Utilisation des ressources CIMENT dans le cadre du projet epimed:

Les besoins spécifiques de la bioinformatique pour l'analyse des données d'exome.

Quentin Testard, Julien Thevenon, Laure Raymond, Jean-François Taly

25/10/18
My thesis objectives

- **Industrialized** processes for exome analysis ⇒ **Workflow**

- **Speed up** / use **novels approaches** for exome analysis ⇒ **Containers**

- **Master** pipeline execution

- **Control** pipeline possible failures ⇒ **Pipeline certification**

- **Will be used in routine health diagnosis activity**
DNA sequencing

- Human Genome Project: 13 years, > 3 billion $
- James Watson Genome: 2 months, 2 million $
- Individual Genome: 15 days, 3000-5000 $
- Individual Genome: 3 days, 1000 $
- Individual Genome: 1h, 100 $

2003 2007 2013 2015 2025
Bioinformatics analysis

Sample collection and experimental design

Sequencing

Data management
- Raw reads (FASTA, FASTQ)
- Mapped reads (BAM, CRAM, MRQ)
- Data reduction
- High-level summaries (VCF, Peaks, RPMK)

Downstream analyses
(differential expression, novel TARs, regulatory networks, ...)

Time

Pre-NGS (Approximately 2000)
Now (Approximately 2010)
Future (Approximately 2020)
The Epimed initiative

Medical Epigenetics and Bioinformatics

• Objectives:

Facilitate a translational research in epigenetics, between the fundamental research teams and the medical teams.

Analyze large-scale whole-genome data that is essential to understand the epigenome.

Organize an interactive database associating "omics" data with biological and clinical data.

Community

Collaborative network of about 50 people:
• IAB, CHU, TIMC-IMAG, LJK, CEA
• International collaborations

Sophie Rousseaux (DR)
epigenetics, medical research

Anne-Laure Vitte (IE)
molecular biology

Florent Chuffart (IR)
statistics, computing

Ekaterina Flin (IR)
databases, web systems
The Epimed infrastructure

Web application
epimed.imag.fr/database

High Performance Computing

A dedicated node in Luke cluster

Database
epimed-db.imag.fr

Data storage
SUMMER
Exome and rare Mendelian disorders

- Mostly genetic related
- Mostly monogenic/mendelian diseases
- Quite difficult to diagnose:
  - Often long to diagnose => 25 % between 5 to 25 years with traditional methods
- Severe:
  - Child mortality rates around 30% under 5 y/o
- Numerous:
  - Around 7000 different diseases
  - 7-8% of total worldwide population (around 30M people in Europe)
- Exome analysis:
  - Compare sample DNA material to a human « reference »
  - Find discordance between both
  - Relate difference to patient clinical features (phenotype)
  - Resolve 40% of undiagnosed case
Bioinformatics analysis

Raw Unmapped Reads
- uBAM or FASTQ
  - Map to Reference
  - Raw Mapped Reads
    - BAM
  - Mark Duplicates
  - Recalibrate Base Quality Scores
  - Analysis-Ready Reads
    - BAM

Analysis-Ready Reads
- BAM
  - Call Variants Per-Sample
    - HaplotypeCaller in GVCF mode
    - GVCF
    - SNPs
    - Indels
  - Consolidate GVCFs
  - Joint-Call Cohort
    - GenotypeGVCFs
    - Raw SNPs + Indels
      - VCF

Raw SNPs + Indels
- VCF
  - Filter Variants
  - Refine Genotypes
  - Annotate Variants
  - Evaluate Callset
    - Troubleshoot
    - Use in project

From FastQ data to high confidence variant calls: the Genome Analysis Toolkit best practices pipeline.

Van der Auwera GA1, Camargo MG1, Hartl C1, Pakin H1, Del Angel O1, Levy-Moonshine A1, Jordan T1, Sharter H1, Rozen D1, Thibault J1, Banks E1, Gambetta D2, Atkinson D1, Garcia S1, DelFralco MA1

https://software.broadinstitute.org/gatk/
## « Required » ressources

- **Cluster Dahu CIMENT :**
  - 72 Dell C6420 (2304 cores)
  - 2x Intel Skylake Gold 6130
  - 192 GB RAM
  - SSD 240GB + SSD 446GB + HDD 4TB

<table>
<thead>
<tr>
<th>Analysis step</th>
<th>Input</th>
<th>Time</th>
<th>RAM</th>
<th>CPU</th>
<th>Size output</th>
<th>Read disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapping</td>
<td>$\approx 2 \times 4$ - 5 Gb</td>
<td>$\approx 50 - 60$ min</td>
<td>$\approx 24$ Gb</td>
<td>$\approx 16000%$</td>
<td>$\approx 8 - 10$ Gb</td>
<td>$\approx 30 - 35$ Gb</td>
</tr>
<tr>
<td>Mark Duplicates</td>
<td>$\approx 8 - 10$ Gb</td>
<td>$\approx 20 - 25$ min</td>
<td>$\approx 36 - 38$ Gb</td>
<td>$\approx 10000%$</td>
<td>$\approx 10 - 12$ Gb</td>
<td>$\approx 20 - 21$ Gb</td>
</tr>
<tr>
<td>Base Recalibration 1</td>
<td>$\approx 10 - 12$ Gb</td>
<td>$\approx 55 - 60$ min</td>
<td>$\approx 36 - 38$ Gb</td>
<td>$\approx 10000%$</td>
<td>$\approx 18 - 20$ Gb</td>
<td>$\approx 45 - 50$ Gb</td>
</tr>
<tr>
<td>Base Recalibration 2</td>
<td>$\approx 18 - 20$ Gb</td>
<td>$\approx 60 - 70$ min</td>
<td>$\approx 36 - 38$ Gb</td>
<td>$\approx 10000%$</td>
<td>$\approx 18 - 20$ Gb</td>
<td>$\approx 55 - 65$ Gb</td>
</tr>
<tr>
<td>Haplotypecaller</td>
<td>$\approx 18 - 20$ Gb</td>
<td>$\approx 30 - 40$ min</td>
<td>$\approx 36 - 38$ Gb</td>
<td>$\approx 10000%$</td>
<td>$\approx 50 - 90$ MB</td>
<td>$\approx 30 - 40$ Gb</td>
</tr>
<tr>
<td>CombineGVCFs</td>
<td>$X \times 50 - 90$ Mb + $0,5 - 3$ Gb</td>
<td>$\approx 10 - 40$ min</td>
<td>$\approx 36 - 38$ Gb</td>
<td>$\approx 10000%$</td>
<td>$24 \times 0,5 - 3$ Gb</td>
<td>$\approx 10 - 40$ Gb</td>
</tr>
<tr>
<td>GenotypeGVCFs</td>
<td>$24 \times 0,5 - 3$ Gb</td>
<td>$\approx 1 - 30$ min</td>
<td>$\approx 36 - 38$ Gb</td>
<td>$\approx 10000%$</td>
<td>$X \times 4 - 5$ Mb</td>
<td>$\approx 1 - 12$ Gb</td>
</tr>
<tr>
<td>Annotation</td>
<td>$X \times 4 - 5$ Mb</td>
<td>$\approx 18 - 20$ min</td>
<td>$\approx 48 - 50$ Gb</td>
<td>$\approx 8000%$</td>
<td>$X \times 30 - 40$ Mb</td>
<td>$\approx 14 - 20$ Gb</td>
</tr>
</tbody>
</table>
Used tools

**Nextflow**

Nextflow enables reproducible computational workflows.

Di Tommaso P, Chatzou M, Floden EI, Barja PP, Palumbo E, Notredame C.

**GATK**

The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data.

process fastqc {
  publishDir "${params.resultDir}/fastqc", mode: 'copy'
  input:
  file fastq from fastq_list
  output:
  file '*.zip' into fastq_out_zip
  file '*.html' into fastq_out_html
  val "${params.resultDir}" into fastqc_rep
  "fastqc $fastq"
}

$fastqc {container = "${params.singularityDir}/fastqc-0.11.15.img"}

- [https://www.nextflow.io/](https://www.nextflow.io/)
- Dataflow manager
- « Crash » management and « resume » fonction
- Automatic pipeline launching
- « Compatible » with OAR, CIMENT batch scheduler
- Summary reports

process fastqc {
  publishDir "${params.resultDir}/fastqc", mode: 'copy'
  input:
  file fastq from fastq_list
  output:
  file '*.zip' into fastq_out_zip
  file '*.html' into fastq_out_html
  val "${params.resultDir}" into fastqc_rep
  "fastqc $fastq"
  ...
  fastqc $fastq
  ...
}

$fastqc {container = "${params.singularityDir}/fastqc-0.11.15.img"}
My « future » pipeline

- **Robust** (managed to process 387 samples during a single week-end)

- Every pipeline version will be « Vmized » with all its dependencies (reproductibility for each version)

- **Fully fonctionnal** and **documented exome version** coming **Q1 2019**

- **Coming next, genome version !**
And soon … The Human Genome!

- We need **novel approaches to speed up** the analysis:
  - **Splitting** operation per **logical unit** (chromosomes):
    => **Reformatting** my code
  
  - Speeding up by **massively parallelizing processes**:
    
    => **GPU programming**
    
    => **MapReduce Spark**

**Tools already exist!**
GPU

The Parabricks Advantage

- **Equivalent Result**: Parabricks’ pipeline generates results equivalent to the baseline GATK 4 Best practices pipeline (using the same algorithm).
- **50 Genomes Per Day**: On a single server, Parabricks’ software provides high throughput and can process up to 50 whole genomes per day.
- **30-50X faster**: Parabricks’ software can run on local GPU servers, on any major public or private cloud with access to GPUs.
- **Deterministic Reproducible**: By running 30-50 times faster, Parabricks can reduce computing costs up to 50% compared to GPU-only solutions.
- **On-prem & Cloud**: Any configuration of Parabricks software on any platform with any number of resources generates the exact same results every execution.
- **Up-To-Date**: Parabricks’ accelerated software supports multiple versions of BWA-Mem, Picard and GATK and will support all future versions of these tools.

Performance Comparison

- **Benchmark**
  - Baseline: 32CPUs (BASELINE)
  - DGX-1: 53
  - AWS: 100
  - GOOGLE: 115
  - AZURE: 118

- **GPU Utilization**
  - 4 GPU: 60
  - 8 GPU: 35

- **Accuracy**
  - BAM: 100%
  - VCF: 99.99%
Native Spark tools are included into GATK4 for mapping and variant calling (two of the most parallelizable steps)
Why Spark?

- Permit to **massively parallelize** processes

- Spark Bioinformatics tools « cut » **biological data** into **chunks** and do « **bioinformatic** » **operations** on it
GATK4 Spark benchmarking

- Toulouse LIPM Cluster:
  - 8 physical nodes
  - 252 cores
  - 642 Go RAM memory
  - 34T HDD

- One supplementary frontal node to distribute Spark jobs

- The same cluster have been used for Spark and SGE jobs, but not at the same time
Spark benchmarking

- The **GATK4 non-Spark tools** are poorly parallelized
- The **GATK4 Spark tools** are still in Beta
- There is a significative speed improvement with Spark tools

Genome: Plasmopara halstedii (75 Mbases), fastq.gz: 2x12Go, 3 repeats

Source: Axel Verdier INRA UMR LIPM Toulouse
Conclusion

• Currently **CIMENT luke/Dahu** HPC nodes are **sufficient** for our **exome analyses**

• But for **high-scale genome analysis** we will need to **step up**:
  - Produce a **better code**
  - Use **bigger HPC infrastructures**
  - Use **Spark** or **GPU programmed tools and infrastructures**

• It’s not that hard! **Tools already exist** and are indeed **speeding up genome analyses**

• We need to **proceed with caution**. We are working with **health datasets**. **No mistake allowed**.
Thank you!