Getting Started with the Genome Analysis Toolkit (GATK)

Matt Hanna

Created March 16, 2009
Updated April 22, 2009

1 Build Prerequisites

GATK requires JDK 1.6 and Ant 1.7.1 to compile.

2 Getting and Building the Source

GATK is located in the Sting svn repository, and compiles using a build.xml in the root directory. Download and build the source as follows:

```
svn co https://svnrepos/Sting/trunk Sting
cd Sting
ant
```

3 Getting Started

The core concept behind GATK is the walker, a class that implements the three core operations: filtering, mapping, and reducing.

filter reduces the size of the dataset by applying a predicate.

map Applies a function to each individual element in a dataset, effectively 'mapping' it to a new element.

reduce Inductively combines the elements of a list. The base case is supplied by the reduceInit() function, and the inductive step is performed by the reduce() function.

Users of the GATK will provide a walker to run their analyses. The engine will produce a result by first filtering the dataset, running a map operation, and finally reducing the map operation to a single result.

4 Creating a Walker

To be loaded by GATK, the walker must satisfy the following properties:

1. It must be a loose class, not packaged into a jar file.
2. It must be in the unnamed package (in other words, the source should not start with a package declaration).
3. It must subclass one of the basic walkers in the org.broadinstitute.sting.gatk.walkers package: ReadWalker or LociWalker.
4. It must live in the directory $STING_HOME/dist/walkers.

5 Example

This walker will print output for each read it sees, eventually computing the total number of reads by mapping every read to 1 and summing all the 1s to realize the total number of reads.

Copy the following text into the file $STING_HOME/dist/walkers/HelloWalker.java:

```
import net.sf.samtools.SAMRecord;

import org.broadinstitute.sting.gatk.LocusContext;
import org.broadinstitute.sting.gatk.walkers.ReadWalker;

/**
 * Define a class extending from ReadWalker with types
 * <MapType,ReduceType>.
 */
public class HelloWalker extends ReadWalker<Integer,Long> {
    private Long currentRead = 0L;

    // Maps each read to the value 1.
    public Integer map(LocusContext context, SAMRecord read) {
        System.out.printf("Hello read %d%n", ++currentRead );
        return 1;
    }

    // Provides an initial value for the reduce function.
    public Long reduceInit() { return 0L; }

    // Defines how to compute the reduction given a value in the list.
    public Long reduce(Integer value, Long sum) {
        return sum + value;
    }
}
```

To compile the walker:

```
setenv CLASSPATH $STING_HOME/dist/GenomeAnalysisTK.jar:$STING_HOME/dist/sam-1.0.jar
javac HelloWalker.java
```

To run the walker:

```
mkdir $STING_HOME/dist/walkers
java -Xmx4096m -jar dist/GenomeAnalysisTK.jar \
     -R/seq/references/Homo_sapiens_assembly18/v0/Homo_sapiens_assembly18.fasta \
     -I /broad/1KG/legacy_data/trio/na12878.bam -T Hello \
     -L chr1:10000000-10000100 -l WARN
```

This command will run the walker across a subsection of chromosome 1, operating on reads which align to that subsection. If you'd like to see more information from the GATK on what it's doing, you can change the logging level (-l) to INFO.