SPRITE: A Fast Parallel SNP Detection Pipeline

Vasudevan Rengasamy and Kamesh Madduri
The Pennsylvania State University

Outline

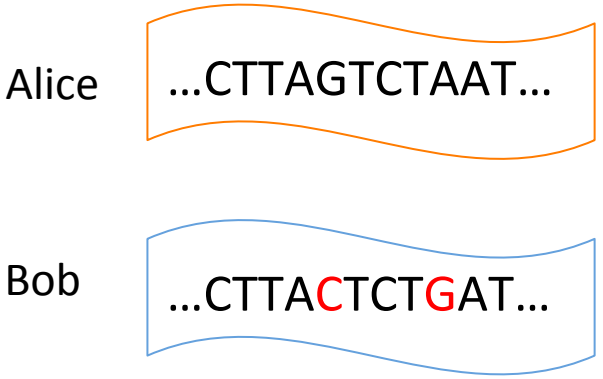
- Introduction
- SPRITE Pipeline
  - PRUNE : Parallel Short-Read Alignment
  - SAMPA : Parallel In-Memory sort
  - PARSNIP : Parallel Counting-based SNP caller
- SPRITE^{+} : In Memory SPRITE
- Performance and Quality Comparisons
- Conclusions

What are SNPs?

Genetic variation resulting from single base flips

Most common type of mutation (~90 %)

Accurate detection plays vital role in identifying disease risk, studying drug efficacy, etc



Goal

Our Goal: Creating a Fast SNP detection pipeline with good accuracy on high coverage sequence data

Genome Data Analysis

~ 250 GB

Short Reads

Quality Assessment

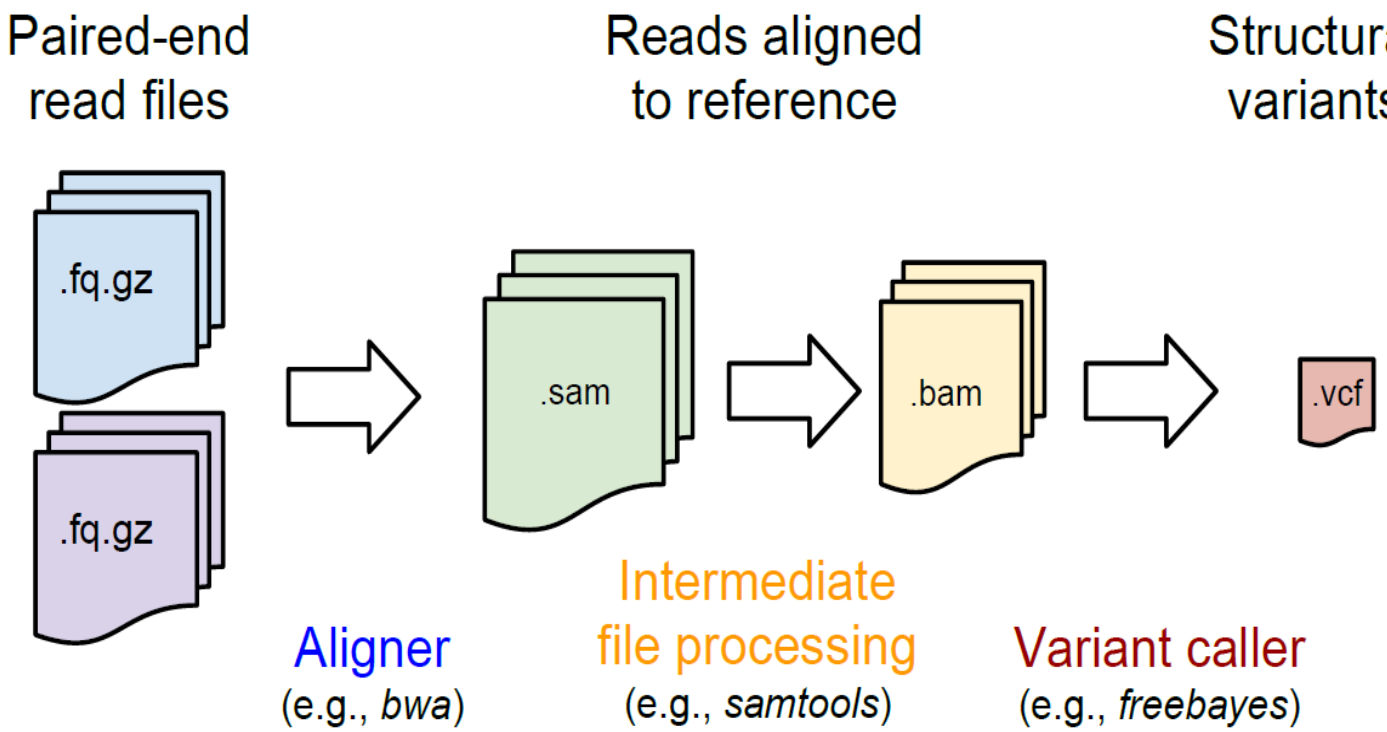
Read Alignment

Variant Identification

Annotation

Prioritization, Filtering
Downstream analysis

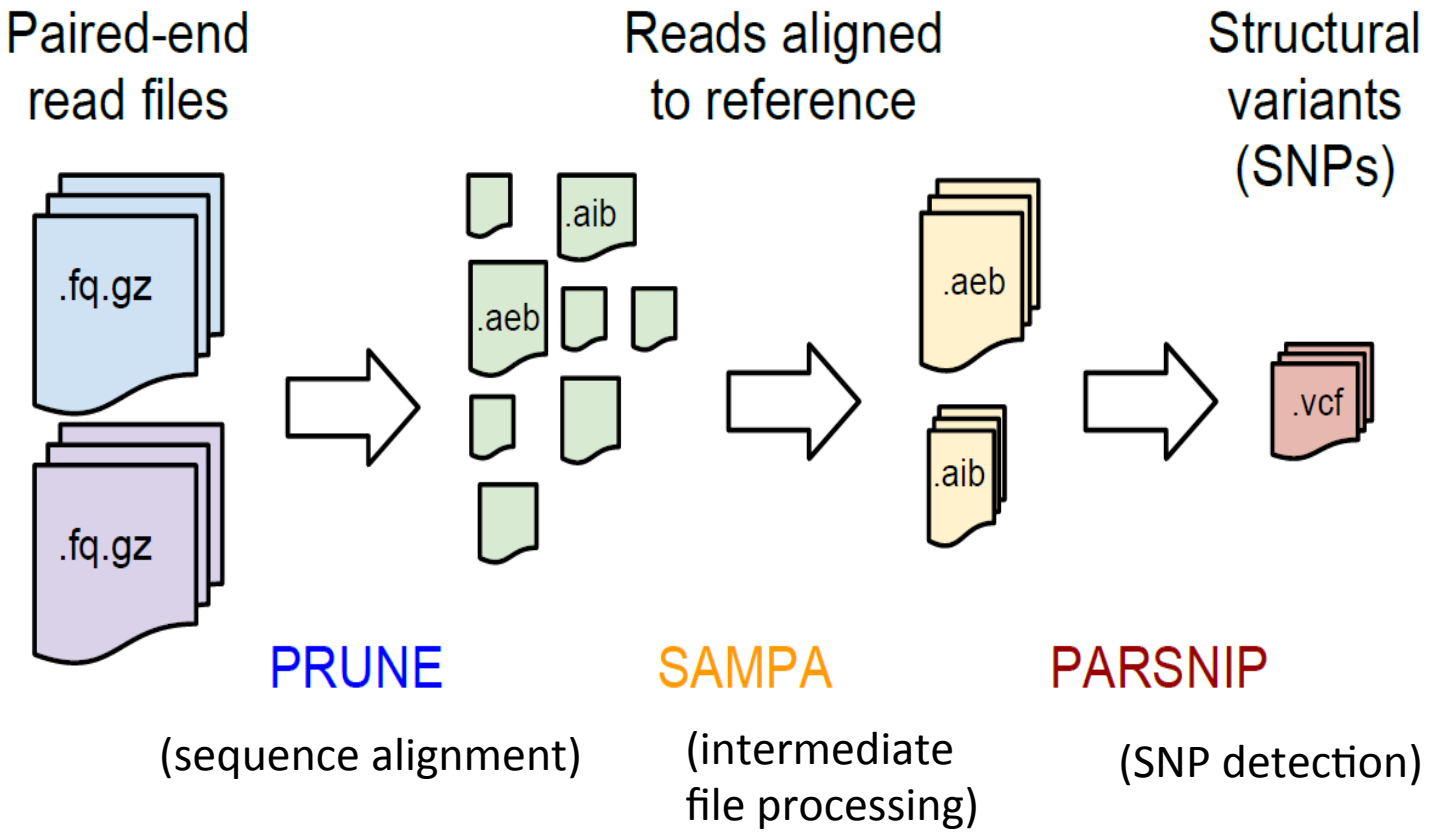
Variant Detection Pipelines



Outline

- Introduction
- **SPRITE Pipeline**
  - PRUNE : Parallel Short-Read Alignment
  - SAMPA : Parallel In-Memory sort
  - PARSNIP : Parallel Counting-based SNP caller
- SPRITE^{+} : In Memory SPRITE
- Performance and Quality Comparisons
- Conclusions

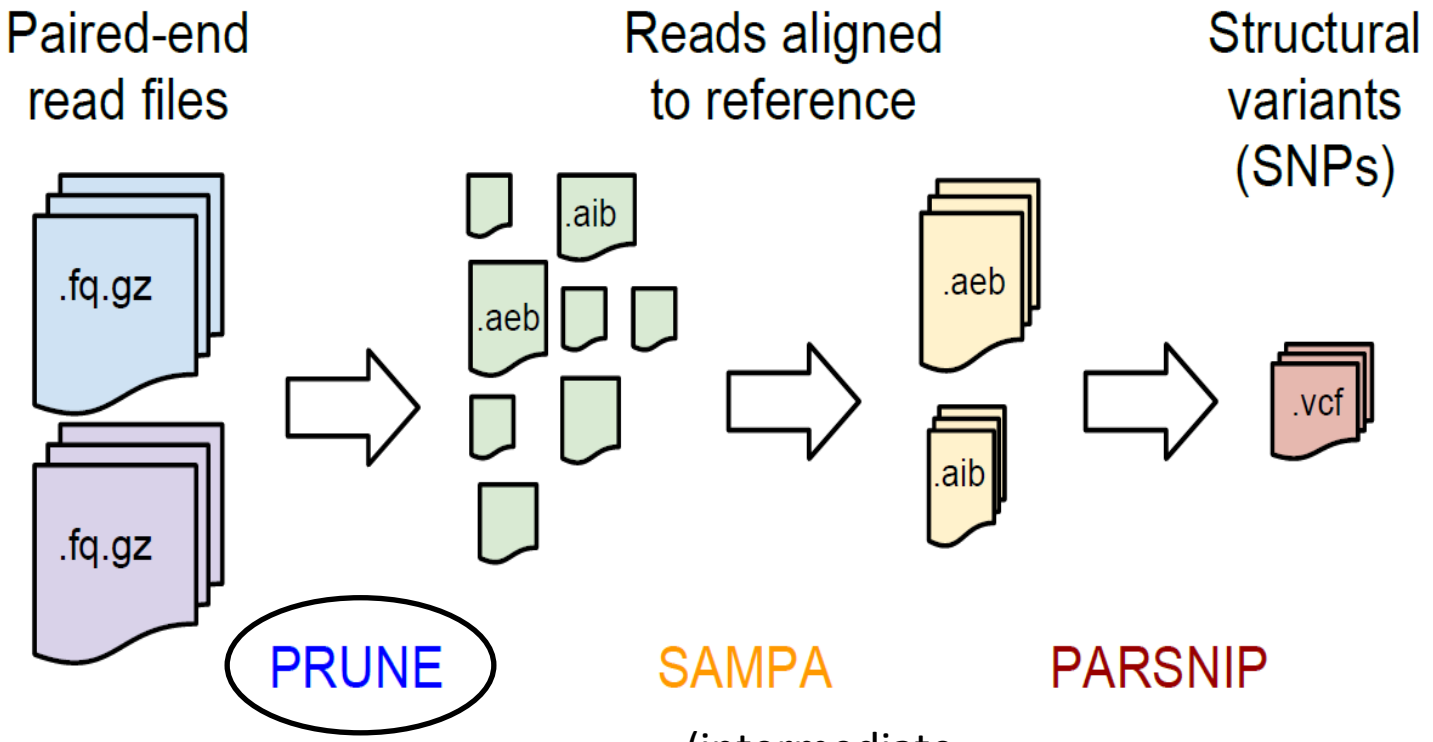
SPRITE: HPC pipeline for SNP detection



Outline

- Introduction
- SPRITE Pipeline
  - **PRUNE : Parallel Short-Read Alignment**
  - SAMPA : Parallel In-Memory sort
  - PARSNIP : Parallel Counting-based SNP caller
- SPRITE^{+} : In Memory SPRITE
- Performance and Quality Comparisons
- Conclusions

SPRITE: HPC pipeline for SNP detection



Based on BWA-MEM
Changes to support multi-node parallelism

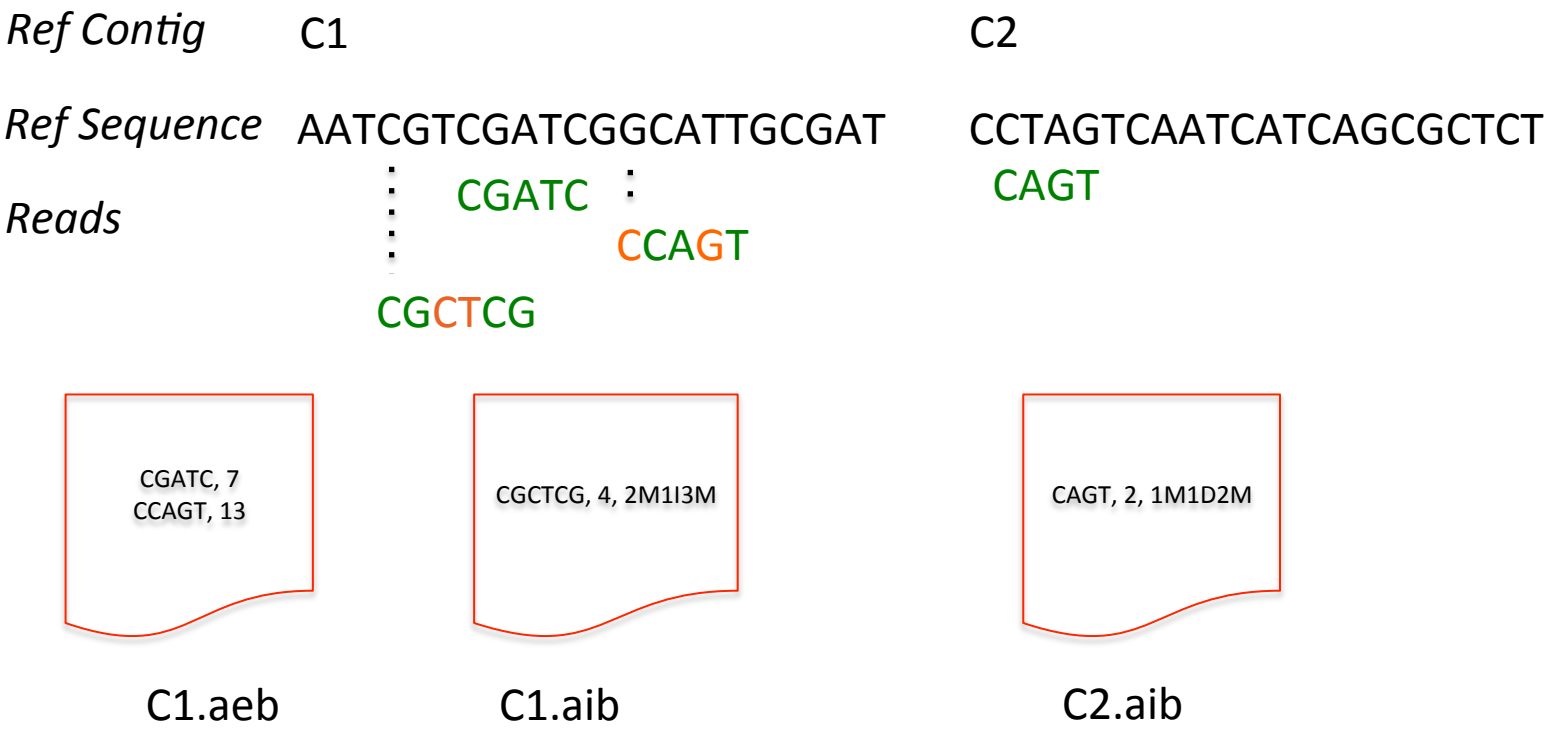
(intermediate file processing)

(SNP detection)

PRUNE: Overview

- Current highlights
  1. Fast read partitioning
  2. I/O reduction due to separating Exact and Inexact alignment records
  3. Fixed length output records
  4. Opportunity for coarse-grained parallelism by separating contigs in output
- Future work opportunities
  1. Currently assume that paired-end FASTQ files have same read length
  2. Overhead due to SAM record parsing could be further reduced

PRUNE: Alignment Output



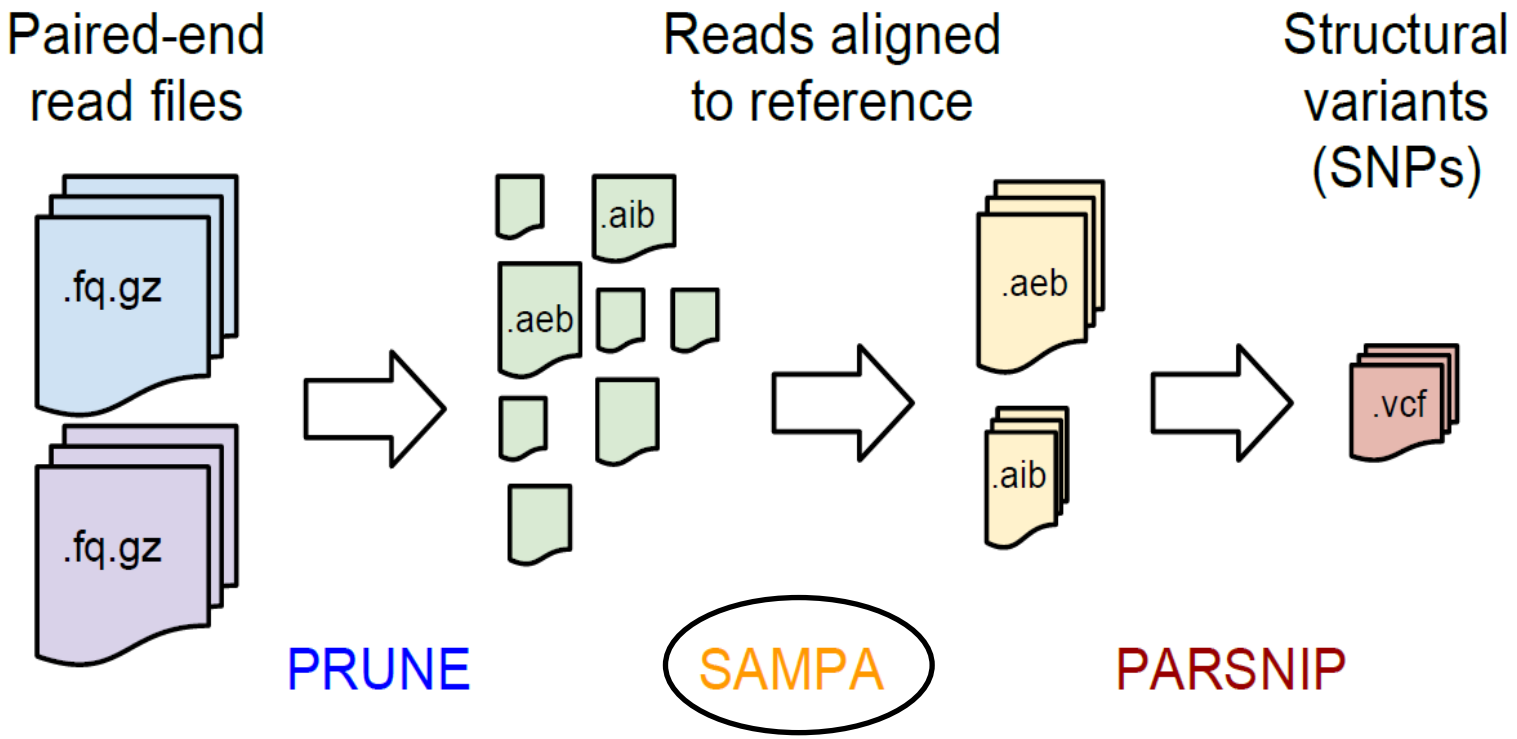
Separate Outputs for Exact and Inexact Alignments

Separate Output Files for Contigs

Outline

- Introduction
- SPRITE Pipeline
  - PRUNE : Parallel Short-Read Alignment
  - **SAMPA : Parallel In-Memory sort**
  - PARSNIP : Parallel Counting-based SNP caller
- SPRITE^{+} : In Memory SPRITE
- Performance and Quality Comparisons
- Conclusions

SPRITE: HPC pipeline for SNP detection



New tool
Parallel in-memory sort
Operates on binary intermediate files

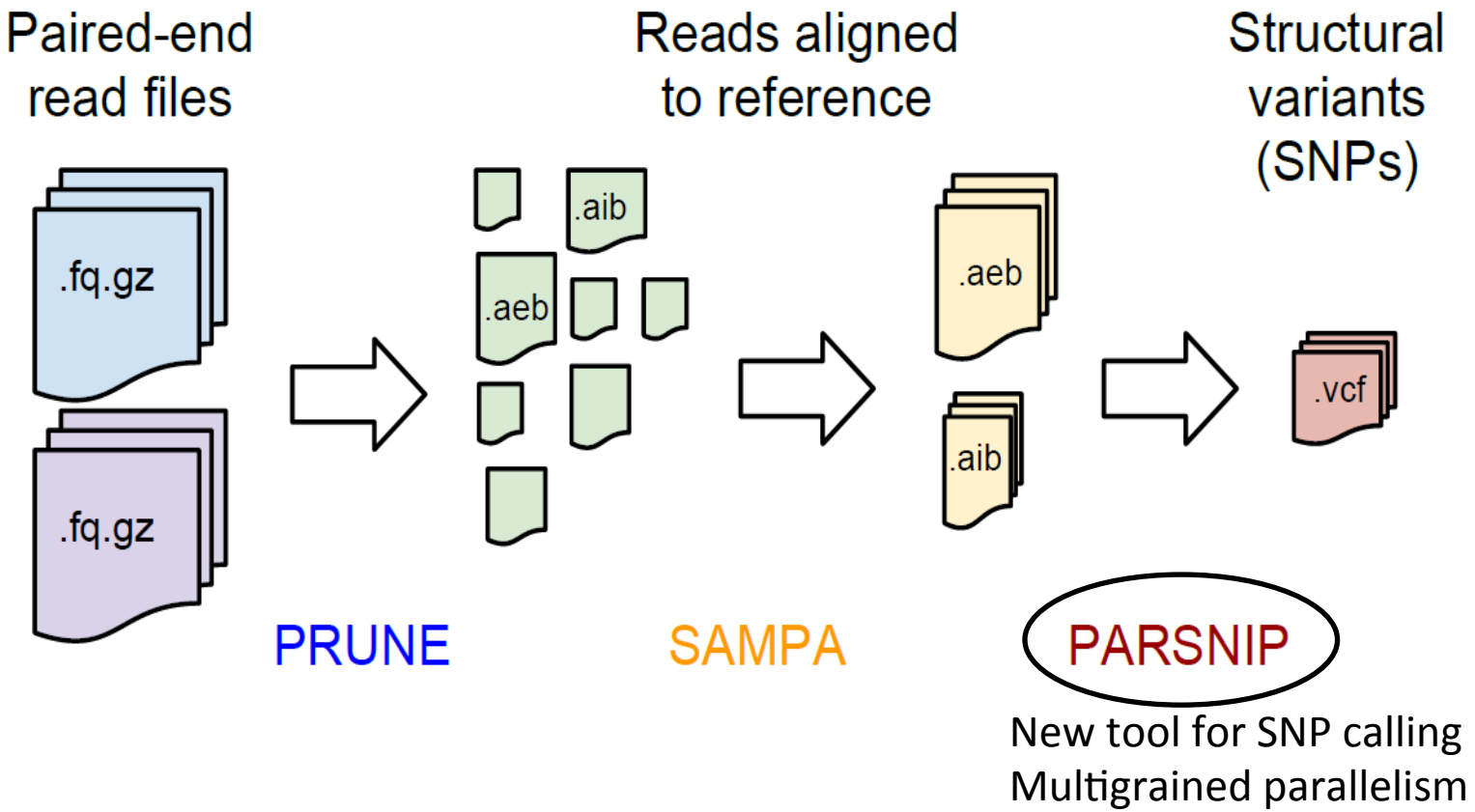
SAMPA: Advantages/Drawbacks

- Current Highlights
  - Coarse-grained parallelism due to separate files for contigs
  - Only one contig needs to be memory resident
  - In-memory, Linear time sort
- Future work opportunities
  - Hybrid parallelism

Outline

- Introduction
- SPRITE Pipeline
  - PRUNE : Parallel Short-Read Alignment
  - SAMPA : Parallel In-Memory sort
  - **PARSNIP : Parallel Counting-based SNP caller**
- SPRITE^{+} : In Memory SPRITE
- Performance and Quality Comparisons
- Conclusions

SPRITE: HPC pipeline for SNP detection



New tool for SNP calling
Multigrained parallelism

PARSNIP: Advantages and Drawbacks

- Current highlights
  - Simple and very fast
  - Accuracy comparable to state-of-the-art SNP callers for high coverage data
  - Easily tunable filters
- Future work opportunities
  - Improve accuracy on low coverage sequence data

PARSNIP Overview

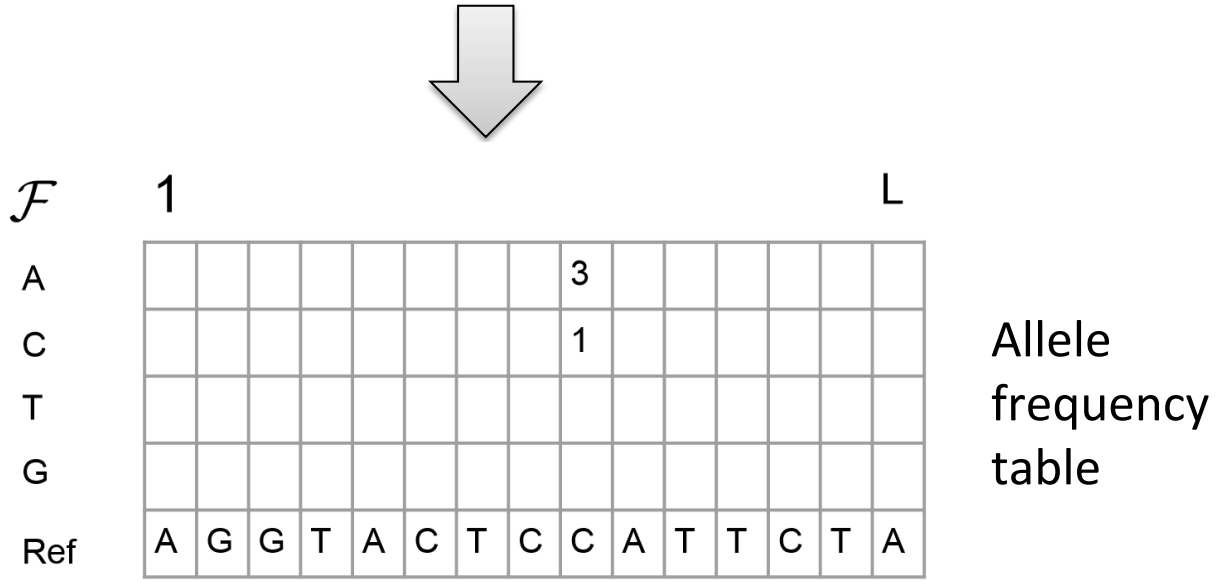
GTACTCGTCGCTTGCGT**A**TTTTGGT→

ATTTTGGTCGCTGGACTTGTCGTCGCTTTA→

GTACTCGTCGCTTGCGT**A**TTTTGGTCGCTGGACTTGTC→

TTGCGT**C**TTTTGGTCGCTGG→

….



Allele frequency table

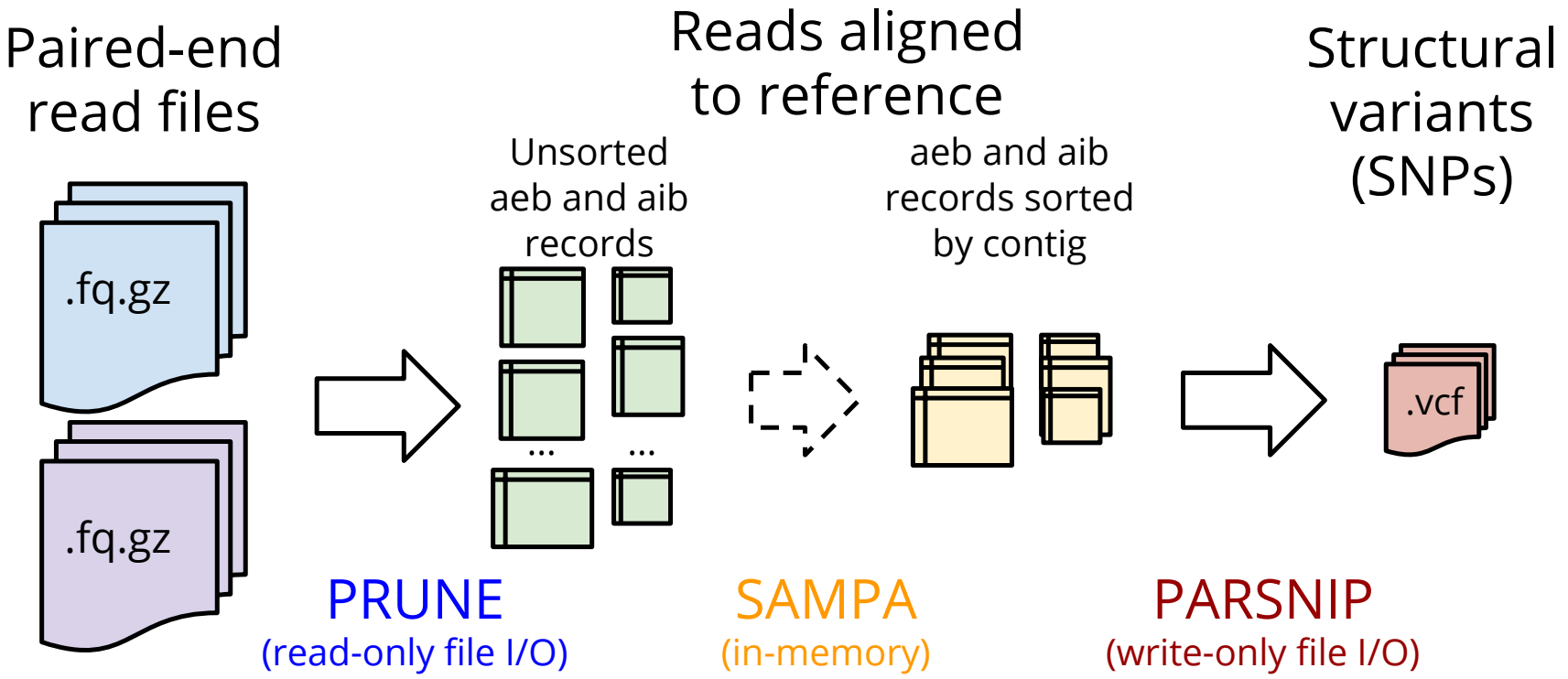
PARSNIP: Filters

Filter	Description	Default setting
DP	Read Depth	>1
AAC	Alternate Allele Count	>1
MQ	Average mapping quality of alternate allele	>20
AAF	Fraction of alternate allele count	>20%
SB	Strand bias	>0

Outline

- Introduction
- SPRITE Pipeline
  - PRUNE : Parallel Short-Read Alignment
  - SAMPA : Parallel In-Memory sort
  - PARSNIP : Parallel Counting-based SNP caller
- **SPRITE+ : In Memory SPRITE**
- Performance and Quality Comparisons
- Conclusions

SPRITE+: In-memory Implementation



Current highlights: No overhead due to disk I/O, better speedup

Future work opportunities: Reduce memory requirement per node

Outline

- Introduction
- SPRITE Pipeline
  - PRUNE : Parallel Short-Read Alignment
  - SAMPA : Parallel In-Memory sort
  - PARSNIP : Parallel Counting-based SNP caller
- SPRITE+ : In Memory SPRITE
- **Performance and Quality Comparisons**
- Conclusions

Experimental Setup

System Description

- Stampede supercomputer at TACC
  - Each node has
    - Two 8-core Intel Xeon E5 (Sandy Bridge) processors
    - 32GB DDR3 memory
  - Lustre-based Scratch file system

Experimental Setup

- Illumina platinum genome sequence data set NA12878
  - 50X sequencing depth
  - Sequencer: Illumina HiSeq 2000
- Reference human genome
  - Human genome version 19 (hg19)
  - 93 contigs

End-to-End Pipeline Execution

Tools used in pipeline stages

Stage	RefPipeline	SpeedSeq	SPRITE
Alignment	BWA-MEM 0.7.12	BWA-MEM 0.7.12	PRUNE
Alignment Output Processing	SAMtools-1.1	SAMBLASTER v0.1.21, Sambamba v0.4.7	SAMPA
SNP Calling	GATK v3.2.2	FreeBayes v0.9.16-1	PARSNIP

End-to-End Pipeline Execution

Single-node parallel execution time

Pipeline Stage	RefPipeline	SpeedSeq		SPRITE	
	Time	Time	Speedup	Time (Minutes)	Speedup
Overall	1378	836	1.65x	699.5	1.97x
Alignment	580	670	1.65x	692	.84x
Alignment Output Processing	526			4	131.5x
SNP Calling	270	166	1.63x	3.5	77x

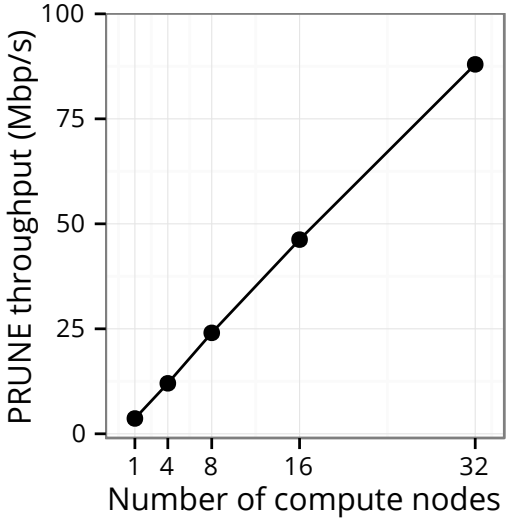
The presentation and code are available at sprite-psu.sourgeforge.net

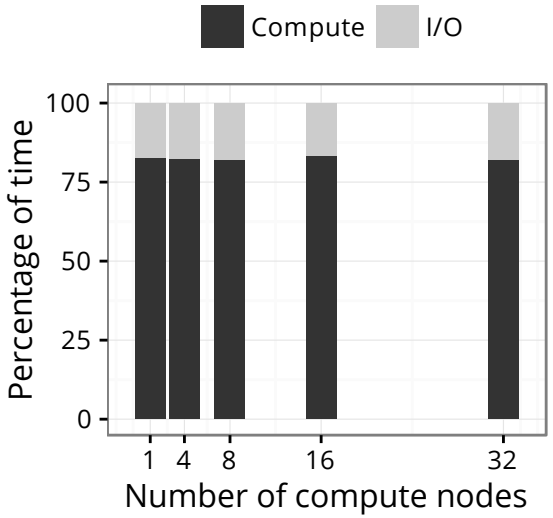
End-to-End Pipeline Execution

Multi-node execution time

Pipeline Stage	SPRITE, 16 nodes		SPRITE^{+},16 nodes	
	Time	Speedup	Time	Speedup
Overall	56.40	12.4X	48.0	14.6X
PRUNE	54.60	12.67X	46.80	14.78X
SAMPA	1.13	3.54X	0.80	5X
PARSNIP	0.68	5.14X	0.37	9.46X

PRUNE Parallel Scaling

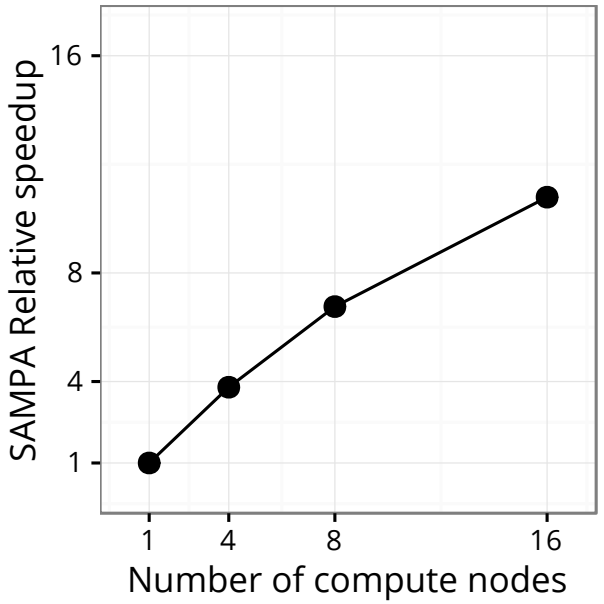




1 MPI process per node, 16 threads per process

24.6 X speedup on 32 nodes

SAMPA Parallel Scaling

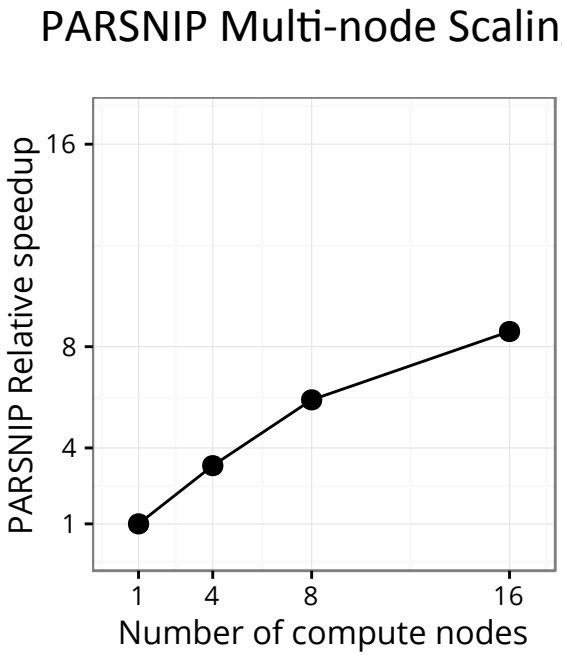


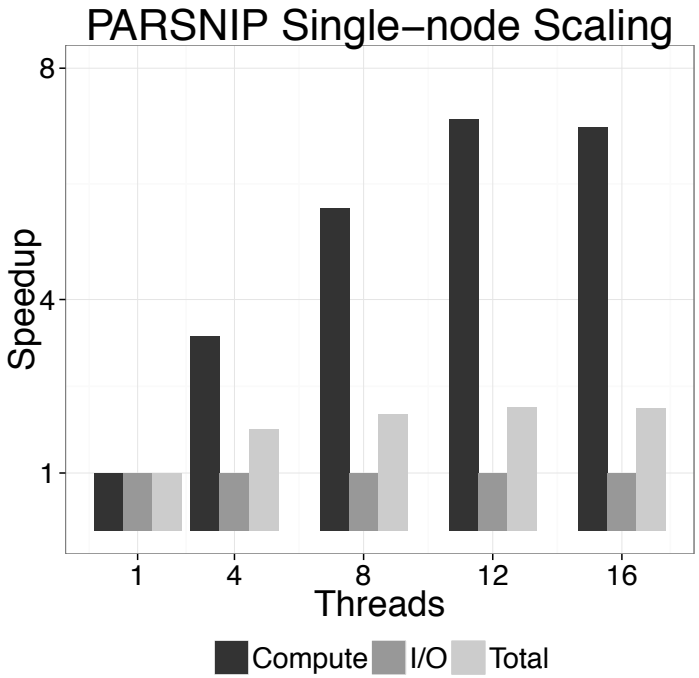
11X speedup for 16 MPI processes

SAMPA takes less than a minute using 16 processes

Coarse-grained parallelism limits scalability beyond 8 MPI processes

PARSNIP Parallel Scaling





Compute part scales well up to 12 cores

PARSNIP completes in 24.6 seconds on 16 cores

Coarse-grained work partitioning limits scalability

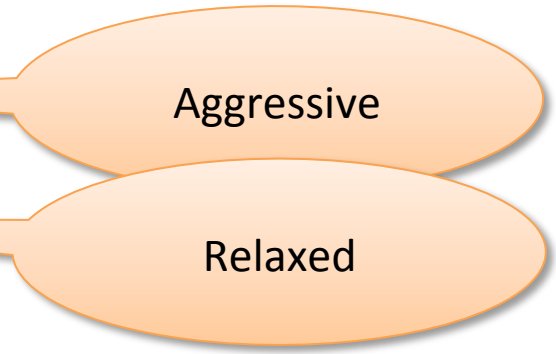
Evaluating Accuracy

- Compare PARSNIP's accuracy with GATK HaplotypeCaller and FreeBayes
- Ground truth SNP calls
  - NIST GIAB v2.19
    - Contains high confidence SNP calls for NA12878 data sample.
  - Illumina high quality calls v7.0
    - Derived from variants called by multiple pipelines on CEPH pedigree trio 1463

Evaluating Accuracy: SPRITE Configurations

- Evaluate 3 SPRITE Configurations using different filter settings

Configuration	MQ	SB	AAF
SPRITE-1	>20	>0.1	>20%
SPRITE-2	>30	>0.2	>25%
SPRITE-3	>0	>=0	>20%



Accuracy Metrics

Sensitivity = True SNPs detected/True SNPs in Ground Truth

Precision = True SNPs detected/Total SNPs detected

Ideal scenario: 100% Sensitivity and Precision

Evaluating Accuracy

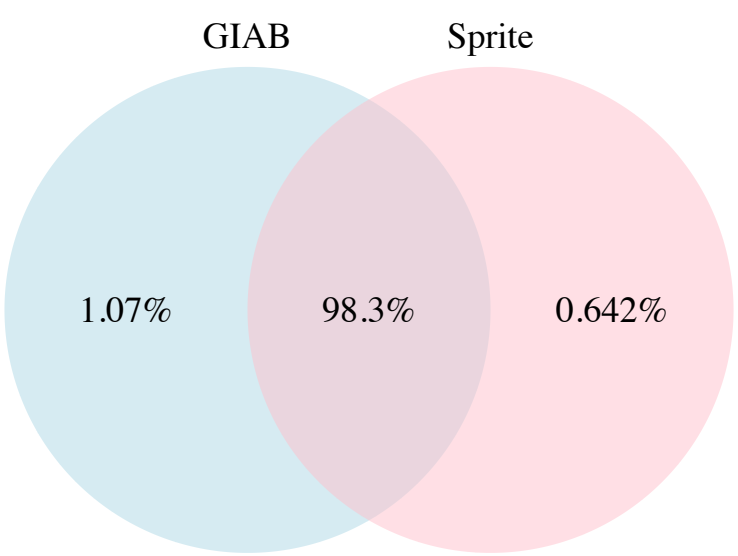
Pipeline	NIST GIAB v2.19		Illumina High Confidence calls v7.0	
	Sensitivity	Precision	Sensitivity	Precision
RefPipeline	99.55	**99.48**	97.5	99.7
SpeedSeq	99.51	99.32	97.5	99.41
SPRITE-1	99.46	98.71	97.3	98.88
SPRITE-2	98.93	99.35	95.9	99.18
SPRITE-3	**99.63**	92.12	98.3	97.53

High sensitivity due to l
filter setting

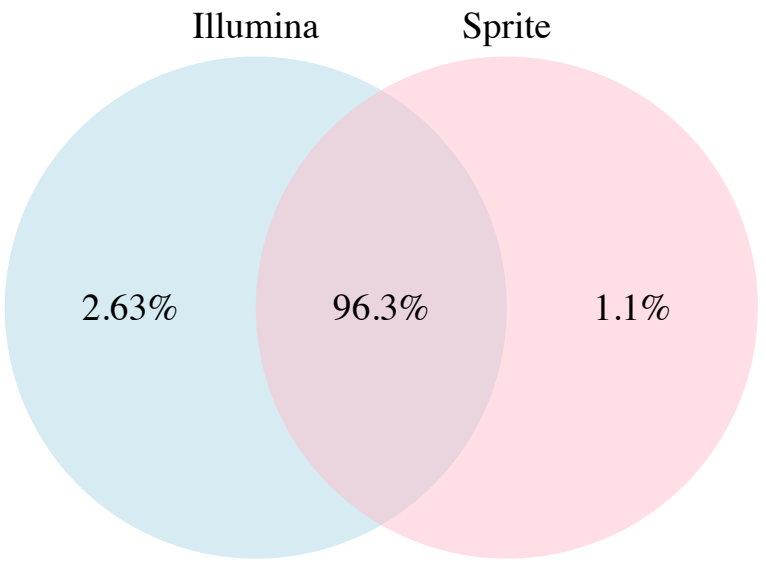
High precision due to
aggressive filter setting

Evaluating Accuracy: Overlap with ground truth

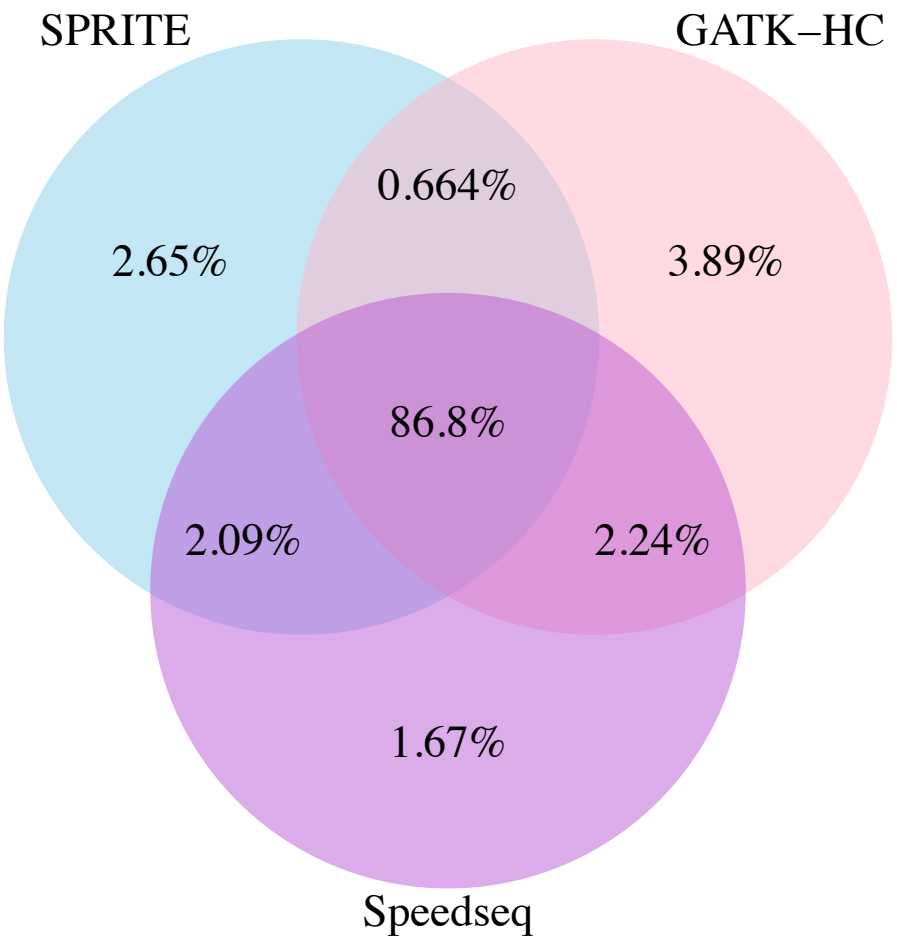
Overlap with NIST GIAB v2.19



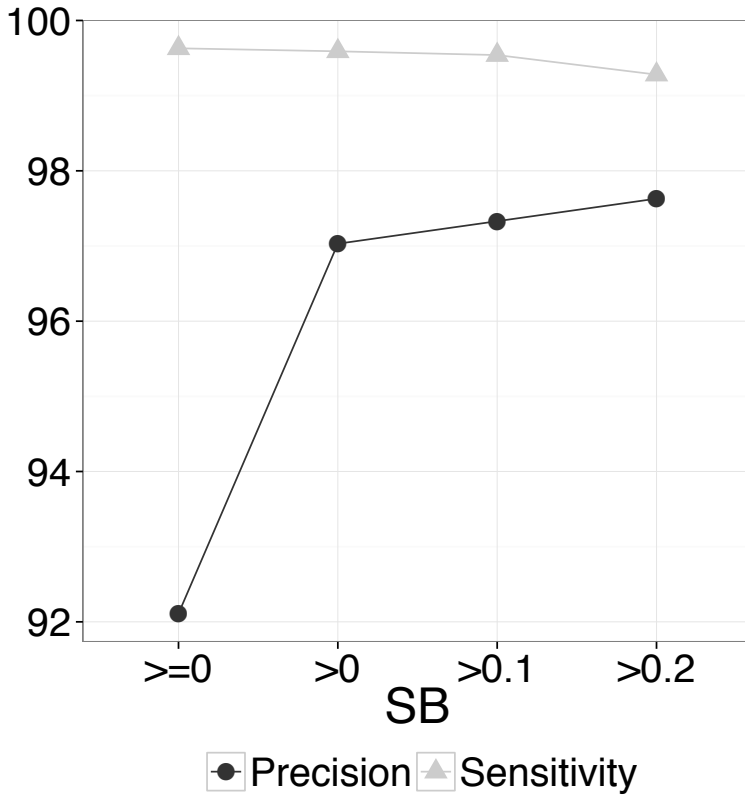
Overlap with Illumina v7.0



Evaluating Accuracy: Overlap with Other Variant Callers



Evaluating Accuracy: Filter Settings



Conclusions

- Developed SPRITE pipeline to detect SNP in High-Depth donor genome
- Optimized alignment, intermediate file processing and SNP calling stages
- SPRITE^{+} reduces overhead due to intermediate file I/O
- 1.97X Single node speedup, 28.7X speedup using 16 nodes over reference pipeline
- Accuracy comparable to state-of-the-art tools for a High-Depth Sequenced genome

Questions?

Project web site: sites.psu.edu/XPSGenomics

THANK YOU

The presentation and code are available at
sprite-psu.sourgeforge.net