An Update on
The Seal Hadoop-based Sequence Processing Toolbox

Luca Pireddu, Simone Leo, Gianluigi Zanetti

CRS4—Distributed Computing Group

July 20, 2013
Outline

1. Motivation

2. Introducing Seal

3. New additions

4. Conclusion
Regular advances to sequencing technologies are

- lowering sequencing costs
- increasing acquisition speed

Data production rate is growing exponentially

- E.g., 1 Illumina HT machine can now produce about 9 TB of raw data per month

Processing capacity is not growing this fast!
Growth of data

- In recent years there has been a steady increase in the amount of digitized data available
- Rise of data-driven businesses

Google apparently processed 24 PB/day in 2009
- That’s about 20000 Illumina run directories... per day!
- Relative to theirs, our problem doesn’t seem so big
How can do they do that?
Scaling

Change!

Adopt a new computational paradigm

- Scale horizontally, using lots of machines
- Write software that accepts and handles hardware failure
- Spread the data
  - split it into parts
  - distribute them on the processing nodes
- Move the computation to the data
Those ideas are already implemented in an open source solution

- Refactors distribution and robustness into a reusable framework
- Not a second-class citizen: this is the system used by Twitter, Facebook, Yahoo, LinkedIn, and others
- Maybe those processing a lot of sequencing data should try it...
### CRS4 Sequencing and Genotyping Platform
- Currently the largest sequencing center in Italy

**Sequencing Equipment:** 3 Illumina HiSeq2000, plus older sequencers

**Sequencing Capacity:** about 5 Tbases/month

### Since Sept. 2010 we’ve sequenced about...
- over 2000 whole-genome samples (mostly low-pass, some high-coverage)
- 800 RNA samples
- 100 exomes
- a handful ($\approx 30$) of ChIP-Seq samples
We needed to scale

Decided to trying doing so with Hadoop

But...

...software has to be written specifically for Hadoop
1 Motivation

2 Introducing Seal

3 New additions

4 Conclusion
Seal is:

- a suite of distributed tools for processing HT sequencing data
- runs on the Hadoop MapReduce framework

Goals

<table>
<thead>
<tr>
<th>Scalable</th>
<th>In cluster size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>In data size</td>
</tr>
<tr>
<td>Robust</td>
<td>Resilient to node failure and transient cluster problems</td>
</tr>
<tr>
<td>Sufficient</td>
<td>Implement all data-intensive steps of our sequencing pipeline</td>
</tr>
</tbody>
</table>
At the time of the last Seal publication we only had two tools:

**Seqal**
- Hadoop-based read aligner
- Incorporates BWA’s alignment code
- Simultaneously identifies PCR duplicates

**Prq**
- Required for Seqal
- Reformat read and mate into the same record

CRE_242:1:2204:1453;1918#0 READ1 QUAL1 READ2 QUAL2
Outline

1 Motivation

2 Introducing Seal

3 New additions

4 Conclusion
Bcl to Qseq

Description

Extract reads in qseq format from Illumina bcl files, using Hadoop

- Wraps Illumina’s own bclToQseq utility
- Automatically runs many instances in parallel on Hadoop cluster
  - Based on Pydoop
- Supports all the original utility’s features; adds Hadoop benefits
Demux

Description

Separate (demultiplex) samples in multiplexed runs.

- Analogous to functionality provided by Illumina’s tools, but scalable
- Separates samples into their own directory
- Can, optionally, also separate reads by number (i.e. 1, 2)
- Can allow for substitution errors in barcodes
RecabTable

Description

Collect base quality statistics for recalibration using Hadoop

- Equivalent to GATK CountCovariatesWalker
- Supported factors (hard-coded):
  - Read group
  - Base quality score
  - Sequencing cycle
  - Dinucleotide
- Generates a table that can be fed to the GATK base quality recalibrator
ReadSort

Description

Distributed sorting of read alignments

- Sorting required to create files usable by downstream software
- ReadSort uses an algorithm based on TeraSort
  - Divides work among all nodes

Getting data out of the Seal environment:
- ReadSort leaves data in n sorted files
- `merge_alignments` program provided to concatenate them all
Seal now uses HadoopBAM (I/O library for sequencing file formats on Hadoop)

Introduced support for data in multiple file formats
  - Qseq
  - Fastq
  - SAM

Both input and output

Also supports transparent *distributed* compression and decompression
  - Codecs: snappy, bzip2, gzip (gzip input files not splittable)
Galaxy integration

- We have implemented a Galaxy wrappers for the Seal tools
  - A bit tricky since Hadoop doesn’t follow Galaxy’s model
- Not directly in the Seal project
- Plan to release them later this year
Outline

1. Motivation
2. Introducing Seal
3. New additions
4. Conclusion
Production use

Seal-based pipeline has been in use for over a year

- Our experience has been positive
- Scales well
- Significantly improved processing throughput
- Significantly lowered operational effort
  - Jobs fail much less frequently, and they are relatively easy to monitor
  - Robustness is important for automation
Future development

- Complete Hadoop 2 compatibility
- Base quality recalibration (complete the workflow)
- Optimization
- Support RNA expression analysis
- Support for efficient columnar file formats
- Support for sequencing platforms other than Illumina

Too bad we won’t have the time to do all this. Pull requests welcome!
Try it and contribute!

Repository

https://github.com/crs4/seal

- Might see more frequent activity at
  https://github.com/ilveroluca/seal
- Next release should arrive soon
  - Currently updating documentation

- Web site: http://biodoop-seal.sf.net
Try it and contribute!

Repository

https://github.com/crs4/seal

- Might see more frequent activity at https://github.com/ilveroluca/seal
- Next release should arrive soon
  - Currently updating documentation

Web site: http://biodoop-seal.sf.net

Thank you!