# *Jnomics*—A cloud-scale sequence analysis suite

**Matthew A. Titmus[1,2], Fnu Sneh Lata[3], Eric Antoniou[3], James Gurtowski[1], W. Richard McCombie[3] and Michael C. Schatz[1,2]**

[1] Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, Cold Spring Harbor, NY
[2] Department of Molecular and Cellular Biology, Stony Brook University, Stony Brook, NY
[3] Stanley Institute for Cognitive Genomics, Cold Spring Harbor Laboratory, Cold Spring Harbor, NY

**http://jnomics.sourceforge.net**

## Abstract

In 2000 the Human Genome Project released the "rough draft" of the human genome, the result of ten years of work by researchers in seven countries at a code of $3 billion. Continuing advancements in second-generation sequencing technologies have made it possible for a single laboratory sequence a whole human genome in a matter of days or weeks at less than one millionth of the cost. In total, current worldwide second-generation sequencing capacity exceeds 13 Pbp/year, and continues to increase by 5x each year.

As impressive as this revolution in whole-genome sequence speed and cost may be, however, the storage and analysis of such massive volumes of data has become a primary challenge, necessitating equally revolutionary advancements in computational genomics.

To help meet this challenge, our lab has been applying recent innovations in high performance computing distributed computing – particularly distributed computing – to the challenge of large-scale genomic storage and analysis by creating *Jnomics*, a Java-based toolkit and API based on Google's MapReduce framework, which allows rapid development of parallelized genomic analysis pipelines using components constructed from the Jnomics API and/or existing binaries executed in a distributed fashion.

## Hadoop – a distributed computing framework

*Apache Hadoop* is an open-source Java implementation of the MapReduce framework introduced by Google in 2004. Contributors include Yahoo, Facebook, Twitter, Amazon.

Hadoop is enables distributed computation on large data sets across large computing clusters, potentially allowing applications to work with petabytes of data spread over thousands of nodes.

Benefits: Linearly scalable, reliable, easy to program, runs on commodity computers

Challenges: Map-reduce is not suitable for all problems.

### Hadoop for next-generation sequencing analysis

**CloudBurst**
Highly-sensitive short read mapping with MapReduce
*100x speedup mapping on 96 cores @ Amazon*
http://cloudburst-bio.sf.net (Schatz, 2009)

**Myrna**
Cloud-scale differential gene expression for RNA-seq
*Expression of 1.1 billion RNA-Seq reads in ~2 hours for ~$66*
(Langmead, Hansen, Leek, 2010) http://bowtie-bio.sf.net/myrna/

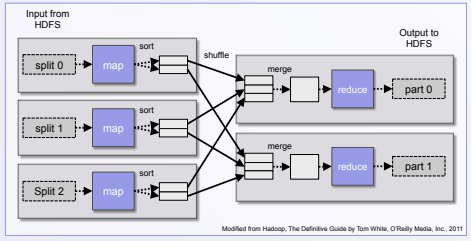
**Quake**
Quality-aware error correction of short read
*Correct 97.9% of errors with 99.9% accuracy*
http://www.cbcb.umd.edu/software/quake/ (Kelley, Schatz, Salzberg, 2010)

**Genome Indexing**
Rapid parallel construction of genome index
*Construct the BWT of the human genome in 9 minutes*
(Menon, Bhat, Schatz, 2011*) http://code.google.com/p/genome-indexing/

### Hadoop MapReduce: distributed computation

Distributed computation in *three phases,* running *in parallel.*

- **Map**: Worker nodes process sub-problem, report results as key:value pairs.
- **Shuffle**: Values from all nodes are grouped by key.
- **Reduce**: Worker nodes process grouped values to produce final output.




### Hadoop HDFS: Distributed File System

- Allows storage of *petabyte size files* as 64 MB blocks across multiple nodes.
- Filesystem tree and block locations are maintained by a *namenode* (master).
- Blocks are replicated among several *datanodes*. Under-replicated blocks are asynchronously replicated across the cluster.

## Distributed genomics analysis with *Jnomics*

Jnomics was designed from the ground up to be intuitive enough to let scientists spend time doing science, while also providing a powerful open-source Java API that lets developers modify, extend, or add functionality:

- **Minimal configuration:** *Jnomics* provides a number of tools out-of-the-box that allow you to distribute a variety of common genomic tasks, including sorting, merging, filtering, selection.
- **File-format agnostic:** *Jnomics* allows users to seamlessly read and write many common formats (SAM, BED, fastq), largely eliminating time-consuming format conversions that add significant overhead to genomics pipelines.
- **Parallelization of existing tools:** Although many excellent genomic tools already exist, very few of these are designed to operate in a distributed environment. Jnomics allows the user to *distribute the execution of existing tools*, allowing an easy transition from serial to distributed analyses. *Jnomics* currently supports BWA and Novoalign (with more to come!); the *Jnomics* API allows components to be added easily.

## Command line examples

### Example 1. Using *Jnomics* to merge files and convert formats

It is trivial to simultaneously merge multiple files and convert them to another format. Given a pair of fastq files:

**input_1.fq:**

```
@READ_NAME/1
GATTACAGATTACA
+
HHHII9DAAACCEF
```

**input_2.fq**

```
@READ_NAME/2
ACTGACTGACTG
+
DDDBBACACCCD
```

**The `jnomics processor` command: distributed sequencing read processing and transformation**

```
$ jnomics processor -in input_1.fq input_2.fq -out combined --out-format sam
READ_NAME  69   *  0  0  0  *  *  0  0  GATTACAGATTACA  HHHII9DAAACCEF
READ_NAME  133  *  0  0  0  *  *  0  0  ACTGACTGACTG    DDDBBACACCCD
```

### Example 2. Using *Jnomics* for distributed read alignment with BWA

It is just as simple to run a distributed BWA job. In this example, the output of the previous is being used as the input. The default output format is SAM.

```
$ jnomics bwa -in combined -out bwaout --aln-args "-q 20" --sampe-args "-a 400"
```

## API example

*Jnomics* provides an open source Java API that makes it simple to create distributed genomic analysis tools.

- *JnomicsTool*: Provides flexible and versatile command line parameter handling.
- *JnomicsMapper* and *JnomicsReducer*: Used to implement map and reduce functions.
- *QueryTemplate* and *SequencingRead*: Reads are provided to the mapper and reducer as one or more *SequencingRead* objects contained within a *QueryTemplate* instance. *Jnomics* automatically combines reads from the same template.

Below is an example of a complete *Jnomics* tool that inputs sequencing reads from an input file and keeps only paired reads.

```
public class FilterUnpaired extends JnomicsTool {
    /**
     * A mapper that writes paired reads, and ignores all others.
     */
    public static class FilterUnpairedMapper
            extends JnomicsMapper<Writable, QueryTemplate, Writable, QueryTemplate> {

        protected void map(Writable key, QueryTemplate value, Context context)
                throws IOException, InterruptedException {

            if (value.size() == 2)
                context.write(key, value); }
    }

    /** The entry point of the tool, replacing main(String[]).
     *  Standard commands and input files are handled automatically.
     */
    public int run(String[] args) throws Exception {
        getJob().setReducerClass(FilterUnpairedReducer.class);
        return getJob().waitForCompletion(true) ? 0 : 1;
    }
}
```

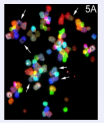
*References*

1. Mitelman *et al.* (2007) The impact of translocations and gene fusions on cancer causation. *Nature Reviews Cancer* 7:223-245
2. Dean, J., Ghemawat, S. (2004) MapReduce: Simplified Data Processing on Large Clusters. *Sixth Symposium on Operating System Design and Implementation*
3. Borthakur, D. (2007). The Hadoop Distributed File System: Architecture and Design. The Apache Software Foundation. Retrieved November 2, 2011, from http://hadoop.apache.org/commondocs/r0.18.0/hdfs_design.pdf
4. Schatz, MC. (2009) CloudBurst: highly sensitive read mapping with MapReduce. Bioinformatics 25(11):1363-9.
5. Quinlan et al. (2010) Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome. *Genome Research* 20(5):623-35
6. Bayani, JM, Squire, JA (2002) Applications of SKY in cancer cytogenetics. *Cancer Invest.* 20(3):373-86

## *Jnomics* case study: structural variations in cancer

### Structural variations

Structural variations (SVs) – balanced or unbalanced chromosomal rearrangements such as insertions, deletions, inversions, and large tandem duplications – represent a major source of genetic variation in humans. SVs can also underlie clinically significant phenotypes by creating copy number alterations in dosage-sensitive genes or rearrangements introducing gain of function mutations. This is particularly evident in carcinogensis: an analysis of available data suggests that gene f*usions occur in all malignancies, and that they account for 20% of human cancer morbidity.*



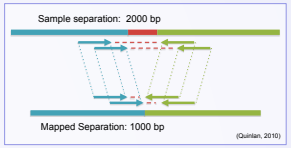
(Bayani et al., 2002)

### Hydra Discordant Pair Analysis

Illumina sequencing generates reads in pairs from both ends of a fragment with a known separation.

SVs can be inferred from discordant pairs that map to the reference with unexpected distance or orientation.

Multiple discordant read pairs are clustered to pinpoint breakpoints.

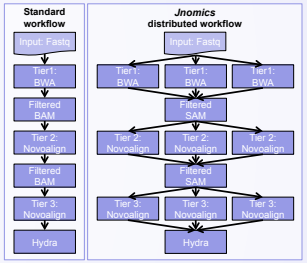



### *Jnomics* vs. standard SV workflows

**Discordant pairs selection by tiered alignments:** each tier aligns the sample reads to the reference; discordant pairs are selected, filtered by quality. Each tiers realigns discordant pairs with progressively greater sensitivity.

Hydra clusters the final discordant pairs, from which it infers breakpoints that indicate structural variations.

The standard (serial) workflow is highly resource intensive, requiring *several weeks per single genome.*

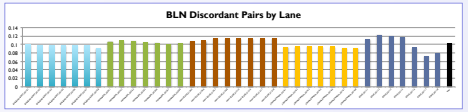



*Jnomics* allows us to parallelize most of the pipeline, and removes several file type conversion steps between BAM, SAM, fastq, and BED.

### Pair analysis of esophageal cancer

Samples of normal, dysplastic Barrett's esophagus, and frank carcinoma from the same individual were sequenced using Illumina paired-end protocol and evaluated using the Hydra structural variation workflow.

- **BLN** (Normal Tissue) – 1.56B reads; discordant pairs: 16% (Tier 1); 10% (final)
- **BLB** (Barrett's esophagus) –1.84B reads; discordant pairs: 17% (Tier 1); 11% (final)
- **BLL** (Esophageal adenocarcinoma) – 1.77B reads, 50% (Tier 1); 14% (final)




### *Jnomics* structural variations

Circos plot of high-confidence SVs specific to pathologic samples.

- **Red**: SV's present only in cancer (BLL) sample.
- **Green**: SV's in cancer (BLL) and pre-cancer (BLB) samples.

A detailed analysis of disrupted and fusion genes in progress.

Preliminary analysis suggests a number of breaks in known oncogenes.