string
* phenotypes: array
* element: string
* race: string
* study\_id: string
* visible: boolean
* release\_id: string

## phenotypes

[Open Access]

Description to be added…

Code to read in file:

phenotypes = spark.read.parquet('/sbgenomics/project-files/dataservice/phenotypes/phenotypes\_re\_000019')

#### Columns

* kf\_id: string
* modified\_at: string
* created\_at: string
* age\_at\_event\_days: integer
* hpo\_id\_phenotype: string
* observed: string
* participant\_id: string
* source\_text\_phenotype: string
* snomed\_id\_phenotype: string
* external\_id: string
* visible: boolean
* study\_id: string
* release\_id: string

## sequencing\_experiments

[Open Access]

Description to be added…

Code to read in file:

sequencing\_experiments = spark.read.parquet('/sbgenomics/project-files/dataservice/sequencing\_experiments/sequencing\_experiments\_re\_000019')

#### Columns

* kf\_id: string
* created\_at: string
* modified\_at: string
* experiment\_date: string
* experiment\_strategy: string
* center: string
* library\_name: string
* library\_prep: string
* library\_selection: string
* library\_strand: string
* is\_paired\_end: boolean
* platform: string
* instrument\_model: string
* max\_insert\_size: long
* mean\_insert\_size: double
* mean\_depth: double
* total\_reads: long
* mean\_read\_length: double
* external\_id: string
* genomic\_files: array
* element: string
* sequencing\_center\_id: string
* visible: boolean
* study\_id: string
* release\_id: string

## studies

[Open Access]

Description to be added…

Code to read in file:

studies = spark.read.parquet('/sbgenomics/project-files/dataservice/studies/studies\_re\_000019')

#### Columns

* attribution: string
* created\_at: string
* data\_access\_authority: string
* external\_id: string
* kf\_id: string
* modified\_at: string
* name: string
* participants: array
* element: string
* release\_status: string
* study\_files: array
* element: string
* version: string
* short\_name: string
* short\_code: string
* domain: string
* program: string
* visible: boolean
* study\_id: string
* release\_id: string

## study\_files

[Open Access]

Description to be added…

Code to read in file:

study\_files = spark.read.parquet('/sbgenomics/project-files/dataservice/study\_files/study\_files\_re\_000019')

#### Columns

* kf\_id: string
* modified\_at: string
* created\_at: string
* file\_name: string
* study\_id: string
* visible: boolean
* release\_id: string

# Variant Workbench Functions

## Count

To count the number of rows in a dataframe, use the COUNT function.

For example:

row\_count = variants.count()

print(f'The DataFrame has {row\_count} rows.')

## Select

Each file contains many columns, some of which you will not need. To select specific columns of interest, use the SELECT function.

For example:

variants.select('start','reference','alternate','name', 'end','hgvsg','variant\_class','studies','chromosome','clin\_sig','clinvar\_id')

## Drop

Similar to Select, Drop is used to select specific columns that can be removed from the existing data frame.

For example:

variants.drop('clin\_sig','clinvar\_id')

For additional examples, see <https://sparkbyexamples.com/pyspark/pyspark-drop-column-from-dataframe/>

## Filter

Each table can contain millions of rows. In order filter the table to a subset of rows, you can use the FILTER or WHERE functions (functionally equivalent).

For example:

**from** **pyspark.sql** **import** functions **as** F

variants.where(F.array\_contains(variants.studies,"SD\_PREASA7S"))

variants.where(F.col('contigName') == "22")

clinvar.where((F.col('contigName')=="22") & (F.col('INFO\_CLNSIG')=="Pathogenic"))

For additional examples, see <https://sparkbyexamples.com/pyspark/pyspark-where-filter/>

## Join

Often, there is information from multiple tables that you want to combine, usually when you want to filter all the information based on a single column. To do this, you can join two tables by using the JOIN function.

For example:

Combined\_table = variants.join(clinvar,[’column\_to\_join\_on’])

Or, to join on multiple columns, you can do:

cond = ['contigName', 'start','referenceAllele', 'alternateAlleles']

Combined\_table = variants.join(clinvar,[’column\_to\_join\_on’])

For additional examples and explanation, see <https://sparkbyexamples.com/pyspark/pyspark-join-two-or-multiple-dataframes/> and <https://sparkbyexamples.com/pyspark/pyspark-join-multiple-columns/> and

<https://sparkbyexamples.com/pyspark/pyspark-join-explained-with-examples/>

## Show

For a quick way to look at the table, use SHOW. This does not format the table in an easy-to-read manner and is generally only recommended for small tables (few columns). Also, it’s recommended not to output the entire table, but rather the first few rows.

For example:

clinvar.show(n=10)

## printSchema

In order to see the columns and the variable types of a table, one can use the printSchema() function.

For example:

variants.printSchema()

## Convert to Pandas dataframe

In order to easily view a table or to export the table in .csv format, it is necessary to convert an RDD (distributed data frame) into a Pandas data frame. This is a very computationally heavy function and is only recommended one a very small data frame is created (using select, filter, etc.).

To see the first 10 rows of a table, use:

variants.limit(10).toPandas()

To save a file as a single csv, convert to Pandas data frame and then save:

Filtered\_df.toPandas().to\_csv(…)

You can save a partitioned data frame (without converting to Pandas) using pyspark:

Filtered\_df.write.csv(…)

For more on writing files with pyspark, see <https://sparkbyexamples.com/pyspark/pyspark-write-dataframe-to-csv-file/>