SparkBWA

SparkBWA is a tool that integrates the Burrows-Wheeler Aligner-BWA on a Apache Spark framework running on the top of Hadoop. The current version of SparkBWA (v0.2, October 2016) supports the following BWA algorithms:

- BWA-MEM
- BWA-backtrack
- BWA-SW

All of them work with single-reads and paired-end reads.

If you use SparkBWA, please cite this article:

> José M. Abuin, Juan C. Pichel, Tomás F. Pena and Jorge Amigo. "SparkBWA: Speeding Up the Alignment of High-Throughput DNA Sequencing Data". PLoS ONE 11(5), pp. 1-21, 2016.

A version for Hadoop is available here.

Structure

Since version 0.2 the project keeps a standard Maven structure. The source code is in the *src/main* folder. Inside it, we can find two subfolders:

- **java** - Here is where the Java code is stored.
- **native** - Here the BWA native code (C) and the glue logic for JNI is stored.

Getting started

Requirements

Requirements to build SparkBWA are the same than the ones to build BWA, with the only exception that the *JAVA_HOME* environment variable should be defined. If not, you can define it in the */src/main/native/Makefile.common* file.

It is also needed to include the flag *-fPIC* in the *Makefile* of the considered BWA version. To do this, the user just need to add this option to the end of the *CFLAGS* variable in the BWA Makefile. Considering bwa-0.7.15, the original Makefile contains:

```
CFLAGS=		-g -Wall -Wno-unused-function -O2
```

and after the change it should be:

```
CFLAGS=		-g -Wall -Wno-unused-function -O2 -fPIC
```

Additionaly, and as SparkBWA is built with Maven since version 0.2, also have it in the user computer is needed.

Building

The default way to build SparkBWA is:

```
git clone https://github.com/citiususc/SparkBWA.git
cd SparkBWA
mvn package
```

This will create the *target* folder, which will contain the *jar* file needed to run SparkBWA:

- SparkBWA-0.2.jar - jar file to launch with Spark.

Configuring Spark

Since version 0.2 there is no need of configuring any Spark parameter. The only requirement is that the YARN containers need to have at least 10GB of memory available (for the human genome case).

Running SparkBWA

SparkBWA requires a working Hadoop cluster. Users should take into account that at least 10 GB of memory per map/YARN container are required (each map loads into memory the bwa index - refrence genome). Also, note that SparkBWA uses disk space in the */tmp* directory or in the configured Hadoop or Spark temporary folder.

Here it is an example of how to executeSparkBWA using the BWA-MEM algorithm with paired-end reads. The example assumes that our index is stored in all the cluster nodes at */Data/HumanBase/* . The index can be obtained from BWA using "bwa index".

First, we get the input FASTQ reads from the 1000 Genomes Project ftp:

```
wget ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase3/data/NA12750/sequence_read/ERR000589_1.filt.fastq.gz
wget ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase3/data/NA12750/sequence_read/ERR000589_2.filt.fastq.gz
```

Next, the downloaded files should be uncompressed:

```
gzip -d ERR000589_1.filt.fastq.gz
gzip -d ERR000589_2.filt.fastq.gz
```

and uploaded to HDFS:

```
hdfs dfs -copyFromLocal ERR000589_1.filt.fastq ERR000589_1.filt.fastq
hdfs dfs -copyFromLocal ERR000589_2.filt.fastq ERR000589_2.filt.fastq
```

Finally, we can execute SparkBWA on the cluster. Again, we assume that Spark is stored at*spark_dir*:

```
spark_dir/bin/spark-submit --class com.github.sparkbwa.SparkBWA --master yarn-cluster
--driver-memory 1500m --executor-memory 10g --executor-cores 1 --verbose
--num-executors 32 SparkBWA-0.2.jar -m -r -p --index /Data/HumanBase/hg38 -n 32
-w "-R @RG\tID:foo\tLB:bar\tPL:illumina\tPU:illumina\tSM:ERR000589"
ERR000589_1.filt.fastq ERR000589_2.filt.fastq Output_ERR000589
```

Options used:

- -m - Sequence alignment algorithm.
- -p - Use paired-end reads.
- -w "args" - Can be used to pass arguments directly to BWA (ex. "-t 4" to specify the amount of threads to use per instance of BWA).
- --index index_prefix - Index prefix is specified. The index must be available in all the cluster nodes at the same location.
- The last three arguments are the input and output HDFS files.

After the execution, in order to move the output to the local filesystem use:

```
hdfs dfs -copyToLocal Output_ERR000589/* ./
```

In case of not using a reducer, the output will be split into several pieces (files). If we want to put it together we can use "samtools merge".

If you want to check all the available options, execute the command:

```
spark_dir/bin/spark-submit --class com.github.sparkbwa.SparkBWA SparkBWA-0.2.jar -h
```

The result is:

```
SparkBWA performs genomic alignment using bwa in a Hadoop/YARN cluster
 usage: spark-submit --class com.github.sparkbwa.SparkBWA SparkBWA-0.2.jar
      [-a | -b | -m]  [-f | -k] [-h] [-i <Index prefix>]   [-n <Number of
      partitions>] [-p | -s] [-r]  [-w <"BWA arguments">]
      <FASTQ file 1> [FASTQ file 2] <SAM file output>
Help options:
  -h, --help                          Shows this help

Input FASTQ reads options:
  -p, --paired                         Paired reads will be used as input FASTQ reads
  -s, --single                        Single reads will be used as input FASTQ reads

Sorting options:
  -f, --hdfs                         The HDFS is used to perform the input FASTQ reads sort
  -k, --spark                         the Spark engine is used to perform the input FASTQ reads sort

BWA algorithm options:
  -a, --aln                          The ALN algorithm will be used
  -b, --bwasw                           The bwasw algorithm will be used
  -m, --mem                           The MEM algorithm will be used

Index options:
  -i, --index <Index prefix>                  Prefix for the index created by bwa to use - setIndexPath(string)

Spark options:
  -n, --partitions <Number of partitions>         Number of partitions to divide input - setPartitionNumber(int)

Reducer options:
  -r, --reducer                        The program is going to merge all the final results in a reducer phase

BWA arguments options:
  -w, --bwa <"BWA arguments">                  Arguments passed directly to BWA
```

Accuracy

SparkBWA should be as accurate as running BWA normally. Below are GCAT alignment benchmarks which proves this.

MEM * Single-reads (400 bp) * Pair-ended reads (100 bp) * Pair-ended reads (150 bp)

BWA-backtrack * Single-reads (100 bp) * Pair-ended reads (250 bp)

BWA-SW * Single-reads (400 bp) * Pair-ended reads (250 bp)

Frequently asked questions (FAQs)

1. I can not build the tool because *jni_md.h* or *jni.h* is missing.

1. I can not build the tool because *jni_md.h* or *jni.h* is missing.

You need to set correctly your *JAVA_HOME* environment variable or you can set it in Makefile.common.

INFORMACIÓN

Investigadores
José Manuel Abuín Mosquera
Juan Carlos Pichel Campos
Tomás Fernández Pena
Jorge Amigo Lechuga

Licenza

DESCARGAR

Repositorio Gitlab

Descargar de Gitlab

Repositorio Github

PUBLICACIÓNS

SparkBWA: Speeding Up the Alignment of High-Throughput DNA Sequencing Data
PLoS One, 2016

PROXECTOS DE INVESTIGACIÓN

SHSCAP: Solucións hardware e software para a computación de altas prestacións