New solutions for Big Data Analysis and Visualization

From *HPC* to *cloud*-based solutions

Barcelona, February 2013

Nacho Medina
imedina@cipf.es
http://bioinfo.cipf.es/imedina
Head of the Computational Biology Unit
Computational Medicine Institute
Centro de Investigación Príncipe Felipe (CIPF)
(Valencia, Spain)
Introduction
Some results
Next steps
Conclusions
Next-Generation Sequencing (NGS) technology is changing the way how researchers perform experiments. Many new experiments are being conducted by sequencing: exome re-sequencing, RNA-seq, Meth-seq, ChIP-seq, ... 

NGS is allowing researches to: 
- Find exome and genomic variants responsible of diseases
- Study the whole transcriptome of a phenotype
- Establish the methylation state of a condition
- Locate DNA binding proteins

But experiments have increased data size by 1000x when compared with microarrays, i.e. from MB to hundreds of GB in transcriptomics

Data processing and analysis are becoming a bottleneck and a nightmare, from days or weeks with microarrays to months with NGS, and it will be worse as more data become available
Introduction

**Big data challenges and solutions**

- “Big data is a collection of data sets *so large and complex* that it becomes difficult to process using on-hand database management tools or traditional data processing applications”

- Big data is *not a new scenario* for other science areas: meteorology, physics, internet search, finance, business, ...

- Which are the *main Big data challenges*?: curation, search, sharing, storage, analysis and visualization

- We need to study and use *new computational technologies* available:
  - High-Performance Computing (HPC): multi-core CPUs, SSE/AVX, GPUs
  - Distributed computing: Apache Hadoop MapReduce, MPI
  - Distributed and NoSQL databases: Apache Cassandra, HBase, ...
  - Web apps: HTML5 (SVG, WebGL, ...), Javascript, RESTful WS, ...
  - Clouds: Amazon AWS, Google Cloud, Microsoft Azure, ...
  - Science: Machine learning, data mining, clustering, probabilistic graphicals models, visualization, ...
Introduction
Brief presentation of Computational Biology Unit

Part of Institute of Computational Medicine, http://bioinfo.cipf.es

http://bioinfo.cipf.es/compbio

---

**Joaquin Dopazo's Group**: long experience in genomic data analysis for more than 10 years

- Many methodologies and applications developed: GEPAS, Babelomics (FatiGO, SNOW,...), NetworkMiner, ...
- More than 10 NARs and many papers with thousands of users, more than 250 jobs executed daily
- Today computational requirements in biology makes necessary more advanced computing solutions, so the Computational Computing Unit has been created: Genomic Variants analysis, NGS data analysis, Computational Systems biology, Databases. But also HPC computing, Machine Learning, cloud-based solutions, ...

---

*Big data* in biology. New software and solutions are needed. Efficiency matters.

This specialized unit consists of Computer Scientists such as Scientific programmers, HPC developers and Web and Distributed developers to produce efficient software
**Introduction**

**Motivation and goals**

- Poor performance, **Software** in Bioinformatics is in general slow and not designed for running in clusters or clouds like in other science fields. NGS push computer needs, new solutions and technologies are needed: *more analysis, performance, scalable, cloud*

- As data analysis group we focus in **analysis**: *It's analysis, stupid!*
  - ~1-2 weeks sample **sequencing**, ~2-4 weeks **preprocessing and read mapping**, this is *necessary to be improved but is not enough!*
  - **Several months of data analysis**, lack of software for biological analysis, i.e. which are the genes with a coverage >30x that are differentially expressed in a dataset of 50 BAMs accounting for 500GB of data with a p-value<0.05?

- **Goals**: to develop new generation of software in bioinformatics to be
  - **Very fast** software with a **low memory** footprint and **integrated analysis toolkit**
  - be able to handle and analyze TB of data, store data **efficiently** to be queried
  - distribute computation, no data. Make use of **cloud computing**

- Software **must** exploit current hardware and be **fast and efficient** in a **standard workstation, HPC cluster or in a cloud environment**
Some results
NGS pipeline, a HPC implementation

NGS sequencer

Fastq file, up to hundreds of GB per run

QC and preprocessing

HPG Aligner, short read aligner

SAM/BAM file

QC and preprocessing

Variant calling analysis

VCF file

Variant VCF viewer

HTML5+SVG Web based viewer

HPG Variant, Variant analysis

Consequence type, GWAS, regulatory variants and system biology information

GATK and SAM mPileup HPC Implementation. Statistics genomic tests

Other analysis (HPC4Genomics consortium)

RNA-seq (mRNA sequenced)
DNA assembly (not a real analysis)
Meth-seq
Copy Number
Transcript isoform
...

More info at:
http://bioinfo.cipf.es/docs/compbio/projects/hpg/doku.php

HPG suite
High-Performance Genomics

Double mapping strategy:
Burrows-Wheeler Transform (GPU Nvidia CUDA) + Smith-Waterman (CPU OpenMP+SSE/AVX)
Some results

**HPG Aligner**, the first HPC DNA and RNA-seq aligner

- Current read aligners software tend to fit in one of these groups:
  - **Very fast, but no too sensitive**: no gaps, no indels, rna-seq...
  - **Slow, but very sensitive**: up to 1 day by sample
- Current aligners show **bad performance** with long reads
- Current read Aligner algorithms
  - **Burrows-Wheeler Transform (BWT)**: very fast! No sensitive
  - **Smith-Waterman (SW)**: very sensitive but very slow
- Hybrid approach (papers in preparation):
  - **HPG-BWT** implemented with OpenMP and Nvidia CUDA
  - **HPG-SW** implemented using OpenMP and SSE (~26x in 8-core)

![Graphs showing alignment time for BWT and SW]

- Fastq file
- Architecture:
  - **HPG-BWT** OpenMP+GPU
  - Find CALs with **HPG-BWT** (seeds)
  - **HPG-SW** OpenMP+SSE
  - SAM file

- Testing BWT-GPU

- 10x compared to BFAST
- ~50x with SSE
First results show an amazing **performance** and the best **sensitivity**

### DNA 2M simulated datasets

<table>
<thead>
<tr>
<th>Program</th>
<th>100nt %mapped</th>
<th>150nt %mapped</th>
<th>250nt %mapped</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPG Aligner <strong>dna mode</strong></td>
<td>96.22%</td>
<td>96.98%</td>
<td>97.83%</td>
</tr>
<tr>
<td></td>
<td>1.26min</td>
<td>1.98min</td>
<td>3.7min</td>
</tr>
<tr>
<td>BWA 0.6.2</td>
<td>93.58%</td>
<td>92.6%</td>
<td>98.0%</td>
</tr>
<tr>
<td></td>
<td>4.3min</td>
<td>6.3min</td>
<td>9.7min</td>
</tr>
<tr>
<td>Bowtie 0.12.8</td>
<td>79.95%</td>
<td>60.11%</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1.8min</td>
<td>2.42min</td>
<td></td>
</tr>
<tr>
<td>Bowtie2 2.0.0</td>
<td>94.71%</td>
<td>96.75%</td>
<td>98%</td>
</tr>
<tr>
<td></td>
<td>2.48</td>
<td>3.25</td>
<td>5.75</td>
</tr>
</tbody>
</table>

### RNA-seq 1M simulated datasets

<table>
<thead>
<tr>
<th>Program</th>
<th>75nt %mapped</th>
<th>100nt %mapped</th>
<th>150nt %mapped</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPG Aligner <strong>rna mode</strong></td>
<td>96.5%</td>
<td>97.1%</td>
<td>97.7%</td>
</tr>
<tr>
<td></td>
<td>1.38min</td>
<td>2.41min</td>
<td>5.91min</td>
</tr>
<tr>
<td>TopHat 2.0.4</td>
<td>74.4%</td>
<td>62.1%</td>
<td>37.1%</td>
</tr>
<tr>
<td></td>
<td>15.6min</td>
<td>20.1min</td>
<td>36.2min</td>
</tr>
</tbody>
</table>

**Notes:**
- Max errors allowed
- Similar results are obtained with real datasets
- **TopHat** doubles disk space and big memory needs
- **10x faster!!**

### Notes:
- SE with 2M reads datasets, **5% random reads**
- Tests in a **12-core** machine, **no GPU used**
- **INDELS** supported
- GEM, Soap2 and BFAST outperformed
- Similar results are obtained with real datasets
- **Papers in preparation**
Some results

**HPG Aligner, RNA-seq results**

Comparative results with TopHat2 using both Bowtie and Bowtie2

**Hardware scalability tests**

![Graph a](image1)

![Graph b](image2)
Some results

**HPG Variant**, suite of tools for variant analysis

- **HPG Variant**, a suite of tools for HPC-based genomic variant analysis
  - VARIANT = **VAR**iant **AN**alysis **T**ool

- Three tools are already implemented: **vcf**, **gwas** and **effect**. Implemented using **OpenMP**, **Nvidia CUDA** and **MPI** for large clusters.

- **VCF**: C library and tool: allows to analyze large VCFs files with a low memory footprint: stats, filter, split, merge, ... (*paper in preparation*)
  - Example: `hpg-variant vcf --stats --vcf-file ceu.vcf`

- **GWAS**: suite of tools for gwas variant analysis (~**Plink**)  
  - association, TDT  
  - Hardy-Weinberg, LOH  
  - Epistasis  
  - Example: `hpg-variant gwas --tdt --vcf-file tumor.vcf`

- **EFFECT**: A CLI and web application, it's a cloud-based genomic variant **effect** predictor tool has been implemented ([http://variant.bioinfo.cipf.es](http://variant.bioinfo.cipf.es), *published in NAR*)
Based on **CellBase** *(published in NAR)*, a comprehensive integrative database and **RESTful Web Services API**, more than 250GB of data and 90 SQL tables exported in TXT and JSON:

- Core features: genes, transcripts, exons, cytobands, proteins (UniProt), ...
- Variation: dbSNP and Ensembl SNPs, HapMap, 1000Genomes, Cosmic, ...
- Functional: 40 OBO ontologies (Gene Ontology), Interpro, ...
- Regulatory: TFBS, miRNA targets, conserved regions, ...
- System biology: Interactome (IntAct), Reactome database, co-expressed genes, ...

- NoSQL based version coming, scale to TB

**Project:** [http://bioinfo.cipf.es/compbio/cellbase](http://bioinfo.cipf.es/compbio/cellbase)

**Wiki:** [http://docs.bioinfo.cipf.es/projects/cellbase/wiki](http://docs.bioinfo.cipf.es/projects/cellbase/wiki)
Some results

**Genome Maps**, a HTML5+SVG data visualization

- Genome scale data **visualization** is an important part of the data analysis: *heavy data issue! Do not move data!*

- **Features of Genome Maps** ([www.genomemaps.org](http://www.genomemaps.org), under review)
  - First 100% HTML5 web based: **HTML5+SVG** (Google Maps inspired)
  - Always updated, **no browser plugins or installation**, USA Amazon AWS
  - Data taken from **CellBase**, remote **NGS data**, local **files** and **DAS servers**: genes, transcripts, exons, SNPs, TFBS, miRNA targets, ...
  - Other features: Multi species, API oriented, integrable, plugin framework, ...

---

Project: [http://bioinfo.cipt.es/compbio/genomemaps](http://bioinfo.cipt.es/compbio/genomemaps)
Next steps

*cloud*-based and open solutions

- *cloud*-based environment integration ready, codename: GASC
  - **Storage**: efficient storage and data retrieval of ~TB, transparent connection to other clouds such as Amazon AWS or Microsoft Azure
  - **Analysis**: many tools ready to use (aligners, GATK, …), users can upload their tools to extend functionality, SGE queue, …
  - **Search and access**: data is indexed and can be queried efficiently, RESTful WS allows users to access data and analysis programatically
  - **Sharing**: users can share their data and analysis, public and private data
  - **Visualization**: HTML5-SVG based web applications to visualize data

- Open development initiative
  - HPG project, CellBase, Genome Maps, GASC, … *released as open source development initiative*
  - Source code controlled with Git, hosted freely in *GitHub*
  - Scientist are *encouraged* to collaborate and extend functionality, a HPC4G consortium from universities already created
High-throughput technologies such as NGS is pushing Biology into *Big Data*

Bioinformatics must learn how to deal with this huge amount of data

This new scenario demands new solutions, new computational technologies must be used

Open development model allows researchers to join forces and build up better solutions