

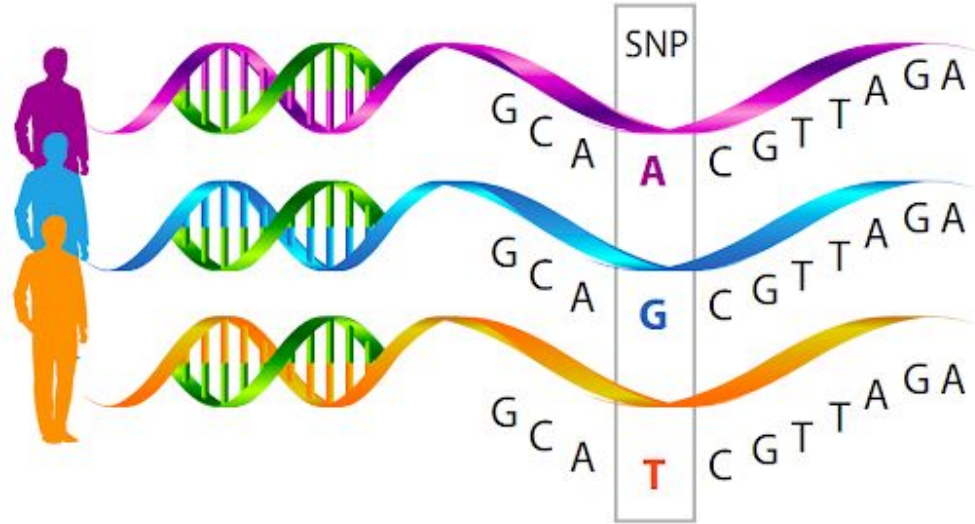
Genomic Sequencing Analysis Parallelization

Kar-Tong Tan
Nripsuta Saxena
Divyam Misra
Andrew Lund



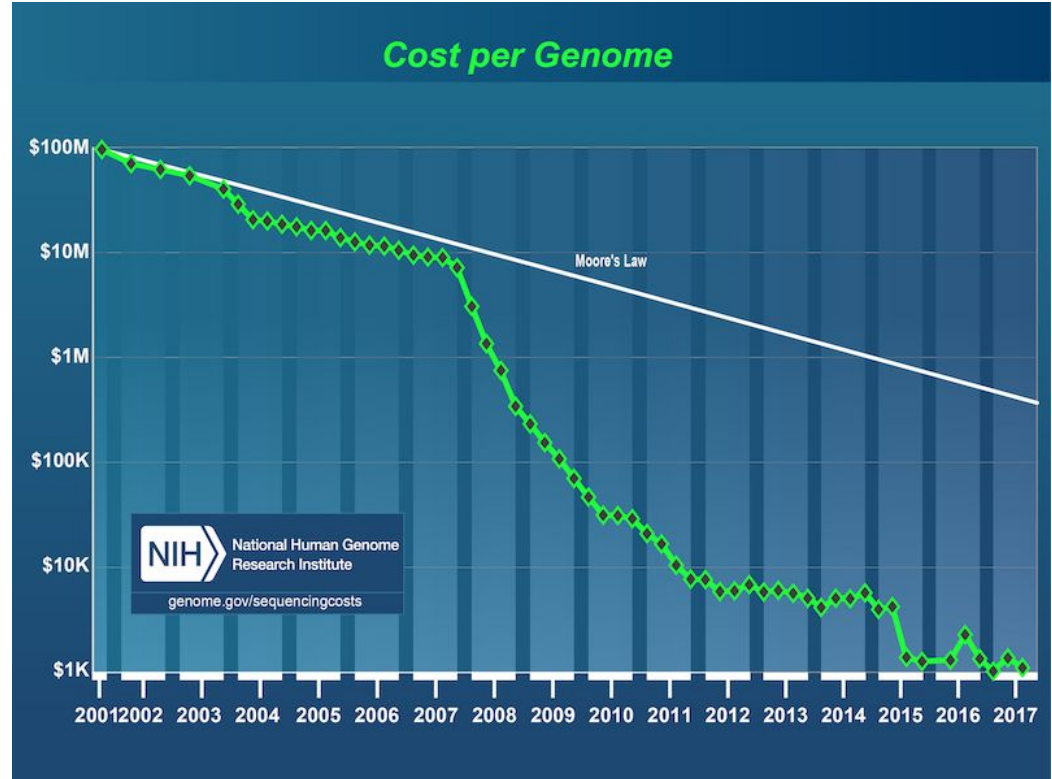
Introduction

- What is genomic sequencing?
 - SNPs
- Why is it important?
 - Disease diagnosis
 - Predictive testing
 - Pharmacogenetic testing



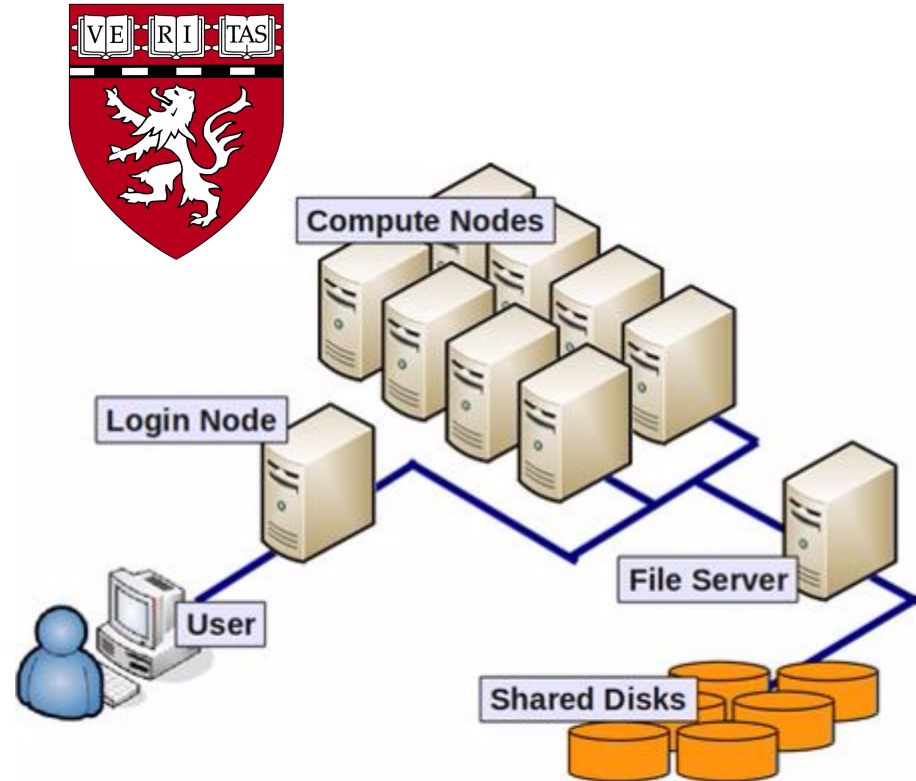
What is the problem?

1. Need for Parallelization
2. Need for **Efficient** Parallelization



Harvard Medical School Research Computing

- 8,000 cores
 - 32 or 28 cores per node (capped at 20)
 - 256 GB RAM per node
- SLURM batch system
- Supports OpenMP, MPI
- Currently does not support Spark



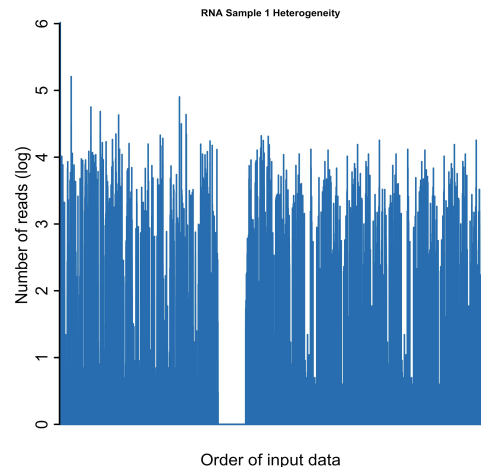
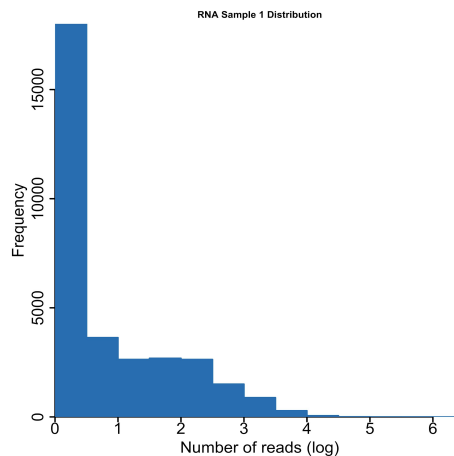
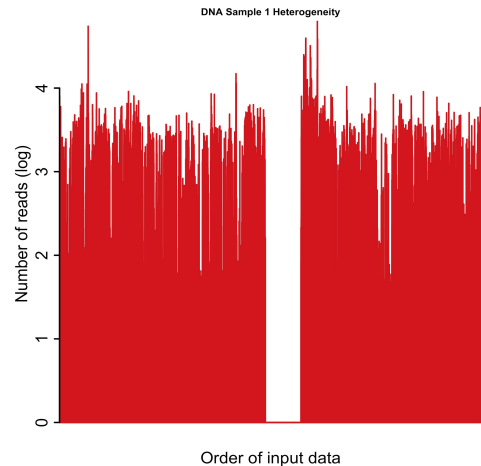
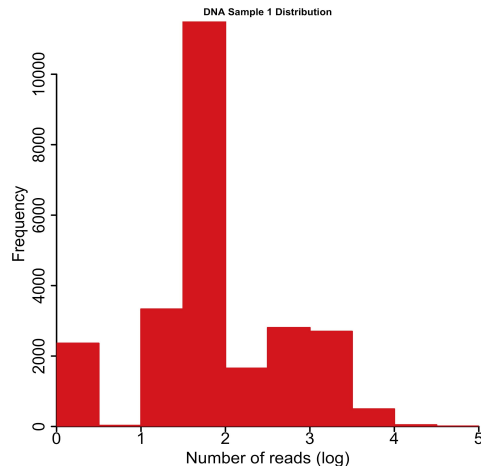
Model & Data

SAMtools SNP Analysis Suite:

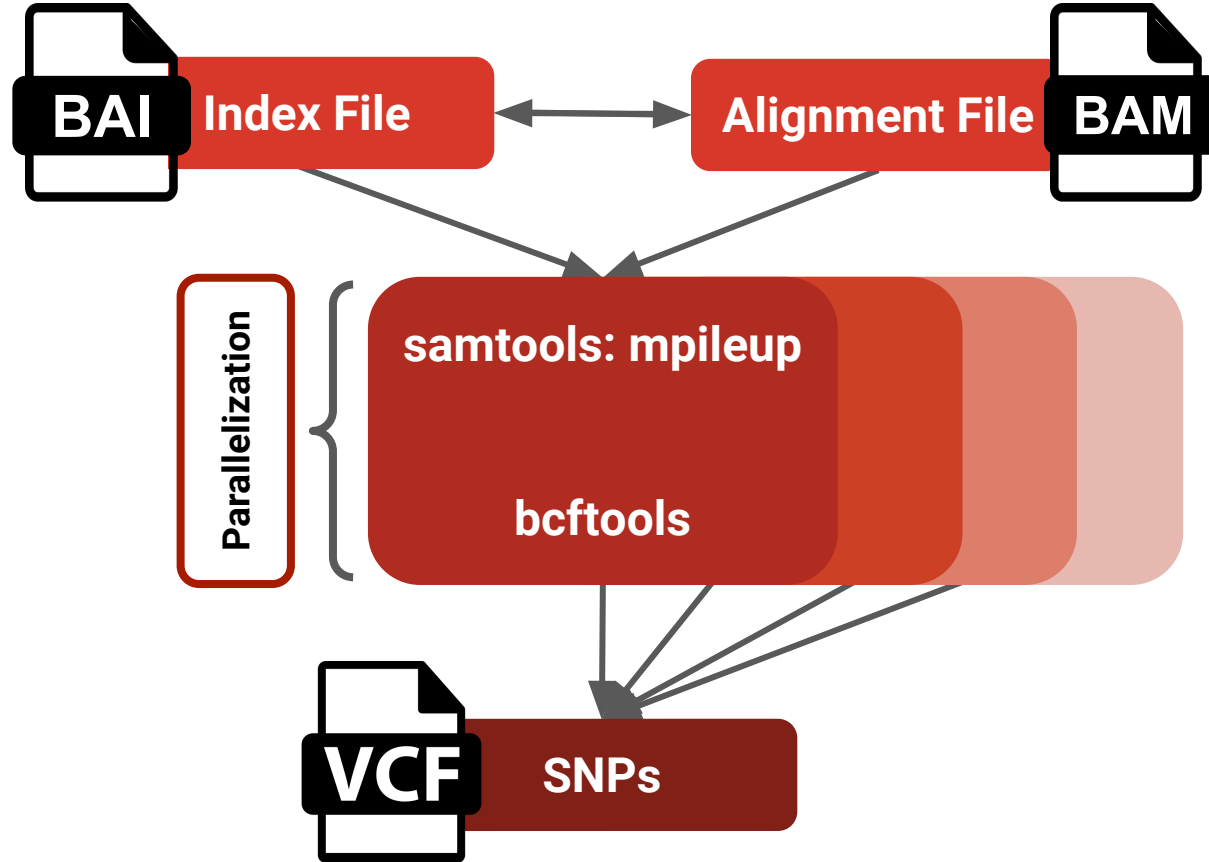
- 93,000 lines (mostly C)
- SAMtools, BCFtools, HTSlib

Genome Data:

- 2 people: DNA & RNA
- Alignment files (.bam) is ~10GB
- Index files (.bai) is ~5MB
- Very heterogeneous



SAMtools Parallelization Strategy



Profiling

Each sample counts as 0.01 seconds.

| % time | cumulative seconds | | self calls | self s/call | total s/call | name |
|--------------|-----------------------|-------------|---------------|----------------|-----------------|-----------------------|
| 51.10 | 1.16 | 1.16 | 1 | 1.16 | 2.27 | mpileup |
| 18.06 | 1.57 | 0.41 | 4519010 | 0.00 | 0.00 | bam_plp_next |
| 12.11 | 1.85 | 0.28 | 26195998 | 0.00 | 0.00 | pileup_seq |
| 9.69 | 2.07 | 0.22 | 26896170 | 0.00 | 0.00 | resolve_cigar2 |
| 5.73 | 2.20 | 0.13 | 4213250 | 0.00 | 0.00 | bam_mplp_auto |
| 0.44 | 2.21 | 0.01 | 353719 | 0.00 | 0.00 | kh_get_olap_hash |
| 0.44 | 2.22 | 0.01 | 330015 | 0.00 | 0.00 | bam_read1 |
| 0.44 | 2.23 | 0.01 | 305760 | 0.00 | 0.00 | mp_free |
| 0.44 | 2.24 | 0.01 | 305760 | 0.00 | 0.00 | mplp_func |
| 0.44 | 2.25 | 0.01 | 305759 | 0.00 | 0.00 | bam_copy1 |
| 0.44 | 2.26 | 0.01 | 17474 | 0.00 | 0.00 | tweak_overlap_quality |
| 0.44 | 2.27 | 0.01 | 2670 | 0.00 | 0.00 | printw |
| 0.22 | 2.27 | 0.01 | 4213249 | 0.00 | 0.00 | mplp_get_ref |

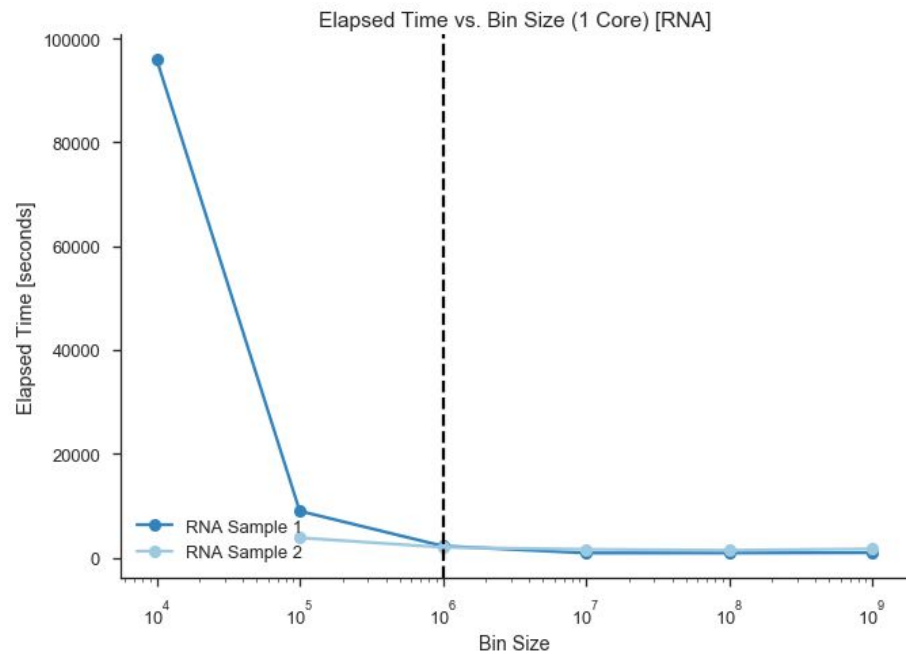
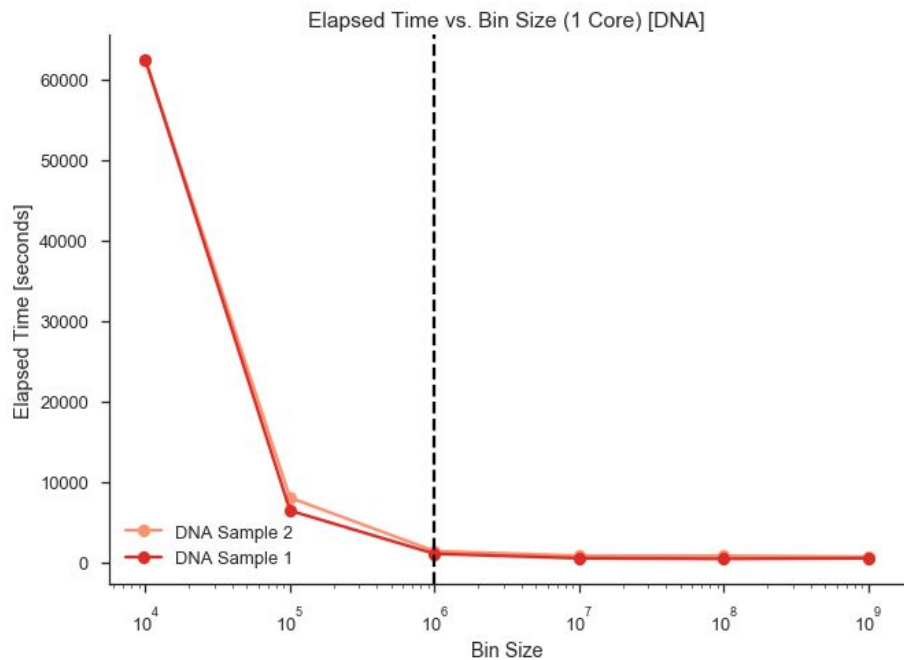
Parallelization Techniques

1. Single-node parallelization (binning)
2. MPI
3. Load balancing simulator
4. OpenMP

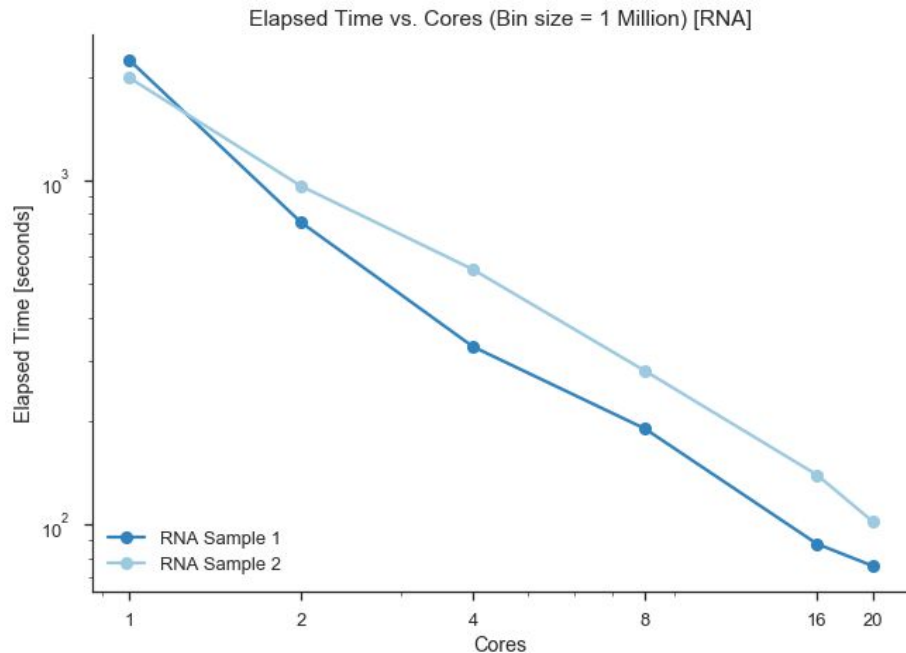
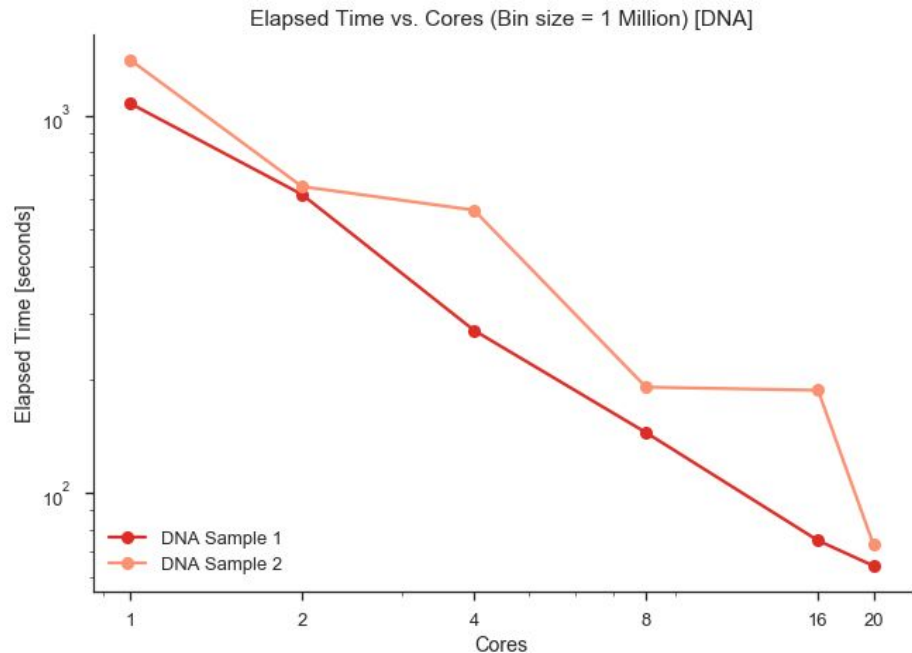
Parallelization Techniques

1. Single-node parallelization (binning)
2. MPI
3. Load balancing simulator
4. OpenMP

Optimal Bin Size

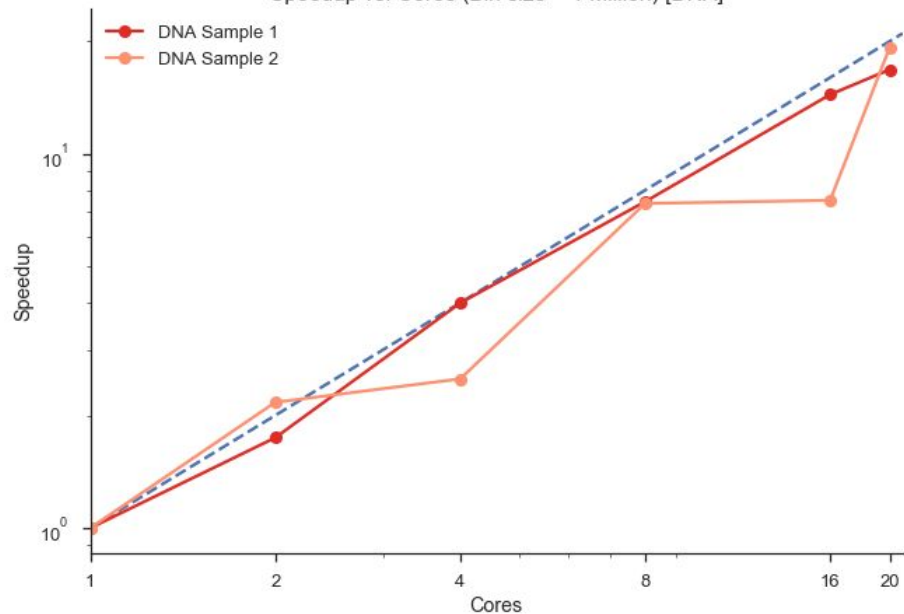


Results: Single-node Parallelization Execution Time

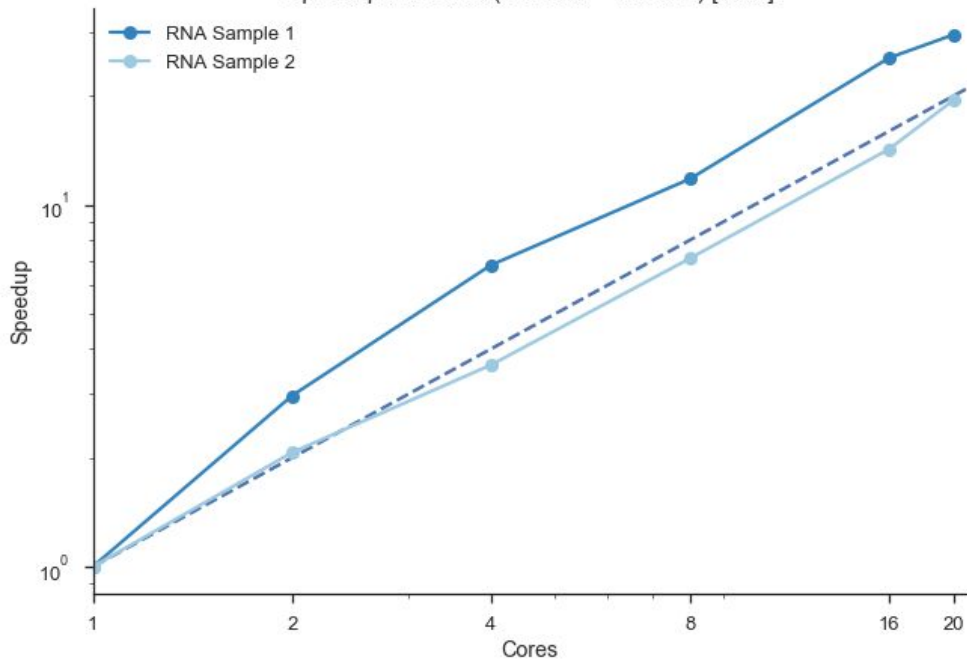


Results: Single-node Parallelization Speedup

Speedup vs. Cores (Bin size = 1 Million) [DNA]



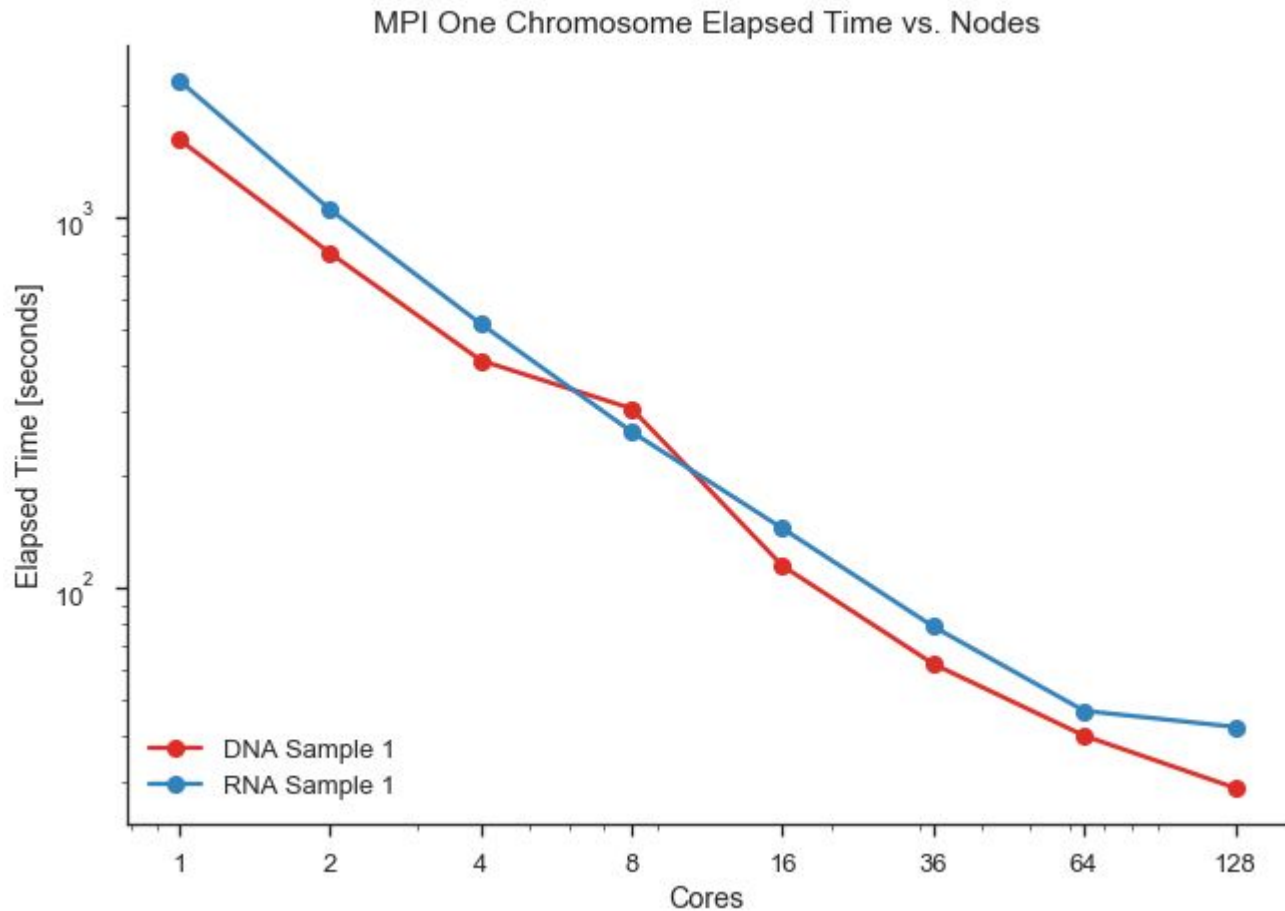
Speedup vs. Cores (Bin size = 1 Million) [RNA]



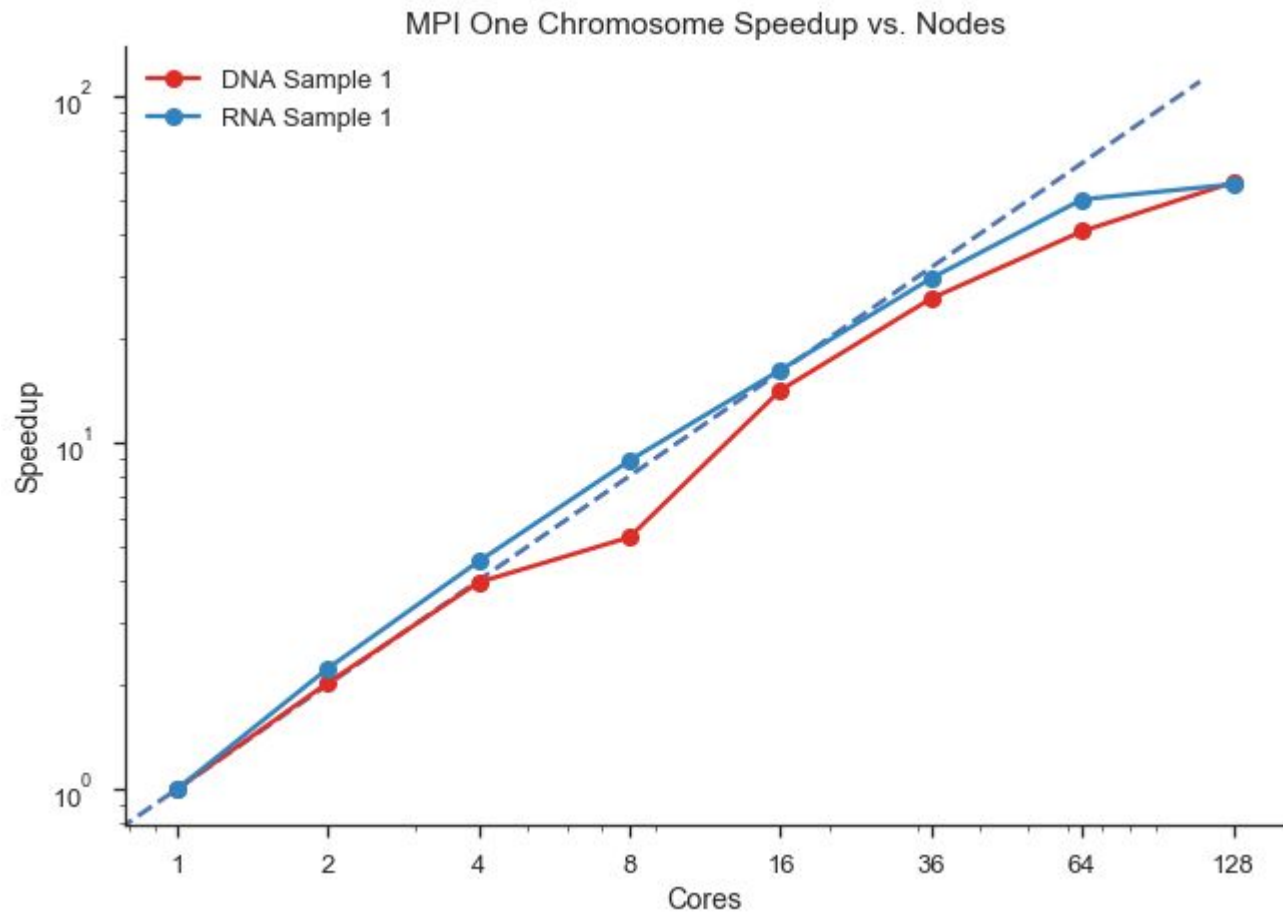
Parallelization Techniques

1. Single-node parallelization (binning)
2. MPI
3. Load balancing simulator
4. OpenMP

Results: MPI Runtime



Results: MPI Speedup



Parallelization Techniques

1. Single-node parallelization (binning)
2. MPI
3. Load balancing simulator
4. OpenMP

Index Files (.bai)

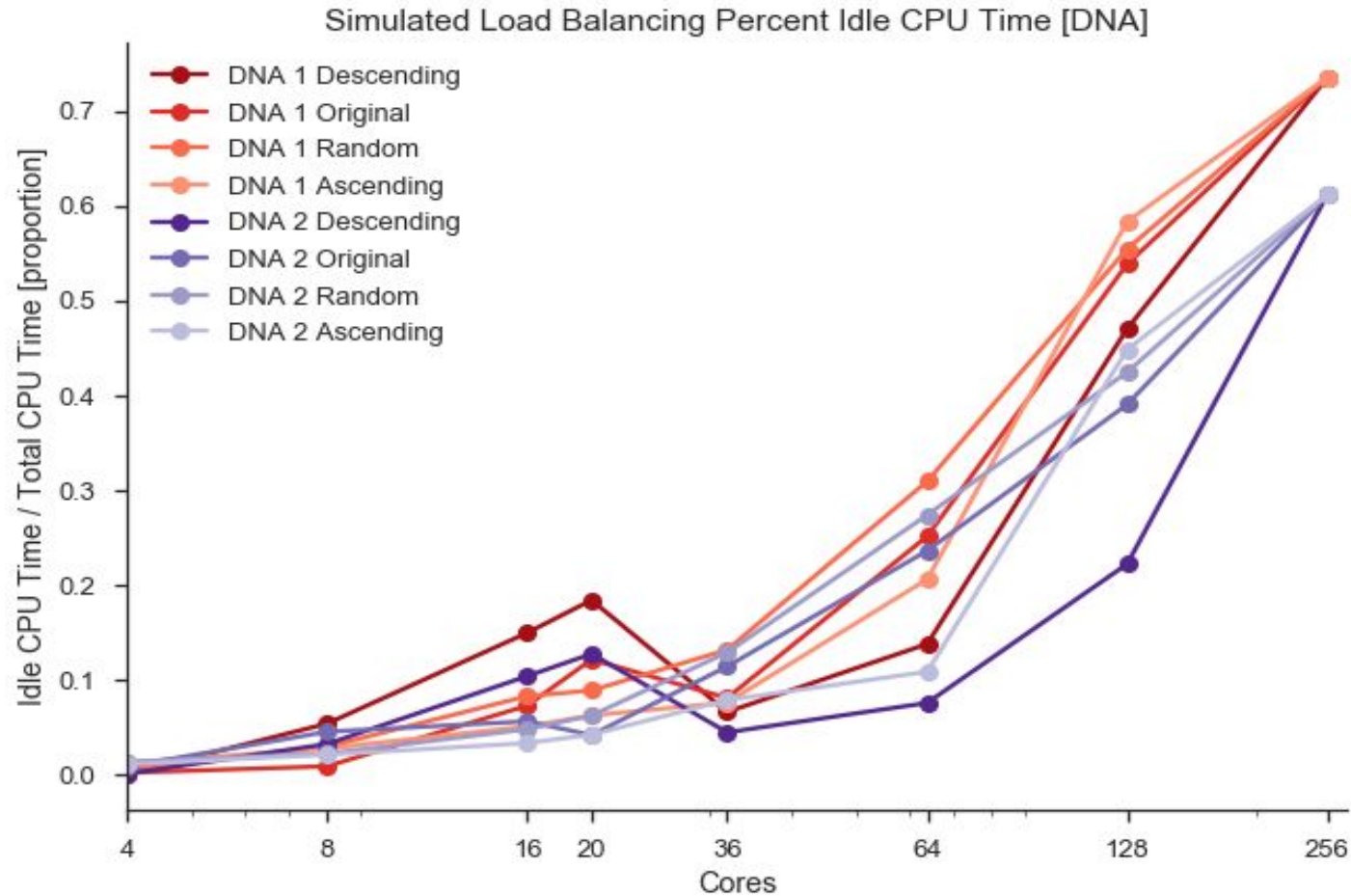
What: “Table of contents” to alignment file

Our Idea: Use to quickly determine data distribution & optimal load balancing strategy

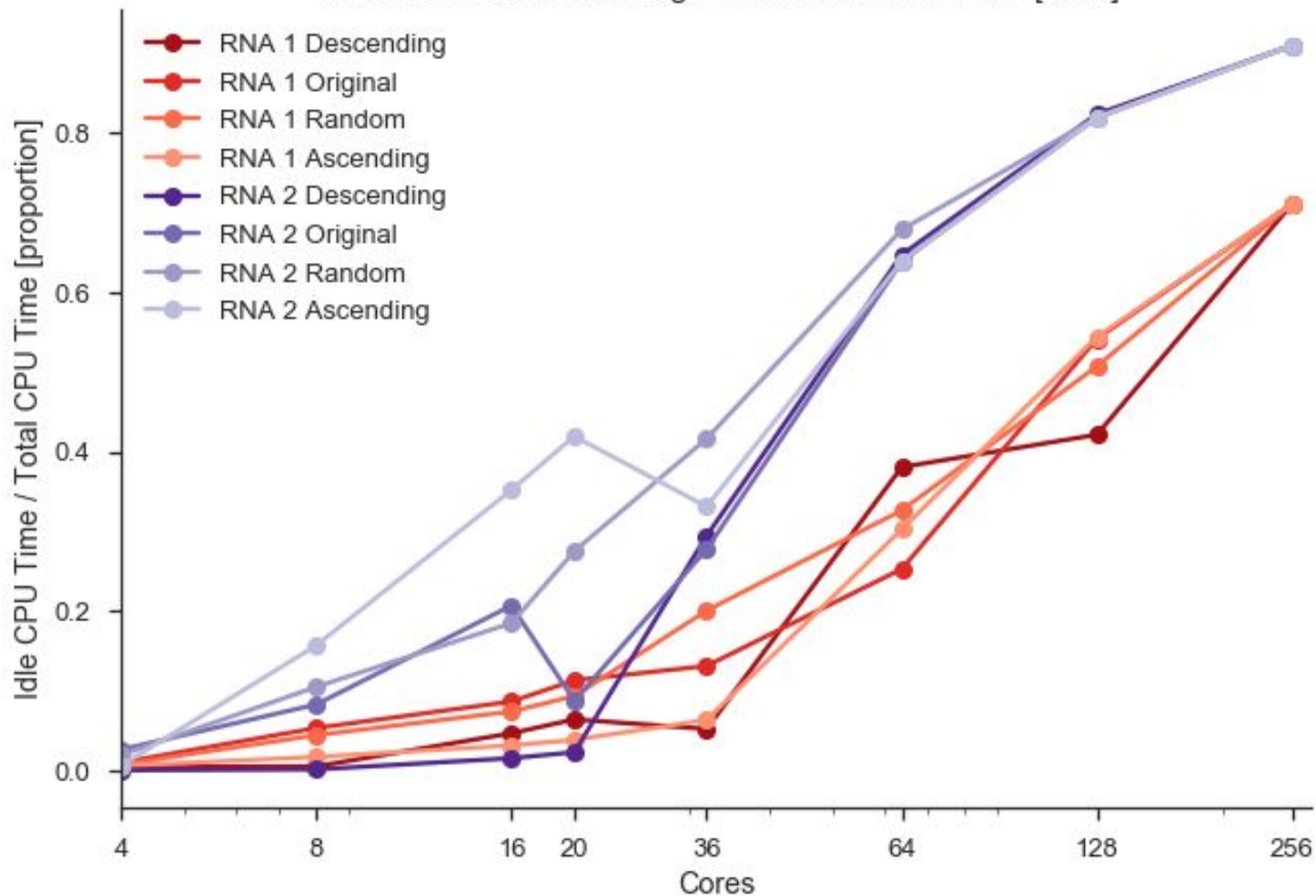
Problem: Binary, hard to read, no domain knowledge

Work-around: Load Balancing Simulator

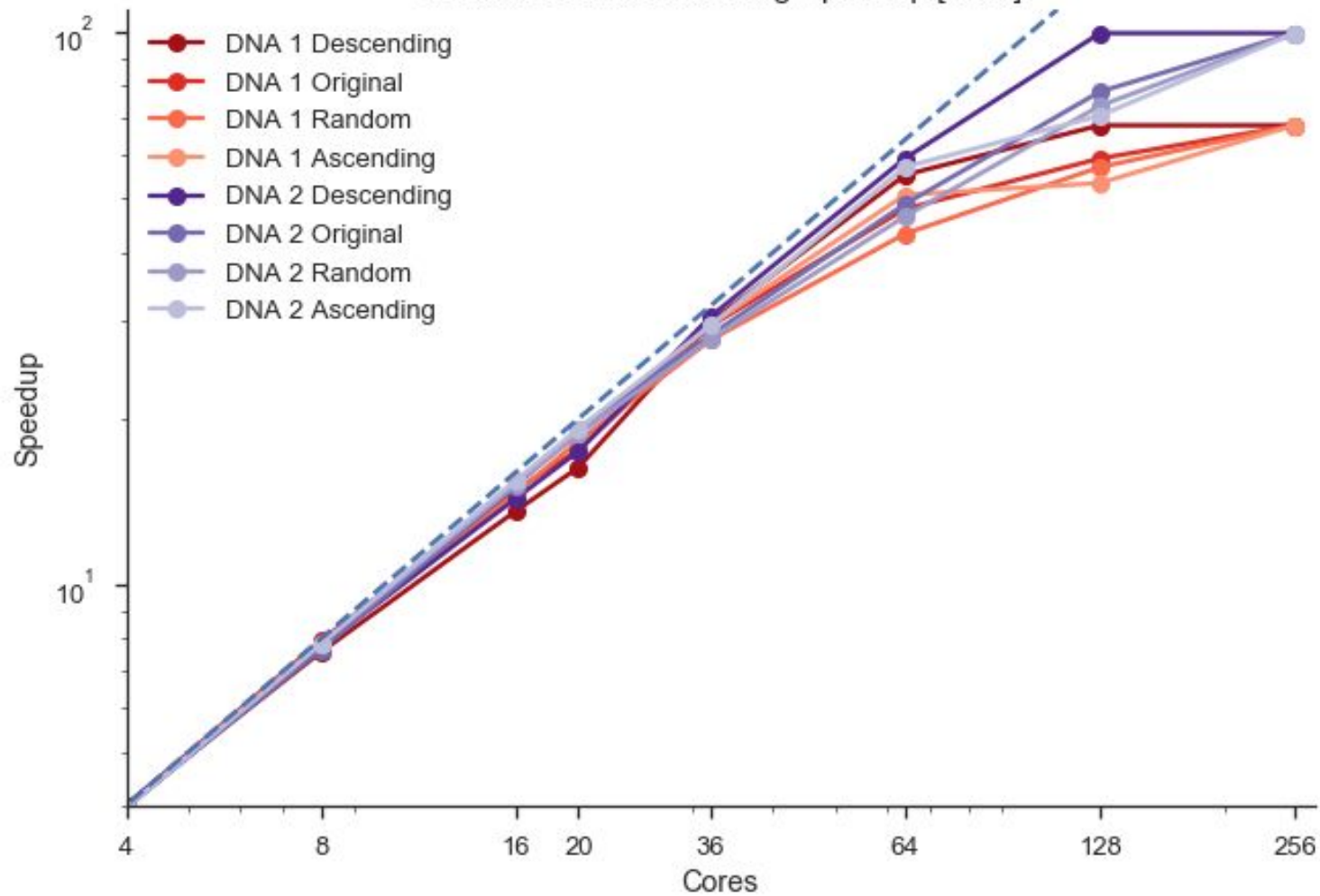
Results: Load Balancing Simulator



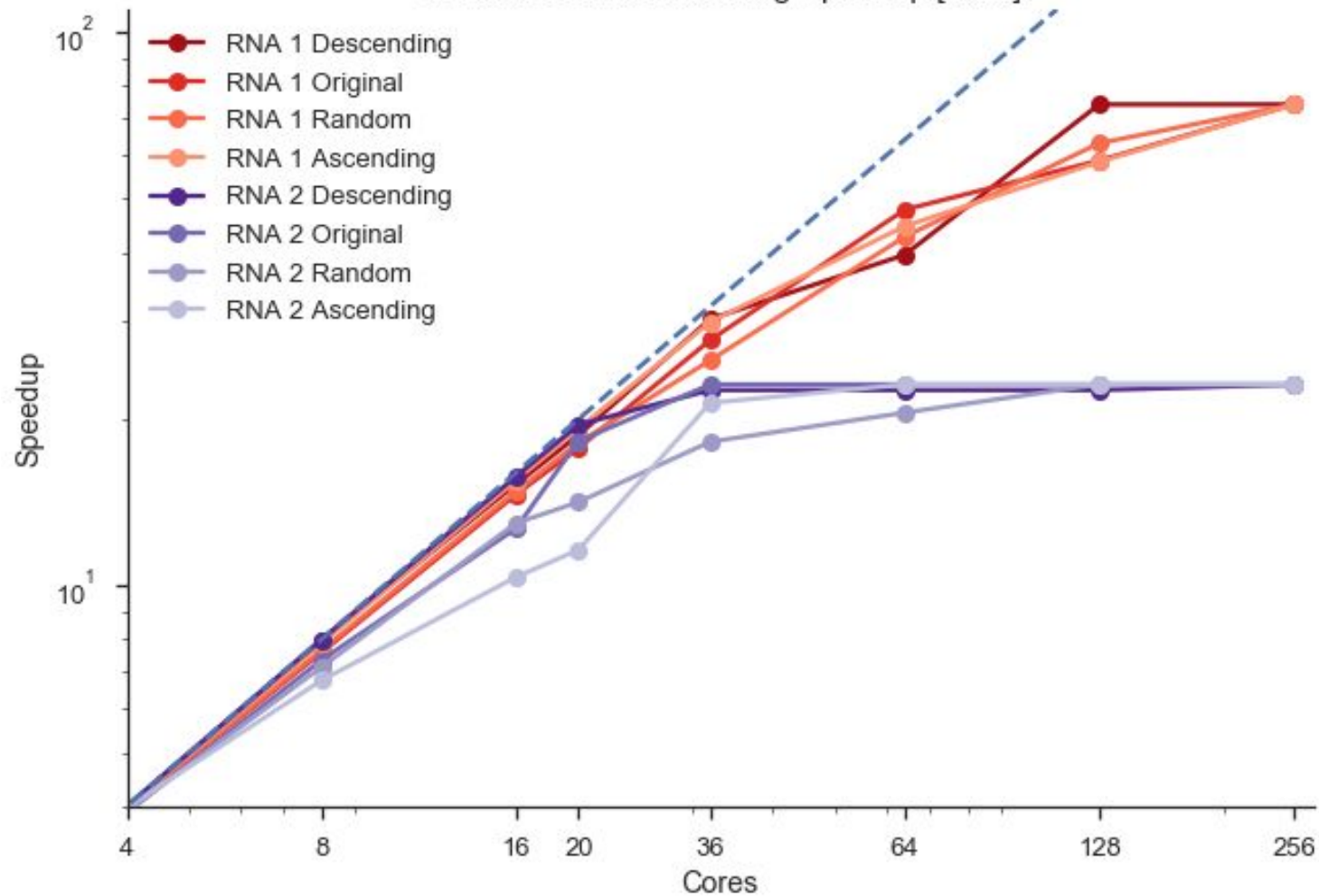
| Cores | RNA 1 Descending | RNA 1 Original | RNA 1 Random | RNA 1 Ascending | RNA 2 Descending | RNA 2 Original | RNA 2 Random | RNA 2 Ascending |
|-------|------------------|----------------|--------------|-----------------|------------------|----------------|--------------|-----------------|
| 4 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 8 | 0.05 | 0.05 | 0.02 | 0.02 | 0.08 | 0.08 | 0.10 | 0.15 |
| 16 | 0.05 | 0.08 | 0.08 | 0.05 | 0.20 | 0.20 | 0.18 | 0.35 |
| 20 | 0.07 | 0.10 | 0.05 | 0.05 | 0.02 | 0.08 | 0.28 | 0.42 |
| 36 | 0.05 | 0.13 | 0.07 | 0.07 | 0.28 | 0.28 | 0.28 | 0.42 |
| 64 | 0.38 | 0.25 | 0.32 | 0.30 | 0.65 | 0.65 | 0.65 | 0.68 |
| 128 | 0.42 | 0.50 | 0.55 | 0.55 | 0.82 | 0.82 | 0.82 | 0.82 |
| 256 | 0.72 | 0.72 | 0.72 | 0.72 | 0.90 | 0.90 | 0.90 | 0.90 |



Simulated Load Balancing Speedup [DNA]



Simulated Load Balancing Speedup [RNA]



Parallelization Techniques

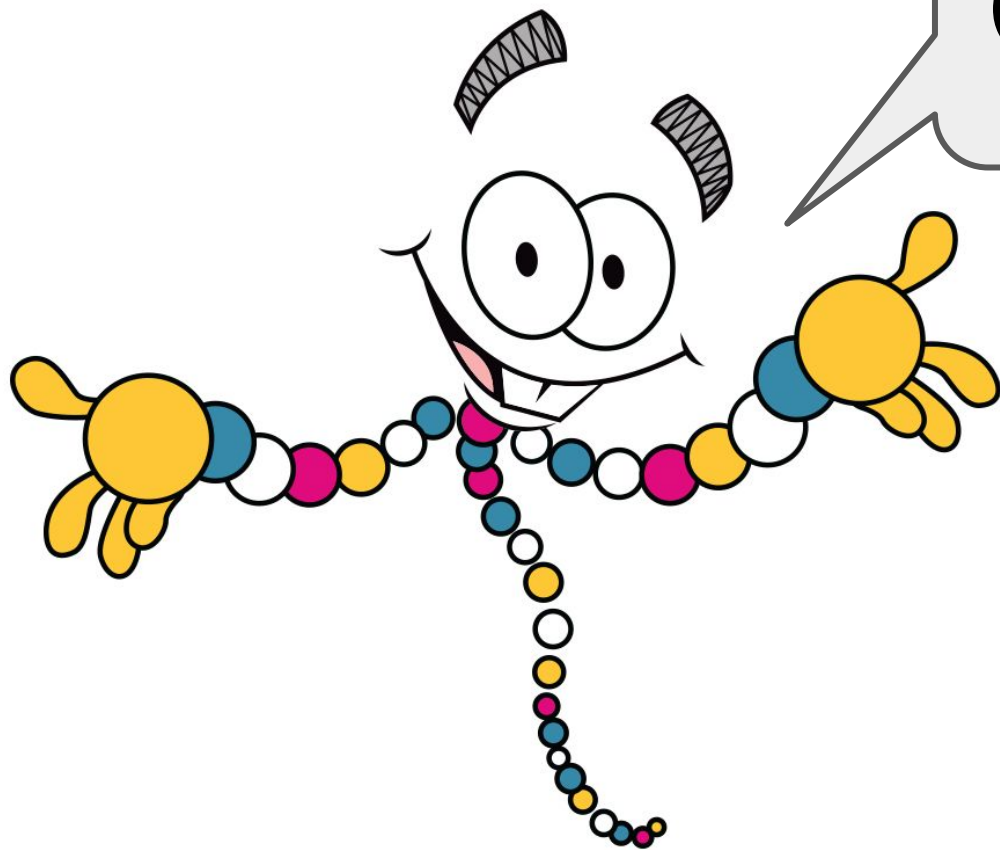
1. Single-node parallelization (binning)
2. MPI
3. Load balancing simulator
4. OpenMP

Results: OpenMP

| Function | 1 thread time (seconds) | 8 threads time (seconds) |
|------------------------|-------------------------|--------------------------|
| no pragmas | 22.3 | --- |
| mpileup (8 pragmas) | 37.5 | 37.4 |
| pileup_seq (2 pragmas) | 103.7 | 50.5 |

Conclusions

- Binning, MPI, and Load Balancing worked great
- OpenMP is highly complex to implement
- Transforming the index file to human-readable text is challenging!
- Load balancing simulator is a great analysis tool
- Will apply these techniques on a ~400TB dataset on a supercomputer



Questions?