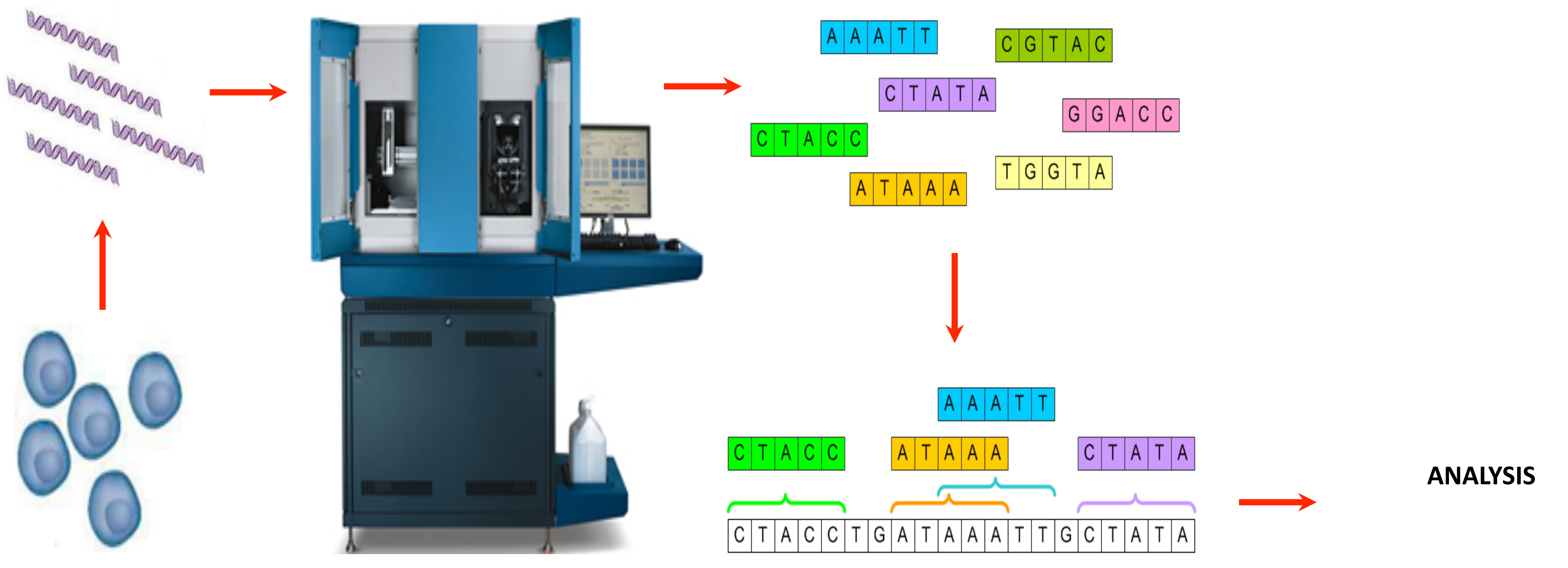
# Parallel Short Sequence Mapping

Doruk Bozdag*, Ayat Hatem*^, Umit Catalyurek*^
*Biomedical Informatics, ^Electrical and Computer Eng., The Ohio State University

This work was supported in parts by the NIH/NCI grant R01CA141090; the DOE grant DE-FC02-06ER2775; by the NSF grants CNS-0643969, OCI-0904809, OCI-0904802 and CNS-0403342; and an allocation of computing time from the Ohio Supercomputer Center.

## The Problem

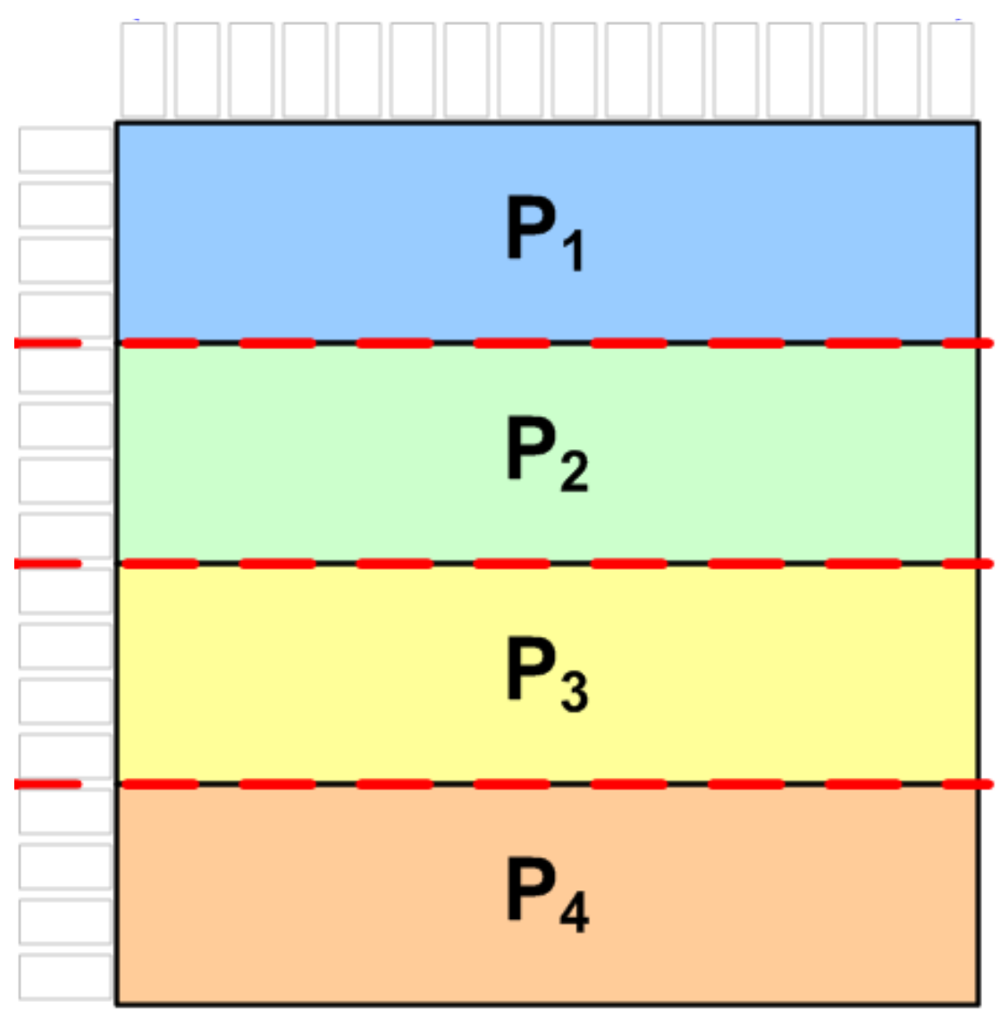


- Next generation sequencing instruments (SOLiD, Solexa, 454) can sequence up to 20 billion bases in a single run
  - SOLiD 3 system can generate 400M reads of length 35-50 bases in a single run
- Reads should be efficiently mapped to a reference genome
  - Human genome: 3 billion bases
- Sequentially mapping reads generated at a single run to a human genome takes time on the order of days
- Fast, resource efficient, parallel algorithms that can handle mismatches are required

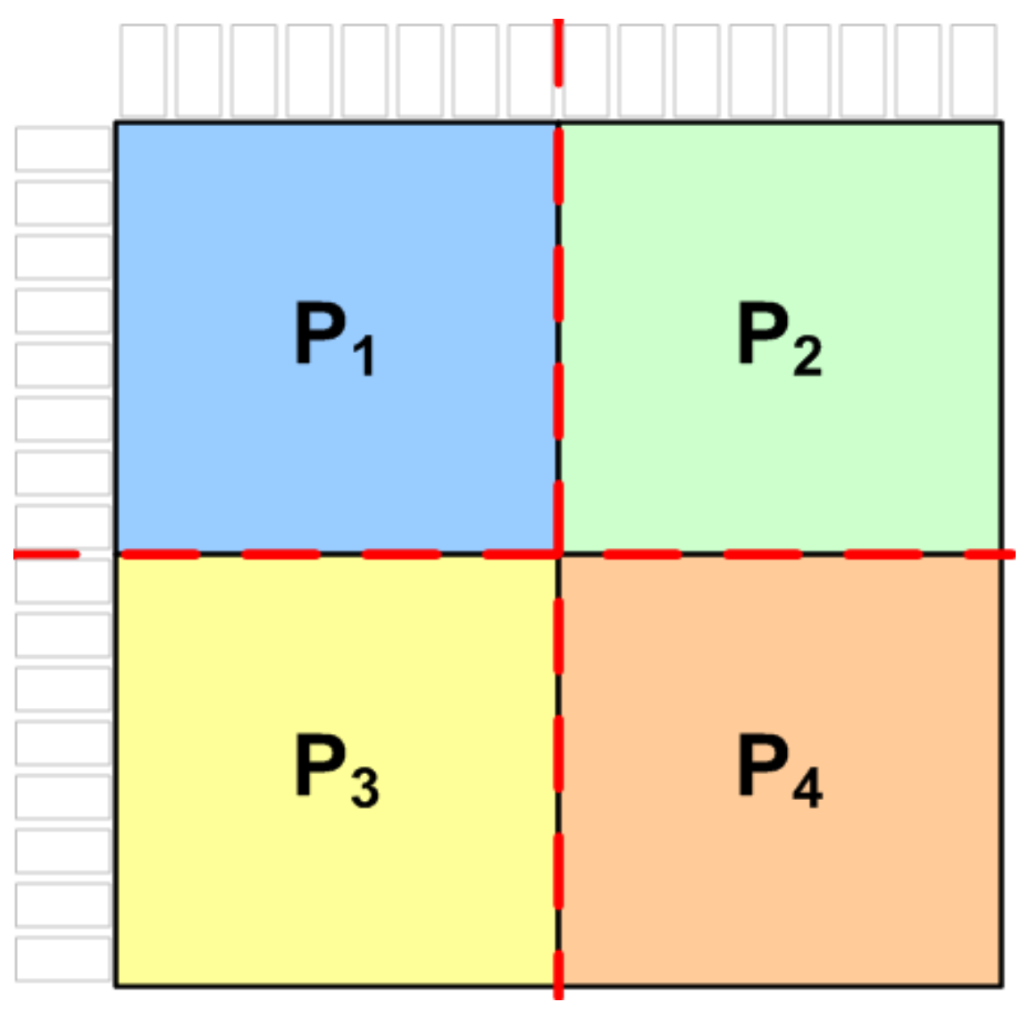
## Parallelization Methods

**Parameters**

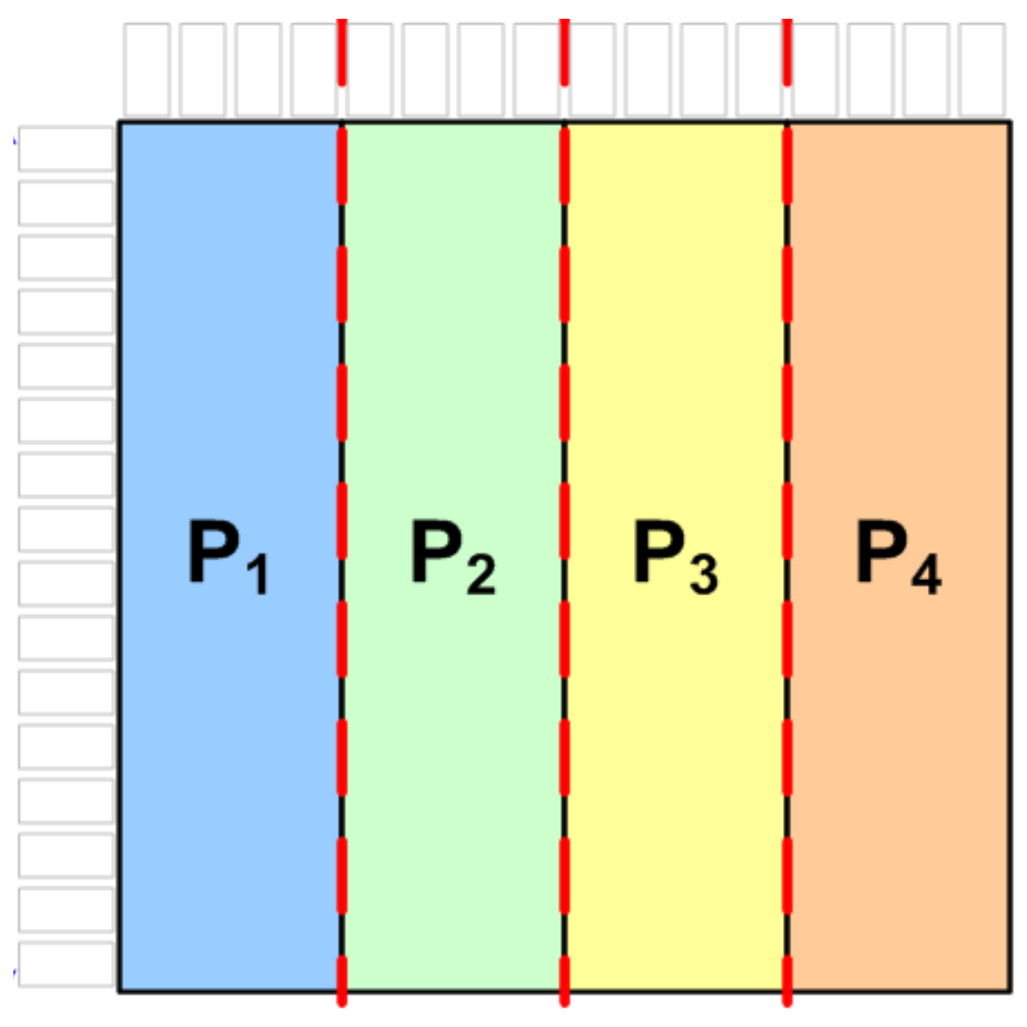
G: Genome size
R: Number of reads
N: Number of nodes
NR: Number of read parts
NG: Number of genome parts



- Partitioning reads is useful when R is large and G is small

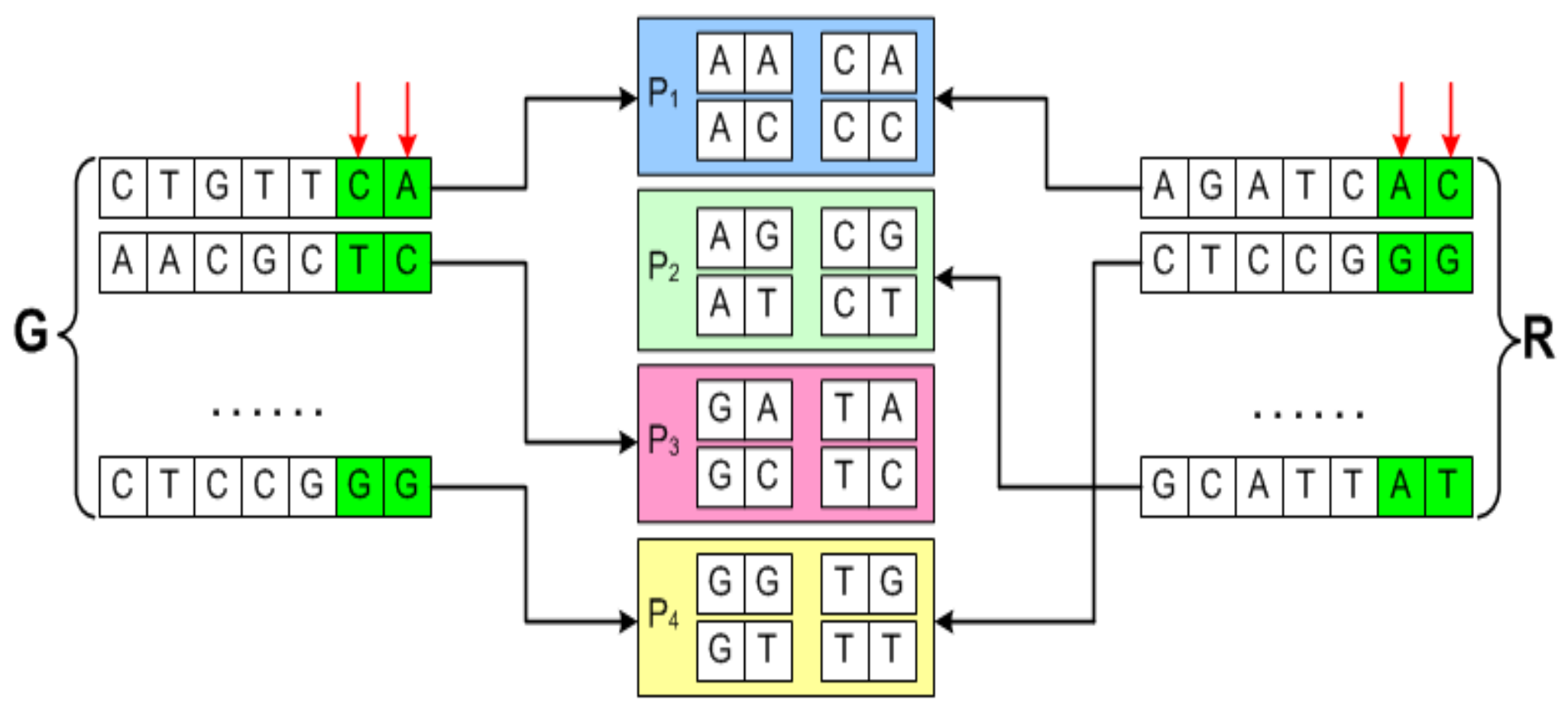


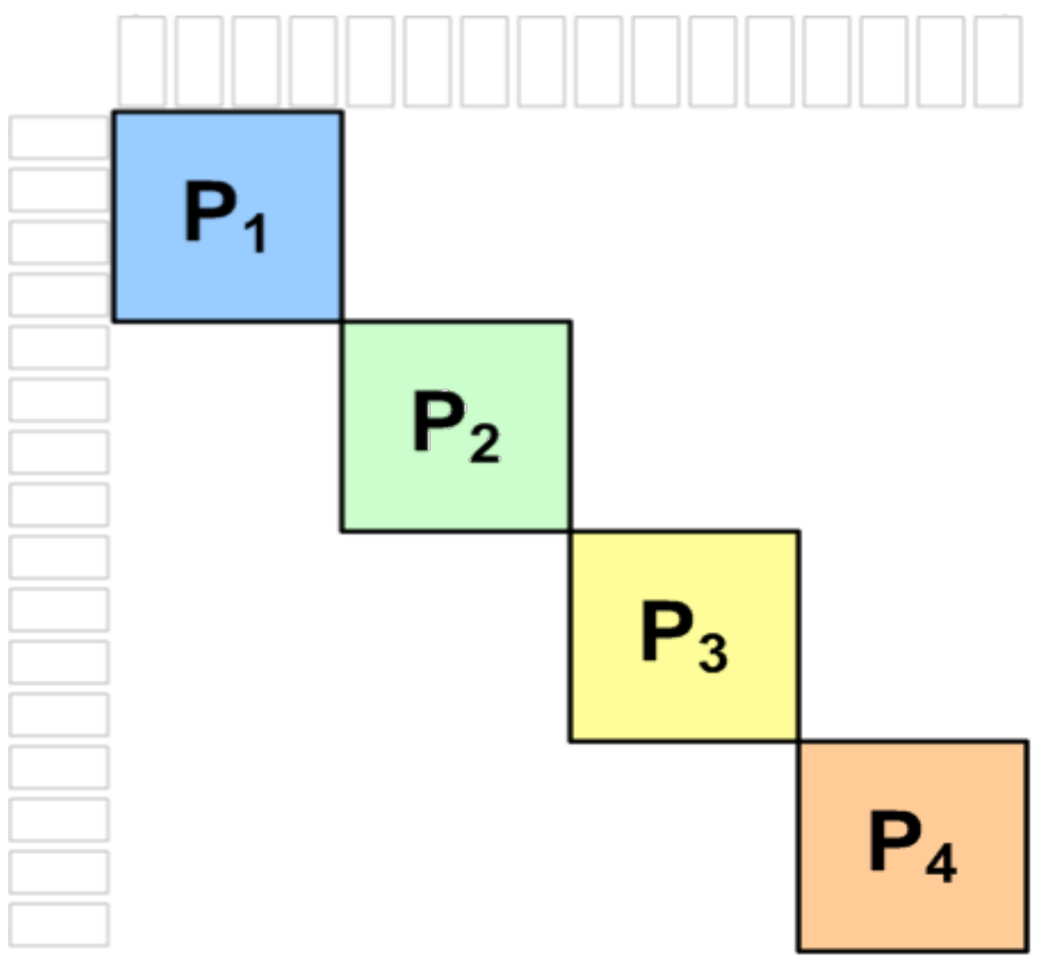
- Partitioning both reads and genome is useful unless G >> R or G << R



- Partitioning genome is useful when G is large and R is small

- In a previous work, we had also introduced a new dimension to partition the load for hashing based methods:
  - Assign a set of suffixes S to each node
  - Each node only processes reads and genome sub-sequences ending with assigned suffixes
  - Under perfect balance, G and R are partitioned equally
- Useful for medium values of N
  - Additional scanning steps are not scalable
- D. Bozdag, C. Barbacioru, U. Catalyurek, “Parallel Short Sequence Mapping”, IPDPS’09.

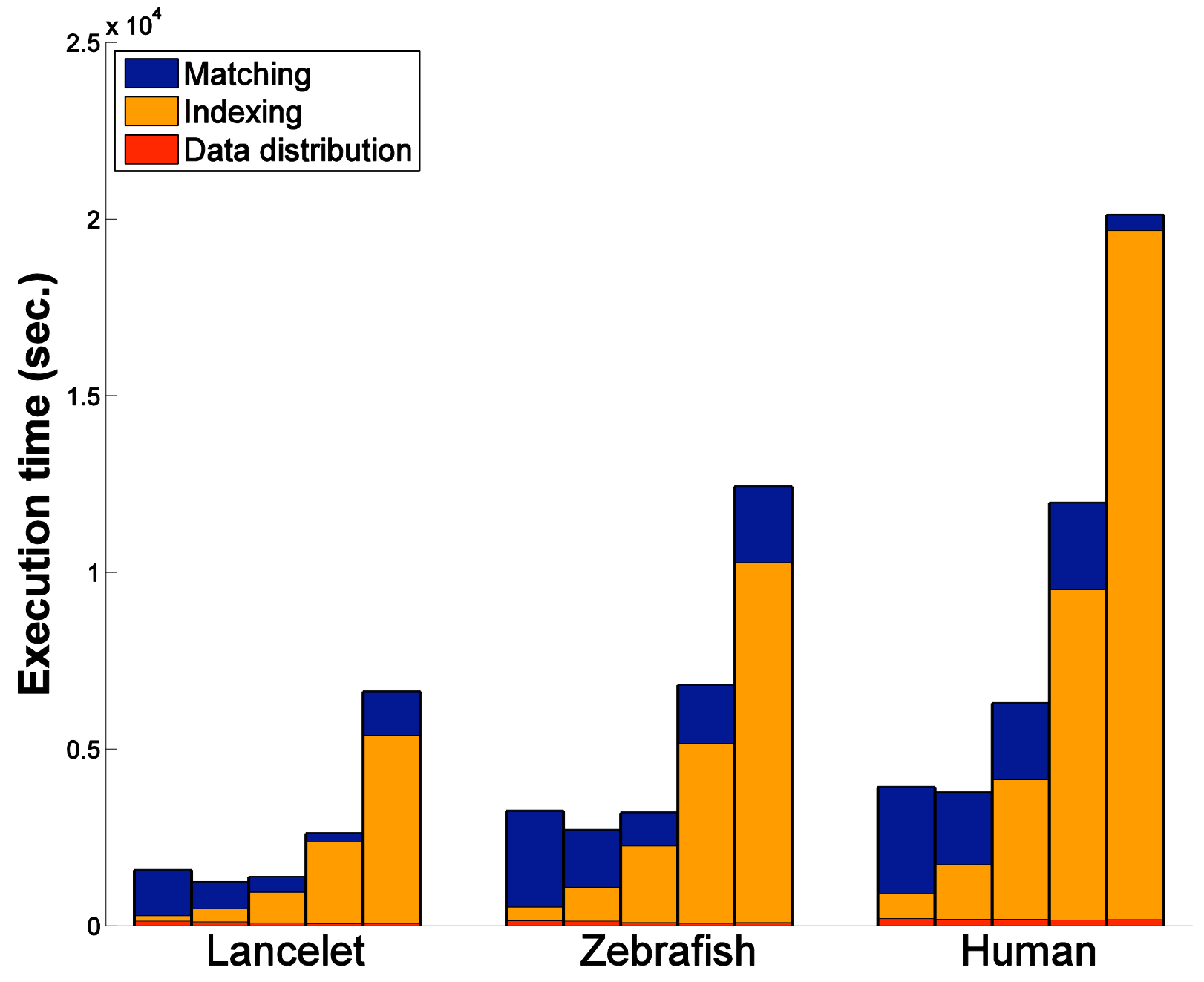




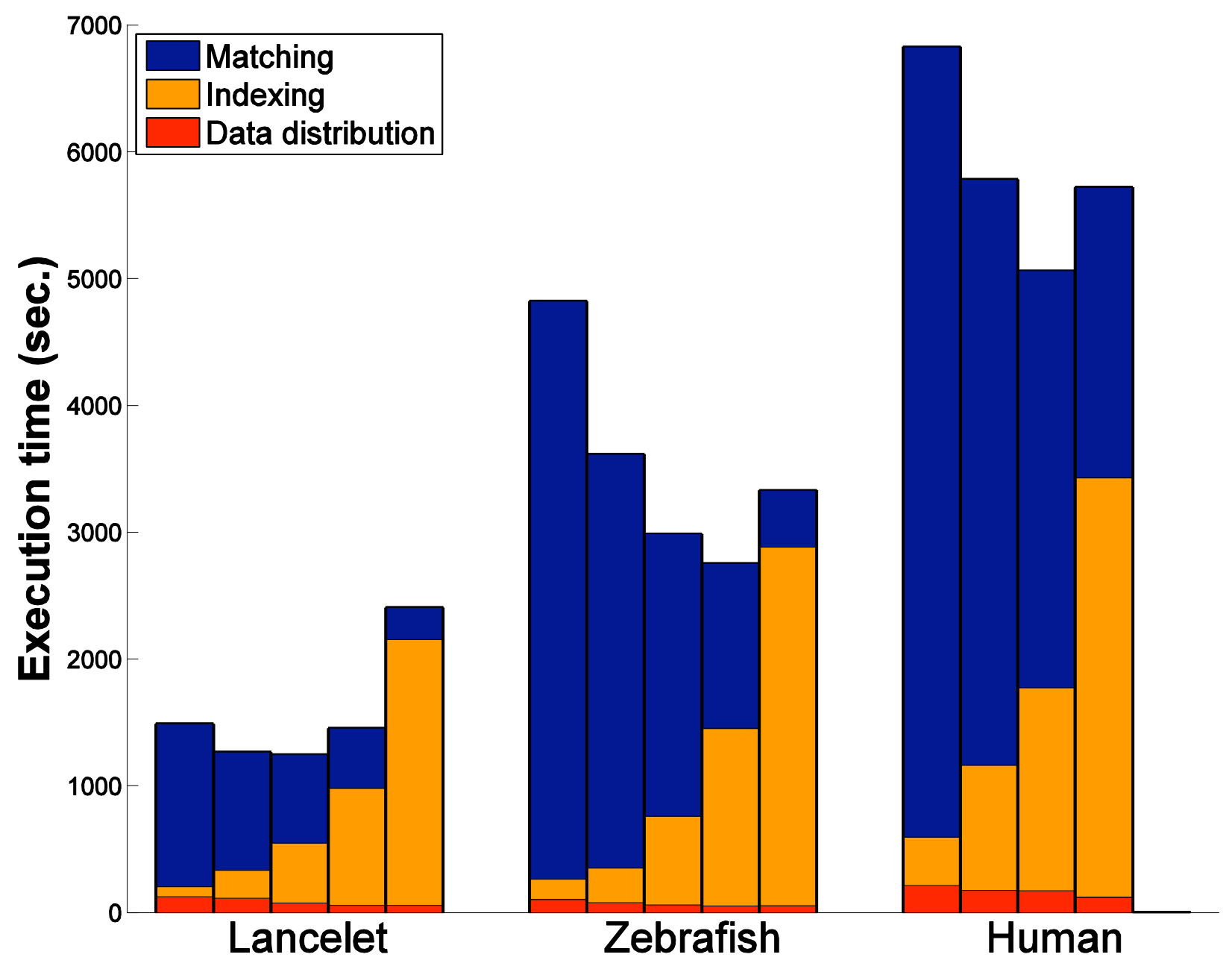
## Results and Conclusions

- We applied reads and genome partitioning methods for parallelizing Bowtie and BWA, recently introduced Burrows Wheeler Transform based short sequence mapping tools.
- Results from various input scenarios on 16 dual-core Opteron nodes are given.
  - Each group of five bars correspond to the following NRxNG configurations from left to right: 1x16, 2x8, 4x4, 8x2, 16x1. Two threads are used in matching phase.
  - The reads were generated using wgsim tool, part of samtools package.

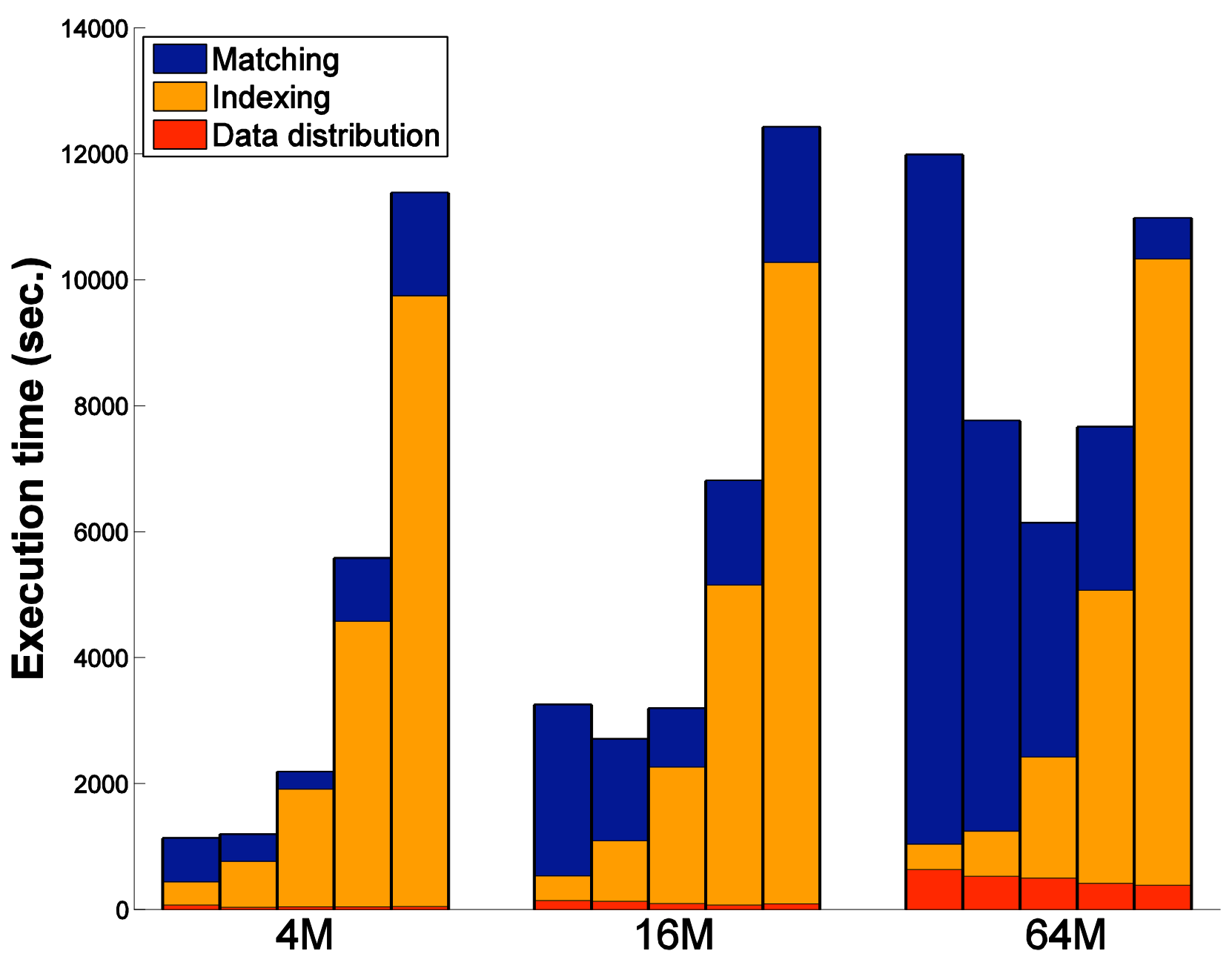
Mapping 16M reads to different genomes using Bowtie



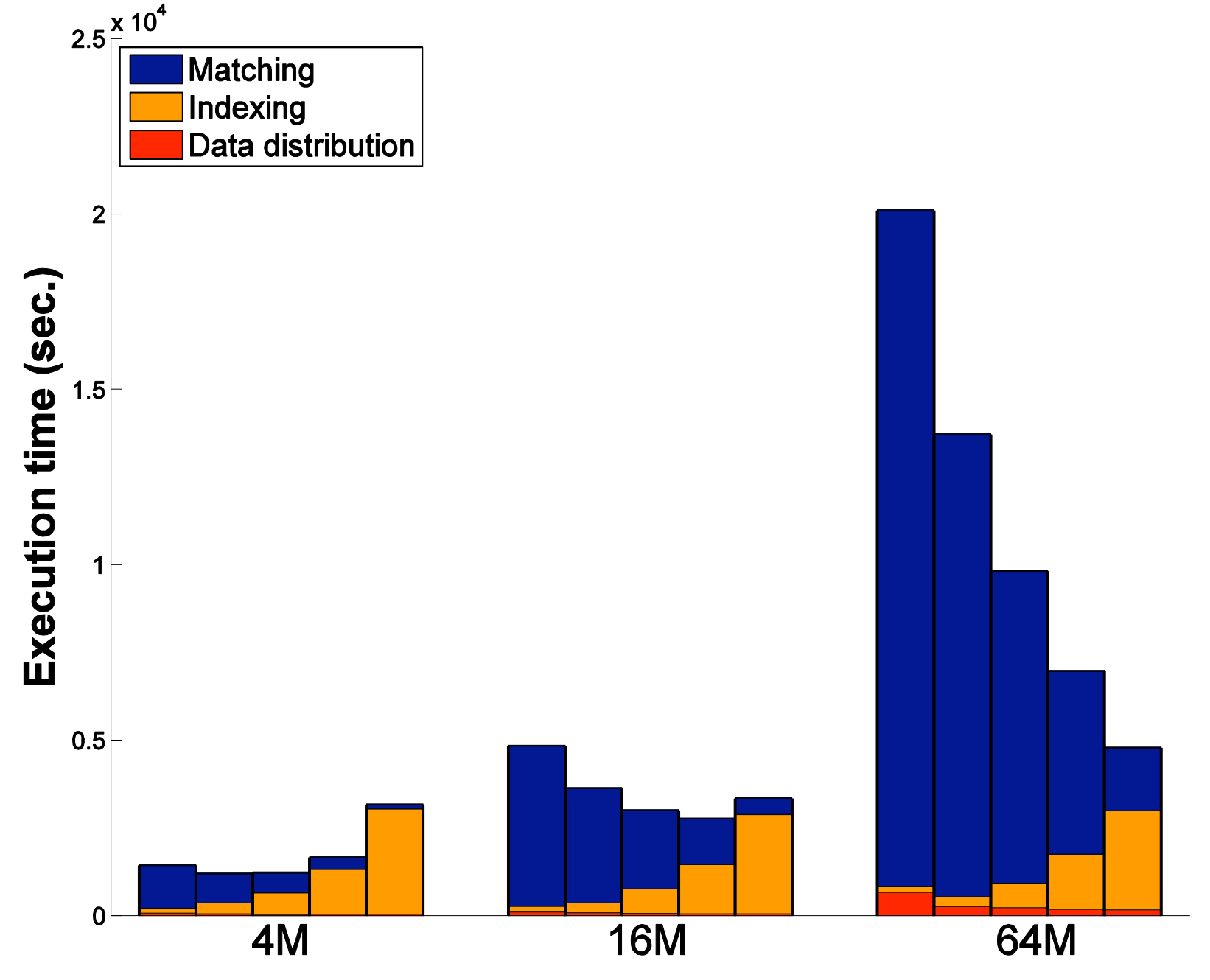
Mapping 16M reads to different genomes using BWA



Mapping different number of reads to Zebrafish genome using Bowtie



Mapping different number of reads to Zebrafish genome using BWA



**Conclusions**

- In general, indexing time is almost halved when NG is doubled. However, NR does not have the same effect on the matching time.
- Partitioning the genome also helps reducing the matching time.
- The best NRxNG configuration depends on the values of R, G and N. There’s no single “best configuration”.