NGS Tools and Formats for Secondary Analysis: A Primer

by Gabe Rudy, VP of Product Development at Golden Helix

After the publication of my three-part series: A Hitchhiker's Guide to Next Generation Sequencing, we have had quite a few requests for more details on the file formats and programs involved in getting data ready for tertiary analysis. Sequencing service providers most likely will have a set of tools orchestrated into a secondary analysis pipeline to process the sequence data from the machine to the point of being deliverable for tertiary analysis. But there are good reasons to understand this pipeline yourself in more detail. For one, you may want to know what to expect or ask from a service provider, internal core lab, or collaborating bioinformatician. Or you may simply want to learn about the types of files produced by various pipelines and what that means for your tertiary analysis.

We will focus here on tools for the secondary analysis of human DNA. Why? Well, although sequence data in general looks deceptively similar in its raw format, the tools and methods used for RNA sequencing versus DNA sequencing get more specialized the further down the pipeline you go. For example, compared to DNA, a fairly different analysis pipeline is necessary for assembling RNA due to the complexity of going from already transcribed and spliced transcripts (the transcriptome) back to the genome. Looking at the differential expression of genes or transcripts between samples or genes as you do in a RNAseq experiment requires a completely different set of secondary and tertiary analysis tools than DNA re-sequencing.

So in particular, we will look at the DNA re-sequencing tools. Here is the basic workflow:

1. Take the raw short sequencing reads and do some quality checks and clean-up, thereby producing a "clean" FASTQ file per sample.
2. On a per-sample basis, assemble the short reads for the sample to a reference genome. This will produce a Sequence Alignment/Map (SAM) file per sample (as well as a file of reads that were not aligned).
3. Process the aligned reads per sample to build the consensus sequence and call where there are differences to the reference (variant sites such as those in a VCF file).

But first, let's review some file formats.

File Formats Glossary

Fortunately, for the most part, each major step in the analysis pipeline has a corresponding standardized file format. Unfortunately, while file formats seem universal, they tend to be riddled with special cases and sometimes even blatantly improper behavior from programs failing to adhere to the format specifications in either reading or writing files. Here is a list of file formats you are likely to encounter and what they are suited for:

FASTA/FASTQ – a series of sequences and their base qualities:

The FASTA file format is a simple text format that contains one or more records consisting of a start indicator, a comment, and then a sequence of nucleotides or peptides encoded as letters of the alphabet. The start indicator is most commonly ">" and the comment usually annotates how the sequence was produced (i.e. the platform or program and information about the sample). Often, the reference sequence for species are in FASTA files with one record per chromosome. Although almost exclusively thought of as a file format, FASTA can also refer to the analysis method published by Bill Pearson in 1985 that produced these files as its output.

FASTQ is a file format initially developed by the Wellcome Trust Sanger Institute to bundle a FASTA sequence and its quality data. There are a couple variants to the format that encode the quality score differently, specifically the original Sanger version and the

Solexa variant (which was updated to the Illumina variant). Most programs can auto-detect the differences between these by looking at the information in the comment field of a sequence.

While FASTQ has quickly become the standard for storing short read, high throughput sequencing data, each platform may have native formats that get converted into FASTQ. For example, Illumina's "pipeline" software for primary analysis outputs the QSEQ format, which can be directly converted to FASTQ. The ABI SOLiD solid platform uses a 2 base encoding for their nucleotide sequences and hence uses a "Color Space" FASTA file (CSFASTA). (Although you can convert this to FASTQ, you would lose the advantages of the SOLiD platform. Some aligners/assemblers specialize in operating in "color space.")

SAM/BAM – a series of sequences and their alignment to a reference genome:

The SAM (Sequence Alignment/Map) format is a very versatile and near-standardized format for storing many aligned nucleotide sequences. BAM is simply a more compact binary equivalent of the SAM format, allowing for trivial conversions back and forth between the two formats. The venerable SAMtools utilities provide ample power to manipulate these files such as sorting, indexing, merging, and building reports of per-base-pair alignment data (see pileup discussion below).

One thing to note is the difference between sorted and unsorted SAM files. Most tools require the data to be sorted so as to be able to operate on data within genomic intervals of interest. Since sorting doesn't lose any information, it's usually a standard step of most pipelines and what you will end up seeing in use. But it's good to be aware that SAM/BAM files may be unsorted.

SAMtools is a Swiss Army Knife of operations that can be used on SAM/BAM files. One popular utility of SAMtools is to create a text-based representation of all the reads that form a "pileup" at a given chromosome locus with its pileup command. This aggregation of information to the base-pair level instead of the sequence read level can facilitate calling variants. The deprecated pileup and varFilter SAM commands do just that and can be combined with some simple Unix command line wizardry to get down to a decent set of quality variant calls. If you are using varFilter, it's useful to know that it essentially adds six additional columns of computed variant information to the standard pileup format. So if you see a "pileup" file with more than the expected columns, it's most likely a pileup file that varFilter was used on.

VCF – variants between a sample and the reference genome:

Standardized by the 1000 Genomes Project, the Variant Call Format (VCF) is now in version 4.0 of the specification. Previous versions were primarily subsets of the 4.0 specification or simply had less defined behavior for specific fields.

What gives VCF its allure and popularity can also be a point of frustration: its flexibility. You can specify pretty much any type of genetic variant at the single sample, multi-sample, or even whole population level in one format. You can encode any combination of genotype, quality, and pretty much anything else you would like to attach to variants. And while the header can provide meta-information about what to expect in a given VCF file, there are no guarantees that the included variants adhere to the header "tags" or that there is even a header at all!

As you can imagine, it can be quite a feat to create a VCF importer that performs consistently and still takes advantage of the flexibility that the format offers.

Stages of a DNA Re-Sequencing Pipeline

Okay, so now we can recognize all those file extensions on that external hard drive or FTP folder that you were given. As I stated in the beginning, these are all simply inputs and outputs to the three step process of QA filtering, aligning and variant calling.

Let's now go into more details on these steps and cover the most common tools used in practice for each. Note that this list will most likely be dated in a matter of months, but it certainly reflects the state of things at this time.

Sequence Quality Assurance

Like any finely tuned and powerfully scientific instrument, it may serve you to follow the precedent of US foreign relations in dealing with powerful entities: Trust, but verify.

In the case of looking at the millions of short reads that come from today's high throughput sequencing platforms, there are an number of great ways to verify that the reads are of good quality and that the samples you intended to sequence were not contaminated or degraded to useless levels.

The quality assurance step will process the FASTQ files that are produced by the sequencer machine and output reports and potentially "cleaned up" FASTQ files for further analysis.

FastQC provides a nice static report that measures various metrics of quality of the sequences themselves (with examples of a "good" sequence run versus a "bad" one). FastQ Screen will give you the confidence that the species of the sample you thought you were sequencing is, in fact, accurate and that bacterial or viral contamination is minimal. FASTX is another tool with a variety of FASTQ and FASTA manipulation and quality reporting abilities.

Sequence Alignment

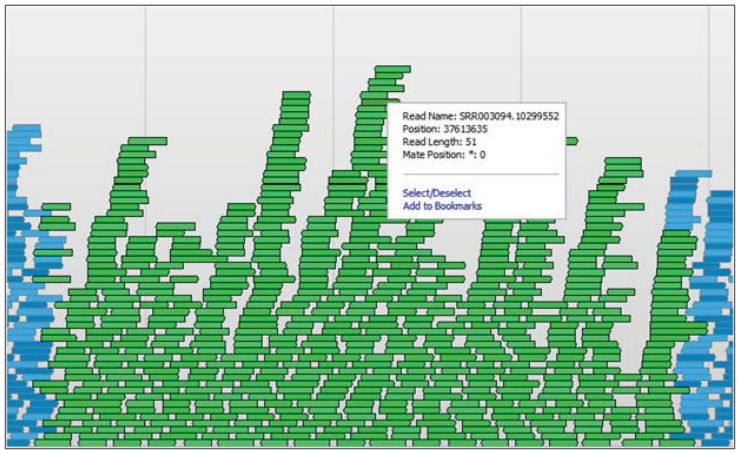
In the sequence alignment step, a single FASTQ file (representing a single biological sample's short reads from the sequencer) and a reference genome sequence (such as the hg18 or hg19 human reference sequence) are used as input and a SAM file are produced as output.

A whole separate article could be devoted to the advances, intricacies, and pitfalls of various sequence aligners. Like any choice of algorithms, the point (or points) of compromise often fall around trade-offs of time, accuracy, and specificity for your exact problem.

The time and accuracy trade-off is fairly well represented between the computationally expensive Maq aligner and the more efficient group of BWA, SOAP, and Bowtie. For the most part, the efficient group can reach the accuracy of Maq and others like BLAT with the right parameters but, as you can see, you are not starved for choices. For the most part, BWA is a good general purpose choice but, as always, Your Mileage May Vary.

Variant Calling

Now that we have aligned sequences in a SAM/BAM format for each sample, we can make variant calls for each sample, producing a variant call file, such as VCF. This will be the output required for tertiary analysis that would then combine the variant call files of multiple samples to be able to ask the questions and do the analyses specific to a given experiment.



Pileup from the Savant Genome Browser

Similar to aligners, there are a number of "variant callers" in the wild with various degrees of functionality. A variant caller essentially calls the genotype of a sample at a given locus. So for example, you may see evidence in the aligned sequences that roughly half of the sequences have a "G" at a given position and the others have the reference sequence allele, "T". This would be called as a "G/T" genotype at that position. Positions that are all perfect or nearly all perfect matches with the reference sequence are not reported. While simple in theory, there are various ways to account for the inherent presence of sequencing errors, the biases of the aligner program, and the challenges of detecting small insertions or deletions that account for the differences in variant callers.

The BGI bioinformatics group released SOAPsnp which they often use on their internal and external projects. The Broad Institute developed a very general framework for doing analysis work in a distributed environment called GATK which contains a variant caller with plenty of options to explore.

But for the most part, expect to see usage of SAMtools for doing variant calling of the basic Single Nucleotide Variant (SNV) sort.

Let's take a look at what running SAMtools on aligned sequence data in a file called "mysample.sorted.bam" would look like being compared to the hg18 human reference sequence in a file called "hg18.fa". If you hope to never be running command line programs of this nature, feel free to skip to the next section.

There are two workflows for producing a filtered variant list using SAM tools. The newest uses their mpileup tool and looks something like this in practice:

```
$ samtools mpileup -ugf hg18.fa mysample.sorted.bam | bcftools view -bvcg - > mysample_var.raw.bcf
$ bcftools view mysample_var.raw.bcf | vcfutils.pl varFilter -D1500 > mysample_var.flt.vcf
```

A previous (and some would say more vetted) variant calling workflow used a Perl script to filter the raw pileup data and looked something like this:

```
$ samtools pileup -vcf hg18.fa mysample.sorted.bam > mysample.raw.pileup -c
$ samtools.pl varFilter -D1500 mysample.raw.pileup | awk '$6>20' > mysample.final.pileup
```

As you would expect, the details are in your choice of how to filter down reported differences from the reference sequence to a list that can reasonably be trusted to represent the underlying biology. In the above examples, filters like -D1500 remove sketchy variants that have over 1,500 coverage (most likely bad alignment). Other common filters include only trusting variants with a depth of coverage greater than say, 30, or have a computed Phred quality score greater than 20 (as was done above with "| awk '$g>20'").

Finally, a note on "indels" (short insertions and deletions). It is commonly understood that the standard pipeline optimized for detecting SNVs does not perform well in detecting real insertions and deletions. Instead, you most likely want to do a local de novo assembly of the reads in the region of a suspected indel to get a proper reflection of the sample's consensus sequence for that region. Then a proper comparison of that consensus to the reference can be made to report insertions or deletions. This is the technique Complete Genomics uses in their whole genome sequencing service. A similar method is implemented in the Dindel package from the Sanger institute. Other programs that call indels include GATK mentioned above and SOAPindel from BGI.

Onward

And that's it! The output of the variant caller should be a VCF file or similar per sample and you are ready for tertiary analysis.

With all these tools and options, it can be overwhelming to be in the shoes of a bioinformatician with a stream of sequence data pouring out of your Illumina HiSeq 2000 and a growing line of customers in need of actionable data. Most core labs of this nature either have someone dedicated to the design and maintenance of a pipeline built on these tools, or look to supported commercial solutions that will provide equivalent functionality. Commercial solutions may be built on their own algorithms for alignment and variant calling such as CLC Bio. There is certainly a case to be made for better algorithms, but commercial solutions may also simply provide an easy-to-use user interface, a support team, and a best-of-breed solution based on open-source algorithms. Finally, offerings such as DNAnexus can slurp up data directly from the sequencer and perform all the secondary analysis on the cloud.

If you're a researcher, you may have a choice of sequencing service providers whether an internal core lab or an external provider such as BGI or Expression Analysis. In that case, you now will have a better idea of what to ask for to be well prepared for your tertiary analysis.

For more information about Golden Helix, visit
www.goldenhelix.com

About Gabe Rudy

Gabe Rudy is GHI's Vice President of Product Development. Gabe is continually scouting the fast changing fields of both genetics analysis and software development. This in turn leads to curating features and infrastructure improvements so that the GHI development team can regularly provide our valued customers with top-of-the-line software releases to further accelerate their research. Gabe joined Golden Helix while still an undergrad student at MSU-Bozeman and then ventured to the University of Utah where he received his masters degree in Computer Science. Outside of the office Gabe enjoys taking advantage of what life in the Rocky Mountains has to offer: white water kayaking, skiing, and running. However, with a brand new addition to the Rudy family, Gabe truly loves spending time with his son and family.

About Golden Helix

We are inspired by significance. Not only statistical, but technological, scientific, and personal significance. It's embodied in everything we and our customers do. And we believe the only way to achieve significance is by transcending the status quo. Every day we strive for extraordinary analytic and technological advancements that empower scientists around the world to pursue that which is truly significant: from uncovering the genetic causes of disease and transforming drug discovery to developing genetic diagnostics and advancing the quest for personalized medicine.