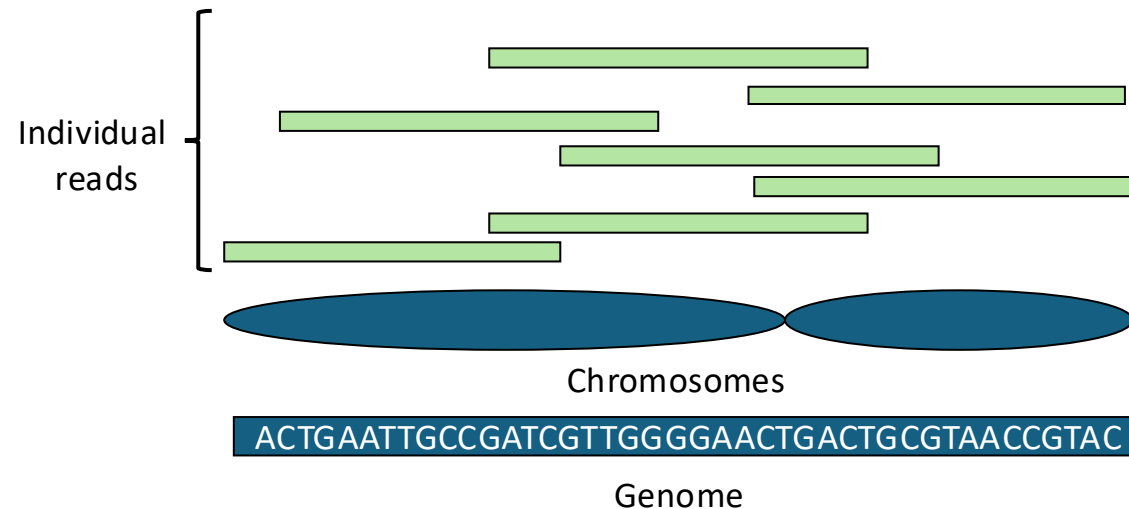


VCFIntegrator: Merging VCF Files with Conflict Resolution

Alina Skrylnik ^{1 2}, Anton Dignös ¹, Christian Fuchsberger ², Giuseppe Tallini ¹, Johann Gamper ¹

Reference genome and genomic variations

- Reference genome: curated representative DNA sequence used as a common coordinate system for the organism
- It defines chromosome positions and reference alleles for comparing individual genomes



Reference genome: A T C G C **A** G
Example sequence: A T C G C **C** T

↑
SNP

- SNP (single nucleotide polymorphism): common variant at a single DNA base position
- SNPs can serve as markers for identifying genomic regions associated with diseases or other traits

Such data are stored in Variant Call Format (VCF) files¹

VCF file structure

The diagram illustrates the VCF file structure. The header section (lines 1-11) contains metadata and format definitions. The table section (lines 12-16) contains the variant data. Brackets indicate that the header is a single block, the table is a single block, and the table is divided into 'Fixed fields' (columns 1-8) and 'Samples' (columns 9-12).

```
##fileformat=VCFv4.4
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,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">
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT NA000001 NA000002 NA000003
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
```

header

table

Fixed fields

Samples

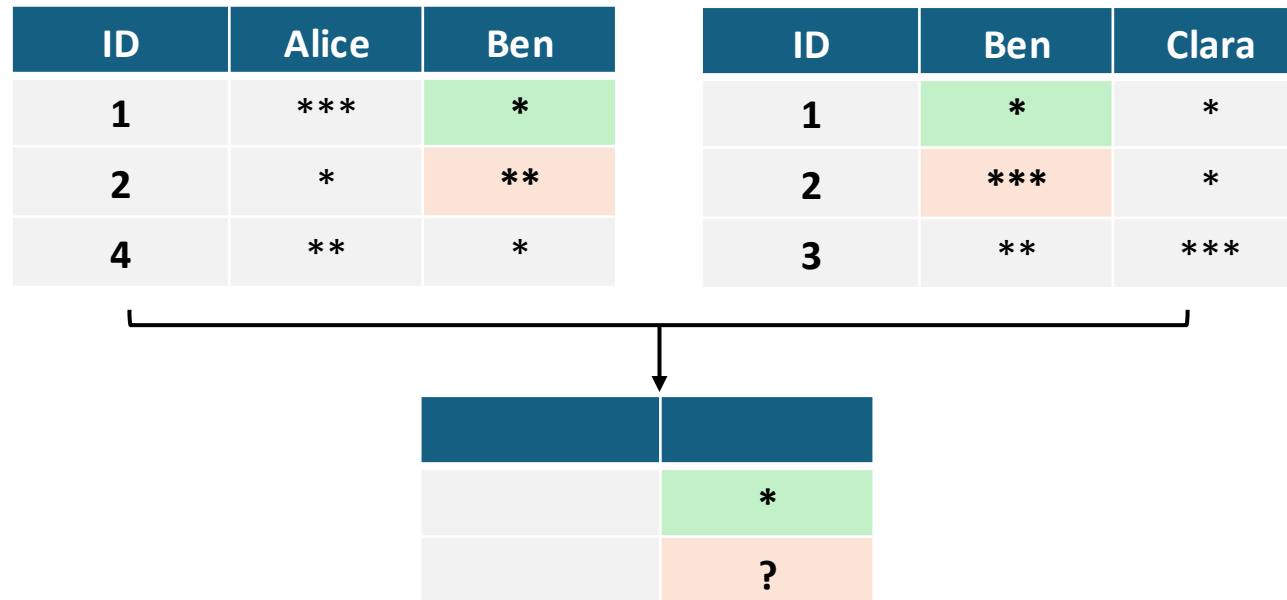
- **#CHROM, POS:** chromosome number and nucleotide position within the chromosome
- **REF, ALT:** reference nucleotide and alternative nucleotide(s) that may occur at that position
- **FORMAT:** two-letter format of the genotype fields reported for each sample
- **GT (genotype):** two allele indices separated by | or /. 0 – REF allele, 1, 2, ... – first, second, ... allele in ALT

Problem

VCF files:

1. Come from multiple technologies: WGS, WES, statistical imputation
2. Large-scale: up to 10^6 rows, 10^5 columns, **GB-scale** files
3. May partially overlap and contain conflicting values

How to integrate files without losing information?



Tools for VCF files integration

Merging (columns)

POS	S1	+	POS	S2	=	POS	S1	S2
100	0/1		100	0/0		100	0/1	0/0
200	1/1		200	0/1		200	1/1	0/1

Concatenation (rows)

POS	S1	+	POS	S1	=	POS	S1
100	0/1		200	1/1		100	0/1
						200	1/1

Retain only shared samples

Picard MergeVcfs²

Non-shared samples are discarded.
Potentially redundant records remain unchanged.

Preserve duplicate entries

VCFTools³ and BCFtools⁴

Duplicate samples are added as separate columns with modified IDs (e.g., 2_file_sampleID in VCFTools and 2:sampleID in BCFtools), rather than being consolidated into a single sample column.

Adjudicate and handle other types of variation

Minos⁵ and DRAGEN⁶

Minos requires an additional reads file and targets one sample. DRAGEN focuses on structural variants (SVs) rather than standard SNP merging.

General VCF integration does not consolidate conflicting calls for the same sample and position

Our proposed solution

VCFIntegrator: conflict-aware merging of VCF files with overlapping records

1 Preserves information

Records with the different REF alleles remain separate; ALT alleles are combined

2 Resolves sample conflicts

Decision order: majority vote → genotype quality → file priority

3 Controls memory without compromising runtime

Sorted files are streamed, processed rows are released

VCFIntegrator

Algorithm 1: VCFINTEGRATOR

Input: Set $\mathcal{F} = \{F_1, \dots, F_n\}$ of VCF files

Output: Merged VCF file F

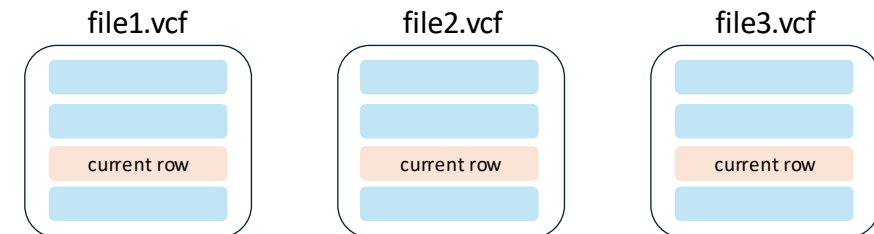
// Initialization

```
1  $p_i \leftarrow$  pointer to first record in  $F_i$  for  $i = 1, \dots, n$ ;  
2  $\mathcal{S} \leftarrow$  set of sample IDs in all files  $F_i$  for  $i = 1, \dots, n$ ;  
3  $F \leftarrow$  empty VCF file with a column for each sample in  $\mathcal{S}$ ;  
  
// Process all records with interleaved file scans  
4 while not all data processed do  
5    $(chr, pos) \leftarrow$  position of least advanced pointer in  $p_1, \dots, p_n$ ;  
6    $G \leftarrow \emptyset$ ;  
7   foreach  $F_i \in \mathcal{F}$  do  
8      $G \leftarrow G \cup \{r \in F_i \mid r.CHR = chr \wedge r.POS = pos\}$ ;  
9     Advance  $p_i$  to the record following  $(chr, pos)$ ;  
10  foreach  $ref \in \{r.REF \mid r \in G\}$  do  
11     $R \leftarrow \{r \in G \mid r.REF = ref\}$ ;  
12     $F \leftarrow F \cup \{\text{MERGE}(R, \mathcal{S})\}$ ;  
13 return  $F$ ;
```

Bitmap indexing of samples

S_1	1	1	1
S_2	1	1	0
S_3	0	0	1

File integration



Read the next shared position

Collect rows sorted by position, group by REF,
then call MERGE algorithm

Merged VCF

VCFIntegrator

Algorithm 2: Algorithm MERGE

Input: Set $R = \{r_1, \dots, r_m\}$ of records, set of samples $\mathcal{S}$

Output: Merged record r

```

1  $r \leftarrow$  empty record;
2  $r.CHR \leftarrow r_1.CHR$ ;
3  $r.POS \leftarrow r_1.POS$ ;
4  $r.REF \leftarrow r_1.REF$ ;
5  $r.ALT \leftarrow \{r'.ALT \mid r' \in R\}$ ;
6  $r.FORMAT \leftarrow r_1.FORMAT \circ \dots \circ r_m.FORMAT$ ;
7 foreach sample  $S \in \mathcal{S}$  do
8   if  $r_1.S = \dots = r_m.S$  then
9      $r.S \leftarrow r_1.S$ ;
10  else if  $\nexists r' \in R$  with a value  $r'.S$  then
11     $r.S \leftarrow ./.$ ; // empty value
12  else if majority of  $r_i \in R$  has genotype  $r_i.S = v$  then
13     $r.S \leftarrow v$ ;
14  else if  $\exists R' \subseteq R$  with PL or GQ values for  $r \in R'$  then
15     $r.S \leftarrow$  genotype with highest PL or GQ value in  $R'$ ;
16  else
17     $r.S \leftarrow$  genotype of the first record in the user-specified preference list;
18  Remap  $r.S$  to  $r.ALT$  if ( $r.ALT$  changed wrt to the input records);
19 return  $r$ ;

```

CHR	POS	REF	ALT	S1
22	478	G	A	0/0

Same sample and position
Different genotype calls

CHR	POS	REF	ALT	S1
22	478	G	A	0/1

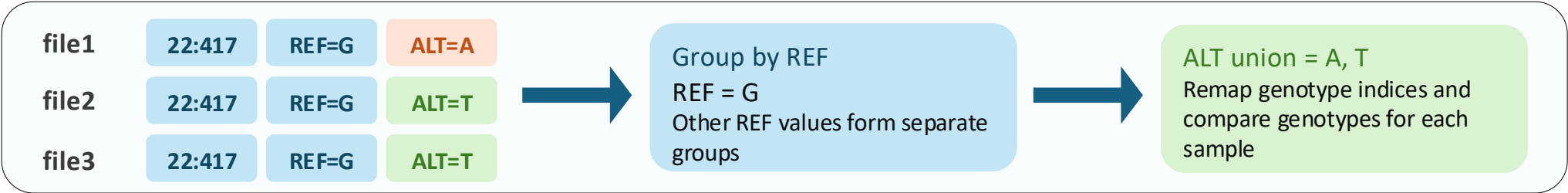
0/0 occurs twice
Output GT = 0/0

CHR	POS	REF	ALT	S1
22	478	G	A	0/0

Write merged row,
release processed rows
from memory

Conflicts resolution

RECORD PREPARATION BEFORE GENOTYPE COMPARISON



DECISION ORDER

- 1 Same genotype
Keep the shared GT
- 2 Majority voting
Use the most **frequent** GT
- 3 Genotype quality
Select GT with the **highest GQ**
- 4 File priority
Use the GT from the file with the highest priority

RUNNING EXAMPLES FOR EACH STRATEGY

22:480, S1	0/1	0/1	0/1	All calls agree	0/1	preserved
22:478, S1	0/0	0/1	0/0	2 votes for 0/0	0/0	majority
22:417, S2	1/0	0/1	./.	PL gives GQ 30	1/0:88	88 > 30

GQ: 88 PL: 0, 30, 85

Genotype quality (GQ) calculation

PL: 0, 30, 85

GQ = second-lowest PL – lowest PL = 30 – 0 = 30

File 1 has higher confidence in the genotype value with GQ: 88

Fallback rule: when majority vote and quality values are not available, *use the user-specified priority*

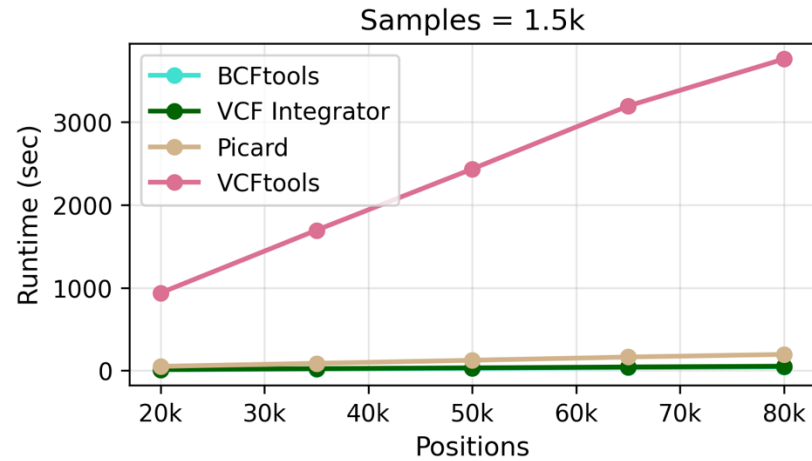
Experimental setup

- Input files – **.vcf.gz** format (compressed VCF files)
- **3** files (minimum for a majority vote)
- Synthetically generated from HAPNEST dataset⁷
- 3 scenarios: full overlap, partial overlap (90% common), 20% GT conflicts
- For VCFIntegrator only: varying sizes of GT and REF-ALT conflicts

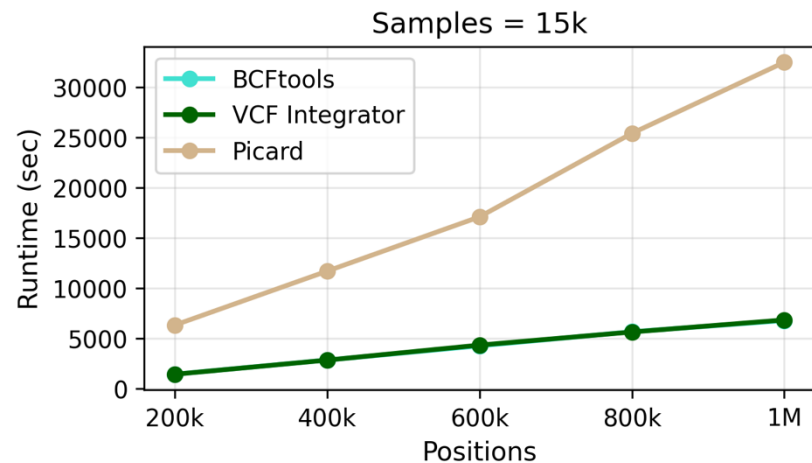
Datasets	Size		Conflicts ratio		
	Samples [k]	Positions [k]	Overlap [%]	GT [%]	REF-ALT [%]
Small	0.5, 1, 1.5 , 2, 2.5	20, 35, 50 , 65, 80	90 – 100	0 – 50	0 – 50
Large	5, 10, 15 , 20, 25	200, 400, 600 , 800, 1000	90 – 100	0 – 50	0 – 50

VCFIntegrator is comparable for a runtime with the second fastest competitor: BCFtools

Small datasets

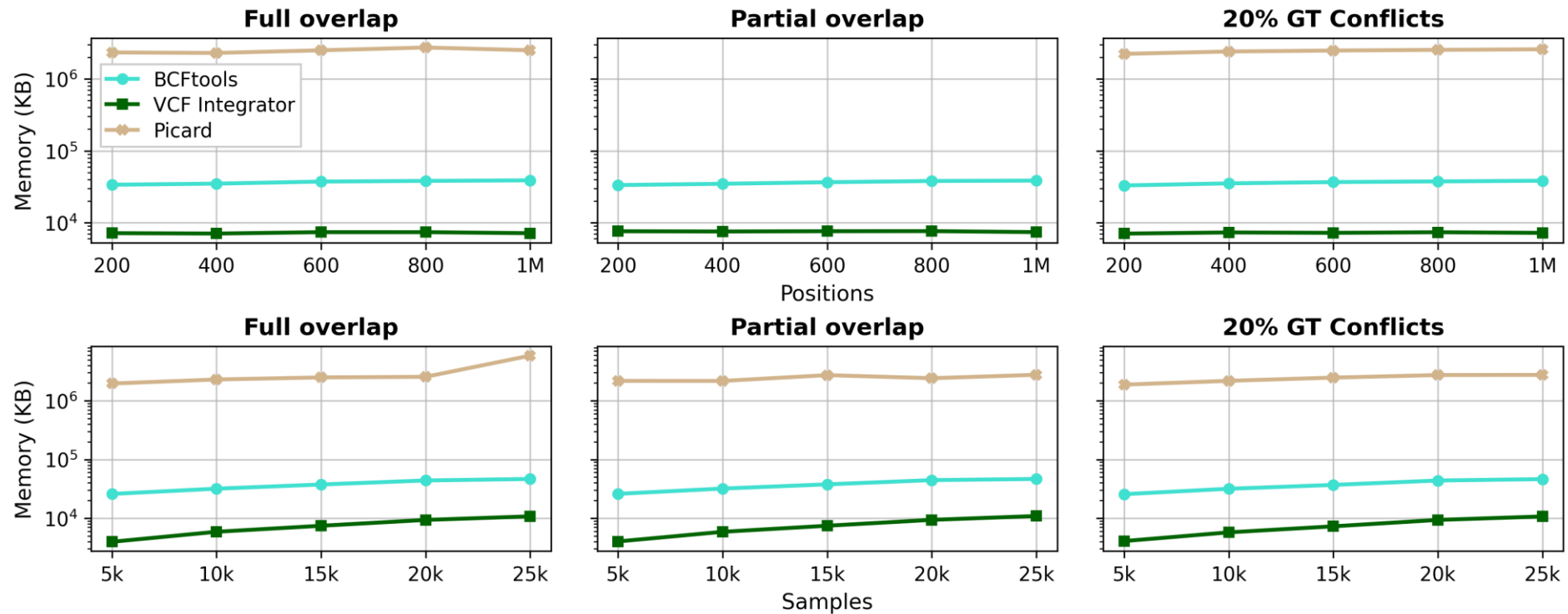


Large datasets

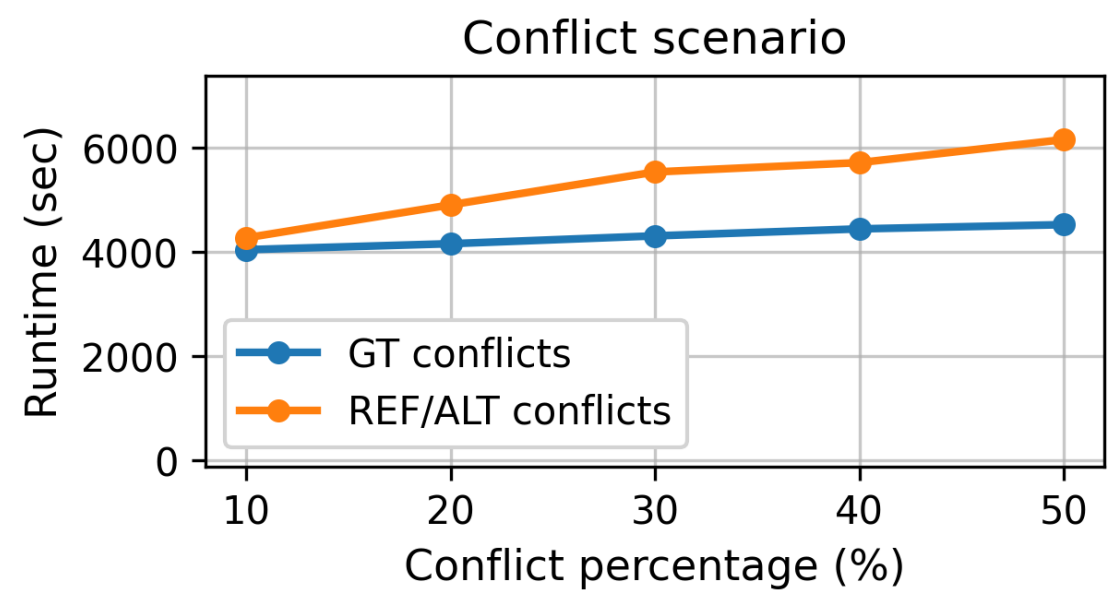


Average runtime difference (sec)			
Scenario	BCFtools	VCFIntegrator	Diff (%)
Full overlap (S)	4177.12	4197.57	+0.5
Full overlap (P)	4349.20	4238.31	-2.5
Partial overlap (S)	4063.74	4493.51	+10.6
Partial overlap (P)	4469.65	4556.65	+1.9
GT conflicts (S)	4541.09	4372.85	-3.7
GT conflicts (P)	4763.05	4459.68	-6.4

VCFIntegrator is the most memory-efficient in all scenarios across the algorithms



Increasing the conflict percentage only slightly affects runtime in the REF/ALT conflict scenario



Dataset dimensions: 15k columns × 600k rows

Conclusion & Future Work

- Efficiently handled conflicts in large-dimensional VCF files and under high conflict rates
 - Outperformed BCFtools by up to **79%** in memory usage
 - Maintained comparable runtime performance within a range of approximately **±6-10%**
-
- Integrate .bcf files merging (binary counterpart of VCF)
 - Extend genotype confidence metrics (e.g., genotype-specific PL values)
 - Account for insertions and deletions

References

1. SAMtools: The Variant Call Format Specification (2024), <https://samtools.github.io/hts-specs/VCFv4.4.pdf>, version 4.4, October 9, 2024
2. Broad Institute: Picard toolkit. <https://broadinstitute.github.io/picard/> (2019)
3. Danecek, P., et al.: The Variant Call Format and VCFtools. *Bioinformatics* 27(15), 2156–2158 (06 2011).
<https://doi.org/10.1093/bioinformatics/btr330>
4. Danecek, P., et al.: Twelve years of SAMtools and BCFtools. *GigaScience* 10(2) (02 2021). <https://doi.org/10.1093/gigascience/giab008>
5. Hunt, M., et al.: Minos: variant adjudication and joint genotyping of cohorts of bacterial genomes. *Genome Biology* 23(1), 147 (2022).
<https://doi.org/10.1186/s13059-022-02714-x>
6. Behera, S., et al.: Comprehensive genome analysis and variant detection at scale using DRAGEN. *Nature Biotechnology* 43(7), 1177–1191 (2025). <https://doi.org/10.1038/s41587-024-02382-1>
7. Wharrie, S., et al.: HAPNEST: efficient, large-scale generation and evaluation of synthetic datasets for genotypes and phenotypes. *Bioinformatics* 39(9), btad535 (08 2023). <https://doi.org/10.1093/bioinformatics/btad535>