Week 12, Lecture 23

István Albert

Bioinformatics Consulting Center
Penn State, 2015
Variant calling: a world unto its own

Why is it so difficult to call variants?

Genome == Information

Information will vary as much as needed to allow the organism to survive

there is reason for it to be easy to understand
How to tell when you know something?
Bioinformatics in 2015

• We live in an era of massive data generation

• It is also driven by technology

• Current technologies are extremely inefficient \(\rightarrow\) everything needs to be measured hundreds of times

• Most of the data is redundant and most of the information is lost

• Information that is not lost \(\rightarrow\) that’s what Bioinformatics of 2015 is about
Best practices evolve very fast
some things stay the same

Conceptually always the same

Step 1 → Step 2 → Step 3 → Step 4

Can be automatized at the command line

What you actually may change (a lot)
“How do we call SNPS?”

Always depends on the complexity of the information.

Easy problems ➔ can be solved by any tool
Complex problems ➔ only some results are correct
Difficult problems ➔ needs human intervention
Even comparing SNP calls is hard (we can’t even tell which SNP callers are correct)

Blue Collar Bioinformatics

Validating multiple cancer variant callers and prioritization in tumor-only samples
leave a comment »

Overview
The post discusses work validating multiple cancer variant callers in bcbio-nextgen using a synthetic reference call set from the ICGC–TCGA DREAM challenge. We’ve previously validated germline variant calling methods, but cancer calling is additionally challenging. Tumor samples have mixed cellularity due to contaminating normal sample, and consist of multiple sub-clones with different somatic variations. Low-frequency sub-clonal variations can be critical to understand disease progression but are more difficult to detect with high sensitivity and precision.
Personalized Medicine

All about the variation

One of the fastest growing fields of Bioinformatics

One with the most promise and hype
ABOUT THE 1000 GENOMES PROJECT

Project Overview
Project Design
Use of the Project data and samples
Samples included in the project
Publications and project documents

PROJECT OVERVIEW

Recent improvements in sequencing technology ("next-gen" sequencing platforms) have sharply reduced the cost of sequencing. The 1000 Genomes Project is the first project to sequence the genomes of a large number of people, to provide a comprehensive resource on human genetic variation.

As with other major human genome reference projects, data from the 1000 Genomes Project will be made available quickly to the worldwide scientific community through freely accessible public databases. (See Data use statement.)

The goal of the 1000 Genomes Project is to find most genetic variants that have frequencies of at least 1% in the populations studied. This goal can be attained by sequencing many individuals lightly. To sequence a person's genome, many copies of the DNA are broken into short pieces and each piece is sequenced. The many copies of DNA mean that the DNA pieces are more- or-less randomly distributed across the genome. The pieces are then aligned to the reference sequence and joined together. To find the complete genomic sequence of one person with current sequencing platforms requires sequencing that person's DNA the equivalent of about 28 times (called 28X). If the amount of sequence done is only an average of once across the genome (1X), then much of the sequence will be missed, because some genomic locations will be covered by several pieces while others will have none. The deeper the sequencing coverage, the more of the genome will be covered at least once. Also, people are diploid; the deeper the sequencing coverage, the more likely that both chromosomes at a location will be included. In addition, deeper coverage is particularly useful for detecting structural variants, and allows sequencing errors to be corrected.

Sequencing is still too expensive to deeply sequence the many samples being studied for this project. However, any particular region of the genome generally contains a limited number of haplotypes. Data can be combined across many samples to allow efficient detection of most of the variants in a region. The Project currently plans to sequence each sample to about 4X coverage; at this depth sequencing cannot provide the complete genotype of each sample, but should allow the detection of most variants with frequencies as low as 1%. Combining the data from 2500 samples should allow highly accurate estimation (imputation) of the variants and genotypes for each sample that were not seen directly by the light sequencing.
The Genome Analysis Toolkit or GATK is a software package for analysis of high-throughput sequencing data, developed by the Data Science and Data Engineering group at the Broad Institute. The toolkit offers a wide variety of tools, with a primary focus on variant discovery and genotyping as well as strong emphasis on data quality assurance. Its robust architecture, powerful processing engine and high-performance computing features make it capable of taking on projects of any size.
GATK is well documented but not easy to use, lots of gotchas – but it works!

If this is your first rodeo, you're probably asking yourself:

- **What can GATK do for me?** Identify variants in a bunch of sample sequences, with great sensitivity and specificity.
- **How do I get GATK to do that?** You run the recommended Best Practices steps, one by one, from start to finish, as described in the Best Practices documentation.
- **No but really, how do I know what to do?** For each step in the Best Practices, there is a tutorial that details how to run the tools involved, with example commands. The idea is to daisy-chain all those tutorials in the order that they're referenced in the Best Practices doc into a pipeline.
- **Oh, you mean I can just copy/paste all the tutorial commands as they are?** Not quite, because there are a few things that need to be tweaked. For example, the tutorials use the `-L/--intervals` argument to restrict analysis for demo purposes, but depending on your data and experimental design, you may need to remove it (e.g., for WGS) or adapt it (for WES). Hopefully it's explained clearly enough in the tutorials.
- **Why don't you just provide one script that runs all the tools?** It's really hard to build and maintain a one-size-fits-all pipeline solution. Really really hard. And not nearly as much fun as developing new analysis methods. We do provide a pipinglining program called Queue that has the advantage of understanding GATK argument syntax natively, but you still have to actually write scripts yourself in Scala to use it. Sorry. Maybe one day we will be able to offer GATK analysis on the Cloud. But not today.
Better Ingredients - Better SNP calls

Base recalibration
(makes the read quality score better reflect the predictive values)

Mark (remove) duplicates
(identical reads are likely artificial, hence will magnify errors)

Realign around indels
(Perform local realignment around indels to correct mapping-related artifacts.)

Refine genotypes
(compare to known variants and establish stricter discovery to de-novo calls)
Homework 23

• Install the picard tool

• Use alignments that you have previously generated

• Use the picard to perform the following:
  
  – Mark the duplicates in your data. How many of your alignment are duplicates?

  – Add read group information to your data. How does your alignment file show the read groups?