Introduction to GEMINI

Aaron Quinlan
University of Utah

quinlanlab.org

Please refer to the following Github Gist to find each command for this session.
Commands should be copy/pasted from this Gist

https://gist.github.com/arq5x/9e1928638397ba45da2e#file-gemini-intro-sh

Genome analysis options

Free:

Write code yourself

VariantTools (complex traits focus)

Exomiser

PLINK/seq

xBrowse

GEMINI

Commercial:

Ingenuity Variant Analysis

GoldenHelix

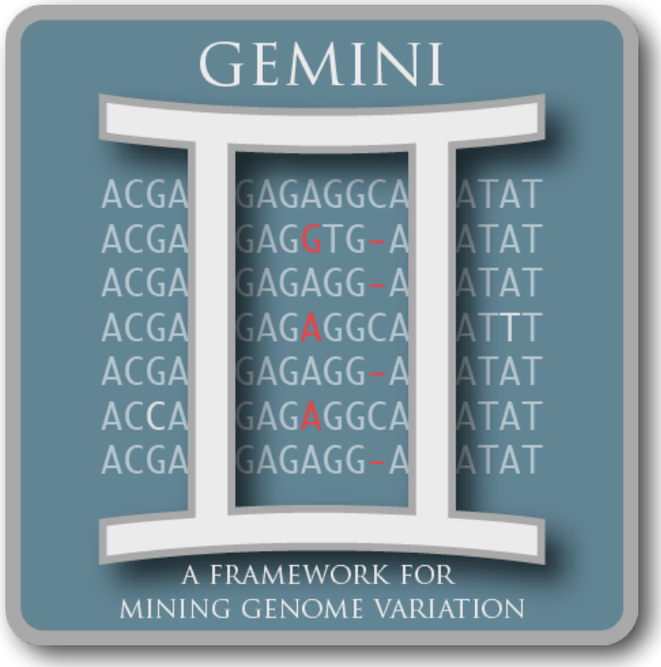
What is GEMINI?

Software package for exploring genetic variation

- Integrates annotations from many different sources (ClinVar, dbSNP, ENCODE, UCSC, 1000 Genomes, ESP, KEGG, etc.)

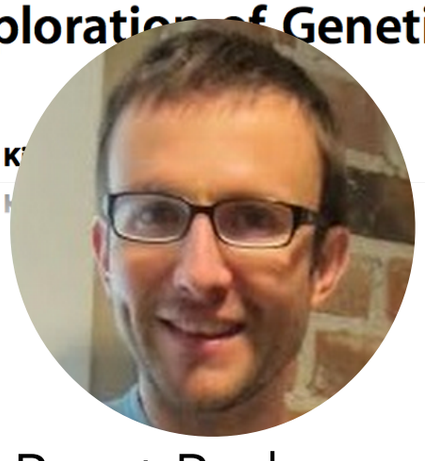
What can you do with Gemini?

- Load a VCF into an "easy to use" database
- Query (fetch data) from database based on annotations or subject genotypes
- Analyze simple genetic models
- More advanced pathway, protein-protein interaction analyses



github.com/arq5x/gemini

Uma Paila



Brent Pedersen

Brad Chapman

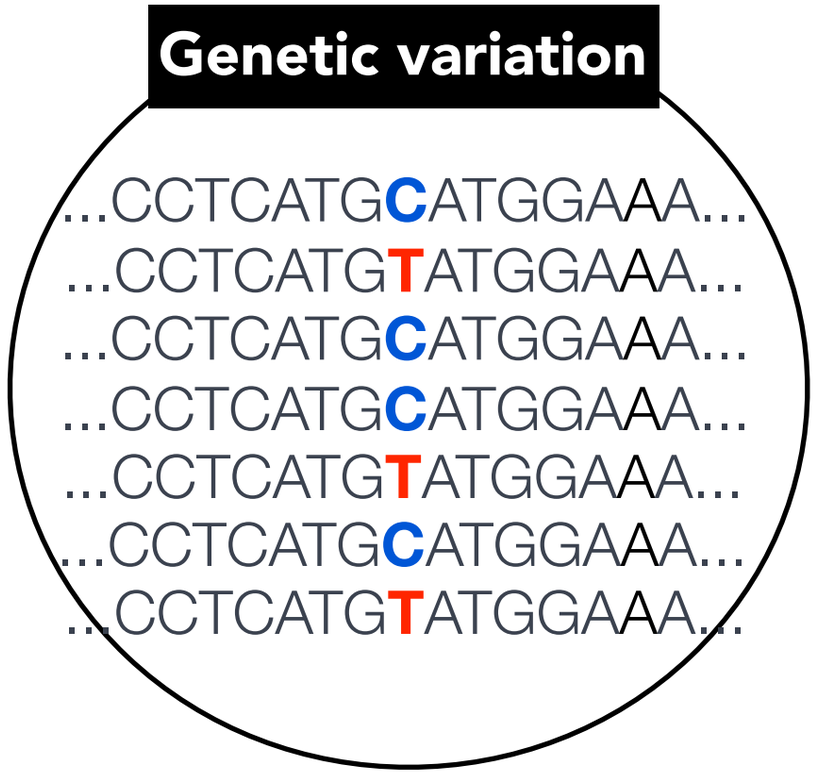
OPEN ACCESS Freely available online

PLOS | COMPUTATIONAL BIOLOGY

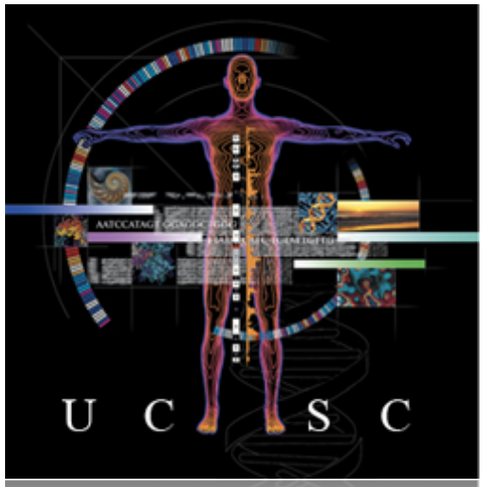
GEMINI: Integrative Exploration of Genetic Variation and Genome Annotations

Umadevi Paila[1], Brad A. Chapman[2], Rory K

Why GEMINI? Annotations provide context

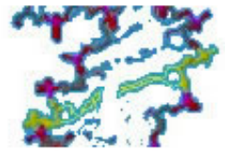


Why GEMINI? Annotations provide context



Conservation
Repeat elements
Genome Gaps
Cytobands
Gene annotations
"Mappability"
DeCIPHER
ISGA

dbSNP
Short Genetic Variations



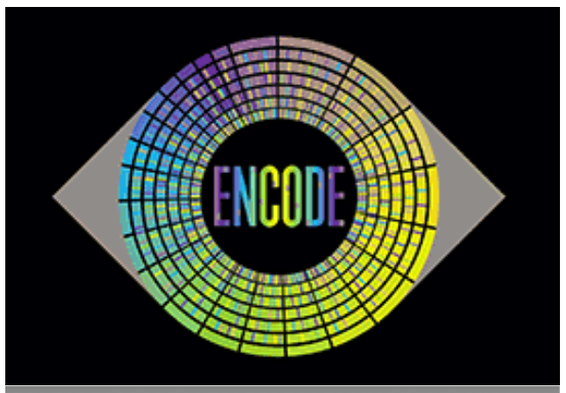
ClinVar

Genetic variation

...CCTCATG**C**ATGGAAA...
...CCTCATG**T**ATGGAAA...
...CCTCATG**C**ATGGAAA...
...CCTCATG**C**ATGGAAA...
...CCTCATG**T**ATGGAAA...
...CCTCATG**C**ATGGAAA...
...CCTCATG**T**ATGGAAA...

OMIM
Online Mendelian Inheritance in Man

1000 Genomes
A Deep Catalog of Human Genetic Variation



Chromatin marks
DNA methylation
RNA expression
TF binding

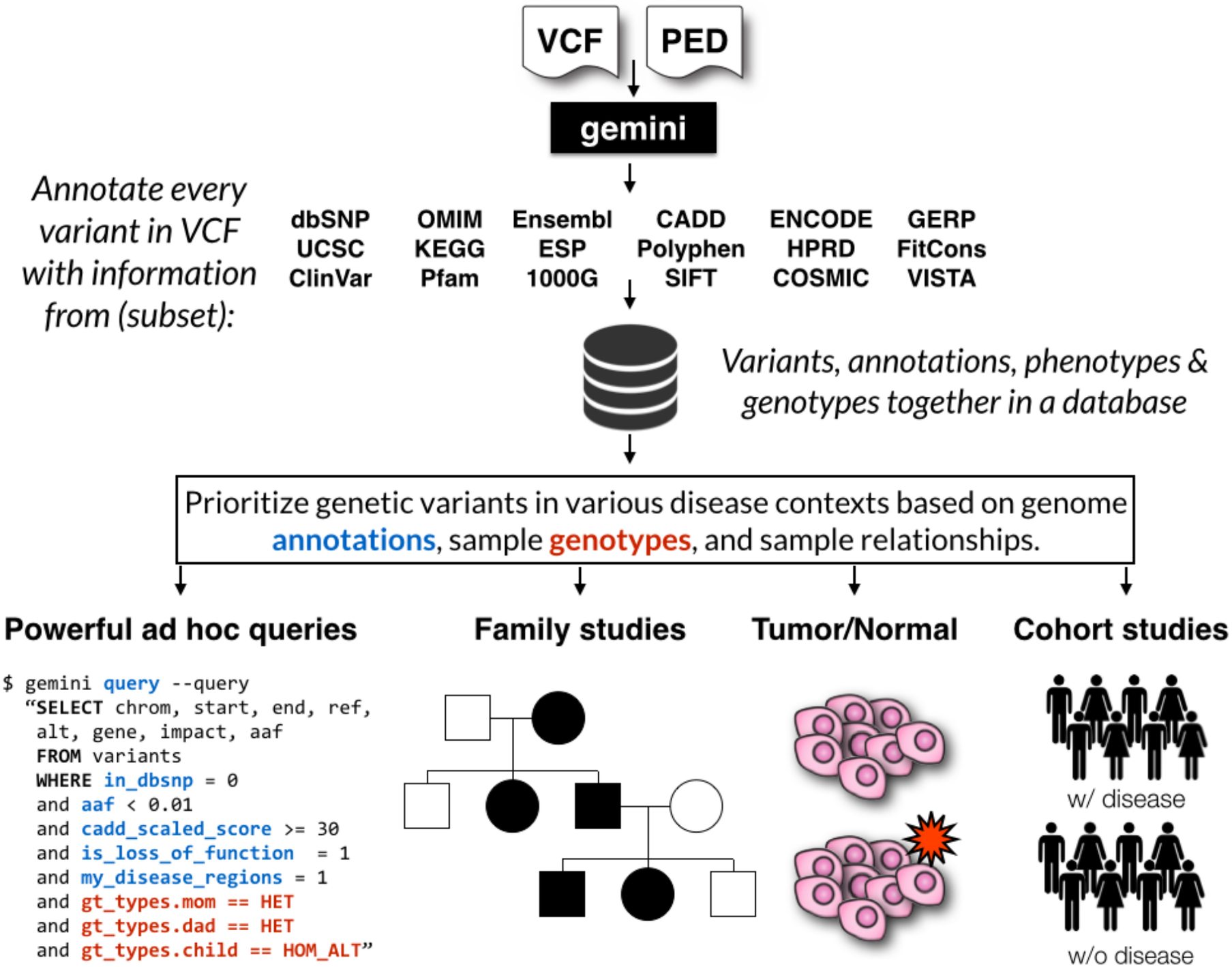
Pfam

KEGG Kyoto Encyclopedia of Genes and Genomes

HGMD®

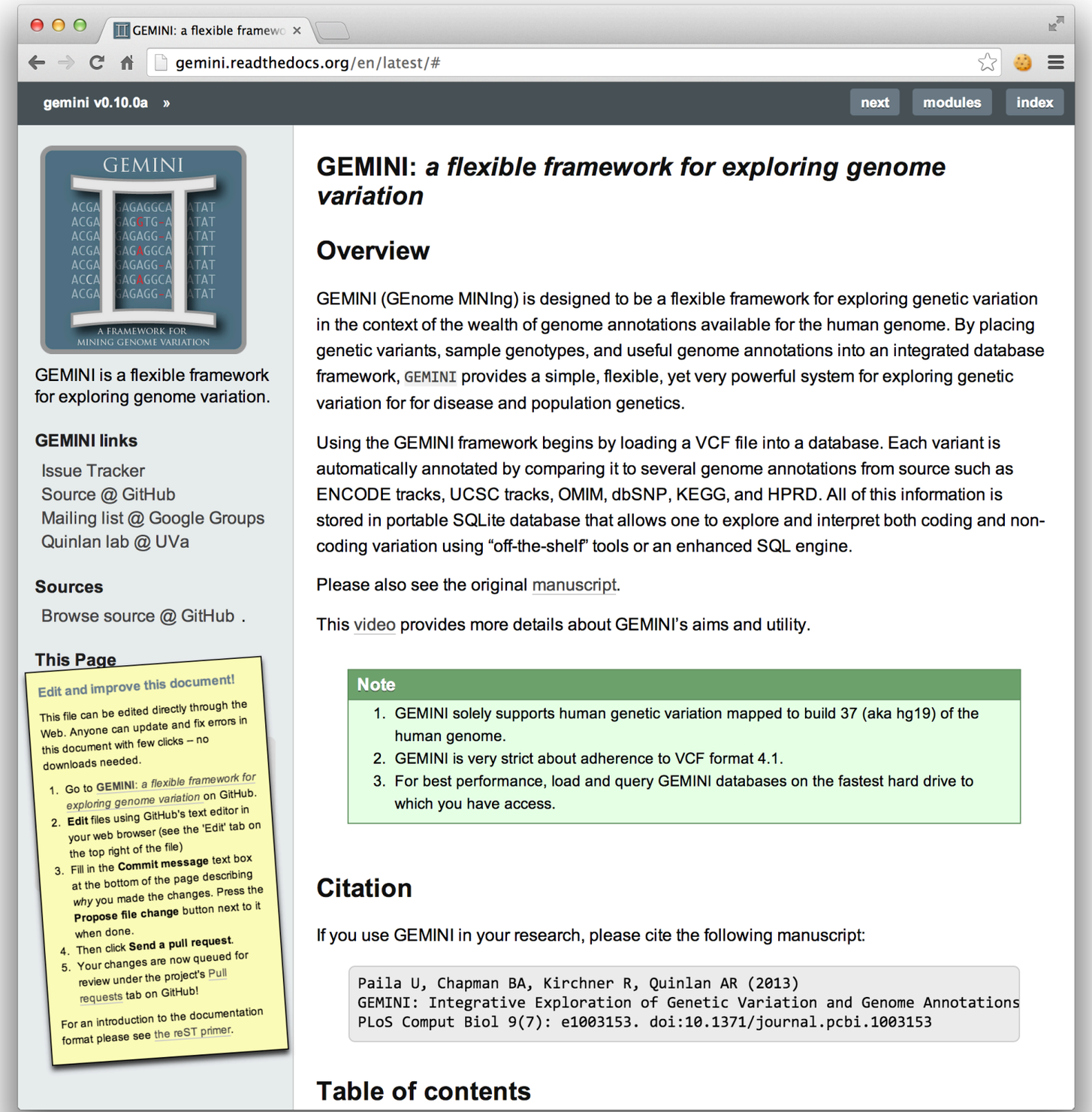
Human Protein Reference Database

GEMINI Framework



GEMINI documentation

http://gemini.readthedocs.org



Data setup: download tutorial files.

```
$ curl https://s3.amazonaws.com/gemini-tutorials/
learnSQL.db > learnSQL.db

$ curl https://s3.amazonaws.com/gemini-tutorials/
learnSQL2.db > learnSQL2.db

$ curl https://s3.amazonaws.com/gemini-tutorials/
chr22.VEP.vcf > chr22.VEP.vcf

$ curl https://s3.amazonaws.com/gemini-tutorials/
trio.ped > trio.ped
```

Note: copy and paste the full commands from the Github Gist to avoid errors injected by PDF conversion

GEMINI uses a database. Uses SQL to "talk" to it.

- SQL databases to organize genotypes and annotations
- SQL = Structured Query Language
- Data stored in tables

samples							
sample_id	name	family_id	paternal_id	maternal_id	sex	phenotype	ethnicity
1	John	1	Bob	Sue	1	2	CEU
2	Bob	1	0	0	1	1	CEU
3	Mary	1	0	0	2	1	CEU
4	Sue	2	0	0	2	2	YRI

- Query (ask the database) to report data matching your requirements
- Can combine queries, act on results from previous query

GEMINI **query** allows one to use SQL to query

samples							
sample_id	name	family_id	paternal_id	maternal_id	sex	phenotype	ethnicity
1	John	1	Bob	Sue	1	2	CEU
2	Bob	1	0	0	1	1	CEU
3	Mary	1	0	0	2	1	CEU
4	Sue	2	0	0	2	2	YRI

```
gemini query -q "SELECT name FROM samples" learnSQL.db
```

John
Bob
Mary
Sue

Filtering rows with the `WHERE (==)` clause

samples							
sample_id	name	family_id	paternal_id	maternal_id	sex	phenotype	ethnicity
1	John	1	Bob	Sue	1	2	CEU
2	Bob	1	0	0	1	1	CEU
3	Mary	1	0	0	2	1	CEU
4	Sue	2	0	0	2	2	YRI

```
gemini query -q "SELECT name FROM samples WHERE phenotype == 2"
                learnSQL.db
```

John
Sue

- phenotype == 2 gets rows where the phenotype value is equal to 2

Filtering rows with the `WHERE (<>)` clause

samples							
sample_id	name	family_id	paternal_id	maternal_id	sex	phenotype	ethnicity
1	John	1	Bob	Sue	1	2	CEU
2	Bob	1	0	0	1	1	CEU
3	Mary	1	0	0	2	1	CEU
4	Sue	2	0	0	2	2	YRI

```
gemini query -q "SELECT name FROM samples WHERE phenotype <> 2"
                learnSQL.db
```

Bob
Mary

- phenotype <> 2 gets rows where the phenotype value is NOT equal to 2

Filtering rows with the **`WHERE (<)`** clause

samples							
sample_id	name	family_id	paternal_id	maternal_id	sex	phenotype	ethnicity
1	John	1	Bob	Sue	1	2	CEU
2	Bob	1	0	0	1	1	CEU
3	Mary	1	0	0	2	1	CEU
4	Sue	2	0	0	2	2	YRI

```
gemini query -q "SELECT name FROM samples WHERE sample_id < 3"
                                                learnSQL.db
```

- Many numeric comparisons possible: <=, >=, >, <

Filtering rows with the **WHERE (NULL)** clause

samples							
sample_id	name	family_id	paternal_id	maternal_id	sex	phenotype	ethnicity
1	John	1	Bob	Sue	1	2	CEU
2	Bob	1	0	0	1	1	CEU
3	Mary	1	0	0	2	1	CEU
4	Sue	2	0	0	2	2	YRI
5	Greg	3	0	0	1	2	

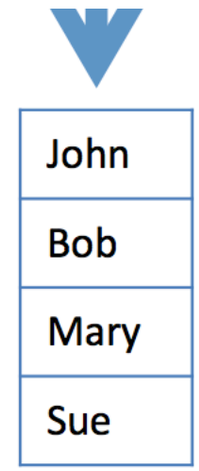
```
gemini query -q "SELECT name FROM samples WHERE ethnicity IS
NULL" learnSQL2.db
```

- When value is missing (empty), it is "NULL"

Filtering rows with the **WHERE (NOT NULL)** clause

samples							
sample_id	**name**	**family_id**	**paternal_id**	**maternal_id**	**sex**	**phenotype**	**ethnicity**
1	John	1	Bob	Sue	1	2	CEU
2	Bob	1	0	0	1	1	CEU
3	Mary	1	0	0	2	1	CEU
4	Sue	2	0	0	2	2	YRI
5	Greg	3	0	0	1	2	

```
gemini query -q "SELECT name FROM samples WHERE ethnicity IS NOT
                NULL" learnSQL2.db
```



The * wildcard

fakevariants				
chrom	**start**	**end**	**rsid**	**in_dbsnp**
chr1	1	2	rs123	1
chr1	3	4		0
chr2	1	2	rs456	1
chr2	3	4		0

```
gemini query -q "SELECT * FROM fakevariants" learnSQL2.db
```

chr1	1	2	rs123	1
chr1	3	4		0
chr2	1	2	rs456	1
chr2	3	4		0

Booleans

- Boolean values are TRUE or FALSE
  - TRUE is equivalent to a value of 1
  - FALSE is equivalent to value of 0

fakevariants				
chrom	**start**	**end**	**rsid**	**in_dbsnp**
chr1	1	2	rs123	1
chr1	3	4		0
chr2	1	2	rs456	1
chr2	3	4		0

Booleans (True)

fakevariants				
chrom	start	end	rsid	in_dbsnp
chr1	1	2	rs123	1
chr1	3	4		0
chr2	1	2	rs456	1
chr2	3	4		0

```
gemini query -q "SELECT chrom,start,end FROM fakevariants
        WHERE in_dbsnp == 1" learnSQL2.db
```

OR

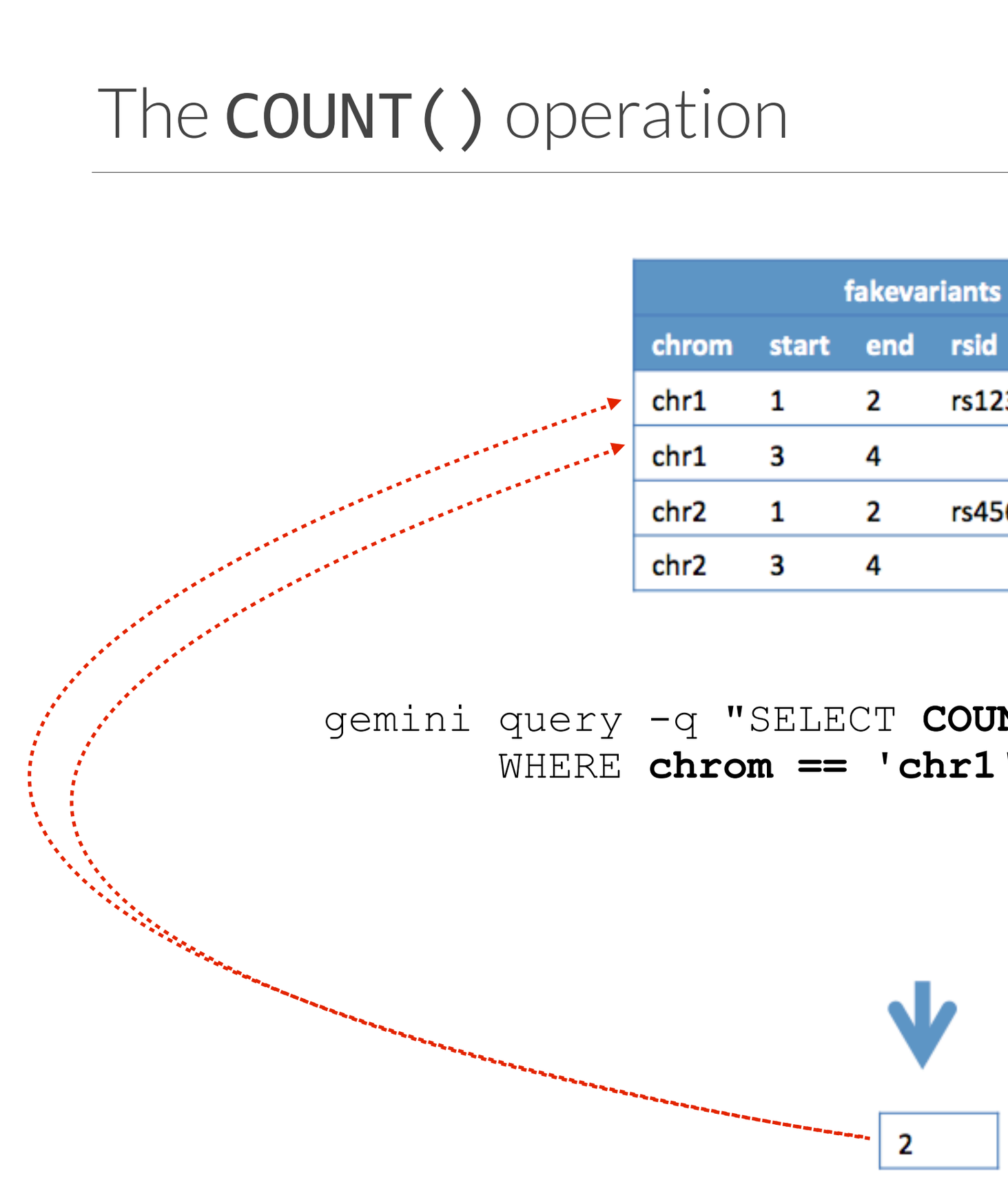
```
gemini query -q "SELECT chrom,start,end FROM fakevariants
        WHERE in_dbsnp" learnSQL2.db
```

chr1	1	2
chr2	1	2

The **COUNT( )** operation

fakevariants				
chrom	**start**	**end**	**rsid**	**in_dbsnp**
chr1	1	2	rs123	1
chr1	3	4		0
chr2	1	2	rs456	1
chr2	3	4		0

```
gemini query -q "SELECT COUNT(*) FROM fakevariants
        WHERE chrom == 'chr1' " learnSQL2.db
```



Multiple **`WHERE`** clauses

fakevariants				
chrom	**start**	**end**	**rsid**	**in_dbsnp**
chr1	1	2	rs123	1
chr1	3	4		0
chr2	1	2	rs456	1
chr2	3	4		0

```
gemini query -q "SELECT COUNT(*) FROM fakevariants
                 WHERE chrom == 'chr1'
                 AND in_dbsnp == 0 " learnSQL2.db
```

1

Using GEMINI

Annotating genetic variants in the VCF file.

- For gene-based annotations, Gemini supports SnpEff and Variant Effect Predictor (VEP)
- Can add custom annotations as well (e.g. SeattleSeq)
- We will use VEP in this session

VEP webpage: http://uswest.ensembl.org/info/docs/variation/vep/vep_script.html
SeattleSeq: http://snp.gs.washington.edu
SNPEff: http://snpeff.sourceforge.net/

Annotation with VEP (done for you; see GEMINI docs)

```
#perl ~/software/variant_effect_predictor/variant_effect_predictor/
variant_effect_predictor.pl -i chr22.vcf -o chr22.VEP.vcf --vcf \
--cache --dir ~/software/variant_effect_predictor/references \
--sift b --polyphen b --symbol --numbers --biotype --total_length \
--fields
Consequence,Codons,Amino_acids,Gene,SYMBOL,Feature,EXON,PolyPhen,SIFT,Protein_pos
ition,BIOTYPE
```

-i chr22.vcf -o chr22.VEP.vcf	Provide names of input and ouput vcfs
--vcf	Create output in vcf format
--cache --dir data/variant_effect_predictor/references	Use a copy of the VEP database stored locally (faster)
--compress "gunzip -c"	Tells VEP how to uncompress vcf file
--terms so	Style for reporting functional consequence of variants
--sift b --polyphen b	Include **both** score and prediction by SIFT and Polyphen
--hgnc	Include HGNC gene name if available
--numbers	Include number of affected exon/intron
--fields Consequence,Codons,Amino_acids,Gene,HGNC,Feature, EXON,PolyPhen,SIFT	Fields to include in output

Annotation with VEP

Before…

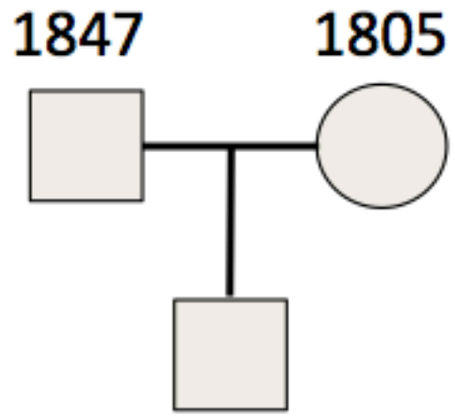
```
22      16157603        .       G       C       382.96  PASS    .       GT:AD:DP:GQ:PL  0/1:4,18:22:0.32:370,0,0         0/1:5,2:7:17.67:46,0,18 ./.:.:.:.:.
22      16157635        .       G       A       15.67   LowQual;QDFilter;QUALFilter     .       GT:AD:DP:GQ:PL  0/0:5,5:10:3:0,3,27      0/1:7,5:12:20.61:47,0,21        ./.:.:.:.:.
22      16157883        .       T       C       104.13  ABFilter;QDFilter       .       GT:AD:DP:GQ:PL  0/1:52,14:66:99:137,0,1177       0/0:88,3:91:99:0,182,2160       ./.:.:.:.:.
22      16159060        .       G       A       11777.6 PASS    .       GT:AD:DP:GQ:PL  1/1:0,49:49:99:1865,144,0       1/1:0,87:87:99:3311,262,0       1/1:3,192:195:99:6602,490,0
22      16255645        .       T       C       235.22  ABFilter;QDFilter       .       GT:AD:DP:GQ:PL  0/1:30,6:36:20.52:155,0,21      0/1:27,7:34:99:119,0,182        0/0:246,1:250:99:0,183,2040
```

After…

```
22      16157603        .       G       C       382.96  PASS    CSQ=C|ENSG00000232775|ENST00000609679|Transcript|intron_variant&nc_transcript_variant|-/552|||||||1/2||1|AP000525.10|Clone_bas
ed_vega_gene|||processed_transcript,C|ENSG00000206195|ENST00000437781|Transcript|intron_variant&nc_transcript_variant|-/2398|||||||7/7||-1|AP000525.9|Clone_based_vega_gene|||lincRNA,C|ENSG00
000272872|ENST00000608286|Transcript|upstream_gene_variant|-/694|||||||||2837|-1|LL22NC03-N14H11.1|Clone_based_vega_gene|||sense_intronic,C|ENSG00000206195|ENST00000383038|Transcript|downstre
am_gene_variant|-/1394|||||||||1195|-1|AP000525.9|Clone_based_vega_gene|||processed_transcript,C|ENSG00000206195|ENST00000447898|Transcript|intron_variant&nc_transcript_variant|-/4136|||||||3
/3||-1|AP000525.9|Clone_based_vega_gene|||lincRNA,C|ENSG00000206195|ENST00000607933|Transcript|downstream_gene_variant|-/642|||||||||1226|-1|AP000525.9|Clone_based_vega_gene|||processed_trans
cript,C|ENSG00000232775|ENST00000440946|Transcript|upstream_gene_variant|-/1160|||||||||4463|1|AP000525.10|Clone_based_vega_gene|||processed_transcript,C|ENSG00000206195|ENST00000413768|Trans
cript|intron_variant&nc_transcript_variant|-/2102|||||||7/7||-1|AP000525.9|Clone_based_vega_gene|||lincRNA      GT:AD:DP:GQ:PL  0/1:4,18:22:0.32:370,0,0         0/1:5,2:7:17.67:46,0,18 ./.:.:
.:.:.
22      16157635        .       G       A       15.67   LowQual;QDFilter;QUALFilter     CSQ=A|ENSG00000232775|ENST00000609679|Transcript|intron_variant&nc_transcript_variant|-/552|||||||1/2|
|1|AP000525.10|Clone_based_vega_gene|||processed_transcript,A|ENSG00000206195|ENST00000437781|Transcript|intron_variant&nc_transcript_variant|-/2398|||||||7/7||-1|AP000525.9|Clone_based_vega
_gene|||lincRNA,A|ENSG00000272872|ENST00000608286|Transcript|upstream_gene_variant|-/694|||||||||2869|-1|LL22NC03-N14H11.1|Clone_based_vega_gene|||sense_intronic,A|ENSG00000206195|ENST0000038
3038|Transcript|downstream_gene_variant|-/1394|||||||||1163|-1|AP000525.9|Clone_based_vega_gene|||processed_transcript,A|ENSG00000206195|ENST00000447898|Transcript|intron_variant&nc_transcrip
t_variant|-/4136|||||||3/3||-1|AP000525.9|Clone_based_vega_gene|||lincRNA,A|ENSG00000206195|ENST00000607933|Transcript|downstream_gene_variant|-/642|||||||||1194|-1|AP000525.9|Clone_based_veg
a_gene|||processed_transcript,A|ENSG00000232775|ENST00000440946|Transcript|upstream_gene_variant|-/1160|||||||||4431|1|AP000525.10|Clone_based_vega_gene|||processed_transcript,A|ENSG000002061
95|ENST00000413768|Transcript|intron_variant&nc_transcript_variant|-/2102|||||||7/7||-1|AP000525.9|Clone_based_vega_gene|||lincRNA      GT:AD:DP:GQ:PL  0/0:5,5:10:3:0,3,27      0/1:7,5:12:20.
61:47,0,21      ./.:.:.:.:.
22      16157883        .       T       C       104.13  ABFilter;QDFilter       CSQ=C|ENSG00000232775|ENST00000609679|Transcript|intron_variant&nc_transcript_variant|-/552|||||||1/2||1|AP000
525.10|Clone_based_vega_gene|||processed_transcript,C|ENSG00000206195|ENST00000437781|Transcript|intron_variant&nc_transcript_variant|-/2398|||||||7/7||-1|AP000525.9|Clone_based_vega_gene|||
lincRNA,C|ENSG00000272872|ENST00000608286|Transcript|upstream_gene_variant|-/694|||||||||3117|-1|LL22NC03-N14H11.1|Clone_based_vega_gene|||sense_intronic,C|ENSG00000206195|ENST00000383038|Tra
nscript|downstream_gene_variant|-/1394|||||||||915|-1|AP000525.9|Clone_based_vega_gene|||processed_transcript,C|ENSG00000206195|ENST00000447898|Transcript|intron_variant&nc_transcript_variant
|-/4136|||||||3/3||-1|AP000525.9|Clone_based_vega_gene|||lincRNA,C|ENSG00000206195|ENST00000607933|Transcript|downstream_gene_variant|-/642|||||||||946|-1|AP000525.9|Clone_based_vega_gene|||p
rocessed_transcript,C|ENSG00000232775|ENST00000440946|Transcript|upstream_gene_variant|-/1160|||||||||4183|1|AP000525.10|Clone_based_vega_gene|||processed_transcript,C|ENSG00000206195|ENST000
00413768|Transcript|intron_variant&nc_transcript_variant|-/2102|||||||7/7||-1|AP000525.9|Clone_based_vega_gene|||lincRNA      GT:AD:DP:GQ:PL  0/1:52,14:66:99:137,0,1177      0/0:88,3:91:99
:0,182,2160     ./.:.:.:.:.
22      16159060        .       G       A       11777.6 PASS    CSQ=A|ENSG00000232775|ENST00000609679|Transcript|intron_variant&nc_transcript_variant|-/552|||||||1/2||1|AP000525.10|Clone_bas
ed_vega_gene|||processed_transcript,A|ENSG00000206195|ENST00000437781|Transcript|intron_variant&nc_transcript_variant|-/2398|||||||7/7||-1|AP000525.9|Clone_based_vega_gene|||lincRNA,A|ENSG00
000272872|ENST00000608286|Transcript|upstream_gene_variant|-/694|||||||||4294|-1|LL22NC03-N14H11.1|Clone_based_vega_gene|||sense_intronic,A|ENSG00000206195|ENST00000383038|Transcript|non_codi
ng_exon_variant&nc_transcript_variant|1132/1394||||||8/8|||-1|AP000525.9|Clone_based_vega_gene|||processed_transcript,A|ENSG00000206195|ENST00000447898|Transcript|intron_variant&nc_transcrip
t_variant|-/4136|||||||3/3||-1|AP000525.9|Clone_based_vega_gene|||lincRNA,A|ENSG00000206195|ENST00000607933|Transcript|non_coding_exon_variant&nc_transcript_variant|411/642||||||1/1|||-1|AP0
00525.9|Clone_based_vega_gene|||processed_transcript,A|ENSG00000232775|ENST00000440946|Transcript|upstream_gene_variant|-/1160|||||||||3006|1|AP000525.10|Clone_based_vega_gene|||processed_tra
nscript,A|ENSG00000206195|ENST00000413768|Transcript|intron_variant&nc_transcript_variant|-/2102|||||||7/7||-1|AP000525.9|Clone_based_vega_gene|||lincRNA      GT:AD:DP:GQ:PL  1/1:0,49:49:99
:1865,144,0     1/1:0,87:87:99:3311,262,0       1/1:3,192:195:99:6602,490,0
22      16255645        .       T       C       235.22  ABFilter;QDFilter       CSQ=C|ENSG00000241838|ENST00000417657|Transcript|non_coding_exon_variant&nc_transcript_variant|833/1123||||||1
/1|||-1|LA16c-3G11.7|Clone_based_vega_gene|||processed_pseudogene,C|ENSG00000198062|ENST00000452800|Transcript|downstream_gene_variant|-/1964|-/912|-/303||||||796|-1|POTEH|HGNC|||nonsense_me
diated_decay,C|ENSG00000198062|ENST00000343518|Transcript|downstream_gene_variant|-/1928|-/1638|-/545||||||796|-1|POTEH|HGNC|||protein_coding,C|ENSG00000241838|ENST00000339523|Transcript|dow
nstream_gene_variant|-/309|||||||||274|-1|LA16c-3G11.7|Clone_based_vega_gene|||processed_pseudogene      GT:AD:DP:GQ:PL  0/1:30,6:36:20.52:155,0,21      0/1:27,7:34:99:119,0,182        0/0:24
6,1:250:99:0,183,2040
```

Creating PED files.

- VCF contains genotypes and annotations
- PED file provides family/pedigree, sex, phenotype information
- missing values = 0 or -9



Family ID	Subject ID	Father ID	Mother ID	Sex 1 = male 2 = female	Phenotype 1 = unaffected 2 = affected	Ancestry
1	4805	1847	1805	2	0	CEU
1	1847	0	0	1	0	CEU
1	1805	0	0	2	0	CEU

Loading a VCF file into a GEMINI database

```
gemini load -v chr22.VEP.vcf \
            -p trio.ped \
            -t VEP \
            --cores 4 \
            --skip-gene-tables \
        chr22.db
```

- Use Gemini to load annotated VCF and associated PED file into chr22.db
- Also load per-position GERP scores
- Use 4 cores
  - optional
  - faster, but most laptops have only 2 cores

What is in the database?

```
gemini db_info chr22.db
```

- `table_name` column shows whether information stored applies to:
  - `variants`
  - `variant_impacts`
  - `samples`
- `types` column shows the type of information
  - `text` - just plain text (e.g. "indel" or "SNP")
  - `integer` - a whole number (e.g. "start" position)
  - `float` - a number with decimal places (e.g. "call_rate")
  - `blob` - special data type interpreted by Gemini (genotype data)
  - `bool` - can be true or false (e.g. "in_dbsnp")

Full database details

http://gemini.readthedocs.org/en/latest/content/database_schema.html

Queries of the samples in the database

Which samples/subjects are in your database?

```
gemini query -q "SELECT name FROM samples" --header chr22.db
```

Does the rest of the info match your PED file?

```
gemini query -q "SELECT * FROM samples" --header chr22.db
```

Querying variants. ***Basics***.

How many novel (i.e., not in dbSNP) are observed in these samples?

```
gemini query -q "SELECT COUNT(*) \
                 FROM variants \
                 WHERE in_dbsnp == 0" --header chr22.db
```

How many variants passed GATK filters?

```
gemini query -q "SELECT COUNT(*) \
                 FROM variants \
                 WHERE filter is NULL" --header chr22.db
```

Querying variants. ***Basics***.

Let's examine variants with GATK filter PASS in the MLC1 gene

```
gemini query -q "SELECT * FROM variants WHERE
filter is NULL and gene = 'MLC1' " --header chr22.db
```

Let's instead focus the analysis to a specific set of columns

```
gemini query -q "SELECT rs_ids, aaf_esp_ea, impact,
clinvar_disease_name, clinvar_sig
FROM variants
WHERE filter is NULL and gene = 'MLC1' " --header chr22.db
```

Querying variants. *Basics*.

How many variants are rare and in a disease-associated gene?

```
gemini query -q "SELECT COUNT(*) from variants WHERE
clinvar_disease_name is not NULL and aaf_esp_ea <= 0.01" \
chr22.db
```

List the genes

```
gemini query -q "SELECT gene from variants \
WHERE clinvar_disease_name is not NULL and aaf_esp_ea <= 0.01" \
chr22.db
```

Querying variants. ***Sample genotypes***.

For each individual, Gemini gives access to genotype, depth, genotype quality and genotype likelihoods at each variant

`gt_types.subjectID`
  `HOM_REF`
  `HET`
  `HOM_ALT`

`gt_quals.subjectID`
  genotype quality

`gt_depths.subjectID`
  total number of reads in this subject at position

`gt_ref_depths.subjectID`
  number of **reference allele** reads in this subject at position

`gt_alt_depths.subjectID`
  number of **alternate allele** reads in this subject at position

Querying variants. ***Sample genotypes queries.***

At how many sites does subject 1805 have a non-reference allele?

```
gemini query -q "SELECT * from variants" \
             --gt-filter "gt_types.1805 <> HOM_REF" \
             --header \
             chr22.db \
| wc -l
```

At how many sites do subject 1805 ***and*** subject 4805 both have a non-reference allele?

```
gemini query -q "SELECT * from variants" \
             --gt-filter "(gt_types.1805 <> HOM_REF AND \
                          gt_types.4805 <> HOM_REF)" \
             chr22.db \
| wc -l
```

List the genotypes for sample 1805 and 4805

```
gemini query -q "SELECT gts.1805, gts.4805 from variants" \
             --gt-filter "(gt_types.1805 <> HOM_REF and \
                          gt_types.4805 <> HOM_REF)" \
             chr22.db
```

Querying variants. ***Sample genotypes wildcards.***

At which variants are every sample heterozygous?

```
gemini query -q "SELECT chrom, start, end, ref, alt, \
                       gene, impact, (gts).(*) \
                  FROM variants" \
               --gt-filter "(gt_types).(*).(==HET).(all)" \
               --header \
               chr22.db
```

At which variants are all of the female samples reference homozygotes?

```
gemini query -q "SELECT chrom, start, end, ref, alt, \
                       gene, impact, (gts).(*) \
                  FROM variants" \
               --gt-filter "(gt_types).(sex==2).(==HOM_REF).(all)" \
               --header \
               chr22.db
```

Note

The syntax of the wildcard `--gt-filters` is `(COLUMN).(SAMPLE_WILDCARD).(SAMPLE_WILDCARD_RULE).(RULE_ENFORCEMENT).`

Querying variants. *The "any" wildcard*

At which variants is ***any female*** sample homozygous for the alternate allele?

```
gemini query -q "SELECT chrom, start, end, ref, alt, \
                       gene, impact, (gts).(*) \
                   FROM variants" \
                --gt-filter "(gt_types).(sex==2).(!=HOM_REF).(any)" \
                --header \
                chr22.db
```

Querying variants. *The "none" wildcard*

At which variants are ***none of the female*** samples homozygous for the reference allele?

```
gemini query -q "SELECT chrom, start, end, ref, alt, \
                       gene, impact, (gts).(*) \
                    FROM variants" \
                --gt-filter "(gt_types).(sex==2).(==HOM_REF).(none)" \
                --header \
                chr22.db
```

Querying variants. *The "count" wildcard*

Identify suspicious variants. Cases where at least 2 of the samples have UNKNOWN genotypes

```
gemini query -q "SELECT chrom, start, end, ref, alt, \
                       gene, impact, (gts).(*) \
                   FROM variants" \
                --gt-filter "(gt_types).(*).(==UNKNOWN).(count >= 2)" \
                --header \
                chr22.db
```

Wildcards work on all of the "genotype" columns in GEMINI

Genotype information

gts	BLOB	A compressed binary vector of sample genotypes (e.g., "A/A", "A|G", "G/G") - Extracted from the VCF `GT` genotype tag.
gt_types	BLOB	A compressed binary vector of numeric genotype "types" (e.g., 0, 1, 2) - Inferred from the VCF `GT` genotype tag.
gt_phases	BLOB	A compressed binary vector of sample genotype phases (e.g., False, True, False) - Extracted from the VCF `GT` genotype tag's allele delimiter e.g., `A/G` means an unphased genotype. Value is **FALSE**. e.g., `A|G` means a phased genotype. Value is **TRUE**.
gt_depths	BLOB	A compressed binary vector of the depth of aligned sequence observed for each sample - Extracted from the VCF `DP` genotype tag.
gt_ref_depths	BLOB	A compressed binary vector of the depth of reference alleles observed for each sample - Extracted from the VCF `AD` genotype tag.
gt_alt_depths	BLOB	A compressed binary vector of the depth of alternate alleles observed for each sample - Extracted from the VCF `AD` genotype tag.
gt_quals	BLOB	A compressed binary vector of the genotype quality (PHRED scale) estimates for each sample - Extracted from the VCF `GQ` genotype tag.

http://gemini.readthedocs.org/en/latest/content/database_schema.html#genotype-information

Wildcards can be applied to other genotype columns

Identify variants that are likely to have high quality genotypes
(i.e., aligned depth >=50 for all samples)

```
gemini query -q "SELECT chrom, start, end, ref, alt, \
                       gene, impact, (gts).(*), (gt_depths).(*) \
                FROM variants" \
            --gt-filter "(gt_depths).(*).(>=50).(all)" \
            --header \
            chr22.db
```

Variant statistics

Get some basic statistics on variants in samples

```
gemini stats --gts-by-sample chr22.db | column -t
```

```
sample  num_hom_ref  num_het  num_hom_alt  num_unknown  total
1805    860          1031     496          58           2445
1847    676          1297     418          54           2445
4805    662          1242     478          63           2445
```

Calculate transition/transversion ratio

```
gemini stats --tstv chr22.db | column -t
```

```
ts    tv   ts/tv
1594  698  2.2837
```

Variant statistics --summarize

Add "**`--summarize`**" to summarize genotypes by sample for any custom query

```
gemini stats --summarize \
"SELECT * from variants WHERE in_dbsnp = 0" \
chr22.db | column -t
```

```
sample  total  num_het  num_hom_alt
1805    85     73       12
1847    94     75       19
4805    168    148      20
```

```
gemini stats --summarize \
"SELECT * from variants WHERE in_dbsnp = 1" \
chr22.db | column -t
```

```
sample  total  num_het  num_hom_alt
1805    1442   958      484
1847    1621   1222     399
4805    1552   1094     458
```

References

Resources mentioned in these slides:

VEP webpage:

http://uswest.ensembl.org/info/docs/variation/vep/vep_script.html

SeattleSeq:

http://snp.gs.washington.edu

SNPEff:

http://snpeff.sourceforge.net/

Gemini Documentation:

https://gemini.readthedocs.org

Annotations and information available for Gemini:

https://gemini.readthedocs.org/en/latest/content/database_schema.html

To learn more about SQL on your own:

http://software-carpentry.org/4_0/databases/