Exploring disease genetics among thousands of human genomes with GEMINI

Aaron Quinlan | University of Virginia | Jun 26, 2013

SciPy 2013 | Austin, Texas

quinlanlab.org

github.com/arq5x/gemini


@arq5x


@aaronquinlan

Acknowledgements

Uma Paila*

Postdoctoral Fellow

github.com/udp3f

Brad Chapman

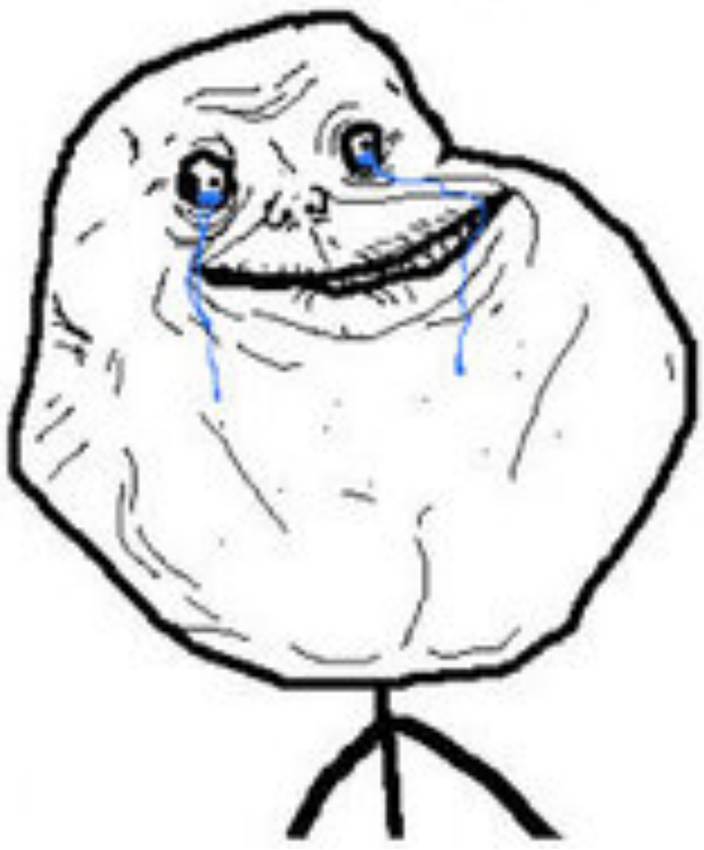
Oliver Hofmann

Rory Kirchner

National Human Genome Research Institute

R01HG006693-01

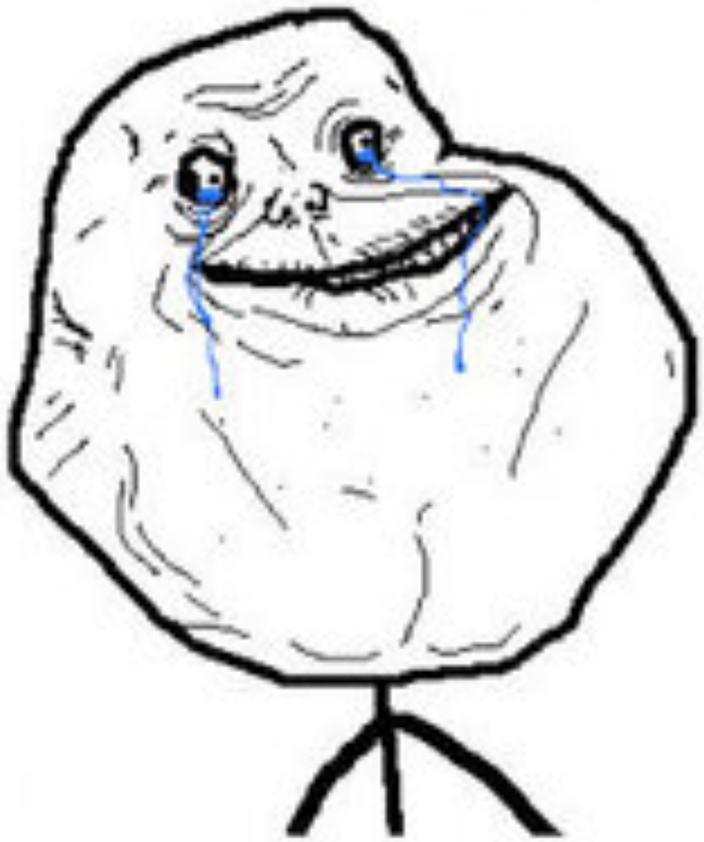
Motivation: Sequencing genomes? Easy. Understanding them? Hard.



Pre-2008 sadness

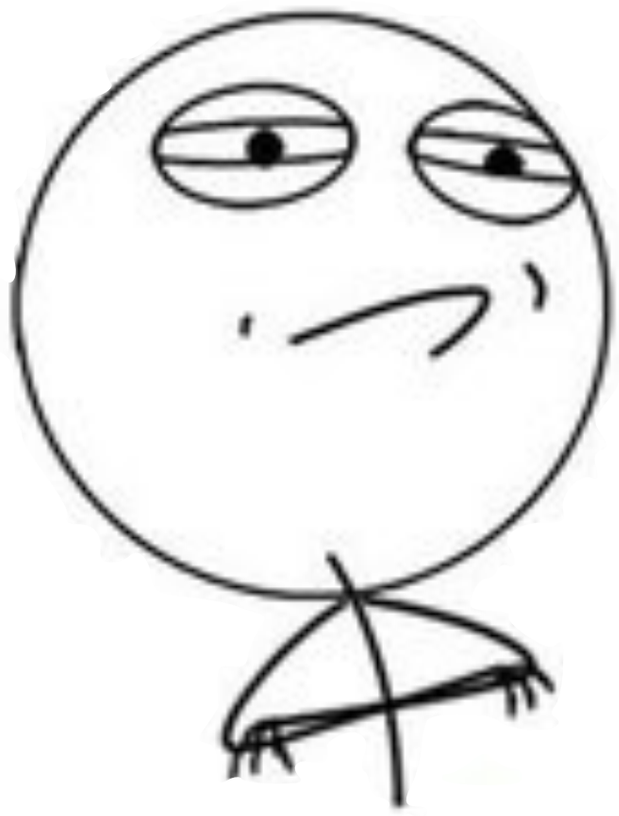
Sequencing human genomes was once very laborious and expensive

Motivation: Sequencing genomes? Easy. Understanding them? Hard.



Pre-2008 sadness

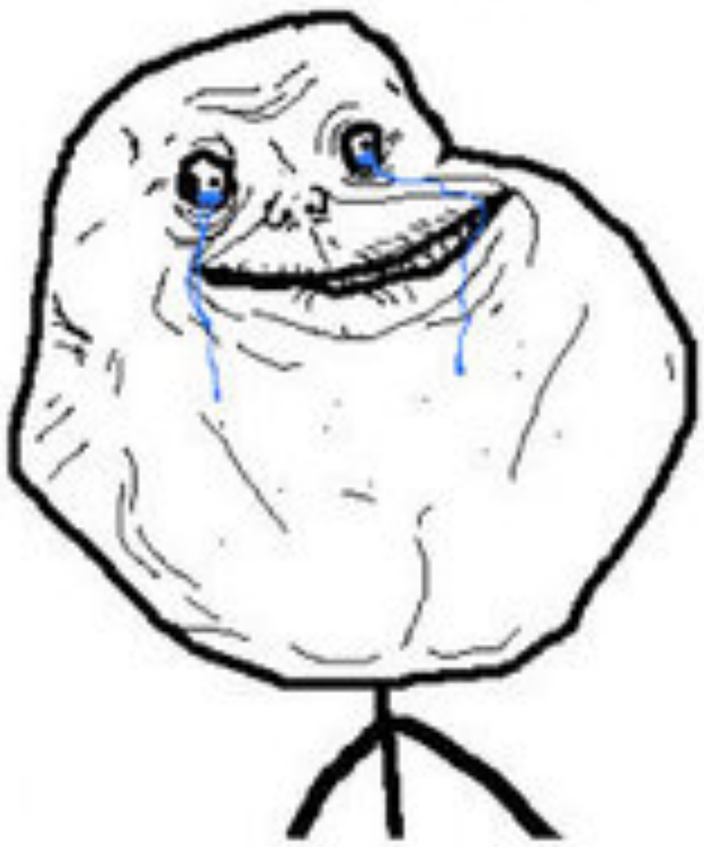
Sequencing human genomes was once very laborious and expensive



Now it is not.

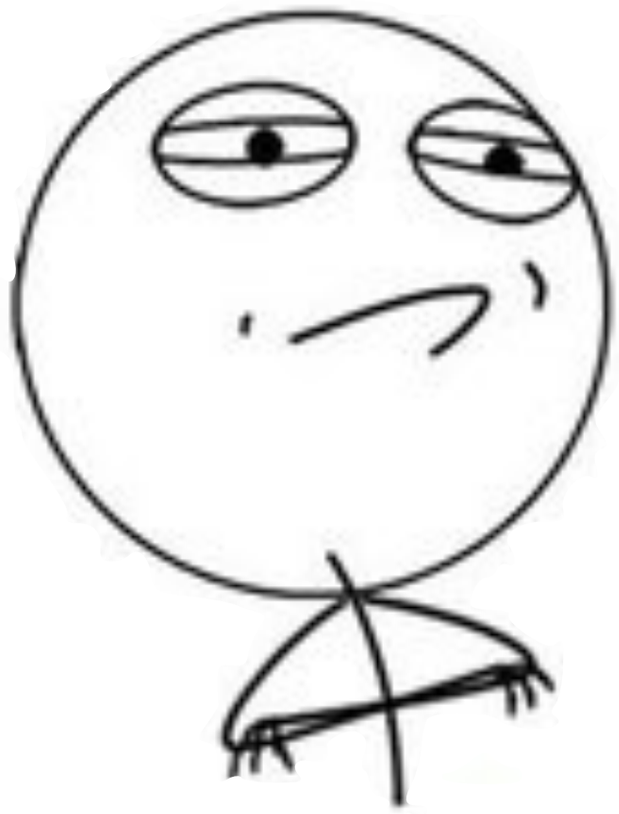
Right, Time to solve some diseases!

Motivation: Sequencing genomes? Easy. Understanding them? Hard.



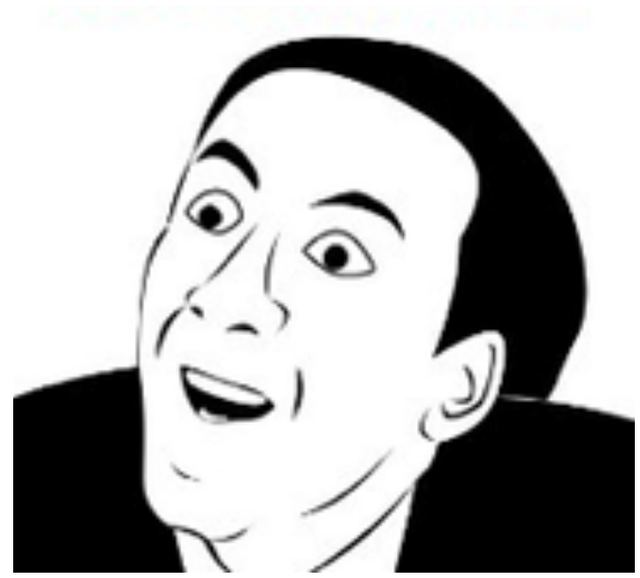
Pre-2008 sadness

Sequencing human genomes was once very laborious and expensive



Now it is not.

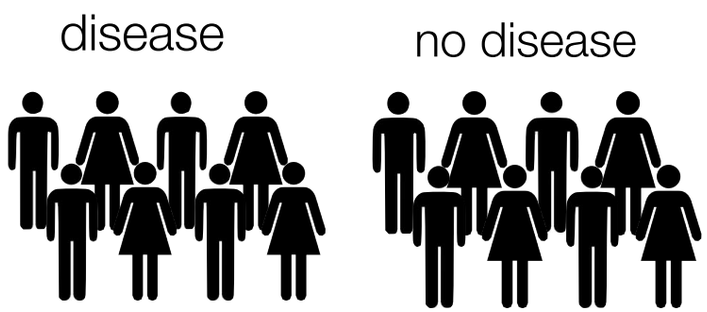
Right, Time to solve some diseases!

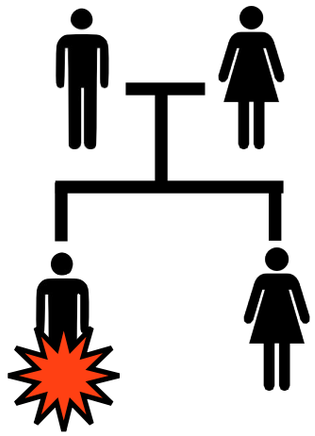


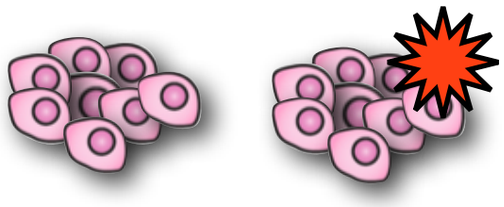
Okay, guy.

It's...complicated.

Typical genetics study designs

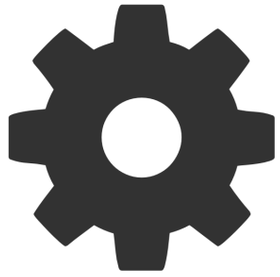



family studies



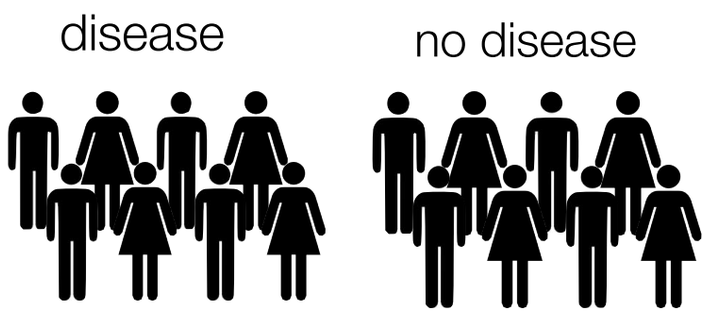
cancer genomics

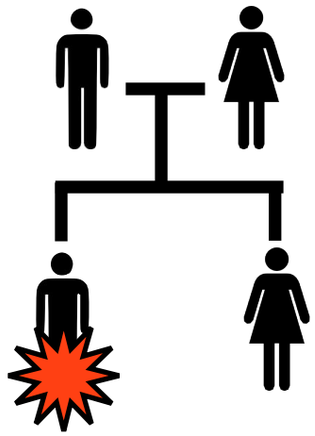
Sequence the genome of every sample in the study

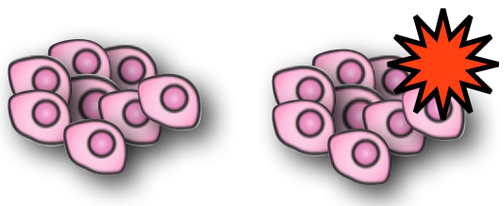


Align DNA.
Remove artifacts.
Find genetic variants.

Typical genetics study designs

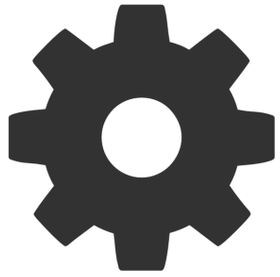



family studies



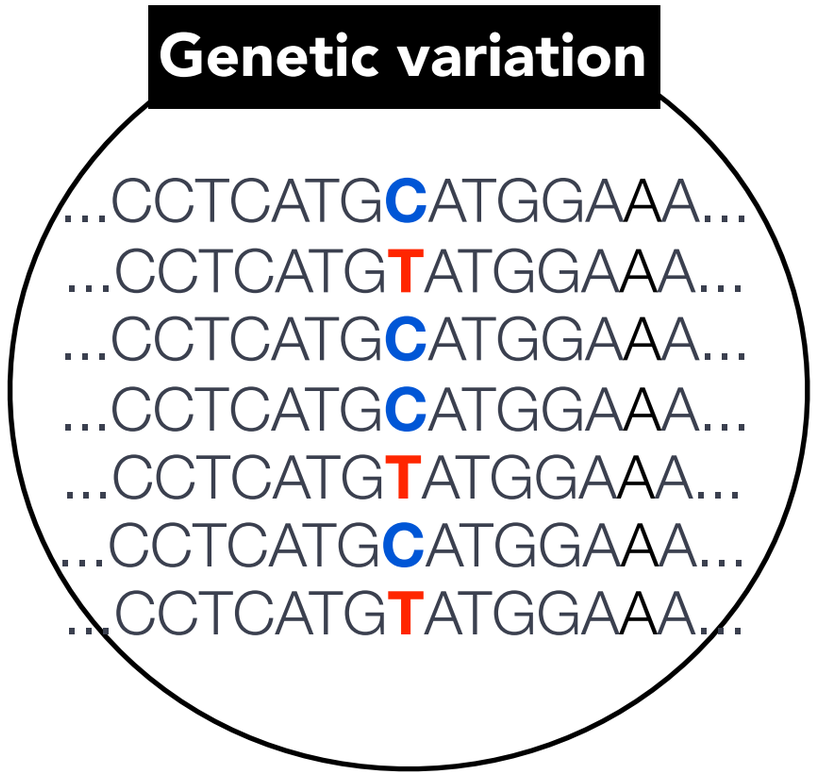
cancer genomics


Sequence the genome of every sample in the study

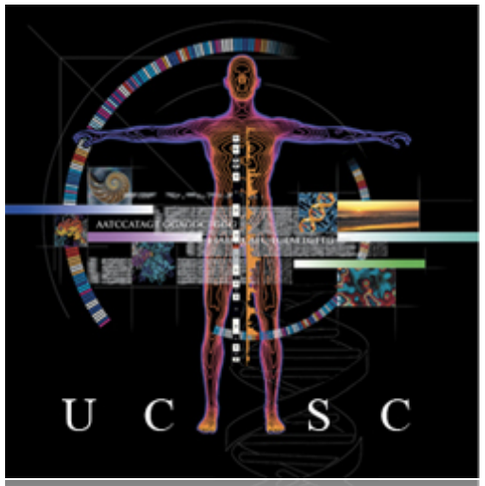

Align DNA.
Remove artifacts.
Find genetic variants.

Interpret.
Prioritize.
Repeat. Like, a lot.

Analytical challenges: data integration

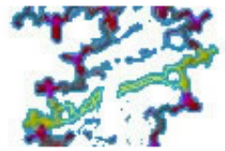


Analytical challenges: data integration



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

dbSNP
Short Genetic Variations



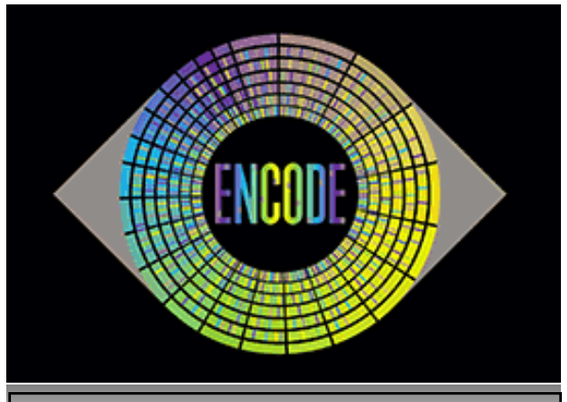
ClinVar

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

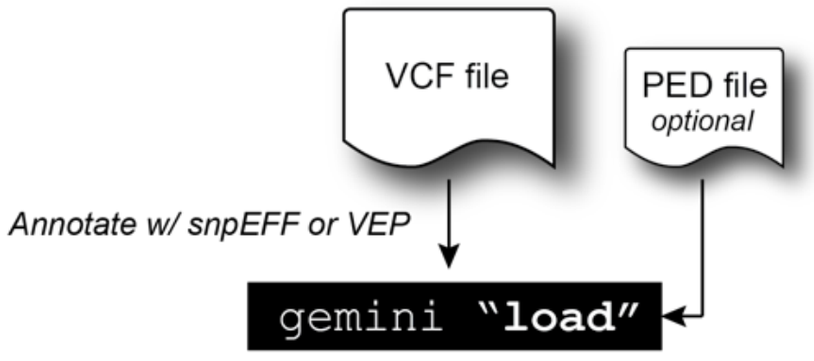
Pfam

KEGG Kyoto Encyclopedia of Genes and Genomes

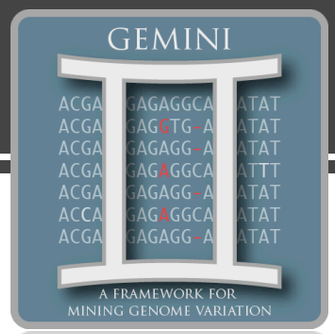
HGMD®

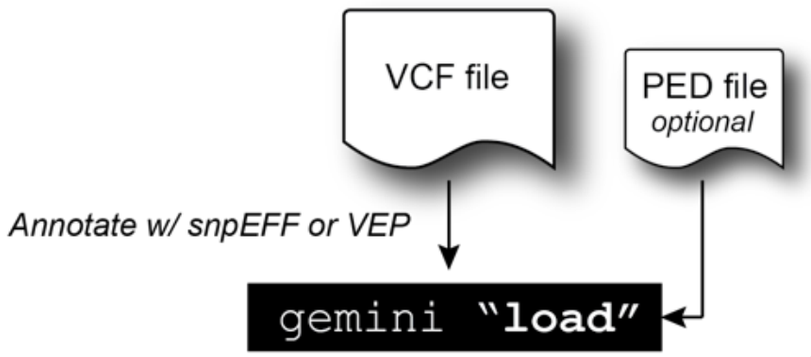
Human Protein Reference Database

The GEMINI framework



The GEMINI framework





cyvcf, pysam, bx-python, tabix

Parallelized loading on SGE, Torque, LFS with ***IPython.parallel***

Annotation source	Variants Table
From VCF	**Core**: chrom, ref. allele, alt. allele, id, qual, filter, ...
From VCF	**Variant info**: depth, strand bias, allele balance, ...
Computed	**Statistics**: type, call rate, Pi, allele freq., HWE, ...
snpEff, VEP, KEGG*, HPRD*	**Gene**: gene, transcript, impact, LoF, SIFT, pathway, ...
1000G, dbSNP, ESP, HapMap	**Population**: rsId, ESP and 1000G allele freq., recomb.
ClinVar	**Disease**: OMIM, clinical significance, disease, ID
UCSC	**Genome**: Conservation, RptMasker, CpG, SegDup...
UCSC	**Mappability**: gaps; Illumina, SOLiD, Ion mappability
ENCODE	**Regulation**: TF binding, DNase1, chrom. segment.
User defined	**Custom**: New columns based upon overlaps between variants and researcher-defined genome annotations.
Computed	**Genotypes**: genotype, type (e.g., HET), phase, depth, number of hets, hom_ref, hom_alt, unknown, etc. **Individual samples**: *gts.sample1, gt_types.sample2*

Variant Impacts Table

Variant impacts for each gene/transcript

Samples Table

Sample Id, sex, phenotype, relatives...
One entry / sample in VCF / PED file.

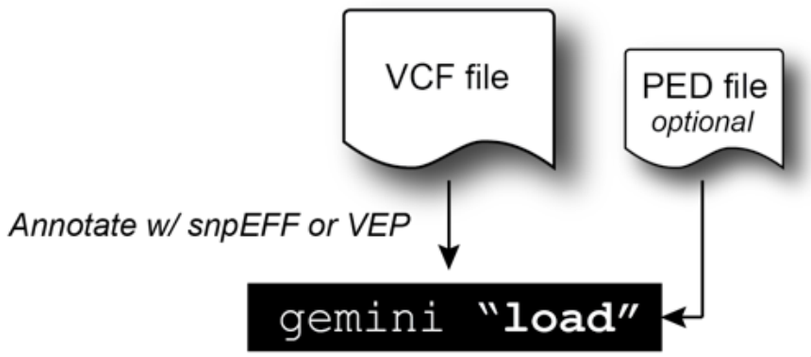
Resources Table

The name and version of all annotation files used by GEMINI.

Version Table

Tracks the GEMINI software version that was used to create the database.

The GEMINI framework



cyvcf, pysam, bx-python, tabix

Parallelized loading on SGE, Torque, LFS with ***IPython.parallel***

User-defined annotations

Annotation source	Variants Table
From VCF	**Core**: chrom, ref. allele, alt. allele, id, qual, filter, ...
From VCF	**Variant info**: depth, strand bias, allele balance, ...
Computed	**Statistics**: type, call rate, Pi, allele freq., HWE, ...
snpEff, VEP, KEGG*, HPRD*	**Gene**: gene, transcript, impact, LoF, SIFT, pathway, ...
1000G, dbSNP, ESP, HapMap	**Population**: rsId, ESP and 1000G allele freq., recomb.
ClinVar	**Disease**: OMIM, clinical significance, disease, ID
UCSC	**Genome**: Conservation, RptMasker, CpG, SegDup...
UCSC	**Mappability**: gaps; Illumina, SOLiD, Ion mappability
ENCODE	**Regulation**: TF binding, DNase1, chrom. segment.
User defined	**Custom**: New columns based upon overlaps between variants and researcher-defined genome annotations.
Computed	**Genotypes**: genotype, type (e.g., HET), phase, depth, number of hets, hom_ref, hom_alt, unknown, etc. **Individual samples**: *gts.sample1, gt_types.sample2*

Variant Impacts Table

Variant impacts for each gene/transcript

Samples Table

Sample Id, sex, phenotype, relatives...
One entry / sample in VCF / PED file.

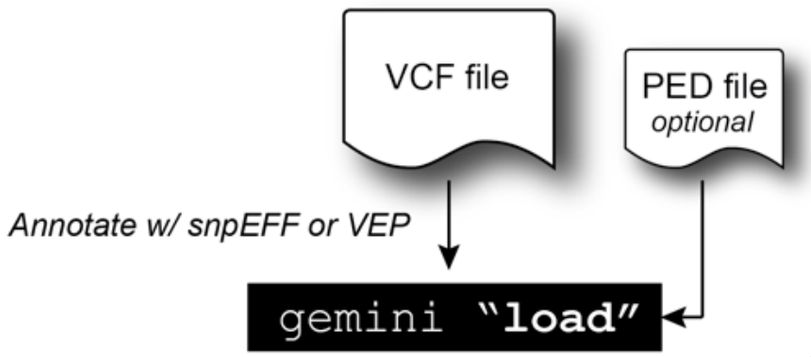
Resources Table

The name and version of all annotation files used by GEMINI.

Version Table

Tracks the GEMINI software version that was used to create the database.

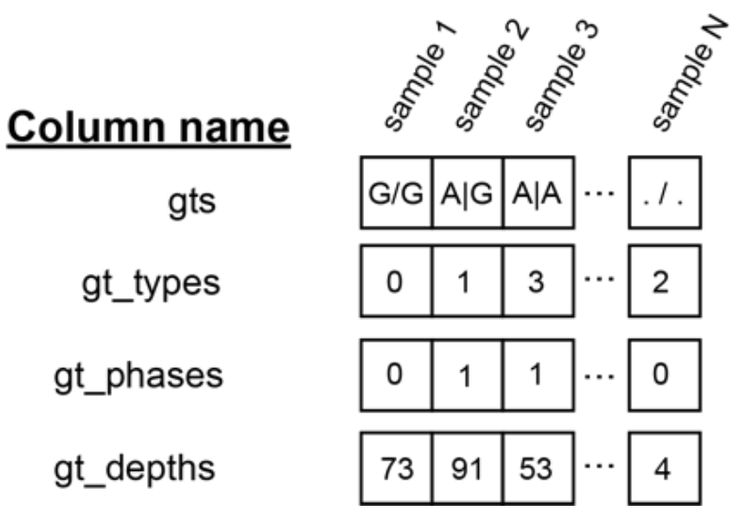
The GEMINI framework



cyvcf, pysam, bx-python, tabix

Parallelized loading on SGE, Torque, LFS with ***IPython.parallel***

User-defined annotations



Access to (and filtering upon) individual genotypes (compressed NUMPY arrays stored as SQLite BLOBs in DB)

Annotation source	Variants Table
From VCF	**Core**: chrom, ref. allele, alt. allele, id, qual, filter, ...
From VCF	**Variant info**: depth, strand bias, allele balance, ...
Computed	**Statistics**: type, call rate, Pi, allele freq., HWE, ...
snpEff, VEP, KEGG*, HPRD*	**Gene**: gene, transcript, impact, LoF, SIFT, pathway, ...
1000G, dbSNP, ESP, HapMap	**Population**: rsId, ESP and 1000G allele freq., recomb.
ClinVar	**Disease**: OMIM, clinical significance, disease, ID
UCSC	**Genome**: Conservation, RptMasker, CpG, SegDup...
UCSC	**Mappability**: gaps; Illumina, SOLiD, Ion mappability
ENCODE	**Regulation**: TF binding, DNase1, chrom. segment.
User defined	**Custom**: New columns based upon overlaps between variants and researcher-defined genome annotations.
Computed	**Genotypes**: genotype, type (e.g., HET), phase, depth, number of hets, hom_ref, hom_alt, unknown, etc. **Individual samples**: *gts.sample1, gt_types.sample2*

Variant Impacts Table

Variant impacts for each gene/transcript

Samples Table

Sample Id, sex, phenotype, relatives...
One entry / sample in VCF / PED file.

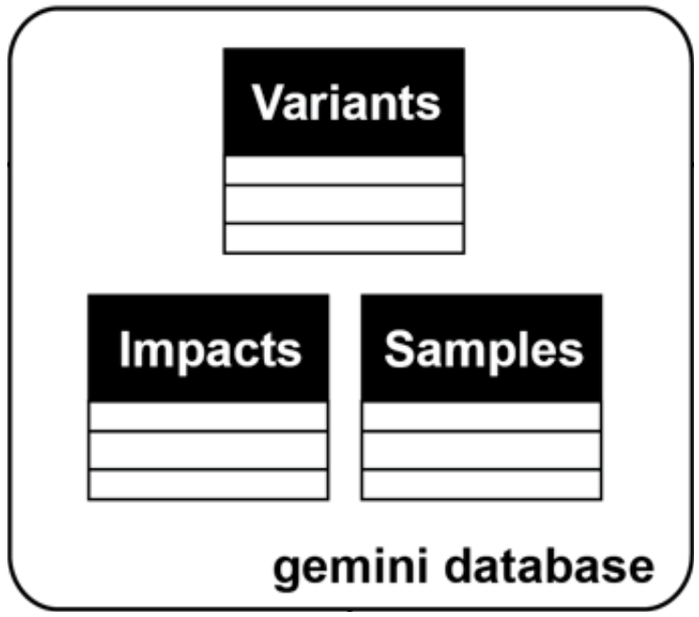
Resources Table

The name and version of all annotation files used by GEMINI.

Version Table

Tracks the GEMINI software version that was used to create the database.

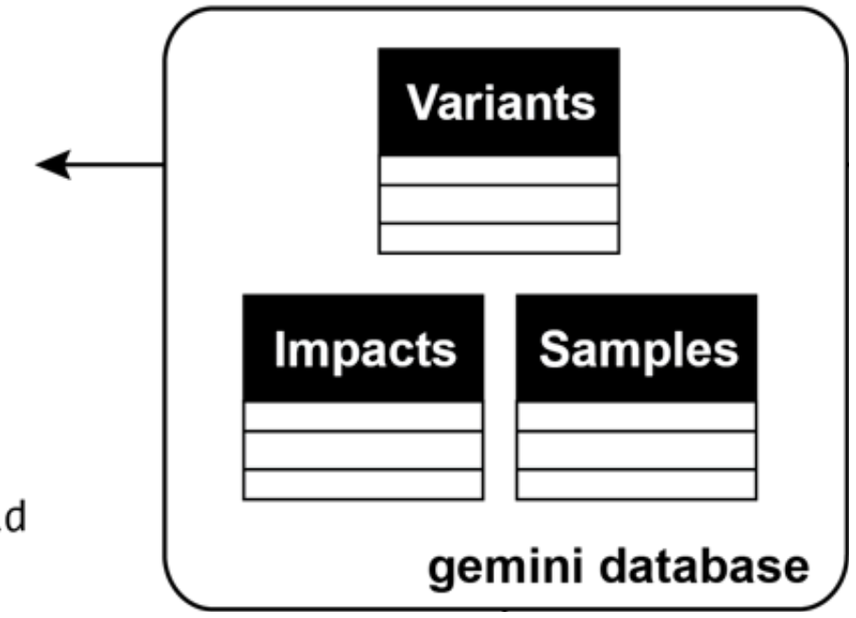
Mining variation with GEMINI



Mining variation with GEMINI

ad hoc data exploration

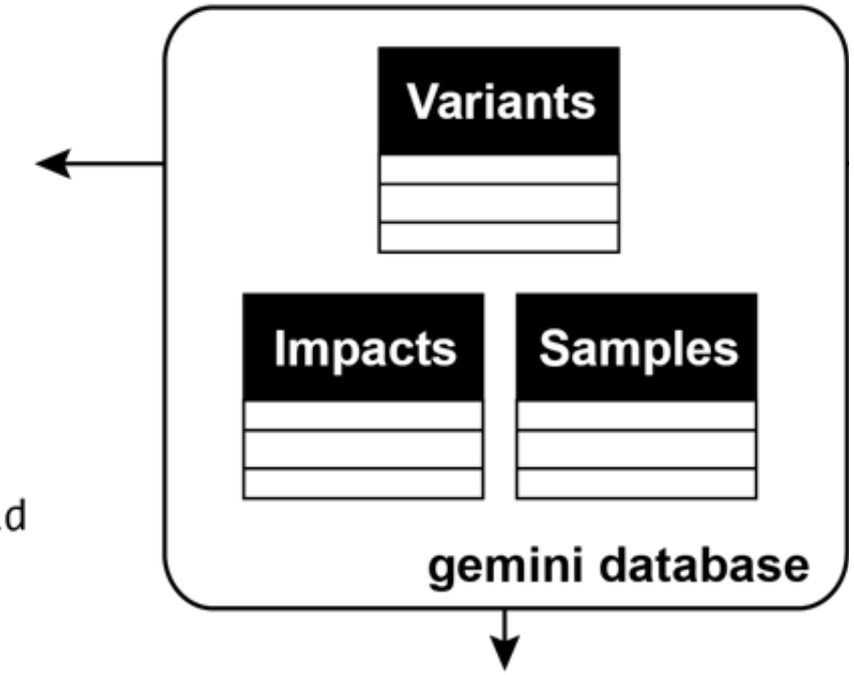
```
gemini query
--query
"select chrom, start, end,
        ref, alt, gene,
        impact, aaf, gts.kid
 from variants
 where in_dbsnp = 0
 and aaf < 0.01
 and is_lof  = 1
 and my_disease_regions = 1"
--gt-filter
"gt_types.mom == HET
 and
 gt_types.dad == HET
 and
 gt_types.proband == HOM_ALT"
```



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"
```



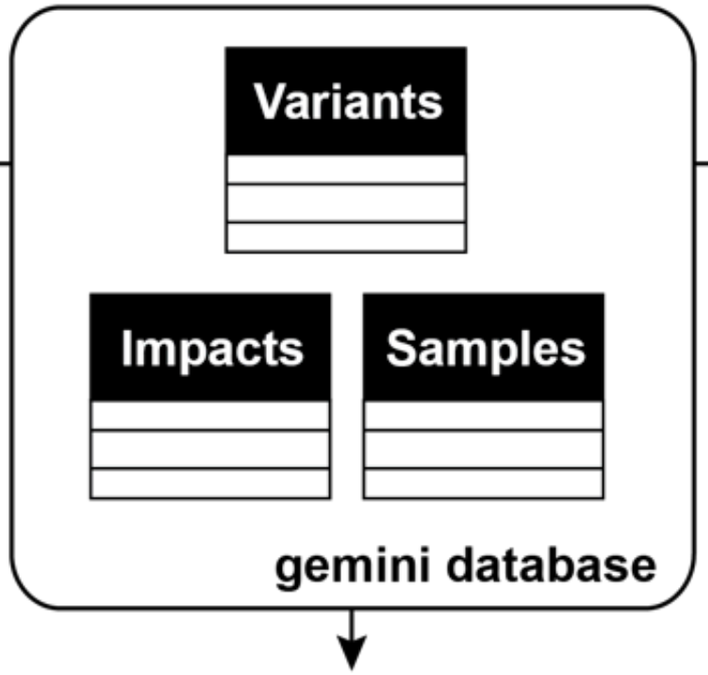
Built-in tools and analyses

Tool	*Description*
region	*extract variants spec. genomic intervals or genes*
stats	*compute variant statisics (SFS, Ts/Tv, counts, etc.)*
annotate	*add new columns based on custom annotations*
windower	*compute variant statistics across genome "windows"*
comp_hets	*identify candidate compund 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 statisics (SFS, Ts/Tv, counts, etc.)*
`annotate`	*add new columns based on custom annotations*
`windower`	*compute variant statistics across genome "windows"*
`comp_hets`	*identify candidate compund 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 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 into a NUMPY array
    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 into a NUMPY array
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 into a NUMPY array
genotype_types = row.gt_types

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

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

Queries scale to studies with 1000s of samples

Experiment	Illumina "platinum" Trio	1046 samples
Return all novel variants `select * from variants where in_dbsnp = 0`	24 sec (N=345,028)	11 sec (N=87,939)
Return all loss-of-func. variants `select * from variants where is_lof = 1`	2 sec (N=1,126)	177 sec (N=13,049)

Return all rare, loss-of-func. variants
`select * from variants`
`where is_lof = 1 and aaf < 0.01`
	2 sec
	(N=112)
		152 sec
		(N=12,683)

Filtering variants based on sample genotype criteria
`select * from variants where is_lof = 1"`
`--gt-filter "gt_types.NA12878 == HET"`
	2 sec
	(N=487)
		194 sec
		(N=384)

Find somatic mutations in cancer in 5 minutes

```
##############################################################
# load a VCF for a tumor / normal pair into gemini.
# - use 4 cores
# - assume VCF has been annotated with snpEff
##############################################################
$ gemini load -v tumor-normal.vcf -t snpEff --cores 4 tumor-normal.vcf.db


##############################################################
# Identify novel somatic mutations in the tumor that are likely to
# impact gene function.
##############################################################
$ gemini query -q "select chrom, start, end, ref, alt, type, gene \
                   from variants
                   where impact_severity !='LOW'
                   and in_dbsnp = 0" \
               --gt-filter "gt_types.TUMOR == HET and
                            gt_types.NORMAL == HOM_REF and
                            gt_alt_depths.NORMAL == 0" \
               tumor-normal.vcf.db
```

https://gist.github.com/arq5x/5769167

Summary

- Flexible framework for mining genetic variation.
- Integrates important genome annotations.
- Query access to individual genotypes
- Extensible for new analyses and tool dev.

- Free. Open source. github.com/arq5x/gemini
- Well documented. gemini.readthedocs.org
- Extensible, portable, & reproducible
- *In press* at PLoS Computational Biology