Analysis of high-throughput sequencing data

Simon Anders
EBI
Outline

Overview on high-throughput sequencing (HTS) technologies, focusing on Solexa’s GenomAnalyzer as example
Software requirements to work with HTS software
Use of SolexaPipeline and Maq
Bioconductor support for HTS (Biostrings, ShortRead, TileQC, HilbertCurveView)
Overview

In recent years, new sequencing schemes, also called
• high-throughput sequencing (HTS)
• massively-parallel sequencing
• flow-cell sequencing
have been proposed.

Commerically available are the devices from
• Roche (was: 454)
• Illumina (was: Solexa)
• Applied Biosystems ("SOLiD system")
• Helicos ("HelicoScope")
Core ideas
The two core differences of HTS to Sanger capillary sequencing:

1. The library is not constructed by cloning, but by PCR, where the fragments are separated by physico-chemical means (emulsion PCR or bridge PCR).

2. Very many fragments are sequenced in parallel in a flow cell (as opposed to a capillary), observed by a microscope with CCD camera.
Solexa workflow

- Bridge PCR to prepare "polonies"

- Sequencing: 35 cycles x 4 bases, with micrographs taken in 300 tiles x 8 lanes  ⇒ 1 Terabyte image data

- "SolexaPipeline" ⇒ Sequences and alignment
Solexa: flow cell

Up to eight samples can be loaded onto the flow cell for simultaneous analysis on the Illumina Genome Analyzer.
Solexa: sample preparation

1. PREPARE GENOMIC DNA SAMPLE
   Randomly fragment genomic DNA and ligate adapters to both ends of the fragments.

2. ATTACH DNA TO SURFACE
   Bind single-stranded fragments randomly to the inside surface of the flow cell channels.

3. BRIDGE AMPLIFICATION
   Add unlabeled nucleotides and enzyme to initiate solid-phase bridge amplification.
Solexa: sample preparation

4. Fragments become double stranded.

The enzyme incorporates nucleotides to build double-stranded bridges on the solid-phase substrate.

5. Denature the double-stranded molecules.

Denaturation leaves single-stranded templates anchored to the substrate.

6. Complete amplification.

Several million dense clusters of double-stranded DNA are generated in each channel of the flow cell.
Solexa: sequencing

7. Determine First Base

First chemistry cycle: to initiate the first sequencing cycle, add all four labeled reversible terminators, primers and DNA polymerase enzyme to the flow cell.

8. Image First Base

After laser excitation, capture the image of emitted fluorescence from each cluster on the flow cell. Record the identity of the first base for each cluster.

9. Determine Second Base

Second chemistry cycle: to initiate the next sequencing cycle, add all four labeled reversible terminators and enzyme to the flow cell.
Solexa: sequencing

10. IMAGE SECOND CHEMISTRY CYCLE
After laser excitation, collect the image data as before. Record the identity of the second base for each cluster.

11. SEQUENCE READS OVER MULTIPLE CHEMISTRY CYCLES
Repeat cycles of sequencing to determine the sequence of bases in a given fragment a single base at a time.

12. ALIGN DATA
Align data, compare to a reference, and identify sequence differences.
Roche 454

- presented 2005, first on market
- emulsion PCR
- pyrosequencing
- read length: 250 bp
- paired read separation: 3 kb
- 300 Mb per day
- $84 per Mb
- error rate: around 5% per bp
Illumina/Solexa

- second on market
- bridge PCR
- polymerase-based sequencing-by-synthesis
- read length: 32..40 bp
- paired read separation: 200 bp
- 400 Mb per day
- $6 per Mb (reagents only: $2)
- error rate: 1.5% per bp
Applied Biosystems SOLiD

- third on market (since less than a year)
- emulsion PCR
- ligation-based sequencing
- read length: 35 bp
- paired read separation: 3 kb
- 600 Mb per day
- $6 per Mb  (reagents only: $0.80)
- error rate: <0.1% (manufacturer claim)
Helicos

- new on the market
- no amplification
- single-molecule sequencing-by-synthesis
- read length: 25..45 bp
- 1200 Mb per day
- reagents: $2.50 per Mb
- error rate: <1% (manufacturer claim)

Comparison data:
E Mardis, Trends in Genetics 24 (2008) 133
Use cases for HTS

denovo sequencing of small genomes
transcriptome analysis, ESTs, expression profiling
environmental sampling (metagenomics)
ChIP-Seq, miR-Seq and similar studies
resequencing, SNP finding
reading bar codes

Established procedures for assembly, alignment, counting
statistics etc. may not be suitable.
Coming soon: targeted sequencing

Currently, one always samples the whole genome, which is wasteful if one is interested in only a specific region.

Microarrays allow to select fragments of interest.
Alignment software

Short-read alignment comes with its own challenges, and is best done with specialized software.

- **Eland** (supplied by Solexa)
  - fast, but limited flexibility
- **Maq** (developed by Heng Li at Sanger)
  - slower, but very powerful
  - uses and outputs quality scores
  - graphical browser (Maqview)
- **Biostrings alignment functions** (H. Pages, Seattle)
  - very fast, well integrated in R/Bioconductor
  - not yet released
Assembly software

Standard assembly software for capillary shot-gun sequencing gives unsatisfactory results.

For specialized techniques, see e.g. Velvet (D. Zerbino, E. Birney; EBI), which uses de Bruijn graphs.
R, Perl, Python, or C/C++?

Some thoughts:
• R offers superb statistics capabilities and the Bioconductor infrastructure.
• However, R is hindered by the call-by-value semantics and the lack of persistent, mutable and referenceable objects.
• Perl is popular for text processing. Python offers the same with better language design.
• BioPerl seems closer to HTS needs than Bioconductor (more sequence-level tools) but might be too slow as well.
• C/C++ will always be needed for high-performance components but is too unwieldy for infrastructure.
SWIG allows easy binding of C/C++ to Perl and Python. (R support possible)
Bioconductor support

Bioconductor support for HTS is still at an early stage. So far, we have these packages:

- **Biostrings**: Infrastructure to handle large amounts of character or genomic data; certain alignment functionality (H. Pages)
- **ShortReads***: Reading in data from the SolexaPipeline and Maq (M. Morgan)
- **HilbertCurveDisplay***: Visualization of genomic data (S. An.)
- **TileQC**: raw-data quality control tools (P. Dolan, D. Denver)

* not yet released (available from SVN)
How to provide the data

Processing software may work in a streaming fashion, by reading in all data into memory, or (as a compromise) via a database.

- streaming: most software (SolexaPipeline, Maq, ...)
- SQL database: e.g., TileQC
- work from RAM: e.g., BioStrings, ShortReads
SolexaPipeline

- "Firecrest": Identifying clusters
  ⇒ typically 3..5 mio clusters per lane

- "Bustard": Base calling
  ⇒ sequence for each cluster,
      with Phred-like scores

- "Gerald" and "Eland": Aligning to reference
Firecrest output

Large tab-separated text files with one row per identified cluster, specifying:
- lane index and tile index
- x and y coordinates of cluster on tile
- for each cycle a group of four numbers, specifying the fluorescence intensity for A, C, G, and T.
Bustard output

Two TSV text files, with one row per cluster:

"seq.txt" file:
lane and tile index, x and y coordinates
the called sequence as string of A, C, G, T

"prb.txt" file:
Phred-like scores, ranging from -40 to 40;
one value per called base
Example data

We use data from A. Barski et al.: "High-resolution profiling of histone methylations in the human genome", Cell **129** (2007) 823

We look at their Solexa ChIP-Seq data for H3K4me1 (7 lanes) and H3K4me3 (4 lanes).

NCBI Short Read Archive (SRA), accession number SRA000206.
Data reduction in ChIP-Seq

← raw images
→ reads and quality scores
→ aligned reads
→ filtered aligned reads
→ pile-up vectors
Using Maq: Preparation

Get a reference genome in FASTA format and convert it into one big BFA ("binary FASTA") file with the command `maq fasta2bfa`.

Convert the *_seq.txt and *_prb.txt files produced by Bustard to FASTQ and on to BFQ ("binary FASTQ"). Take care to convert the quality scales correctly. Make sure each BFQ file contains around 1 to 2 mio reads (2 to 5 files per lane).
Using Maq: Mapping

Simply call

```
maq map out.map reference.bfa
   reads1.bfq reads2.bfq ...
```

Use a compute cluster if available. On an LSF system, you just need to prefix `bsub` before the command.

Some rudimentary scripting skills (shell, Perl, or Python) are worth acquiring for this.
FASTQ format

FASTQ = "FASTA with basecall quality scores"

Each read has four lines, e.g.:

@R:2:176:692:881
GGGTCGGCCGTCCAGCGACCTCTC
+
IIIIIIIIIC>I89=/>&)9*=3
## FASTQ quality scores

<table>
<thead>
<tr>
<th>quality score $q$</th>
<th>error prob $p$</th>
<th>characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>! &quot;#$%&amp;'()*</td>
</tr>
<tr>
<td>10</td>
<td>0.1</td>
<td>+ ,-./*01234</td>
</tr>
<tr>
<td>20</td>
<td>0.01</td>
<td>5 6789:;&lt;=&gt;</td>
</tr>
<tr>
<td>30</td>
<td>0.001</td>
<td>? @ABCDEFGH</td>
</tr>
<tr>
<td>40</td>
<td>0.0001</td>
<td>I</td>
</tr>
</tbody>
</table>

**Sanger institute scale:**  
\[ q = -10 \log_{10} p \]  
\[ 0 \leq q \leq 40 \]

**Solexa scale:**  
\[ q = -10 \log_{10} \left( \frac{p}{1-p} \right) \]  
\[ -40 \leq q \leq 40 \]
Mapping output

See text file for an example of an output produced by MAQ.
Strand and fragment direction

Solexa sequencing is not strand-specific. The 'strand' column actually specifies the direction of the fragment.

[Diagram showing '+' and '-' strands with start and length labels]
Biostrings and ShortRead

Let’s load a lane of Solexa data, aligned with Maq, to see these two packages in action.

[ see text file ]
The AlignedRead class

An AlignedRead object holds many short reads with their alignments.

Slots:

- **raw data**
  - id
  - sread
  - quality

- **alignment data**
  - chromosome
  - position
  - strand
  - alignQuality

- **extra info**
  - alignData
XString objects

An XString object holds a potentially very long string within a given alphabet:

BString arbitrary characters
DNAString only IUPAC nucleotide symbols
RNAString only IUPAC nucleotide symbols
AAString only amino acid symbols
XStringView objects

An XStringView object holds subviews of an XString without containing a copy of the string.

Again, we have BStringView, DNASTringView, RNASStringView, and AAStringView.

[ see example in text file ]
XStringSet objects

An XStringView object holds many (typically short) strings.

As before, we have BStringSet, DNASStringSet, RNASSStringSet, and AASStringSet.

The function alphabetByCycle tallies the letters (or nucleotides, or amino acids) by position within the string.

[ see example in text file ]
> barplot( rowSums( alphabetByCycle( quality( ar.me1$run4_lane8 ) )[ 33 + 0:40, ] ) )
A "pile-up" is an integer vector with one element per base pair in a chromosome, tallying the number of reads (or fragments) mapping onto each base pair. It is the essential intermediate data type in ChIP-Seq studies. Having uniform fragment lengths is useful. Paired-end reads are even better.

[Show Maqview to demonstrate.]
Pile-up plot for chromosome 10

H3K4me1

H3K4me3
Hilbert Curve Visualization

The Hilbert curve is a way to "fold up" a one-dimensional line such that it fills a two-dimensional square.

This allows for a detailed "bird’s eye view" onto, e.g., pile-up data.
Hilbert curve, level 1
Hilbert curve, level 2
Hilbert curve, level 3
Hilbert curve, level 4
Hilbert curve, level 5
Pile-up plot for chromosome 10

H3K4me1

H3K4me3
Hilbert plots of chromosome 10:

H3K4me1  H3K4me3
GUI for Hilbert curve display
Callback for drill-down

When clicking on a pixel, a user-supplied R function may be called. This function is provided information on the position of the clicked pixel and the portion of the data vector surrounding it.

This function may produce a linear or custom plot, or do customized statistics or exploration.
Zooming in with GenomeGraphs

H3K4me1

H3K4me3
GenomeGraphs (S. Durrick) produces genomic plots using biomaRt to get data from Ensembl.
3-color Hilbert plot

red:  
    H3K4me1

green:  
    H3K4me3

blue:  
    exons
Pile-up around TSSs:

only data from chromosome 10

black: H3K4me1
red: H3K4me3
TileQC: Example
Bioconductor wishlist

- Common data structure to represent output of all popular short-read alignment tools
- Tools to perform quality-control statistics on base-call and alignment quality scores
- Calculations for depth and coverage.
- Data reduction functions (pile-up etc.)
- Genomic visualization tools
- Peak finders, peak validity statistics
- Output filters to write GFF, BED, etc.
Conclusion

- Analysis of HTS data requires novel tools.
- Much software is being developed at the moment.
- The Bioconductor project is getting ready for HTS.

If you are interested in HTS analysis with Bioconductor, subscribe to the "bioc-sig-sequencing" mailing list.