DNA Sequencing

Ben Langmead

Please sign guestbook (www.langmead-lab.org/teaching-materials) to tell me briefly how you are using the slides. For original Keynote files, email me (ben.langmead@gmail.com).
A profound implication of the central dogma is that nearly all the information necessary to construct and operate a living thing is contained in its DNA.\textsuperscript{2} We call the complete complement of DNA (and therefore the collection of all the genes) in a particular species its genome. That is why genome sequencing projects, which determine the exact sequence of all the DNA in an organism, are so important.

Genomics technology

Sanger DNA sequencing
1977-1990s

DNA Microarrays
Since mid-1990s

2nd-generation DNA sequencing
Since ~2007

3rd-generation & single-molecule DNA sequencing
Since ~2010

Fred Sanger
1918-2013

“Chain termination” sequencing
Sanger sequencing

First practical method invented by Fred Sanger in 1977. Initially used to sequence shorter genomes, e.g. viral genomes 10,000s of bases long.
Sanger sequencing

From "DNA" documentary, episode 3
Genomics technology

- **Sanger DNA sequencing**: 1977-1990s
- **DNA Microarrays**: Since mid-1990s
- **2nd-generation DNA sequencing**: Since ~2007
- **3rd-generation & single-molecule DNA sequencing**: Since ~2010
No sequencing technology yet invented can read much more than 10,000 nucleotides at a time with reasonable cost, throughput, accuracy.

Instead, there’s a vigorous race to see whose sequencer can read “short” fragments of DNA (around 100s of nucleotides) with best cost, throughput, accuracy.

Decoding DNA With Semiconductors
By NICHOLAS WADE
Published: July 20, 2011

Cost of Gene Sequencing Falls, Raising Hopes for Medical Advances
By JOHN MARKOFF
Published: March 7, 2012

Company Unveils DNA Sequencing Device Meant to Be Portable, Disposable and Cheap
By ANDREW POLLACK
Published: February 17, 2012

Source: nytimes.com
Sequencing

Since 2005, many DNA sequencing instruments have been described and released. They are based on a few different principles:

- Synthesis / ligation
- SMRT cell
- Nanopore

Sequencing by synthesis (“massively parallel sequencing”) provides greatest throughput, and is the most prevalent today.

DNA: double helix


TCACACTGAGCGGTGCTG
Your genome

CGTCTGGGGGGTATGCACGCGATAGCATTGCGAGACGCTGGAGCCGGAGCACCCTATGTCGCAGTATCTGTCTTTGATTCCTG

Reads

GTATGCACGCGATAG TATGTCGCAGTATCT CACCCATATGTCGCAG GAGACGCTGGAGCCG
Your genome:

CGTCTGGGGGGTATGCACGCGATAGCATTGCGAGACGCTGGAGCCGGAGCACCCTATGTCGCAGTATCTGTCTTTGATTCCTG

Reads:

GTATGCACGCGATAG  TATGTCGCAGTATCT  CACCCATATGTCGAGC  GAGACGCTGAGCCCG

TAGCATTGCGAGACG  GGTATGCACGCGATA  TGGAGCCGGAGCACC  CGCTGGAGCCGGAGC

CGTCTGGGGGGTATGCACGCGATAGCATTGCGAGACGCTGGAGCCGGAGCACCCTATGTCGCAGTATCTGTCTTTGATTCCTG
Reads

Your genome
Reads

Your genome

CGTCTGGGGGGTATGCACGCGATAGCATTGCGAGACGCTGGAGCCGGAGCACCCTATGTCGCAGTATCTGTCTTTGATTCCTG

GTATGCACGCAGTATGCTCGAGTATCT CACCCTATGTCGCAGTCAGA GAGACGCTGGAGGCCCG
TAGCATTCGCGAGAGCG TGGATGCAGCAGACGCTGGAGCCGGAGCACCCTATGTCGCAGTATCTGTCTTTGATTCCTG
TGTCCTGGATTTCTG CGCAGATAGCATATGCAGCTCTGACCCTATGTCGCAGTATCTGTCTTTGATTCCTG
GACGTGGAGCGAGCA GCACCCTATAGTCCGAGA GTATGCACGCGATAGCATTGCGAGACGCTGGAGCCGGAGCACCCTATGTCGCAGTATCTGTCTTTGATTCCTG
CGAGAACGCTTTGCG CATATGTCGATACT CACCCTATGTCGCAGTATCTGTCTTTGATTCCTG
CTACGTTCAATATT GCACCTACGTCTAC GACGTGGAGCGAGCA GCACCCTATAGTCCGAGA GTATGCACGCGATAGCATTGCGAGACGCTGGAGCCGGAGCACCCTATGTCGCAGTATCTGTCTTTGATTCCTG
GATCAGCCAGTCTATCATACCTTGAACCGACGGAGCTCT CACCCTATGTCGCAGTATCTGTCTTTGATTCCTG
CGAGAACGCTTTGCG CATATGTCGATACT CACCCTATGTCGCAGTATCTGTCTTTGATTCCTG
CTACGTTCAATATT GCACCTACGTCTAC GACGTGGAGCGAGCA GCACCCTATAGTCCGAGA GTATGCACGCGATAGCATTGCGAGACGCTGGAGCCGGAGCACCCTATGTCGCAGTATCTGTCTTTGATTCCTG
GATCAGCCAGTCTATCATACCTTGAACCGACGGAGCTCT CACCCTATGTCGCAGTATCTGTCTTTGATTCCTG

Your genome

CGTCTGGGGGGTATGCACGCGATAGCATTGCGAGACGCTGGAGCCGGAGCACCCTATGTCGCAGTATCTGTCTTTGATTCCTG
Reads

Your genome
Reads

Your genome
Double stranded DNA (double helix)  Double stranded DNA (lego version)
Single stranded templates
Input DNA
CCATAGTATATCTCGGCTCTAGGCCCTCATTTTTT
CCATAGTATATCTCGGCTCTAGGCCCTCATTTTTT
CCATAGTATATCTCGGCTCTAGGCCCTCATTTTTT
CCATAGTATATCTCGGCTCTAGGCCCTCATTTTTT

Cut into snippets
CCATAGTA TATCTCGG CTCTAGGCCCTC ATTTTTTT
CCA TAGTATAT CTCGGCTCTAGGCCCTCA TTTTTTT
CCATAGTAT ATCTCGGCTCTAG GCCCTCA TTTTTTT
CCATAG TATATCTCT CGGCTCTAGGCCCT CATTTTTTT

Deposit on slide

Template (billions of them!)
DNA polymerase

“Terminator”
Remove terminators
DNA polymerase

Repeat!
Sequencing by synthesis
Sequencing by synthesis

Cycle 1

Cycle 2

Cycle 3

Cycle 4

Cycle 5

Cycle 6

complement

complement

complement

complement

complement

complement

G

A

T

A

C

C

C

A

T

G

C

C

C

A

T

G
Sequencing by synthesis

Actual Illumina HiSeq 3000 image

http://dnatech.genomecenter.ucdavis.edu/2015/05/07/first-hiseq-3000-data-download/
Sequencing by synthesis

Billions of templates on a slide

Massively parallel: photograph captures all templates simultaneously

Terminators are “speed bumps,” keeping reactions in sync
Cluster of clones
Unterminated

Ahead of schedule

Unterminated
\[ Q = -10 \cdot \log_{10} p \]

- Base quality
- Probability that base call is incorrect

- \( Q = 10 \) → 1 in 10 chance call is incorrect
- \( Q = 20 \) → 1 in 100
- \( Q = 30 \) → 1 in 1,000
Call: orange (C)

Estimate $p$, probability incorrect:
non-orange light / total light

$p = 3 \text{ green} / 9 \text{ total} = 1/3$

$Q = -10 \log_{10} \frac{1}{3} = 4.77$
A read in FASTQ format

<table>
<thead>
<tr>
<th>Name</th>
<th>Sequence (ignore)</th>
<th>Base qualities</th>
</tr>
</thead>
<tbody>
<tr>
<td>@ERR194146.1</td>
<td>ACATCTGGTTTACTCTTCAAGGCCATAAAGCCTAAATAGCCCACACGTTCCCCTTTAAAT</td>
<td>?@@FFBFFDDHHBCEAFEGIIDHGH@GDHHHGEHID@C?GGDG@FHIHG@FHBEG:G</td>
</tr>
</tbody>
</table>
FASTQ

$ head -20 SRA_HISEQ2000_FC1.shuffle.2M.1.fastq
@509.6.64.20524.149722
AGCTCTGGTAGACCATGGGAGCTAGCGCTTAGGACCTTCCTCCACCCCTGAATATGCTTCCTTGCTGNTTGCTGAACTATGGAGAAGCGTTTATTAT
+
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
@509.4.62.19231.2763
GGTTGATAAGCAAGCTCTATTTTTGTCATATACCTGTTGTCTCTGATTCTCTGCTGATCGTGAAGTGCGCGGNTCTAGCTGACAGCACTCTTTTGATCTCATT
+
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
@509.6.47.3027.76579
CCTTTTCGACTAGAGCTGCAAGTGCCAAATATTCACCTGTCAAGTACTAACAAGAGTGTTTTCAAAAATCTGCATAATCAAAAGAAAAATGTTCAACTCTT
+
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
@509.2.7.2951.186312
AAAGATACAACTACCACTCTTTGAGACACACCTAAGCATAAGGACGTTAAGAGAAAATTAAATAGCACATAATGCGCCACATCAAAAAGTTAGA
+
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
@509.6.25.8102.140546
GGCACATTCACACACCTACCTCATTGACATCCTTCCACATCGTTCAAGAGATGCTCAACCAAGAAAGATCTGGANTCAGAGACACACAGCTGATTACATACCTACGTTT
+
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
$
Base qualities

Bases and qualities line up:

AGCTCTGGTGACCCATGGGCAGCTGCTAGGGA

HHHHHHHHHHHHHHHHHHHHHHGCGC5FEFFFGHHHHHHH

Base quality is ASCII-encoded version of \( Q = -10 \log_{10} p \)
| ASCII | Character | Decimal | Hexadecimal | Symbol
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>&lt;NUL&gt;</td>
<td>32</td>
<td>00101000</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>&lt;SOH&gt;</td>
<td>33</td>
<td>00101001</td>
<td>!</td>
</tr>
<tr>
<td>2</td>
<td>&lt;STX&gt;</td>
<td>34</td>
<td>00101001</td>
<td>&quot;</td>
</tr>
<tr>
<td>3</td>
<td>&lt;ETX&gt;</td>
<td>35</td>
<td>00101010</td>
<td>#</td>
</tr>
<tr>
<td>4</td>
<td>&lt;EOT&gt;</td>
<td>36</td>
<td>00101011</td>
<td>$</td>
</tr>
<tr>
<td>5</td>
<td>&lt;ENQ&gt;</td>
<td>37</td>
<td>00101100</td>
<td>%</td>
</tr>
<tr>
<td>6</td>
<td>&lt;ACK&gt;</td>
<td>38</td>
<td>00101101</td>
<td>&amp;</td>
</tr>
<tr>
<td>7</td>
<td>&lt;BEL&gt;</td>
<td>39</td>
<td>00101110</td>
<td>'</td>
</tr>
<tr>
<td>8</td>
<td>&lt;BS&gt;</td>
<td>40</td>
<td>00101111</td>
<td>(</td>
</tr>
<tr>
<td>9</td>
<td>&lt;TAB&gt;</td>
<td>41</td>
<td>00101111</td>
<td>)</td>
</tr>
<tr>
<td>10</td>
<td>&lt;LF&gt;</td>
<td>42</td>
<td>00101110</td>
<td>+</td>
</tr>
<tr>
<td>11</td>
<td>&lt;VT&gt;</td>
<td>43</td>
<td>00101101</td>
<td>-</td>
</tr>
<tr>
<td>12</td>
<td>&lt;FF&gt;</td>
<td>44</td>
<td>00101100</td>
<td>.</td>
</tr>
<tr>
<td>13</td>
<td>&lt;CR&gt;</td>
<td>45</td>
<td>00101011</td>
<td>/</td>
</tr>
<tr>
<td>14</td>
<td>&lt;SO&gt;</td>
<td>46</td>
<td>00101010</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>&lt;SI&gt;</td>
<td>47</td>
<td>00101001</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>&lt;DLE&gt;</td>
<td>48</td>
<td>00101000</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>&lt;DC1&gt;</td>
<td>49</td>
<td>00101001</td>
<td>3</td>
</tr>
<tr>
<td>18</td>
<td>&lt;DC2&gt;</td>
<td>50</td>
<td>00101010</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>&lt;DC3&gt;</td>
<td>51</td>
<td>00101011</td>
<td>5</td>
</tr>
<tr>
<td>20</td>
<td>&lt;DC4&gt;</td>
<td>52</td>
<td>00101100</td>
<td>6</td>
</tr>
<tr>
<td>21</td>
<td>&lt;NAK&gt;</td>
<td>53</td>
<td>00101101</td>
<td>7</td>
</tr>
<tr>
<td>22</td>
<td>&lt;SYN&gt;</td>
<td>54</td>
<td>00101110</td>
<td>8</td>
</tr>
<tr>
<td>23</td>
<td>&lt;ETB&gt;</td>
<td>55</td>
<td>00101111</td>
<td>9</td>
</tr>
<tr>
<td>24</td>
<td>&lt;CAN&gt;</td>
<td>56</td>
<td>00100100</td>
<td>&lt;</td>
</tr>
<tr>
<td>25</td>
<td>&lt;EM&gt;</td>
<td>57</td>
<td>00100101</td>
<td>=</td>
</tr>
<tr>
<td>26</td>
<td>&lt;SUB&gt;</td>
<td>58</td>
<td>00100110</td>
<td>&gt;</td>
</tr>
<tr>
<td>27</td>
<td>&lt;ESC&gt;</td>
<td>59</td>
<td>00100111</td>
<td>?</td>
</tr>
<tr>
<td>28</td>
<td>&lt;FS&gt;</td>
<td>60</td>
<td>00101000</td>
<td>[</td>
</tr>
<tr>
<td>29</td>
<td>&lt;GS&gt;</td>
<td>61</td>
<td>00101001</td>
<td>\</td>
</tr>
<tr>
<td>30</td>
<td>&lt;RS&gt;</td>
<td>62</td>
<td>00101010</td>
<td>]</td>
</tr>
<tr>
<td>31</td>
<td>&lt;US&gt;</td>
<td>63</td>
<td>00101011</td>
<td>^</td>
</tr>
</tbody>
</table>
Base qualities

Usual ASCII encoding is “Phred+33”:

take Q, rounded to integer, add 33, convert to character

```
def QtoPhred33(Q):
    """ Turn Q into Phred+33 ASCII-encoded quality """
    return chr(int(round(Q)) + 33)
```

(converts character to integer according to ASCII table)

```
def phred33ToQ(qual):
    """ Turn Phred+33 ASCII-encoded quality into Q """
    return ord(qual) - 33
```

(converts integer to character according to ASCII table)