Sequence Assembly and Next Generation Sequencing Informatics
CPBS7711
Oct 14, 2014

Sonia Leach, PhD
Assistant Professor
Center for Genes, Environment, and Health
National Jewish Health
sonia.leach@gmail.com
## Applications of next-generation DNA sequencing

<table>
<thead>
<tr>
<th>Method</th>
<th>Sequencing to determine:</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNA-Seq</td>
<td>A genome sequence</td>
</tr>
<tr>
<td>Targeted DNA-Seq</td>
<td>A subset of a genome (for example, an exome)</td>
</tr>
<tr>
<td>Methyl-Seq</td>
<td>Sites of DNA methylation, genome-wide</td>
</tr>
<tr>
<td>Targeted methyl-Seq</td>
<td>DNA methylation in a subset of the genome</td>
</tr>
<tr>
<td>DNase-Seq, Sono-Seq and FAIRE-Seq</td>
<td>Active regulatory chromatin (that is, nucleosome-depleted)</td>
</tr>
<tr>
<td>MAINE-Seq</td>
<td>Histone-bound DNA (nucleosome positioning)</td>
</tr>
<tr>
<td>ChIP-Seq</td>
<td>Protein-DNA interactions (using chromatin immunoprecipitation)</td>
</tr>
<tr>
<td>RIP-Seq, CLIP-Seq, HITS-CLIP</td>
<td>Protein-RNA interactions</td>
</tr>
<tr>
<td>RNA-Seq</td>
<td>RNA (that is, the transcriptome)</td>
</tr>
<tr>
<td>FRT-Seq</td>
<td>Amplification-free, strand-specific transcriptome sequencing</td>
</tr>
<tr>
<td>NET-Seq</td>
<td>Nascent transcription</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Sequencing to determine:</th>
</tr>
</thead>
<tbody>
<tr>
<td>HI-C</td>
<td>Three-dimensional genome structure</td>
</tr>
<tr>
<td>Chia-PET</td>
<td>Long-range interactions mediated by a protein</td>
</tr>
<tr>
<td>Ribo-Seq</td>
<td>Ribosome-protected mRNA fragments (that is, active translation)</td>
</tr>
<tr>
<td>TRAP</td>
<td>Genetically targeted purification of polysomal mRNAs</td>
</tr>
<tr>
<td>PARS</td>
<td>Parallel analysis of RNA structure</td>
</tr>
<tr>
<td>Synthetic saturation mutagenesis</td>
<td>Functional consequences of genetic variation</td>
</tr>
<tr>
<td>Immuno-Seq</td>
<td>The B-cell and T-cell repertoires</td>
</tr>
<tr>
<td>Deep protein mutagenesis</td>
<td>Protein binding activity of synthetic peptide libraries or variants</td>
</tr>
<tr>
<td>PhIT-Seq</td>
<td>Relative fitness of cells containing disruptive insertions in diverse genes</td>
</tr>
</tbody>
</table>

http://www.nature.com/nbt/journal/v30/n11/full/nbt.2421.html
seq (From Costello lectures)
Introduction

• Sequencing technologies
  – 1st Gen: Gel based sequencing and Capillary electrophoresis
  – 2nd Gen: Next-Generation Sequencers:
    • Sequencing by Synthesis: Pyrosequencing with ePCR (Roche 454), Reversible Dye-termination with cluster/bridge amplification (Illumina Solexa), Semiconductor (Ion Torrent PGM/Proton)
    • Sequencing by Ligation: ePCR and diColor system (ABI SOLiD)
  – 3rd Generation Sequencers:
    • Single molecule sequencing (ABI SMS, PacBio SMRT, Helicos), nanopore sequencing, ...

• Sequencing applications
  – ChIP Seq – transcription factor binding and chromatin modification
  – RNA Seq – transcript detection (boundaries, novel exons), quantification
  – Whole Genome/Exome Sequencing - sequence variant detection

• De novo assembly versus mapping to reference sequence
  – Human Genome Project (Hierarchical versus Shotgun Sequencing)
    • Contig assembly and ordering
  – Mapping/Alignment Programs
    • Repetitive regions (interspersed repeats (SINEs, LINEs, LTR elements, and DNA transposons), satellite sequences, and low-complexity sequences)
    • Polymorphisms and structural variants (insertions, deletions, substitutions, inversions, translocation)
Early Sequencing Technologies (1977)

• Maxam-Gilbert Sequencing
  – double-stranded DNA segment ends labeled with $^{32}$P, divided into 4 samples, treated with chemical specific to 1-2 of 4 bases to nick DNA at few sites, treated with piperidine to destroy nicked base, creates fragments of length from $^{32}$P to nicked base, run fragments in 4 lanes on acrylamide gel to separate by size, auto-radiograph, read sequence off gel
  – fallen out of favor due to technical complexity prohibiting its use in standard molecular biology kits, extensive use of hazardous chemicals, and difficulties with scale-up

• Sanger Sequencing (chain termination method)
  – single-stranded DNA template and short primer for one end, add polymerase with normal bases (dNTPs) and special dideoxynucleotides (ddNTPs) that cannot extend chain at certain ratio to ensure variable length fragments, each ending with ddNTP is incorporated. Create one sample with each ddNTP flavor, run in 4 lanes on gel, auto-radiograph and read sequence off gel

• Gilbert and Sanger shared 1980 Nobel Prize in Chemistry for sequencing
Maxam-Gilbert Sequencing

Sanger Sequencing

From Recombinant DNA, 2nd Ed, Watson et al
Evolution of Sequencing

Gel Sequencing and radioactive tags

Capillary Sequencing and fluorescent tags

Adapted from D. Pollock
Base call Quality

- Used to assess sequence quality and determine accurate consensus sequence (originally from Phred program 1998).
- Quality scores $Q$ logarithmically linked to base-calling error probabilities $P$:
  $$Q = -10 \log_{10} P$$
- Thus 10 means 1 in 10 probability base called incorrectly, 20 means 1 in 100 (or 99% base call accuracy), 30 means 1 in 1000 (or 99.9%). Typical Q20 or Q30 for ‘high quality’ bases (max out at 35-40, <10 unusable)
Rate of sequencing over past 30 years

Evolution of Sequencing

Adapted from D. Pollock
The Next-Generation Process

1. DNA Prep
   - Randomly shear DNA + end repair + size select

2. Library Prep
   - Append sequencing adapters

3. Chip Prep
   - Layout of library on sequencing slide or wells

4. Sequencing
   - For each library fragment – determine the order and identity of bases at either end of the fragment

5. Raw Analysis
   - Image processing + base calling
     - Base calls + associated quality: Fastq/BAM
Illumina GAII & Friends

- 'PCR on a chip' - primer-ligated fragments (1 on each end) hybridize to slide which has complement of both primers, DNA extended, original molecule denatured and washed away. Extended molecule bends and hybridizes to 2nd PCR primer (bridge amplification). Molecule extended on bridge, denatured, rebridged, extended, denatured, etc for 35 cycles, slide contains 'cluster' of double-stranded bridges - denatured
  - For excellent graphical explanation of bridge amplification: http://seq.molbiol.ru/sch_clon_ampl.html

- Sequencing by Synthesis, with reversible termination so have competitive incorporation of all 4 bases per cycle, compile images across cycles to get sequence (note: fixed length sequences)
  - Technical issues: dephasing, error increases with length, polymerase problems with transversions (pyrimidine/purine) /transitions (A/G or C/T)

http://youtu.be/45vNetkGspQ (3:40-6:00)
http://www.youtube.com/watch?v=HtuUFUnYB9Y (with Techs) (Aidan Flynn)
In four-channel SBS, four images are necessary to capture the unique fluorescent dyes for each base. In contrast, two-channel SBS requires only two images to determine all four base calls.
Roche 454

• Adapter-ligated DNA fragments (400-500 bp) fixed to small DNA-capture beads in a water-in-oil emulsion for **emulsion-based PCR** in **PicoTiterPlate**, a fiber optic chip, mixed with Enzyme Beads containing also **ATP sulfurylase + luciferase**.

• 4 DNA nucleotides added sequentially in fixed order, extending DNA strand, 'sequencing-by-synthesis’ (pyrosequencing) whereby pyrophosphate released upon nucleotide incorporation. Signal strength proportional to number of nucleotides - linear up to 8 **homopolymers** (3.3% error A5, ~50% for A8)

Center for Genes, Environment, and Health

Roche / 454 FLX

http://www.youtube.com/watch?v=bFNjxKHP8Jc
Roche 454 Process Overview

1) Prepare Adapter Ligated ssDNA Library
2) Clonal Amplification on 28 μ beads
3) Load beads and enzymes in PicoTiterPlate™
4) Perform Sequencing by synthesis on the 454 Instrument

http://mammoth.psu.edu/howToSeqMammoth.html
Example of a Flowgram
Roche 454

Key sequence = TCAG for identifying wells and calibration

http://mammoth.psu.edu/howToSeqMammoth.html
Roche 454 Acyclic Flow (2013)

- As sequencing proceeds, failure to incorporate nucleotide in given flow cycle means have to wait for next flow cycle to finish, thereby missing all other bases in flow, causing bead to go out of phase
- Now, not flow cycle, but flow group
  - ACGAGCTCGTGCTAGTATGACACTCATG
  - Not T Not A Not C Not G

of repeated set of bases (minus 1) to make sure get all other bases before BASE, allows cycles to ‘reset’ flowgrams over bead, and informatics can correct, allowing longer sequence reads
Ion Torrent PGM/Proton

http://www.youtube.com/watch?v=MxkYa9XcvBQ (0:40-2:23)

- Adapter-ligated DNA fragments (400-500 bp) fixed to small DNA-capture beads in a water-in-oil emulsion for **emulsion-based PCR** in Ion Chip, a semiconductor chip.

- 4 DNA nucleotides added sequentially in fixed order, extending DNA strand, semiconductor-sequencing whereby hydrogen H\(^+\) released upon nucleotide incorporation, which changes well pH, and measured by voltmeter. Signal strength proportional to number of nucleotides.

Example Ion Torrent PGM Output

Run Summary

Unaligned

- **706M** Total Bases
- **83** Key Signal
- **83%** Bead Loading

ISP Density

- **6,060,871** Total Reads
- **65%** Usable Sequence

ISP Summary

- 83% Loading 9,406,057 17% Empty Wells
- 99% Enrichment 9,325,915 1% No Template
- 72% Clean 6,739,789 28% Polycytosil
- 90% Final Library 6,090,871 1% Test Fragments 1% Adapter Dimer 0% Low Quality

Read Length

- **116 bp** Mean Read Length

Center for Genes, Environment, and Health
Are homopolymers a problem, really?

T-cell receptor alpha chain (courtesy John Kappler NJH)

- **Somatic mutation** of TCR genes from bone marrow **germline template** means potentially different VJ combination in *every* T-cell
- Need long sequences (>200bp) to see each VJ combo in mRNA (must map V,J separately to ‘reference’, each mRNA also contains random 2-15bp sequence between V-J)
- (Also happens in immunoglobulin genes of B-cells…)
Are homopolymers a problem, really?

- Representative sequences for Jα
- Note high sequence similarity among different Jα and many homopolymer stretches
## Histogram of Homopolymer Lengths

454 TCR RNA-seq

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8011010</td>
<td>6720315</td>
<td>7886821</td>
<td>8369939</td>
</tr>
<tr>
<td>2</td>
<td>1231699</td>
<td>1699846</td>
<td>2103490</td>
<td>1795892</td>
</tr>
<tr>
<td>3</td>
<td>601868</td>
<td>478223</td>
<td>643908</td>
<td>379664</td>
</tr>
<tr>
<td>4</td>
<td>124496</td>
<td>81120</td>
<td>141134</td>
<td>146001</td>
</tr>
<tr>
<td>5</td>
<td>3269</td>
<td>26830</td>
<td>18969</td>
<td>204361</td>
</tr>
<tr>
<td>6</td>
<td>46</td>
<td>1040</td>
<td>14769</td>
<td>10358</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>493</td>
<td>20378</td>
<td>453</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>65</td>
<td>14603</td>
<td>24</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>16</td>
<td>6437</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>7</td>
<td>1223</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>4</td>
<td>261</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>140</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>44</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Ion Torrent ChIPSeq

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>105103373</td>
<td>89525821</td>
<td>89531100</td>
<td>104702528</td>
</tr>
<tr>
<td>2</td>
<td>21761365</td>
<td>23225902</td>
<td>23701785</td>
<td>21410250</td>
</tr>
<tr>
<td>3</td>
<td>11239514</td>
<td>3803885</td>
<td>4040473</td>
<td>10786013</td>
</tr>
<tr>
<td>4</td>
<td>6741716</td>
<td>863941</td>
<td>939610</td>
<td>5952429</td>
</tr>
<tr>
<td>5</td>
<td>2615588</td>
<td>169608</td>
<td>198168</td>
<td>2277292</td>
</tr>
<tr>
<td>6</td>
<td>581643</td>
<td>20948</td>
<td>39730</td>
<td>505266</td>
</tr>
<tr>
<td>7</td>
<td>84868</td>
<td>4666</td>
<td>8725</td>
<td>91205</td>
</tr>
<tr>
<td>8</td>
<td>51957</td>
<td>1299</td>
<td>2967</td>
<td>50248</td>
</tr>
<tr>
<td>9</td>
<td>17012</td>
<td>824</td>
<td>2019</td>
<td>18384</td>
</tr>
<tr>
<td>10</td>
<td>15998</td>
<td>547</td>
<td>1989</td>
<td>15408</td>
</tr>
<tr>
<td>11</td>
<td>10615</td>
<td>310</td>
<td>1557</td>
<td>10267</td>
</tr>
<tr>
<td>12</td>
<td>7014</td>
<td>159</td>
<td>1078</td>
<td>7010</td>
</tr>
<tr>
<td>13</td>
<td>5330</td>
<td>73</td>
<td>715</td>
<td>5253</td>
</tr>
<tr>
<td>14</td>
<td>4300</td>
<td>22</td>
<td>410</td>
<td>3985</td>
</tr>
<tr>
<td>15</td>
<td>3287</td>
<td>3</td>
<td>226</td>
<td>3016</td>
</tr>
<tr>
<td>16</td>
<td>2529</td>
<td>5</td>
<td>131</td>
<td>2292</td>
</tr>
<tr>
<td>17</td>
<td>2041</td>
<td>0</td>
<td>71</td>
<td>1599</td>
</tr>
<tr>
<td>18</td>
<td>1462</td>
<td>1</td>
<td>53</td>
<td>1204</td>
</tr>
<tr>
<td>19</td>
<td>1168</td>
<td>1</td>
<td>42</td>
<td>918</td>
</tr>
<tr>
<td>20</td>
<td>1047</td>
<td>0</td>
<td>29</td>
<td>718</td>
</tr>
<tr>
<td>21</td>
<td>925</td>
<td>5</td>
<td>42</td>
<td>708</td>
</tr>
<tr>
<td>22</td>
<td>891</td>
<td>3</td>
<td>12</td>
<td>1114</td>
</tr>
</tbody>
</table>
Ability of mappers to correct

- TMAP CIGAR OP from BAM file

Ex: ~100bp, 31 CIGAR ops (Loc: chr7:140,842,278-140,845,266)

15M1I4M1I8M1I6M1I12M2I4M3I8M2I4M2I4M1I4M1I5M2I6M1I16M1D12M1D34M1D20M

ATCAGTGATCAAGAGAAATACCACCCACAGGGCTTTCCCCCACAAGGACCACAAATCCTG
GAAGTAGAGGGCCAATTTCCTTCAAAACTGGGGATTACCCCTCTTTTCTCAGATACGT
CTAGCTTTATGTCAAGTGACAAAGG GCCAACCAGAACAAACAGACAGTTCGGGAGGGAGCAT
GGTGTAGACCAAAGG
Reads with CIGAR Ops

- Large (consistent) percentage of reads have complicated CIGAR strings – which is somewhat reflective of the mapper

<table>
<thead>
<tr>
<th></th>
<th>% with 1 op</th>
<th>% with 1 or 2 ops</th>
<th>% with &gt;= 5 ops</th>
<th>% with &gt;= 10 ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illumina NasEP1255</td>
<td>65%</td>
<td>(only 1,3,5 ops)</td>
<td>3%</td>
<td>0.000001%</td>
</tr>
<tr>
<td>Proton Input P1</td>
<td>31%</td>
<td>45%</td>
<td>30%</td>
<td>4%</td>
</tr>
<tr>
<td>Proton IRF4P</td>
<td>28%</td>
<td>43%</td>
<td>31%</td>
<td>5%</td>
</tr>
<tr>
<td>Torrent GATA3</td>
<td>32%</td>
<td>47%</td>
<td>28%</td>
<td>5%</td>
</tr>
<tr>
<td>Torrent InputCD8</td>
<td>32%</td>
<td>47%</td>
<td>27%</td>
<td>4%</td>
</tr>
</tbody>
</table>
ABI SOLiD

Fragment Library (Targeted Resequencing, ChIP-Seq, SmallRNA, Multiplexing)

Mate-pair Library (Whole Genome Sequencing, Structural Variation)

Note: unlike 454/Illumina paired-ends, mate-pairs sequenced in same direction

Images copyright 2009 ABI
• Sequencing by **ligation** with fluorescently-labeled **di-base** probes compete (2 bases, 3 degenerate N, 3 universal Z)

• Successive cycles means each di-base seen twice.
ABI SOLiD

- Di-base encoding allows accurate and fast detection/disambiguation of sequencing errors/valid polymorphisms

Color space

Base space

- “Invalid” changes are inconsistent with SNP and likely sequencing errors

Images copyright 2009 ABI
Paired End/Mate Pair Reads

- **Illumina**: paired end is both ends of ~300 bp fragment (shorter than a 454 read, shorter than most TEs)
- **454**: paired ends span ~3, 8, 20 Kb
- **SOLiD**: mate pairs span ~1, 2, 3, 5, 10 Kb

Single reads can map to multiple positions

Paired read often map uniquely


Adapted from D. Pollock
Evolution of Sequencing

3rd Generation Sequencing (Single Molecule)

Adapted from D. Pollock
3rd-Generation Sequencing

- **PacBio Single Molecule Real Time (SMRT):** polymerase and single DNA molecule fixed to surface (zero-mode waveguides ZMW). Each of 4 DNA bases labelled with fluorescent dye, when incorporated by polymerase, generates flash of light, records time and color series of flashes.

- **Helicos:** first single molecule sequencer, seq by synthesis + pyroseq from individual ACGT (without PCR to introduce bias), rather than thousands of identical frags sequenced simultaneously to get a signal bright enough to interpret.


From PMID: 20686191
Single Molecule Sequencing

- Currently plagued by high error rates (dye perhaps not incorporated on dNTP, reaction too slow or fast for camera, etc) causes errors and **phasing** problems, as in 2nd generation technologies
  - PacBio approaches problem by creating circular construct with DNA for repeated re-reads of same molecule, Helicos 2-pass sequencing
- PacBio strobe sequencing randomly turns on light to mimic paired ends – useful for de novo seq of scaffolding contigs (discontinued development 2011)
- Because real-time capture of incorporation, can distinguish nucleotide type by lag-time between incorporation (2009 PacBio reports exciting implications for detecting methylated C, or A (plants) …)

Nanopore Technology

- Oxford Nanopore (PMID: 19350039 (2009)), uses alpha-hemolysin nanopores (αHL), can discriminate between the four bases of DNA (and methyl C)

- Lazslo @ UW: using Mycobacterium smegmatis porin A (MspA) not base-pair resolution: roughly a 4-mer, but report that each 4-mer has distinct signal (PMID: 24964173 (2014))

From Oxford Nanopore Technologies movie:
<table>
<thead>
<tr>
<th>Platform</th>
<th>Template Preparation</th>
<th>Chemistry</th>
<th>Max Read length (bases)</th>
<th>Run Times (days)</th>
<th>Max Gb per Run</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roche 454</td>
<td>Clonal-emPCR</td>
<td>Pyrosequencing</td>
<td>400‡</td>
<td>0.42</td>
<td>0.40-0.60</td>
</tr>
<tr>
<td>GS FLX Titanium</td>
<td>Clonal-emPCR</td>
<td>Pyrosequencing</td>
<td>400‡</td>
<td>0.42</td>
<td>0.035</td>
</tr>
<tr>
<td>Illumina MiSeq</td>
<td>Clonal Bridge Amplification</td>
<td>Reversible Dye Terminator</td>
<td>2×300</td>
<td>0.17-2.7</td>
<td>15</td>
</tr>
<tr>
<td>Illumina HiSeq</td>
<td>Clonal Bridge Amplification</td>
<td>Reversible Dye Terminator</td>
<td>2×150</td>
<td>1-11</td>
<td>600</td>
</tr>
<tr>
<td>Illumina Genome Analyzer IIx</td>
<td>Clonal Bridge Amplification</td>
<td>Reversible Dye Terminator (11)[12]</td>
<td>2×150</td>
<td>2-14</td>
<td>95</td>
</tr>
<tr>
<td>Life Technologies SOLiD4</td>
<td>Clonal-emPCR</td>
<td>Oligonucleotide 8-mer Chained Ligation (13)</td>
<td>35-50 SOLiD5500:35-75</td>
<td>4-7</td>
<td>35-50</td>
</tr>
<tr>
<td>Life Technologies Ion Proton</td>
<td>Clonal-emPCR</td>
<td>Native dNTPs, proton detection</td>
<td>200 PGM:300</td>
<td>0.5</td>
<td>100 PGM:1.5</td>
</tr>
<tr>
<td>Complete Genomics</td>
<td>Gridded DNA-nanoballs</td>
<td>Oligonucleotide 9-mer Unchained Ligation (15)[16][17]</td>
<td>7×10</td>
<td>11</td>
<td>3000</td>
</tr>
<tr>
<td>Helicos Biosciences Heliscope</td>
<td>Single Molecule</td>
<td>Reversible Dye Terminator</td>
<td>35‡</td>
<td>8</td>
<td>25</td>
</tr>
<tr>
<td>Pacific Biosciences SMRT</td>
<td>Single Molecule</td>
<td>Phospholinked Fluorescent Nucleotides</td>
<td>10,000 (N50); 30,000+ (max) (18)</td>
<td>0.08</td>
<td>0.5 (19)</td>
</tr>
</tbody>
</table>

From Wikipedia 2014

Center for Genes, Environment, and Health
Comparison and Usage

• **Short Reads (ABI/Illumina/Ion)**
  - Resequencing b/c high throughput runs (coverage)
  - SNP detection (coverage)
  - Micro RNA (23 bp)
  - Counting (e.g., transcriptome profiling, ChIPSeq)
    - Adjustable dynamic range ($$)
  - Hard to place near repetitive elements
  - Harder to assemble *de novo*
  - 35-400 bp reads intermediate

• **Long Reads (454/Ion/single mol)**
  - Many fewer reads (lower coverage)
  - Much longer (implications for assembly, indel detection)
  - Better for de novo sequencing (can bridge scaffolds)
  - Better in ability to span repetitive regions (find unique seq)
  - Amplicon and tagging
    - longer reads detect varying copy numbers btwn individuals
  - Meta-genomics
    - 16S RNA, need 500bp reads to see small variant region among species

Adapted from D. Pollock
PR Space versus Science Space

- **Alignment problems from length** - 97% E coli uniquely aligned with 18bp reads, only 90% of human with 30bp reads – see Voelkerding, 2009

- **Flow and phasing** - skip basecall in cycle

- **Data quality and Error rate**
  - Variation along sequence (higher error at 3’ end as reagents begin to fail, error for next base in elongation depends on identity of previous base, homopolymers)
  - Quality scores (Equivalence? – can use GATK for calibration)

- **Length distribution versus average**
  - mate pair/paired end insert size

- **Coverage versus GC content**

- **Raw versus recovered sequence** - Uncallable bases waste reads - How much coverage with different methods? What about multi-mapped reads?

- **Tagging (barcodes) and multiplexing** - Variation in coverage for each individual in multiplex

Adapted from D. Pollock
Moore’s law ~2x/2yrs

1. The accelerating pace of change...
   - Agricultural Revolution: 8,000 years
   - Industrial Revolution: 120 years
   - Lightbulb: 90 years
   - Moon landing: 22 years
   - World Wide Web: 9 years
   - Human genome sequenced

2. ...and exponential growth in computing power...
   - Computer technology, shown here climbing dramatically by powers of 10, is now progressing more each hour than it did in its entire first 90 years

3. ...will lead to the Singularity
   - Apple II
   - UNIVAC I
   - Colossus
   - Analytical engine
   - Power Mac G4

Computer Rankings
- By calculations per second per $1,000

http://humanswlord.files.wordpress.com/2014/01/moores-law-graph-gif.png
The Sequencing Explosion

- Cost of computing (Moore's Law)
- Cost of Sequencing

- $3 billion
- $1,000

Human Genomes Sequenced

Log Scale

- 100,000
- 10,000
- 1,000
- 100
- 10
- 1
- 0.1

Adapted from The Economist
Ewan Birney’s slide, ECCB2014
Additional (Overlooked) Costs of Sequencing

• **Storage**
  – dbGaP (NCBI), EGA (Euro), DDBJ (Japan), CGHub (NCI), BGI (China)… etc: esp TCGA >2PB dataset, CGHub delivers 10+PB data
  – Which level: img/dat, spch/dat.gz, fastq.gz, bam? Encode Qvals (ACGT:2bits, 0-40 qual: 5.3bits)?
  – Sequence Graph Reference Genome (Haussler arxiv 1404), DNA Digital Storage? (Goldman&Birney, et al: PMID: 23354052)

• **Management**
  – Data Validation, Tracking, Reporting, Distribution

• **Access**
  – Example: iPlant Collaboration (NSF) cyber infrastructure: data storage facility; interactive, web-based, analytical platform; cloud infrastructure; web authentication and security services; support for scaling computational algorithms to run on large, high-speed computers; education and training; people w/ expertise in all of above)

• **Accessibility**
  – Cloud computing solution for big projects, algorithms for identifying ‘slices’ of data

• **Security**
  – Identifiable by DNA (92 SNPs PMID: 19937056), ethics vs practical considerations
  – Non US researchers hesitant to collaborate after Snowden incident

http://www.politigenomics.com/2009/02/the-scale-up.html
(Goldman & Birney, et al: PMID: 23354052)

Introduction

• Sequencing technologies
  – 1st Gen: Gel based sequencing and Capillary electrophoresis
  – 2nd Gen: Next-Generation Sequencers:
    • Sequencing by Synthesis: Pyrosequencing with ePCR (Roche 454), Reversible Dye-termination with cluster/bridge amplification (Illumina Solexa), Semiconductor (Ion Torrent PGM/Proton)
    • Sequencing by Ligation: ePCR and diColor system (ABI SOLiD)
  – 3rd Generation Sequencers:
    • Single molecule sequencing (ABI SMS, PacBio SMRT, Helicos), nanopore sequencing, …

• Sequencing applications
  – ChIP Seq – transcription factor binding and chromatin modification
  – RNA Seq – transcript detection (boundaries, novel exons), quantification
  – Whole Genome/Exome Sequencing - sequence variant detection

• De novo assembly versus mapping to reference sequence
  – Human Genome Project (Hierarchical versus Shotgun Sequencing)
    • Contig assembly and ordering
  – Mapping/Alignment Programs
    • Repetitive regions (interspersed repeats (SINEs, LiNEs, LTR elements, and DNA transposons), satellite sequences, and low-complexity sequences)
    • Polymorphisms and structural variants (insertions, deletions, substitutions, inversions, translocation)
Introduction

• **Sequencing technologies**
  – 1\textsuperscript{st} Gen: Gel based sequencing and Capillary electrophoresis
  – 2\textsuperscript{nd} Gen: Next-Generation Sequencers:
    • Sequencing by Synthesis: Pyrosequencing with ePCR (Roche 454), Reversible Dye-termination with cluster/bridge amplification (Illumina Solexa), Semiconductor (Ion Torrent PGM/Proton)
    • Sequencing by Ligation: ePCR and diColor system (ABI SOLiD)
  – 3\textsuperscript{rd} Generation Sequencers:
    • Single molecule sequencing (ABI SMS, PacBio SMRT, Helicos), nanopore sequencing, …

• **Sequencing applications**
  – ChIP Seq – transcription factor binding and chromatin modification
  – RNA Seq – transcript detection (boundaries, novel exons), quantification
  – Whole Genome/Exome Sequencing - sequence variant detection

• **De novo assembly versus mapping to reference sequence**
  – Human Genome Project (Hierarchical versus Shotgun Sequencing)
    • Contig assembly and ordering
  – Mapping/Alignment Programs
    • Repetitive regions (interspersed repeats (SINEs, LINES, LTR elements, and DNA transposons), satellite sequences, and low-complexity sequences)
    • Polymorphisms and structural variants (insertions, deletions, substitutions, inversions, translocation)
Human Genome Project

15 FEBRUARY 2001

Today: UCD core $4000 on Illumina, 30x coverage, Complete Genomics $3000, Proton brags $1000

16 FEBRUARY 2001
Other Sequencing Projects

- 1000 Genomes Project
- MalariaGEN – sequencing of 1000s of malaria isolates
- Cancer Genomes: ICGC, TCGA
- Mouse Genomes Project – 17 common lab strains
- 1001 Genomes Project – Arabidopsis wgs variation
- UK10K – 10k healthy and disease affected Individuals
- COPDgene 500-600 patients (Pfizer)
- ...

Center for Genes, Environment, and Health
Exome Sequencing

- Randomly shear whole DNA
- Select fragments using pre-designed capture array
  - Typically 120bp, tiled across exon
- Amplify on array
- Sequence fragments
RNA-Seq

- cDNA made from transcript, sequenced, mapped to genome
  - Seq more sensitive than micro-arrays since do not have to design array probes or decide chip content, especially important for microRNA (~22nt total)
  - Seq can be used to distinguish isoforms, allelic expression, sequence variants
  - Seq more expensive than microarrays and suffer from 5’/3’ biases, high abundance transcripts (housekeeping, ribosomal) which make up the majority of sequencing data (e.g. in some tissues 5% of the genes represent up to 75% of the reads sequenced).

- Mapping to reference genome means recognizing known/novel splice junctions, determining exon boundaries, potentially discovering novel exons. Also map to transcriptome seqs.

- Quantification done at exon or transcript level (what about when exons shared?)

- Differential RNA expression between samples measures diff in coverage, isoform use or splice events (exon skip/intron retain)
ChIP-Seq
(Chromatin Immunoprecipitation + Sequencing)

1. Cell Nucleus
2. Crosslink Protein and Shear DNA
3. Add Protein-Specific Antibody
4. Immunoprecipitate and purify complexes
5. Reverse Crosslinks, Purify DNA and prepare for sequencing
6. Sequence DNA fragment and map to genome

- Requires read alignment and peak finding

Image from wikipedia

Data from Dr Barbara Wold
Sequence Variant Detection

Sequence Variant Detection

- **Single Nucleotide Polymorphisms**
  - Compute genotype from distribution of 4 nucleotides for reads mapping to same reference location
    - Libraries can contain PCR duplicate reads, inflates coverage, use unique (for WGS, not RNA-seq)
  - Use coverage, read position to determine accuracy
    - SOLiD accurate due to dibase encoding and rules for valid adjacent
  - Homozygous SNPs more accurate than heterozygous SNPs
    - Typically cut-off for minimum number of reads calling each variant of het

Likely sequencing error
Non-called bases or indels? Phasing?

Voelkerding et al, Clin Chem 2009
Sequence Variant Detection

- Small Insertion-Deletions can be detected with gapped-alignment of reads
- Large Insertion-Deletions suggested by deviations from expected insert size for paired reads
Sequence Variant Detection

- Transversions suggested by correct spacing but unexpected orientation of paired ends, also suggested by drop in coverage for normal pairs
- Translocations suggested by unexpected, mapping of paired ends to different chromosomes
- In all examples, note importance of good quality reference genome, long length of insert for paired reads (mix of insert size also beneficial)
Sequence Variant Detection

• Copy Number Variation

• Can use difference from expected coverage to indicate CNVs (e.g. see CNV-seq by Xie and Tammi, Bioinformatics, 2009)

• SOLiD algorithm: compute log-ratio of coverage to expected coverage (GC corrected), use HMM to estimate copy number, take Viterbi sequence

X-Axis: Window IDs (windows of size of 5000 mappable bases)
Old School: the basics of comparing sequences
Alignment of two sequences

- What is the best way to align the sequences $S = AB$ and $T = XYZ$?

- But ‘best’ implies can score each alignment
  - provide score per alignment (not practical as space of sequences grows)
  - instead sum scores $\sigma(a,b)$ of matching each character pair (ex. BLOSUM62)
  - could penalize for extended gaps, ex) low score for $A----B$ vs $-XYZ-$

$$\sigma(a,b) = \text{score for matching character } a \text{ in } S \text{ with } b \text{ in } T$$

ex) $a,b \in \{A,B,C,X,Y,-\}$

score: match any non-space
discourage gaps in $S$

$\sigma(-,*) = -2$, $\sigma(*,-) = -1$, otherwise $\sigma(*,*) = 1$

- $- A B$
  - $A B C$
  - $- A - B$
  - $- A - B$
  - $A - - B$
  - $A - - B$
  - $- A - - B$
  - $- A - - B$

Note how score can’t distinguish between last 3 alignments
(need affine gap penalties, as in Miller’s lecture)
Alignment of two sequences

• In example above, still had to calculate total score for each alignment and number of computations grows exponentially as length of sequence increases

• What if you could take a shortcut?
  – Imagine that you magically had the optimal score for most of S and T and just needed to decide what to do with the last character of each sequence (i.e. $S_n$ and $T_m$)?
  • Then only need to consider 3 possibilities!!

- But how did we get the optimal alignment for say $S'$=$S_1$..$S_{n-1}$ and $T'$=$T_1$...$T_{m-1}$?
- Imagine that you magically had the optimal score for most of $S'$ and $T'$, and just needed to decide what to do with the last character of each $S'$ and $T'$ (i.e. $S_{n-1}$, $T_{m-1}$)?
  …. but wait, we just answered that question for $S_n$ and $T_m$!!
- Can repeat question until down to $S_1$ and $T_1$, then propagate to answer for $S_n$ and $T_m$
Dynamic Programming

• Quote Miller’s ‘Sequence Search & Alignment’ lecture:
  – Break a problem into subproblems and solve them.
  – Combine solutions of the subproblems to construct
    the solution of the original problem.

\[
\begin{align*}
S &= \text{optimally aligned up to } i-1 \\
T &= \text{optimally aligned up to } j-1 \\
\text{Match } S_i \text{ and } T_j \\
\text{Score: best score for aligning} \\
(S_1 \ldots S_{i-1}, T_1 \ldots T_{j-1}) &+ \sigma(S_i, T_j)
\end{align*}
\]

\[
\begin{align*}
S &= \text{optimally aligned up to } i-1 \\
T &= \text{optimally aligned up to } j-1 \\
\text{Delete } S_i \\
\text{Score: best score for aligning} \\
(S_1 \ldots S_{i-1}, T_1 \ldots T_{j-1}) &+ \sigma(S_i, -)
\end{align*}
\]

\[
\begin{align*}
S &= \text{optimally aligned up to } i \\
T &= \text{optimally aligned up to } j-1 \\
\text{Insert ‘-’ into } S \\
\text{Score: best score for aligning} \\
(S_1 \ldots S_i, T_1 \ldots T_{j-1}) &+ \sigma(-, T_j)
\end{align*}
\]

\[
\text{BestAlign}(i,j) = \max \left\{ \begin{array}{l}
\text{BestAlign}(i-1,j-1) + \sigma(S_i, T_j) \\
\text{BestAlign}(i-1,j) + \sigma(S_i, -) \\
\text{BestAlign}(i,j-1) + \sigma(-, T_j) \\
\end{array} \right. 
\]

Combination

Problem expressed in terms of i and j

Subproblems expressed in terms of i-1 and j-1

Center for Genes, Environment, and Health
Needleman-Wunsch: Dynamic Programming

Global alignment: match all characters in S with characters in T, allowing for spaces in either string to make length equal

\[ V(i,j) \] : optimal score of \( S_1 \ldots S_i \) and \( T_1 \ldots T_j \) given score \( \sigma(a,b) \) for matching characters \( a \) in \( S \) and \( b \) in \( T \)

Base conditions:

\[ V(i,0) = \sum_{k=0}^{i} \sigma(S_k,-) \]

(essentially matches all characters in \( S \) up to \( i \) with ‘-‘, leaving \( T \) alone)

\[ V(0,j) = \sum_{k=0}^{j} \sigma(-,T_k) \]

(essentially matches all characters in \( T \) up to \( j \) with ‘-‘, leaving \( S \) alone)

Recurrence:

for \( 1 \leq i \leq n, 1 \leq j \leq m \)

\[ V(i,j) = \max \left\{ \begin{array}{ll}
V(i-1,j-1) + \sigma(S_i,T_j) & \text{(Match } S_i,T_j) \\
V(i-1,j) + \sigma(S_i,-) & \text{(Delete } S_i) \\
V(i,j-1) + \sigma(-,T_j) & \text{(Insert ‘-‘ in } S) 
\end{array} \right\} \]

Need for when \( i =1 \) or \( j =1 \) (i.e. first character of \( S \) or \( T \) need answer for \( V(i,j) \) in terms of \( V(i-1,*) \) and \( V(*,j-1) \))

\[ V(i,j) \text{ is same as BestAlign on previous slide} \]
**Needleman-Wunsch: Dynamic Programming**

- Create table $V(i,j)$ where each cell shows ‘optimal alignment for $S_i..S_i$ against $T_j..T_j$’, denoted by \{…\}

<table>
<thead>
<tr>
<th></th>
<th>Tj</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Si</td>
<td></td>
<td>0</td>
<td>c</td>
<td>a</td>
<td>d</td>
<td>b</td>
<td>d</td>
</tr>
<tr>
<td>0</td>
<td>T:</td>
<td>-</td>
<td>c</td>
<td>ca</td>
<td>cad</td>
<td>cadb</td>
<td>cadbd</td>
</tr>
<tr>
<td></td>
<td>S:</td>
<td>-</td>
<td></td>
<td>--</td>
<td>--</td>
<td>----</td>
<td>-----</td>
</tr>
<tr>
<td>1</td>
<td>a</td>
<td>{c}</td>
<td>{a}</td>
<td>{ca}</td>
<td>{cad}</td>
<td>{cadb}</td>
<td>{cadbd}</td>
</tr>
<tr>
<td></td>
<td>a</td>
<td>{a}</td>
<td></td>
<td>{a}</td>
<td>{a}</td>
<td>{a}</td>
<td>{a}</td>
</tr>
<tr>
<td>2</td>
<td>c</td>
<td>{c}</td>
<td>{ac}</td>
<td>{ca}</td>
<td>{cad}</td>
<td>{cadb}</td>
<td>{cadbd}</td>
</tr>
<tr>
<td></td>
<td>ac</td>
<td>{ac}</td>
<td></td>
<td>{ac}</td>
<td>{ac}</td>
<td>{ac}</td>
<td>{ac}</td>
</tr>
<tr>
<td>3</td>
<td>b</td>
<td>{c}</td>
<td>{acb}</td>
<td>{ca}</td>
<td>{cad}</td>
<td>{cadb}</td>
<td>{cadbd}</td>
</tr>
<tr>
<td></td>
<td>acb</td>
<td>{acb}</td>
<td></td>
<td>{acb}</td>
<td>{acb}</td>
<td>{acb}</td>
<td>{acb}</td>
</tr>
<tr>
<td>4</td>
<td>c</td>
<td>{c}</td>
<td>{acbc}</td>
<td>{ca}</td>
<td>{cad}</td>
<td>{cadb}</td>
<td>{cadbd}</td>
</tr>
<tr>
<td></td>
<td>acbc</td>
<td>{acbc}</td>
<td></td>
<td>{acbc}</td>
<td>{acbc}</td>
<td>{acbc}</td>
<td>{acbc}</td>
</tr>
<tr>
<td>5</td>
<td>d</td>
<td>{c}</td>
<td>{acbcd}</td>
<td>{ca}</td>
<td>{cad}</td>
<td>{cadb}</td>
<td>{cadbd}</td>
</tr>
<tr>
<td></td>
<td>acbcd</td>
<td>{acbcd}</td>
<td></td>
<td>{acbcd}</td>
<td>{acbcd}</td>
<td>{acbcd}</td>
<td>{acbcd}</td>
</tr>
<tr>
<td>6</td>
<td>b</td>
<td>{c}</td>
<td>{abcbd}</td>
<td>{ca}</td>
<td>{cad}</td>
<td>{cadb}</td>
<td>{cadbd}</td>
</tr>
<tr>
<td></td>
<td>abcdb</td>
<td>{abcdb}</td>
<td></td>
<td>{abcdb}</td>
<td>{abcdb}</td>
<td>{abcdb}</td>
<td>{abcdb}</td>
</tr>
</tbody>
</table>
Needleman-Wunsch: Dynamic Programming

Global alignment: match all characters in S with characters in T, allowing for spaces in either string to make length equal

\[ V(i,j) : \text{optimal score of } S_1...S_i \text{ and } T_1..T_j \text{ given score } \sigma(a,b) \text{ for matching characters } a \text{ in } S \text{ and } b \text{ in } T \]

Base conditions:

\[ V(i,0) = \sum_{k=0}^{i} \sigma(S_k,-) \]
\[ V(0,j) = \sum_{k=0}^{j} \sigma(-,T_k) \]

Recurrence:

for \( 1 \leq i \leq n, 1 \leq j \leq m \)

\[ V(i,j) = \max \begin{cases} V(i-1,j-1) + \sigma(S_i,T_j) \\ V(i-1,j) + \sigma(S_i,-) \\ V(i,j-1) + \sigma(-,T_j) \end{cases} \]

\( \sigma = -1 \text{ mismatch, } +2 \text{ match} \)

<table>
<thead>
<tr>
<th>Si</th>
<th>Tj</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
</tr>
<tr>
<td>1 a</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 c</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 b</td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 c</td>
<td>-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 d</td>
<td>-5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 b</td>
<td>-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Initialize with Base Conditions
Needleman-Wunsch: Dynamic Programming

Global alignment: match all characters in S with characters in T, allowing for spaces in either string to make length equal

\[ V(i,j) : \text{optimal score of } S_1 \ldots S_i \text{ and } T_1 \ldots T_j \text{ given score } \sigma(a,b) \text{ for matching characters } a \text{ in } S \text{ and } b \text{ in } T \]

Base conditions:

\[ V(i,0) = \sum_{k=0}^{i} \sigma(S_k, -) \]

\[ V(0,j) = \sum_{k=0}^{j} \sigma(-, T_k) \]

Recurrence:

for \(1 \leq i \leq n, 1 \leq j \leq m\)

\[ V(i,j) = \max \left\{ V(i-1,j-1) + \sigma(S_i, T_j), V(i-1,j) + \sigma(S_i, -), V(i,j-1) + \sigma(-, T_j) \right\} \]

\[ \sigma = -1 \text{ mismatch, } +2 \text{ match} \]

<table>
<thead>
<tr>
<th>(V(i,j))</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(V(0,0))</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
</tr>
<tr>
<td>(V(1,0))</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
<td>-6</td>
</tr>
<tr>
<td>(V(2,1))</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
<td>-6</td>
<td>-7</td>
</tr>
<tr>
<td>(V(3,2))</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
<td>-6</td>
<td>-7</td>
<td>-8</td>
</tr>
<tr>
<td>(V(4,3))</td>
<td>-4</td>
<td>-5</td>
<td>-6</td>
<td>-7</td>
<td>-8</td>
<td>-9</td>
</tr>
<tr>
<td>(V(5,4))</td>
<td>-5</td>
<td>-6</td>
<td>-7</td>
<td>-8</td>
<td>-9</td>
<td>-10</td>
</tr>
</tbody>
</table>

Note that this cell shows score for optimal alignment of \(i=0\) characters from \(S\) against all \(j=m\) characters from \(T\) i.e.,

\[ S: \quad - \quad - \quad - \quad - \quad s: a \quad c \quad b \quad c \quad d \quad b \]

\[ T: c \quad a \quad d \quad b \quad d \]
Needleman-Wunsch: Dynamic Programming

Global alignment: match all characters in S with characters in T, allowing for spaces in either string to make length equal

\( V(i,j) : \) optimal score of \( S_1...S_i \) and \( T_1...T_j \) given score \( \sigma(a,b) \) for matching characters \( a \) in \( S \) and \( b \) in \( T \)

**Base conditions:**

\[
V(i,0) = \sum_{k=0}^{i} \sigma(S_k, -)
\]

\[
V(0,j) = \sum_{k=0}^{j} \sigma(-, T_k)
\]

**Recurrence:**

for \( 1 \leq i \leq n, 1 \leq j \leq m \)

\[
V(i,j) = \max \begin{cases} 
V(i-1,j-1) + \sigma(S_i, T_j) \\
V(i-1,j) + \sigma(S_i, -) \\
V(i,j-1) + \sigma(-, T_j)
\end{cases}
\]

\[\sigma = -1 \text{ mismatch, } +2 \text{ match}\]

<table>
<thead>
<tr>
<th>Si</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
</tr>
<tr>
<td>1</td>
<td>a</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>c</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>b</td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>c</td>
<td>-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>d</td>
<td>-5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>b</td>
<td>-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Likewise, this cell shows score for optimal alignment of all \( i=n \) characters from \( S \) against \( j=0 \) characters from \( T \) i.e.,

\( S: \ a \ c \ b \ c \ d \ b \)

\( T: \ - \ - \ - \ - \ - \ - \ c \ a \ d \ b \ d \)
Needleman-Wunsch: Dynamic Programming

Global alignment: match all characters in S with characters in T, allowing for spaces in either string to make length equal

V(i,j) : optimal score of S₁…Sᵢ and T₁…Tⱼ given score σ(a,b) for matching characters a in S and b in T

Base conditions:
V(i,0) = ∑ₖ₌₀ᵢ σ(Sₖ,-)
V(0,j) = ∑ₖ₌₀ⱼ σ(-,Tₖ)

Recurrence:
for 1≤i≤n, 1≤j≤m

V(i,j) = max

  V(i-1,j-1) + σ(Sᵢ,Tⱼ) (match (M)) -4
  V(i-1,j) + σ(Sᵢ,-) (delete (D)) -5
  V(i,j-1) + σ(-,Tⱼ) (insert - (I)) -6

σ = -1 mismatch, +2 match

<table>
<thead>
<tr>
<th>Si</th>
<th>Tj</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
<td>-6</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-4</td>
<td>-5</td>
<td>-6</td>
<td>-7</td>
<td>-8</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-5</td>
<td>-6</td>
<td>-7</td>
<td>-8</td>
<td>-9</td>
<td></td>
</tr>
</tbody>
</table>

Now do cell V(i=1,j=1) which only depends on cells V(0,0), V(0,1) and V(1,0)
### Needleman-Wunsch: Dynamic Programming

\[ \sigma = -1 \text{ mismatch, } +2 \text{ match} \]

<table>
<thead>
<tr>
<th>( T_j )</th>
<th>( 0 )</th>
<th>( 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_i )</td>
<td>( - )</td>
<td>( -1 )</td>
</tr>
<tr>
<td>( 0 )</td>
<td>0</td>
<td>( -1 )</td>
</tr>
<tr>
<td>( 1 )</td>
<td>( -1 )</td>
<td>( -1 )</td>
</tr>
<tr>
<td>( a )</td>
<td>( -1 )</td>
<td>( -1 )</td>
</tr>
</tbody>
</table>

**Equation:**

- M: \( V(0,0) + \sigma(a,c) = 0 - 1 = -1 \)
- D: \( V(0,1) + \sigma(a,-) = -1 + 1 = 0 \)
- I: \( V(1,0) + \sigma(-,c) = -1 + 1 = 0 \)

\[ V(1,1) = \max \{ -1, 0, 0 \} = -1 \]

Record M as operation here to allow backtracking later.
Needleman-Wunsch: Dynamic Programming

Global alignment: match all characters in $S$ with characters in $T$, allowing for spaces in either string to make length equal

$V(i,j)$: optimal score of $S_1...S_i$ and $T_1..T_j$ given score $\sigma(a,b)$ for matching characters $a$ in $S$ and $b$ in $T$

**Base conditions:**

$V(i,0) = \sum_{k=0}^{i} \sigma(S_k,-)$

$V(0,j) = \sum_{k=0}^{j} \sigma(-,T_k)$

**Recurrence:**

for $1 \leq i \leq n$, $1 \leq j \leq m$

$V(i,j) = \max \begin{cases} V(i-1,j-1) + \sigma(S_i,T_j) \text{ (match (M))} \\ V(i-1,j) + \sigma(S_i,-) \text{ (delete (D))} \\ V(i,j-1) + \sigma(-,T_j) \text{ (insert - (I))} \end{cases}$

$\sigma = -1$ mismatch, +2 match

<table>
<thead>
<tr>
<th>Si</th>
<th>Tj</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
</tr>
<tr>
<td>1 a</td>
<td>-1</td>
<td>-1</td>
<td>M-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 c</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 b</td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 d</td>
<td>-4</td>
<td>(match (M))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 d</td>
<td>-5</td>
<td>(delete (D))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 b</td>
<td>-6</td>
<td>(insert - (I))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Now do cell $V(i=1,j=2)$ which only depends on cells $V(0,1)$, $V(0,2)$ and $V(1,1)$ and so on
### Needleman-Wunsch: Dynamic Programming

\( \sigma = -1 \) mismatch, +2 match

<table>
<thead>
<tr>
<th>( \text{Tj} )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
</tr>
<tr>
<td>1 a</td>
<td>-1</td>
<td><strong>M:</strong> ( V(0,0) + \sigma(a,c) = 0 + 1 = -1 )</td>
<td><strong>D:</strong> ( V(0,1) + \sigma(a,-) = -1 + 1 = -2 )</td>
<td><strong>I:</strong> ( V(1,0) + \sigma(-,c) = 1 + 1 = -2 )</td>
<td><strong>M:</strong> ( V(0,1) + \sigma(a,a) = -1 + 2 = 1 )</td>
<td><strong>D:</strong> ( V(0,2) + \sigma(a,-) = -2 + 1 = -3 )</td>
</tr>
<tr>
<td>2 c</td>
<td>-2</td>
<td><strong>M:</strong> ( V(1,0) + \sigma(c,c) = -1 + 2 = 1 )</td>
<td><strong>D:</strong> ( V(1,1) + \sigma(c,-) = -1 + 1 = -2 )</td>
<td><strong>I:</strong> ( V(2,0) + \sigma(-,c) = 2 + 1 = -3 )</td>
<td><strong>M:</strong> ( V(1,1) + \sigma(c,a) = -1 + 1 = -2 )</td>
<td><strong>D:</strong> ( V(1,2) + \sigma(c,-) = 1 + 1 = 0 )</td>
</tr>
<tr>
<td>3 b</td>
<td>-3</td>
<td><strong>M:</strong> ( V(2,0) + \sigma(b,c) = -2 + 1 = -3 )</td>
<td><strong>D:</strong> ( V(2,1) + \sigma(b,-) = 1 + 1 = 0 )</td>
<td><strong>I:</strong> ( V(3,0) + \sigma(-,c) = -3 + 1 = -4 )</td>
<td><strong>M:</strong> ( V(2,1) + \sigma(b,a) = 1 + 1 = 0 )</td>
<td><strong>D:</strong> ( V(2,2) + \sigma(b,-) = 0 + 1 = -1 )</td>
</tr>
<tr>
<td>4 c</td>
<td>-4</td>
<td><strong>M:</strong> ( V(3,0) + \sigma(c,a) = -3 + 2 = 1 )</td>
<td><strong>D:</strong> ( V(3,1) + \sigma(c,-) = 0 + 1 = -1 )</td>
<td><strong>I:</strong> ( V(4,0) + \sigma(-,c) = -4 + 1 = -5 )</td>
<td><strong>M:</strong> ( V(3,1) + \sigma(c,a) = 0 + 1 = -1 )</td>
<td><strong>D:</strong> ( V(3,2) + \sigma(c,-) = 0 + 1 = -1 )</td>
</tr>
<tr>
<td>5 d</td>
<td>-5</td>
<td><strong>M:</strong> ( V(4,0) + \sigma(d,c) = -4 + 1 = -5 )</td>
<td><strong>D:</strong> ( V(4,1) + \sigma(d,-) = 1 + 1 = 2 )</td>
<td><strong>I:</strong> ( V(5,0) + \sigma(-,c) = 5 + 1 = 6 )</td>
<td><strong>M:</strong> ( V(4,1) + \sigma(d,a) = 1 + 1 = 2 )</td>
<td><strong>D:</strong> ( V(4,2) + \sigma(d,-) = 1 + 1 = 2 )</td>
</tr>
</tbody>
</table>
| 6 b | -6 | **M:** \( V(5,0) + \sigma(b,c) = -5 + 1 = -6 \) | **D:** \( V(5,1) + \sigma(b,-) = 2 + 1 = 3 \) | **I:** \( V(6,0) + \sigma(-,c) = -6 + 1 = -7 \) | **M:** \( V(5,1) + \sigma(b,a) = -2 + 1 = -3 \) | **D:** \( V(5,2) + \sigma(b,-) = 2 + 1 = 3 \) | **I:** \( V(6,1) + \sigma(-,a) = -3 + 1 = -4 \) | **M:** \( V(5,2) + \sigma(b,d) = -2 + 1 = -3 \) | **D:** \( V(5,3) + \sigma(b,-) = 1 + 1 = 0 \) | **I:** \( V(6,2) + \sigma(-,d) = 3 + 1 = -4 \) | **M:** \( V(5,3) + \sigma(b,b) = 1 + 2 = 3 \) | **D:** \( V(5,4) + \sigma(b,-) = 0 + 1 = 1 \) | **I:** \( V(6,3) + \sigma(-,b) = 0 + 1 = 1 \) | **M:** \( V(5,4) + \sigma(b,d) = 0 + 1 = 1 \) | **D:** \( V(5,5) + \sigma(b,-) = 3 + 1 = 2 \) | **I:** \( V(6,4) + \sigma(-,d) = 3 + 1 = 2 \)
Needleman-Wunsch: Dynamic Programming

Global alignment: match all characters in $S$ with characters in $T$, allowing for spaces in either string to make length equal

$V(i,j)$: optimal score of $S_1...S_i$ and $T_1..T_j$ given score $\sigma(a,b)$ for matching characters $a$ in $S$ and $b$ in $T$

Base conditions:

- $V(i,0) = \sum_{k=0}^{i} \sigma(S_k,-)$
- $V(0,j) = \sum_{k=0}^{j} \sigma(-,T_k)$

Recurrence:

- for $1 \leq i \leq n$, $1 \leq j \leq m$

$$V(i,j) = \max \begin{cases} V(i-1,j-1) + \sigma(S_i,T_j) \\ V(i-1,j) + \sigma(S_i,-) \\ V(i,j-1) + \sigma(-,T_j) \end{cases}$$

$\sigma = -1$ mismatch, $+2$ match

<table>
<thead>
<tr>
<th></th>
<th>Tj</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Si</td>
<td></td>
<td>0</td>
<td>c</td>
<td>a</td>
<td>d</td>
<td>b</td>
<td>d</td>
</tr>
<tr>
<td>0..</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
<td></td>
</tr>
<tr>
<td>1 a</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td></td>
</tr>
<tr>
<td>2 c</td>
<td>-2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td></td>
</tr>
<tr>
<td>3 b</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4 c</td>
<td>-4</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5 d</td>
<td>-5</td>
<td>-2</td>
<td>-2</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>6 b</td>
<td>-6</td>
<td>-3</td>
<td>-3</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
Needleman-Wunsch: Dynamic Programming

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Si</td>
<td></td>
<td>c</td>
<td>a</td>
<td>d</td>
<td>b</td>
<td>d</td>
</tr>
<tr>
<td>0</td>
<td>-</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
</tr>
<tr>
<td>1</td>
<td>a</td>
<td>-1</td>
<td>M: 1</td>
<td>M: 1</td>
<td>I: 0</td>
<td>I: -1</td>
</tr>
<tr>
<td>2</td>
<td>c</td>
<td>-2</td>
<td>M: 1</td>
<td>D: 0</td>
<td>I: 0</td>
<td>M: -1</td>
</tr>
<tr>
<td>3</td>
<td>b</td>
<td>-3</td>
<td>D: 0</td>
<td>M: 0</td>
<td>M: -1</td>
<td>D: -1</td>
</tr>
<tr>
<td>4</td>
<td>c</td>
<td>-4</td>
<td>M: -1</td>
<td>M: -1</td>
<td>M: -1</td>
<td>D: 1</td>
</tr>
<tr>
<td>5</td>
<td>d</td>
<td>-5</td>
<td>D: -2</td>
<td>M: -2</td>
<td>M: 1</td>
<td>D: 0</td>
</tr>
<tr>
<td>6</td>
<td>b</td>
<td>-6</td>
<td>D: -3</td>
<td>M: -3</td>
<td>D: 0</td>
<td>M: 3</td>
</tr>
</tbody>
</table>

- Now aligning all S to all T means **backtrack** from cell V(6,5)
  - Multiple ops in cell mean multiple possible paths
- Ex) (shown in red) S: -acbcdb
  - T: cadb-d-
- This matrix path representation also in Miller’s lecture

Center for Genes, Environment, and Health
Smith-Waterman: Dynamic Programming

Local Alignment: find substring a in S and b in T whose optimal global alignment is maximum over all pairs of substrings

\[ V(i,j) : \text{optimal score of } S_1...S_i \text{ and } T_1..T_j \text{ given score } \sigma(a,b) \text{ for matching characters a in S and b in T} \]

**Global Alignment**

Base conditions:
\[ V(i,0) = \sum_{k=0}^{i} \sigma(S_k,-) \]
\[ V(0,j) = \sum_{k=0}^{j} \sigma(-,T_k) \]

Recurrence:
\[ V(i,j) = \max \begin{cases} V(i-1,j-1) + \sigma(S_i,T_j) \\ V(i-1,j) + \sigma(S_i,-) \\ V(i,j-1) + \sigma(-,T_j) \end{cases} \]

**Local Alignment**

Base conditions:
\[ V(i,0) = 0 \quad V(0,j) = 0 \]
\[ \sigma(x,y) = \begin{cases} \geq 0 & \text{if } x=y \\ \leq 0 & \text{if } x\neq y \text{ or one is -} \end{cases} \]

Recurrence:
\[ V(i,j) = \max \begin{cases} 0 \\ V(i-1,j-1) + \sigma(S_i,T_j) \\ V(i-1,j) + \sigma(S_i,-) \\ V(i,j-1) + \sigma(-,T_j) \end{cases} \]
\[ V(i^*,j^*) = \max V(i,j) \]
Computational Complexity

• Global/Local alignment:
  – Time complexity $O(mn)$
  – Space complexity $O(m)$ with some tricks

• If human genome is $3 \times 10^9$ bp and we have 600 million sequences of length 150, how long will this take?
  – Assuming 2.2 GHz computer ($2.2 \times 10^9$ ops per second)
    • $3 \times 10^9 \times 600 \times 10^6 / 2.2 \times 10^9 / 60 / 60 / 24 = 9469.69$ days
  – Obviously, we need a faster way.....
One Sequence versus Genome

- **Basic Local Alignment Search Tool (BLAST)**
  - Compare query to database of library sequences
  - Heuristic, faster than calculating optimal (i.e. Smith-Waterman)
  - Input: FASTA sequence & weight matrix, like BLOSUM62
  - Algorithm: find initial words (seeds) in query and library (hit sequences), expand high scoring seeds with ungapped alignment

- **BLAST-Like Alignment Tool (BLAT)**
  - Faster than BLAST, precomputes index of non-overlapping k-mers in genome
How to use BLAT (the easy way)
### Mouse BLAT Results

#### BLAT Search Results

<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>QUERY</th>
<th>SCORE</th>
<th>START</th>
<th>END</th>
<th>QSIZE</th>
<th>IDENTITY</th>
<th>CHRO</th>
<th>STRAND</th>
<th>START</th>
<th>END</th>
<th>SPAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>135</td>
<td>1</td>
<td>180</td>
<td>180</td>
<td>88.7%</td>
<td>7</td>
<td>+</td>
<td>148029624</td>
<td>148029786</td>
<td>165</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>36</td>
<td>8</td>
<td>131</td>
<td>180</td>
<td>71.5%</td>
<td>19</td>
<td>+</td>
<td>46572952</td>
<td>46573057</td>
<td>106</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>30</td>
<td>111</td>
<td>156</td>
<td>180</td>
<td>88.6%</td>
<td>10</td>
<td>-</td>
<td>118750347</td>
<td>118750391</td>
<td>45</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>29</td>
<td>96</td>
<td>131</td>
<td>180</td>
<td>91.5%</td>
<td>X</td>
<td>-</td>
<td>136658996</td>
<td>136659031</td>
<td>36</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>29</td>
<td>100</td>
<td>131</td>
<td>180</td>
<td>96.9%</td>
<td>15</td>
<td>-</td>
<td>92642190</td>
<td>92642222</td>
<td>33</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>28</td>
<td>92</td>
<td>133</td>
<td>180</td>
<td>96.8%</td>
<td>6</td>
<td>-</td>
<td>117139048</td>
<td>117139089</td>
<td>42</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>28</td>
<td>93</td>
<td>131</td>
<td>180</td>
<td>86.9%</td>
<td>2</td>
<td>-</td>
<td>3447436</td>
<td>3447474</td>
<td>39</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>28</td>
<td>94</td>
<td>127</td>
<td>180</td>
<td>91.2%</td>
<td>18</td>
<td>+</td>
<td>56741649</td>
<td>56741682</td>
<td>34</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>27</td>
<td>112</td>
<td>144</td>
<td>180</td>
<td>96.7%</td>
<td>19</td>
<td>-</td>
<td>4137859</td>
<td>4137896</td>
<td>38</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>26</td>
<td>96</td>
<td>131</td>
<td>180</td>
<td>96.6%</td>
<td>10</td>
<td>-</td>
<td>86215205</td>
<td>86215243</td>
<td>39</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>26</td>
<td>96</td>
<td>131</td>
<td>180</td>
<td>89.7%</td>
<td>4</td>
<td>+</td>
<td>116931002</td>
<td>116931036</td>
<td>35</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>25</td>
<td>100</td>
<td>133</td>
<td>180</td>
<td>88.3%</td>
<td>13</td>
<td>-</td>
<td>108252923</td>
<td>108252957</td>
<td>35</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>23</td>
<td>110</td>
<td>133</td>
<td>180</td>
<td>100.0%</td>
<td>2</td>
<td>-</td>
<td>18841397</td>
<td>18841421</td>
<td>25</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>22</td>
<td>102</td>
<td>127</td>
<td>180</td>
<td>92.4%</td>
<td>7</td>
<td>-</td>
<td>140183726</td>
<td>140183751</td>
<td>26</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>22</td>
<td>111</td>
<td>133</td>
<td>180</td>
<td>100.0%</td>
<td>16</td>
<td>-</td>
<td>32227453</td>
<td>32227476</td>
<td>24</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>22</td>
<td>111</td>
<td>133</td>
<td>180</td>
<td>100.0%</td>
<td>14</td>
<td>-</td>
<td>31622869</td>
<td>31622892</td>
<td>24</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>22</td>
<td>111</td>
<td>133</td>
<td>180</td>
<td>100.0%</td>
<td>7</td>
<td>+</td>
<td>151248430</td>
<td>151248453</td>
<td>24</td>
</tr>
<tr>
<td>browser details</td>
<td>YourSeq</td>
<td>22</td>
<td>111</td>
<td>133</td>
<td>180</td>
<td>100.0%</td>
<td>2</td>
<td>+</td>
<td>169088471</td>
<td>169088494</td>
<td>24</td>
</tr>
</tbody>
</table>
Alignment of YourSeq and chr7:148029624-148029788

Click on links in the frame to the left to navigate through the alignment. Matching bases in cDNA and genomic sequences are colored blue and capitalized. Light blue bases mark the boundaries of gaps in either sequence (often splice sites).

**cDNA YourSeq**

```
ATCGTGATAC AGAGAGAAGA CCCGCCAGAC GAGGTTCGCC ACAGACCAAA 50
TCCTGAGAAG TAGGGGcacaat ctcCTTCAAT ACTGgGggtt acCCCCTc7T 100
CTCAGATAAC TCTACCTTAAC GCTGAAGTGAC AAGGAAACAA CAAACAAAGA 150
CAGTGTGCTCA GCGATGCTGT TAAGACCAAGG
```

**Genomic chr7**:

```
gccctgtctct tcatgaggttct ctaaacctgct ttcctttatcact cacccagggag 148029573
agttgcttag ggggttggtctg gcccataagat ggttgtgagct tctccccatcata 148029622
ATCGTGATAC AGAGAGAAGA CCCGCCAGAC GAGGTTCGCC ACAGACCAAA 50
TCCTGAGAAG TAGGGGcacaat ctcCTTCAAT ACTGgGggtt acCCCCTc7T 100
CTCAGATAAC TCTACCTTAAC GCTGAAGTGAC AAGGAAACAA CAAACAAAGA 150
CAGTGTGCTCA GCGATGCTGT TAAGACCAAGG
```

**Side by Side Alignment**

```
000000001 atcagtgtcatacagagataatccccccccacagggcctttccccacagaccata 000000049
000000050 tagaggcc 0000000066
000000075 cttcacaactggg 0000000086
```

80
Sequence Assembly
Sequence Assembly

A fundamental goal of DNA sequencing has been to generate large, continuous regions of DNA sequence.

Whole-genome shotgun proven to be the most cost-effective and least labour intensive method of sequencing:

- Human genome completed by a BAC-by-BAC strategy.

Capillary sequencing reads ~600-800bp in length:

- Overlap based assembly algorithms (phrap, phusion, arachne, pcap…)
- Compute all overlaps of reads and then resolve the overlaps to generate the assembly.

Volume + read length of data from next-gen sequencing machines meant that the read-centric overlap approaches were not feasible:

- 1980’s Pevzner et al. introduced an alternative assembly framework based on de Bruijn graph.
- Based on a idea of a graph with fixed-length subsequences (k-mers).
- Key is that not storing read sequences – just k-mer abundance information in a graph structure.
Sequence Assembly

Sequence reads → Sequence contigs → Scaffolds → Mapped scaffolds → Genome map

Sequence-tagged sites (STS) – 200-500bp sequence, occurs 1 time in genome at known location

Green, E.D. (2001) NRG 2, 573-580
De Bruijn Graph Construction 1

Genome is sampled with random sequencing 7bp reads (e.g. Illumina or 454)
Note errors in the reads are represented in red

Flicek & Birney (2009) Nat Meth, 6
The $k$-mers in the reads (4-mers in this example) are collected into nodes and the coverage at each node is recorded (numbers at nodes)

Features

- continuous linear stretches within the graph
- Sequencing errors are low frequency tips in the graph

Note: base with error is last base in $k$-mer that starts ‘tip’
De Bruijn Graph Construction 3

Graph is simplified to combine nodes that are associated with the continuous linear stretches into single, larger nodes of various k-mer sizes.

Error correction removes the tips and bubbles that result from sequencing errors.

Final graph structure that accurately and completely describes in the original genome sequence.
Next-gen Assemblers

First de Bruijn based assembler was Newbler
  ▸ Adapted it to handle main source of error in 454 data – indels in homopolymer tracts
Many de Bruijn assemblers subsequently developed
  ▸ SHARCQS, VCAKE, VELVET, EULER-SR, EDENA, ABySS and ALLPATHS
  ▸ Most can use mate-pair information
Few next-gen assemblers capable of assembling mammalian sized genomes out of the box
  ▸ ABySS – distributes de Bruijn graph over a network of computers using MPI protocol
  ▸ SOAP (BGI) and Cortex (unpublished)
    ▸ Key: early removal of spurious errors from the data
Short Read Alignment
There are many short-read aligners...

- Bfast
- BioScope
- Bowtie
- BWA
- CLC bio
- CloudBurst
- Eland/Eland2
- GenomeMapper
- GnuMap
- Karma
- MAQ
- MOM
- Mosaik
- MrFAST/MrsFAST
- NovoAlign
- PASS
- PerM
- RazerS
- RMAP
- SSAHA2
- Segemehl
- SeqMap
- SHRiMP
- Slider/SliderI
- SOAP/SOAP2
- Srprism
- Stampy
- vmatch
- ZOOM
- ......
Short Read Alignment

Next-gen considerations

- Amount of data many orders of magnitude higher => memory + speed
- Different error profiles than the previous-generation technology
  - Roche 454 reads to have insertion or deletion errors at homopolymers
  - Illumina low quality calls can occur anywhere in a read
  - Increasing likelihood of sequence errors toward the end of the read for ABI SOLiD + illumina
- Output of the SOLiD machine is a series of colors representing two nucleotides

Alignment itself is the process of determining the most likely source within the genome sequence for the observed DNA sequencing read.

DNA based alignment mostly

- Very small evolutionary distances (human-human, strains of the reference genome)
- Assumptions about the number of expected mismatches
  - Allow for much faster processing
Short Read Alignment Algorithms

Large and ever growing number of implementations for short-read sequence alignment

Fundamental algorithms can be divided into two approaches
  - Hash table based implementations
  - Burrows-Wheeler transform (BWT)

Both approaches apply to sequence and colour space and all technologies

Heuristics to find potential locations on the genome
  - Slower more accurate alignment run on a subset of potential locations
  - Similar strategy to traditional read alignment algorithms: Blat, Blast, SSAHA2

Constant trade-off: speed vs. sensitivity
Guaranteed high accuracy will generally take longer
Hash Table Based Alignment

Hash table is a common data structure that is able to index complex and non-sequential data in a way that facilitates rapid searching.

Hash of the reads: MAQ, ELAND, ZOOM and SHRIIMP, RMAP
- Smaller but more variable memory requirements

Hash the reference: SOAP, BFAST and MOSAIK, PASS
- Advantage: constant memory cost
Hash Table Based Alignment

Hash is typically built from a set of seeds spaced along the reference or read sequence
Seeds used in the hash table creation and the reads have been associated with the region of the genome
Specialized and accurate alignment algorithm is used to determine the exact placement
  ▸ Gapped and ungapped versions of Smith-Waterman
  ▸ Use the base qualities in order to determine the most likely alignment of a read
  ▸ Use information about where the mate pair aligns to
    ▸ Maq: Require 1 mate to align ungapped and can do full Smith-Waterman to align its mate (e.g. indel in the mate)

Mapping quality
  ▸ A probability measure of the alignment being incorrect
  ▸ Single end mapping quality vs. paired end mapping quality
  ▸ Low complexity regions typically have low mapping qualities
  ▸ Typically represented as a phred score (i.e. Q10 = 1 in 10 incorrect, Q20 = 1 in 100 incorrect)

Differs by aligners, see ex) http://user.list.galaxyproject.org/about-Mapping-Quality-td4366680.html  Kevin Silverstein Feb 8, 2012 and http://genome.sph.umich.edu/wiki/Mapping_Quality_Scores
Other Data Structures in Aligners

<table>
<thead>
<tr>
<th>Data Structure</th>
<th>Memory Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brute Force</td>
<td>(3 GB)</td>
<td>Naive, Slow &amp; Easy</td>
</tr>
<tr>
<td>Suffix Array</td>
<td>(&gt;15 GB)</td>
<td>Vmatch, PacBio Aligner</td>
</tr>
<tr>
<td>Suffix Tree</td>
<td>(&gt;51 GB)</td>
<td>MUMmer, MUMmerGPU</td>
</tr>
<tr>
<td>Hash Table</td>
<td>(&gt;15 GB)</td>
<td>BLAST, MAQ, ZOOM, RMAP, CloudBurst</td>
</tr>
</tbody>
</table>

BANANA
BAN
ANA
NAN
ANA

Vmatch, PacBio Aligner
Binary Search
Tree Searching

BWT Based Alignment (Burrows-Wheeler Transform)

New generation of short read aligners based on BWT
- BWA, SOAP2, BOWTIE

BWT commonly used in data compression – compressed suffix array

FM index data structure
- Ferragina and Manzini - suffix array is much more efficient if it is created from the BWT sequence
- FM index retains the suffix array's ability for rapid subsequence search
- Similar or smaller in size than input genome

Two step creation process
- Sequence order of the reference genome is modified using the BWT
- Final index is created; it is then used for rapid read placement on the genome
BWT Based Alignment

1. All possible rotations

2. Sort

3. Select final column

Genomic sequence

Transform

Li and Durbin (2009) Bioinformatics for further details

(Bioinformatics (2009) 25 (14): 1754-1760)
Let $S(i)$ be the suffix array, let $B[i]$ be the BWT transform of $X$. Let $W=\text{`go'}$ be a query substring of $X$. Then the **SA interval** of all occurrences of $W$ is $[1,2]$. The suffix array values for this interval are 3 and 0 which give all positions of the occurrences of `go'. Procedure called `backward search' uses properties of $W$ and $B$ to efficiently calculate this. See Li and Durbin ([Bioinformatics (2009) 25 (14): 1754-1760] errata [http://bio-bwa.sourceforge.net/](http://bio-bwa.sourceforge.net/))
Let $S(i)$ be the suffix array, let $B[i]$ be the BWT transform of $X$. Let $W='go'$ be a query substring of $X$. Then the SA interval of all occurrences of $W$ is $[1,2]$. The suffix array values for this interval are $3$ and $0$ which give all positions of the occurrences of ‘go’. Procedure called ‘backward search’ uses properties of $W$ and $B$ to efficiently calculate this. See Li and Durbin (Bioinformatics (2009) 25 (14): 1754-1760) errata [link](http://bio-bwa.sourceforge.net/)
**BWA: last-to-front (LF) property**

- Ordering of characters in last column is same ordering of characters in the first column, so if sort last column, can recreate First Column
- Also know in Original sequence, characters in First column follow characters in the Last column
- Can use these to reconstruct X

![Diagram showing string sorting and character reconstruction](image)
BWA: Other Helpful Tutorials

• http://www.homolog.us/blogs/blog/2011/10/05/finding-us-in-homolog-us-part-ii/
Which approach?

BWT implementations are much faster than their hash-based counterparts

- Several times faster still at slightly reduced sensitivity levels
- BOWTIE's reported 30-fold speed increase over hash-based MAQ with small loss in sensitivity
- Limitations to BWT approaches: BWA is only able to find alignments within a certain 'edit distance'
  - 100-bp reads, BWA allows 5 'edits'
  - Important to quality clip reads (-q in BWA)
    - Non-A/C/G/T bases on reads are simply treated as mismatches

Hash based approaches are more suitable for divergent alignments

- Rule of thumb: <2% divergence -> BWT
  - E.g. human alignments
- >2% divergence -> hash based approach
  - E.g. wild mouse strains alignments
Alignment Limitations

Read Length and complexity of the genome
- Very short reads difficult to align confidently to the genome
- Low complexity genomes present difficulties
  - Malaria is 80% AT rich – lots of low complexity AT stretches

Alignment around indels
- Next-gen alignments tend to accumulate false SNPs near true indel positions due to misalignment
- Smith-Waterman scoring schemes generally penalise a SNP less than a gap open
- New tools being developed to do a second pass on a BAM and locally realign the reads around indels and ‘correct’ the read alignments (ex. SRMA)

High density SNP regions
- Seed and extend based aligners can have an upper limit on the number of consecutive SNPs in seed region of read (e.g. Maq – max of 2 mismatches in first 28bp of read)
- BWT based aligners work best at low divergence
Example Misalignments

Newbler on Illumina reads: left-side bias when inappropriate gap penalty

Multiple sequence realignment around indels
Experimental Design

Choosing right sequencing technology to get optimal results for experiment

Experiment 1: “I want to determine the genome of a new fungi species with no closely related reference genome”
- Whole-genome sequencing
- De novo assembly with no reference
  - Longer reads might be more useful – 454?
- Mixture of fragment sizes
  - 200, 500, 3kb, 5kb, 10kb
  - Short range pairing information and long range information for scaffolding

Experiment 2: “I want to measure the relative expression level differences of one yeast species under different environmental conditions”
- Sequence the transcriptome (RNA-seq)
- Illumina or SOLiD sequencing for high depth
  - Multiplex the sequencing into a single lane
- Measure the relative expression levels by aligning
  - e.g. use Cufflinks to detect differential expression across the samples
Experimental Design

Experiment 4: “I want to catalog all of the structural variants in a human cancer cell vs. the normal cell for as little cost as possible”
- Fragment coverage vs. sequence coverage
- SVs are called from discordant read pairs – long range information
  - Sequence coverage not important
  - Require fragment coverage
- Sequence multiple paired libraries with short read length
  - E.g. 1000bp in total capacity
  - 100bp reads = 5RPs = 5 fragments x 500bp per fragment = 2.5Kbp fragment coverage
  - 40bp reads = 25RPs = 25 fragments x 500bp per fragment = 12.5Kbp fragment coverage
  - More fragments sequenced = more independent sources of evidence to call structural variants

Experiment 5: “I have 3 patients with a rare condition and want to find the causitive variant”
- High depth sequencing (20x?) per patient. Illumina or SOLiD or complete genomics
- Exome sequencing – 1 lane per patient
- SNPs + short indels
- Exclude all common variation (dbSNP + 1000Genomes)
- Is there a shared truncating variant? If not – is there a shared truncating structural variant?
Additional Resources


• David Pollock’s slides from previous years (especially math for detecting coverage needed to detect sequence errors)
Not used
Elements of Genetic Material

http://www.landesbioscience.com/curie/images/chapters/Meissner2color.jpg
Mapping RNA-Seq reads

Option 1: map reads to database of transcriptome (i.e coding sequences)

Option 2: do gapped alignments to genomic database
TopHat: Leading contender

- Part of ‘tuxedo’ suite:
  - Bowtie
  - TopHat
  - Cufflinks
  - Cummerbund

- Trapnell et al PMID: 19289445
Two approaches: Public and Celera

- **International Human Genome Sequencing Consortium (IHGSC)**
  - 200 labs in US, 18 countries
  - Shotgun project: Sequence 1 bacterial artificial chromosome (BAC) clone (150-350 Kbp inserts) at a time
  - Use previously established physical map (like STS) to orient BAC in relation to each other
  - STS have single occurrence in genome at known location
  - Planned 15 yrs, 1990-2000 for $2.7B (start $1-$10 per base, end 10-20¢)
Two approaches: Public and Celera

- **Celera: Whole genome Shotgun Assembly**
  - Shoot first, ask questions later
  - Shotgun all DNA, not just within BAC
  - Used combination of fragment and mate pairs with known insert size
  - Started 1998-2000
<table>
<thead>
<tr>
<th></th>
<th>Sang er</th>
<th>Roche 454 GS FLX</th>
<th>ABI SOLiD</th>
<th>Illumina GAI1</th>
<th>Illumina MiSeq</th>
<th>Illumina HiSeq 2000</th>
<th>Ion PGM</th>
<th>Ion Proton</th>
<th>Helicos tSMS</th>
<th>PacBio RS II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample</td>
<td>1 - 5μg</td>
<td>2 -20 μg</td>
<td>&lt;1 μg</td>
<td>1ng-1 μg</td>
<td>50ng-1 μg</td>
<td>&lt;1-10 μg</td>
<td>&lt;2 μg</td>
<td>250-1000ng</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Library Prep</td>
<td>-</td>
<td>3-4 days</td>
<td>2-5 days</td>
<td>6 hrs</td>
<td>1.5h</td>
<td>6 hours</td>
<td>8 hrs</td>
<td>8 hrs</td>
<td>1 day</td>
<td></td>
</tr>
<tr>
<td>Amplification method</td>
<td>-</td>
<td>Bead-based emulsion PCR</td>
<td>Bead-based emulsion PCR</td>
<td>Bridge amplification on flow cell</td>
<td>Bridge amplification on flow cell</td>
<td>Bead-based emulsion PCR</td>
<td>Bead-based emulsion PCR</td>
<td>n/a single molecule</td>
<td>n/a single molecule</td>
<td></td>
</tr>
<tr>
<td>Amplify</td>
<td>-</td>
<td>2 days</td>
<td>2 days</td>
<td>4 hours</td>
<td>Incl in seq</td>
<td>4h</td>
<td>&lt;1 day</td>
<td>&lt;1 day</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sequencing method</td>
<td>Dye terminators</td>
<td>Pyro-sequencing</td>
<td>Ligation( dibase)</td>
<td>Reversible dye terminators SBS</td>
<td>Reversible dye terminators</td>
<td>Semi-conductor</td>
<td>Semi-conductor</td>
<td>Virtual terminator SBS</td>
<td>SMRT cell</td>
<td></td>
</tr>
<tr>
<td>Sequencing</td>
<td>3 h</td>
<td>10 h</td>
<td>6-10 d</td>
<td>2-14 d</td>
<td>6h-3d</td>
<td>3-12d</td>
<td>2.3-7.3 h</td>
<td>2-6 h</td>
<td>8-9 d</td>
<td>10hr</td>
</tr>
<tr>
<td>Read length (bp)</td>
<td>800</td>
<td>400 -800</td>
<td>35-75</td>
<td>36-150</td>
<td>50-250</td>
<td>35-100</td>
<td>35-400</td>
<td>75-250</td>
<td>45-50</td>
<td>4k-23k</td>
</tr>
<tr>
<td>Reads/Run</td>
<td>500 K</td>
<td>600 M</td>
<td>320-640M</td>
<td>12-20 M</td>
<td>3-6 B (200M x 8)</td>
<td>0.5-6 M</td>
<td>90 M</td>
<td>12-20M</td>
<td>47 K</td>
<td></td>
</tr>
<tr>
<td>Bases/Run</td>
<td>800 bp</td>
<td>400-500 Mb</td>
<td>10-20 Gb</td>
<td>3-6 Gb</td>
<td>1-7.5 Gb</td>
<td>7.5-35Gb/lane</td>
<td>30Mb-2.2Gb</td>
<td>7 Gb</td>
<td>21-28 Gb</td>
<td>217 Mb</td>
</tr>
<tr>
<td>Cost/Run</td>
<td>$5/750 bp</td>
<td>~$8k</td>
<td>$25k</td>
<td>~$9k</td>
<td>$1.5k</td>
<td>$2k/lane</td>
<td>$400-$1000</td>
<td>$1500</td>
<td>Out of business!</td>
<td></td>
</tr>
<tr>
<td>Cost/Mb</td>
<td>$5/750 bp</td>
<td>$16</td>
<td>$1.65</td>
<td>$1.50</td>
<td>$0.30</td>
<td>$1.06</td>
<td>$0.45</td>
<td>$0.21</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Mapping to Reference With Discontinuous Words

- For a read length of 15, we can find all alignments with 1 mismatch (15_1) using **discontinuous words** in the 3 schemas of word size 10

<table>
<thead>
<tr>
<th>Schema_15_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>111111111100000</td>
</tr>
<tr>
<td>00000111111111</td>
</tr>
<tr>
<td>11111000001111</td>
</tr>
</tbody>
</table>

- Using **continuous words**, word size must be at most 7 to find all alignments with 1 mismatch
  - 40 times slower than the three schemas above
Schema Set for 25-mer Read, Word Size = 14

# 14 base index on 25, 0 mismatches
00000000001111111111111

# 14 base index on 25, 1 mismatches
111111111111110000000000
111110000000000011111111
00000000001111111111111

# 14 base index on 25, 2 mismatches
000001101111001110110111
1101100010001010101011110
11110111011100100010001001
100011110011110011011010
11001110011110011011010
01111001010001011101010011
101000011100111101101101
010101111011011001100100

More mismatches
longer run time
Local Alignment Strategy

- Map the first 25 colors of the read to allowing 2 mismatches (MM).
- For every hit found (up to the Z-limit), do a local extension
  - Accumulate alignment score (Match = 1, MM = -2 [user defined])
  - Report the best partial alignment (anchored local) based on score
    - Discard if score does not meet minimum cutoff

Read: 0122130123012303201203021123012310231203120103120
    ||||||||| ||||||||| ||||| ||||| ||||| ||||| ||||| |||||
Ref: 0122130023012303201203011123011310231203100101203

- For reads not mapped, shift anchor location and attempt additional mapping
Mapping QV: Working Examples

- Read maps two places, both with one mismatch
  - Each has equal chance of being correct

<table>
<thead>
<tr>
<th>Read: 0122130123012303201203021</th>
<th>Read: 0122130123012303201203021</th>
</tr>
</thead>
<tbody>
<tr>
<td>mqv=0</td>
<td>mqv=0</td>
</tr>
<tr>
<td>Ref: 0122130123012303201203011</td>
<td>Ref: 0122130123012303201203021</td>
</tr>
</tbody>
</table>

- Read maps two places, one with zero mismatches, and one with two mismatches
  - Higher likelihood the true alignment is the perfect alignment

<table>
<thead>
<tr>
<th>Read: T...0120123012303201203011</th>
<th>Read: T...0120123012303201203021</th>
</tr>
</thead>
<tbody>
<tr>
<td>mqv=22</td>
<td>mqv=0</td>
</tr>
<tr>
<td>Ref: ...0120123012303201203011</td>
<td>Ref: ...0120123002303201223021</td>
</tr>
</tbody>
</table>

- Mapping QV largely depends on the difference of the hits
Homework: Due 9/30/2013

• Read

• Compute global alignment table (superimpose backtrack) and show a possible final optimal alignment for the following example, using cost function:

  \[ \sigma = -1 \text{ mismatch, } +2 \text{ match} \]

  \[ \begin{align*}
  S &: \text{ agcg}\text{tg} \\
  T &: \text{ agg}\text{g}\text{c}\text{g}
  \end{align*} \]

• Show all work, using tables like Slide 55 and 57 (template table provided on following slide)
### Needleman-Wunsch: Dynamic Programming

\[ \sigma = XX \text{ mismatch, } XX \text{ match} \]

<table>
<thead>
<tr>
<th>( T_j )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_i )</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

M: \( V(-1,-1) + \sigma(i,j) = M + s = x \)
D: \( V(-1,j) + \sigma(i,-) = D + 1 = y \)
I: \( V(i,-1) + \sigma(-,j) = I + 1 = z \)