Parallel Computing Strategies for NGS Sequence Mapping

Kun Huang

Doruk Bozdag, Terry Camerlengo, Hatice Gulcin Ozer, Joanne Trgovcich, Tea Meulia, Umit Catalyurek

Department of Biomedical Informatics
OSUCCC Biomedical Informatics Shared Resource

Outline

- Introduction
- Different Approaches
- Parallelization Strategies
- Summary

Mapping Sequences to Reference Genome

- Solexa and SOLiD sequencers
  - ~ 35-70 bp DNA segments
- Mapping Methods
  - Need to allow mismatches and gaps
    - SNP locations
    - Sequencing errors
    - Reading errors
  - Indexing and hashing
    - genome
    - sequence reads
- Use of quality scores
- Performance
  - Partitioning the genome or sequence reads

Data Preprocessing

Mapping Sequences to Reference Genome

- ELAND (Cox, unpublished)
  - "Efficient Large-Scale Alignment of Nucleotide Databases" (Solexa Ltd.)
  - Very fast
  - Allow at most 2 mismatches
- SeqMap (Jiang, 2008)
  - "Mapping massive amount of oligonucleotides to the genome"
  - Allow at most 5 mismatches and gaps
- RMAP (Smith, 2008)
  - "Using quality scores and longer reads improves accuracy of Solexa read mapping"
- MapReads (Applied Biosystems)

Mapping Procedure

- Two steps:
  - Data or genome transform
    - Hashing table
    - Borrows-Wheeler transform
  - Mapping
    - Table lookup or index search

Tradeoffs

- Limiting the number of allowed mismatches
- Ignoring insertions and deletions or limiting their number and length
- Ignoring base quality score information
- Limiting the number of reported matching locations
- Imposing constraints on read length
- Ignoring information about errors particular to each sequencing technology

Outline

- Introduction
- Different Approaches
- Parallelization Strategies
- Summary

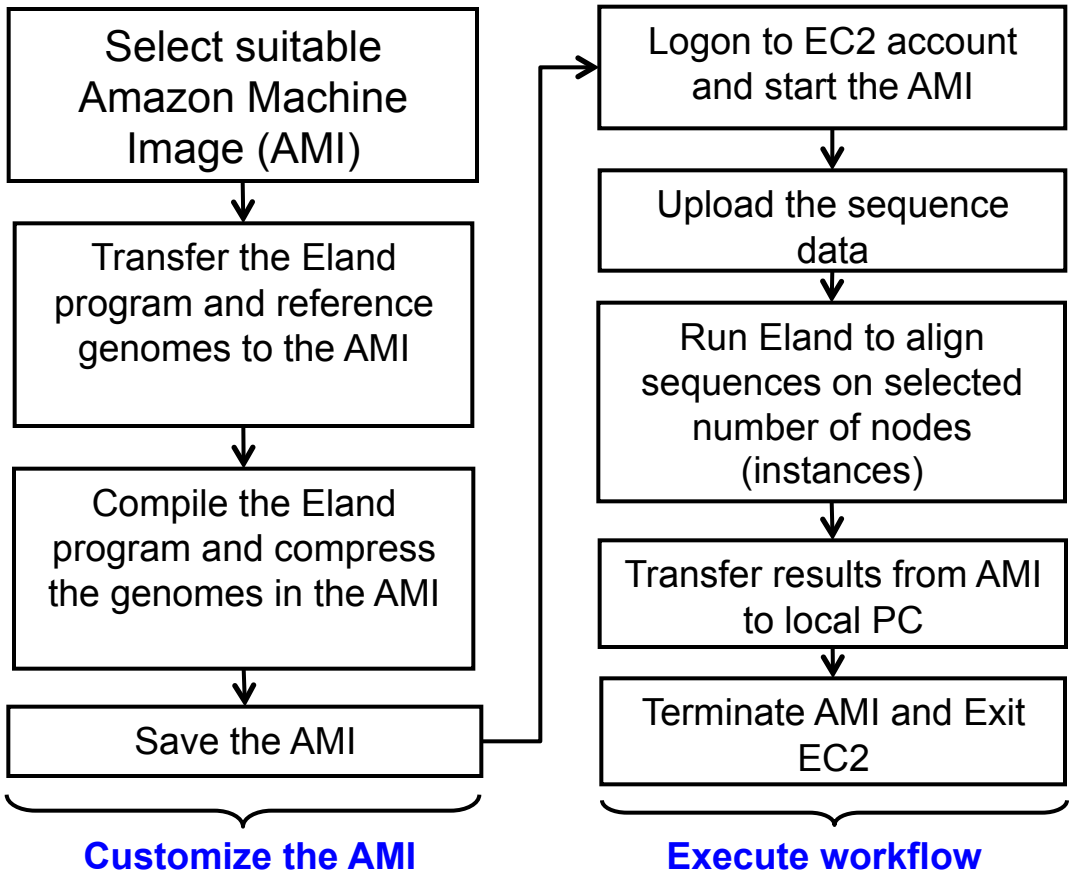
Different Approaches

- Multi-threading
  - Most on the second step (e.g., Bowtie, SOCS)
  - Or can be for reading and writing (e.g., a GMAP implementation)
  - Relatively easy
  - Not scalable

Different Approaches

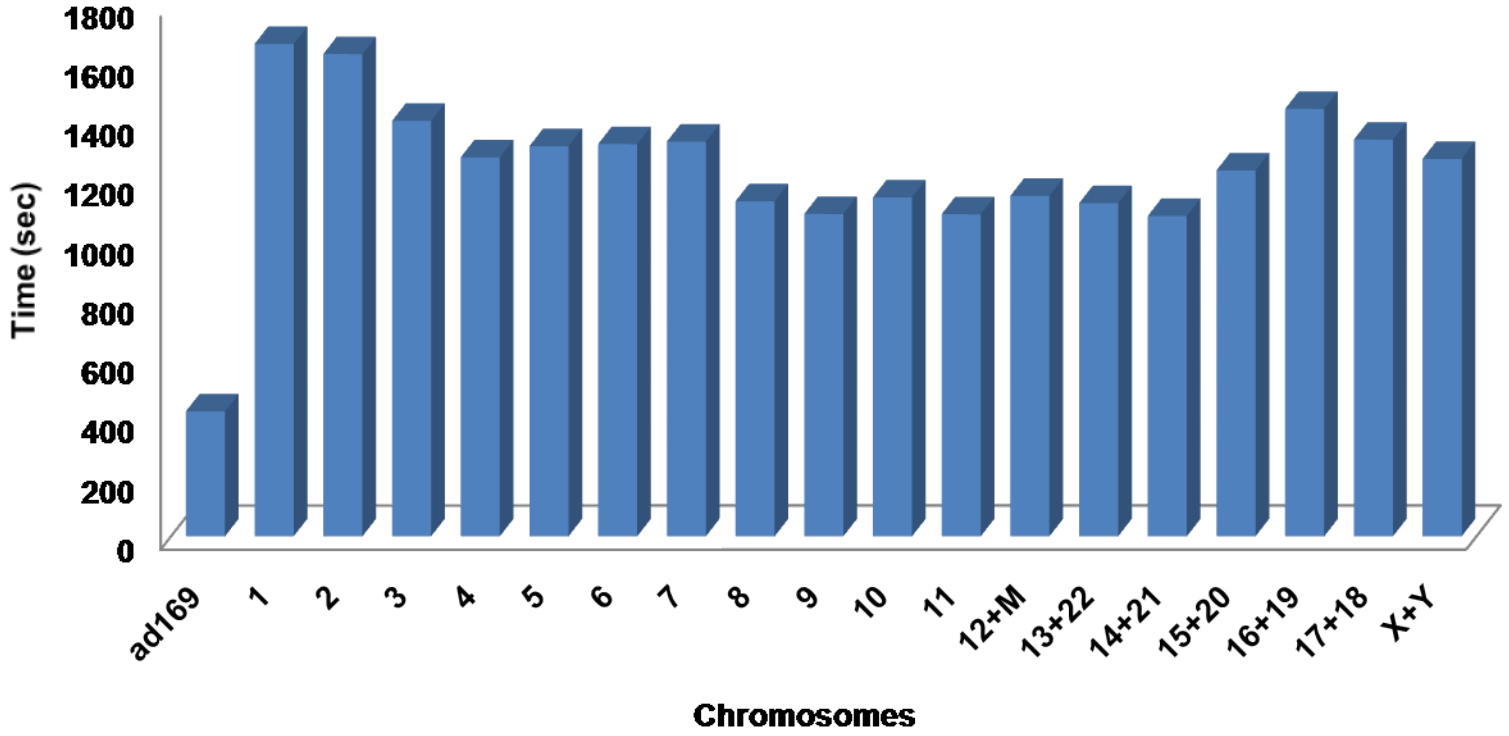
- Cloud computing
  - CloudBurst
    - MapReduce for RMAP
  - Amazon EC2

Experiment Using Amazon EC2



Experiment Using Amazon EC2

- Cloud computing – Amazon Elastic Computing Cloud (EC2)
  - Low-cost : pay per use
  - Easy to maintain and set up
  - Mapping 7.8 million short reads to the human genome in less than 0.5 hour for less than $4



Different Approaches

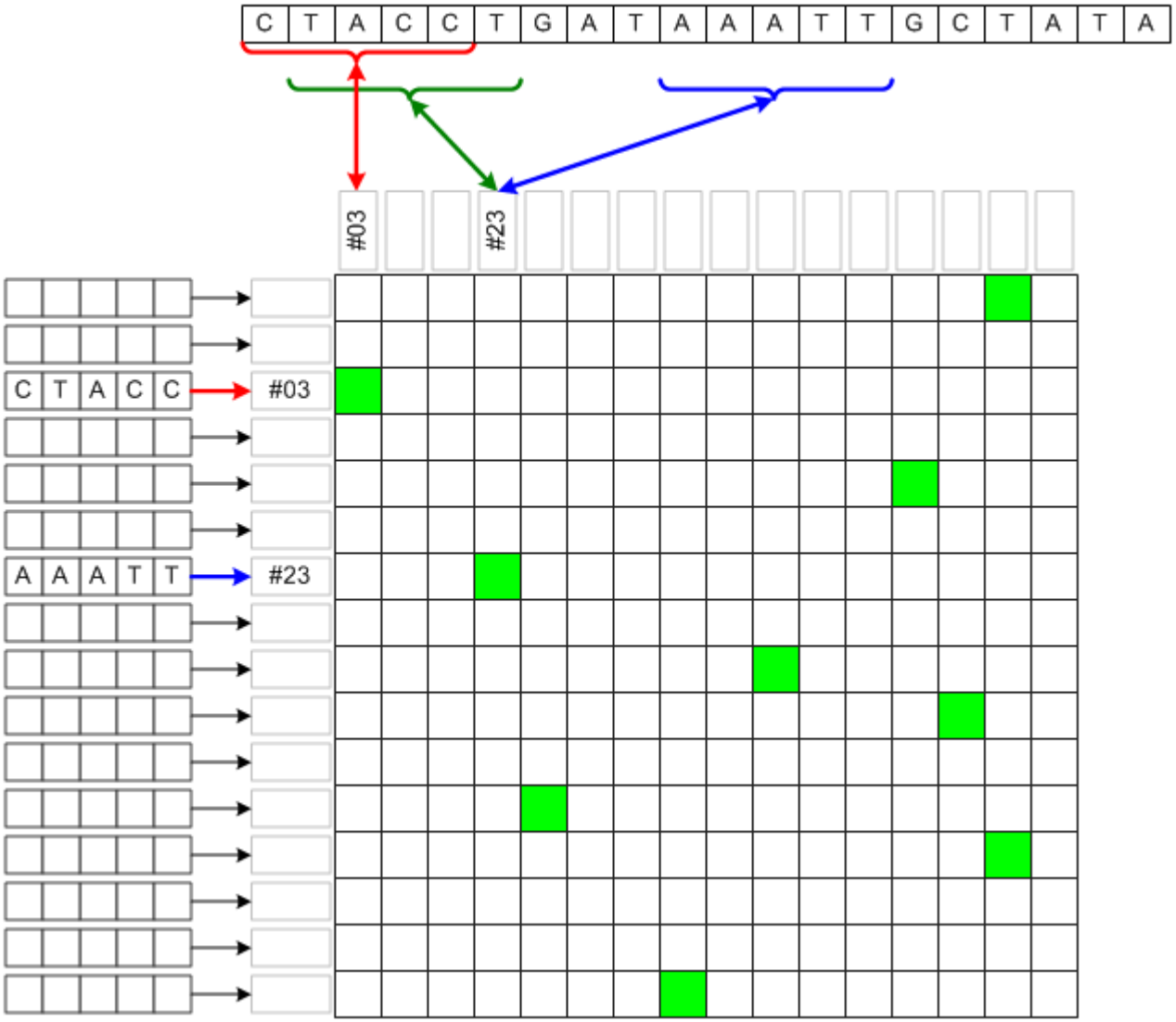
- Computer cluster
  - Scalability
  - Partition the reads
  - Partition the genome
  - Partition both
  - SOAPv1, MapReads, RMAP
  - MPI
  - Middleware - DataCutter

Outline

- Introduction
- Different Approaches
- Parallelization Strategies
- Summary

Mapping Example

- Hash table construction using sliding window
- Table lookup to find matches for each read



Modeling Run Time Costs

c_g : Time to hash a single genome subsequence

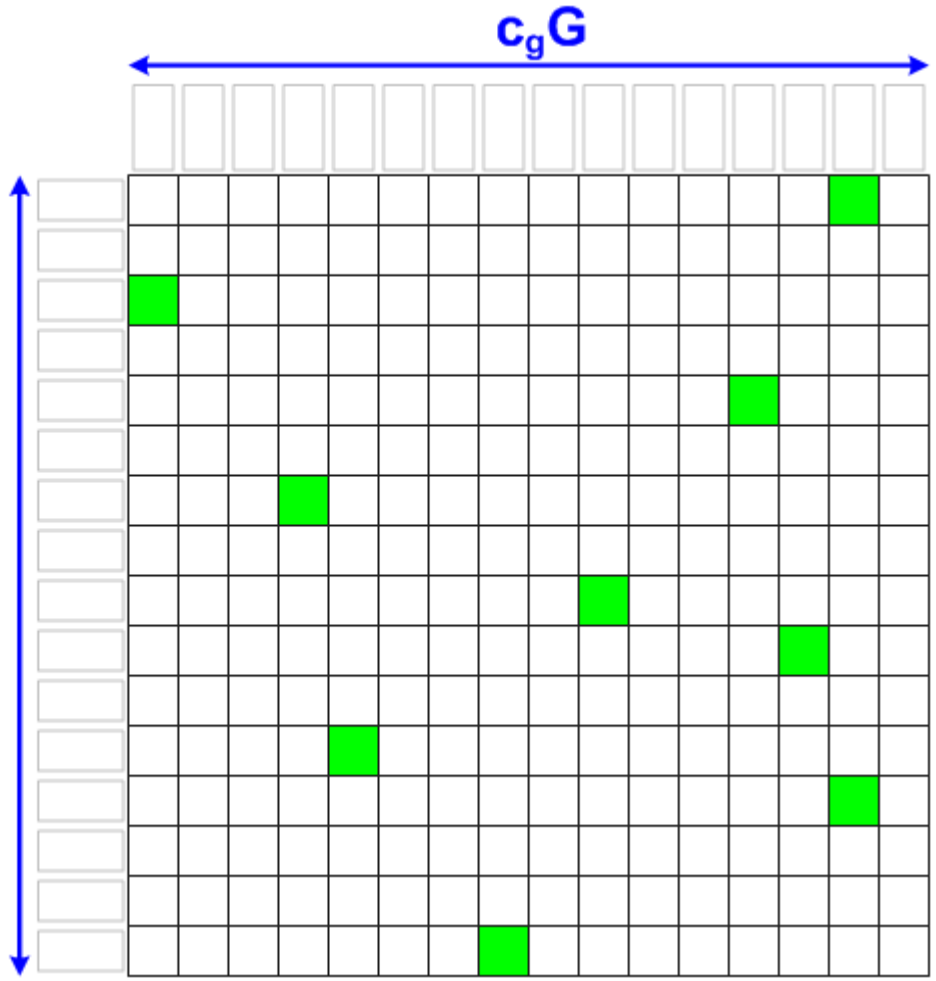
G: Size of genome

c_r : Time to process a single read if no collision

c_c : Time to resolve a collision

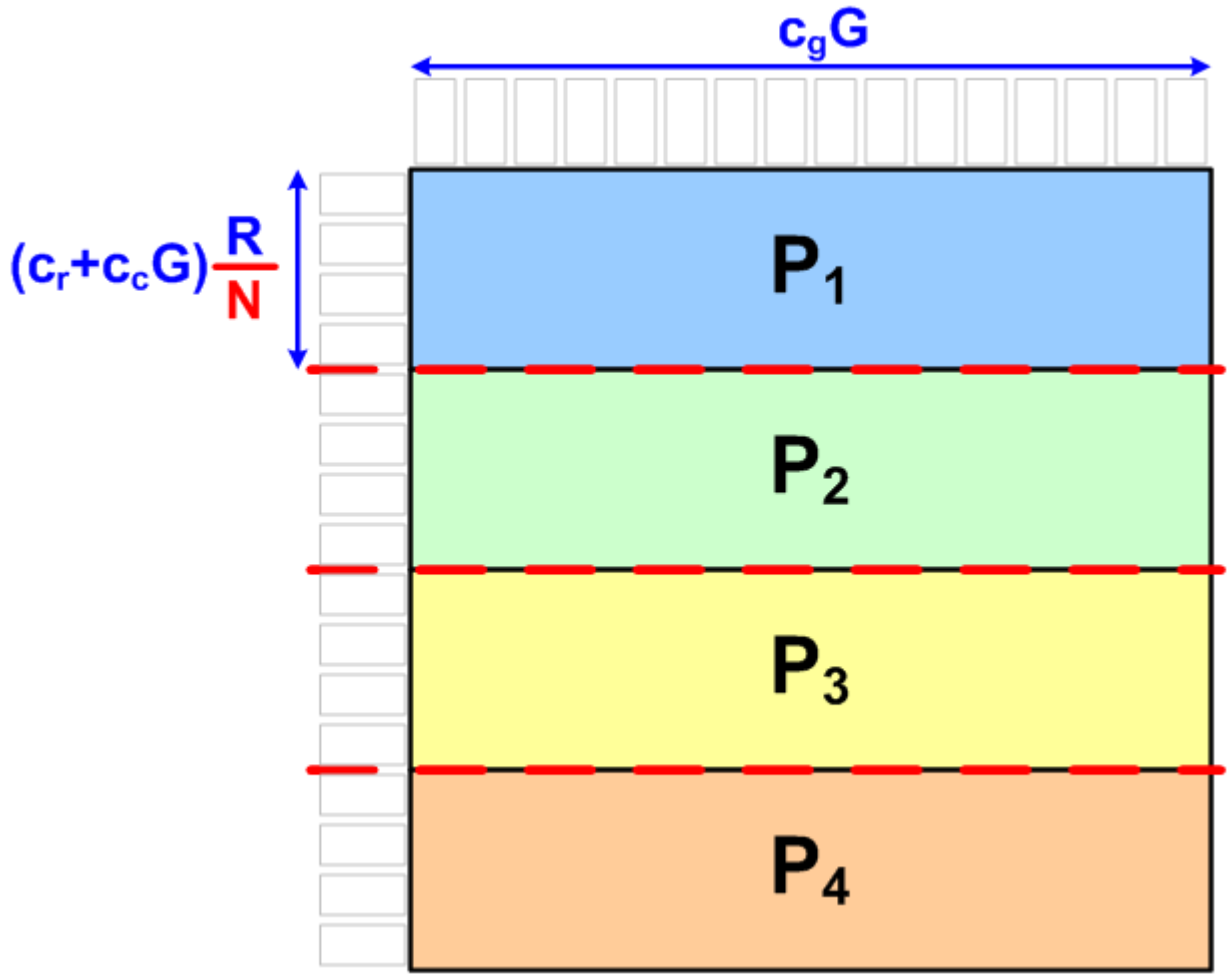
R: Number of reads

N: Number of computation nodes



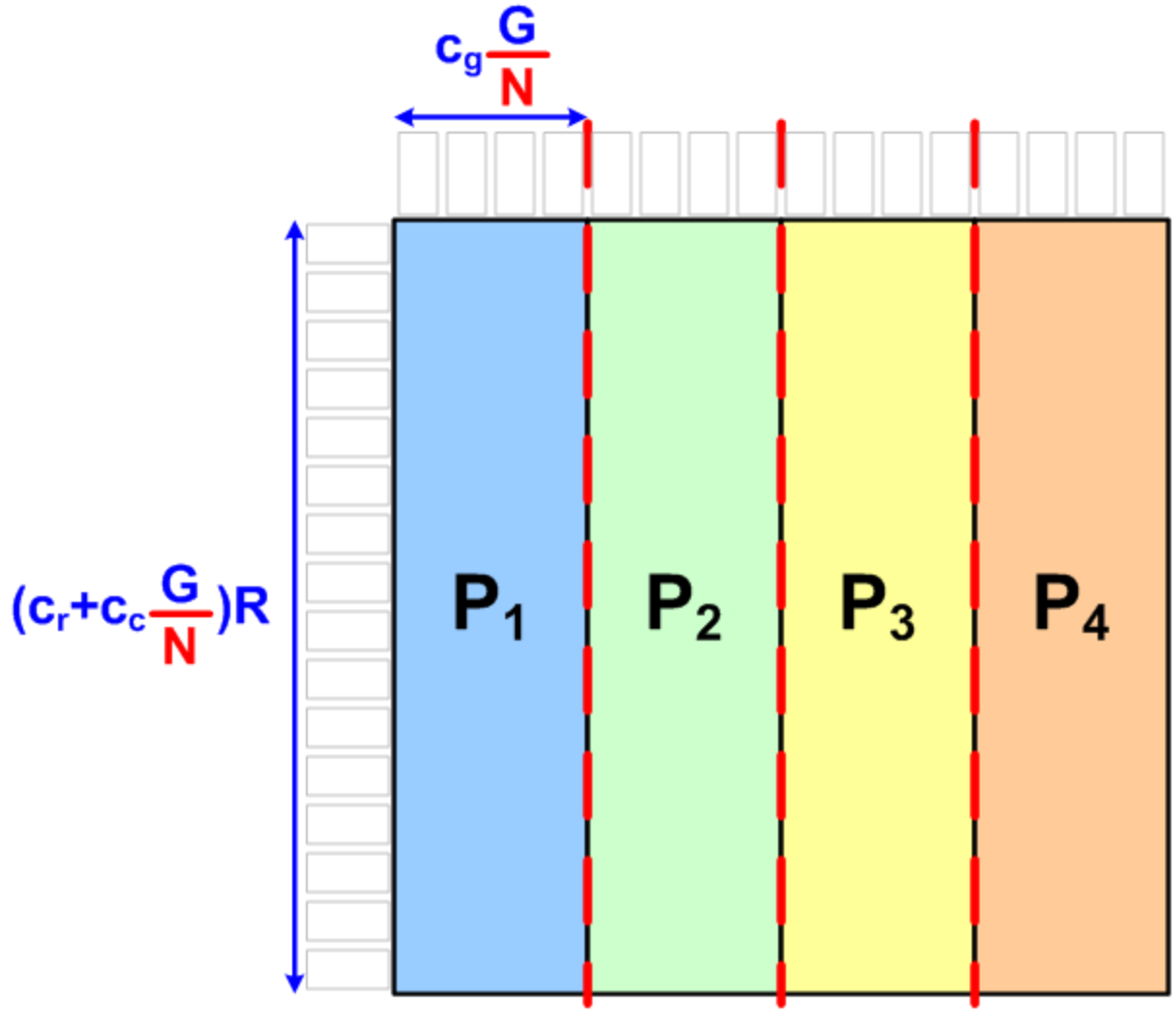
Partition Reads Only (PRO)

- Partition reads into N equal parts.
- Useful when R is large and G is small.
- Memory requirement does not scale



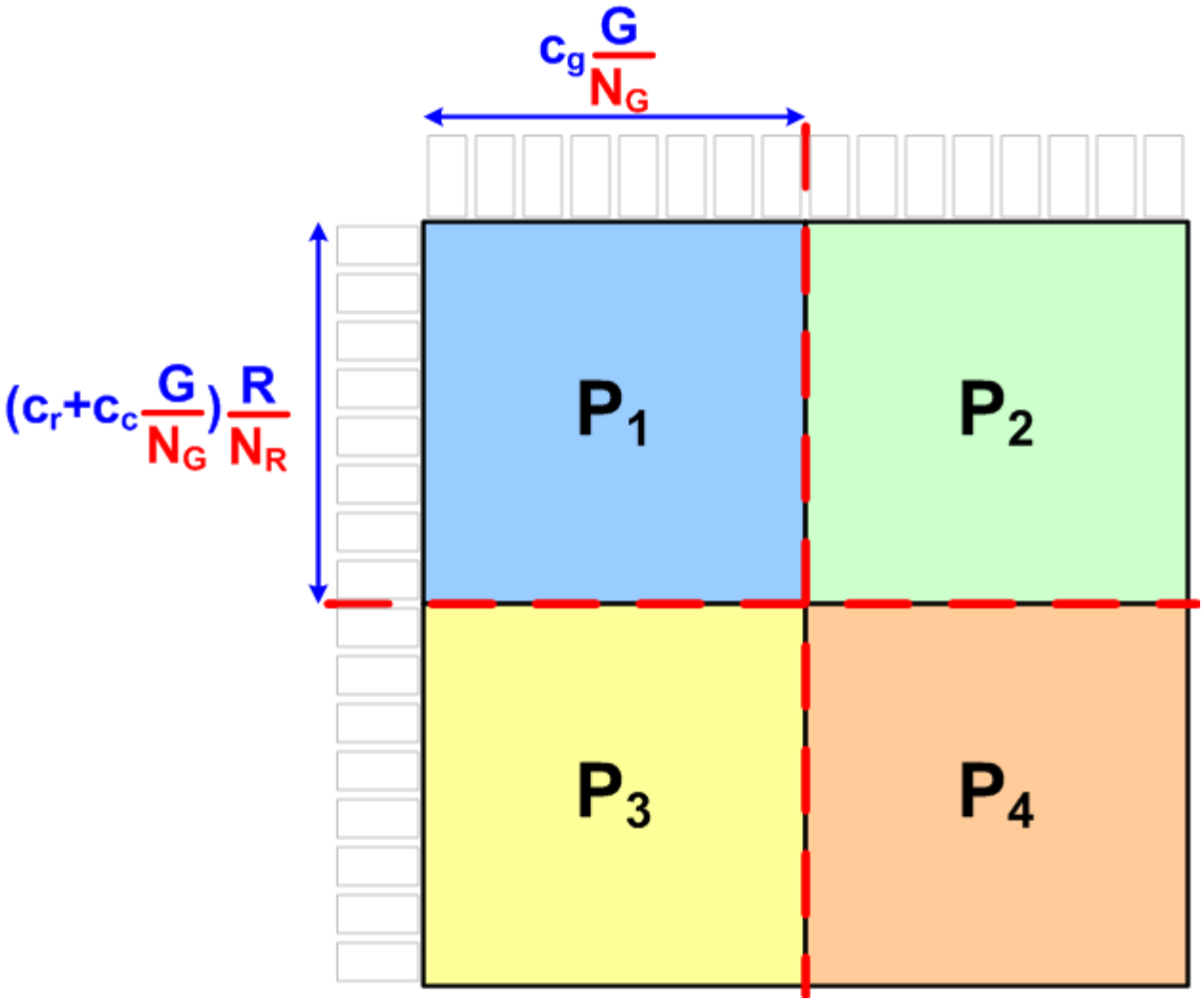
Partition Genome Only (PGO)

- Partition genome into N equal parts
- Useful when G is large and R is small.
- Memory requirement scales perfectly



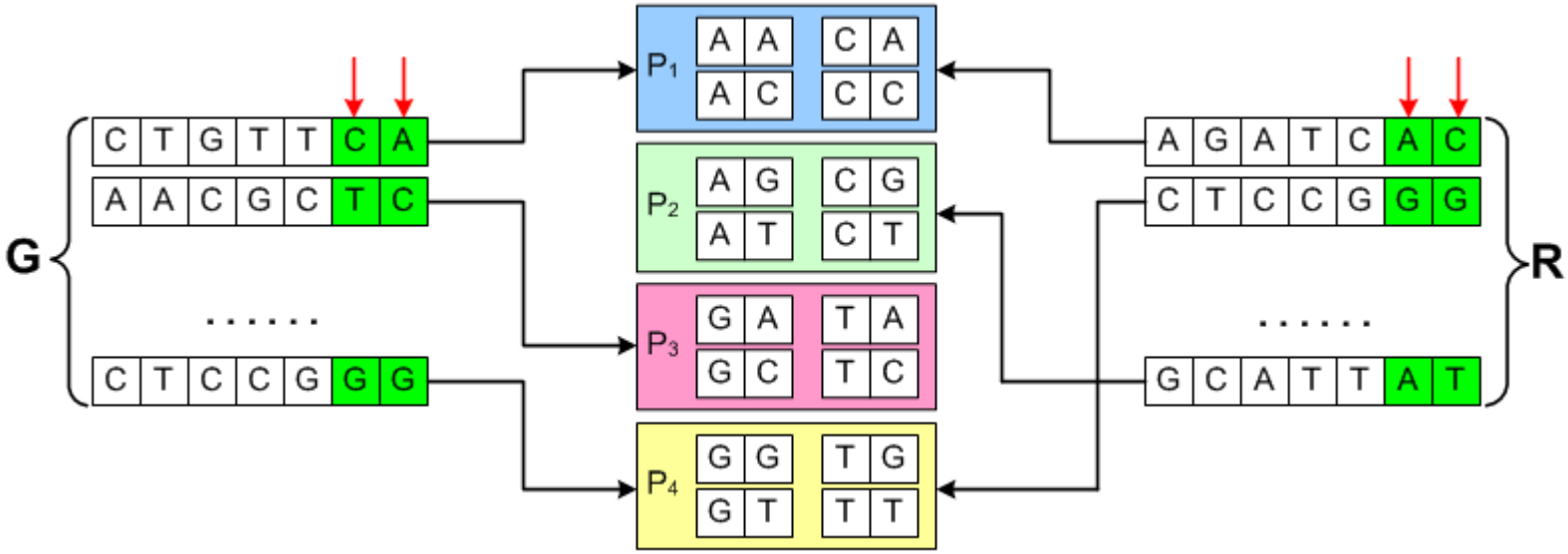
Partition Reads and Genome (PRG)

- A generalization of PRO and PGO
- Nodes are arranged in $N=N_R x N_G$ mesh
- Useful unless $G>>R$ or $G<<R$
- Memory scales worse than PGO, but better than PRO



Suffix Based Assignment

- A new dimension in partitioning load
- Assign a set of suffixes of length **s** to each node
  - 4^s suffixes for a given s
- Each node scans reads and genome subsequences, then only processes those ending with assigned suffixes
  - Only consider the last **s** care positions to handle mismatch cases

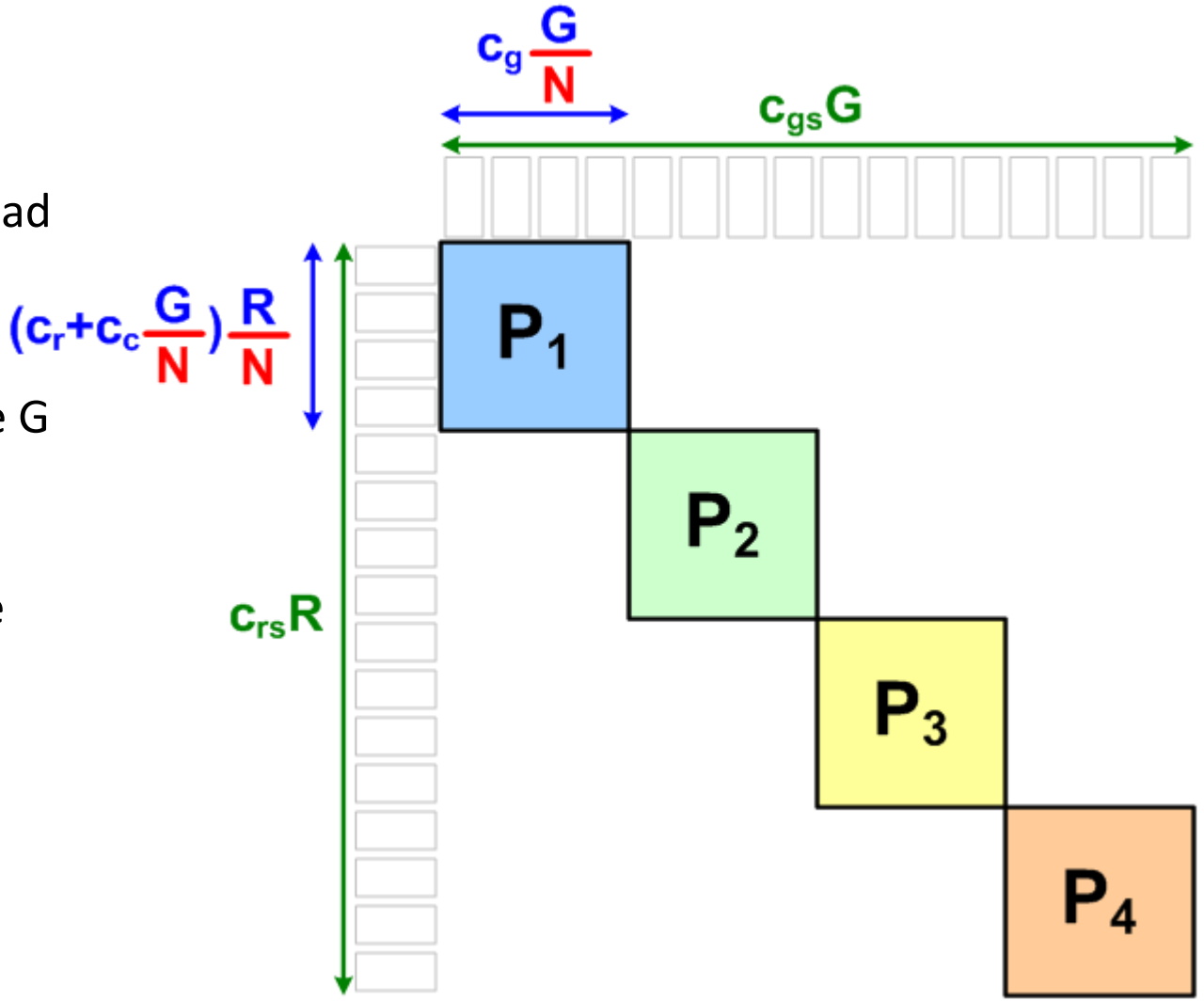


Suffix Based Assignment (SBA)

c_{gs} : Time to compare a genome sequence against suffixes

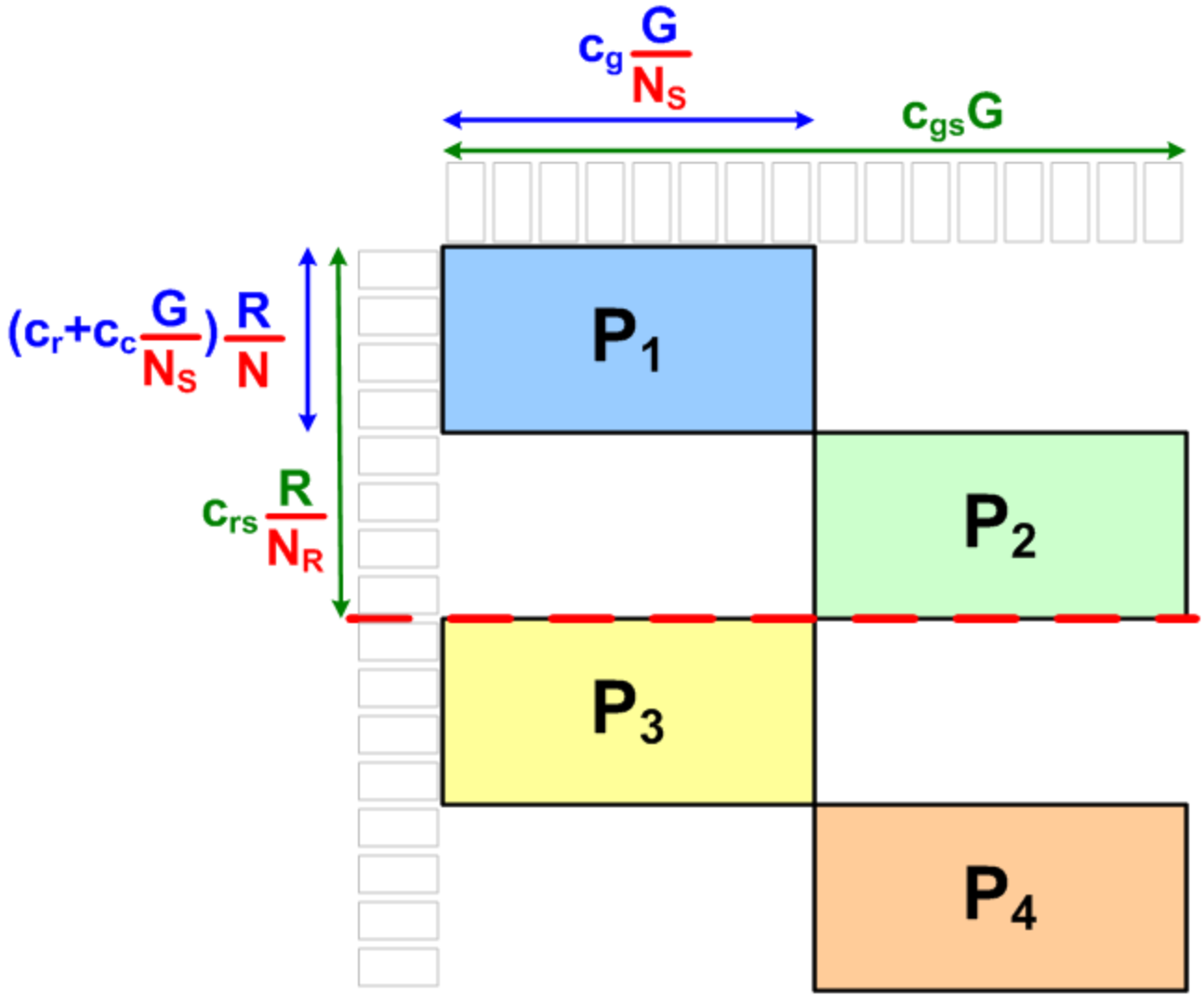
c_{rs} : Time to compare a read against suffixes

- Under perfect balance G and R are partitioned equally
- Limited scalability due to c_{gs} and c_{rs} terms
- Useful for medium values of N
- Memory requirement scales well



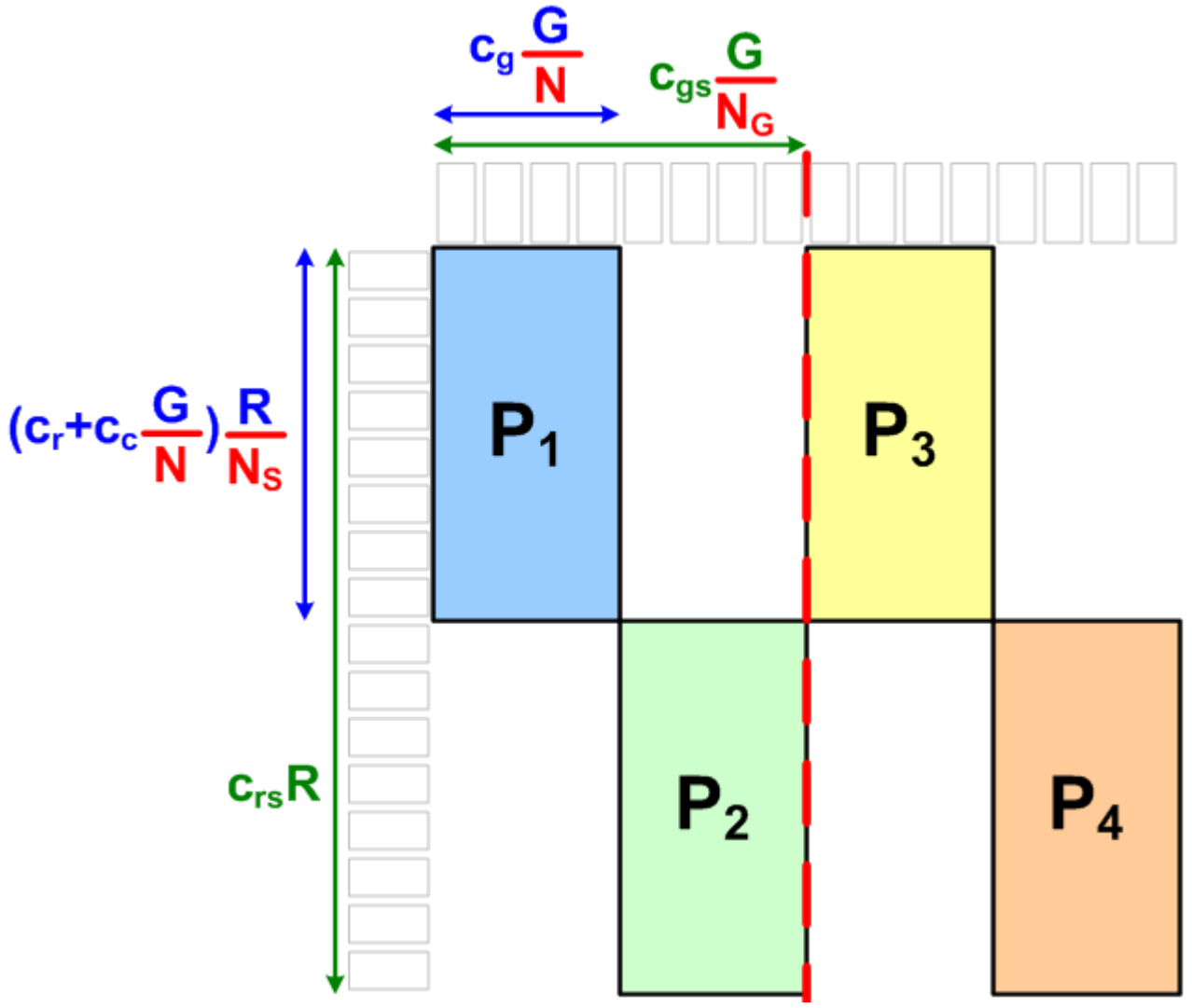
SBA after Partitioning Reads (SPR)

- Partition reads into N_R parts, then apply SBA on each part
- Nodes are arranged in $N=N_R x N_S$ mesh
- Takes advantage of SBA when R is large



SBA after Partitioning Genome (SPG)

- Partition genome into N_G parts, then apply SBA on each part
- Nodes are arranged in $N=N_G x N_S$ mesh
- Takes advantage of SBA when G is large

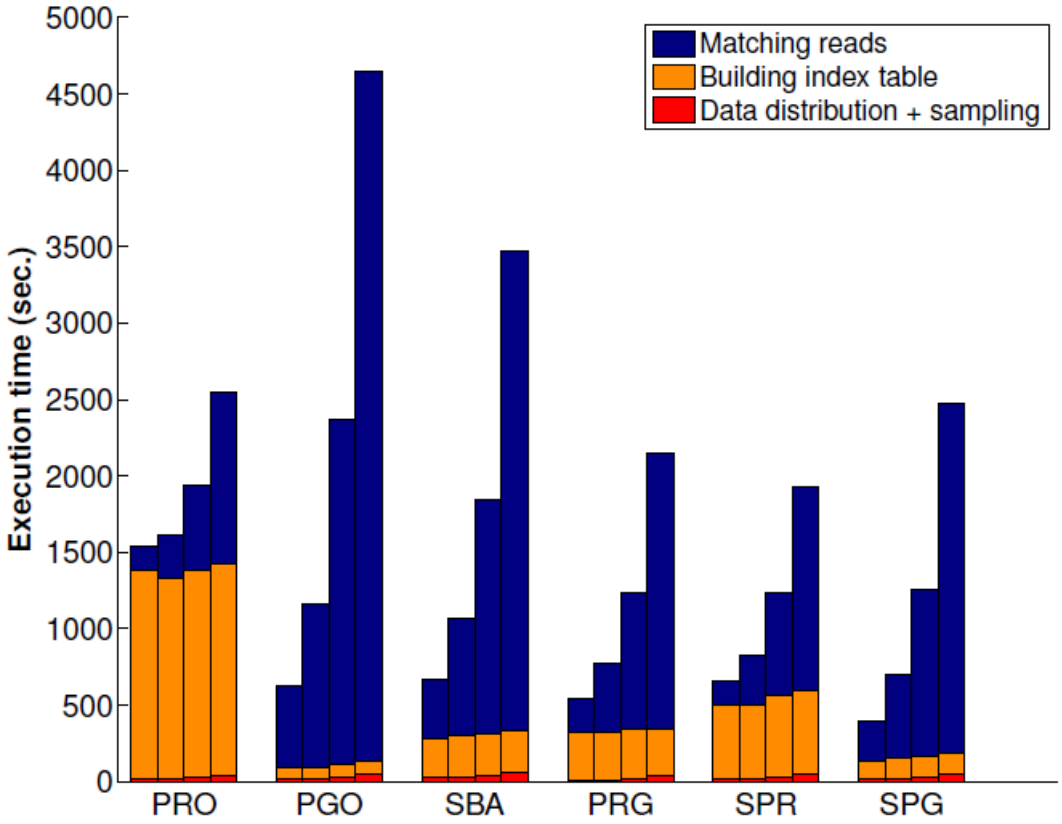


Experimental Setup

- Our implementation is based on *MapReads,* a part of SOLiD System Color Space Mapping Tool
  - Implemented in C using MPI
  - Used default covers with allowing up to 2 mismatches
- Experiments on 64-node dual 2.4GHz Opteron cluster with 8GB memory
- Nodes are interconnected via Infiniband, used MVAPICH v0.9.8
- Reads from a single run of SOLiD system
- Human Genome Build 36.1 (http://genome.uscs.edu)
- $N_R = N_G = N_S = \sqrt{N}$

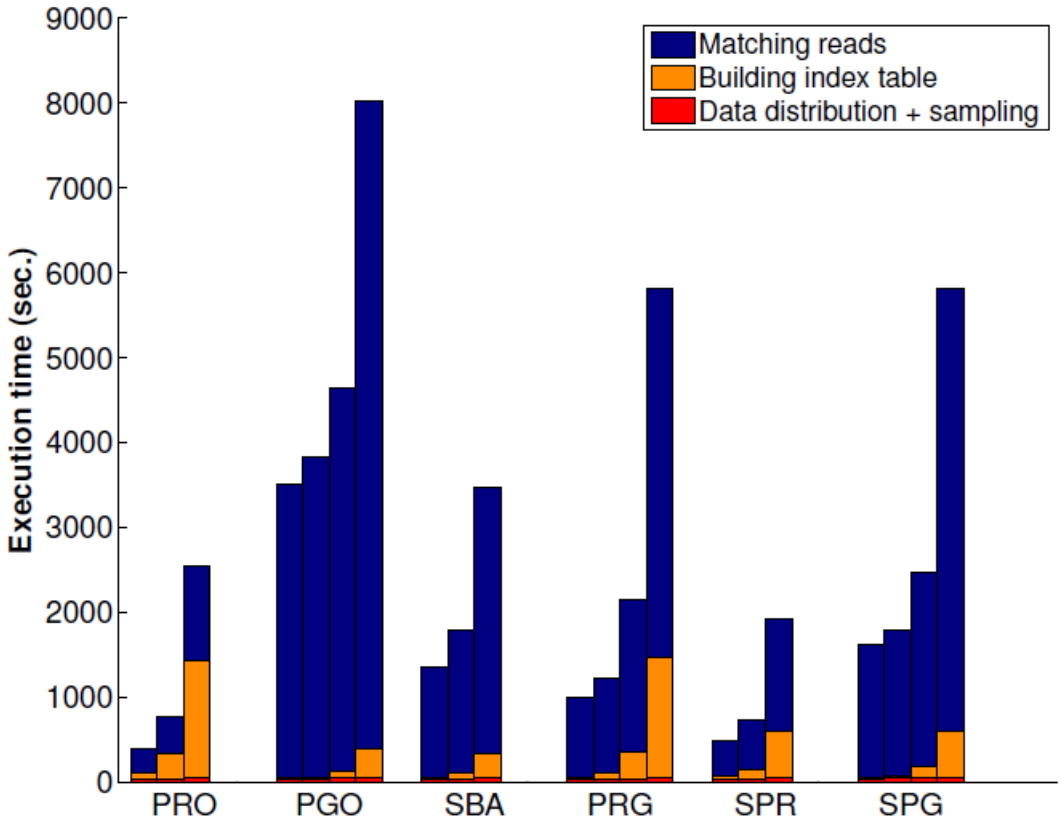
Varying Number of Reads

- G: 800M, **R: (16M, 32M, 64M, 130M)**, L: 50, N:16
- Partitioning reads helps reducing matching time



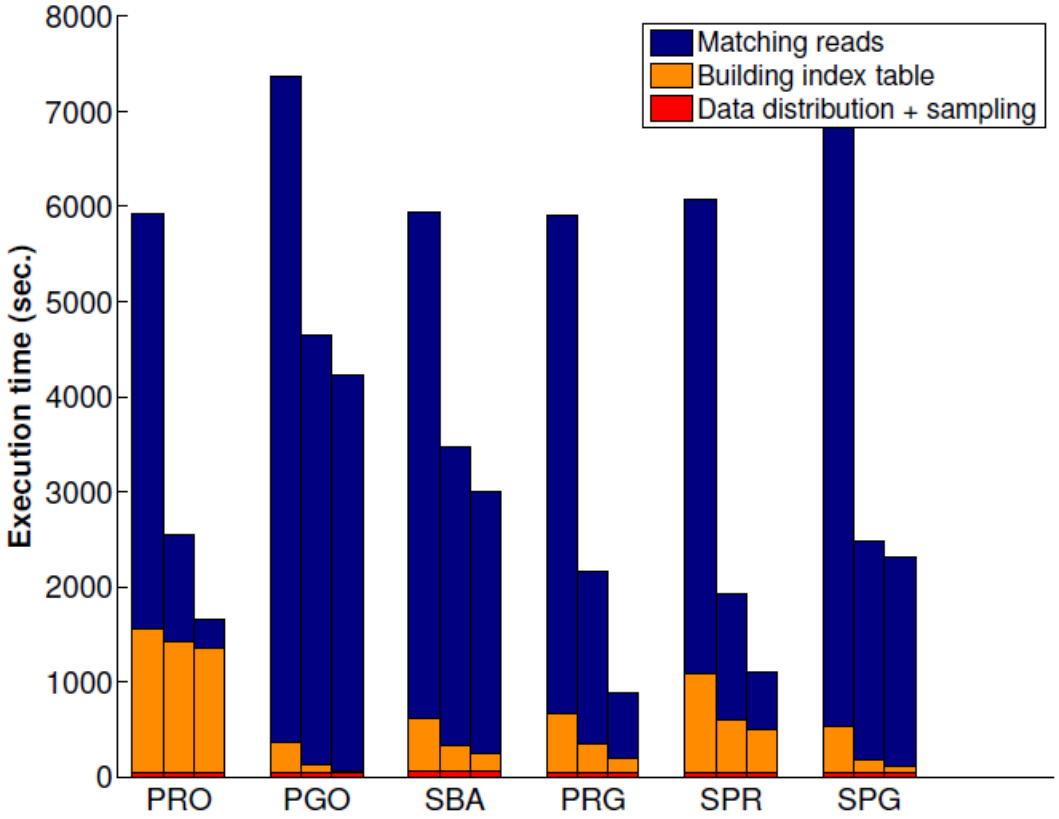
Varying Genome Size

- G: (50M, 200M, 800M, 3080M), R: 130M, N:16
- Partitioning genome helps reducing hashing time



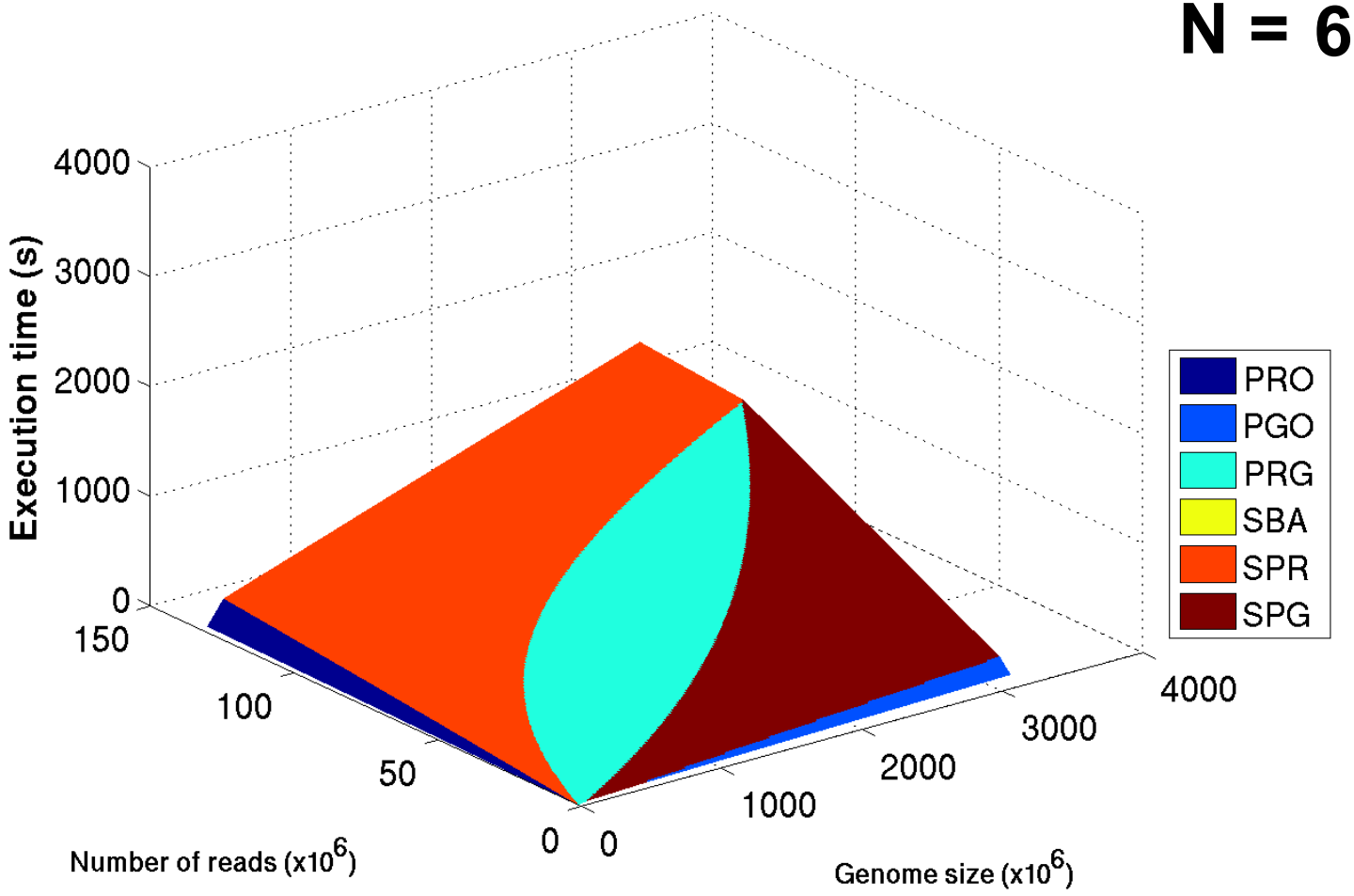
Varying Number of Nodes

- G: 800M, R: 130M, **N: (4, 16, 64)**
- Up to 22x speedup: **From a day to an hour!**



Run Time Prediction

N = 64



Outline

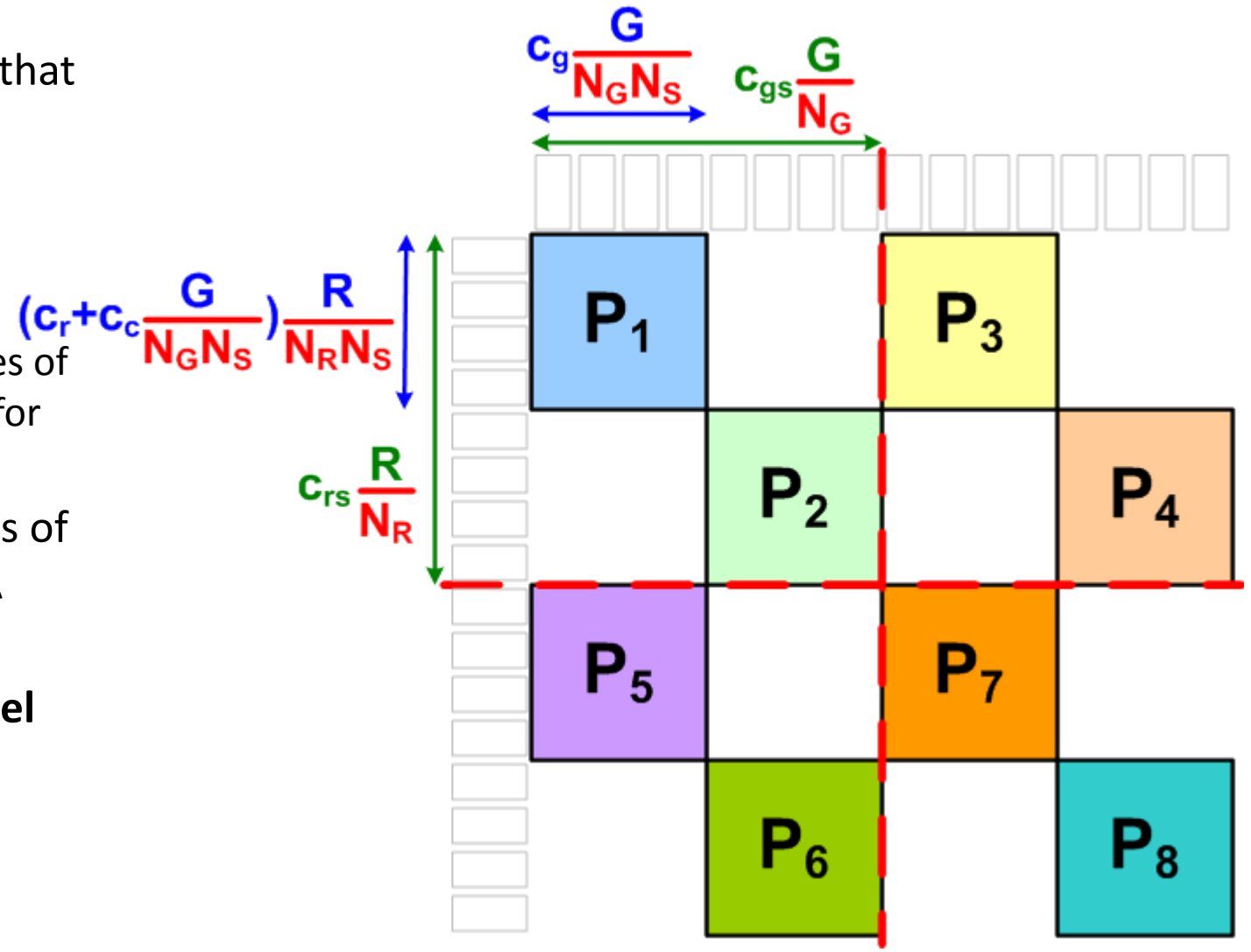
- Introduction
- Different Approaches
- Parallelization Strategies
- Summary

Conclusions

- Three strategies for parallel computing of mapping
- Proposed 6 parallelization methods for short sequence mapping using computer clusters
- Extensively analyzed performance of each method wrt. genome size, number of reads and number of nodes
  - Described theoretical cost models
  - Evaluated performance experimentally
- Proposed a prediction function to select the best method for a given scenario
- Achieved fairly good speedup that allows reducing the mapping time from a day to an hour.

Future Work

- A general model that encompasses all parallelization methods
  - $N=N_R x N_G x N_S$
  - Find best values of N_R, N_G and N_S for given R, G, N
- Investigate causes of imbalance in SBA
- **Develop a web service for parallel short sequence mapping**



Acknowledgement

- Dr. Catalin C. Barbacioru (ABI)
- Dr. Hatice Gulcin Ozer
- Dr. Jeffery Parvin

- CAMDA organizers and committee

- NIH – ICBP (*5U54CA113001-05*), *1R01ES017594-0109*
- NSF, DOE, PhRMA Foundation
- OSUCCC