Next Generation Sequencing

SPH 247

Statistics for Laboratory Data

David M. Rocke

NGS Technologies

- Illumina Sequencing HiSeq 2500, 3000 & MiSeq
- PacBio Sequencing PacBio RSII
- Fluidigm C1 Single Cell Genomics Auto-Prep
- Fluidigm Access Array Target Amplification
- Infinium Genotyping
- BeadXpress Genotyping
- Fluidigm EP1 Genotyping

Illumina Sequencing

- The Illumina sequencing platforms generate up to 94 gigabases of high quality sequence data per lane (HiSeq 3000) or up to 15Gb (MiSeq), using a massively parallel sequencing approach.
- The Illumina systems use sequencing-by-synthesis technology and reversible terminator chemistry, optimized for cost effectiveness and throughput.
- The DNA Technologies and Expression Analysis Cores of the UC Davis Genome Center operate one HiSeq 2500, one HiSeq 3000, and three MiSeqs.
- The BGI@UC Davis facility currently houses one Ion Proton System, five Illumina HiSeq 2000s and three Illumina HiSeq 2500s.
- All are capable of running paired-end reads that allow each library insert to be sequenced from both ends. Data produced from such paired end reads greatly facilitates assemblies and investigation of genomic structure.
- This capability, coupled with chemistry improvements, allows for longer accurate reads and has expanded the utility of these platforms for de novo assemblies.

Cluster Generation and Sequencing by Synthesis (SBS)

- Flow cell preparation (cluster generation): Each flow cell has eight lanes (high-output mode) or two lanes (rapid-output mode), with each lane containing an individual library or a pool of multiplexed libraries.
- One lane is sometimes reserved for a PhiX control for high-output mode, or spiked-in with PhiX for rapid-output mode.
- Typically several libraries of various types will be on a flow cell. Most libraries will use the Illumina sequencing primers.

Read Length

- Low error rates reads of up to 250 bases of paired-end reads for HiSeq or 300 bases of paired-end reads for MiSeq are possible.
- The read length required will depend on experimental needs.
- A single read 50 bp (SR50), paired end 100 bp (PE100), paired end 150 bp (PE150), and rapid-mode paired end 250bp (PE250) are standard read lengths offered by the Core.
- SR50 will provide an unambiguous match to a reference genome and is therefore suitable for most ChIP-seq and mRNA-seq applications where a reference genome is available.
- SR50 is the read length used for small RNA and micro RNA analyses. It can also be used for certain SNP discovery applications, where a reference genome is available. It can be suitable for some bacterial genome analyses, and reduced representation or hybrid-selected libraries, depending on the application.

Read Length

- The MiSeq generates about 12-15 million reads passing filter (using v2 chemistry) or up to 25 million reads PF (v3 chemistry).
- Available Illumina kits include: SR50 (v2 kit), PE75 (v3), PE150 (v2), PE250 (v2), and PE300 (v3 kit).
- Considering the fast turnaround time (1-3 days once libraries are loaded for sequencing), this would be an excellent option for small scale and proof-of-concept projects under a strict timeline.
- Illumina is projecting PE400 read lengths in the near future.

Read Length

- Paired-end reads substantially facilitate assemblies of all genome sizes.
- Paired-end reads can also help resolve differences among repeat regions and thus can be used in transcriptome projects to distinguish family members as well as identify alternative splicing.
- The most commonly run paired-end reads in the Core are PE100 and PE150 (HiSeq), and PE150 and PE250 (MiSeq).
- For de novo sequencing, PacBio RSII is an excellent alternative or complementary option as this generates average median read lengths of ~5kb with long reads tailing off to 10kb, up to 20kb.

RNA-Seq

- As with array analysis, RNA fragments are reverse transcribed before analysis.
- These DNA fragments are sequenced and then mapped to a reference genome (or assembled if necessary).
- The data that are eventually analyzed are counts of fragments that map to a particular gene, exon, or isoform.
- The starting place is a FASTQ file which reports the sequences and their estimated quality.
- This is a variant of FASTA

FASTA

- FASTA format is a text-based format for representing either nucleotide sequences or peptide sequences, in which nucleotides or amino acids are represented using single-letter codes.
- The format also allows for sequence names and comments to precede the sequences. The format originates from the FASTA software package, but has now become a standard in the field of bioinformatics.
- The simplicity of FASTA format makes it easy to manipulate and parse sequences using text-processing tools and scripting languages like Python, Ruby, and Perl.

- A sequence in FASTA format begins with a single-line description, followed by lines of sequence data.
- The description line is distinguished from the sequence data by a greater-than (">") symbol in the first column (i.e., first byte after a newline).
- The word following the ">" symbol is the identifier of the sequence, and the rest of the line is the description (both are optional). There should be no space between the ">" and the first letter of the identifier.
- It is recommended that all lines of text be shorter than 80 characters. The sequence ends if another line starting with a ">" appears; this indicates the start of another sequence.
- Below is a simple example of one sequence in FASTA format. This is the Asian elephant cytochrome b component of mitochondria protein sequence. L = leucine, C = cysteine, Y = tyrosine, …

```
>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPWGQMSFWGATVITNLFSAIPYIGTNLV
EWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPYYTIKDFLG
LLILILLLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVPNKLGGVLALFLSIVIL
GLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEYPYTIIGQMASILYFSIILAFLPIAGX
IENY
```

FASTQ

- FASTQ format is a text-based format for storing both a biological sequence (usually nucleotide sequence) and its corresponding quality scores.
- Both the sequence letter and quality score are encoded with a single ASCII character each for brevity.
- It was originally developed at the Wellcome Trust Sanger Institute to bundle a FASTA sequence and its quality data, but has recently become the de facto standard for storing the output of high throughput sequencing instruments such as the Illumina Genome Analyzer.

- A FASTQ file normally uses four lines per sequence.
  - Line 1 begins with a '@' character and is followed by a sequence identifier and an optional description (like a FASTA title line).
  - Line 2 is the raw sequence letters.
  - Line 3 begins with a '+' character and is optionally followed by the same sequence identifier (and any description) again.
  - Line 4 encodes the quality values for the sequence in Line 2, and must contain the same number of symbols as letters in the sequence.

- A FASTQ file containing a single sequence might look like this:

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

- The character quality codings are derived from the estimated probability that the base call is wrong. With the ASCII numeric value (33–126, Hex 20–7E) used as the code.
- The codes in order from lowest quality to highest are:

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

- The original Sanger FASTQ files also allowed the sequence and quality strings to be wrapped (split over multiple lines), but this is generally discouraged as it can make parsing complicated due to the unfortunate choice of "@" and "+" as markers (these characters can also occur in the quality string).
- Note that line ends are encoded as CR/LF (0D/0A) [Windows] or CR (0D) [original Apple] or LF (0A) [OS X].

FASTQ Quality Coding

- $Q = -10 \log_{10} p$ (Phred quality score).
  - For $p = 0.01$, $Q = 20$
  - For $p = 0.001$, $Q = 30$
  - For $p = 0.50$, $Q = 3$
  - Scores of more than 40 are unusual.
- Sometimes $Q = -10 \log_{10} p/(1 - p)$
- The exact map from ASCII code to quality score varies.
- If you need it, get the information from the core relating exactly to the equipment used.

Bowtie (part of the Tuxedo suite)

- Bowtie is an ultrafast, memory-efficient short read aligner geared toward quickly aligning large sets of short DNA sequences (reads) to large genomes.
- It aligns 35-base-pair reads to the human genome at a rate of 25 million reads per hour on a typical workstation.
- Bowtie indexes the genome with a Burrows-Wheeler index to keep its memory footprint small: for the human genome, the index is typically about 2.2 GB (for unpaired alignment) or 2.9 GB (for paired-end or colorspace alignment).
- Multiple processors can be used simultaneously to achieve greater alignment speed.
    - Macbook Pro 2014: 4 cores;
    - Mac Pro 2015: 6 cores;
    - Linux Server 2010: 16 processors, 64 cores;
    - Dell Desktop Workstation 2015: 2 processors, 32 cores.

- Bowtie can also output alignments in the standard SAM format, allowing Bowtie to interoperate with other tools supporting SAM, including the SAMtools consensus, SNP, and indel callers.
- Bowtie runs on the command line under Windows, Mac OS X, Linux, and Solaris.
- Bowtie is not a general-purpose alignment tool like MUMmer, BLAST or Vmatch.
- Bowtie works best when aligning short reads to large genomes, though it supports arbitrarily small reference sequences (e.g. amplicons) and reads as long as 1024 bases.
- Bowtie is designed to be extremely fast for sets of short reads where
  - (a) many of the reads have at least one good, valid alignment,
  - (b) many of the reads are relatively high-quality, and
  - (c) the number of alignments reported per read is small (close to 1).
- Bowtie does not yet report gapped alignments; this is future work.

SAM

- SAM (Sequence Alignment/Map) format is a generic format for storing large nucleotide sequence alignments. SAM aims to be a format that:
  - Is flexible enough to store all the alignment information generated by various alignment programs;
  - Is simple enough to be easily generated by alignment programs or converted from existing alignment formats;
  - Is compact in file size;
  - Allows most of operations on the alignment to work on a stream without loading the whole alignment into memory;
  - Allows the file to be indexed by genomic position to efficiently retrieve all reads aligning to a locus.
  - SAM Tools provide various utilities for manipulating alignments in the SAM format, including sorting, merging, indexing and generating alignments in a per-position format.

TopHat

- TopHat is a program that aligns RNA-Seq reads to a genome in order to identify exon-exon splice junctions.
- It is built on the ultrafast short read mapping program Bowtie.
- TopHat runs on Linux and OS X.
- TopHat was designed to work with reads produced by the Illumina Genome Analyzer, although users have been successful in using TopHat with reads from other technologies.

- TopHat can find splice junctions without a reference annotation.
- By first mapping RNA-Seq reads to the genome, TopHat identifies potential exons, since many RNA-Seq reads will contiguously align to the genome.
- Using this initial mapping information, TopHat builds a database of possible splice junctions and then maps the reads against these junctions to confirm them.
- Short read sequencing machines can currently produce reads 100bp or longer but many exons are shorter than this so they would be missed in the initial mapping.
- TopHat solves this problem mainly by splitting all input reads into smaller segments which are then mapped independently. The segment alignments are put back together in a final step of the program to produce the end-to-end read alignments.

- TopHat generates its database of possible splice junctions from two sources of evidence. The first and strongest source of evidence for a splice junction is when two segments from the same read (for reads of at least 45bp) are mapped at a certain distance on the same genomic sequence or when an internal segment fails to map—again suggesting that such reads are spanning multiple exons.
- With this approach, "GT-AG", "GC-AG" and "AT-AC" introns will be found ab initio.
- The second source is pairings of "coverage islands", which are distinct regions of piled up reads in the initial mapping. Neighboring islands are often spliced together in the transcriptome, so TopHat looks for ways to join these with an intron.
- We only suggest users use this second option (--coverage-search) for short reads (< 45bp) and with a small number of reads (≤ 10 million). This latter option will only report alignments across "GT-AG" introns

Issues in Mapping Reads

- Assume we have a reference genome.
- A transcript may not map if the sample has a gene variant included in the read that is not in the reference genome.
- Even if there are only rare errors, they can induce false matches, though usually they will induce failed matches.
- Some reads can map to multiple sites. These can be omitted or allocated to the exons.

- Cufflink and Cuffdiff count reads mapping to a gene by counting reads mapping to each apparent isoform and then adding these up, rather than counting reads mapping to the gene as a whole. It is not clear whether this is better or not.
- In any case, at some point, we will be left with a matrix with one column per sample and one row per gene or exon, and with the entries being either counts or allocated counts, usually the former.

Mapping Coverage

- The human exome is about 30 million bases long (1% of the genome) comprising about 180,000 exons of an average length of 200 bases, in round numbers.
- For genome sequencing, coverage (e.g., 6x) is the ratio of the total read lengths to the length of the genome.
- This is harder to define for RNA-Seq.
- If fragments were uniformly distributed (say by starting point), if an exon is of length 200, and a 50-base read is needed for identification, then the starting point needs to be in the first 150 bases in the exon or in the upstream 50 bases, so the expected number of mapped reads is the coverage.
- With 6x coverage, the chance of no read mapping is around 0.2% per exon.