Detecting and Analyzing Genomic Structural Variation using Distributed Computing

Chris Whelan

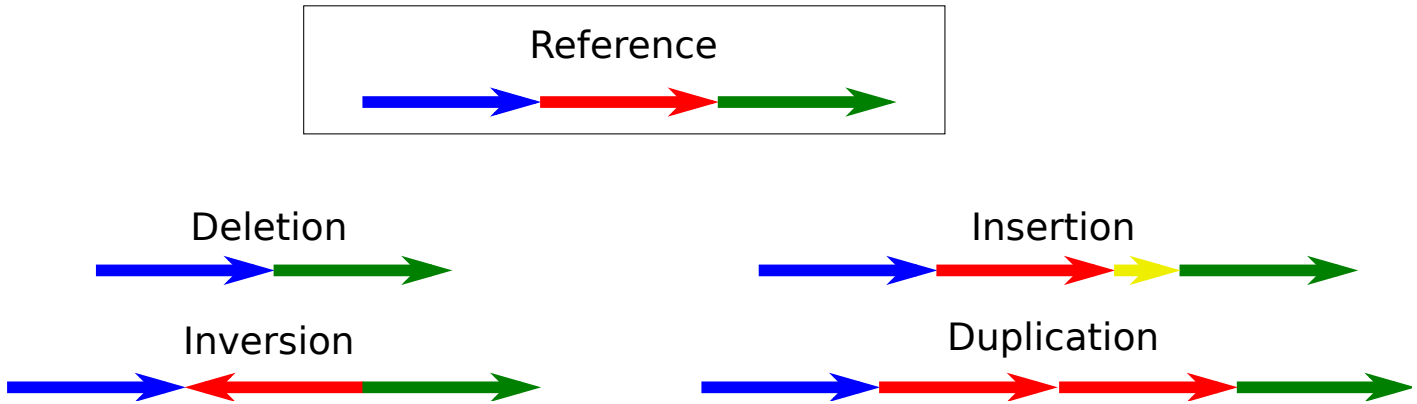
Oregon Health & Science University

February 6, 2014

Outline

Outline

Genomic Structural Variations



- Structural Variations (SVs) are differences between the genomes of individuals and the reference genome sequence that affect at least 40-50 bp of DNA
- SV's explain most variant bases between individuals in the human population (Levy et al. 2007)
- Have effects on phenotype including schizophrenia/autism, cancer-driving mutations, cardiovascular disease, likely many more

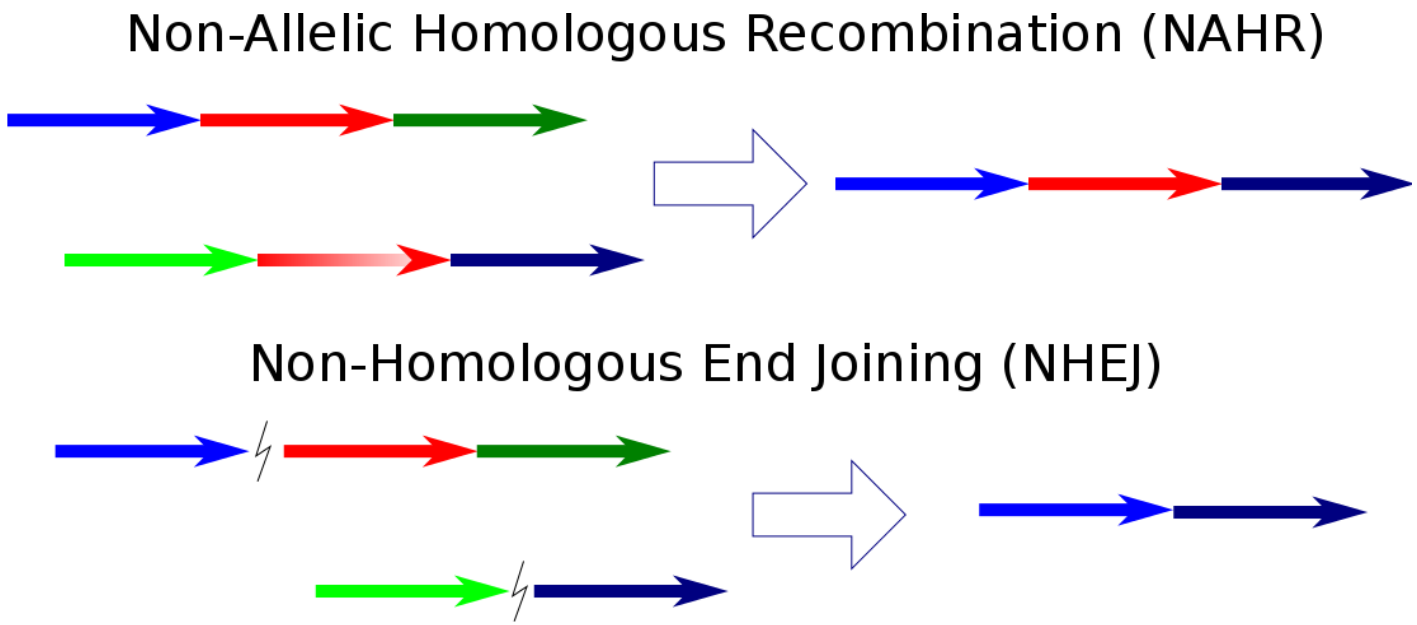
Contributions of this Thesis

- Data analysis and software for determining features enriched or depleted at gibbon genome breakpoints. (Chap. 8, Carbone et al, submitted; Cappozzi et al., *Genome Res.*)
- An algorithmic framework for solving SV detection problems in Hadoop and MapReduce based on the computation of local features along the genome. (Chap. 4, RECOMB-seq 2013)
- Cloudbreak, an open-source Hadoop-based tool for discovering genomic deletions and insertions, which improves accuracy and achieves dramatically faster runtimes over existing approaches. (Chap. 5 & 6, RECOMB-seq 2013)
- A demonstration of the combination of Cloudbreak's local feature model with a statistical sequence labelling technique (conditional random fields) to integrate multiple data sources and improve breakpoint resolution. (Chap. 7)

Outline

Mechanisms and Scales of SV Formation

How do SVs happen? Many mechanisms, including:



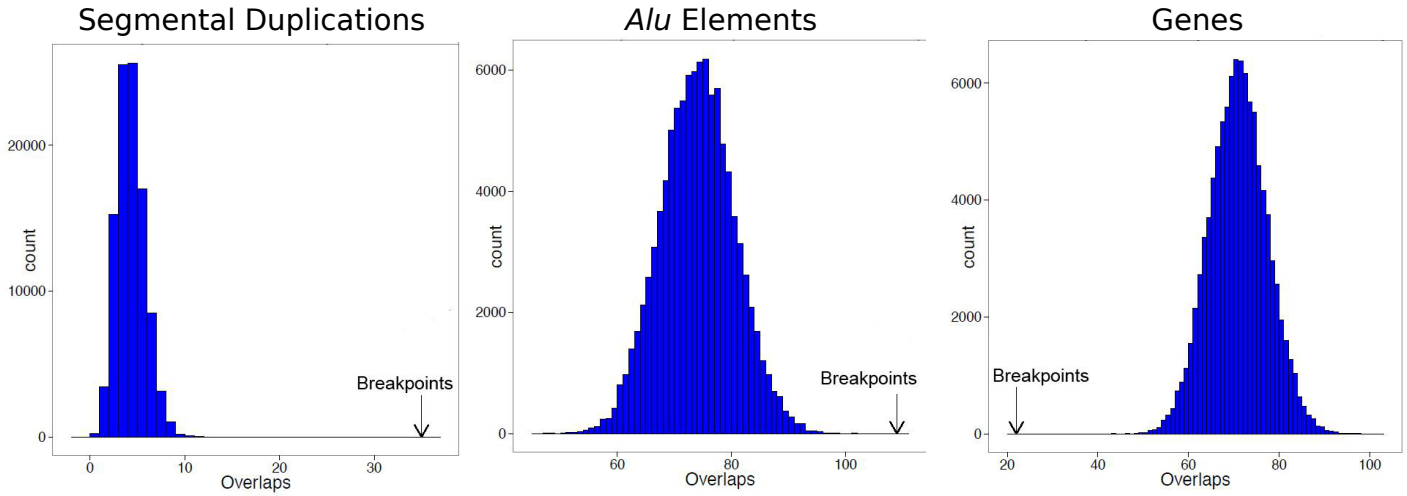
SVs can be present at different population scales

- Cells in a tumor (somatic mutations)
- Individuals in a population
- Between the genomes of different species

SVs in the Gibbon Genome

What can we infer about SV formation mechanisms and timing in a primate genome heavily rearranged by SVs?

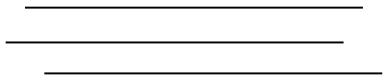
- Gibbons have very different karyotypes from the rest of the apes
- Search for enriched sequence features near evolutionary breakpoints
- Monte Carlo based location permutation approach



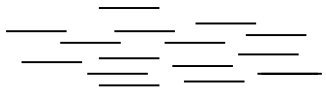
- Run tests, distribute permutations over cluster with Condor: `https://github.com/cwhelan/permuting-feature-enrichment-test`

Outline

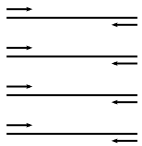
Short-Read Data for Resequencing



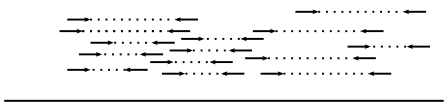
Input DNA from sample



Shear and amplify into many small fragments



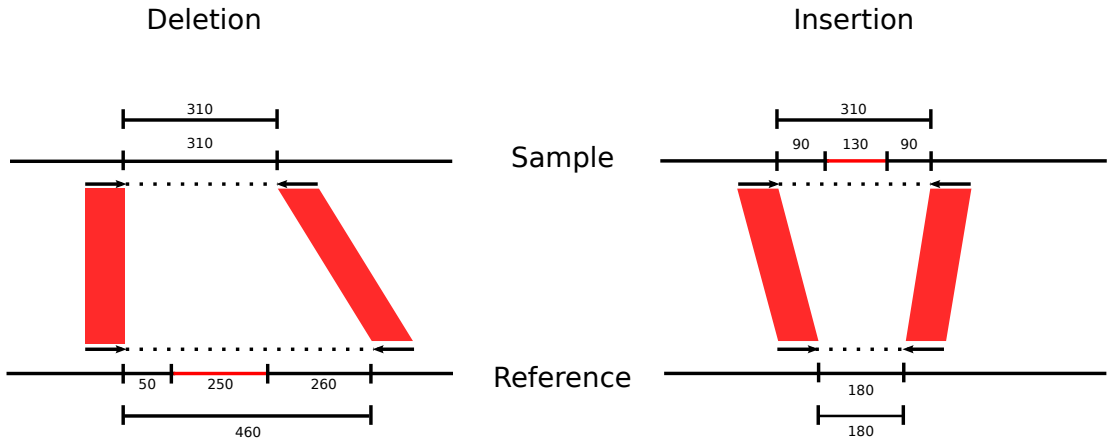
Sequence ends of fragments



Align paired reads to reference genome

SV Detection from Short-Read Sequencing Data

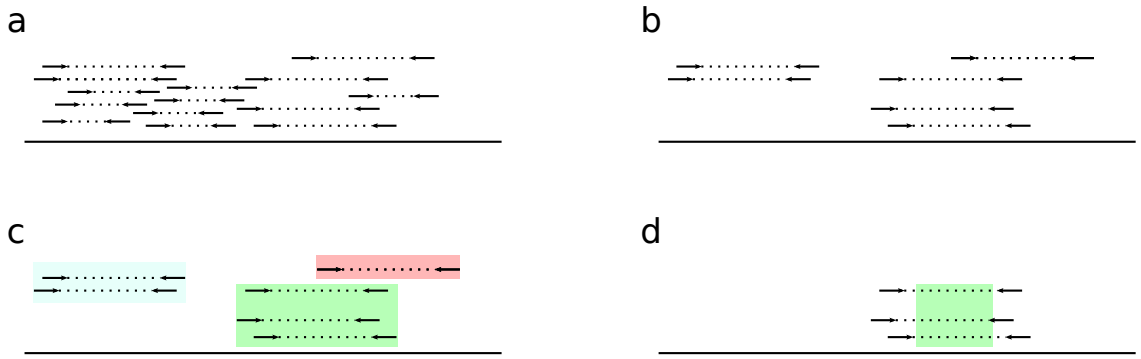
- Multiple signals of SVs in short read sequencing data
- Read-pair (RP) detection most commonly used



- Read depth (RD), Split-Read (SR), Assembly (AS)

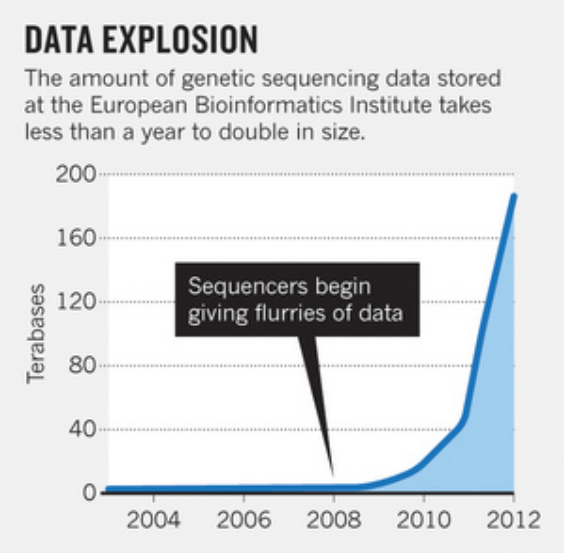
Read-pair approaches extend to consider more data

The Traditional Approach:



- Multiply-mapped read pairs
  - MrFAST - Hormozdiari et al (2009), HYDRA - Quinlan et al. (2010), GASVPro - Sindi et al. (2012)
- Concordant read pairs (Read depth signals, smaller variations)
  - GASVPro, CLEVER - Marschall et al. (2012)
- Split read signals
  - DELLY - Rausch et al (2012)
- All of the above + population-scale signals
  - Genome STRiP - Handsaker et al. (2011)
- Computational requirements increase with the amount of data examined
- How can we scale algorithms to new data sizes?

The amount of sequencing data is growing exponentially



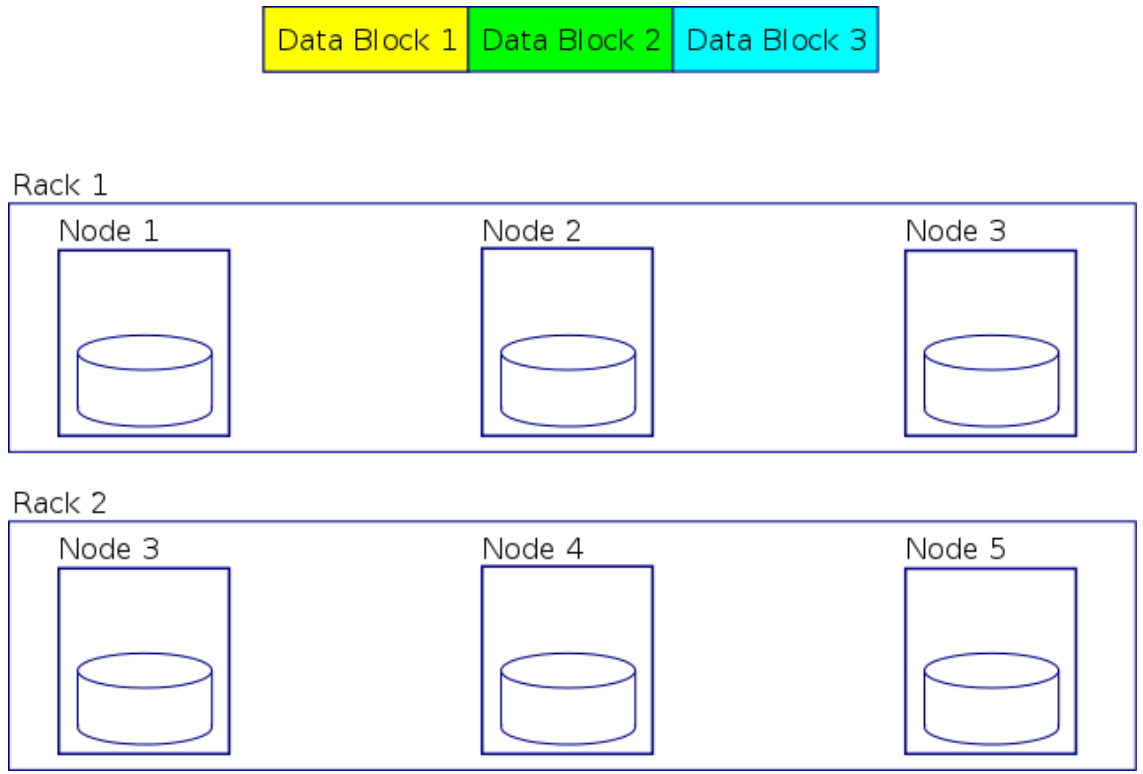
- For large scale research projects and eventual clinical applications, need to improve analysis throughput and latency
- Lots of recent efforts to shorten analysis pipelines (alignment and SNV calling)
- Structural variation detection can be left behind

The MapReduce distributed computing framework

- Created at Google for web-scale data (Dean and Gehemewat 2008)
- Open source Hadoop implementation
- Used by the web giants
- Distributes big data sets across cluster
- Provides a framework for parallelization
- Used in bioinformatics for:
  - Alignment/SNP calling (Crossbow - Langmead 2009)
  - RNA-seq (Myrna - Langmead 2010)

Hadoop/MapReduce is not just a job scheduler

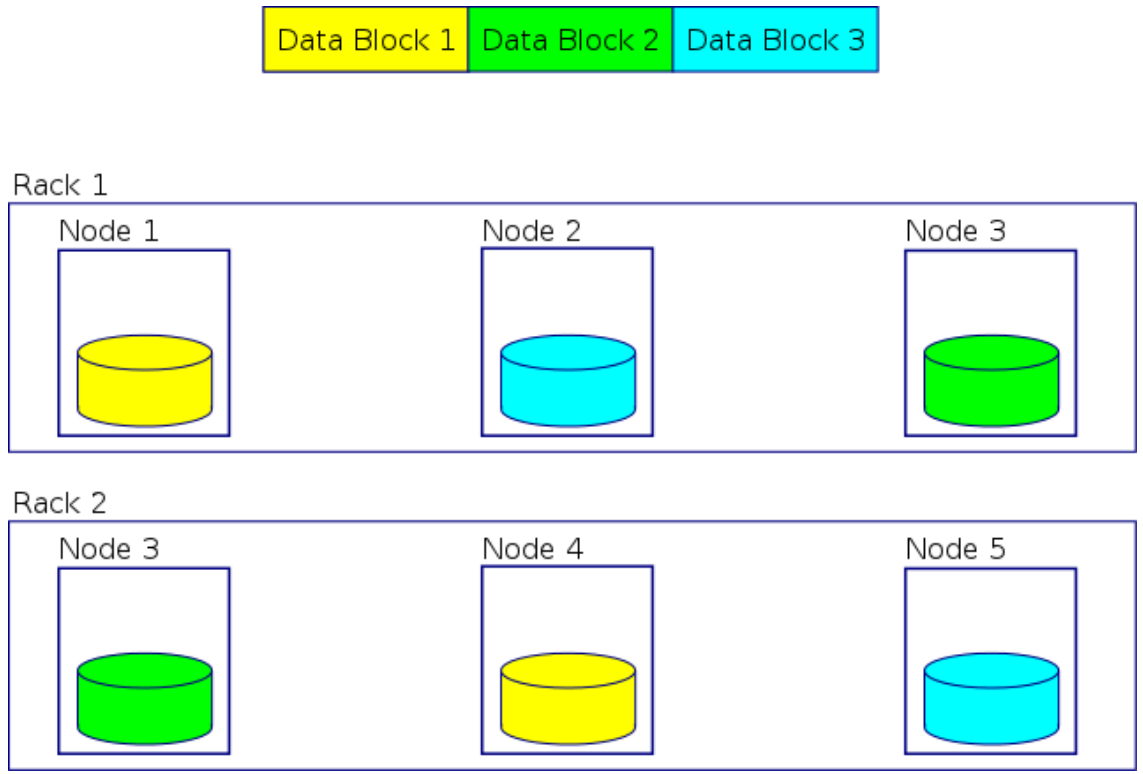
Why not just distribute with a file server and a grid scheduler?



Hadoop/MapReduce is not just a job scheduler

Why not just distribute with a file server and a grid scheduler?

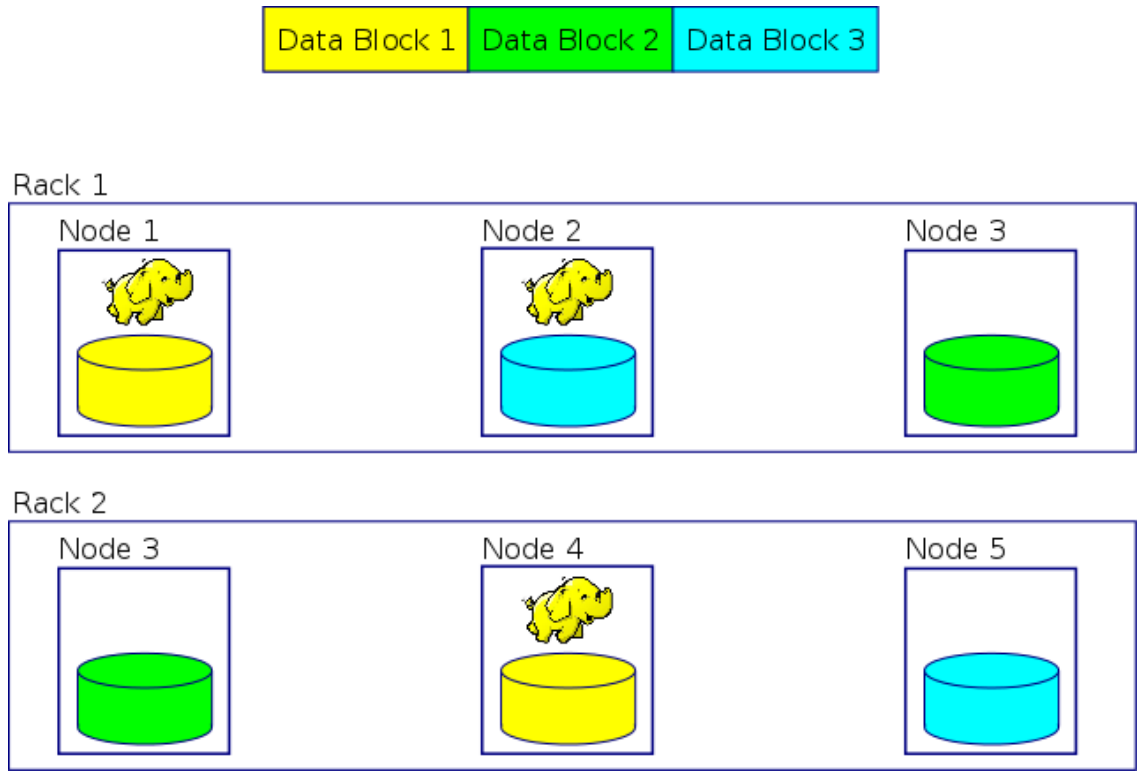
- Automatic redundancy of data



Hadoop/MapReduce is not just a job scheduler

Why not just distribute with a file server and a grid scheduler?

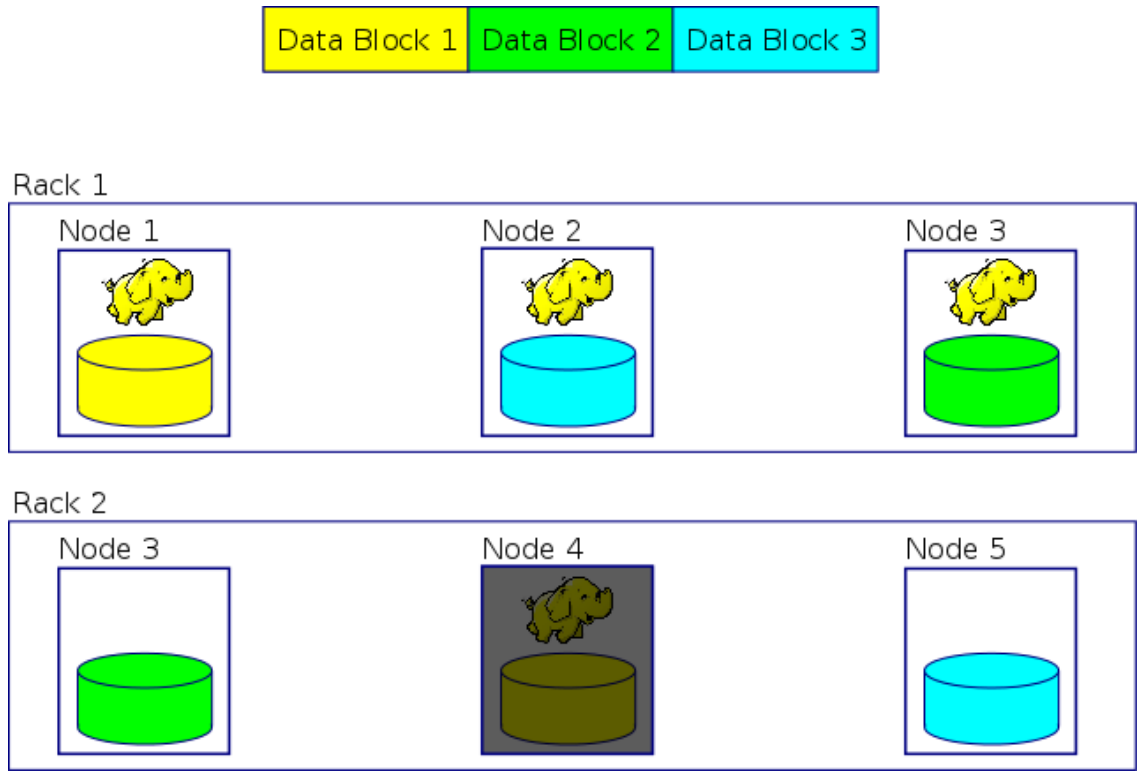
- Automatic redundancy of data
- Takes into account data locality



Hadoop/MapReduce is not just a job scheduler

Why not just distribute with a file server and a grid scheduler?

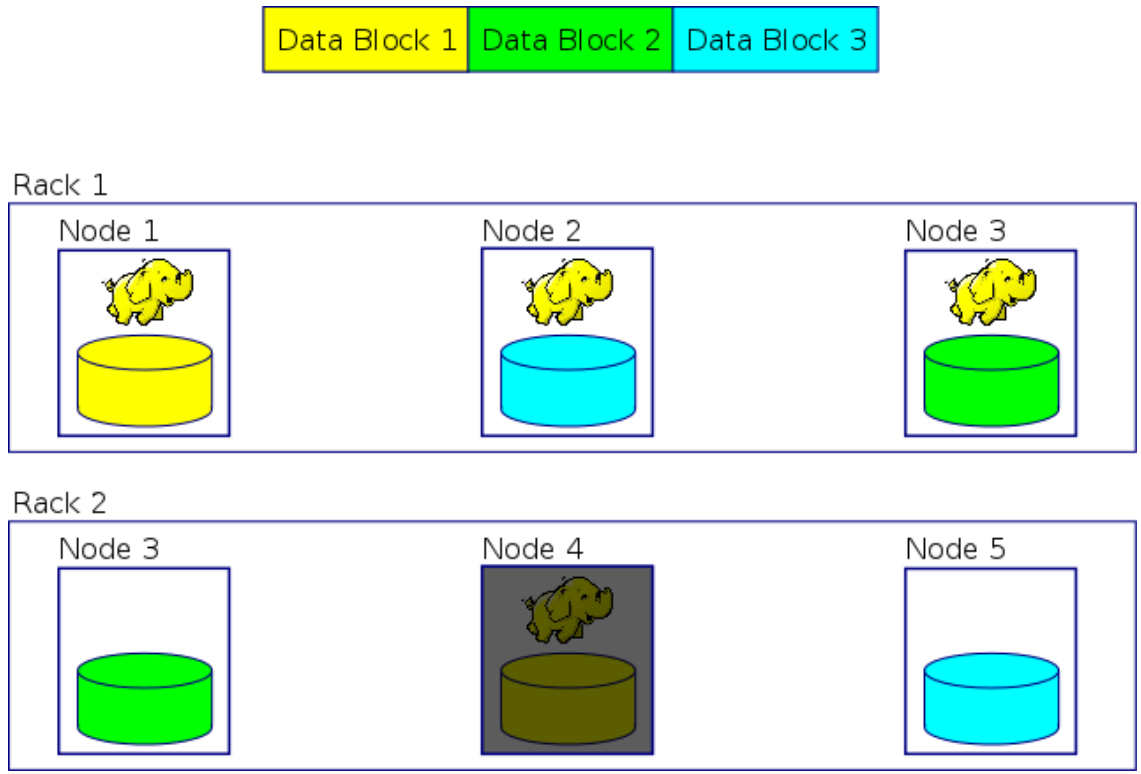
- Automatic redundancy of data
- Takes into account data locality
- Can handle failure of data or workers transparently



Hadoop/MapReduce is not just a job scheduler

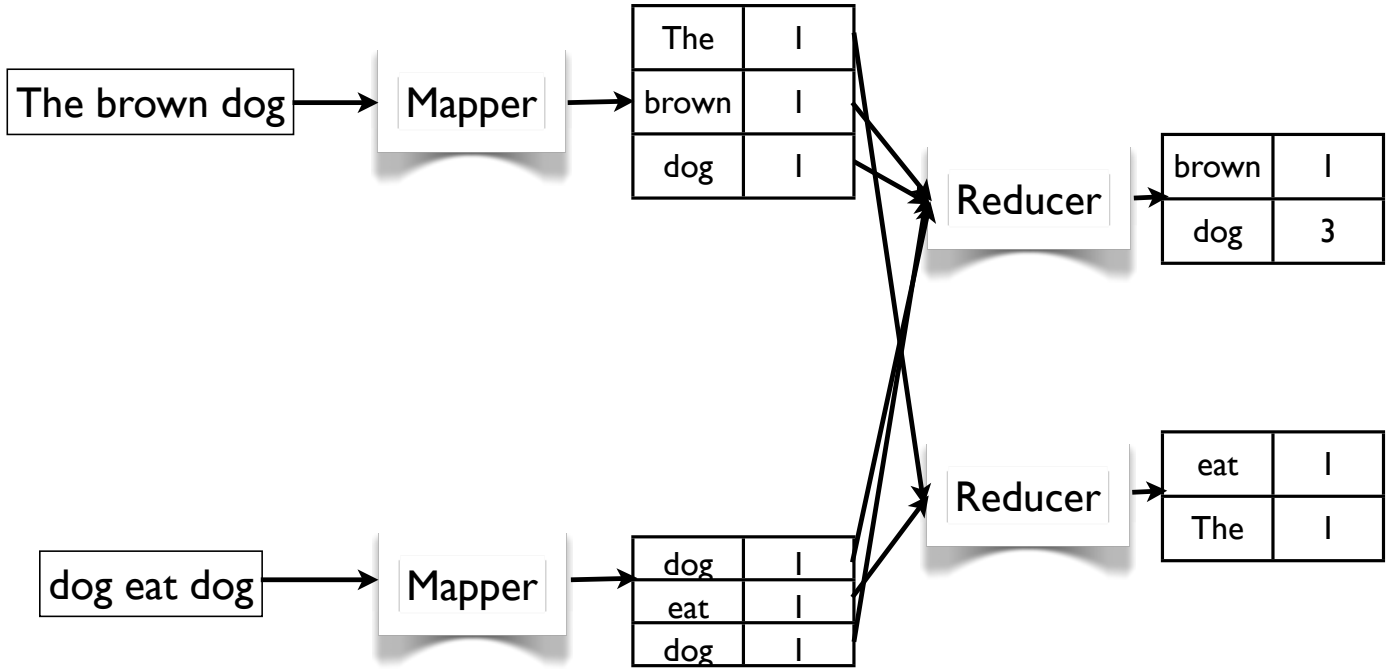
Why not just distribute with a file server and a grid scheduler?

- Automatic redundancy of data
- Takes into account data locality
- Can handle failure of data or workers transparently
- Can scale linearly with the amount of data, number of nodes



Developing algorithms in MapReduce

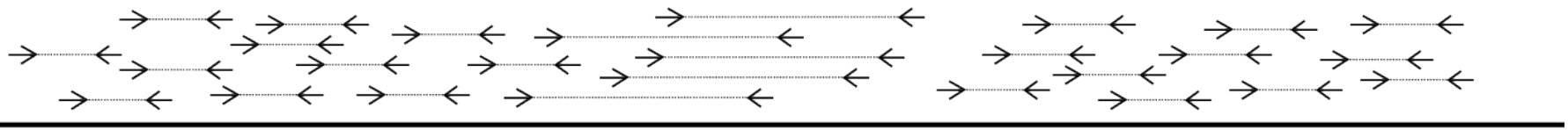
- To get these properties there are constraints on application developers
- MapReduce divides execution into *Map* and *Reduce* tasks



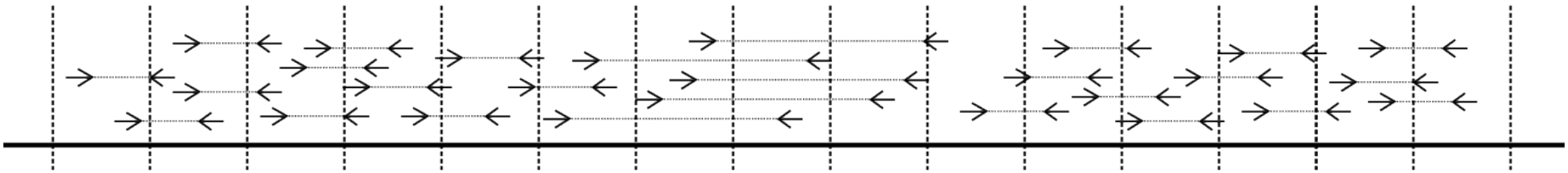
- Need to design algorithms that run independent calculations on small chunks of data
- How can we implement SV calling in this framework?

Outline

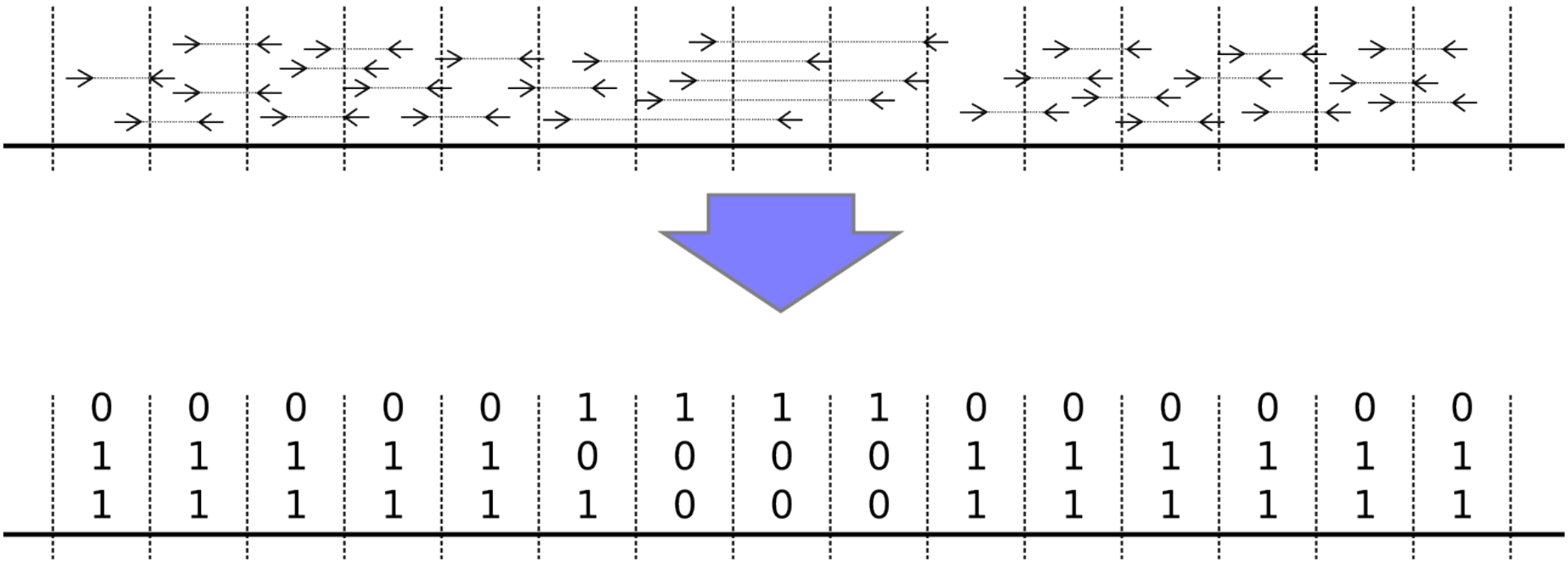
Computing local features enables parallelization



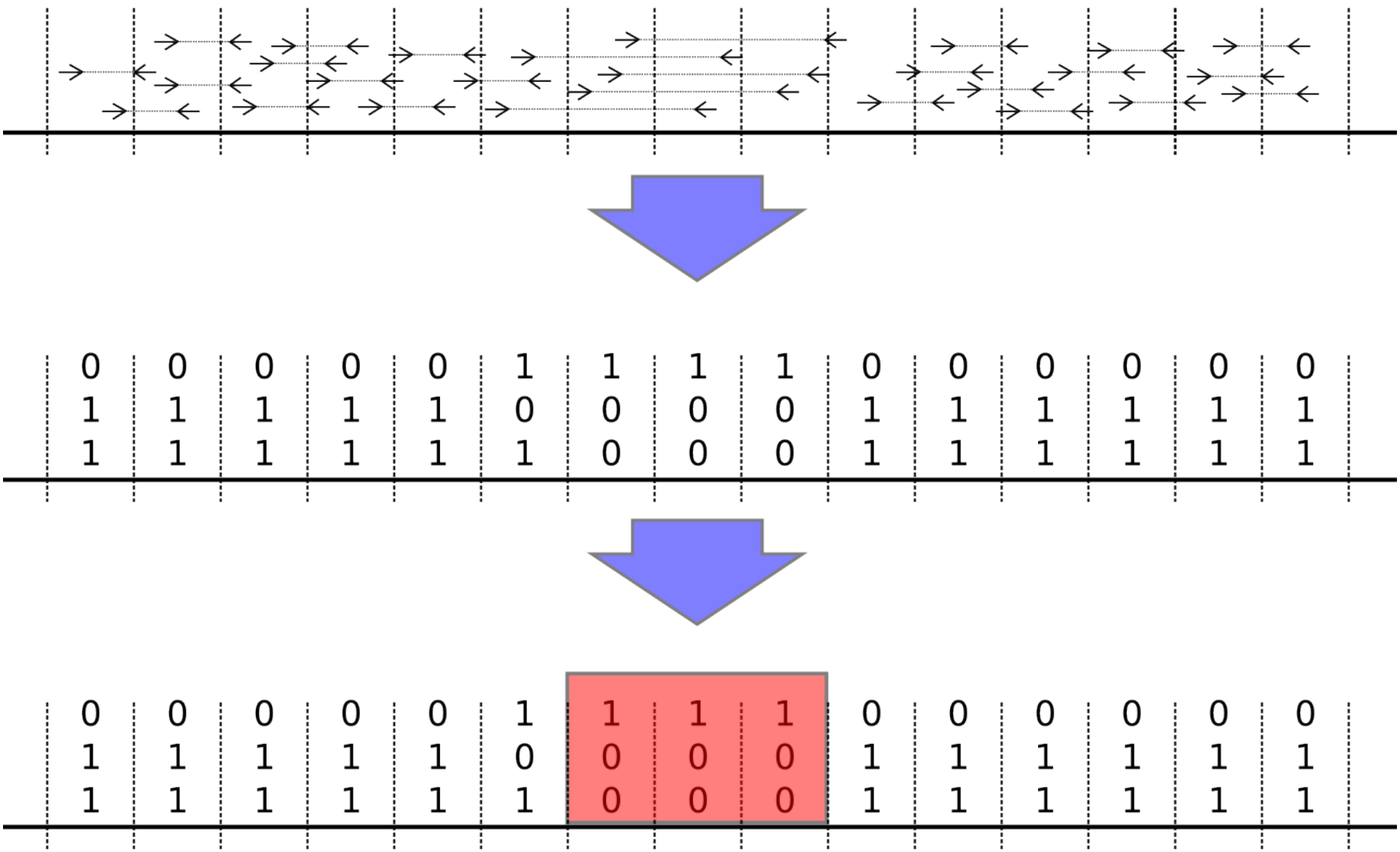
Computing local features enables parallelization



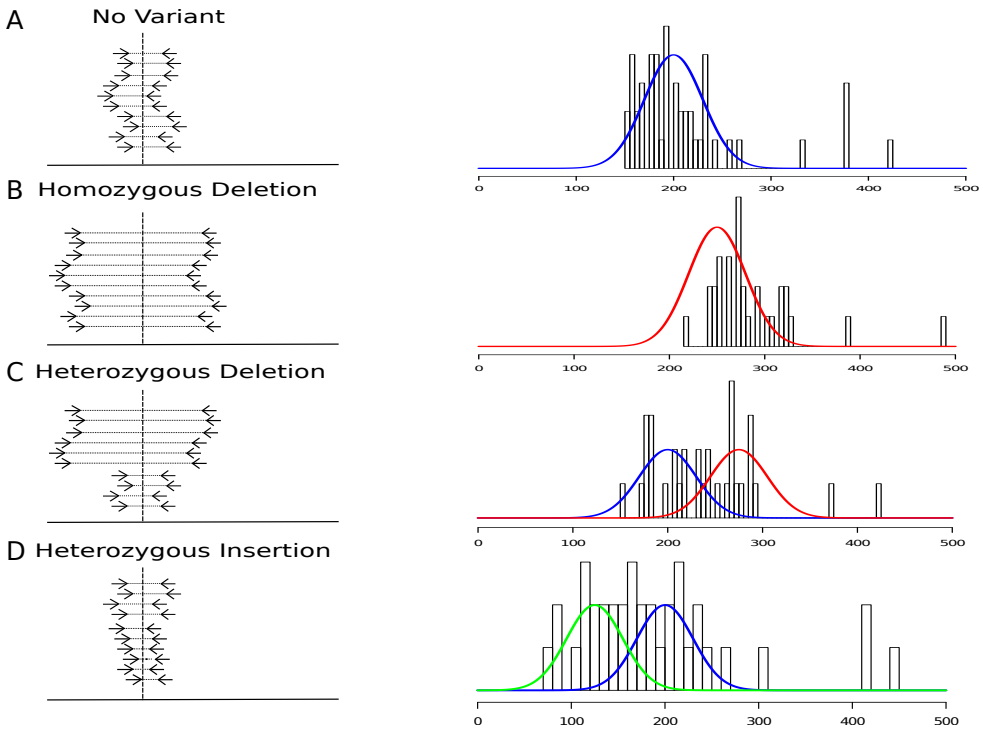
Computing local features enables parallelization



Computing local features enables parallelization

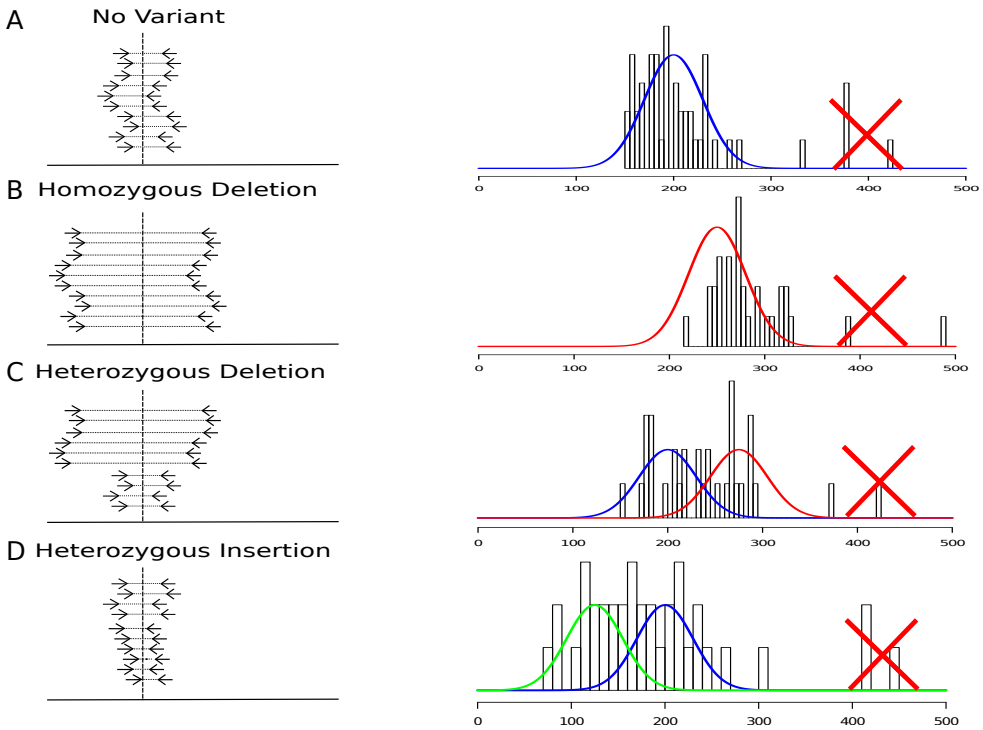


Mixture of distributions provides local features



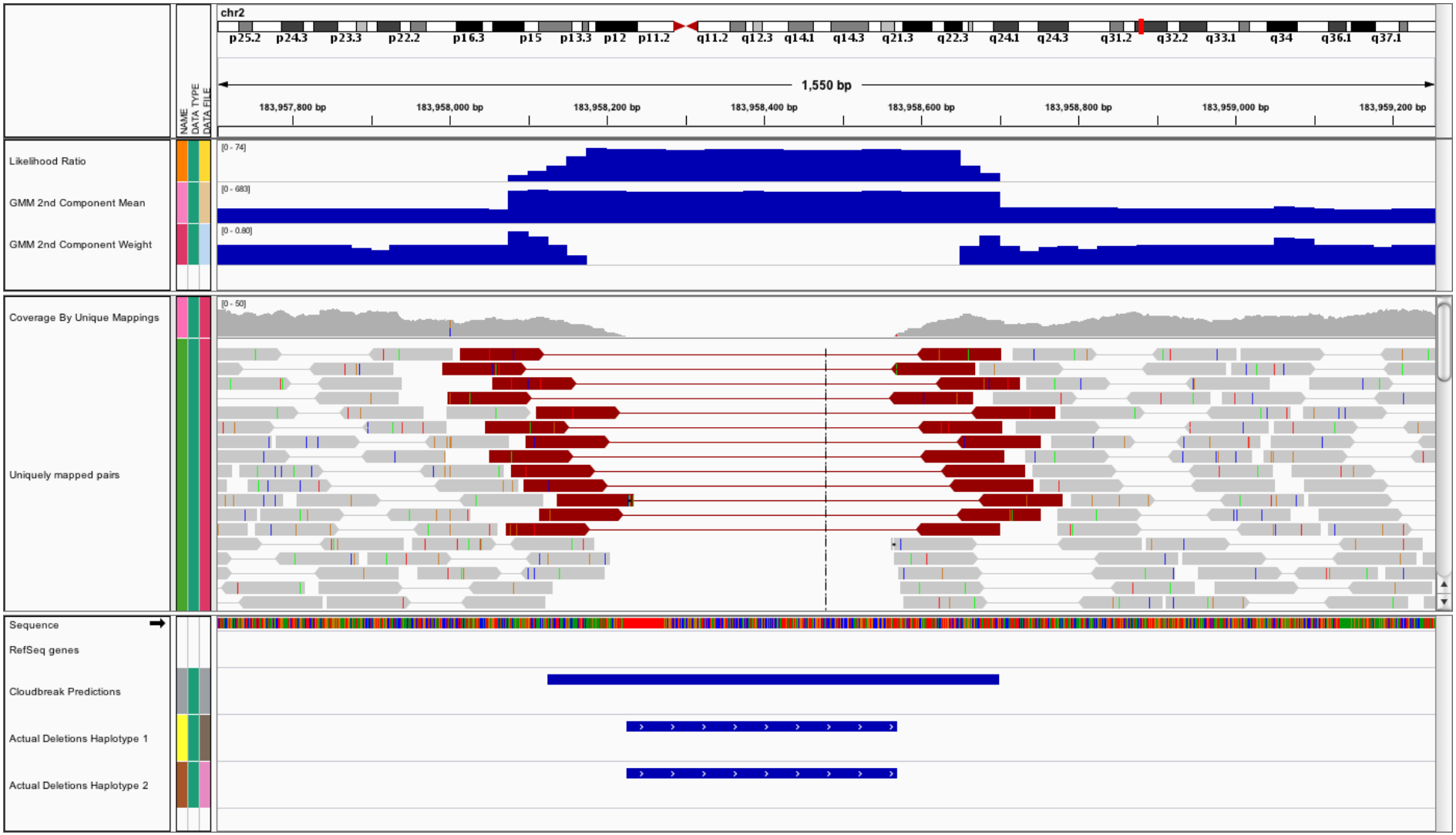
- Single sample, insertions and deletions (40bp - 25kb)
- General idea first proposed by MoDIL (Lee et al. 2009)
- Fit a Gaussian Mixture Model and generate features:
  - LR: likelihood ratio of two-component model to no-variant model
  - μ': estimated mean of second component
  - α: estimated mixing weight of two components

Mixture of distributions provides local features



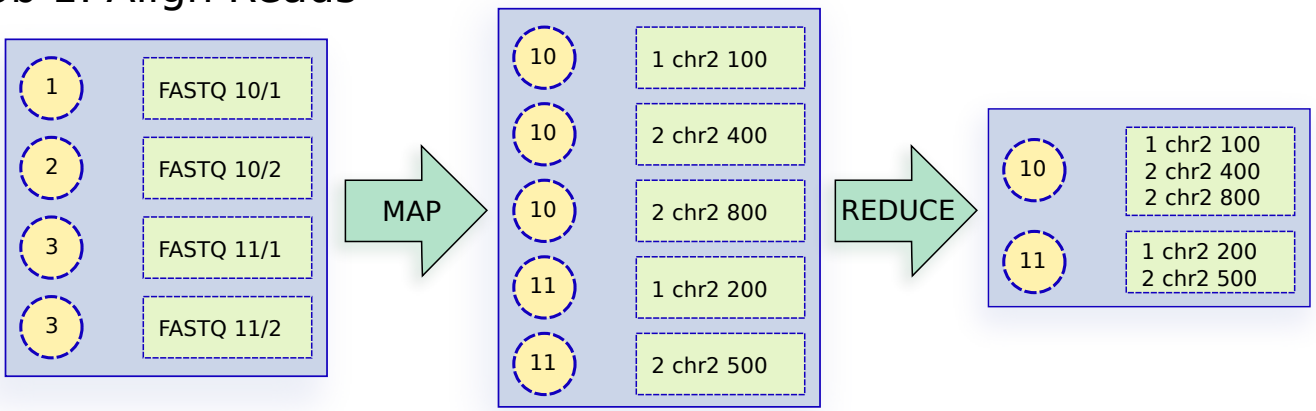
- Single sample, insertions and deletions (40bp - 25kb)
- General idea first proposed by MoDIL (Lee et al. 2009)
- Fit a Gaussian Mixture Model and generate features:
  - LR: likelihood ratio of two-component model to no-variant model
  - μ': estimated mean of second component
  - α: estimated mixing weight of two components

Cloudbreak output example

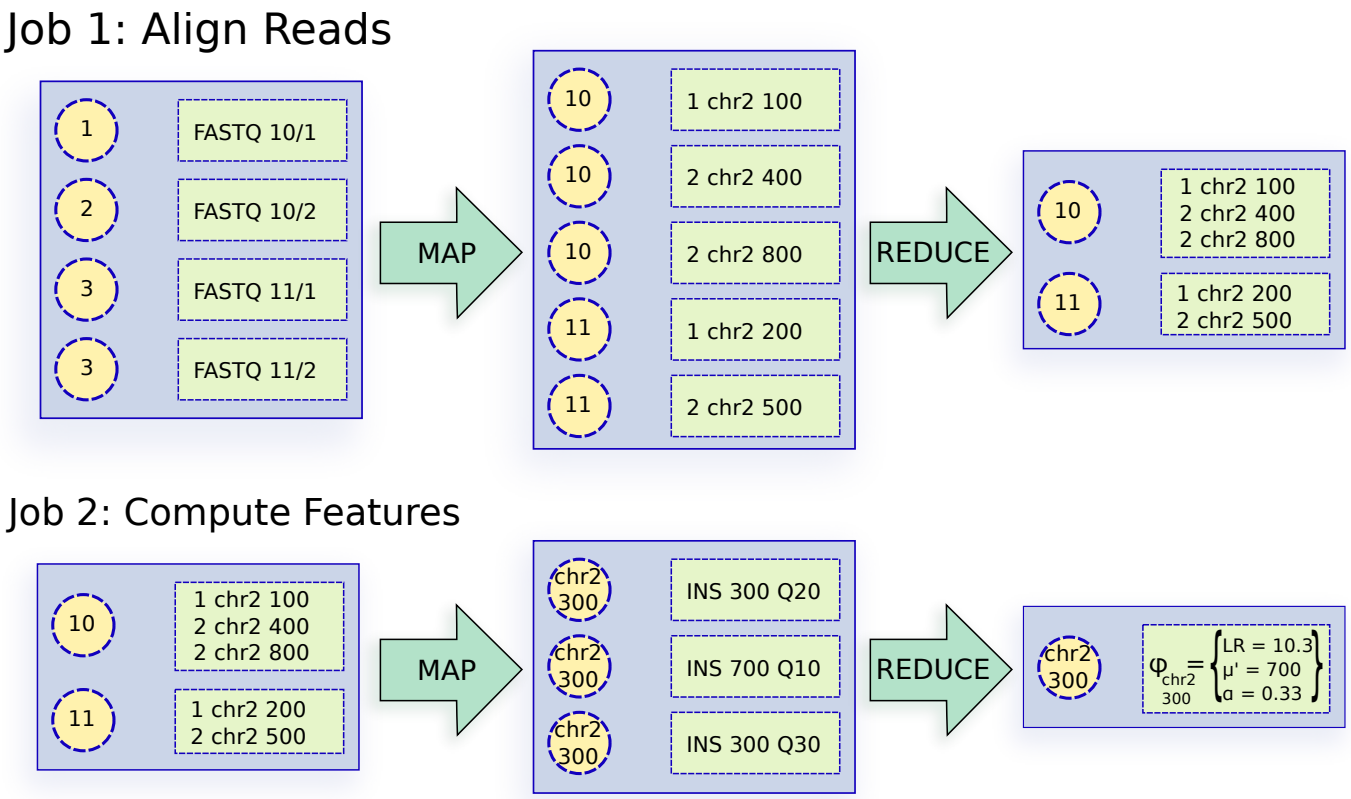


Cloudbreak Algorithm Job 1: Alignment

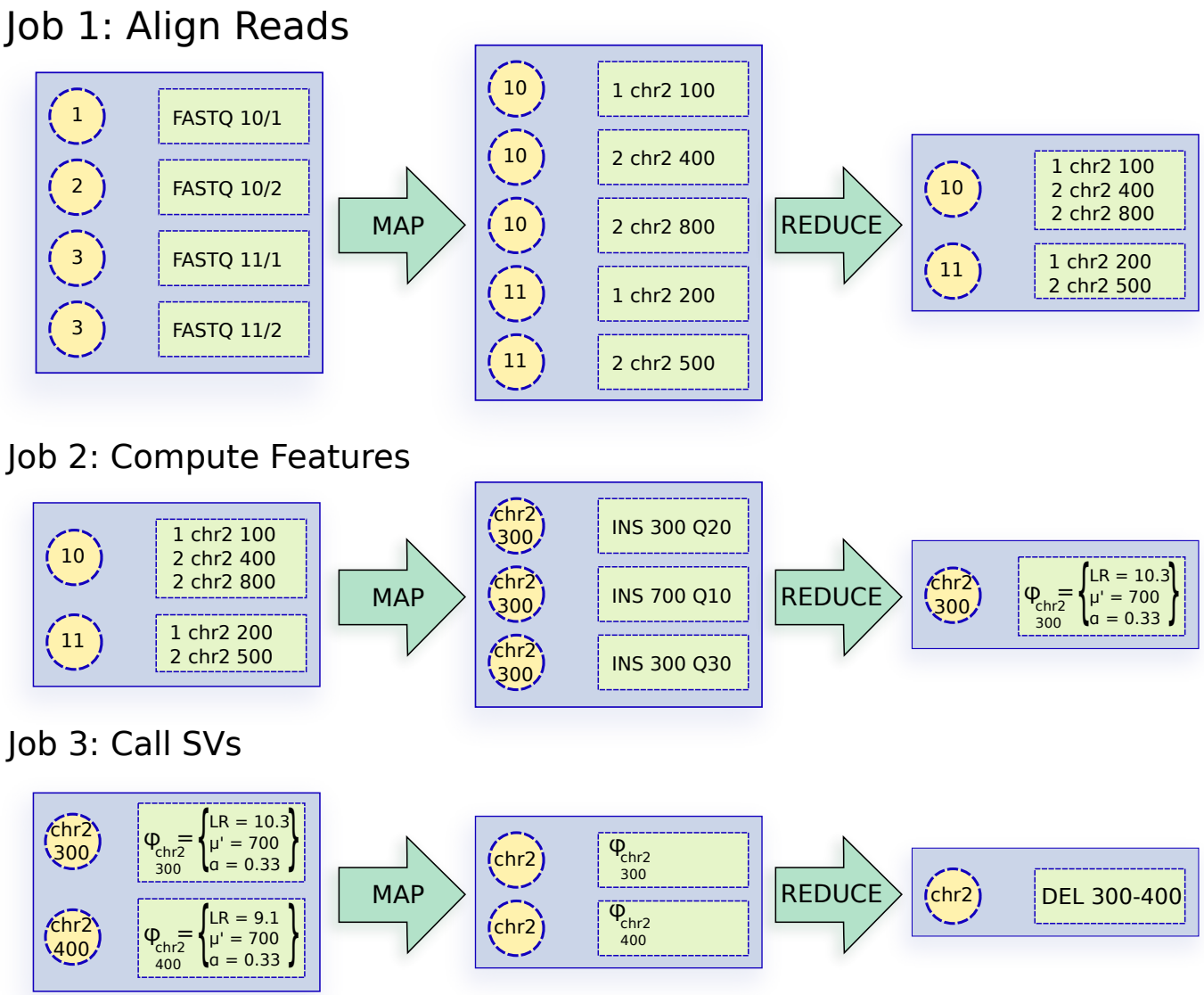
Job 1: Align Reads



Cloudbreak Algorithm Job 2: Compute Features



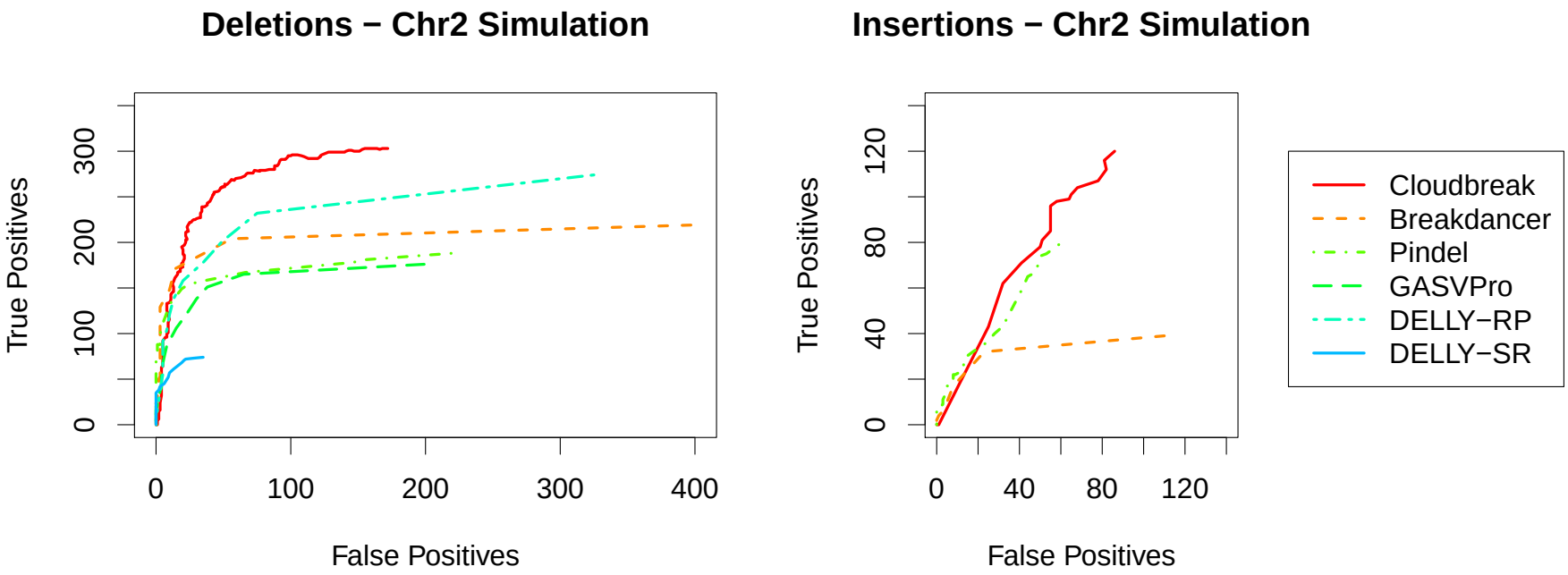
Cloudbreak Algorithm Job 3: Call SVs



Evaluation

- Simulated data
  - Indels from the HuRef genome applied to chr2 reference
  - Diploid
  - 30X, 100bp reads, 100bp internal insert size
- Real data from NA18507
  - 37X, 100bp reads, 112bp internal insert size
  - Gold standard set with merged calls from 1000 Genomes Project, Mills et al. (2011b), Kidd et al. (2008)
- Matching criteria
  - Overlap with length difference < 300bp
- Comparisons
  - RP method Breakdancer (Chen et al. 2009)
  - RP/SR method DELLY (Rausch et al. 2012)
  - RP/RD method GASVPro (Sindi et al. 2012)
  - SR method Pindel (Ye et al. 2009)

ROC Curves for Chromosome 2 Simulation



- Caveat: Methods perform better on simulated data than on real whole genome datasets.

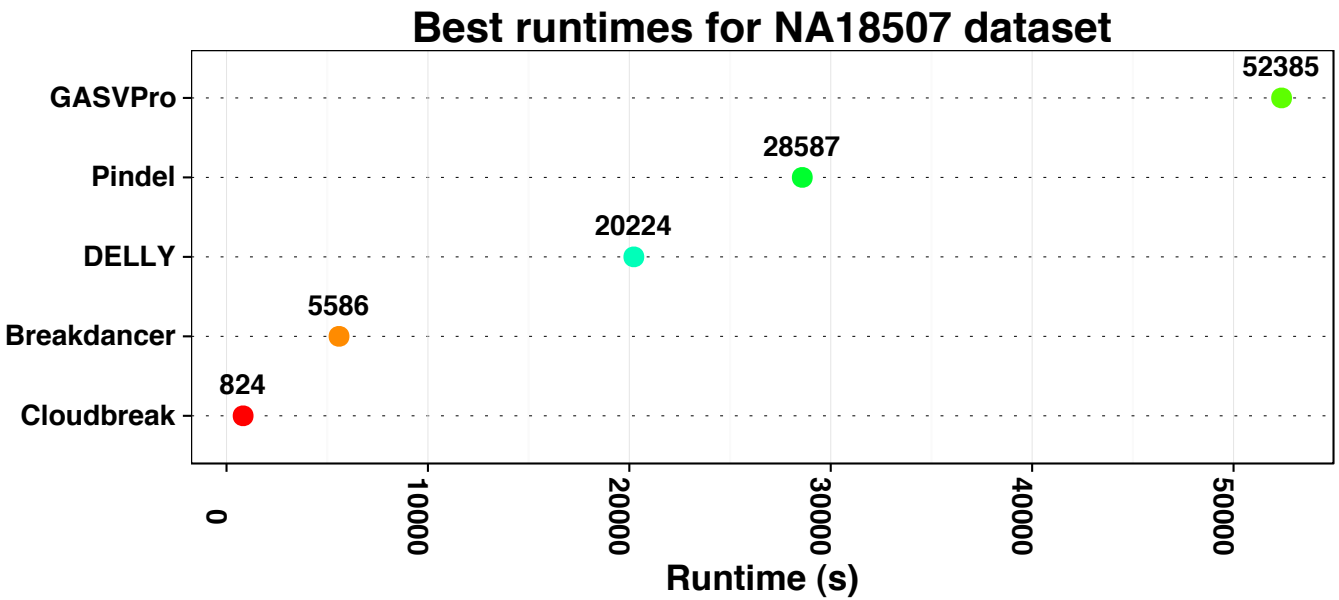
Ability to find simulated variants at a controlled FDR

- The number of simulated deletions found by each tool at a 10% FDR, as well as the number of those deletions that were discovered exclusively by each tool (in parentheses).

	40-100bp	101-250bp	251-500bp	501-1000bp	> 1000bp
Total Number	224	84	82	31	26
Cloudbreak	**68** (17)	**67** (**10**)	**56** (**5**)	**11** (**3**)	**15** (**0**)
Breakdancer	52 (8)	49 (2)	49 (0)	7 (0)	14 (**0**)
GASVPro	35 (2)	26 (0)	26 (0)	2 (0)	6 (**0**)
DELLY-RP	22 (1)	56 (1)	40 (0)	8 (0)	12 (**0**)
DELLY-SR	0 (0)	2 (0)	28 (0)	2 (0)	10 (**0**)
Pindel	60 (**32**)	16 (0)	41 (2)	1 (0)	12 (**0**)

ROC Curves for NA18507

Run Times

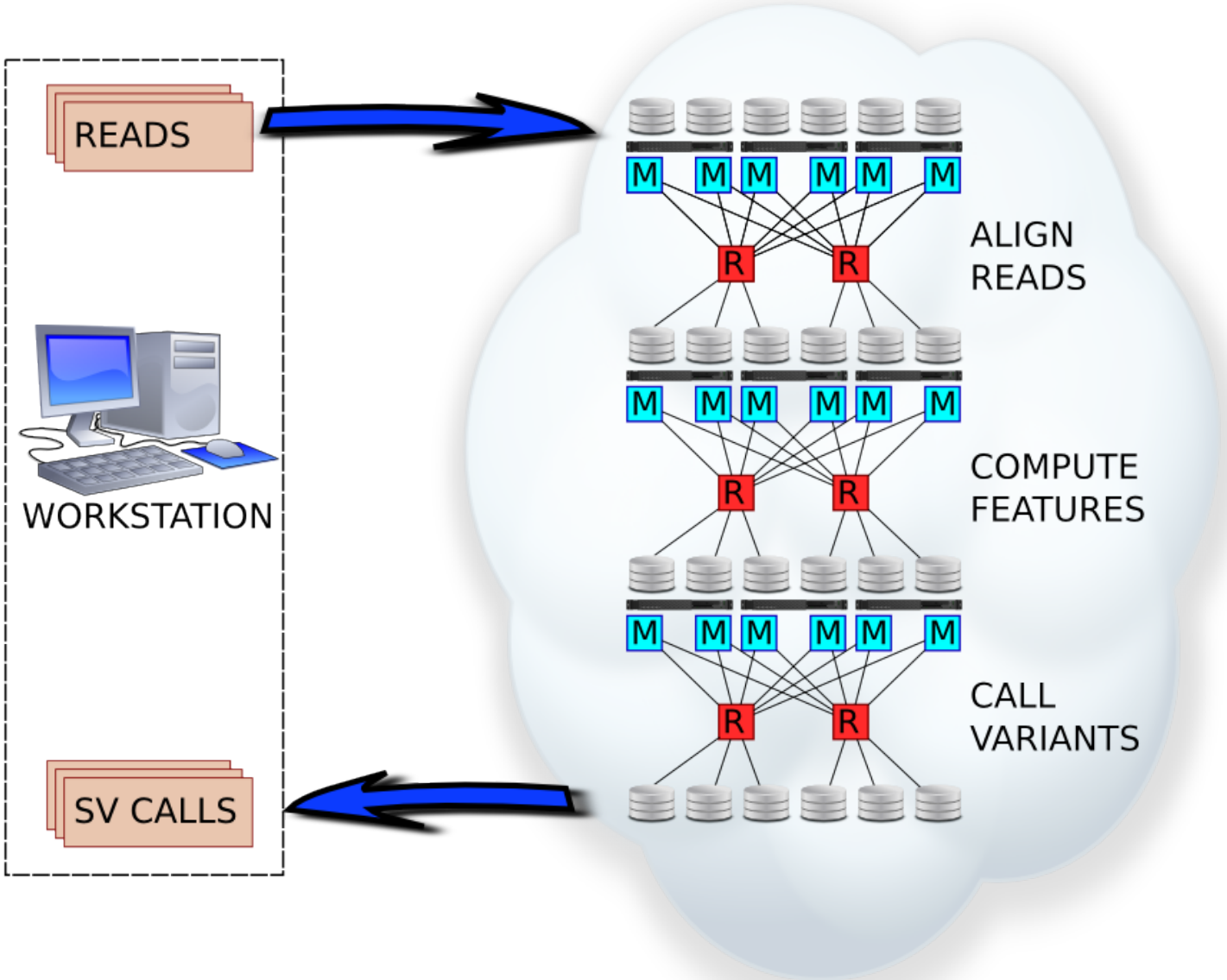


		Simulated Data			NA18507		
	SV Types	Single CPU	Parallel	Proc.	Single CPU	Parallel	Proc.
Cloudbreak	D,I	NA	290	312	NA	824	636
Breakdancer	D,I,V,T	653	NA	NA	134,170	5,586	84
GASVPro	D,V	3,339	NA	NA	52,385	NA	NA
DELLY	D	1,964	NA	NA	30,311	20,224	84
Pindel	D,I,V,P	37,006	4,885	8	284,932	28,587	84
MoDIL	D,I	NA	52,547	250	NA	NA	NA

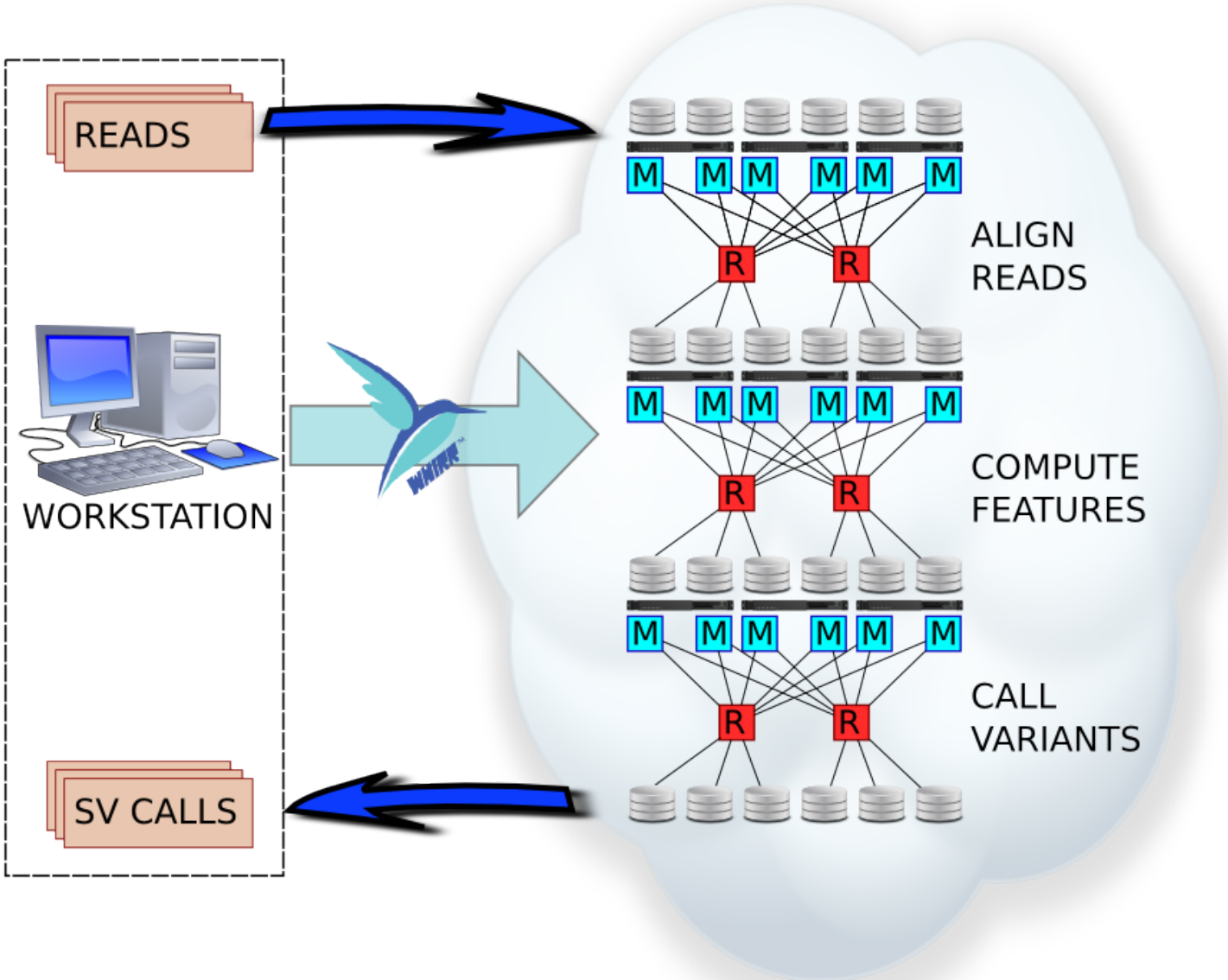
SV Types - D: Deletions; I: Insertions; V: Inversions; T: Translocations; P: Duplications

All times in seconds

Automatic provisioning of clusters in the cloud

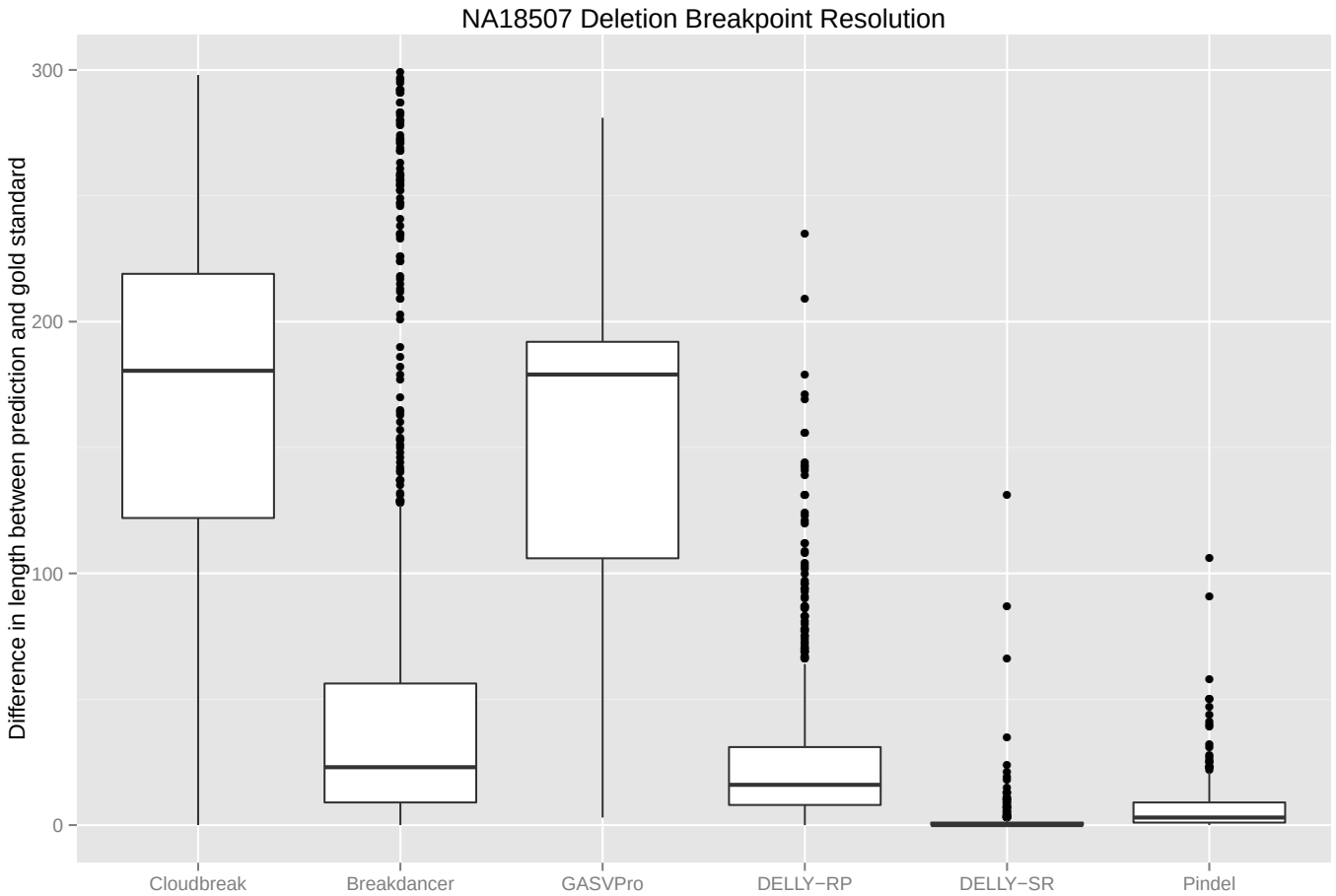


Automatic provisioning of clusters in the cloud



Cloudbreak Limitations

- Very fast to process once in Hadoop, but importing data takes time
- More CPU hours used
- Lower breakpoint resolution



Cloudbreak Summary

- Novel approach to applying MapReduce/Hadoop to the structural variation detection problem using the computation of local features
- Works for insertions and deletions
  - Generalizable to other variant types?
- Ability to accurately genotype calls
- Greatly decreased runtimes at the cost of more CPU hours
- Fast runtimes + accurate region detection but low breakpoint resolution
  - Suggests initial scan before more in depth *in silico* verification (split read mapping; local assembly; discriminative machine learning)
- `http://github.com/cwhelan/cloudbreak`

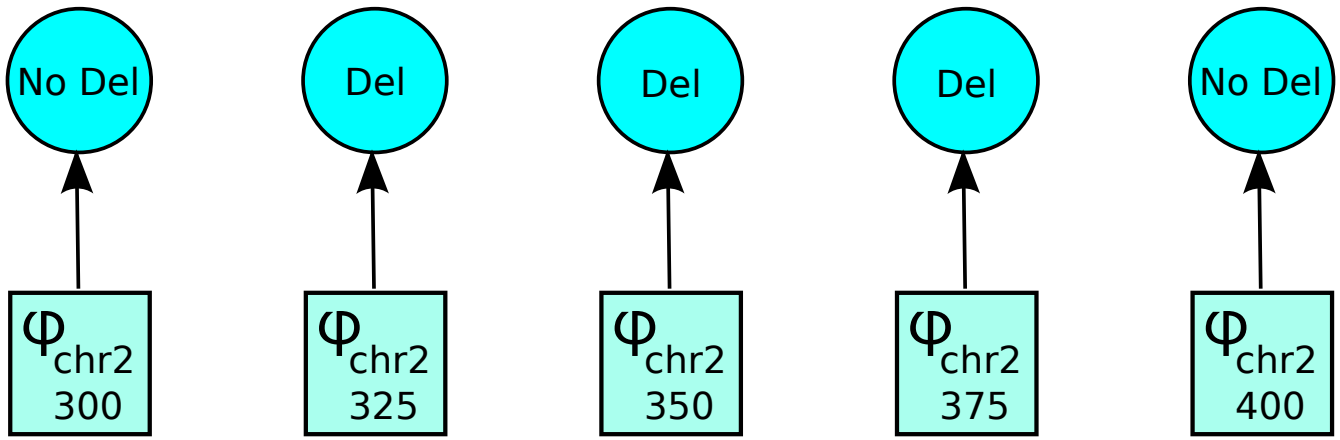
Outline

Can Cloudbreak be extended to consider more signals?

- So far have talked primarily about RP based SV detection
- What about other signals? RD, SR, prior knowledge?
- Some tools integrate multiple signals, but primarily with *ad hoc* rules:
  - DELLY: verify RP predictions with SR mappings
  - GASVPro: look for RD signals to lend support to RP predictions, e.g. in deletion regions or at inversion breakpoints
  - Genome STRiP: combine many signals to genotype loci in a population
- Cloudbreak's formulation in terms of local features lends itself to integration of signals

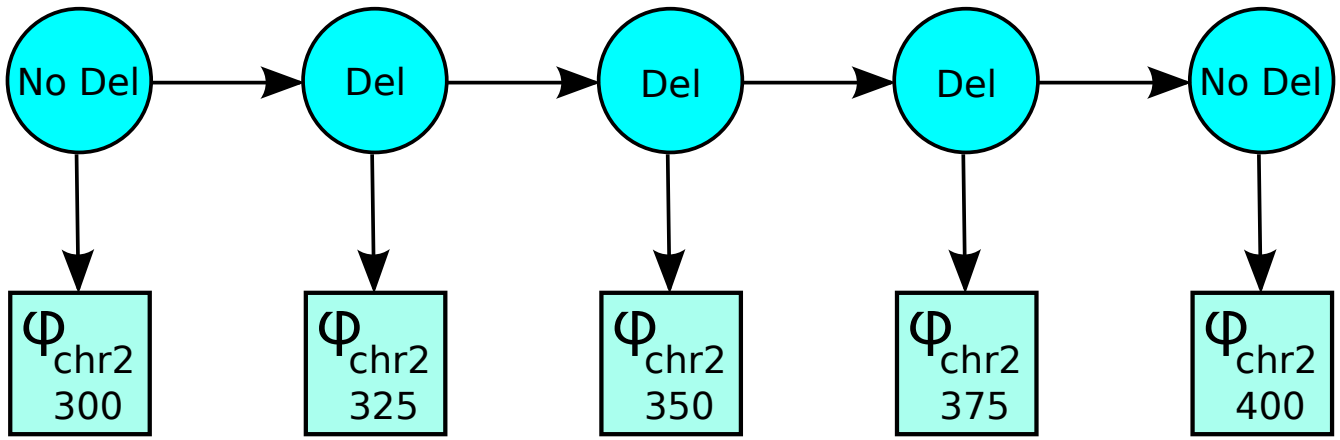
Integrating many features in a sequential labeling problem

- For Cloudbreak, we formulated the problem as finding the label of each window (i.e. Deletion/No Deletion) based on window features
- Let labels be $Y_1, Y_2, ..., Y_n$ and feature observations be $X_1, X_2, ..., X_n$
- Should be able to model sequential relationship of labels...



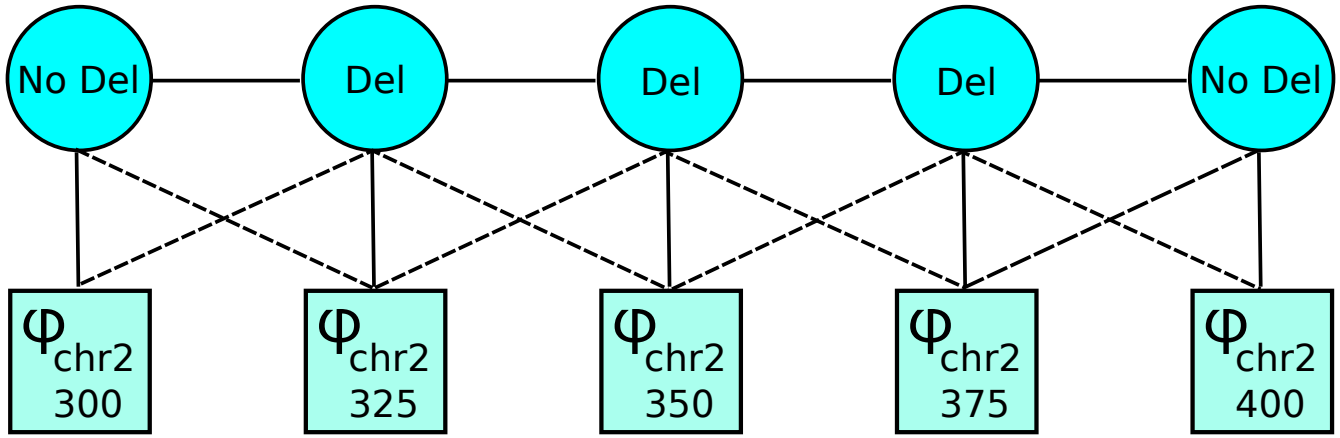
Integrating many features in a sequential labeling problem

- For Cloudbreak, we formulated the problem as finding the label of each window (i.e. Deletion/No Deletion) based on window features
- Let labels be $Y_1, Y_2, ..., Y_n$ and feature observations be $X_1, X_2, ..., X_n$
- Could use a Hidden Markov Model
  - Models the joint probability of labels and observations: $P(X, Y)$.
  - One problem: nearby features might be highly correlated, but HMM assumes that $P(X_i|Y_1, Y_2, ..., Y_i, X_1, X_2, ..., X_{i-1}) = P(X_i|Y_i)$
  - Another: We are modeling $P(X, Y) = P(Y|X)P(X)$, but we only really care about $P(Y|X)$



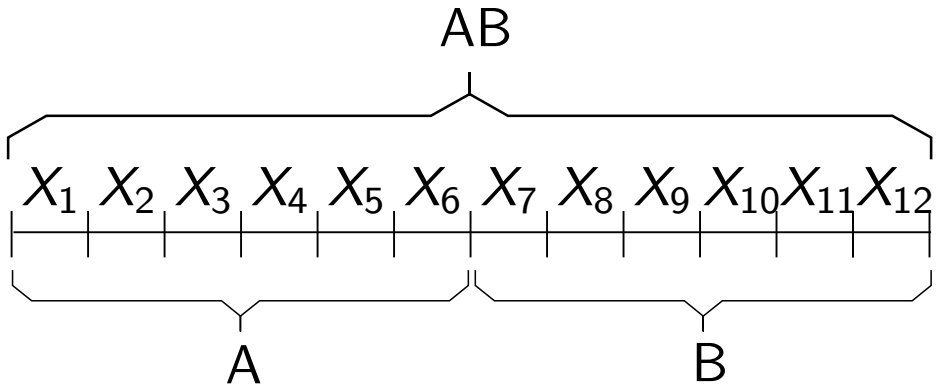
Integrating many features in a sequential labeling problem

- For Cloudbreak, we formulated the problem as finding the label of each window (i.e. Deletion/No Deletion) based on window features
- Let labels be $Y_1, Y_2, ..., Y_n$ and feature observations be $X_1, X_2, ..., X_n$
- Conditional Random Fields
  - Undirected graphical model
  - Labels and observations are related by feature functions, which can incorporate data from any of the observations
  - Can be trained to learn $P(Y|X)$ directly (discriminative)



CRF Features

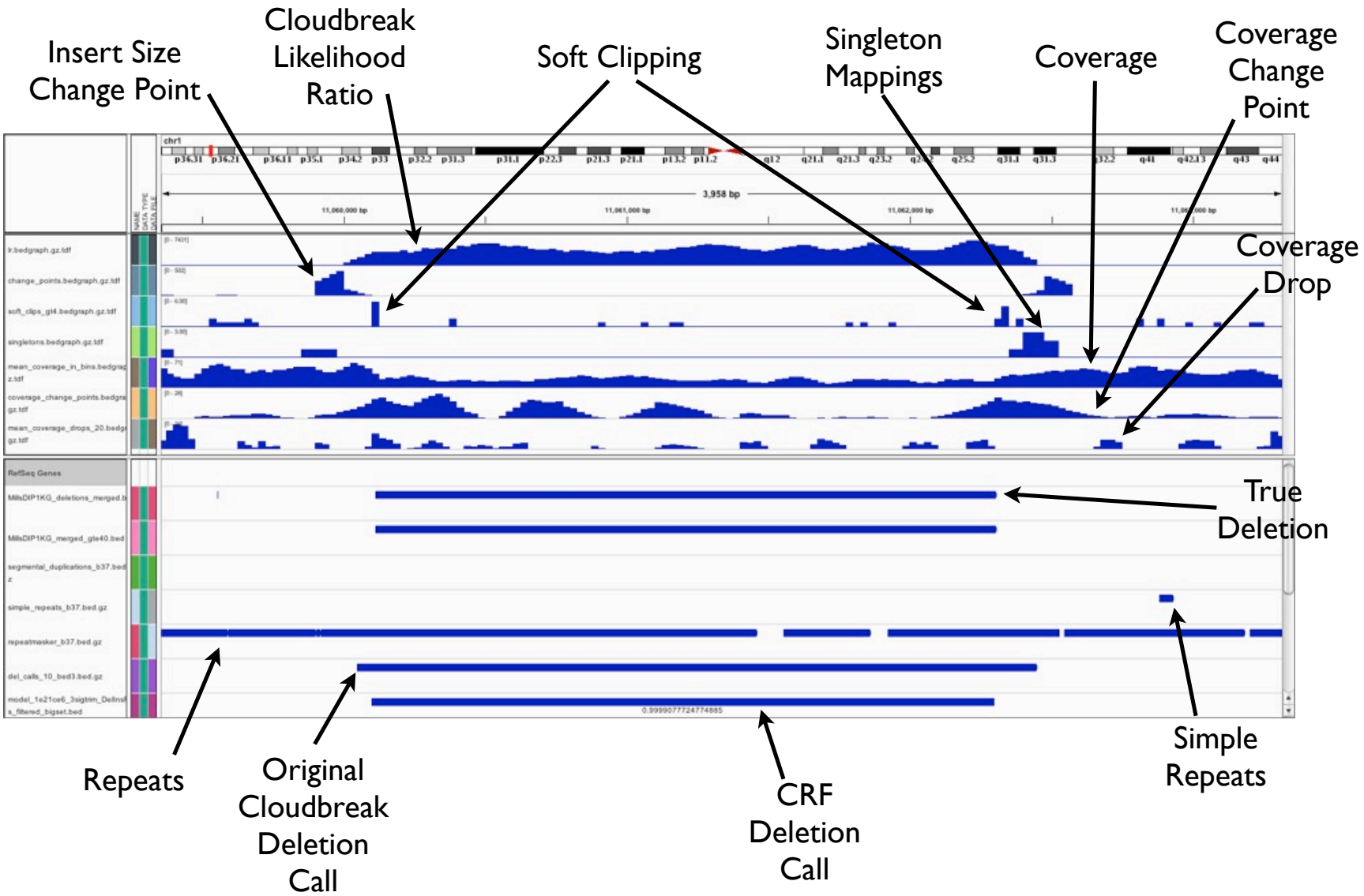
- Insert sizes (RP)
  - GMM Features from Cloudbreak (likelihood ratio, μ', α)
  - Sliding window change point detection



$$CP_6 = \frac{\mathcal{L}(A)\mathcal{L}(B)}{\mathcal{L}(AB)}$$

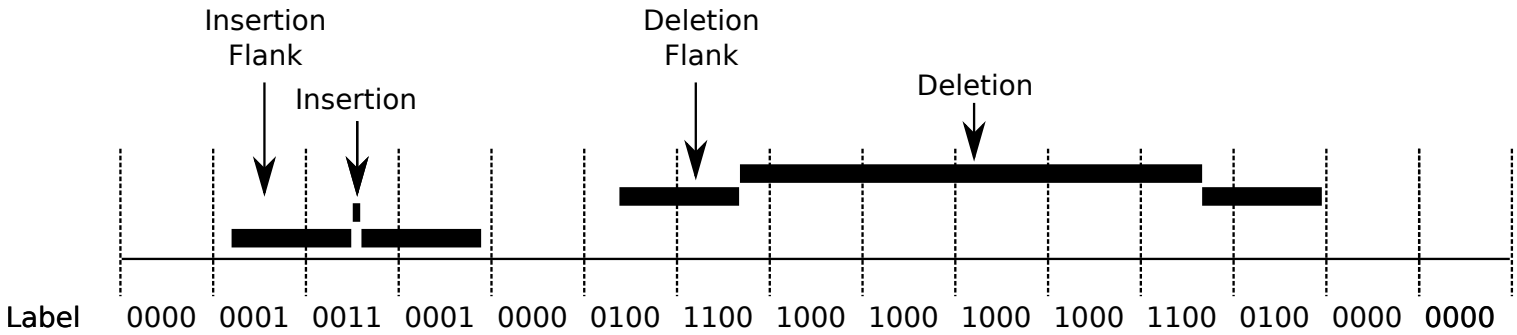
- Depth and depth changes (RD)
  - Average depth of coverage by reads in window
  - Coverage change points
  - Sharp drops in coverage from surrounding neighborhood
- Split-Read (SR) related signals
  - *Soft clipped* mappings
  - Singleton mappings
- Genome annotations
  - Segmental duplications
  - Annotated repeats and simple repeats

CRF Features



CRF Strategy

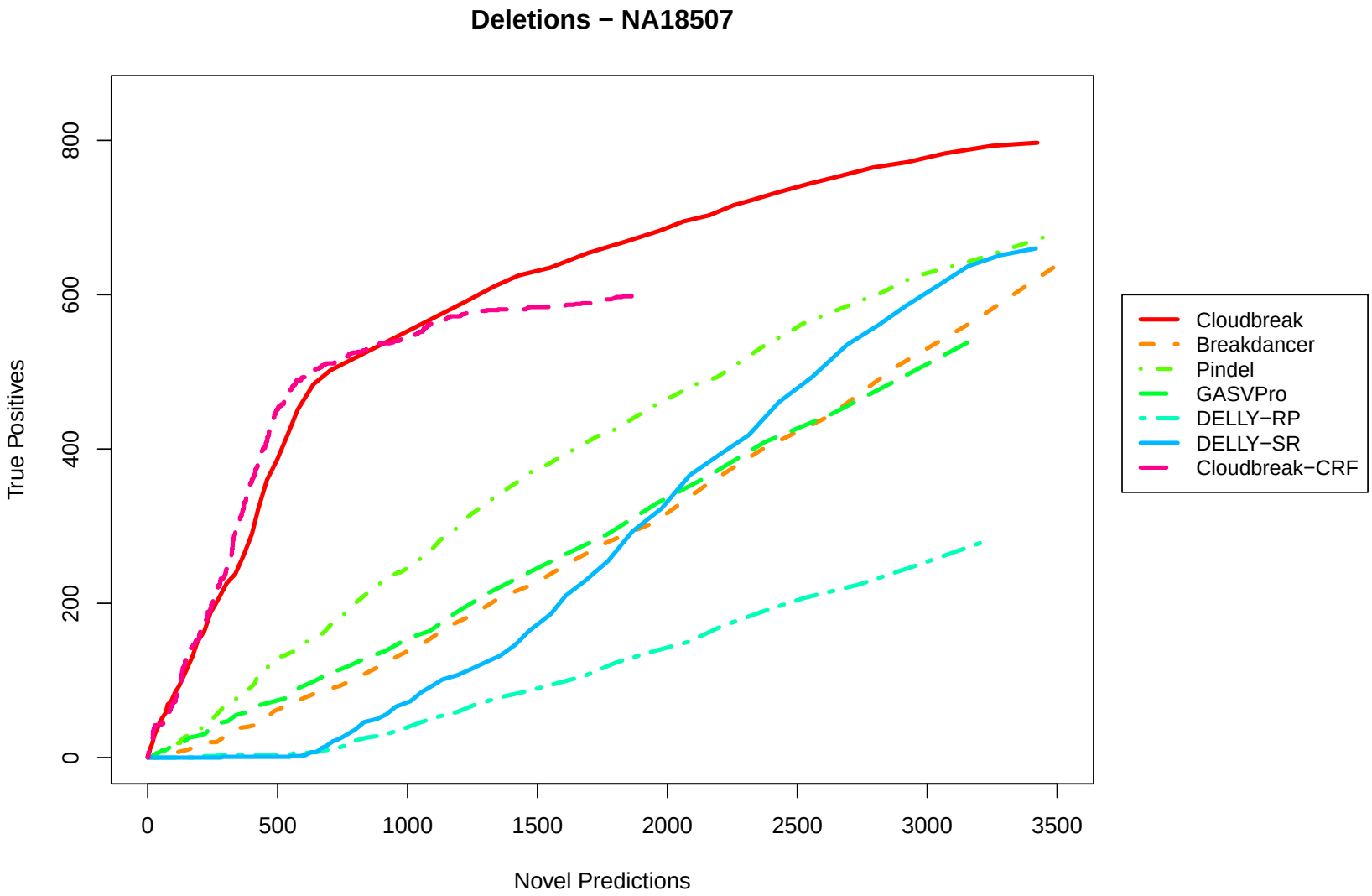
- Label training data



- Train CRF on data from simulated variant genome
  - All insertions and deletions
  - False positive calls from Cloudbreak
- CRF development in Scala using the Factorie library
  - LGBFS optimizer with L_2 regularization
- Use CRF calls to refine Cloudbreak predictions, adjust score
- Test on NA18507 data set with the same gold standard

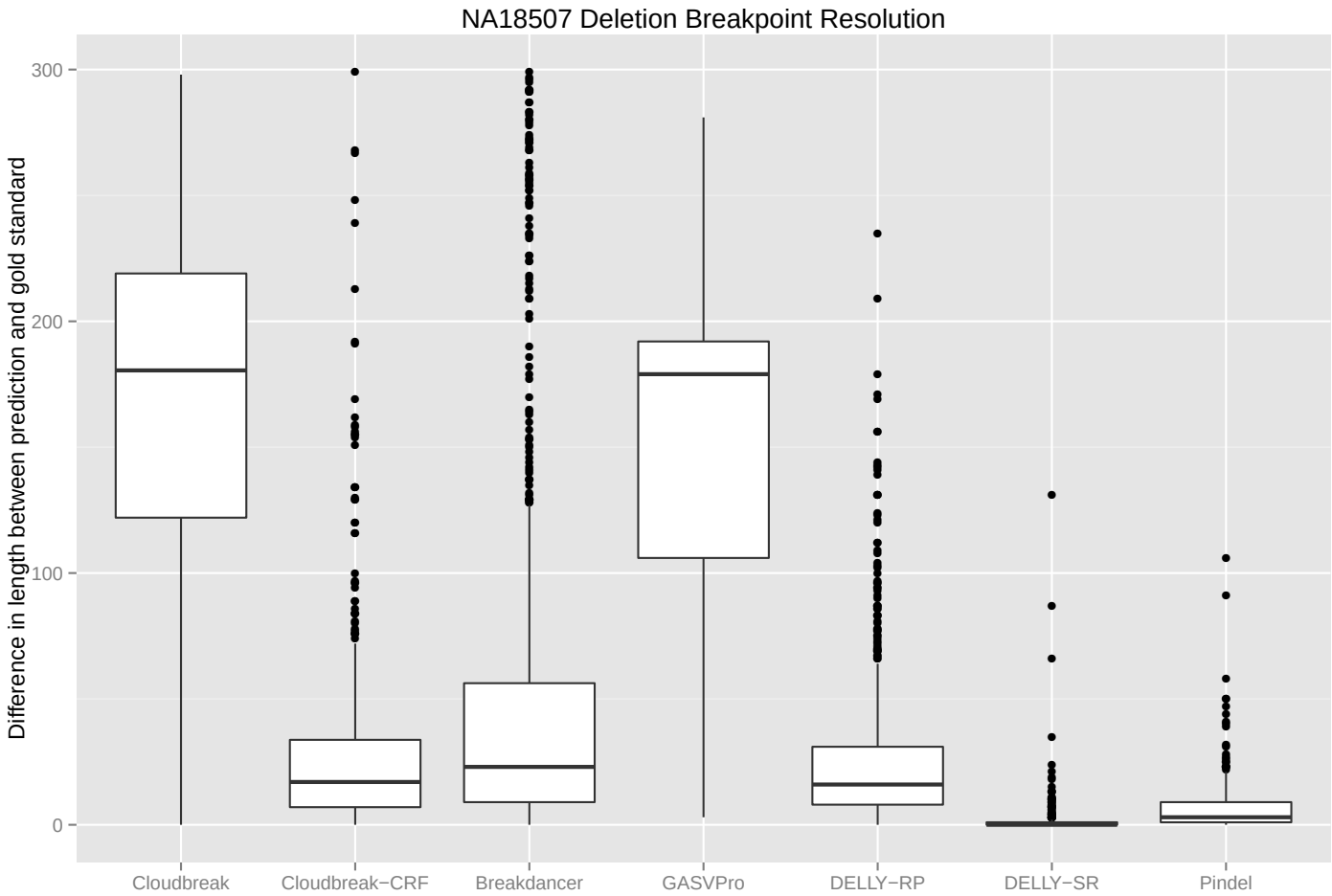
CRF Results

Very modest accuracy improvement



CRF Results

Good improvement in breakpoint resolution



Outline

Summary

- Developed a SV detection algorithm for small deletions and insertions that:
  - Reformulates the problem in terms of local features
  - Has good accuracy
  - Runs very fast on large clusters
  - Has the scalability of Hadoop/MapReduce
- Described a method for integrating multiple signals of structural variation that:
  - Uses discriminative machine learning
  - Improves breakpoint resolution
  - Could be extended to incorporate a wide variety of features

Future work

- Detect other variant types (inversions, MEIs)
- Integrate with other components to create a scalable end-to-end Hadoop genomics pipeline
- Apply ideas from Cloudbreak to population-scale SV callers (Genome STRiP)
- Feature computation algorithms for non diploid or heterogeneous samples (tumors)
- More complete training and testing sets for discriminative machine learning approaches
- Other machine learning techniques - deep learning

What could we do with scalable and accurate SV calling?

- Map how structural variation interacts with coding and regulatory elements across the genome

Science 15 January 2010:
Vol. 327 no. 5963 pp. 302–305
DOI: 10.1126/science.1182213

RESEARCH ARTICLE

Adaptive Evolution of Pelvic Reduction in Sticklebacks by Recurrent Deletion of a *Pitx1* Enhancer

Yingguang Frank Chan[1,*], Melissa E. Marks[1,†], Felicity C. Jones[1], Guadalupe Villarreal Jr.[1,‡], Michael D. Shapiro[1,§], Shannon D. Brady[1], Audrey M. Southwick[2], Devin M. Absher[3], Jane Grimwood[3], Jeremy Schmutz[3], Richard M. Myers[3], Dmitri Petrov[4], Bjarni Jónsson[5], Dolph Schluter[6], Michael A. Bell[7], David M. Kingsley[1,‖]

- Improve understanding of how genomic and epigenetic features interact with structural variants
  - Insights from gibbon genome?
- Enable incorporation of structural variants into clinical genomic pipelines

Acknowledgments

Kemal Sönmez (co-advisor), Lucia Carbone (co-advisor), Izhak Shafran

Carbone Lab: Josh Meyer, Larry Wilhelm, Nathan Lazar, Liz Terhune, Kim Nevonen, Eisa Mahyari

OHSU Center for Spoken Language Understanding

Ability to find simulated variants at maximum sensitivity

- Number of variants found in each size class (number of exclusive predictions for algorithm in that class)

		Prec.	Recall	40-100bp	101-250bp	251-500bp	501-1000bp	> 1000bp
	Total Number			224	84	82	31	26
Deletions	Cloudbreak	0.638	**0.678**	**153** (9)	61 (0)	62 (0)	12 (0)	15 (0)
	BreakDancer	0.356	0.49	89 (0)	54 (0)	53 (0)	8 (0)	15 (0)
	GASVPro	0.146	0.432	83 (2)	32 (0)	55 (0)	8 (0)	15 (0)
	DELLY-RP	0.457	0.613	114 (3)	**68** (0)	**66** (0)	9 (1)	17 (0)
	DELLY-SR	**0.679**	0.166	0 (0)	3 (0)	49 (0)	6 (0)	16 (0)
	Pindel	0.462	0.421	96 (**11**)	24 (0)	48 (0)	5 (0)	15 (0)
	MoDIL	0.132	0.66	123 (6)	66 (**3**)	**66** (**11**)	**17** (**7**)	**23** (**8**)
Insertions	Total Number			199	83	79	21	21
	Cloudbreak	0.451	**0.305**	**79** (**32**)	**32** (**18**)	**11** (8)	1 (0)	0 (0)
	BreakDancer	0.262	0.0968	23 (5)	14 (5)	2 (1)	0 (0)	0 (0)
	Pindel	**0.572**	0.196	52 (25)	5 (1)	10 (**9**)	**3** (**2**)	**9** (**9**)
	MoDIL	0.186	0.0521	14 (1)	4 (0)	1 (0)	2 (**2**)	0 (0)

Ability to find NA18507 deletions by size

- Using the same cutoffs that yielded a 10% FDR on the simulated chromosome 2 data set, adjusted for the difference in coverage from 30X to 37X.

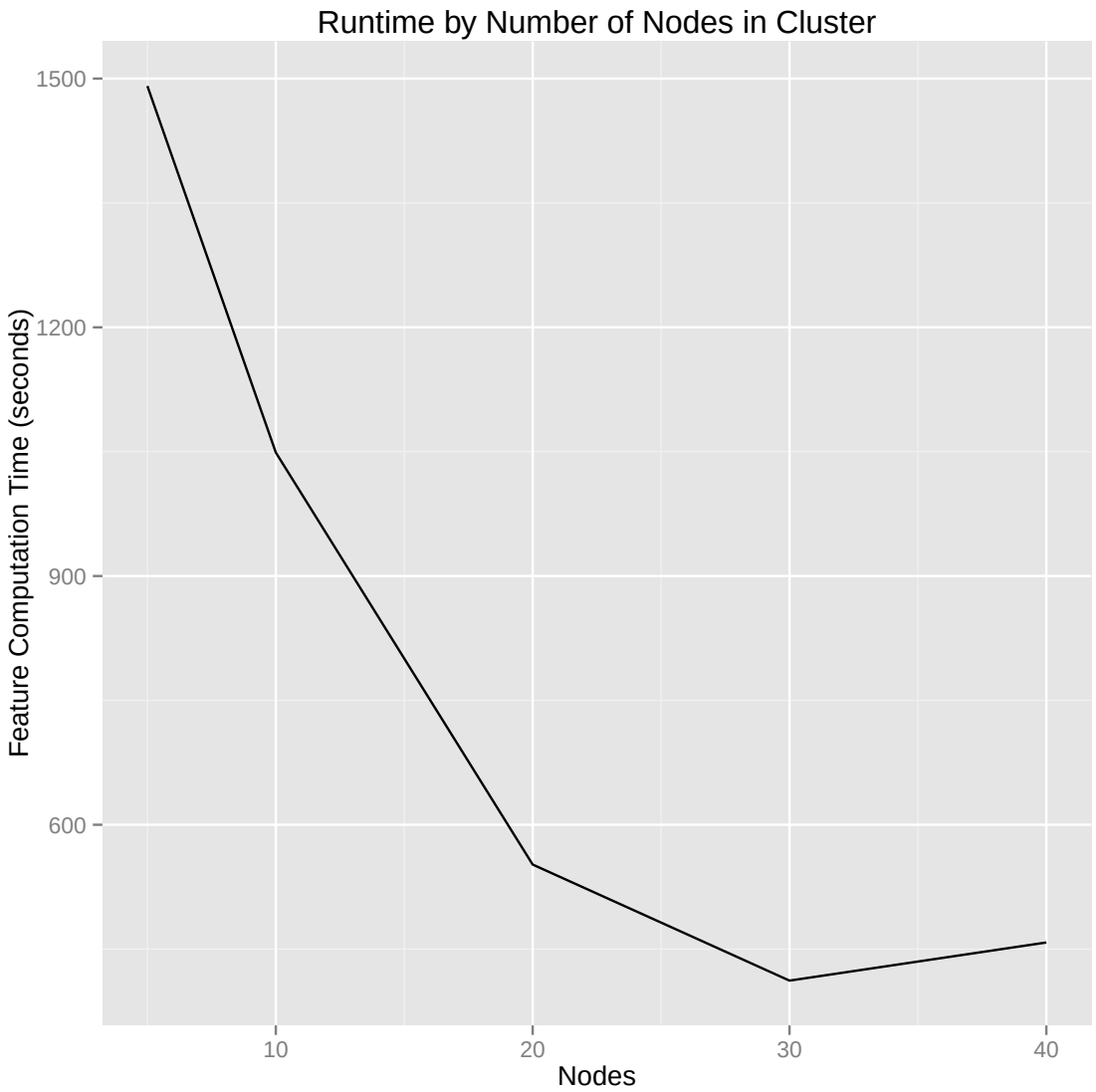
		Prec.	Recall	40-100bp	101-250bp	251-500bp	501-1000bp	> 1000bp
	Total Number			7,462	240	232	147	540
Deletions	Cloudbreak	0.0943	**0.17**	**573 (277)**	**176 (30)**	**197 (18)**	**121 (6)**	**399 (24)**
	BreakDancer	0.137	0.123	261 (29)	136 (3)	178 (0)	114 (0)	371 (0)
	GASVPro	0.147	0.0474	120 (21)	40 (2)	85 (0)	36 (0)	128 (0)
	DELLY-RP	0.0931	0.1	143 (6)	128 (3)	167 (1)	103 (0)	323 (1)
	DELLY-SR	0.153	0.0485	0 (0)	26 (0)	123 (0)	66 (0)	203 (0)
	Pindel	**0.179**	0.0748	149 (8)	61 (0)	149 (0)	69 (1)	217 (0)
Insertions	Total Number			536	114	45	1	0
	Cloudbreak	0.0323	**0.455**	**265 (104)**	**49 (24)**	3 (1)	0 (0)	0 (0)
	BreakDancer	0.0281	0.181	97 (10)	27 (5)	2 (1)	0 (0)	0 (0)
	Pindel	**0.0387**	0.239	144 (45)	14 (7)	**7 (6)**	**1 (1)**	0 (0)

Genotyping Deletions

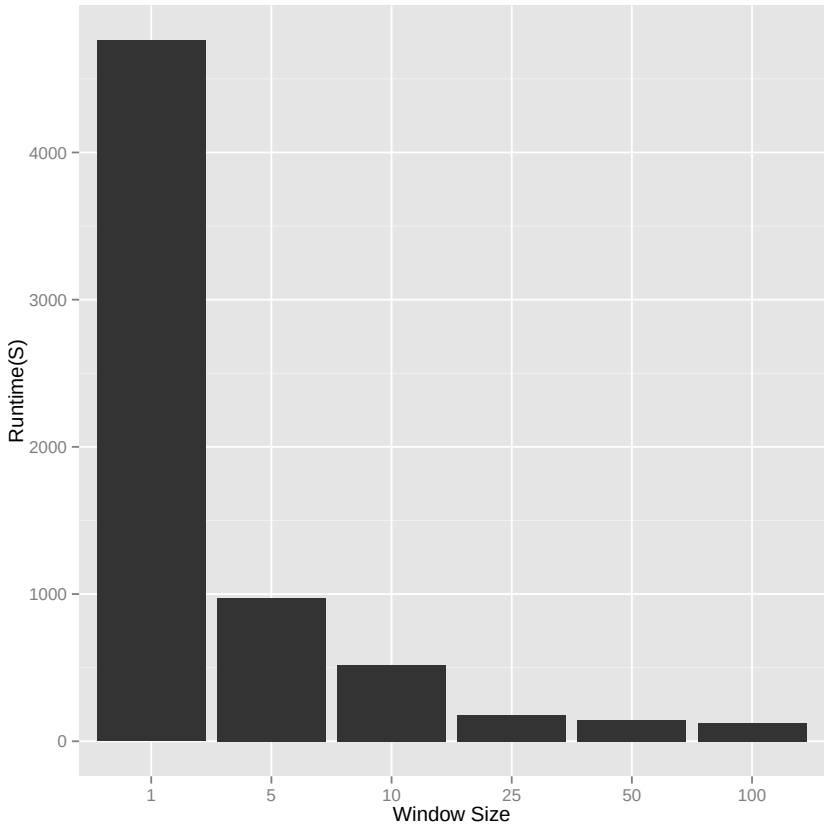
- Use mixing parameter α to predict genotypes.
- Threshold the average value of α in each prediction
  - 92.7% accuracy on simulated data
  - 95.9% accuracy on NA18507 (for calls with genotypes from Genome STRiP)

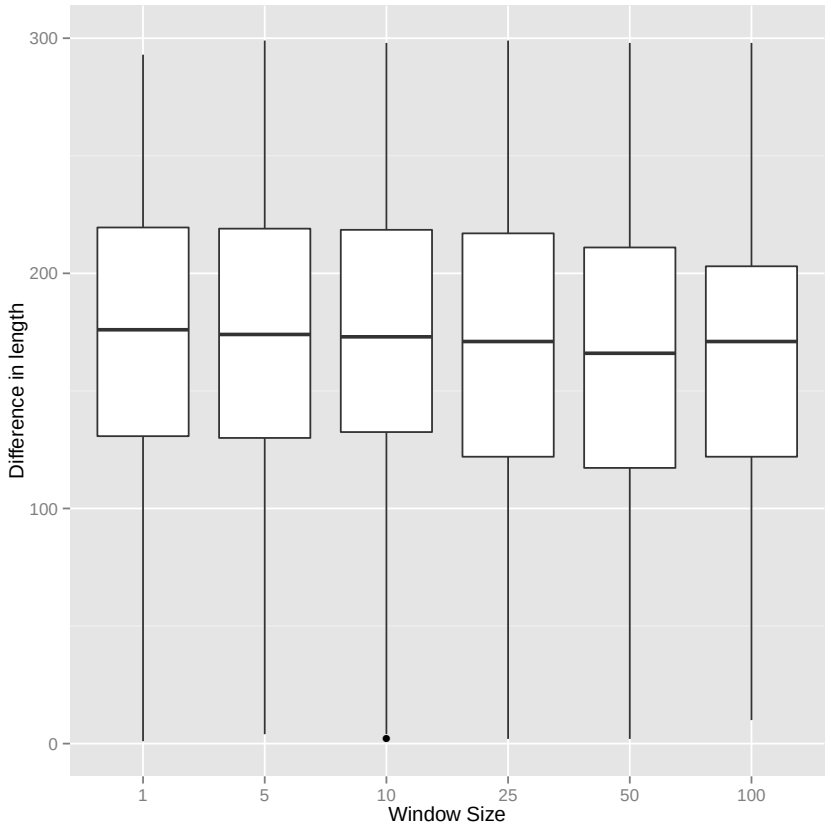
		Actual Genotypes			
		Simulated Data		NA18507	
		Homozygous	Heterozygous	Homozygous	Heterozygous
Predicted Genotypes	Homozygous	35	2	96	21
	Heterozygous	0	39	2	448

Scalability



Effect of Window Size Choice





Window Size	Calls	True Positives	Precision	Recall	F1
1	274	228	0.832	0.57	0.677
5	268	228	0.851	0.57	0.683
10	258	223	0.864	0.557	0.678
25	240	217	0.904	0.542	0.678
50	215	194	0.902	0.485	0.631
100	162	141	0.87	0.352	0.502