bdgenomics.workflows Documentation

Release 0.1.0-SNAPSHOT

Big Data Genomics

Jan 09, 2018

Installation

CHAPTER 1

Introduction

Avocado is a variant caller built on top of Apache Spark to allow rapid variant calling on cluster/cloud computing environments. Avocado is built on ADAM's APIs, and achieves variant calling accuracy that is similar to state-of-the-art tools while being able to drop variant calling latency to approximately 15 minutes when running on a 1,024 core cluster.

CHAPTER 2

Workflows Supported

Avocado is run through the *avocado-submit* command line:

```
./bin/avocado-submit
```

```
Using SPARK_SUBMIT=/usr/local/bin/spark-2.2.1-bin-hadoop2.7/bin/spark-submit

Usage: avocado-submit [<spark-args> --] <avocado-args> [-version]

Choose one of the following commands:

biallelicGenotyper : Call variants under a biallelic model
discover : Discover variants in reads
jointer : Joint call and annotate variants.
mergeDiscovered : Merge variants discovered from reads of multiple samples
reassemble : Reassemble reads to canonicalize variants
trioGenotyper : Call variants in a trio under a biallelic model
```

The *avocado-submit* script follows the same conventions as the *adam-submit* command line, whose documentation can be found here. As a result, just like ADAM, Avocado can be deployed on a local machine, on AWS, an in-house cluster running YARN or SLURM, or using Toil.

Avocado supports several workflows:

- Single sample germline variant calling: Avocado's BiallelicGenotyper runs on a single sample at a time, and can generate both variants-only (VCF) and all-sites (gVCF) output.
- Joint variant calling: Avocado supports jointly calling variants from a collection of gVCF-styled inputs.

Avocado also contains code to reassemble variants, and a pedigree variant caller. However, this code is experimental and is thus unsupported.

For genotyping, Avocado uses a probabilistic model that assumes that sites are biallelic. This model is derived from the biallelic model used by the Mpileup variant caller, but modified to better call multiallelic sites. When used with the INDEL realigner from ADAM, Avocado has >99% accuracy when genotyping SNPs, and >96% accuracy when genotyping INDELs.

2.1 Building Avocado from Source

You will need to have Apache Maven version 3.1.1 or later installed in order to build Avocado.

> **Note:** The default configuration is for Hadoop 2.7.3. If building against a different version of Hadoop, please pass `-Dhadoop.version=<HADOOP_VERSION>` to the Maven command.

```
git clone https://github.com/bigdatagenomics/avocado.git
cd avocado
export MAVEN_OPTS="-Xmx512m -XX:MaxPermSize=128m"
mvn clean package -DskipTests
```

Outputs

```
...
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 9.647s
[INFO] Finished at: Thu May 23 15:50:42 PDT 2013
[INFO] Final Memory: 19M/81M
[INFO] ------------------------------------------------------------------------
```

You might want to take a peek at the `scripts/jenkins-test` script and give it a run. We use this script to test that Avocado is working correctly.

2.1.1 Running Avocado

Avocado is packaged as an überjar and includes all necessary dependencies, except for Apache Hadoop and Apache Spark.

You might want to add the following to your `.bashrc` to make running Avocado easier:

```
alias avocado-submit="${AVOCADO_HOME}/bin/avocado-submit"
```

`$AVOCADO_HOME` should be the path to where you have checked AVOCADO out on your local filesystem. The alias calls a script that wraps the `spark-submit` command to set up Avocado. You will need to have the Spark binaries on your system; prebuilt binaries can be downloaded from the Spark website. Our continuous integration setup builds ADAM against Spark 2.0.0, Scala versions 2.10 and 2.11, and Hadoop versions 2.3.0 and 2.6.0.

Once this alias is in place, you can run Avocado by simply typing `avocado-submit` at the command line.

```
avocado-submit
```

2.2 Single Sample Variant Calling

To call variants using Avocado, use the *BiallelicGenotyper* command. This command discovers possibly variant sites from a collection of reads. The discovered sites are then genotyped using a biallelic probabilistic model. The genotyping model is based off of the biallelic model used by the original Samtools mpileup variant caller, but adds additional components for modeling a site with two minor alleles, as well as reads that do not match any known allele. The full genotyping model is described in Chapter 7 of this thesis.

To run the *BiallelicGenotyper*, you must provide two parameters:

- The path to the input file (in any file format for reads that ADAM can load)

- The path where the output should be saved as Parquet-encoded Genotypes.

You can configure how Avocado discovers variants to genotype, the genotyping phase, and the hard filters that Avocado applies to the called genotypes.

2.2.1 Reads to Evaluate

By default, Avocado only calls variants in autosomal regions. Avocado does this by inspecting the names of the contigs that the reads are mapped to. Avocado assumes that the contig names start with `chr` prefixes. If your reference build does not have these prefixes (i.e., chromosome 1 is `1`, instead of `chr1`), you need to pass the `-is_not_grc` option. To enable calling non-autosomal regions, you can pass:

- `-keep_mitochondrial_chromosome`: to call variants on the mitochondrial chromosome.
- `-autosomal_only`: to call variants on the sex chromosomes.

Additionally, we apply several read quality filters. To disable these filters, you can pass:

- `-keep_duplicate_reads`: To disable discarding reads that have been marked as a PCR duplicate.
- `-keep_non_primary`: To disable discarding reads that have non-primary alignments.
- `-min_mapping_quality`: To change the default mapping quality below which reads are filtered out (default is phred 10).

2.2.2 Variant Discovery

Avocado treats a site as possibly variant if enough alternate alleles are seen with high enough quality. If you already know which variants you want to call, you can skip variant discovery by passing the `-variants_to_call` flag, with a path that ADAM can load as variants. During variant discovery, you can tune the variants that are discovered with the following flags:

- `-min_phred_to_discover_variant`: The minimum Phred quality for a read containing a variant allele to be considered a confident observation of that allele.
- `-min_observations_to_discover_variant`: The minimum number of confident observations of a variant allele for us to choose to score a variant.

2.2.3 Genotyping

By default, Avocado only scores the sites we have identified as possible variants. However, Avocado can also emit genotype likelihoods for sites where no alternate allele was seen. This output is not banded by quality and thus is equivalent to a BP RESOLUTION gVCF. To run in this mode, pass the `-score_all_sites` option on the command line.

The only parameters that the genotyping engine consumes are around copy number. Unless otherwise specified, Avocado assumes that it is running on a diploid sample. To set the default copy number to another value, pass the ploidy with the `-ploidy` option. Additionally, copy number variants can be passed with the `-cnv` flag. The copy number variants should be described in the GFF format that is compatible with the DECA and XHMM copy number variant callers.

2.2.4 Variant Filtration

Avocado applies hard filters separately to SNPs and INDELs. The following hard filters can be applied:

- `min` and `max`:

- `depth`: The read depth covering the site.
- `rms_mapping_quality`: The RMS quality of reads mapped at the site.
- `min` and `max`, separately for `het` and `hom` genotypes:
- `allelic_fraction`: The fraction of reads that support the major vs. minor allele. E.g., if 25 reads support the major (reference) allele and 75 reads support the minor (alternate) allele, the site has an allelic fraction of 0.75. For `hom` genotypes, only `min` is supported.
- `min` only, separately for `het` and `hom` genotypes:
- `quality_by_depth`: The genotype quality divided by the read depth. Sites with a low quality-by-depth may be highly covered (leading to high quality), but with low quality reads.

The default values are subject to change, but are described on the `BiallelicGenotyper`'s command line when `-help` is printed.

2.2.5 Translating to VCF

The `BiallelicGenotyper` saves its output as Apache Parquet, formatted using ADAM's Genotype schema. To translate this representation back to VCF, you can use ADAM's transformGenotypes command, or Avocado's *Jointer* command. We recommend using the `Jointer` command, which will calculate variant `QUAL` for all genotyped sites.

2.3 Joint Variant Calling

Avocado's `Jointer` command supports joint variant calling from gVCF-styled data. The `Jointer` command can also be used to export Apache Parquet Genotype data to VCF, and to joint genotype a collection of samples who all scored the same set of variants. Our joint variant calling approach is is described in Chapter 7 of this thesis.

To run the `Jointer` command, you must provide two parameters:

- The path to all input files to joint genotyping (to load multiple files, use Hadoop's glob syntax.
- The path to save the output to, as a VCF file.

To save the VCF file as a single file (instead of sharded output), pass the `-single` flag.

If run on a single sample, `Jointer` will calculate variant statistics (VCF INFO column attributes) and qualities only. If run on multiple samples, the `Jointer` command will update the called genotypes using a binomial prior that is informed by the observed allele frequency of the variant across all samples with confident calls. If the input data for the multiple samples is in gVCF format, pass the `-from_gvcf` flag.

- genindex
- search