CompBioMed2
March 16th, 2020
LRZ, Germany

Software development and optimisation of bioinformatics pipelines to analyse high-throughput sequencing data in oncology

Philippe Hupé
Bioinformatics facility Curie Core Tech
Institut Curie, INSERM U900, CNRS UMR144

Outline

Outline

Institut Curie, comprehensive cancer center

Research-care continuum initiated by Marie Curie and Claudius Regaud

3 hospitals

- 9.800 new patients / year
- 15.300 patients with treatment (76% women, among them 7.100 breast cancer)
- 500 patients with pediatric cancer

1 multidisciplinary research center

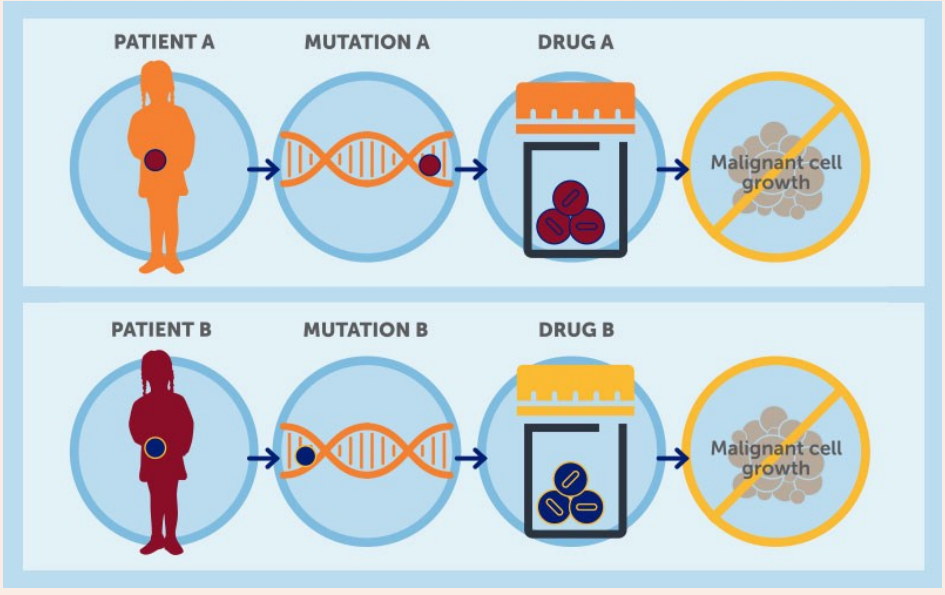
- 12 research units (with INSERM, CNRS and universities)
- 1 department of translational research
- 16 technical core facilities (Curie CORE TECH)

Bioinformatics core facility Unité 900 ($\sim$20 pers.) - created in 2002

- Transversal and institutional missions (management and biostatistical analysis of omics data, HPC, training)
- Asset to be part of a research unit

Support for precision medicine

Tailor the treatment based on genomic alterations in the tumor

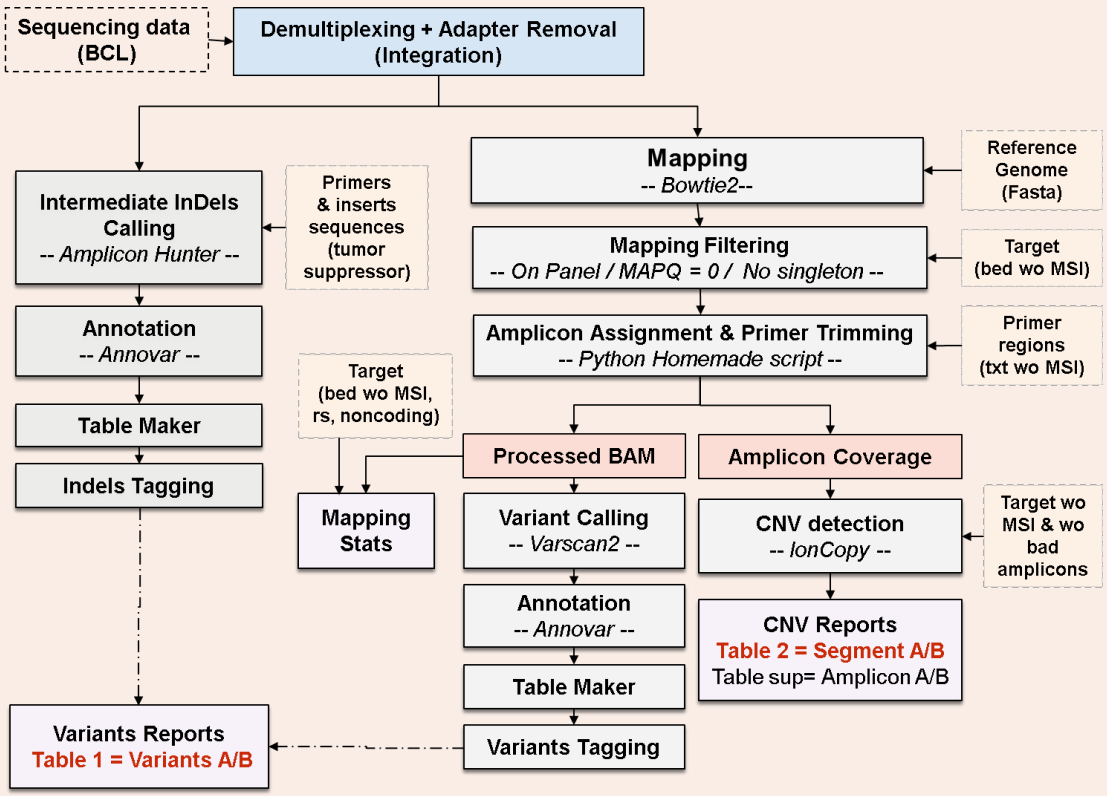


- develop new methods
- implement pipelines for routine production
  - research
  - diagnostic
    - precision medicine for 10.000 patients / year
    - results must be delivered within 48h once the data are available

Outline

Bioinformatics pipeline for high-throughput sequencing data

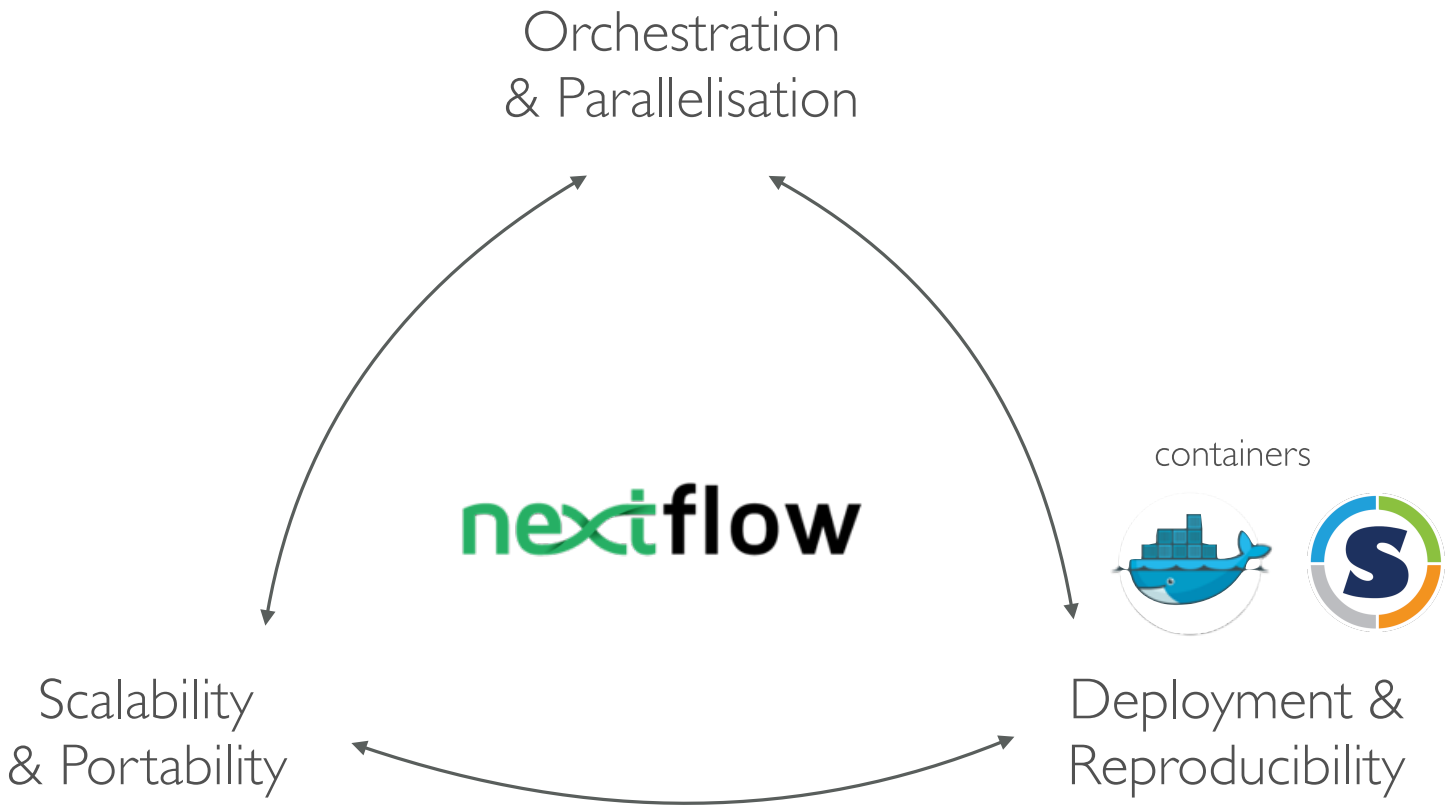
Bioinformatics pipeline to detect and annotate mutations and DNA copy number alterations that are used for the therapeutic decision



- many different steps that can be parallelized
- various languages (bash, python, R, java, C/C++, ...)

Nextflow workflow manager[1]

Overview



[1] adapted from Di Tommaso et al., 2017's slides

A process consists of three blocks

```
process align_sample {

  input:
  file 'reference.fa' from genome_ch
  file 'sample.fq' from reads_ch

  output:
  file 'sample.bam' into bam_ch

  script:
  """
  bwa mem reference.fa sample.fq \
        | samtools sort -o sample.bam
  """

}
```

[1] adapted from Di Tommaso et al., 2017's slides

Input / output are connected by a channel

```
process align_sample {

   input:
   file 'reference.fa' from genome_ch
   file 'sample.fq' from reads_ch

   output:
   file 'sample.bam' into bam_ch


   script:
   """
   bwa mem reference.fa sample.fq \
         | samtools sort -o sample.bam
    """

}
```

```
process index_sample {

   input:
   file 'sample.bam' from bam_ch

   output:
   file 'sample.bai' into bai_ch


   script:
   """
   samtools index sample.bam
    """

}
```

- Processes wait for their input data
- When an input set is ready the process is executed
- They communicate by using dataflow variables (the channels)
- Parallelisation and tasks dependencies are implicitly defined by process in/out declarations

[1] adapted from Di Tommaso et al., 2017's slides

Easy to launch the pipeline on a computing cluster


```
process {
 executor = 'slurm'
 queue = 'my-queue'
 memory = '8 GB'
 cpus = 4
 container = 'user/image'
}
```

[1] adapted from Di Tommaso et al., 2017's slides

nf-geniac

Automatic Configuration GENerator and Installer for nextflow pipeline

One software = one container that can be used by several processes

nextflow

```
process standardUnixCommand {
  label 'onlyLinux'
  label 'smallMem'
  label 'smallCpu'
  publishDir "${params.outputDir}/standardUnixCommand", mode: 'copy'

  input:
  file hello from helloWorldOutputCh

  output:
  file "bonjourMonde.txt"

  script:
  """
  sed -e 's/Hello World/Bonjour Monde/g' ${hello} > bonjourMonde.txt
  """
}

/*
 * process with onlylinux (invoke script from bin/ directory)
 */
process execBinScript {
  label 'onlyLinux'
  label 'smallMem'
  label 'smallCpu'
  publishDir "${params.outputDir}/execBinScript", mode: 'copy'

  output:
  file "execBinScriptResults_*"

  script:
  """
  apMyscript.sh > execBinScriptResults_1.txt
  someScript.sh > execBinScriptResults_2.txt
  """
}

/*
```

BIOCONDA

CONDA

</> source code

DEF DEF DEF

software1 software2 software3

source of the pipeline	availability of the software	generate the recipes	build the container

- Best practises to write the different process and nextflow code and organize the source code repository
- In-house toolbox to automatize container generation and deployement of the pipeline
- The toolbox is as set of CMake scripts

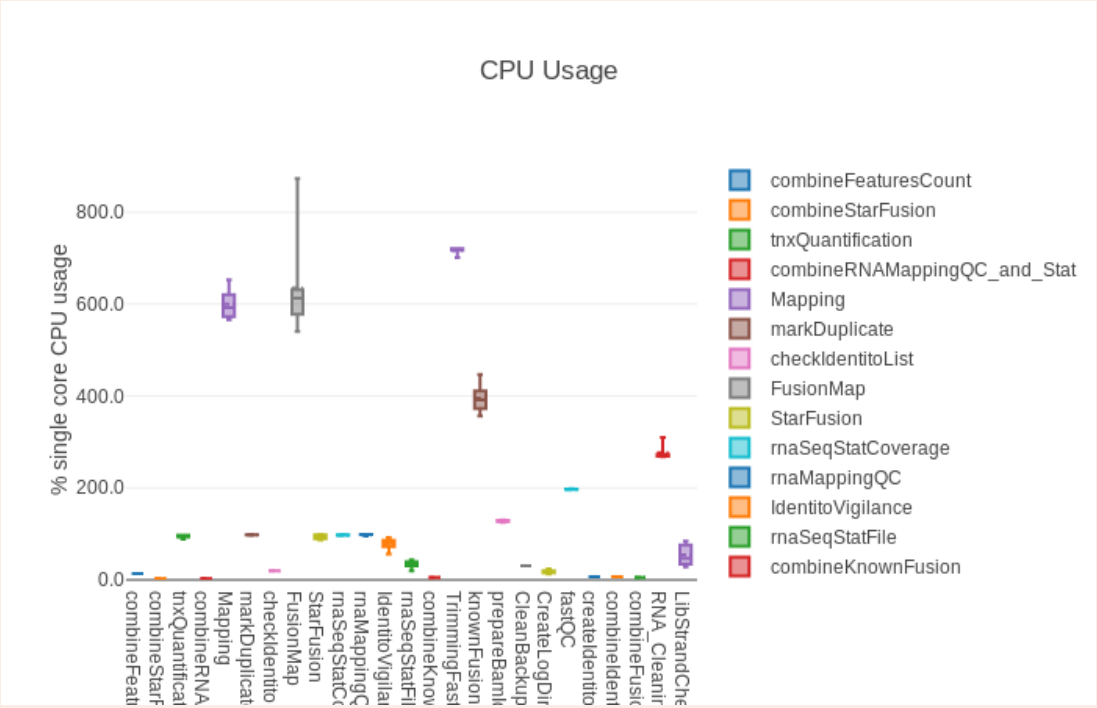
Reduce time to delivery and optimization of HPC resource usage

Ever-growing size and complexity of the data

Need to identify the bottlenecks (CPU, IO, memory, job duration) for each task of bioinformatics pipelines.

Code profiling of the pipeline with nextflow[1] (Workflow Management System)

Bioinformatics pipeline to detect gene fusions from RNA-seq for diagnostic (walltime ∼24H)



[1] Di Tommaso et al., 2017

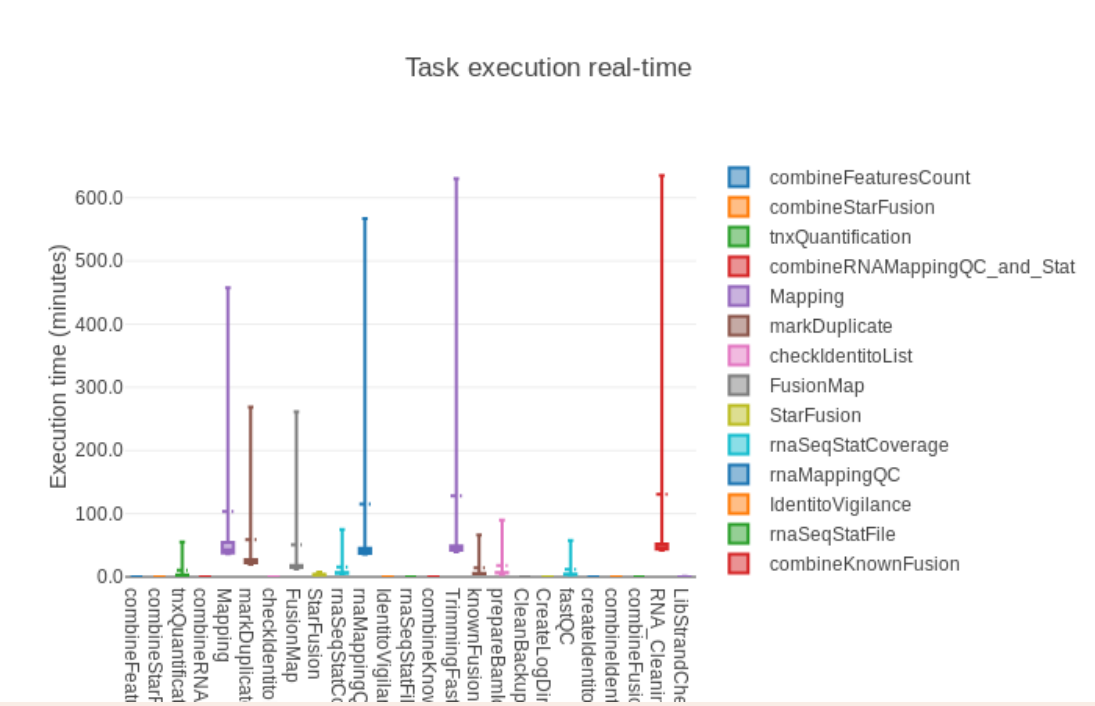
Reduce time to delivery and optimization of HPC resource usage

Ever-growing size and complexity of the data

Need to identify the bottlenecks (CPU, IO, memory, job duration) for each task of bioinformatics pipelines.

Code profiling of the pipeline with nextflow[1] (Workflow Management System)

Bioinformatics pipeline to detect gene fusions from RNA-seq for diagnostic (walltime ∼24H)



[1] Di Tommaso et al., 2017

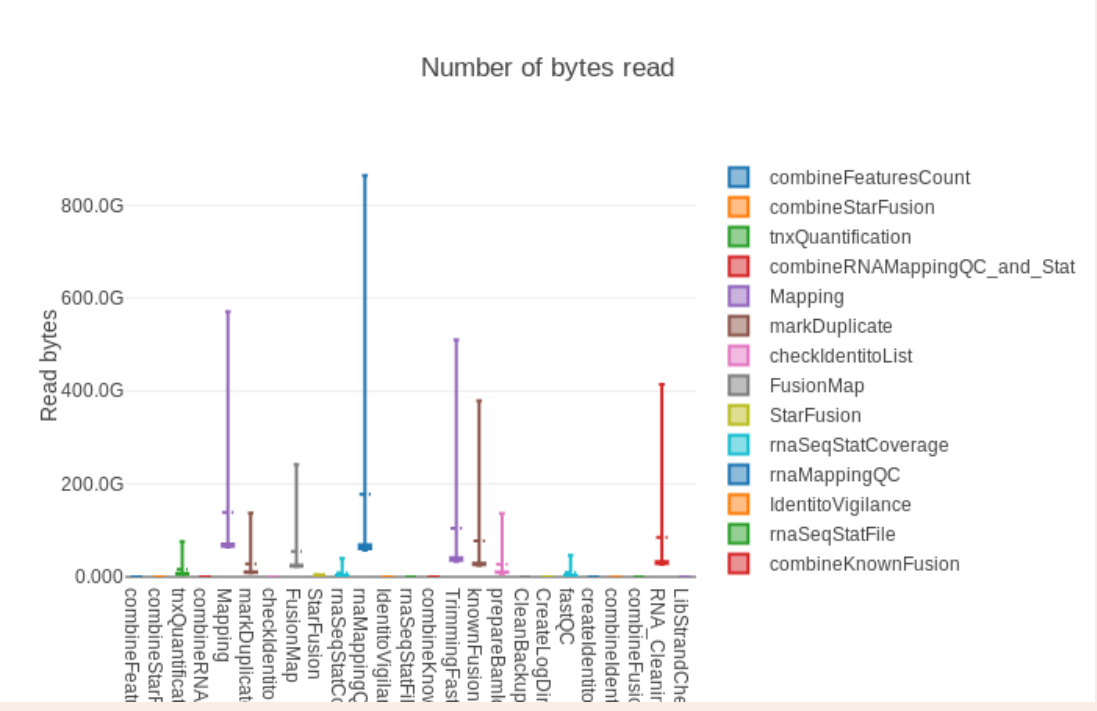
Reduce time to delivery and optimization of HPC resource usage

Ever-growing size and complexity of the data

Need to identify the bottlenecks (CPU, IO, memory, job duration) for each task of bioinformatics pipelines.

Code profiling of the pipeline with nextflow[1] (Workflow Management System)

Bioinformatics pipeline to detect gene fusions from RNA-seq for diagnostic (walltime ∼24H)



[1] Di Tommaso et al., 2017

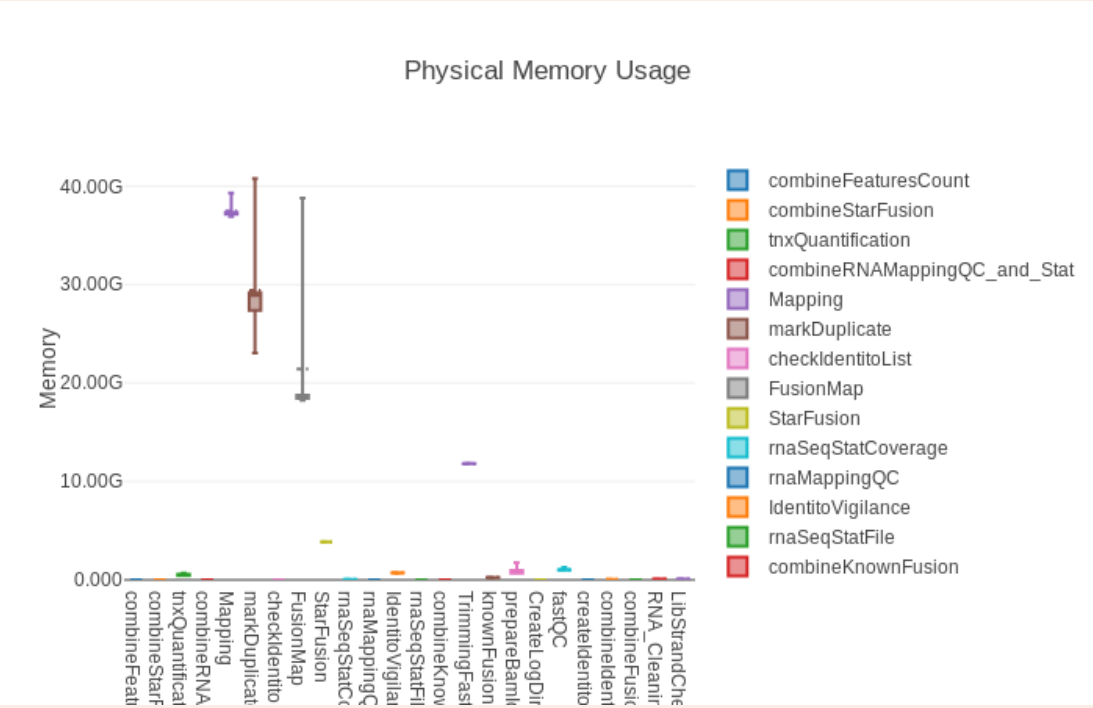
Reduce time to delivery and optimization of HPC resource usage

Ever-growing size and complexity of the data

Need to identify the bottlenecks (CPU, IO, memory, job duration) for each task of bioinformatics pipelines.

Code profiling of the pipeline with nextflow[1] (Workflow Management System)

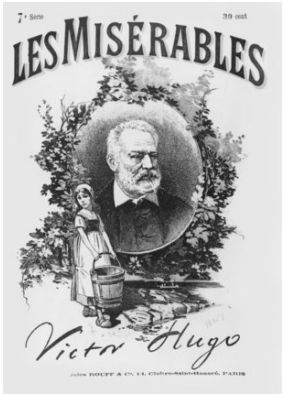
Bioinformatics pipeline to detect gene fusions from RNA-seq for diagnostic (walltime ∼24H)



[1] Di Tommaso et al., 2017

Outline

High-Throughput Sequencing... some figures...

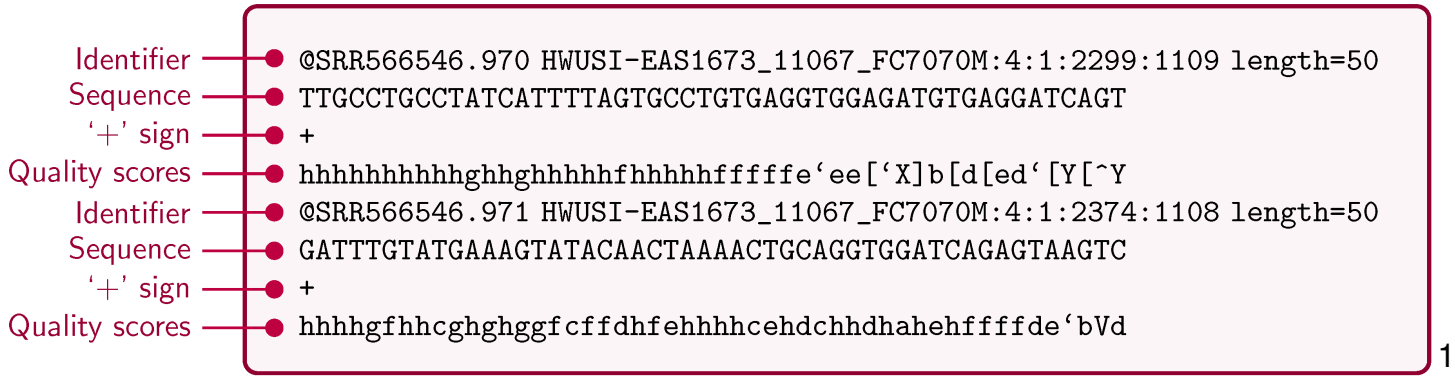


Sequencing with Illumina NovaSeq 6000

- 2 days of experiment
- 6000 billions of bases sequenced
- 10 billions of reads
- 2 millions times **Les Misérables**
- $\sim$ 4,500 GB of data

File format for sequencing data

FASTQ file



[1]

[1] Hosseini et al., 2016, [2] Science February 11th, 2011

File format for sequencing data

FASTQ file

Identifier	@SRR566546.970 HWUSI-EAS1673_11067_FC7070M:4:1:2299:1109 length=50
Sequence	TTGCCTGCCTATCATTTTAGTGCCTGTGAGGTGGAGATGTGAGGATCAGT
'+' sign	+
Quality scores	hhhhhhhhhhghhghhhhhfhhhhhfffffe'ee['X]b[d[ed'[Y[^Y
Identifier	@SRR566546.971 HWUSI-EAS1673_11067_FC7070M:4:1:2374:1108 length=50
Sequence	GATTTGTATGAAAGTATACAACTAAAACTGCAGGTGGATCAGAGTAAGTC
'+' sign	+
Quality scores	hhhhgfhhcghghggfcffdhfehhhhcehdchhdhahehffffde'bVd

[1]

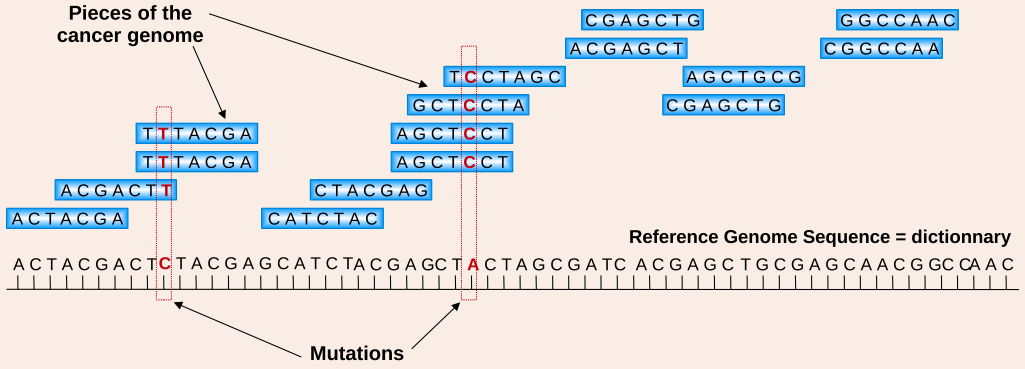
[2]

[1] Hosseini et al., 2016, [2] Science February 11th, 2011

MPI software to analyze sequencing data

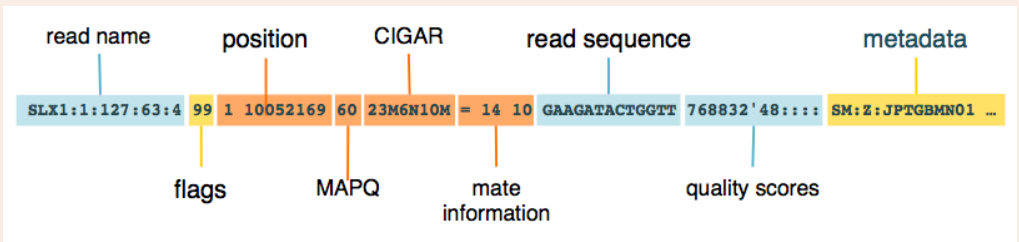
developed by Frédéric Jarlier

Alignment on the human reference genome



code: https://github.com/bioinfo-pf-curie/mpiBWA

Sort the SAM file by position along the genome



[1]

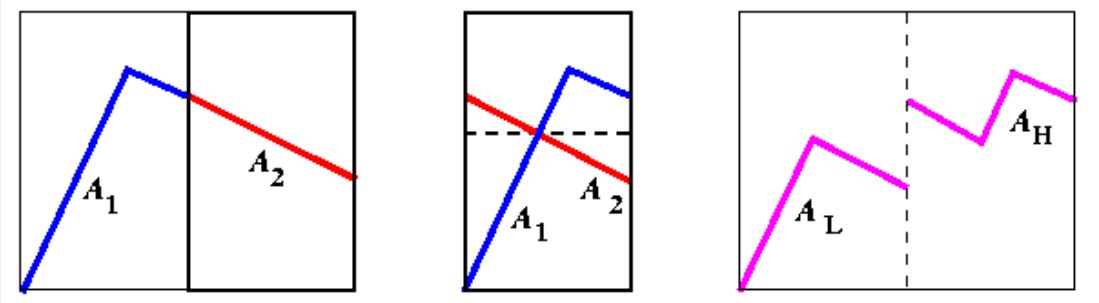
For deep sequencing, the size of the SAM file can be $\sim$ 1TB with $\sim 3 \times 10^9$ lines.

code: https://github.com/bioinfo-pf-curie/mpiSORT

[1] https://gatkforums.broadinstitute.org/gatk/

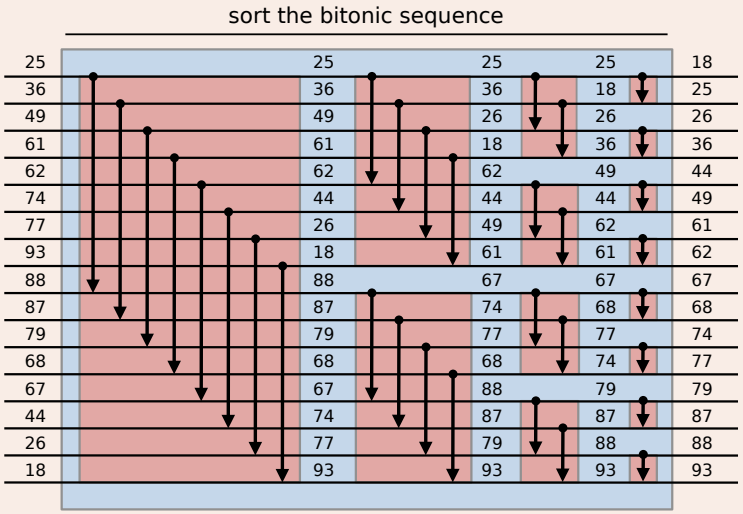
Bitonic sort of $n = 2^k$ elements[1]

Bitonic sequence



2

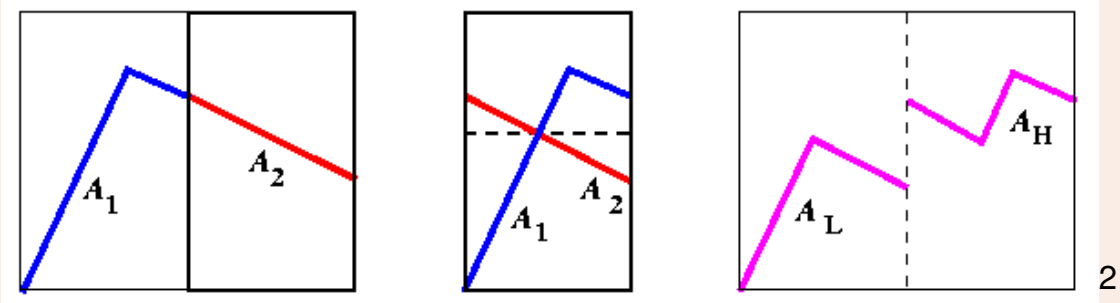
Bitonic sorting network (does not depend on the input data)



[1] Batcher, 1968 [2] http://pages.cs.wisc.edu

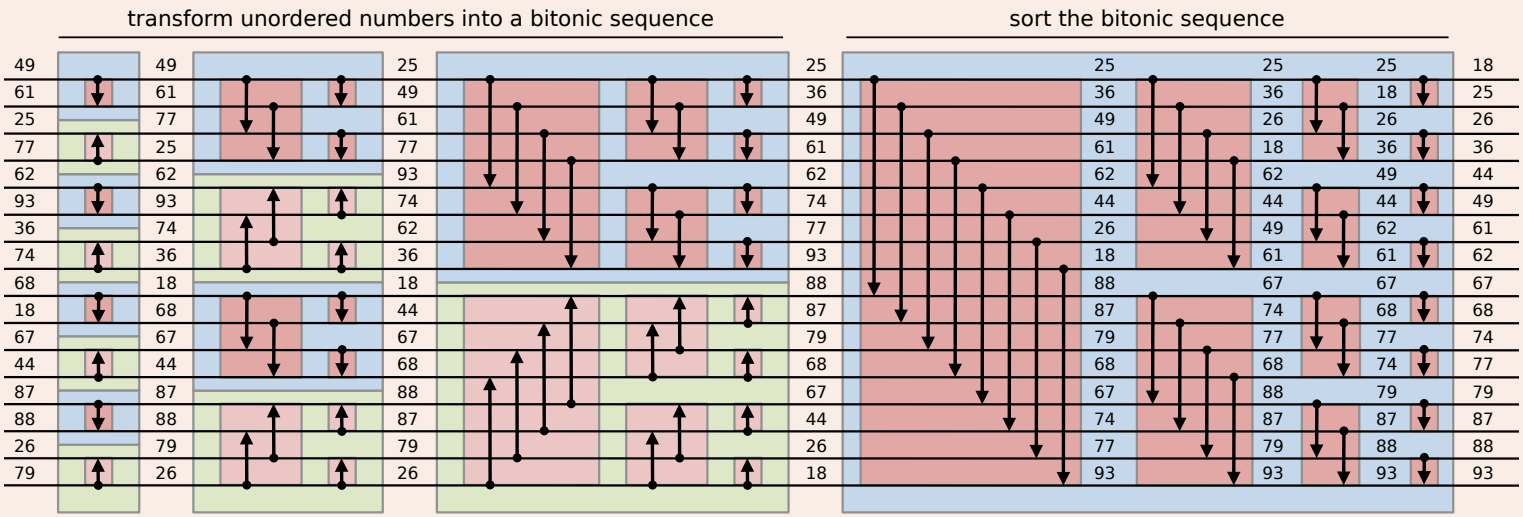
Bitonic sort of $n = 2^k$ elements[1]

Bitonic sequence



2

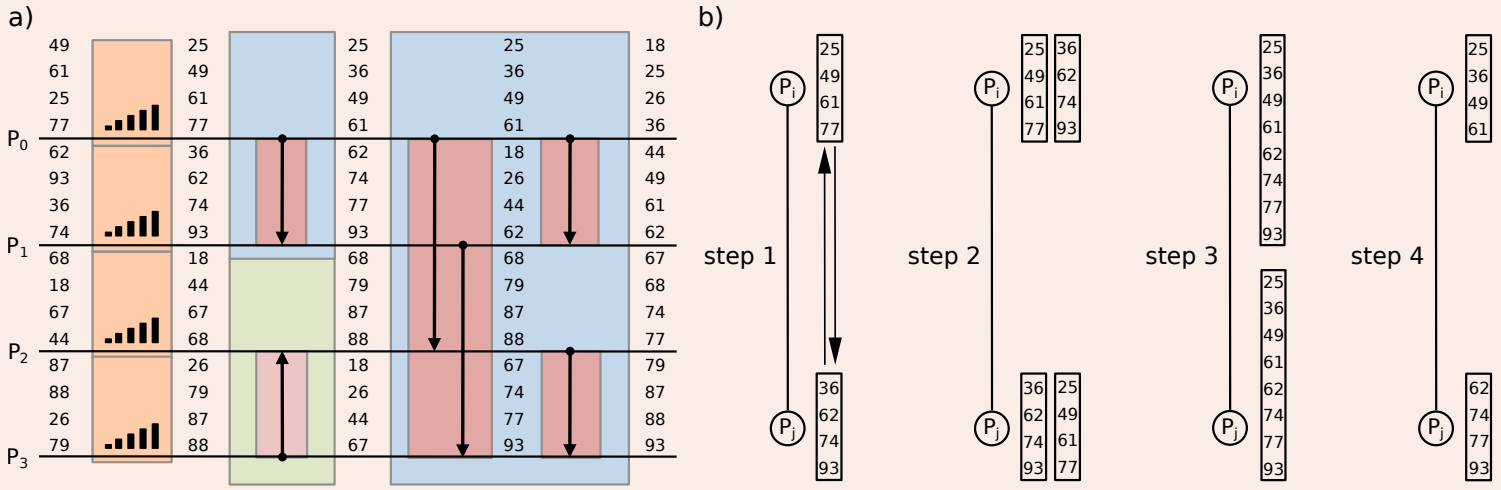
Bitonic sorting network (does not depend on the input data)



[1] Batcher, 1968 [2] http://pages.cs.wisc.edu

Parallel implementation of the bitonic sort

The data are split into blocks attached to the different processors



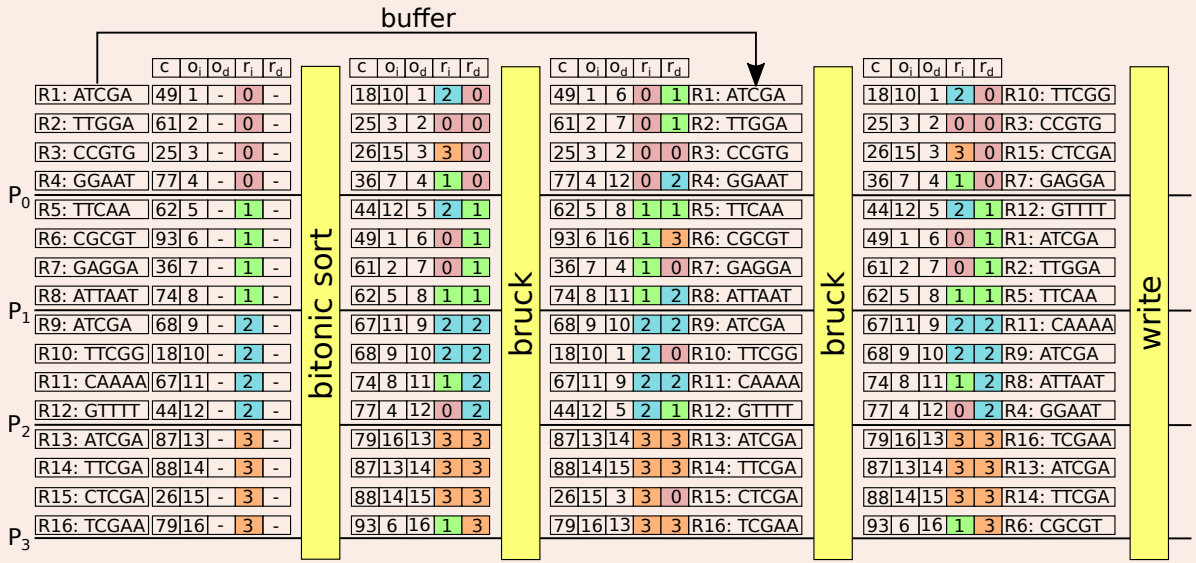
- Start with a merge sort for each block performed by each processor
- Use compare-split operator over the bitonic sorting network

All-to-all communications

Reorganize the data before writing the sorted SAM file

Bruck algorithm[1] exchanges data between $p = 2^m$ processors from different nodes

- Bruck phase is like a joint between two tables (the elements of the table located on different processors)
- Each processor writes the data from a block of contiguous offsets O_d in the sorted destination file
- Avoid random access during writing for better IO performance

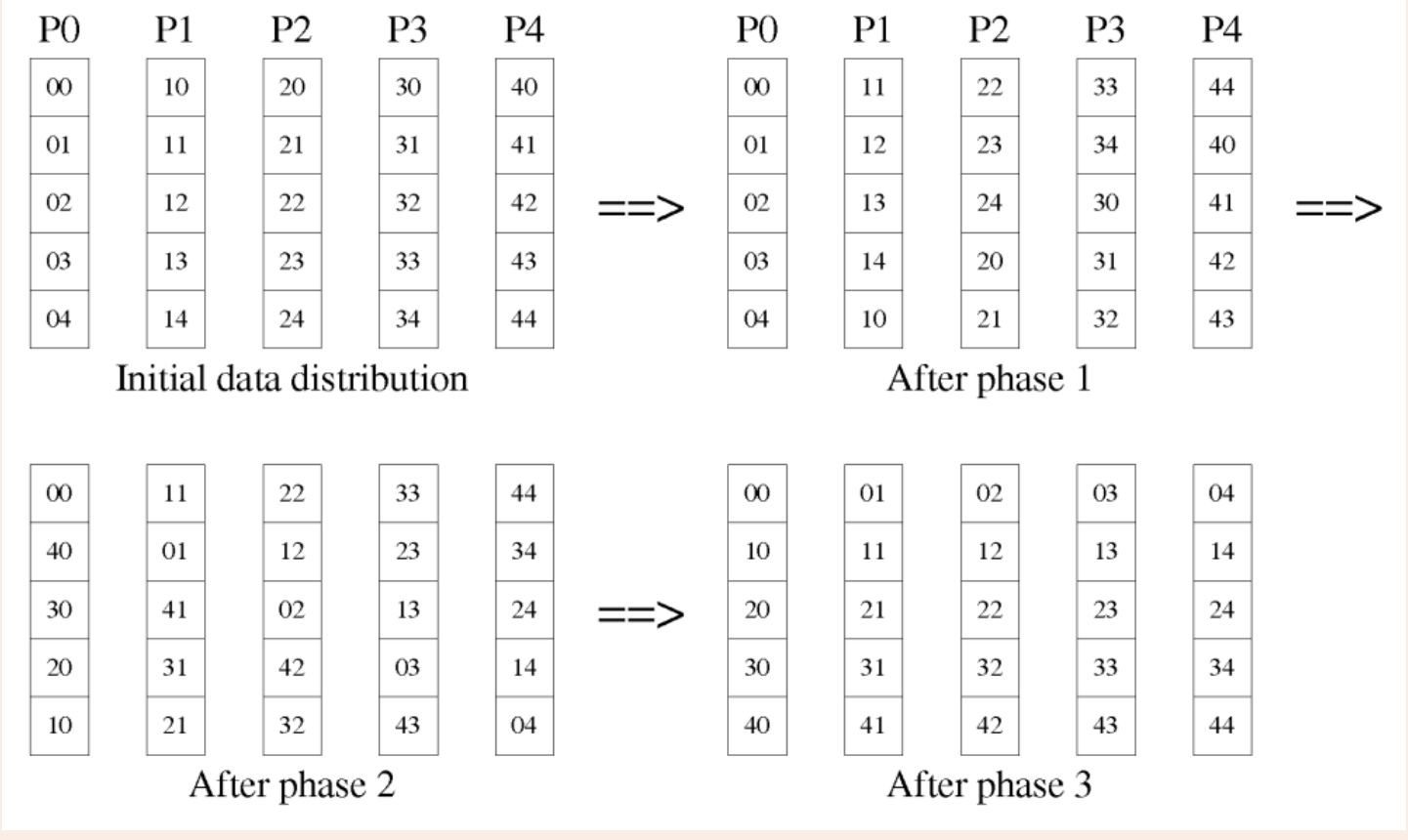


- C is the coordinate on the genome
- O_i the offset of the line from the input SAM file, O_d for the sorted destination SAM file
- R_i is the rank of the proc. that reads the block of the input SAM file
- R_d is the rank of the processor that will write the block of the sorted data in the destination SAM file

[1] Bruck et al., 1997

All-to-all communications

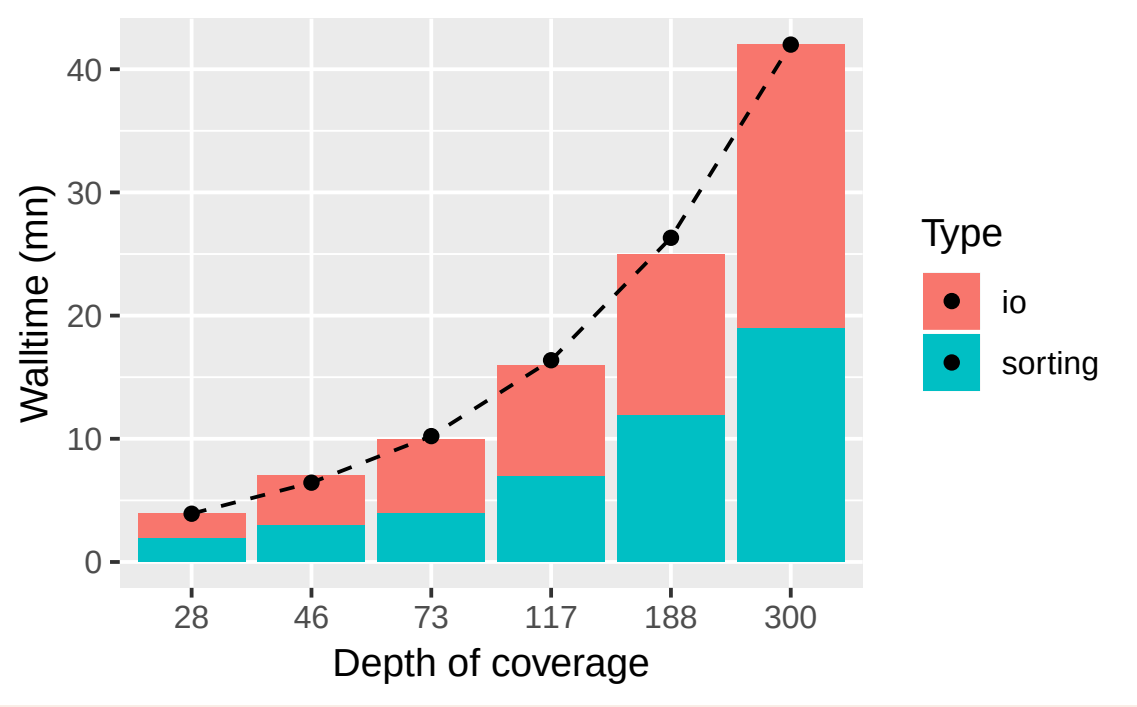
Bruck algorithm[1] principle



[1] Bruck et al., 1997

Performance on the sorting of SAM file with mpiSORT[1]

Walltime on Skylake architecture: 128 cpus on 4 nodes on sample HG001[2]



- Efficient implementation (standard implementation process the data within $\sim$ day)
- Ensure the scalability of the sorting as the size of the data increases
- Tweaking the code using MPI is not straigthforward and takes time

[1] https://github.com/bioinfo-pf-curie/mpiSORT, [2] GIAB HG001 300X sample

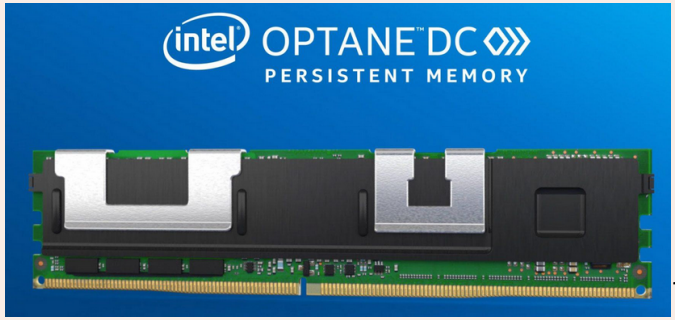
Partnership with intel and GENCI (CINES supercomputer)

mpiSORT is memory bounded

- The data have to fit into the memory
- More cores (and nodes) are required as the input data gets bigger and bigger
- Memory can be a bottleneck

Assess different data placement strategies using different technologies

- Intel Optane DC persistent memory up to 3 TB per CPU socket (in addition to the DRAM in the system)



[1]

[1] https://www.intel.com

Bibliography I

Arango, C., Dernat, R., and Sanabria, J. “Performance Evaluation of Container-based Virtualization for High Performance Computing Environments”. (2017). arXiv: http://arxiv.org/abs/1709.10140v1 [cs.OS].

Batcher, K. E. “Sorting networks and their applications”. *Proceedings of the April 30–May 2, 1968, spring joint computer conference*. 1968, pp. 307–314.

Bruck, J. et al. “Efficient algorithms for all-to-all communications in multiport message-passing systems”. *Parallel and Distributed Systems, IEEE Transactions on (Volume:8 , Issue: 11)* (1997).

Di Tommaso, P. et al. “Nextflow enables reproducible computational workflows.”. *Nature biotechnology* 35 (4 2017), pp. 316–319.

Hosseini, M., Pratas, D., and Pinho, A. “A survey on data compression methods for biological sequences”. *Information* 7.4 (2016), p. 56.

Thank you!

Institut Curie Bioinformatics core facility

Marie Curie

Dans une bonne équipe chacun fait plus que son métier.