Scripting for large-scale sequencing based on Hadoop

André Schumacher and Keijo Heljanko
firstname.lastname@aalto.fi

Luca Pireddu and Gianluigi Zanetti
firstname.lastname@crs4.it

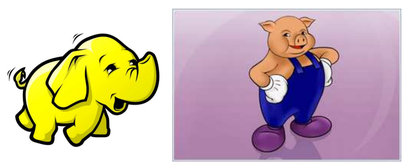
Aleksi Kallio and Eija Korpelainen
firstname.lastname@csc.fi

http://sf.net/projects/seqpig

Web Site

- SeqPig is a tool that facilitates using the Hadoop distributed computing framework to analyse and manipulate NGS data
- Data sizes are increasing faster than processing power and disk read speed
  - SeqPig on Hadoop can help scale the throughput of analysis workflows
- An add-on to Apache Pig (`http://pig.apache.org`); scripts are written in *Pig Latin*
- Scripts are **automatically run as parallel computing jobs**
- Ideal for computing statistics, filtering, projecting and reformatting data
- Supports interactive analysis through its powerful command-line interface
- Parallel scripting environment opens up exciting possibilities for scalable analysis pipelines with a short development cycle

Summary



- Hadoop is the open-source reference implementation of MapReduce
- Widely used in data-intensive industries
  - e.g., Yahoo!, Facebook, Twitter, LinkedIn, etc.
- MapReduce allows the scalable processing of large amounts of data
- Automatically handles splitting/moving data and hardware failures
- Pig is a scripting language and execution engine built on top of Hadoop
- Simplifies the use of Hadoop through its concise SQL-like logic

Hadoop and Pig

SeqPig extends Pig with a number of features for processing sequencing data:

- data input and output components
  - import and export functions for: Fastq, Qseq, SAM and BAM
- extract fields and transform HTS data
  - so you can handle sequencing data like native Pig data (e.g., process SAM flags)
- a collection of scripts and custom functions for frequent tasks
  - implement speed-optimized or simplified task-specific functions

SeqPig

- SeqPig has been tested on Amazon's Elastic MapReduce service
- Users may rent computing time on the cloud to run their SeqPig scripts
- One can also share data in S3 storage buckets with other cloud-enabled software

Running on the cloud

Statistic	Speedup
Read length distribution	28.8
Avg. read quality distribution	28.8
Avg. base quality distribution	28.0
GC-content distribution	8.3
Per-cycle base and base quality	28.0
All of the above	6.9

Data size: 61.42 GB; 16-slave Hadoop cluster vs 1 node for FastQC

Example: speedup over FastQC

Convert qseq into fastq:

```
reads = LOAD 'in.qseq' usingQseqUDFLoader();
STORE reads INTO 'out.fastq' using FastqUDFStorer();
```

Compute a histogram of GC content:

```
reads_by_bases = FOREACH reads
  GENERATE UnalignedReadSplit(sequence, quality);
read_gc = FOREACH reads_by_bases {
  only_gc = FILTER $0 BY readbase == 'G' OR readbase == 'C';
  GENERATE COUNT(only_gc) as count; }
read_gc_counts = FOREACH (GROUP read_gc BY count)
  GENERATE group as gc_count, COUNT_STAR($1) as count;
STORE read_gc_counts INTO 'GC_count_histogram.txt';
```

Compute a histogram of read lenghts:

```
read_len = FOREACH reads GENERATE STRLEN(sequence);
read_len_counts = FOREACH (GROUP read_len BY $0)
   GENERATE group AS len, COUNT_STAR($1) as count;
STORE read_len_counts INTO 'read_len_histogram.txt';
```

Compute a histogram of base composition and base qualities:

```
read_seq_qul = FOREACH reads
  GENERATE sequence, quality;
base_qual_counts = FOREACH (GROUP read_seq_qual ALL)
  GENERATE BaseCounts($1.$0), BaseQualCounts($1.$1);
formatted_base_qual_counts = FOREACH base_qual_counts
  GENERATE FormatBaseCounts($0), FormatBaseQualCounts($1);
```

Example scripts

- Niemenmaa M, Kallio A, Schumacher A, Klemel P, Korpelainen E, and Heljanko K. (2012) Hadoop-BAM: directly manipulating next generation sequencing data in the cloud. Bioinformatics 28(6):876-877. `http://hadoop-bam.sf.net`
- Pireddu,L., Leo,S. and Zanetti,G. (2011). SEAL: a Distributed Short Read Mapping and Duplicate Removal Tool. Bioinformatics. `http://biodoop-seal.sf.net`
- Andrews S., et al. FastQC website: `http://www.bioinformatics.babraham.ac.uk/projects/fastqc`

References