VariantPlaner

Quering many variant without cluster

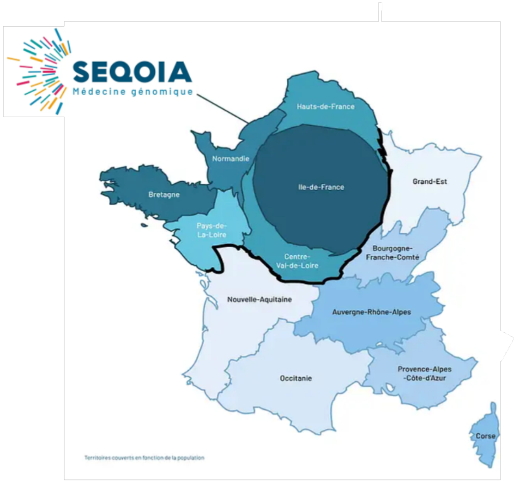
Pierre Marijon, Sacha Schutz

GCS SeqOIA

September 20, 2023

SeqOIA

1082 Prescribers / **40** Prescription assistant



Territoires couverts en fonction de la population

SEQOIA Médecine génomique LBM

Biologist in charge – Pierre BLANC

SEQOIA Médecine génomique GEN

Pierre BLANC

Wet-Lab - Logistics, Reception, Extraction, STHD (Integragen)

SEQOIA Médecine génomique IT

Alban LERMINE

Dry Lab - Bioinformatics

SEQOIA Médecine génomique LMG

Pierre BLANC, Boris KEREN, Jennifer WONG
Damien VASSEUR, Emmanuelle CLAPPIER,

Interpretation of exams
190 Biologists

178 Biologistes

Accreditation ISO 15189 (GC07 & GS07) filed on October 2021

Agreements

16 agreements with non-GCS establishments

GCS SeqOIA

ASSISTANCE PUBLIQUE HÔPITAUX DE PARIS

institut Curie

GUSTAVE ROUSSY CANCER CAMPUS GRAND PARIS

SAKE: SeqOIA dAta laKE

Which sample has:

- denovo variant in these gene/region

SAKE: SeqOIA dAta laKE

Which sample has:

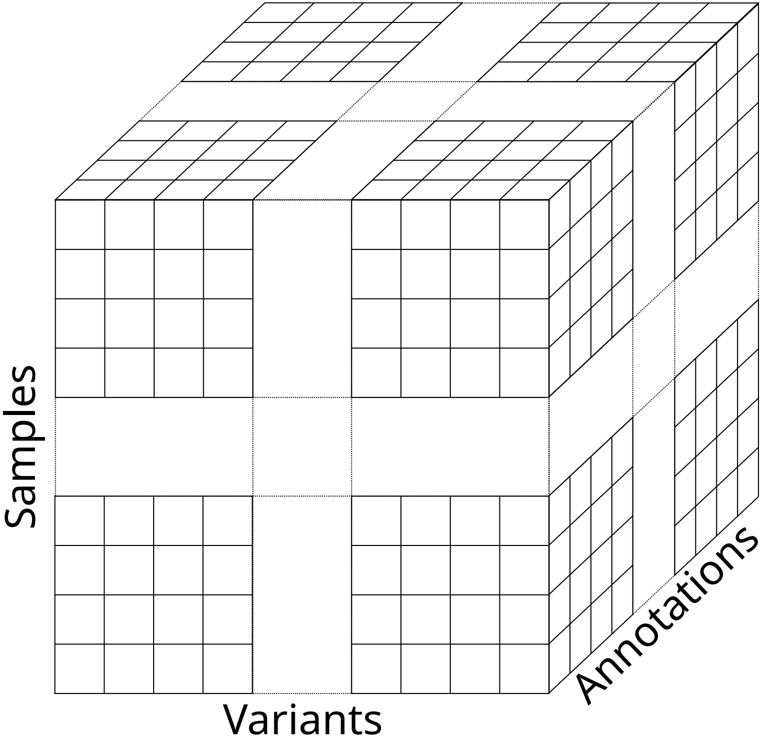
- denovo variant in these gene/region
- variants impact splicing in these gene

SAKE: SEQOIA DATA LAKE

Which sample has:

- denovo variant in these gene/region
- variants impact splicing in these gene
- variants with clinvar state change

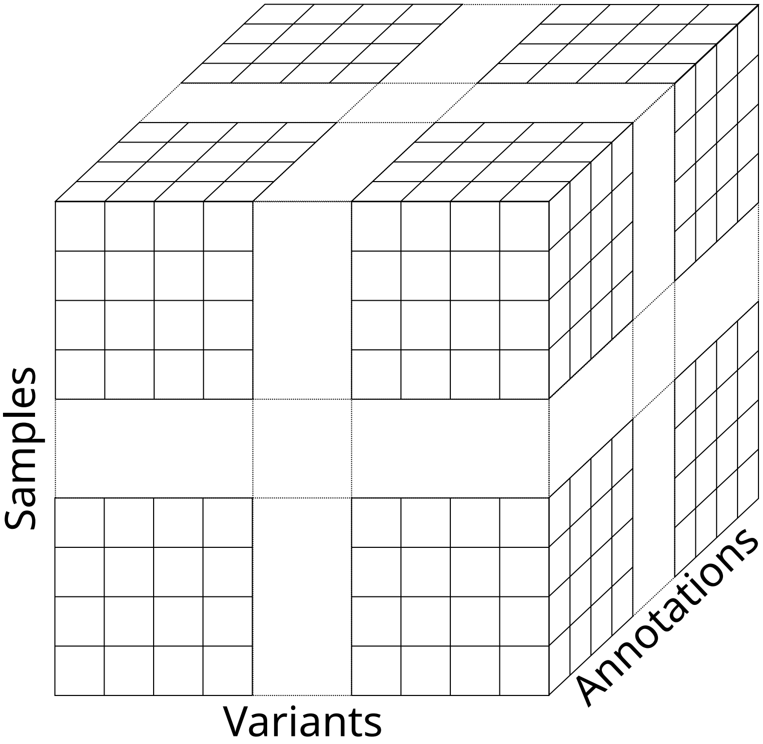
SAKE: SEQOIA DATA LAKE



Which sample has:

- denovo variant in these gene/region
- variants impact splicing in these gene
- variants with clinvar state change

SAKE: SEQOIA DATA LAKE



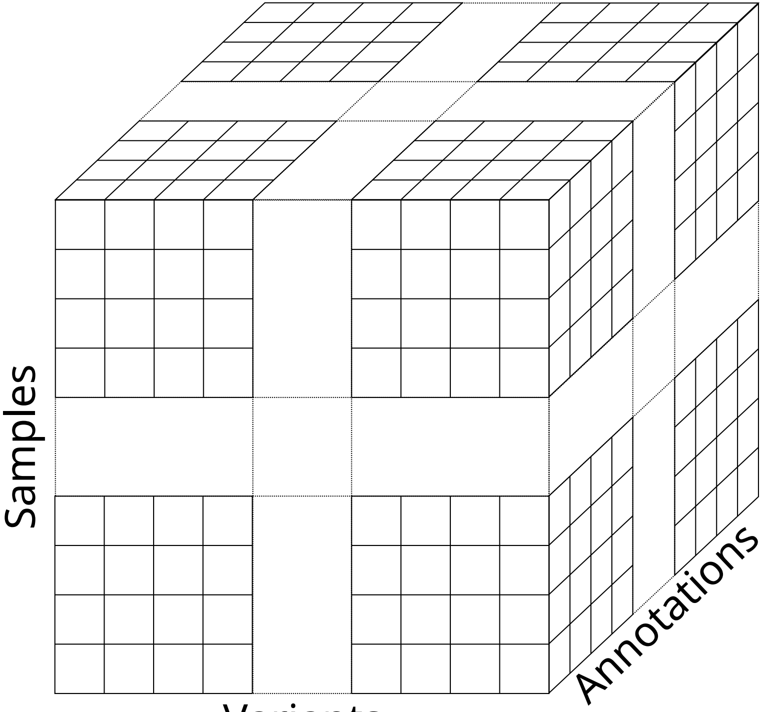
Which sample has:

- denovo variant in these gene/region
- variants impact splicing in these gene
- variants with clinvar state change

Matrix size:

September 1st values

SAKE: SeqOIA dAta laKE



Which sample has:

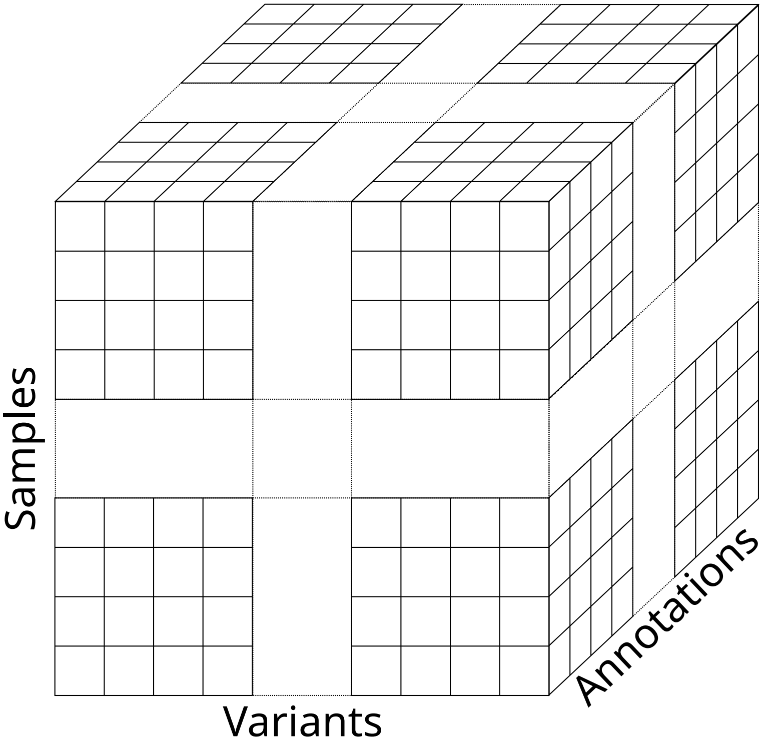
- denovo variant in these gene/region
- variants impact splicing in these gene
- variants with clinvar state change

Matrix size:

- 24,500 samples

September 1st values

SAKE: SEQOIA DATA LAKE



Which sample has:

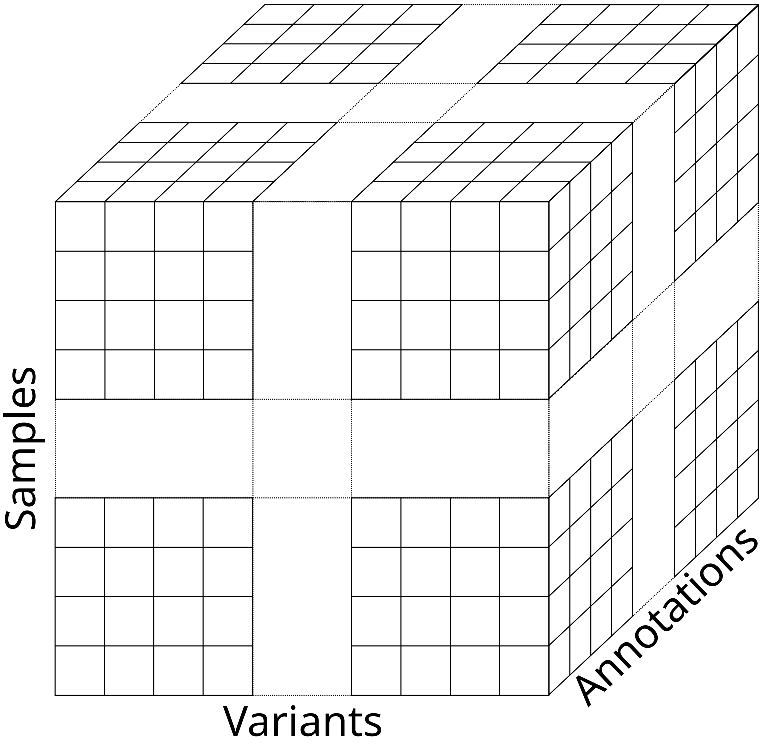
- denovo variant in these gene/region
- variants impact splicing in these gene
- variants with clinvar state change

Matrix size:

- 24,500 samples
- 350,000,000 unique variants

September 1st values

SAKE: SEQOIA DATA LAKE



Which sample has:

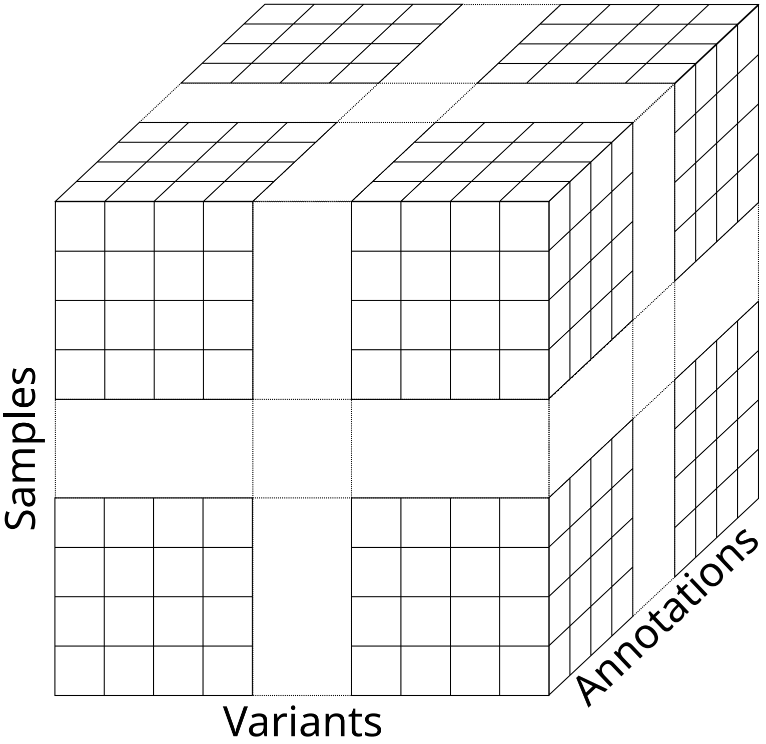
- denovo variant in these gene/region
- variants impact splicing in these gene
- variants with clinvar state change

Matrix size:

- 24,500 samples
- 350,000,000 unique variants
- annotations: genotype, coverage, gnomad, snpeff, spliceAI, clinvar,

September 1st values

SAKE: SEQOIA DATA LAKE



Which sample has:

- denovo variant in these gene/region
- variants impact splicing in these gene
- variants with clinvar state change

Matrix size:

- 24,500 samples
- 350,000,000 unique variants
- annotations: genotype, coverage, gnomad, snpeff, spliceAI, clinvar,
- sparse matrix: 98.5 % variants have 0/0 genotype

September 1st values

SAKE: SeqOIA dAta laKE

```
grep:
```

SAKE: SEQOIA DATA LAKE

`grep`:

- 4.9 Tb of uncompress unannoted vcf

SAKE: SEQOIA DATA LAKE

`grep`:

- 4.9 Tb of uncompress unannoted vcf
- SeqOIA best-ever throughput read: 6Gb/s

SAKE: SeqOIA dAta laKE

`grep`:

- 4.9 Tb of uncompress unannoted vcf
- SeqOIA best-ever throughput read: 6Gb/s
- 864 s → 14 minutes

SAKE: SeqOIA dAta laKE

`grep`:

- 4.9 Tb of uncompress unannoted vcf
- SeqOIA best-ever throughput read: 6Gb/s
- 864 s →14 minutes
- `wc -l *.vcf` →2 hours

SAKE: SeqOIA dAta laKE

`grep`:

- 4.9 Tb of uncompress unannoted vcf
- SeqOIA best-ever throughput read: 6Gb/s
- 864 s →14 minutes
- `wc -l *.vcf` →2 hours

SAKE: SeqOIA dAta laKE

`grep`:

- 4.9 Tb of uncompress unannoted vcf
- SeqOIA best-ever throughput read: 6Gb/s
- 864 s →14 minutes
- `wc -l *.vcf` →2 hours

SAKE: SeqOIA dAta laKE

`grep`:

- 4.9 Tb of uncompress unannoted vcf
- SeqOIA best-ever throughput read: 6Gb/s
- 864 s →14 minutes
- `wc -l *.vcf` →2 hours

hail

Parquet

DuckDB

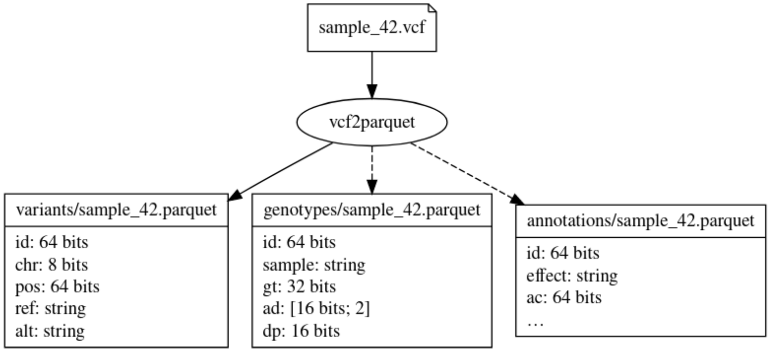
polars

VariantPlaner

- Part of the generalisable SAKE generation pipeline
- Python module and command line
- Based on pola-rs

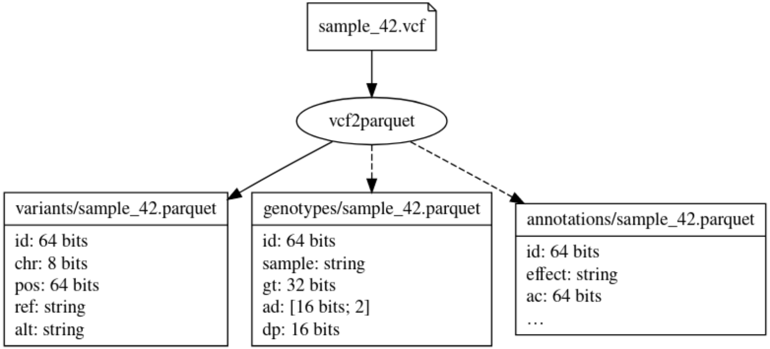
VariantPlaner

vcf2parquet



VariantPlaner

vcf2parquet



sample_42.vcf

vcf2parquet

variants/sample_42.parquet

id: 64 bits
chr: 8 bits
pos: 64 bits
ref: string
alt: string

genotypes/sample_42.parquet

id: 64 bits
sample: string
gt: 32 bits
ad: [16 bits; 2]
dp: 16 bits

annotations/sample_42.parquet

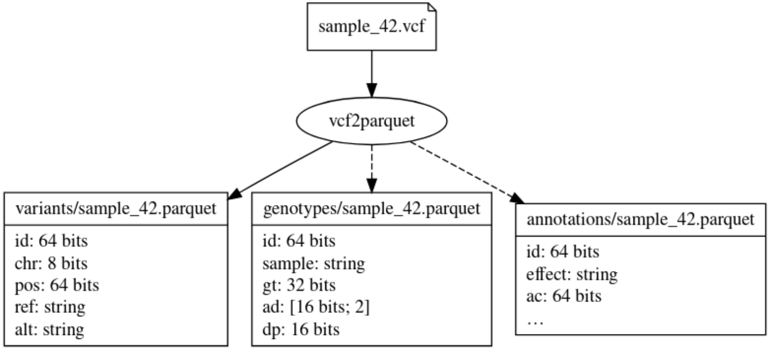
id: 64 bits
effect: string
ac: 64 bits
…

id computation:

- v0.1: hash of `chr`, `pos`, `ref`, `alt`

VariantPlaner

vcf2parquet



sample_42.vcf

vcf2parquet

variants/sample_42.parquet

id: 64 bits
chr: 8 bits
pos: 64 bits
ref: string
alt: string

genotypes/sample_42.parquet

id: 64 bits
sample: string
gt: 32 bits
ad: [16 bits; 2]
dp: 16 bits

annotations/sample_42.parquet

id: 64 bits
effect: string
ac: 64 bits
…

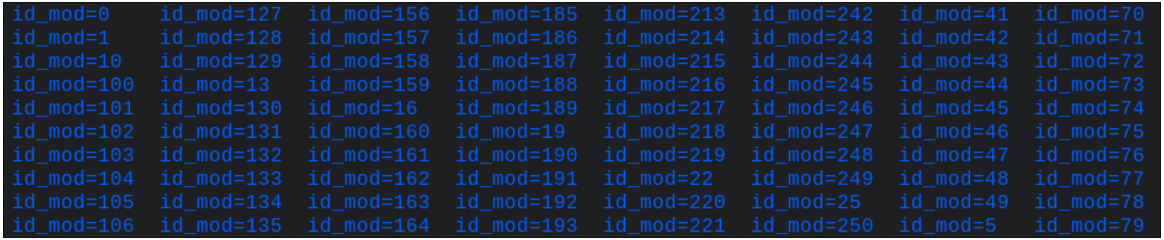
id computation:

- v0.1: hash of `chr, pos, ref, alt`
- V0.2 if $len(ref) + len(alt)$:
  - $\leq 13 \rightarrow$ perfect hash ($\approx$ 96%)
  - $> 13 \rightarrow$ v0.1 hash

VariantPlaner

Genotype Organisation

```
variantplaner struct [-i sample/{}.parquet] genotypes -o genotypes/variants/
```

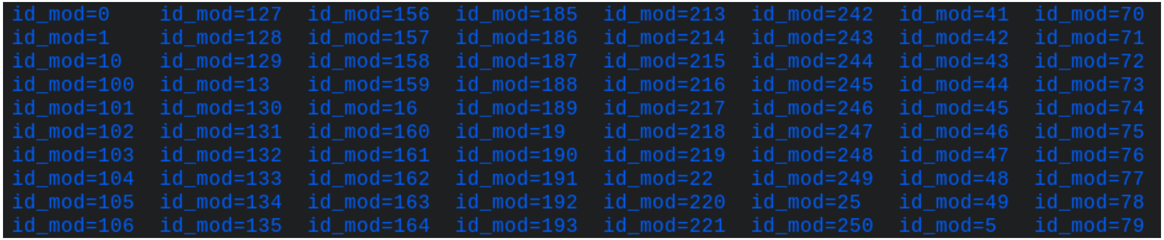


```
id_mod=0    id_mod=127  id_mod=156  id_mod=185  id_mod=213  id_mod=242  id_mod=41  id_mod=70
id_mod=1    id_mod=128  id_mod=157  id_mod=186  id_mod=214  id_mod=243  id_mod=42  id_mod=71
id_mod=10   id_mod=129  id_mod=158  id_mod=187  id_mod=215  id_mod=244  id_mod=43  id_mod=72
id_mod=100  id_mod=13   id_mod=159  id_mod=188  id_mod=216  id_mod=245  id_mod=44  id_mod=73
id_mod=101  id_mod=130  id_mod=16   id_mod=189  id_mod=217  id_mod=246  id_mod=45  id_mod=74
id_mod=102  id_mod=131  id_mod=160  id_mod=19   id_mod=218  id_mod=247  id_mod=46  id_mod=75
id_mod=103  id_mod=132  id_mod=161  id_mod=190  id_mod=219  id_mod=248  id_mod=47  id_mod=76
id_mod=104  id_mod=133  id_mod=162  id_mod=191  id_mod=22   id_mod=249  id_mod=48  id_mod=77
id_mod=105  id_mod=134  id_mod=163  id_mod=192  id_mod=220  id_mod=25   id_mod=49  id_mod=78
id_mod=106  id_mod=135  id_mod=164  id_mod=193  id_mod=221  id_mod=250  id_mod=5   id_mod=79
```

VariantPlaner

Genotype Organisation

```
variantplaner struct [-i sample/{}.parquet] genotypes -o genotypes/variants/
```



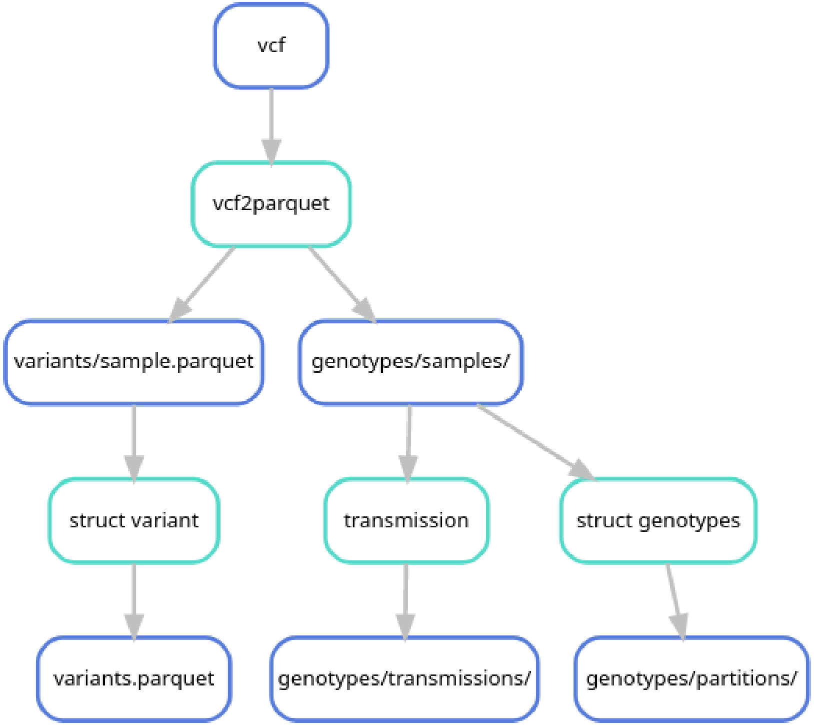
```
id_mod=0    id_mod=127  id_mod=156  id_mod=185  id_mod=213  id_mod=242  id_mod=41  id_mod=70
id_mod=1    id_mod=128  id_mod=157  id_mod=186  id_mod=214  id_mod=243  id_mod=42  id_mod=71
id_mod=10   id_mod=129  id_mod=158  id_mod=187  id_mod=215  id_mod=244  id_mod=43  id_mod=72
id_mod=100  id_mod=13   id_mod=159  id_mod=188  id_mod=216  id_mod=245  id_mod=44  id_mod=73
id_mod=101  id_mod=130  id_mod=16   id_mod=189  id_mod=217  id_mod=246  id_mod=45  id_mod=74
id_mod=102  id_mod=131  id_mod=160  id_mod=19   id_mod=218  id_mod=247  id_mod=46  id_mod=75
id_mod=103  id_mod=132  id_mod=161  id_mod=190  id_mod=219  id_mod=248  id_mod=47  id_mod=76
id_mod=104  id_mod=133  id_mod=162  id_mod=191  id_mod=22   id_mod=249  id_mod=48  id_mod=77
id_mod=105  id_mod=134  id_mod=163  id_mod=192  id_mod=220  id_mod=25   id_mod=49  id_mod=78
id_mod=106  id_mod=135  id_mod=164  id_mod=193  id_mod=221  id_mod=250  id_mod=5   id_mod=79
```

```
variantplaner transmission -i sample/42.parquet -p 42.ped -m
transmissions/42.parquet
```

id	index gt	mother gt	father gt	origin
15225413595434247130	2	0	0	200
12902036237217108692	1	1	0	110
2909135909504078072	1	0	2	102
15241688863478200138	2	3	3	233

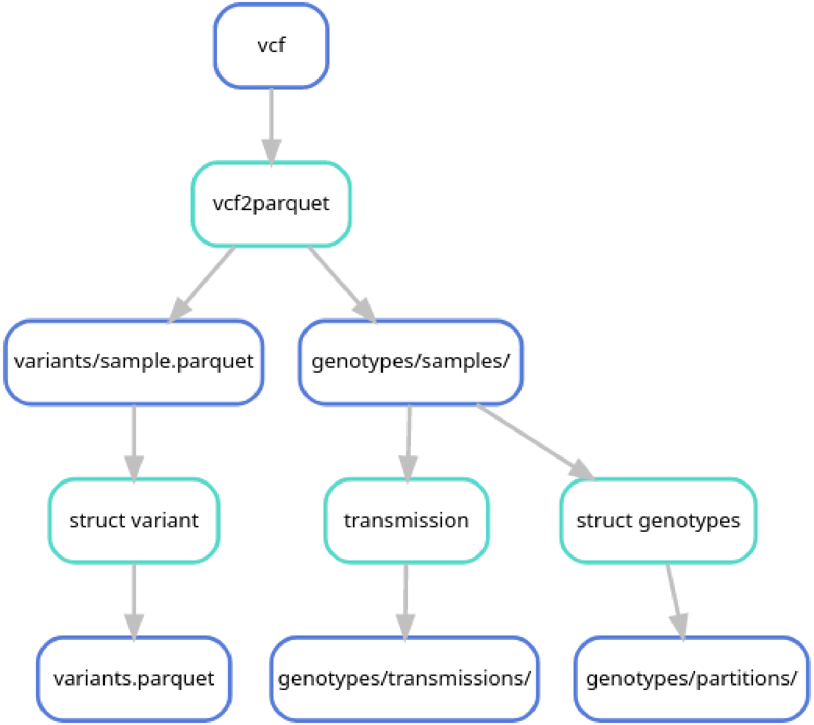
VariantPlaner

Performance: build SAKE



VariantPlaner

Performance: build SAKE



`vcf2parquet`: ~30s per sample

`struct variant`: ~4h 30m for all

`transmission`: ~50s per sample

`struct genotype`: ~2h 50m for all

VariantPlaner

Performance: interrogate SAKE

```
SELECT * FROM Variants WHERE chr="MT"
```

VariantPlaner

Performance: interrogate SAKE

```
SELECT * FROM Variants WHERE chr="MT"
```

Around a second

VariantPlaner

Performance: interrogate SAKE

```
SELECT * FROM Variants WHERE chr="MT"
```

Around a second

```
SELECT * FROM Variants as v JOIN SpliceAI as s ON v.id=s.id
JOIN Recurence as r ON v.id=r.id WHERE v.chr=13 and
v.pos > 6670360 and v.pos < 6694030 and r.ac < 10
```

VariantPlaner

Performance: interrogate SAKE

```
SELECT * FROM Variants WHERE chr="MT"
```

Around a second

```
SELECT * FROM Variants as v JOIN SpliceAI as s ON v.id=s.id
JOIN Recurence as r ON v.id=r.id WHERE v.chr=13 and
v.pos > 6670360 and v.pos < 6694030 and r.ac < 10
```

Around a minutes

VariantPlaner

Performance: interrogate SAKE

```sql
SELECT * FROM Variants WHERE chr="MT"
```

Around a second

```sql
SELECT * FROM Variants as v JOIN SpliceAI as s ON v.id=s.id
JOIN Recurence as r ON v.id=r.id WHERE v.chr=13 and
v.pos > 6670360 and v.pos < 6694030 and r.ac < 10
```

Around a minutes

```sql
SELECT * FROM selected_variant as sv JOIN Genotypes as g ON
sv.id=g.id WHERE g.vaf > 0.1
```

VariantPlaner

Performance: interrogate SAKE

```sql
SELECT * FROM Variants WHERE chr="MT"
```

Around a second

```sql
SELECT * FROM Variants as v JOIN SpliceAI as s ON v.id=s.id
JOIN Recurence as r ON v.id=r.id WHERE v.chr=13 and
v.pos > 6670360 and v.pos < 6694030 and r.ac < 10
```

Around a minutes

```sql
SELECT * FROM selected_variant as sv JOIN Genotypes as g ON
sv.id=g.id WHERE g.vaf > 0.1
```

Around 20 minutes

VariantPlaner

Performance: interrogate SAKE

```
SELECT * FROM Variants WHERE chr="MT"
```

Around a second

```
SELECT * FROM Variants as v JOIN SpliceAI as s ON v.id=s.id
JOIN Recurence as r ON v.id=r.id WHERE v.chr=13 and
v.pos > 6670360 and v.pos < 6694030 and r.ac < 10
```

Around a minutes

```
SELECT * FROM selected_variant as sv JOIN Genotypes as g ON
sv.id=g.id WHERE g.vaf > 0.1
```

Around 20 minutes

```
SELECT * FROM selected_variant_and_genotypes as svg JOIN Transmission as t ON
svg.sample=t.sample WHERE origin = 200 and origin = 100
```

VariantPlaner

Performance: interrogate SAKE

```
SELECT * FROM Variants WHERE chr="MT"
```

Around a second

```
SELECT * FROM Variants as v JOIN SpliceAI as s ON v.id=s.id
JOIN Recurence as r ON v.id=r.id WHERE v.chr=13 and
v.pos > 6670360 and v.pos < 6694030 and r.ac < 10
```

Around a minutes

```
SELECT * FROM selected_variant as sv JOIN Genotypes as g ON
sv.id=g.id WHERE g.vaf > 0.1
```

Around 20 minutes

```
SELECT * FROM selected_variant_and_genotypes as svg JOIN Transmission as t ON
svg.sample=t.sample WHERE origin = 200 and origin = 100
```

Highly variable

Conclusion

VariantPlaner builds an efficient, queryable database of variants:

- With reasonable resources (190Gb of ram)
- Reduce disk usage (SAKE use 3.7Tb)
- Available as a python module and command line
- Open to suggestion and modification

CONCLUSION

VariantPlaner builds an efficient, queryable database of variants:

- With reasonable resources (190Gb of ram)
- Reduce disk usage (SAKE use 3.7Tb)
- Available as a python module and command line
- Open to suggestion and modification

natir/variantplaner

Conclusion

VariantPlaner builds an efficient, queryable database of variants:

- With reasonable resources (190Gb of ram)
- Reduce disk usage (SAKE use 3.7Tb)
- Available as a python module and command line
- Open to suggestion and modification

natir/variantplaner

"Ré-analyse périodique semi-automatisée en génétique constitutionnelle"
Friday morning at 9 hours by Alban Lermine:

References I