Next Generation Sequencing: An Overview

Cavan Reilly

April 16, 2018

Table of contents

Next generation sequencing

Over the last 10 years or so there has been rapid development of methods for *next generation sequencing*.

Here is the process for the Illumina technology (one of the major producers of platforms for next generation sequencing).

The biological sample (e.g. a sample of mRNA molecules) is first randomly fragmented into short molecules.

Then the ends of the fragments are adenylated and adaptor oligos are attached to the ends.

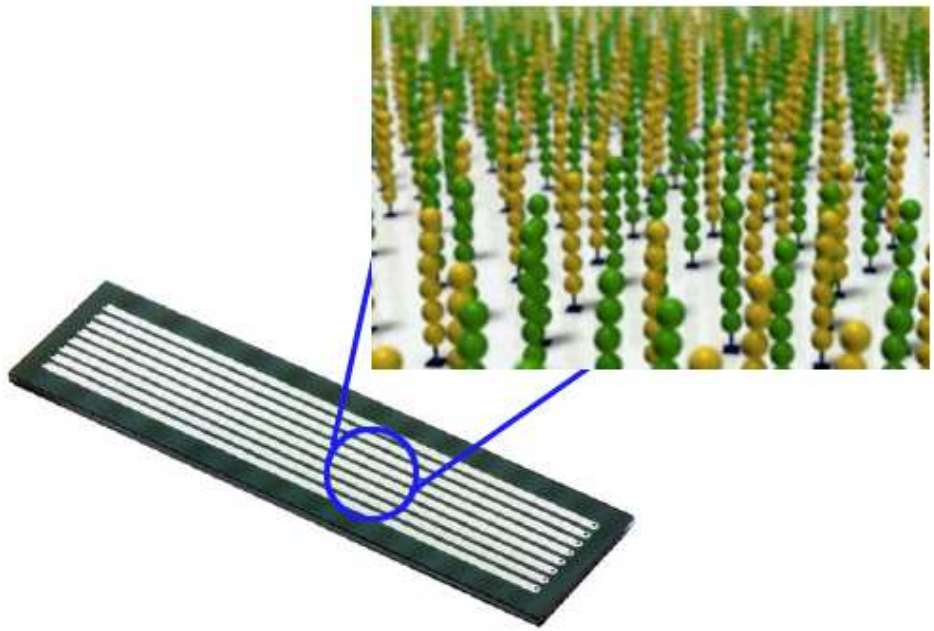
Next generation sequencing

The fragments are then size selected, purified and put on a flow cell.

An Illumina flow cell has 8 lanes and is covered with other oligonucleotides that bind to the adaptors that have been ligated to the fragmented nucleotide molecules from the sample.

The bound fragments are then extended to make copies and these copies bind to the surface of the flow cell.

This is continued until there are many copies of the original fragment resulting in a collection of hundreds of millions of clusters.



Next generation sequencing

The reverse strands are then cleaved off and washed away and sequencing primer is hybridized to the bound DNA.

The individual clusters are then sequenced in parallel, base by base, by hybridizing fluorescently labeled nucleotides.

After each round of extension of the nucleotides a laser excites all of the clusters and a read is made of the base that was just added at each cluster.

If a very short sequence is bound to the flow cell it is possible that the machine will sequence the adaptor sequence-this is referred to as adaptor contamination.

There is also a measure of the quality of the read that is saved along with the read itself.

Next generation sequencing

These quality measures are on the PHRED scale, so if there is an estimated probability of an error of p, the PHRED based score is $-10 \log_{10} p$.

If we fragment someone's DNA we can then sequence the fragments, and if we can then put the fragments back together we can then get the sequence of that person's genome or transcriptome.

This would allow us to determine what alleles this subject has at every locus that displays variation among humans (e.g. SNPs).

Next generation sequencing

There are a number of popular algorithms for putting all of the fragments back together: BWA, Maq, SOAP, ELAND, Bowtie and HISAT.

We'll discuss Bowtie in some detail later (BWA uses the same ideas as Bowtie) and its updated versions, such as HISAT.

ELAND is a proprietary algorithm from Illumina and Maq and SOAP use hash tables and are considerably slower than Bowtie.

Applications

There are many applications of this basic idea:

1. resequencing (DNA-seq)
2. gene expression (RNA-seq)
3. miRNA discovery and quantitation
4. DNA methylation studies
5. ChIP-seq studies
6. metagenomics
7. ultra-deep sequencing of viral genomes

We will focus on resequencing (and SNP calling) and gene expression in this course.

NGS and microarrays

Currently microarrays are not used to study gene expression, almost every researcher would use sequencing based techniques.

Compared to microarrays, RNA-seq

1. higher sensitivity
2. higher dynamic range
3. lower technical variation
4. don't need a sequenced genome (but it helps, a lot)
5. more information about different exon uses (more than 90% of human genes with more than 1 exon have alternative isoforms)

NGS and microarrays

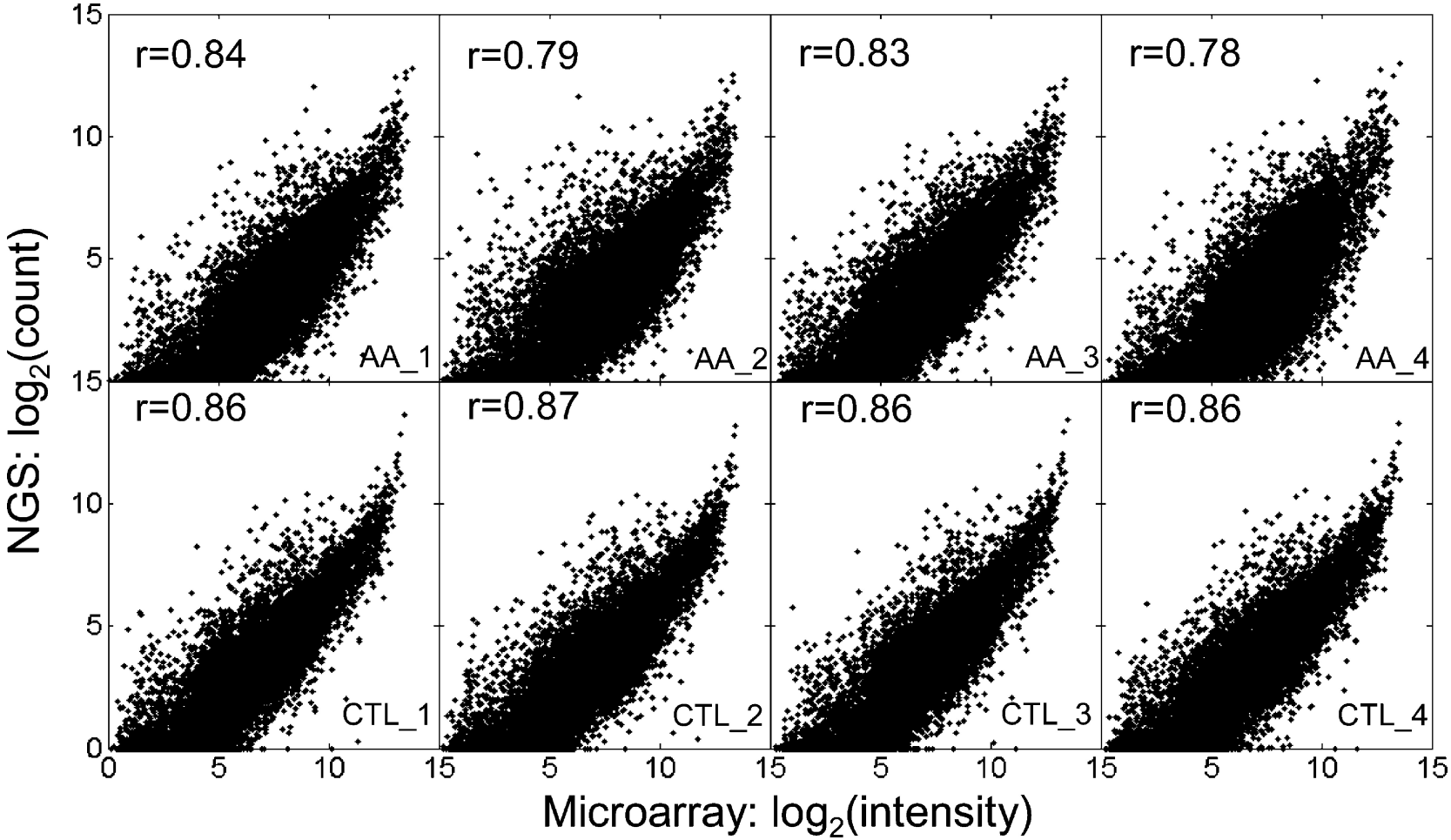
In one head to head comparison, 30% more genes are found to be differentially expressed with the same FDR (Marioni, Mason, Mane, et al. (2008)).

These authors also found that the correlation between normalized intensity values from a microarray and the logarithm of the counts for the same transcript were 0.7-0.8 (this is similar to what others have reported).

The largest differences occurred when the read count was low and the microarray intensity was high which probably reflects cross-hybridization in the microarray.

NGS and microarrays

Others have found slightly higher correlations, here is a figure from a study by Su, Li, Chen, et al. (2011).



Study design

Before addressing the technical details, we will outline some considerations regarding study design specific to NGS technology.

There are several major manufacturers of the technology necessary to generate short reads, and they all have different platforms with resulting differences in the data and processing methods. A few of the major vendors are

1. Illumina, formerly known as Solexa sequencing, they now have MiSeq and HiSeq (the Genome analyzer is the old platform)
2. Roche, which owns 454 Life Sciences, which supports GS FLX+ system
3. Life technologies, which supports Ion-Torrent
4. Pacific Biosciences

Study design

We will focus on the Illumina technology as the University of Minnesota has invested in this platform (the HiSeq 2000 and MiSeq platforms is available in the UMN Genomics Center).

The MiSeq machine produces longer reads than the HiSeq 2000.

We also have some capabilities to do 454 sequencing here.

The 454 platform produces longer reads (up to 800 bp now), and is very popular in the microbiomics literature.

This technology was discontinued in 2013.

Study design

By sequencing larger numbers of fragments one can estimate the sequence more accurately.

The goal of using high sequence *coverage* is that by using more fragments it is more likely that one fragment will cover any random location in the genome.

Clearly one wants to cover the entire genome if the goal is to sequence it, hence the *depth* of coverage is how many overlapping fragments will cover a random location.

By deep sequencing, we mean coverage of 30X to 40X, whereas coverage of 4X or so would be considered low coverage.

Study design

The rule for determining coverage is:

coverage = (length of reads $\times$ number of reads)/ (length of genome)

For a given sequencing budget, there is a tradeoff between the depth of coverage and the number of subjects one can sequence.

The objectives of the study should be considered when determining the depth of coverage.

If the goal is to accurately sequence a tumor cell then deep coverage is appropriate, however if the goal is to identify rare variants, then lower coverage is acceptable.

Study design

For example, the 1000 Genomes project is using about 4X coverage.

A reasonable rule of thumb is that one probably needs 20-30X coverage to get false negatives less than 1% in nonrepeat regions of a duploid genome (Li, Ruan and Durbin, 2008).

If the organism one is sequencing does not have a sequenced genome, the calculations are considerably more complex and one would want greater coverage in this case.

Throughout this treatment we will suppose that the organism under investigation has a sequenced genome.

Study design

Another choice when designing your study regards the use of mate-pairs.

With the Illumina system one can sequence both ends of a fragment simultaneously.

This greatly improves the accuracy of the method and is highly recommended.

If the goal is to characterize copy number variations then getting paired end sequences is crucial.

Quality assessment

We can use the ShortRead package in R to conduct quality assessment and get rid of data that doesn't meet quality criteria.

To explore this we will use some data from dairy cows.

You can find a description of the data along with the actual data in sra format at this site.

`http://trace.ncbi.nlm.nih.gov/Traces/sra/?study=SRP012475`

While the NCBI uses the sra format to store data, the standard format for storing data that arises from a sequencing experiment is the fastq format.

We can convert these files to fastq files using a number of tools-here is how to use one supported at NCBI called fastq-dump.

Quality assessment

To use this you need to download the compressed file, unpack it and then enter the directory that holds the executable and issue the command (more on these steps later).

```
sratoolkit.2.1.15-centos_linux64/bin/fastq-dump SRR490771.sra
```

Then this will generate fastq files which hold the data.

If we open the fastq files we see they hold identifiers and information about the reads followed by the actual reads:

Quality assessment

```
@SRR490756.1 HWI-EAS283:8:1:9:19 length=36
ACGAGCTGANGCGCCGCGGGAAGAGGGCCTACGGGG
+SRR490756.1 HWI-EAS283:8:1:9:19 length=36
AB7A5@A47%8@2@@296263@/.(3'=49<9<8,9
@SRR490756.2 HWI-EAS283:8:1:9:1628 length=36
CCTAATAATNTTTTCCTTTACCCTCTCCACATTAAT
+SRR490756.2 HWI-EAS283:8:1:9:1628 length=36
BCCC@CB>)%)43?BBCB@:>BBCBBBBBB=CB?BC
@SRR490756.3 HWI-EAS283:8:1:9:198 length=36
ATTGAATTTGAGTGTAAACATTCACATAAAGAGAGA
+SRR490756.3 HWI-EAS283:8:1:9:198 length=36
BBB<9ABCA<A2?5>BBB?BBB@BABB>?A*?9?<B
```

The string of letters and symbols encodes quality information for each base in the sequence.

Quality assessment

Then we define the directory where the data is and read in all of the filenames that are there

```
> library(ShortRead)
> dataDir <- "/home/cavan/Documents/NGS/bovineRNA-seq"
> fastqDir <- file.path(dataDir, "fastq")
> fls <- list.files(fastqDir, "fastq$", full=TRUE)
> names(fls) <- sub(".fastq", "", basename(fls))
```

Quality assessment

Then we apply the quality assessment function, qa, to each of the files after we read it into R using the readFastq function

```
> qas <- lapply(seq_along(fls), function(i, fls)
+ qa(readFastq(fls[i]), names(fls)[i]), fls)
```

This can take a long time as it has to read all of these files into R, but note that we do not attempt to read the contents of the files into the R session simultaneously (a couple of objects generated by the call to `readFastq` stored inside R will really bog it down).

Then we rbind the quality assessments together and save to an external location.

```
> qa <- do.call(rbind, qas)
> save(qa, file=file.path("/home/cavan/Documents/NGS/bovineRNA-seq","qa.rda"))
> report(qa, dest="/home/cavan/Documents/NGS/bovineRNA-seq/reportDir")
[1] "/home/cavan/Documents/NGS/bovineRNA-seq/reportDir/index.html"
```

Quality assessment

This generates a directory called reportDir and this directory will hold images and an html file that you can open with a browser to inspect the quality report.

So we now use the browser to do this-we see the data from run 12 is of considerably higher quality than the others but nothing is too bad.

One can implement soft trimming of the sequences to get rid of basecalls that are of poor quality.

A function to do this is as follows (thanks to Jeremy Leipzig who posted this on his blog).

Note that this can require a lot of computational resources for each fastq file.

Quality assessment

```
# softTrim
# trim first position lower than minQuality and all subsequent positions
# omit sequences that after trimming are shorter than minLength
# left trim to firstBase, (1 implies no left trim)
# input:  ShortReadQ reads
# integer minQuality
# integer firstBase
# integer minLength
# output:  ShortReadQ trimmed reads
library("ShortRead")
softTrim <- function(reads, minQuality, firstBase=1, minLength=5){
  qualMat <- as(FastqQuality(quality(quality(reads))),'matrix')
  qualList <-split(qualMat, row(qualMat))
  ends <- as.integer(lapply(qualList, function(x){which(x <
    minQuality)[1]-1}))
  # length=end-start+1, so set start to no more than length+1 to avoid
  # negative-length
  starts <-as.integer(lapply(ends,function(x){min(x+1,firstBase)}))
  # use whatever QualityScore subclass is sent
  newQ <-ShortReadQ(sread=subseq(sread(reads),start=starts,end=ends),
    quality=new(Class=class(quality(reads)),
    quality=subseq(quality(quality(reads)),
    start=starts,end=ends)),id=id(reads))
```

Quality assessment

```
  # apply minLength using srFilter
  lengthCutoff <- srFilter(function(x) { width(x)>=minLength},
    name="length cutoff")
  newQ[lengthCutoff(newQ)]
}
```

To use:

```
> library("ShortRead")
> source("softTrimFunction.R")
> reads <- readFastq("myreads.fq")
> trimmedReads <- softTrim(reads=reads,minQuality=5,firstBase=4,minLength=3)
> writeFastq(trimmedReads,file="trimmed.fq")
```

Quality assessment

When you have mate pair reads things can be more complicated depending on goal.

Some downstream analyses require that the mate pairs are the same length-above procedure can't guarantee that.

Typically use command line tools to do this, e.g. Trimmomatic is popular.

Bowtie background

The keys to understanding how the algorithm implemented by Bowtie are the Burrows Wheeler transform (BWT) and the FM index.

1. the FM index: see Ferragina, P. and Manzini, G. (2000), "Opportunistic data structures with applications"
2. the Burrows Wheeler transform (BWT): see Burrows, M. and Wheeler, D.J. (1994), "A Block-sorting lossless data compression algorithm".

The Burrows Wheeler transform is an invertible transformation of a string (i.e. a sequence of letters).

Computing the BWT

To compute this transform, one

1. determines all rotations of the string
2. sorts the rotations lexicographically to generate an array M
3. saves the last letter of each sorted string in addition to the row of M that corresponds to the original string.