Relational Data Models for Genetic VCF data

Freie Universität Bozen
Libera Università di Bolzano
Università Liedia de Bulsan

Mohamed Sabri Hafidi, Ozan Kahramanoğulları, Anton Dignös, and Johann Gamper

{hmohamedsabri, okahramanogullari, anton.dignoes, johann.gamper}@unibz.it

Challenges

- Genomic data is growing exponentially, reaching petabyte volumes.
- The structure of VCF is compact but not optimized for advanced querying.
- Existing tools struggle with large-scale datasets and complex analyses.
- Merging VCF data with diverse biological sources remains limited.

Objectives

- Population-level storage
- Flexible SQL-based querying
- Optimized performance
- Seamless multi-omics integration

Related Work

- **Command-line tools**: Offer speed but lack flexibility. e.g., *BCFtools, Tabix.*
- **Libraries**: Enable custom analyses but require multiple tools for a complete workflow. e.g., *vcflib, cyvcf2.*
- **Database systems**: Scale well but vary in how they represent the VCF data. e.g., *TileDB-VCF, TheSNPpit.*
- Prior studies explore *relational* and *JSON-based* models but lack broad evaluations.
- We introduce and evaluate novel relational models for VCF data management.

The VCF File Structure

Meta-Information

```
##fileformat=VCFv4.5
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,species="Homo sapiens",taxonomy=x>
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
```

Header

Genotypes

Samples

	#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	NA00001	NA00002	NA00003
Variants	20	14370	rs6054257	G	A	29	PASS	NS=3;DP=14;AF=0.5;DB;H2	GT:GQ:DP:HQ	0|0:48:1:51,51	1|0:48:8:51,51	1/1:43:5:.,.
	20	17330	.	T	A	3	q10	NS=3;DP=11;AF=0.017	GT:GQ:DP:HQ	0|0:49:3:58,50	0|1:3:5:65,3	0/0:41:3
	20	1110696	rs6040355	A	G,T	67	PASS	NS=2;DP=10;AF=0.333,0.667;AA=T;DB	GT:GQ:DP:HQ	1|2:21:6:23,27	2|1:2:0:18,2	2/2:35:4
	20	1230237	.	T	.	47	PASS	NS=3;DP=13;AA=T	GT:GQ:DP:HQ	0|0:54:7:56,60	0|0:48:4:51,51	0/0:61:2
	20	1234567	microsat1	GTC	G,GTCT	50	PASS	NS=3;DP=9;AA=G	GT:GQ:DP	0/1:35:4	0/2:17:2	1/1:40:3

Fixed Fields

Relational Data Models

Wide: each row represents a variant and each column represents a sample

vcf_wide_samples_chunk_1

ln	na00001	na00002	na00003	...
1	0|0:48:1:51,51	1|0:48:8:51,51	1/1:43:5:.	
2	0|0:49:3:58,50	0|1:3:5:65,3	0/0:41:3	
3	1|2:21:6:23,27	2|1:2:0:18,2	2/2:35:4	...
4	0|0:54:7:56,60	0|0:48:4:51,51	0/0:61:2	
5	0/1:35:4	0/2:17:2	1/1:40:3	

Narrow: each row represents a variant-sample pair

vcf_narrow_samples

ln	s_id	genotypes
1	NA00001	0|0:48:1:51,51
1	NA00002	1|0:48:8:51,51
1	NA00003	1/1:43:5:.
2	NA00001	0|0:49:3:58,50
2	NA00002	0|1:3:5:65,3
2	NA00003	0/0:41:3
3	NA00001	1|2:21:6:23,27
3	NA00002	2|1:2:0:18,2
3	NA00003	2/2:35:4
...	...	...

Array Plain: store the genotypes in an array

vcf_array_indices

s_id	e_id
NA00001	1
NA00002	2
NA00003	3
...	...

vcf_array_genotypes

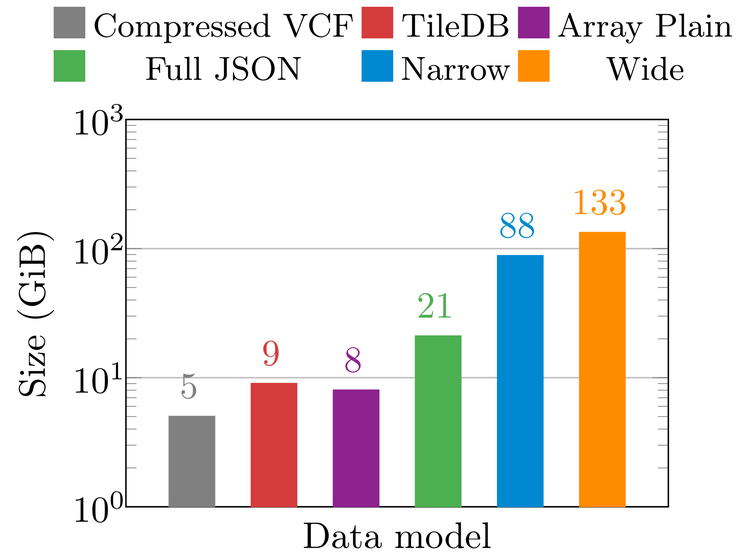
ln	genotypes
1	["0|0:48:1:51,51", "1|0:48:8:51,51", "1/1:43:5:.", ...]
2	["0|0:49:3:58,50", "0|1:3:5:65,3", "0/0:41:3", ...]
3	["1|2:21:6:23,27", "2|1:2:0:18,2", "2/2:35:4", ...]
4	["0|0:54:7:56,60", "0|0:48:4:51,51", "0/0:61:2", ...]
5	["0/1:35:4", "0/2:17:2", "1/1:40:3", ...]

Full JSON: store the sample ID and its genotype in a JSON

vcf_json_samples

ln	sample_genotype
1	{{"s_id": "NA00001", "genotype": {"DP": "1", "GT": "0|0", "GQ": "48", "HQ": "51,51"}}, ...}
2	{{"s_id": "NA00001", "genotype": {"DP": "3", "GT": "0|0", "GQ": "49", "HQ": "58,50"}}, ...}
3	{{"s_id": "NA00001", "genotype": {"DP": "6", "GT": "1|2", "GQ": "21", "HQ": "23,27"}}, ...}
4	{{"s_id": "NA00001", "genotype": {"DP": "7", "GT": "0|0", "GQ": "54", "HQ": "56,60"}}, ...}
5	{{"s_id": "NA00001", "genotype": {"DP": "4", "GT": "0/1", "GQ": "35"}}, ...}

Storage Requirements

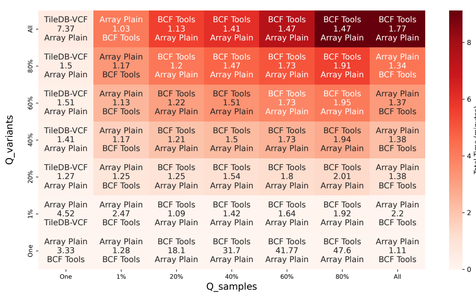


Dataset ∼13k samples and ∼95k variants

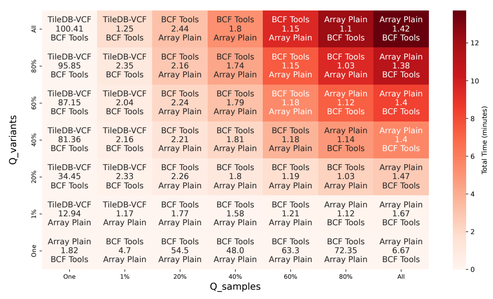
Empirical Experiments

- **Data access**: Retrieves entries for a set of sample IDs and/or variant range. "%" denotes a selected amount, and "All" means no filtering is applied.
- **Variant filtering**: Extends data access pattern and adds a filter on variant's fixed fields.
- **Sample filtering**: Extends data access pattern and adds a filter on genotype values. Returns matching sample IDs.

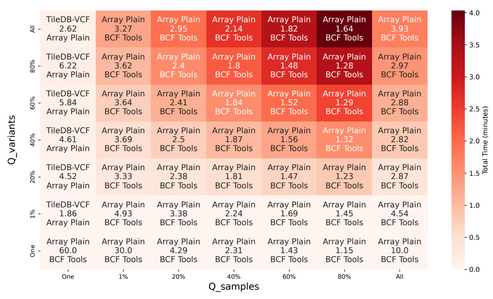
Runtime heatmaps, each cell shows the best- and second-best-performing approaches along with their speed-up factor.



Data access



Sample filtering



Variant filtering

Summary

- RDBMS models rival specialized tools.
- No one-size-fits-all solution: Optimal model depends on query workload.
- Relational models offer superior scalability for large VCF datasets.
- Trade-offs are workload-dependent:
  - Storage needs
  - Query selectivity
  - Dataset size
- Array Plain model – Good all-rounder.

Artifact available:

This work was funded in part by the Autonomous Province of Bozen-Bolzano, call Joint Projects South Tyrol – Germany, project DyHealthNet – CUP: I53C22002980003.

Autonome Provinz Bozen
Provincia autonoma di Bolzano
Provinzia autonoma de Bulsan
SÜDTIROL · ALTO ADIGE

Published in: *Proceedings of the 51st International Conference on Very Large Data Bases*, London, United Kingdom, September 2025.