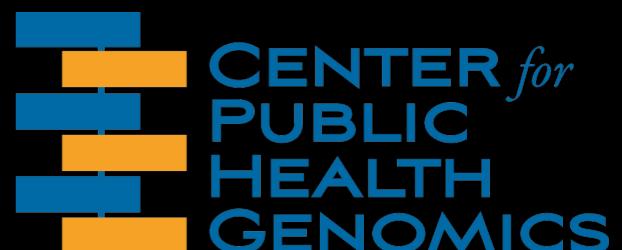


GEMINI: Exploring genetic variation with a tour guide.

Aaron Quinlan

Public Health Sciences
Center for Public Health Genomics

quinlanlab.org
@aaronquinlan



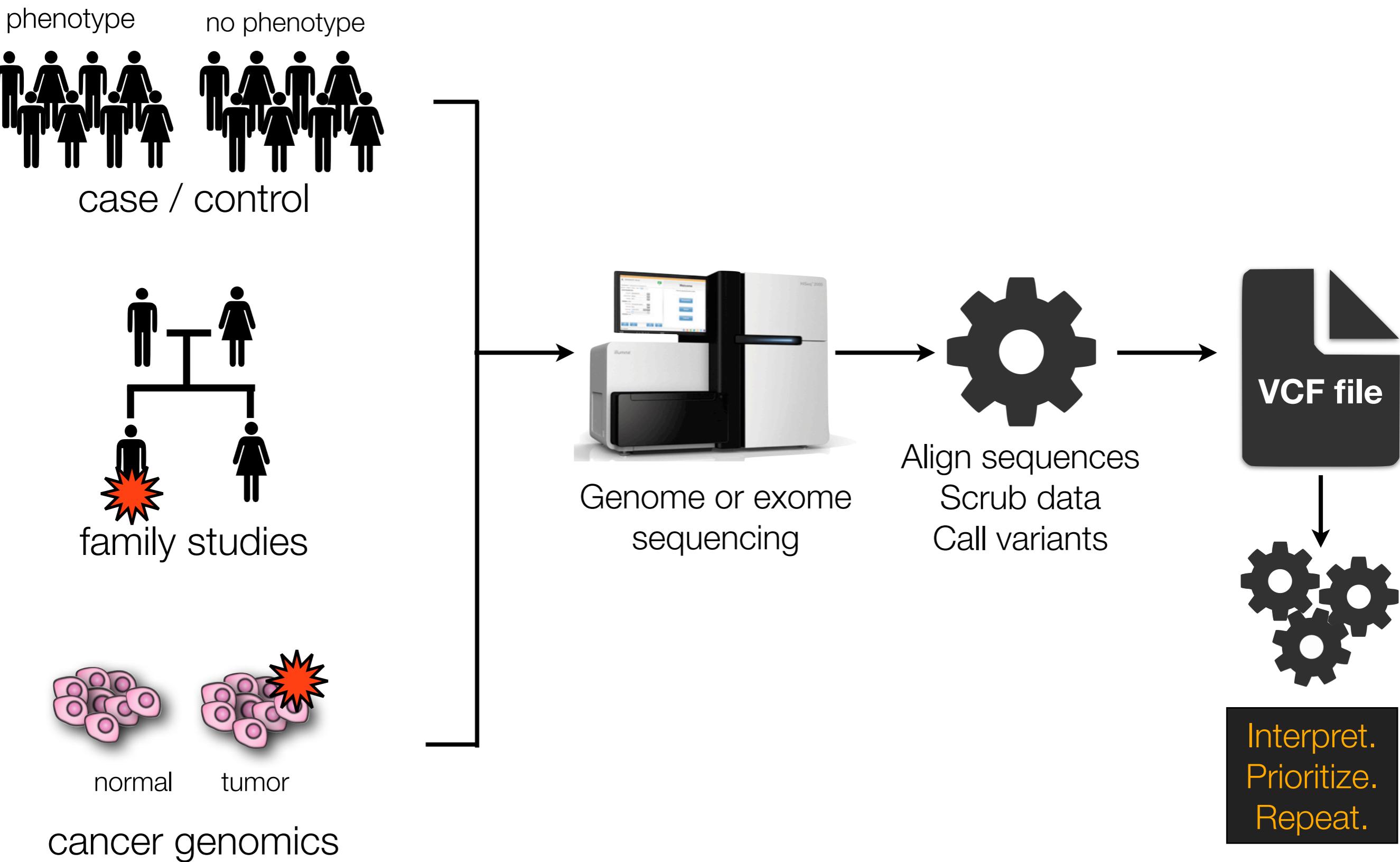
Exploring human genomes: outline

GEMINI

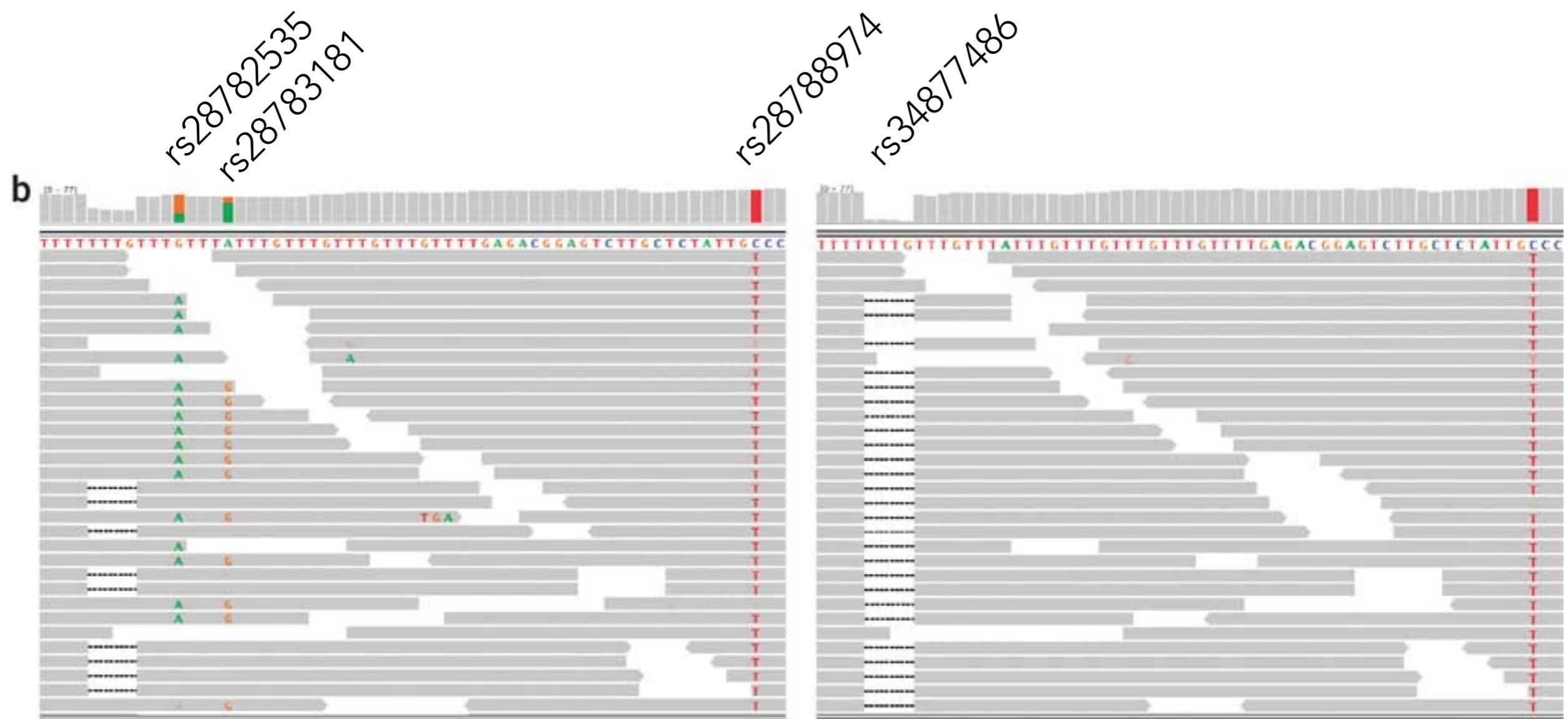
- Technical challenges (many)
- Analytical challenges (more still?)
- Data integration (large; diverse)
- Our solution: GEMINI

A FRAMEWORK FOR
MINING GENOME VARIATION

Typical study designs



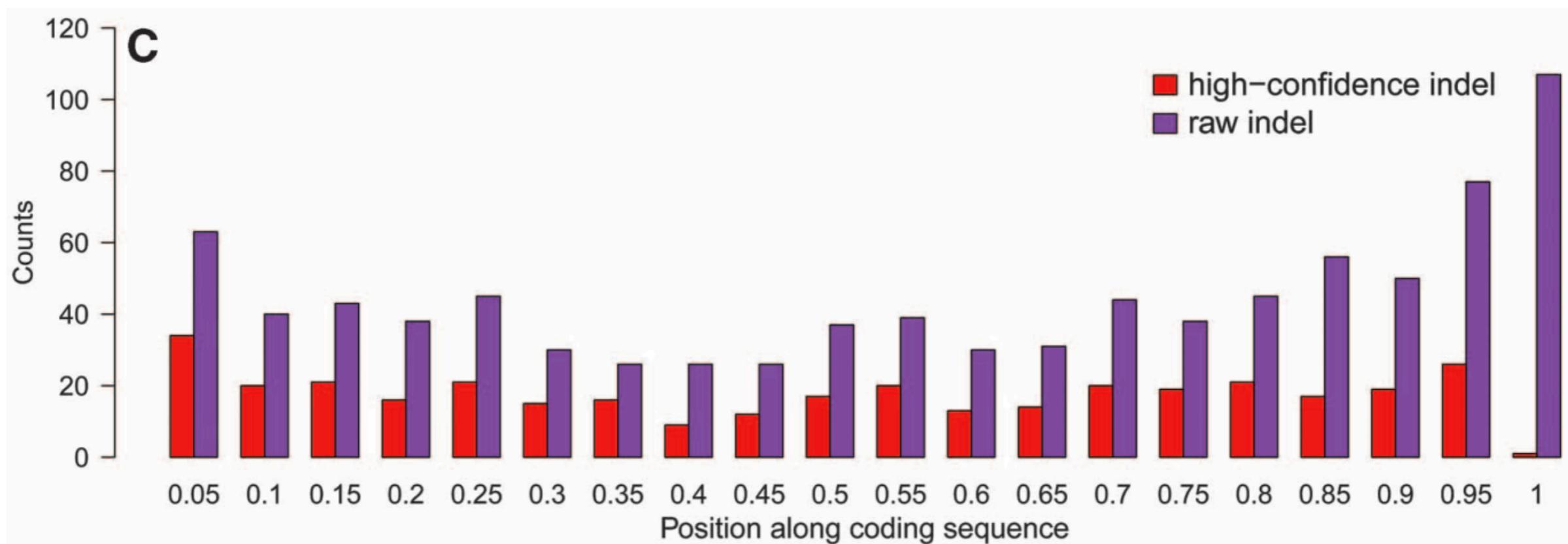
Technical Challenges: alignment issues



“raw” alignments

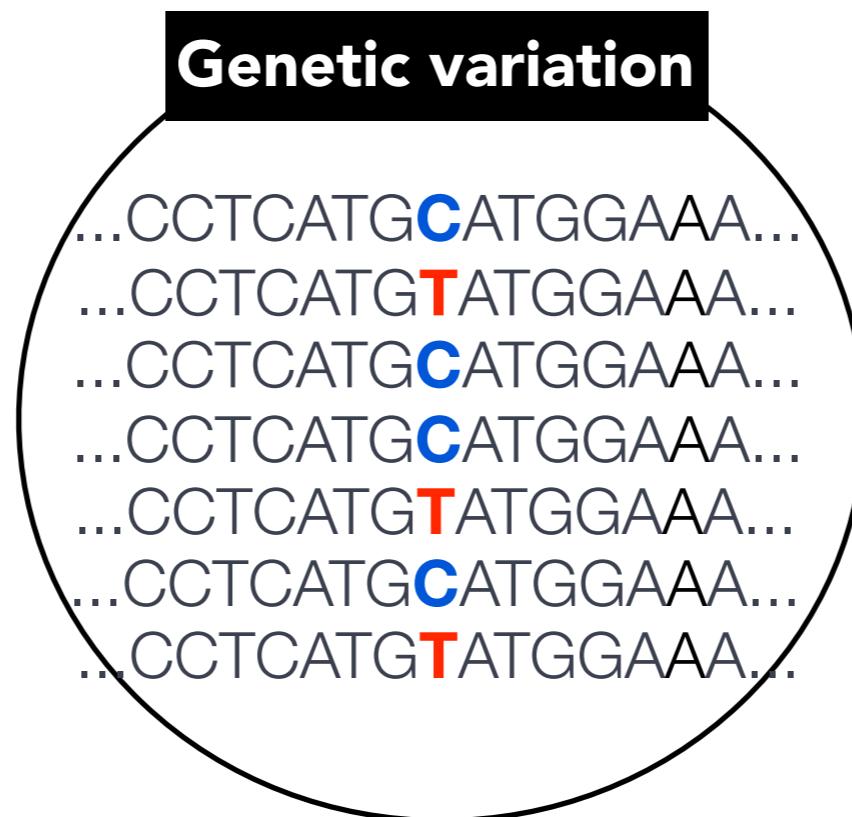
after MSA correction

Technical Challenges: LoF variants

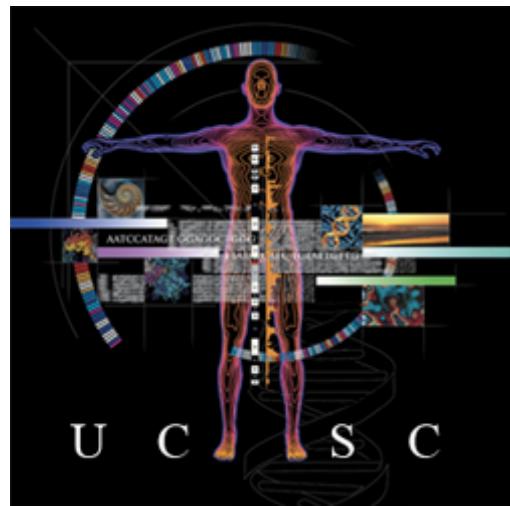


Variant type	Before filtering						After filtering			
	Total	1000G low-coverage average per individual			NA12878	Total	1000G low-coverage average per individual			NA12878
		CEU	CHB+JPT	YRI			CEU	CHB+JPT	YRI	
Stop	1111	85.7 (21.8)	113.4 (26.7)	109.1 (23.7)	115 (25)	565	26.2 (5.2)	27.4 (6.9)	37.2 (6.3)	23 (2)
Splice	658	80.5 (29.5)	98.1 (35.6)	89.0 (30.4)	95 (32)	267	11.2 (1.9)	13.2 (2.5)	13.7 (1.9)	12 (1)
Frameshift indel	1040	217.8 (112.1)	225.5 (121.7)	247.2 (118.7)	348 (159)	337	38.2 (9.2)	36.2 (9.0)	44.0 (8.0)	38 (11)
Large deletion	142	32.4 (12.2)	31.2 (11.8)	31.4 (9.7)	31 (5)	116	28.3 (6.2)	26.7 (5.9)	26.6 (5.5)	24 (4)
Total	2951	416.4 (175.6)	468.2 (195.8)	476.7 (316.0)	654 (286)	1285	103.9 (22.5)	103.5 (24.3)	121.5 (21.7)	97 (18)

Analytical challenges: data integration



Analytical challenges: data integration

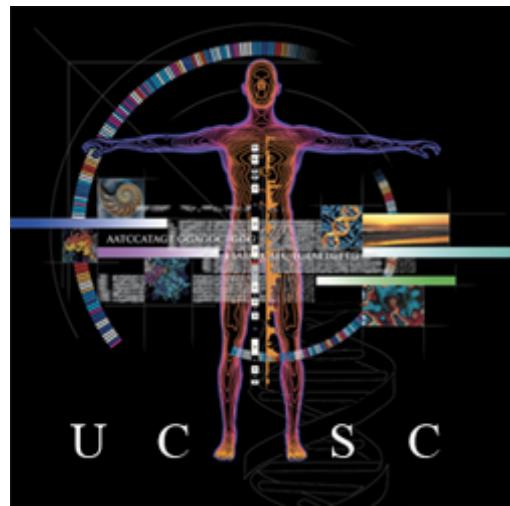


Conservation
Repeat elements
Genome Gaps
Cytobands
Gene annotations
“Mappability”
DeCIPHER
ISGA

Genetic variation

...CCTCATGCATGGAAA...
...CCTCATGTAATGGAAA...
...CCTCATGCATGGAAA...
...CCTCATGCATGGAAA...
...CCTCATGTAATGGAAA...
...CCTCATGCATGGAAA...
...CCTCATGTAATGGAAA...

Analytical challenges: data integration



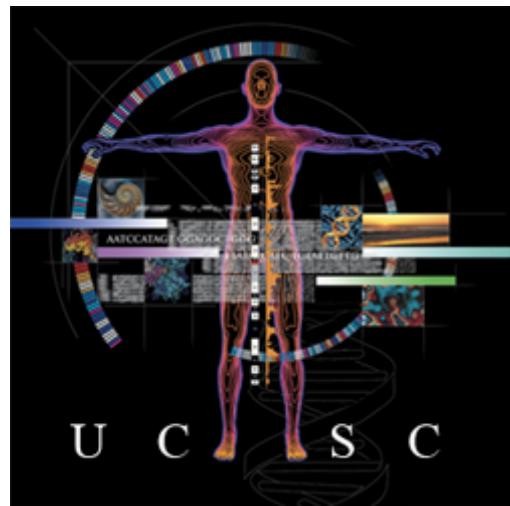
Conservation
Repeat elements
Genome Gaps
Cytobands
Gene annotations
“Mappability”
DeCIPHER
ISGA



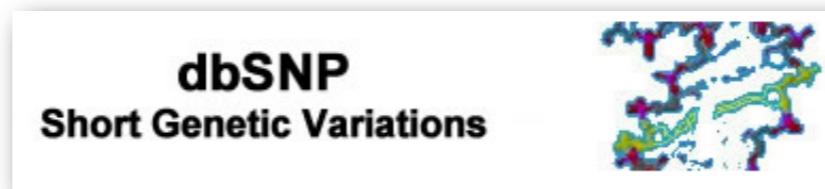
Genetic variation

...CCTCATGCATGGAAA...
...CCTCATG**T**ATGGAAA...
...CCTCATGC**C**ATGGAAA...
...CCTCATGC**C**ATGGAAA...
...CCTCATG**T**ATGGAAA...
...CCTCATGC**C**ATGGAAA...
...CCTCATG**T**ATGGAAA...

Analytical challenges: data integration



Conservation
Repeat elements
Genome Gaps
Cytobands
Gene annotations
“Mappability”
DeCIPHER
ISGA

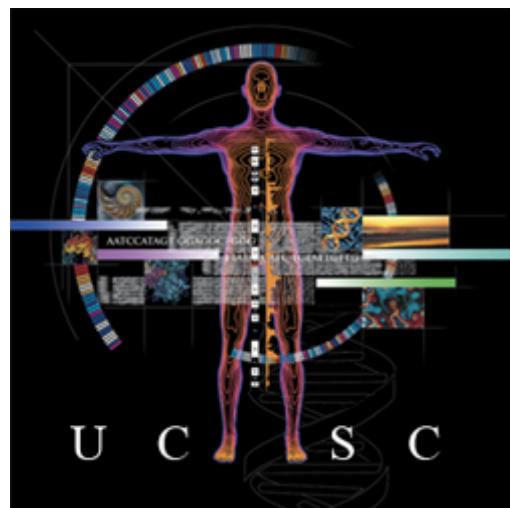


ClinVar

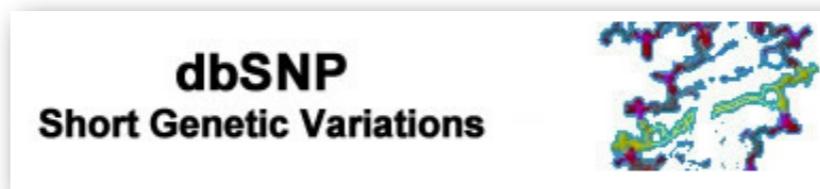
Genetic variation

...CCTCATGCATGGAAA...
...CCTCATG**T**ATGGAAA...
...CCTCATGC**C**ATGGAAA...
...CCTCATGC**C**ATGGAAA...
...CCTCATG**T**ATGGAAA...
...CCTCATGC**C**ATGGAAA...
...CCTCATG**T**ATGGAAA...

Analytical challenges: data integration



Conservation
Repeat elements
Genome Gaps
Cytobands
Gene annotations
“Mappability”
DeCIPHER
ISGA



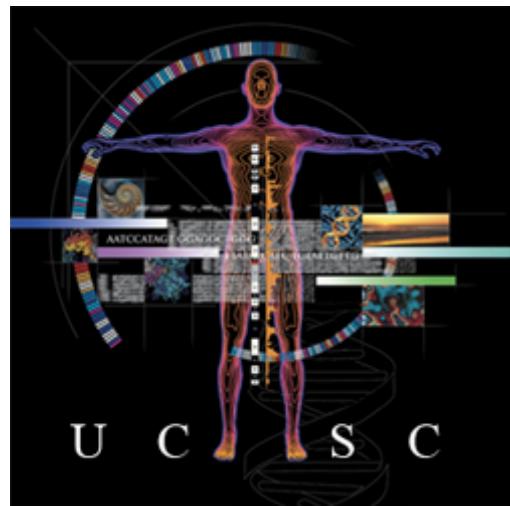
ClinVar

OMIM
Online Mendelian Inheritance in Man

Genetic variation

...CCTCATGCATGGAAA...
...CCTCATGTATGGAAA...
...CCTCATGCATGGAAA...
...CCTCATGCATGGAAA...
...CCTCATGTATGGAAA...
...CCTCATGCATGGAAA...
...CCTCATGTATGGAAA...

Analytical challenges: data integration



Conservation
Repeat elements
Genome Gaps
Cytobands
Gene annotations
“Mappability”
DeCIPHER
ISGA



ClinVar

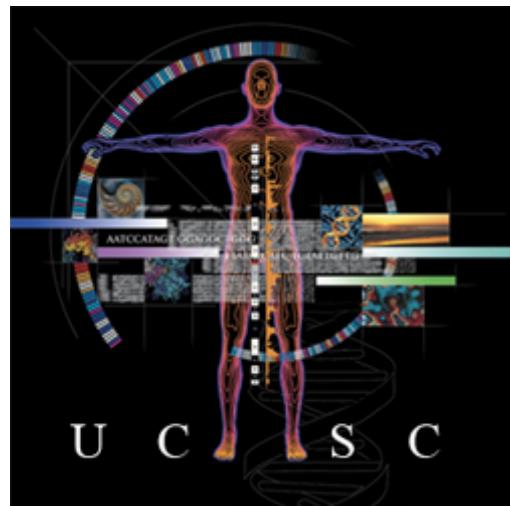
OMIM
Online Mendelian Inheritance in Man

1000 Genomes
A Deep Catalog of Human Genetic Variation

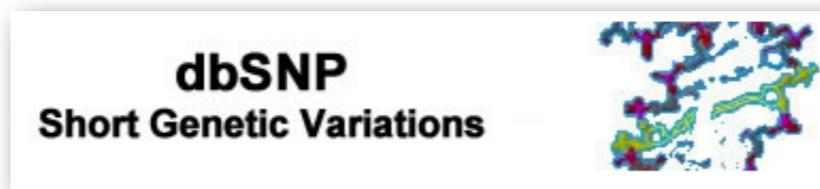
Genetic variation

...CCTCATGCATGGAAA...
...CCTCATG**T**ATGGAAA...
...CCTCATGC**C**ATGGAAA...
...CCTCATGC**C**ATGGAAA...
...CCTCATG**T**ATGGAAA...
...CCTCATGC**C**ATGGAAA...
...CCTCATG**T**ATGGAAA...

Analytical challenges: data integration



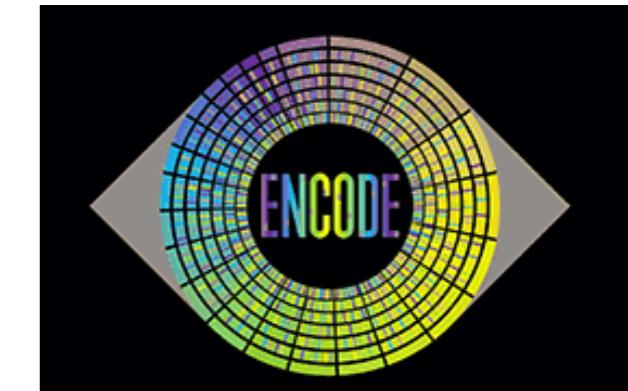
Conservation
Repeat elements
Genome Gaps
Cytobands
Gene annotations
"Mappability"
DeCIPHER
ISGA



ClinVar

OMIM
Online Mendelian Inheritance in Man

1000 Genomes
A Deep Catalog of Human Genetic Variation

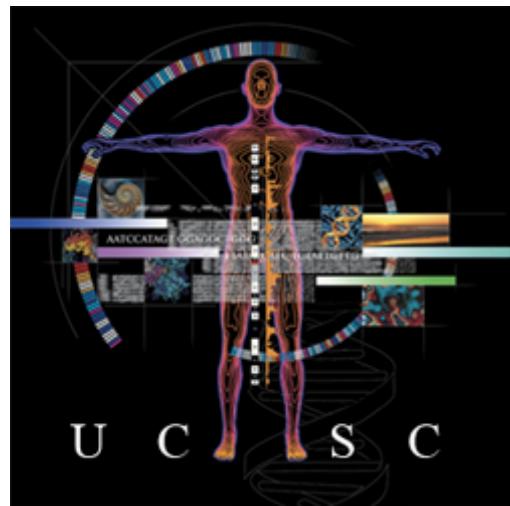


Chromatin marks
DNA methylation
RNA expression
TF binding

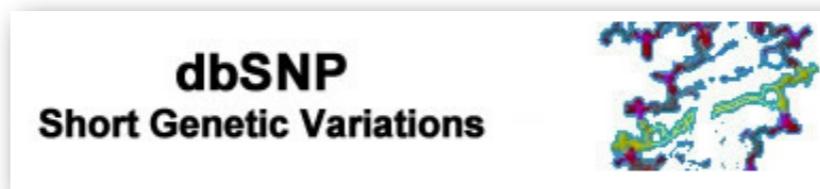
Genetic variation

...CCTCATGCATGGAAA...
...CCTCATGTATGGAAA...
...CCTCATGCATGGAAA...
...CCTCATGCATGGAAA...
...CCTCATGTATGGAAA...
...CCTCATGCATGGAAA...
...CCTCATGTATGGAAA...

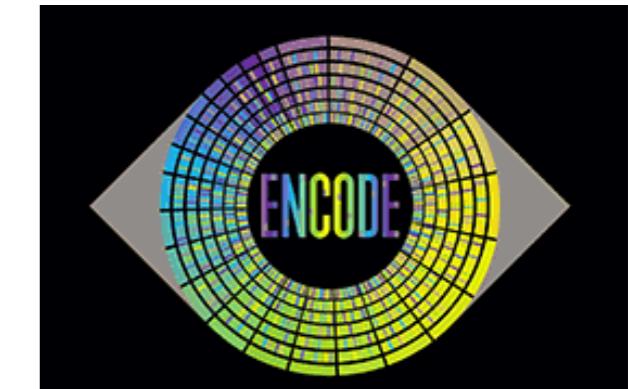
Analytical challenges: data integration



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



ClinVar

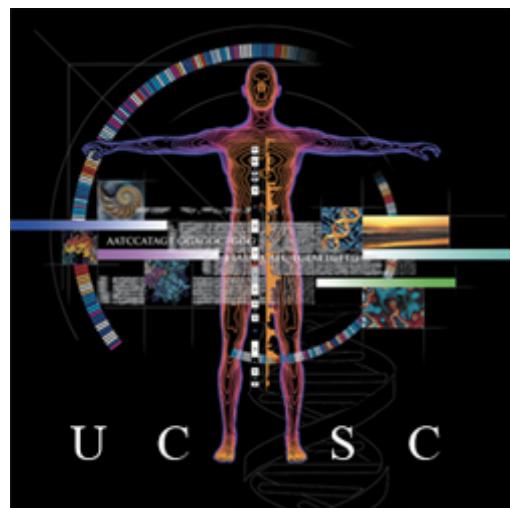


Genetic variation

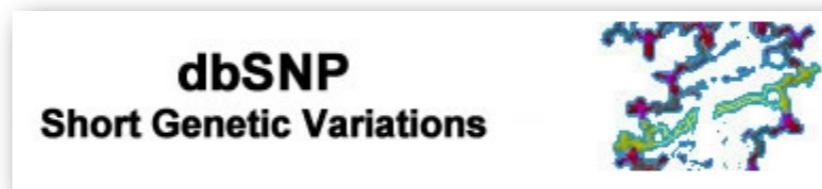
...CCTCATGCATGGAAA...
...CCTCATG**T**ATGGAAA...
...CCTCATGC**C**ATGGAAA...
...CCTCATGC**C**ATGGAAA...
...CCTCATG**T**ATGGAAA...
...CCTCATGC**C**ATGGAAA...
...CCTCATG**T**ATGGAAA...

Chromatin marks
DNA methylation
RNA expression
TF binding

Analytical challenges: data integration



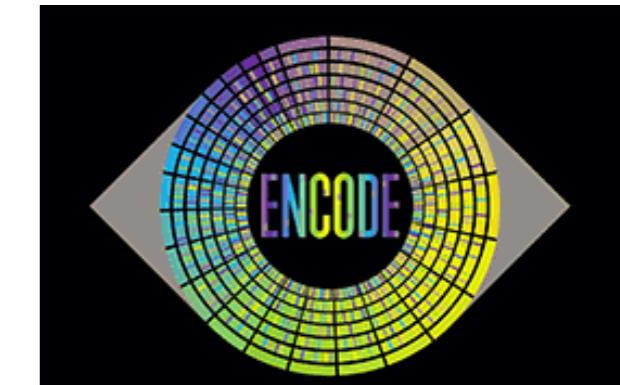
Conservation
Repeat elements
Genome Gaps
Cytobands
Gene annotations
"Mappability"
DeCIPHER
ISGA



ClinVar

OMIM
Online Mendelian Inheritance in Man

1000 Genomes
A Deep Catalog of Human Genetic Variation



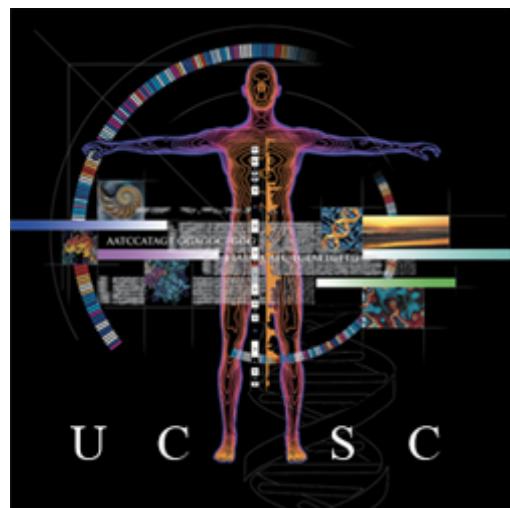
Chromatin marks
DNA methylation
RNA expression
TF binding



Genetic variation

...CCTCATGCATGGAAA...
...CCTCATG**T**ATGGAAA...
...CCTCATGC**T**ATGGAAA...
...CCTCATGC**T**ATGGAAA...
...CCTCATG**T**ATGGAAA...
...CCTCATGC**T**ATGGAAA...
...CCTCATG**T**ATGGAAA...

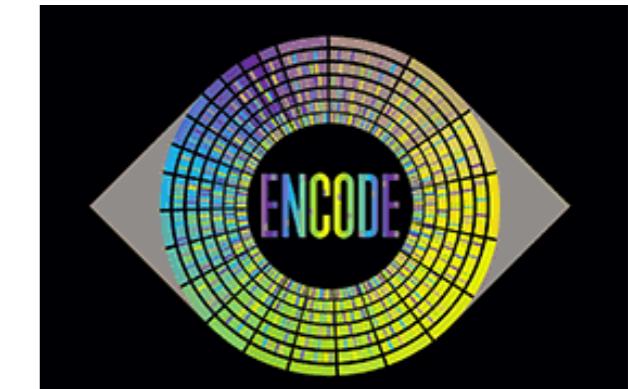
Analytical challenges: data integration



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



ClinVar

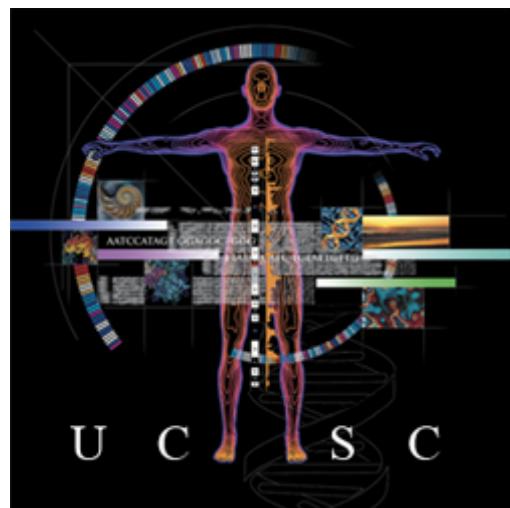


Genetic variation

...CCTCATGCATGGAAA...
...CCTCATGTATGGAAA...
...CCTCATGCATGGAAA...
...CCTCATGCATGGAAA...
...CCTCATGTATGGAAA...
...CCTCATGCATGGAAA...
...CCTCATGTATGGAAA...

Chromatin marks
DNA methylation
RNA expression
TF binding

Analytical challenges: data integration

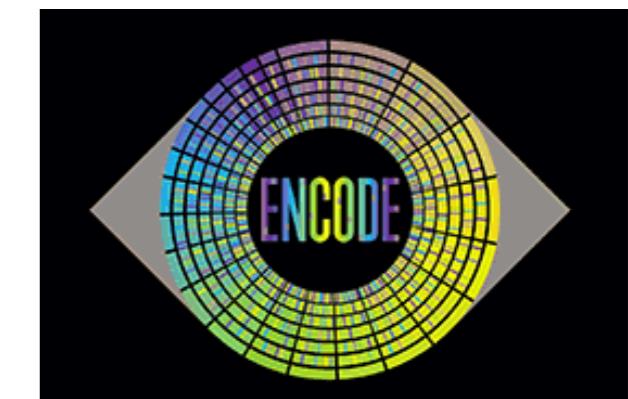


ClinVar

OMIM
Online Mendelian Inheritance in Man

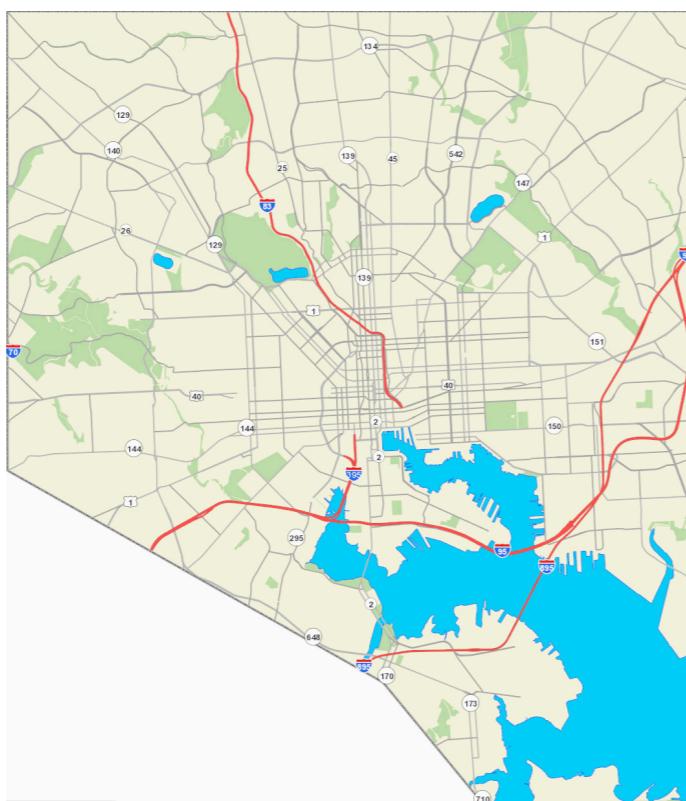
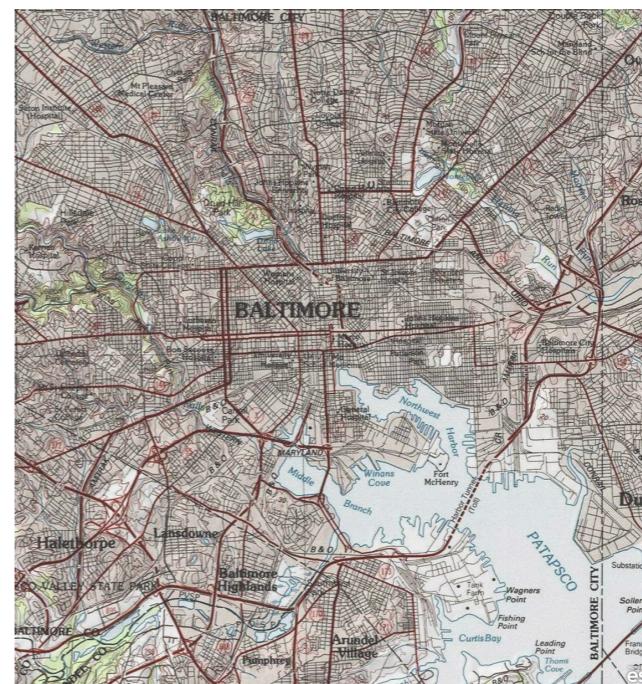
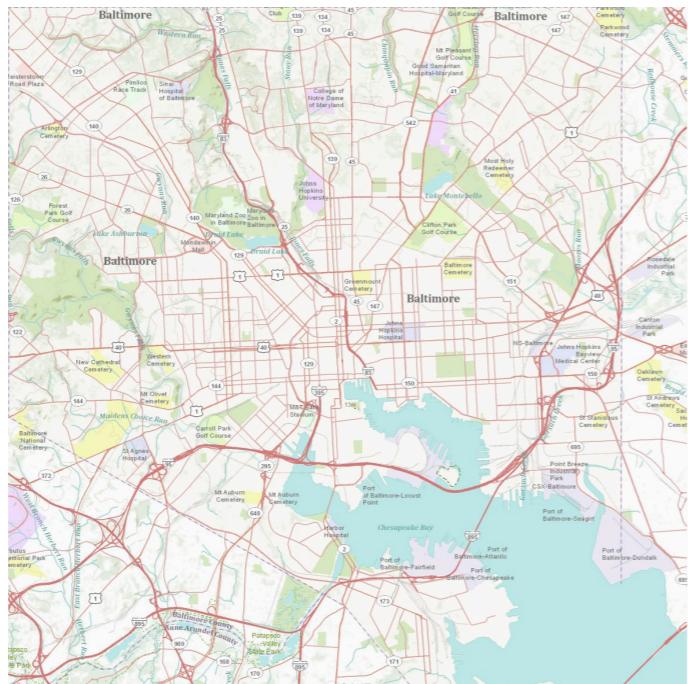
1000 Genomes
A Deep Catalog of Human Genetic Variation

Pfam



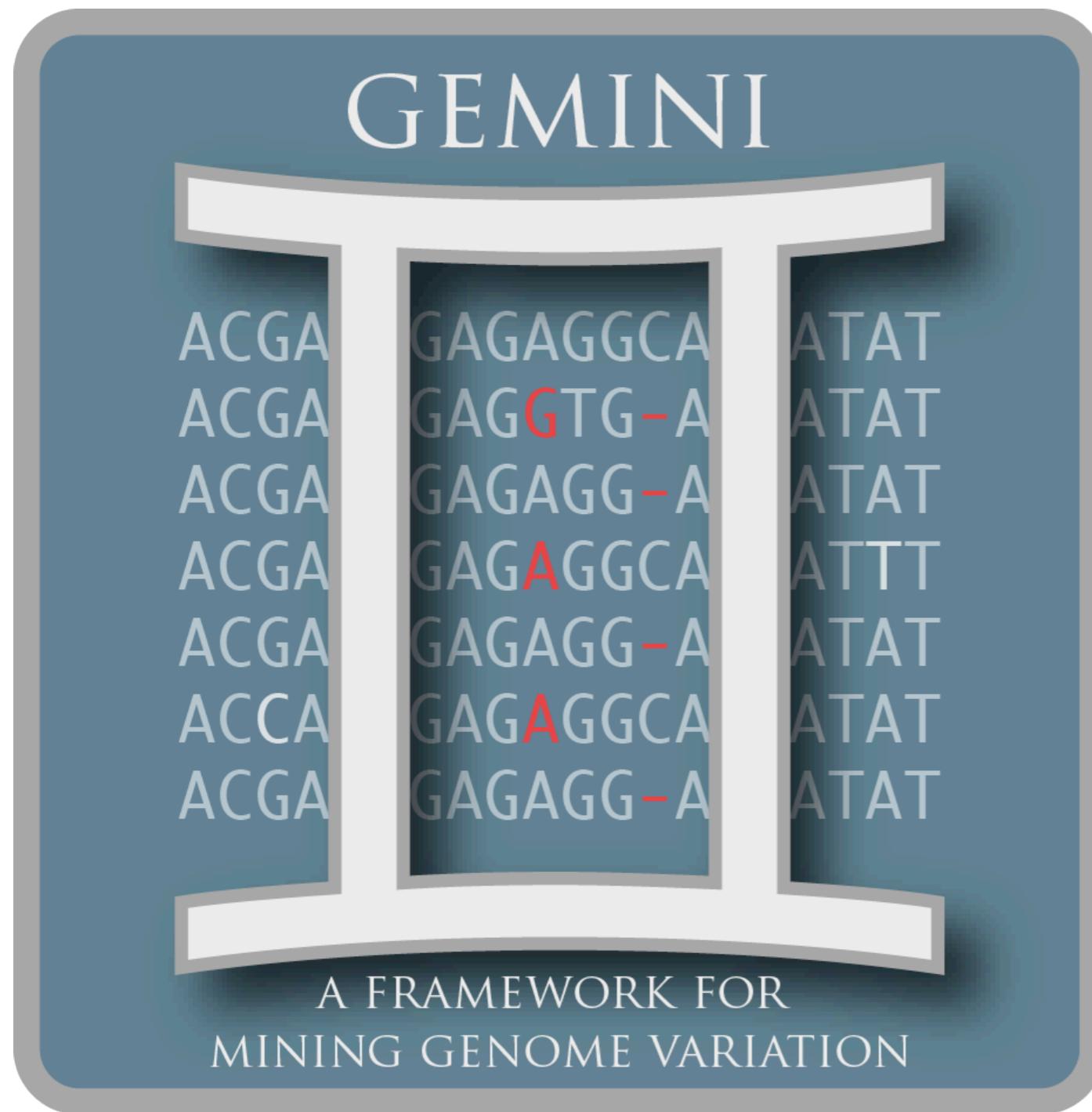
Chromatin marks
DNA methylation
RNA expression
TF binding

Much like map “overlays”



Each layer provides distinct information in different contexts.

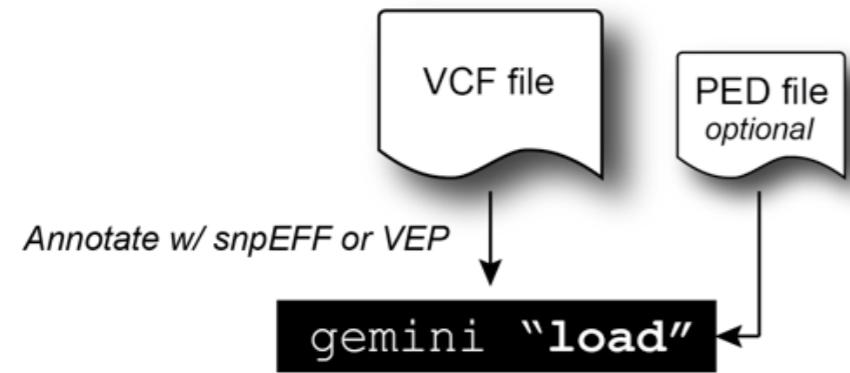
Our solution: GEnome MINIng



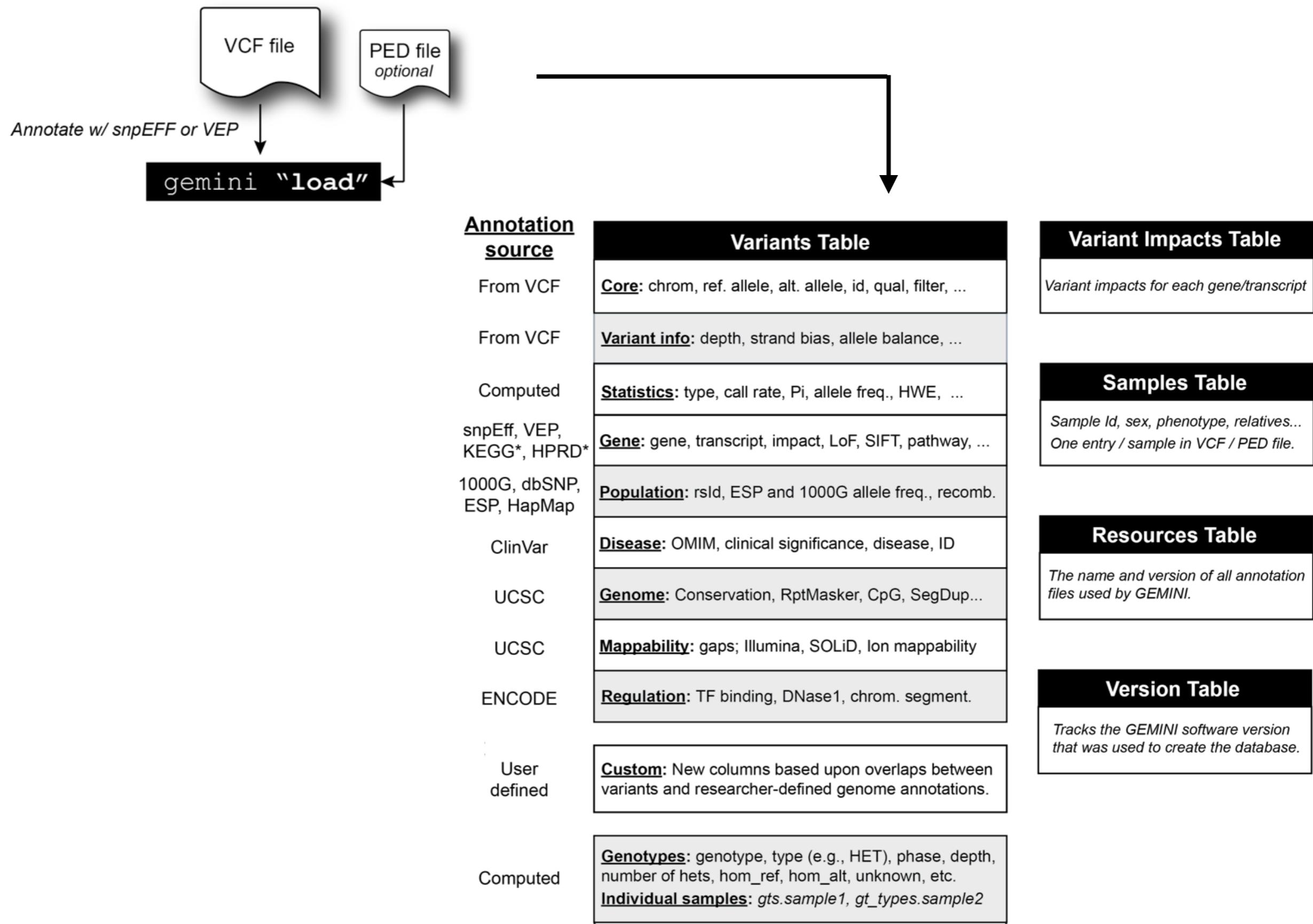
- Powerful
- Flexible
- Comprehensive
- Scalable
- Portable
- Reproducible

Framework for personal, medical, and population genomics

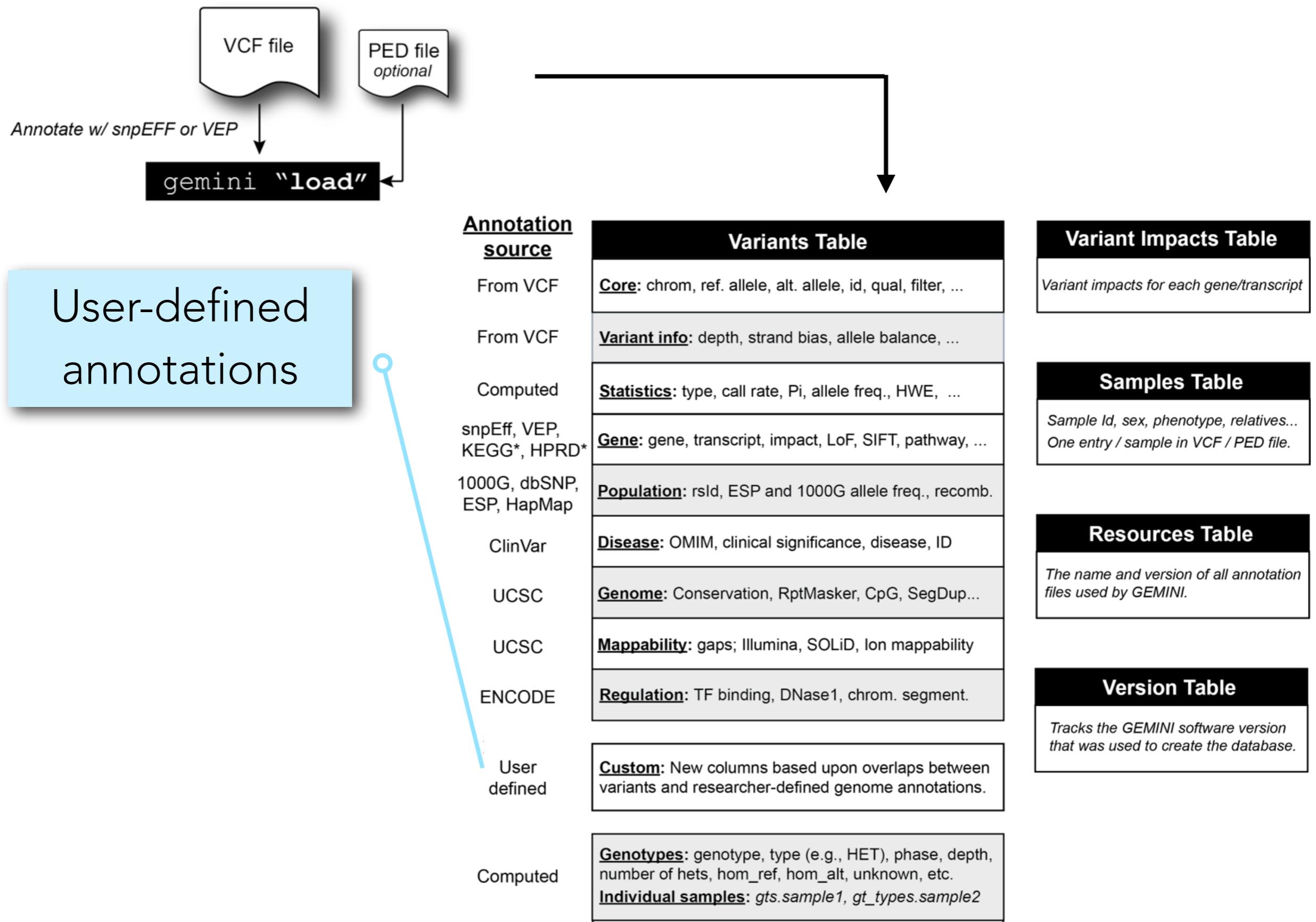
The GEMINI framework



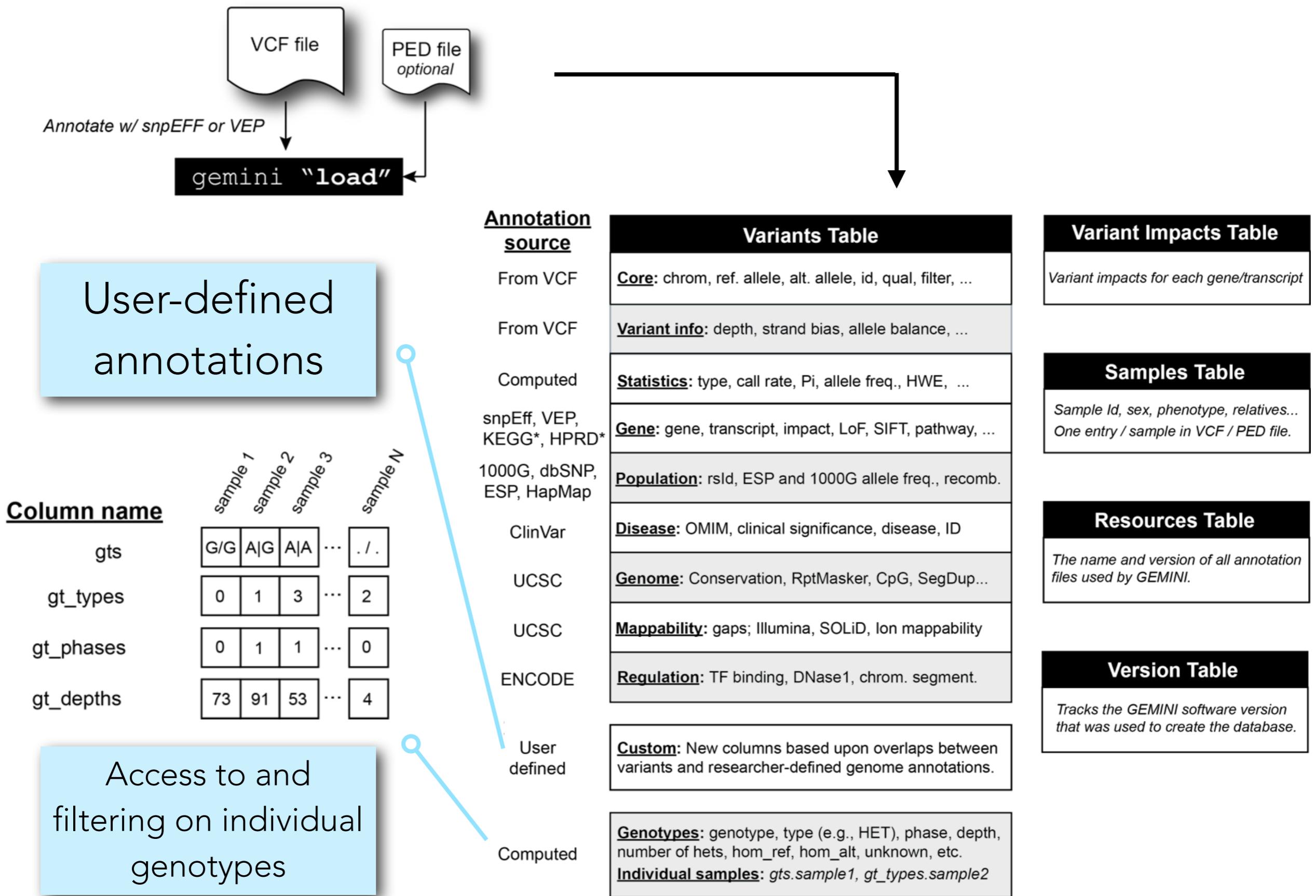
The GEMINI framework



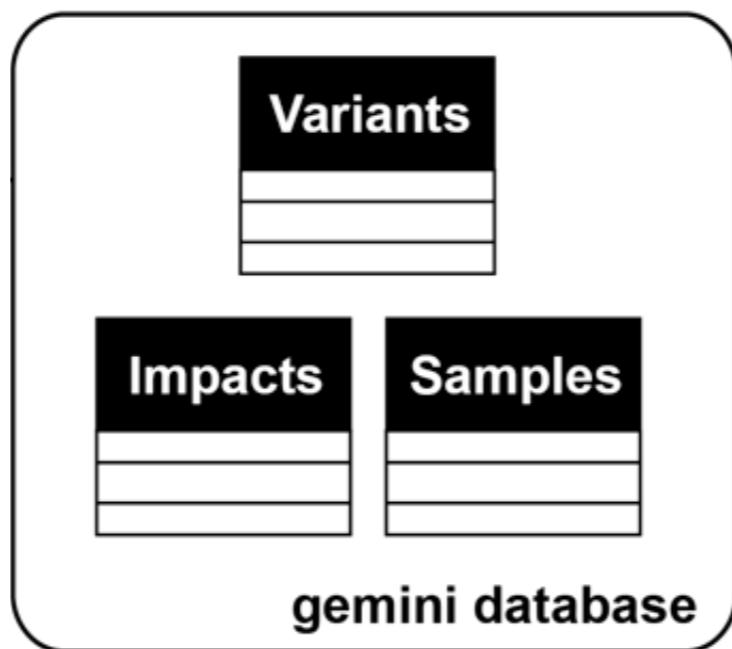
The GEMINI framework



The GEMINI framework



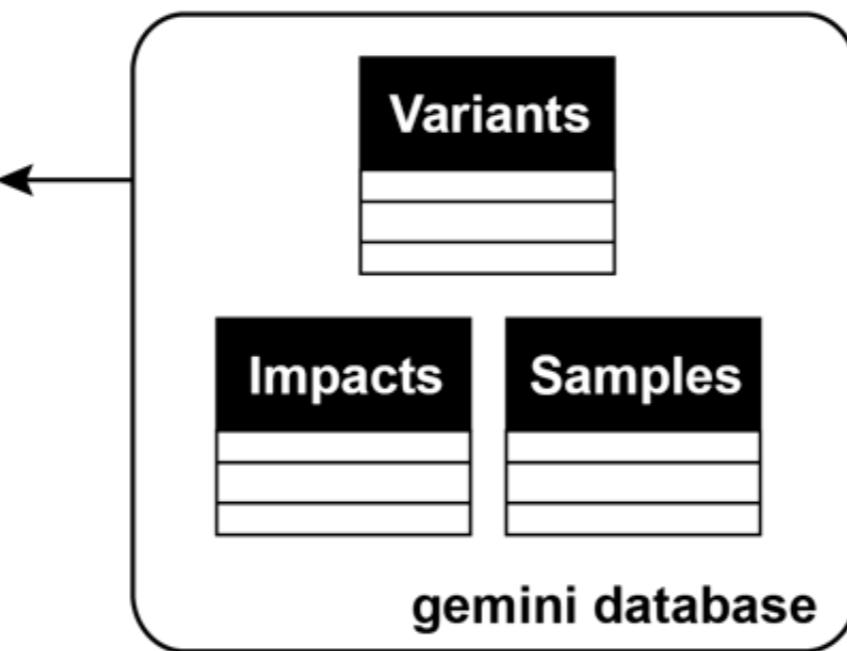
Mining variation with GEMINI



Mining variation with GEMINI

ad hoc data exploration

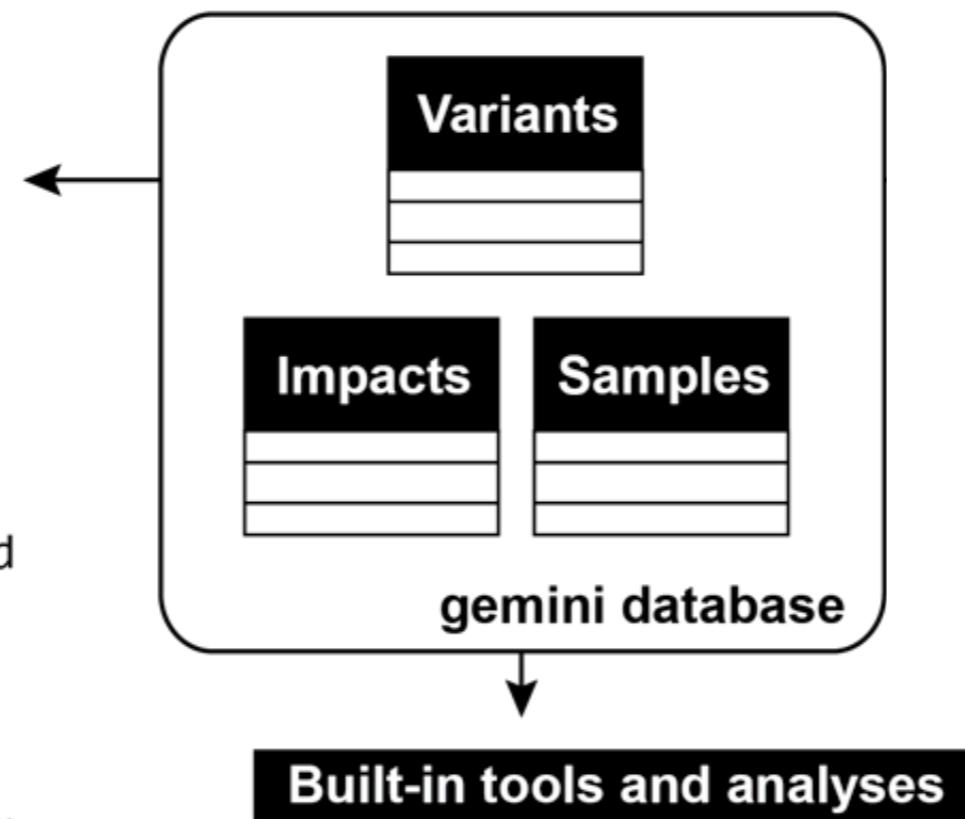
```
gemini query  
--query  
"select chrom, start, end,  
    ref, alt, gene,  
    impact, aaf, gts.kid  
from variants  
where in_dbsnp = 0  
and aaf < 0.01  
and is_lof = 1  
and my_disease_regions = 1"  
  
--gt-filter  
"gt_types.mom == HET  
and  
gt_types.dad == HET  
and  
gt_types.proband == HOM_ALT"
```



Mining variation with GEMINI

ad hoc data exploration

```
gemini query  
--query  
"select chrom, start, end,  
    ref, alt, gene,  
    impact, aaf, gts.kid  
from variants  
where in_dbsnp = 0  
and aaf < 0.01  
and is_lof = 1  
and my_disease_regions = 1"  
  
--gt-filter  
"gt_types.mom == HET  
and  
gt_types.dad == HET  
and  
gt_types.proband == HOM_ALT"
```

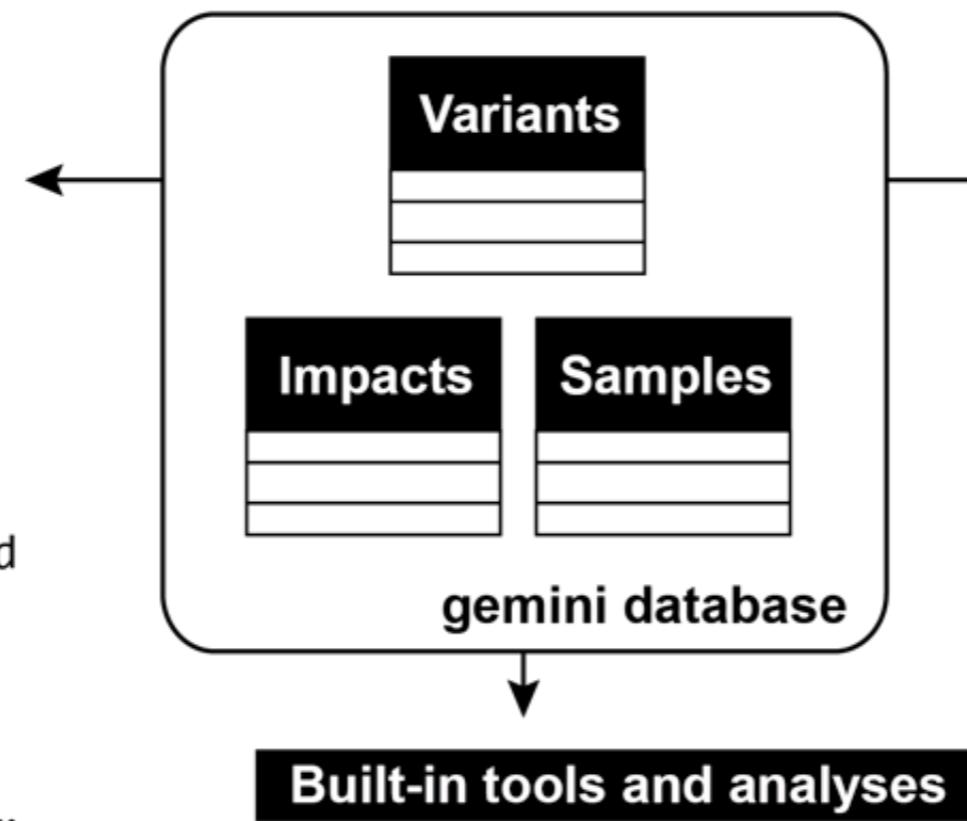


Tool	Description
region	extract variants spec. genomic intervals or genes
stats	compute variant statistics (SFS, Ts/Tv, counts, etc.)
annotate	add new columns based on custom annotations
windower	compute variant statistics across genome “windows”
comp_hets	identify candidate compound heterozygotes
pathways	maps genes and variants to KEGG pathways
lof_sieve	prioritize candidate loss-of-function variants
interact	find protein interactions for genes/variants/samples
auto_rec	identify variants meeting an autosomal recessive model
auto_dom	identify variants meeting an autosomal dominant model
de_novo	identify candidate de novo mutations
browser	launch the interactive gemini web browser interface

Mining variation with GEMINI

ad hoc data exploration

```
gemini query  
--query  
"select chrom, start, end,  
ref, alt, gene,  
impact, aaf, gts.kid  
from variants  
where in_dbsnp = 0  
and aaf < 0.01  
and is_lof = 1  
and my_disease_regions = 1"  
  
--gt-filter  
"gt_types.mom == HET  
and  
gt_types.dad == HET  
and  
gt_types.proband == HOM_ALT"
```



Framework for new tools

- Burden tests
- Population genetics
- Pedigree studies
- Haplotype analysis
- Custom tools and new methods

Built-in tools and analyses

Tool	Description
region	extract variants spec. genomic intervals or genes
stats	compute variant statistics (SFS, Ts/Tv, counts, etc.)
annotate	add new columns based on custom annotations
windower	compute variant statistics across genome “windows”
comp_hets	identify candidate compound heterozygotes
pathways	maps genes and variants to KEGG pathways
lof_sieve	prioritize candidate loss-of-function variants
interact	find protein interactions for genes/variants/samples
auto_rec	identify variants meeting an autosomal recessive model
auto_dom	identify variants meeting an autosomal dominant model
de_novo	identify candidate de novo mutations
browser	launch the interactive gemini web browser interface

Example workflow



Example analyses: *ad hoc* exploration

1. Find all novel or rare (<1%), LoF variants

```
$ gemini query -q "select * from variants  
where is_lof = 1  
and (in_dbsnp = 0 or  
aaf <= 0.01)"  
my.gemini.db
```

Example analyses: *ad hoc* exploration

1. Find all novel or rare (<1%), LoF variants

```
$ gemini query -q "select * from variants  
where is_lof = 1  
and (in_dbsnp = 0 or  
aaf <= 0.01)"  
my.gemini.db
```

2. Find all variants with a known clinical phenotype

```
$ gemini query -q "select * from variants  
where in_omim = 1  
or clinvar_disease_name is not NULL"  
my.gemini.db
```

Example analyses: *ad hoc* exploration

1. Find all novel or rare (<1%), LoF variants

```
$ gemini query -q "select * from variants  
where is_lof = 1  
and (in_dbsnp = 0 or  
aaf <= 0.01)"  
my.gemini.db
```

2. Find all variants with a known clinical phenotype

```
$ gemini query -q "select * from variants  
where in_omim = 1  
or clinvar_disease_name is not NULL"  
my.gemini.db
```

3. Find LoF variants where a sample has a specific genotype

```
$ gemini query -q "select * from variants  
where is_lof = 1"  
--gt_filter "gts.sampleXYZ == 'HET'"  
my.gemini.db
```

Example analyses: *built-in analyses*

1. Find *de novo* mutations

```
$ gemini de_novo my.gemini.db
```

Example analyses: *built-in analyses*

1. Find *de novo* mutations

```
$ gemini de_novo my.gemini.db
```

2. Find compound heterozygotes

```
$ gemini comp_hets my.gemini.db
```

Example analyses: *built-in analyses*

1. Find *de novo* mutations

```
$ gemini de_novo my.gemini.db
```

2. Find compound heterozygotes

```
$ gemini comp_hets my.gemini.db
```

3. Find variants meeting an auto. recessive pattern

```
$ gemini auto_recessive my.gemini.db
```

Example analyses: *built-in analyses*

1. Find de novo mutations

```
$ gemini de_novo my.gemini.db
```

2. Find compound heterozygotes

```
$ gemini comp_hets my.gemini.db
```

3. Find variants meeting an auto. recessive pattern

```
$ gemini auto_recessive my.gemini.db
```

4. Find variants meeting an auto. dominant pattern

```
$ gemini auto_dominant my.gemini.db
```

Example analyses: *built-in analyses*

1. Find *de novo* mutations

```
$ gemini de_novo my.gemini.db
```

2. Find compound heterozygotes

```
$ gemini comp_hets my.gemini.db
```

3. Find variants meeting an auto. recessive pattern

```
$ gemini auto_recessive my.gemini.db
```

4. Find variants meeting an auto. dominant pattern

```
$ gemini auto_dominant my.gemini.db
```

5. Find interacting genes each with LoF variants.

```
$ gemini lof_interactions my.gemini.db
```

Example analyses: *tool / method dev.*

```
from gemini import GeminiQuery  
  
query = GeminiQuery("my.db")
```

Example analyses: *tool / method dev.*

```
from gemini import GeminiQuery

query = GeminiQuery("my.db")
query.run("select * from variants")
for row in query:
    # print specific columns
    print row['chrom'], row['rsid']
```

Example analyses: *tool / method dev.*

```
from gemini import GeminiQuery

query = GeminiQuery("my.db")
query.run("select * from variants")
for row in query:
    # print specific columns
    print row['chrom'], row['rsid']

    # extract sample genotypes
    genotype_types = row.gt_types
```

Example analyses: *tool / method dev.*

```
from gemini import GeminiQuery

query = GeminiQuery("my.db")
query.run("select * from variants")
for row in query:
    # print specific columns
    print row['chrom'], row['rsid']

    # extract sample genotypes
    genotype_types = row.gt_types

    # association test
    if assoc_test(genotype_types) < 1E-8:
        print row
```

Example analyses: *tool / method dev.*

```
from gemini import GeminiQuery

query = GeminiQuery("my.db")
query.run("select * from variants")
for row in query:
    # print specific columns
    print row['chrom'], row['rsid']

    # extract sample genotypes
    genotype_types = row.gt_types

    # association test
    if assoc_test(genotype_types) < 1E-8:
        print row

    # your groundbreaking idea!
    if whizbang_test(genotype_types) < 1E-8:
        print row
```

Example datasets

- Illumina Platinum Trio (CEU trio)
 - <ftp://ftp.platinumgenomes.org/trio/trio.vcf.gz>
 - 6.4 million variants; 3 genotypes / variant
 - VCF is 319 Megabytes
- 1000 genomes VCF from 1046 samples
 - http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase1/analysis_results/integrated_call_sets/
 - 39.7 million variants; 1046 genotypes / variant
 - VCF is 144 Gigabytes

Performance: *database loading and size*

Loading is slowest step, but can be parallelized on clusters or multi-CPU machines.

- Illumina Platinum Trio (CEU trio)
 - 84 minutes with 4 CPUs on a Macbook Pro
 - Database is 1.2 Gigabytes
- 1000 genomes VCF from 1046 samples
 - 28 hours with 30 CPUs on a multi-CPU machine.
 - 2 hours on a cluster using 250 CPUs.
 - Database is 78 Gb versus 144 Gb for original VCF

Performance: *querying the database*

Experiment	Platinum Trio	1046 samples
Return all novel variants <code>select * from variants where in_dbsnp = 0</code>	24 sec (N=345,028)	11 sec (N=87,939)
Return all loss-of-func. variants <code>select * from variants where is_lof = 1</code>	2 sec (N=1,126)	177 sec (N=13,049)
Return all rare, loss-of-func. variants <code>select * from variants where is_lof = 1 and aaf < 0.01</code>	2 sec (N=112)	152 sec (N=12,683)
Filtering variants based on sample genotype criteria <code>select * from variants where is_lof = 1" --gt-filter "gt_types.NA12878 == HET"</code>	2 sec (N=487)	194 sec (N=384)

Filtering variants by specific genotypes requires decompression of sample genotype arrays.
Slower, but thankfully only marginally so. Hope for scalability.

GEMINI browser

SQL query
genotype filters
(e.g., inheritance patterns)
built-in links to IGV

The screenshot shows the Gemini query interface running locally at `localhost:8088/`. The interface has a dark header bar with tabs for "gemini browser", "Query", "Tools", "Docs", "About", and "Contact". The "Query" tab is active.

Gemini database:
`disease.gemini.db`

Query
e.g., `select chrom, start, end, ref, alt, gts.sample1 from variants limit 10`

```
SELECT chrom, start, end, ref, alt, gene
      gts.Mom, gts.Dad, gts.Proband
FROM variants
WHERE in_dbnsfp = 1 and is_coding = 1 and impact = 'non_syn_coding'
```

Genotype Filters
(e.g., `(gt_types.sample1 == HET and gt_types.sample2 == HOM_REF)`)

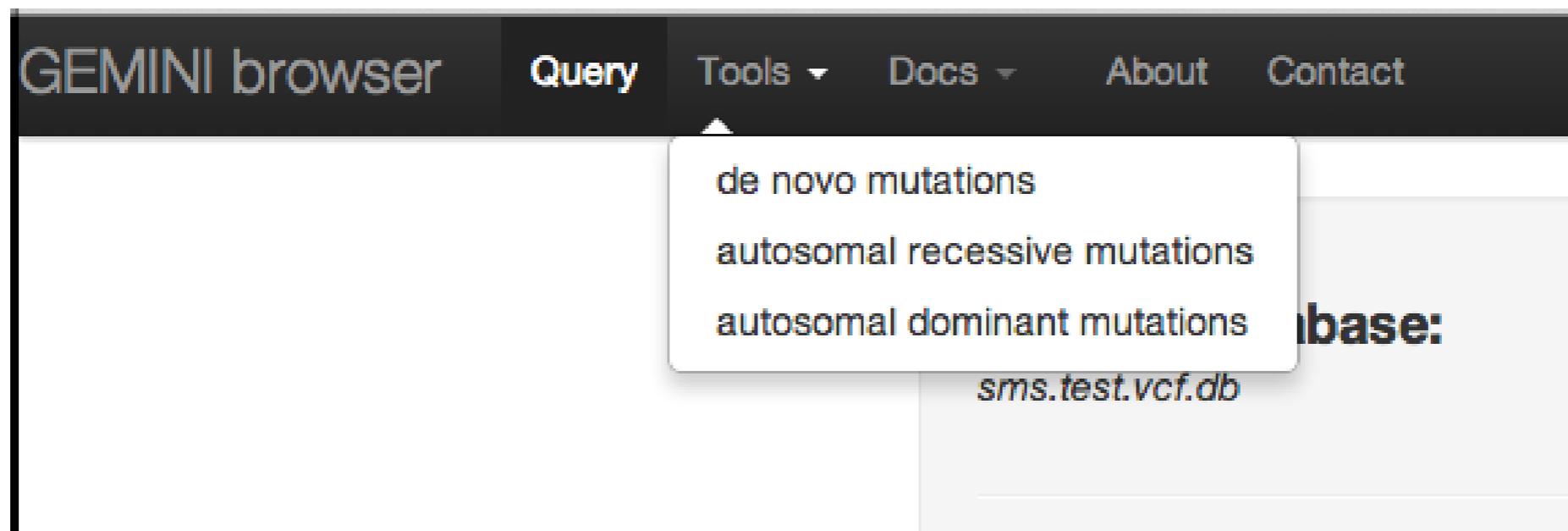
```
(gt_types.Mom == HET and
gt_types.Dad == HET and
gt_types.Proband == HOM_ALT)
```

Add a header?
 Make rows into IGV links (enable port 60151 in IGV)?
Note: If checked, you must include `chrom`, `start`, and `end` in your query.

Submit **Save as text file**

chrom	start	end	ref	alt	gene	gt_types.Mom	gt_types.Dad	gt_types.Proband	chrom	start	end	ref	alt	gene	gt_types.Mom	gt_types.Dad	gt_types.Proband
chr1	47282771	47282772	C	T	CYP4B1				C/T	C/T	T/T						
chr1	57383357	57383358	C	T	C8A				C/T	C/T	T/T						
chr1	62676283	62676284	C	T	L1TD1				C/T	C/T	T/T						
chr1	111861973	111861974	T	C	CHIA				T/C	T/C	C/C						
chr1	118482148	118482149	C	T	WDR3				C/T	C/T	T/T						
chr1	118565952	118565953	G	A	SPAG17				G/A	G/A	A/A						
chr1	118644523	118644524	T	A	SPAG17				T/A	T/A	A/A						
chr1	148754941	148754942	A	T	NBPF16				A/T	A/T	T/T						
chr1	155583936	155583937	A	G	MSTO1				A/G	A/G	G/G						
chr1	156347130	156347131	G	A	RHBG				G/A	G/A	A/A						
chr1	156622251	156622252	G	A	BCAN				G/A	G/A	A/A						
chr1	161967680	161967681	A	G	OLFML2B				A/G	A/G	G/G						

Built-in tools



Built-in documentation

GEMINI query interface ×

localhost:8088/db_schema

Calendar MicrobesOnline Spec Pathogen sequencing PubMed Central, Fig. PubMed Central, Fig. Krona – Root How to Clean and Re http://localhost:808 First steps in data vi

GEMINI browser Query Tools Docs About Contact

VARIANTS TABLE

Core VCF cols.
Variant / PopGen cols.
Genotype info cols
Gene information cols.
Optional VCF cols.
Variant frequency cols.
Disease info. cols.
Genome anno. cols.
Mappability cols.
ENCODE info. cols.

VARIANT_IMPACTS TABLE

SAMPLES TABLE

The *variants* table

Core columns

column_name	type	notes
chrom	STRING	The chromosome on which the variant resides
start	INTEGER	The 0-based start position.
end	INTEGER	The 1-based end position.
variant_id	INTEGER	PRIMARY_KEY
anno_id	INTEGER	Variant transcript number for the most severely affected transcript
ref	STRING	Reference allele
alt	STRING	Alternate allele for the variant
qual	INTEGER	Quality score for the assertion made in ALT
filter	STRING	A string of filters passed/failed in variant calling

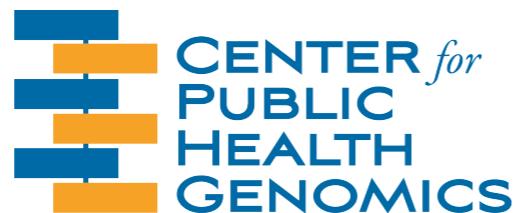
Variant and PopGen info

type	STRING	The type of variant. Any of: [snp, indel]
sub_type	STRING	The variant sub-type. If type is <i>snp</i> : [ts, (transition), tv (transversion)] If type is <i>indel</i> : [ins, (insertion), del (deletion)]
call_rate	FLOAT	The fraction of samples with a valid genotype
num_hom_ref	INTEGER	The total number of homozygotes for the reference (<i>ref</i>) allele
num_het	INTEGER	The total number of heterozygotes observed.
num_hom_alt	INTEGER	The total number of homozygotes for the reference (<i>alt</i>) allele
num_unknown	INTEGER	The total number of unknown genotypes

Our goals.

- Free. Open source.
 - github.com/arg5x/gemini
- Well tested and documented.
 - gemini.readthedocs.org
- Extensible, portable, & reproducible

Acknowledgements



Quinlan Lab



Uma Paila*

Postdoctoral Fellow

github.com/udp3f



Ryan Layer

Graduate Student

Co-mentored with Ira Hall

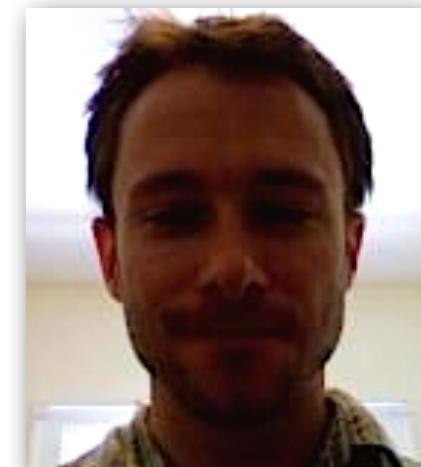
github.com/ryanlayer



Brad Chapman



Oliver Hofmann



Rory Kirchner