High-throughput sequence alignment

Analysis of Biological Sequences 140.638
a little history

human genome project #1 (many U.S. government agencies and large institute) started October 1, 1990. Goal: 10x coverage of human genome, release data daily, no patents

BAC: 150kb

ABI PRISM, 500bp reads from both ends
a little history

human genome project #2. Craig Venter, Perkin-Elmer, started May 1998. Goal: 10x coverage, release data quarterly, patent “only 100-300 important genes”
outcome: both groups finished assemblies “at the same time”

each group required sequences from the other, in order to complete their assemblies.

major contribution from government group: good internal bookkeeping, built mechanism for storing large sequences in GenBank

major contribution from Venter’s group: demonstrated that shotgun sequencing can work
articles

Initial sequencing and analysis of the human genome

International Human Genome Sequencing Consortium*

*A partial list of authors appears on the opposite page. Affiliations are listed at the end of the paper.

The human genome holds an extraordinary trove of information about human development, physiology, medicine and evolution. Here we report the results of an international collaboration to produce and make freely available a draft sequence of the human genome. We also present an initial analysis of the data, describing some of the insights that can be gleaned from the sequence.

The Sequence of the Human Genome

J. Craig Venter,1* Mark D. Adams,1 Eugene W. Myers,1 Peter W. Li,1 Richard J. Mural,1 Granger G. Sutton,1 Hamilton O. Smith,1 Mark Yandell,1 Cheryl A. Evans,1 Robert A. Holt,1 Jeannine D. Gocayne,1 Peter Amanatides,1 Richard M. Ballew,1 Daniel H. Huson,1 Jennifer Russo Wortman,1 Qing Zhang,1 Chinnappa D. Kodira,1 Xiangqun H. Zheng,1 Lin Chen,1 Marian Skupski,1 Gangadharan Subramanian,1 Paul D. Thomas,1 Jinghui Zhang,1 George L. Gabor Miklos,2 Catherine Nelson,3 Samuel Broder,1 Andrew G. Clark,4 Joe Nadeau,5

February 15, 2001 Nature

February 16, 2001 Science
<table>
<thead>
<tr>
<th>high-throughput sequencing platforms (table obsolete by 1/2020)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>second generation</strong></td>
</tr>
<tr>
<td>read length</td>
</tr>
<tr>
<td>Roche 454 (not available)</td>
</tr>
<tr>
<td>Illumina HiSeq 2000/2500/XTen/3000/4000</td>
</tr>
<tr>
<td>Illumina MiSeq/NextSeq/NovaSeq</td>
</tr>
<tr>
<td>SOLiD (not available)</td>
</tr>
<tr>
<td>Ion Torrent</td>
</tr>
<tr>
<td>Pacific Biosciences</td>
</tr>
<tr>
<td>Oxford Nanopore MinION</td>
</tr>
</tbody>
</table>
$1000 genome is here (batteries not included)

range of “normal” genomes is unknown!
copy number variation, chromosome abnormalities, single base changes, indels . . .

availability of data & technology is driving a revolutionary change in genetics and biology
mechanics of high throughput sequencing

A. Library Preparation

Genomic DNA → Fragmentation → Adapters → Ligation → Sequencing Library

Fragment DNA, ligate adapters to both ends

NGS library is prepared by fragmenting a gDNA sample and ligating specialized adapters to both fragment ends.
mechanics of high throughput sequencing

B. Cluster Amplification

Flow Cell

Bridge Amplification
Cycles

Clusters

Library is loaded into a flow cell and the fragments hybridize to the flow cell surface. Each bound fragment is amplified into a clonal cluster through bridge amplification.

hybridize library to the surface of the flow cell
mechanics of high throughput sequencing

DNA synthesis adds fluorescently labeled bases, one per cycle.
Illumina

Illumina Sequencing Technology

DNA (0.1-1.0 ug) → Sample preparation → Cluster growth → Sequencing → Image acquisition → Base calling

TGATAGCAT
mechanics of high throughput sequencing

D. Alignment & Data Analysis

Reads

AGATGGTATTG
GATGCCATTGCAA
GCTTGGCAATTTTGAC
ATGCCATTGCAATT
AGATGGGATTGCAATTTTG

Reference Genome

AGATGGTATTGCAATTTTGACAT

Reads are aligned to a reference sequence with bioinformatics software. After alignment, differences between the reference genome and the newly sequenced reads can be identified.
Primary output of second-generation sequencers: images. lots and lots of images.

Each dot = one bead/well
Each pixel is tracked through subsequent cycles

Image processing is critical, time-consuming, and images are HUGE

2000 panels per slide
5 images per panel, per cycle
6 MB per image
~3TB of images per sequencing run, per slide
Illumina output

read = one contiguous piece of sequence returned from the instrument

fragment = one read that comes from one end of a DNA segment

paired end = two reads that come from both ends of a DNA segment (usually returned in two separate files)
Things to know:

- current slide capacities range from 200M reads/lane to 500M reads/lane
- an entire human genome can be sequenced up to 40x depth in 2-3 lanes
- barcoding is straightforward for applications that don’t require so many reads (ChIP, RNAseq, exome etc etc)
Ion Torrent

very fancy pH meter!
Ion Torrent

one nucleotide at a time washed over wells; incorporation detected by pH change
Ion Torrent

incorporating multiples of the same nucleotide gives more pH change
Ion Torrent

Problems include issues in resolving length of homopolymer repeats, short reads, low throughput

Ion Torrent read:  TCGAAAAAGCCGT
actual sequence:    TCGAAAAGCCGT

Main use (currently) is in sequencing targeted gene panels (genes amplified by PCR prior to sequencing)

• HLA typing
• cancer genes (hotspot panels available)
• trisomy 21 detection
Detecting modified nucleotides during sequencing?

Flusberg et al, Nature Methods 2010
Pacific Biosciences: circular consensus sequences

Both the sense and antisense strand are sequenced (typically several times), and the sequences are merged, greatly increasing accuracy. Most errors are insertions though homopolymers are also problematic.

Travers et al, NAR 2010
nanopores

Idea: single strand of DNA threads through an organic or synthetic pore in a membrane

Nucleotides are detected by their electrical impedance as they pass through the membrane

Jain et al Nature Methods 2015
nanopore sequencing

Oxford Nanopore Technologies:
MinION
PromethION and SmidgION

read lengths 10kb-100kb+
tens of thousands of reads
error ~ 10%
steps in analysis

DNA:
1) align to a reference genome, OR assemble
2) find variation (CNV, SNV, indels, translocations…) or piled-up reads (ChIP, RNAseq)
3) figure out the significance of the variation

RNA:
1) spliced alignment
2) transcriptome assembly
3) isoform usage
4) differential expression
steps in analysis

Cancer
1) tumor-normal sequence differences (somatic variation) and their consequences
2) expression changes

ChIP/methylation
1) locate marks
2) comparison among samples
3) correlation to other variants
Steps in analysis

Many specialized applications!

Microbiome/metagenomics
miRNA and other small RNA species
Ribosomal profiling

And more…
@SRR1067050.4090053/1
GGGTGGTTCCATGTCTTTGCTATTGTAAATAGTGCTGCAATAAACATACATGTGCATGTGTCTTTATAGCAGAATCATTTATATTCCTTTGGGTATATAC
+
@@::@<DD>FBF9,AFAHCBDAFHIAFFE@DFIEF@3DFFFFFDFBB3FD9BBF@DF?DFGFGCGCDGICFCCC@DEE@D>EEFEC??BD>);

name of read

sequence of read

score for each nucleotide in read

reserved field

quality score = \(-10\log_{10}(e)\)

where \(e\) is the probability that the nucleotide is wrong
if \(e = 1/1000\), quality score is 30

quality scores are written as ASCII characters, one per base, adding 33 to each quality score before converting to ASCII
high throughput sequence alignment

Two common goals: align to a reference genome OR do something de novo

perform an optimal alignment using Smith-Waterman or Needleman-Wunsch?
align each read to a database?
new algorithms?
high throughput sequence alignment

Considerations:

- very large number of reads to align: must be efficient, scalable
- read quality varies--bases are not equally reliable
- paired end reads
- non-unique reads (multireads)
- unmapped reads
- data storage
- data transfer
- reproducibility
preparing the reference genome

remember BLAST? reference genome is pre-indexed

Short read alignment programs use similar strategies to increase speed of alignment.

many current strategies, including

q-gram tables

tries (prefix and suffix)

BWT/FM index

all of these use some variation on a seed-and-extend method to get the final alignment
core idea: align small pieces first

Given a short sequence read, I want all matches to a reference, with at most k errors (this number is decided in advance)

I divide my sequence into $k+1$ pieces (seeds), the lengths of which depend on the length of the initial read. At least one of these must have an exact match, if my sequence does align to the reference as above. Why?

$k=4$
q-gram index

Here, the seeds are always created with the same predetermined size $q$, no matter how many mismatches are allowed in the entire read. Seeds are aligned with no mismatches but alignments will be extended.

2-gram index for ttatctctta; looking up ta

Figure 5
A 2-gram index of ttatctctta. To look up all occurrences of the 2-gram ta, one first determines its numerical value code, (ta) = 12. At positions 12 and 13, the lookup table “dir” stores the boundaries 3 and 5 (excluding) of the group of occurrences in the occurrence table “pos.” Therefore, positions 1 and 8 are the only occurrences of ta in the text.

search trees

suffix (and prefix) trees enable searching for seeds of arbitrary length, with or without mismatches.

Index structures are not as memory-efficient as others but searching is fast.
suffix tree for aligning to “banana”

generalized suffix tree for aligning to babb and abab

(j, i), are suffixes of the jth sequence starting at position i.

Burroughs-Wheeler transform & search

original string: gtgaaactggt

<table>
<thead>
<tr>
<th>g t g a a a c t g g t $</th>
</tr>
</thead>
<tbody>
<tr>
<td>t g a a a c t g g t $ g</td>
</tr>
<tr>
<td>g a a a c t g g t $ g t</td>
</tr>
<tr>
<td>a a a c t g g t $ g t g</td>
</tr>
<tr>
<td>a a c t g g t $ g t g a</td>
</tr>
<tr>
<td>a c t g g t $ g t g a a</td>
</tr>
<tr>
<td>c t g g t $ g t g a a a</td>
</tr>
<tr>
<td>t g g t $ g t g a a a c</td>
</tr>
<tr>
<td>g g t $ g t g a a a c t</td>
</tr>
<tr>
<td>g t $ g t g a a a c t g</td>
</tr>
<tr>
<td>t $ g t g a a a c t g g</td>
</tr>
<tr>
<td>$ g t g a a a c t g g t</td>
</tr>
</tbody>
</table>

1) add $ to end
2) create cyclic rotations
**BWT & search**

original string ("database"): `gtgaaactggt`

<table>
<thead>
<tr>
<th>Sort alphabetically</th>
<th><code>gtgaaactggt</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>$</code> <code>g</code> <code>t</code> <code>g</code> <code>a</code> <code>a</code> <code>a</code> <code>c</code> <code>t</code> <code>g</code> <code>g</code> <code>t_0</code></td>
<td></td>
</tr>
<tr>
<td><code>a_0</code> <code>a</code> <code>a</code> <code>c</code> <code>t</code> <code>g</code> <code>g</code> <code>t</code> <code>$</code> <code>c</code> <code>t</code> <code>g_0</code></td>
<td></td>
</tr>
<tr>
<td><code>a_1</code> <code>a</code> <code>c</code> <code>t</code> <code>g</code> <code>g</code> <code>t</code> <code>$</code> <code>c</code> <code>t</code> <code>g</code> <code>a_0</code></td>
<td></td>
</tr>
<tr>
<td><code>a_2</code> <code>c</code> <code>t</code> <code>g</code> <code>g</code> <code>t</code> <code>$</code> <code>c</code> <code>t</code> <code>g</code> <code>a</code> <code>a_1</code></td>
<td></td>
</tr>
<tr>
<td><code>c_0</code> <code>t</code> <code>g</code> <code>g</code> <code>t</code> <code>$</code> <code>g</code> <code>t</code> <code>g</code> <code>a</code> <code>a</code> <code>a_2</code></td>
<td></td>
</tr>
<tr>
<td><code>g_0</code> <code>a</code> <code>a</code> <code>a</code> <code>c</code> <code>t</code> <code>g</code> <code>g</code> <code>t</code> <code>$</code> <code>g</code> <code>t_1</code></td>
<td></td>
</tr>
<tr>
<td><code>g_1</code> <code>g</code> <code>t</code> <code>$</code> <code>g</code> <code>t</code> <code>g</code> <code>a</code> <code>a</code> <code>a</code> <code>c</code> <code>t_2</code></td>
<td></td>
</tr>
<tr>
<td><code>g_2</code> <code>t</code> <code>$</code> <code>g</code> <code>t</code> <code>g</code> <code>a</code> <code>a</code> <code>a</code> <code>c</code> <code>t</code> <code>g_1</code></td>
<td></td>
</tr>
<tr>
<td><code>g_3</code> <code>t</code> <code>g</code> <code>a</code> <code>a</code> <code>a</code> <code>c</code> <code>t</code> <code>g</code> <code>g</code> <code>t</code> <code>$</code></td>
<td></td>
</tr>
<tr>
<td><code>t_0</code> <code>$</code> <code>g</code> <code>t</code> <code>g</code> <code>a</code> <code>a</code> <code>a</code> <code>c</code> <code>t</code> <code>g</code> <code>g_2</code></td>
<td></td>
</tr>
<tr>
<td><code>t_1</code> <code>g</code> <code>a</code> <code>a</code> <code>a</code> <code>c</code> <code>t</code> <code>g</code> <code>g</code> <code>t</code> <code>$</code> <code>g</code> <code>g_3</code></td>
<td></td>
</tr>
<tr>
<td><code>t_2</code> <code>g</code> <code>g</code> <code>t</code> <code>$</code> <code>g</code> <code>t</code> <code>g</code> <code>a</code> <code>a</code> <code>a</code> <code>c_0</code></td>
<td></td>
</tr>
</tbody>
</table>

first column is the "genome dictionary"

last column is the "transformation"

number the nucleotides based on position
# BWT & search

**original string:** `gtgaaactggt`

## match AACTG?

<p>| | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>g</td>
<td>t</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>g</td>
<td>t₀</td>
<td></td>
</tr>
<tr>
<td>a₀</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>c</td>
<td>t</td>
<td>g₀</td>
</tr>
<tr>
<td>a₁</td>
<td>a</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>a₀</td>
</tr>
<tr>
<td>a₂</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>a</td>
<td>a₁</td>
</tr>
<tr>
<td>c₀</td>
<td>t</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>g</td>
<td>t</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>a₂</td>
</tr>
<tr>
<td>g₀</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>g</td>
<td>t₁</td>
</tr>
<tr>
<td>g₁</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>g</td>
<td>t</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>t₂</td>
</tr>
<tr>
<td>g₂</td>
<td>t</td>
<td>$</td>
<td>g</td>
<td>t</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>t</td>
<td>g₁</td>
</tr>
<tr>
<td>g₃</td>
<td>t</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
</tr>
<tr>
<td>t₀</td>
<td>$</td>
<td>g</td>
<td>t</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>g₂</td>
</tr>
<tr>
<td>t₁</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>g₃</td>
</tr>
<tr>
<td>t₂</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>g</td>
<td>t</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>c₀</td>
</tr>
</tbody>
</table>
BWT & search
original string: gtgaaactggt

match AACTG?
BWT & search
original string: gtgaaactggt

match AACTG?

CTG

\[
\begin{array}{cccccccccccc}
\$ & g & t & g & a & a & a & c & t & g & g & t_0 \\
a_0 & a & a & c & t & g & g & t & $ & c & t & g_0 \\
a_1 & a & c & t & g & g & t & $ & c & t & g & a_0 \\
a_2 & c & t & g & g & t & $ & c & t & g & a & a_1 \\
c_0 & t & g & g & t & $ & g & t & g & a & a & a_2 \\
g_0 & a & a & a & c & t & g & g & t & $ & g & t_1 \\
g_1 & g & t & $ & g & t & g & a & a & a & c & t_2 \\
g_2 & t & $ & g & t & g & a & a & a & c & t & g_1 \\
g_3 & t & g & a & a & a & c & t & g & g & t & $ \\
t_0 & $ & g & t & g & a & a & a & c & t & g & g_2 \\
t_1 & g & a & a & a & c & t & g & g & t & $ & g_3 \\
t_2 & g & g & t & $ & g & t & g & a & a & a & c_0 \\
\end{array}
\]
BWT & search
original string: gtgaaactggt

match AACTG?

ACTG

$ g t g a a a c t g g t_0
a_0 a a c t g g t $ c t g_0
a_1 a c t g g t $ c t g a_0
a_2 c t g g t $ c t g a_1
c_0 t g g t $ g t g a a a_2
a_2 g_0 a a a c t g g t $ g t_1
a_1 g_1 g t $ g t g a a a c t_2
g_2 t $ g t g a a a c t g_1
g_3 t g a a a c t g g t $ t_0
a_0 t_0 $ g t g a a a c t g g t_2
t_1 g a a a c t g g t $ g_3
t_2 g g t $ g t g a a a c_0
**BWT & search**

**original string:** gtgaaactggt

**match AACTG?**

<table>
<thead>
<tr>
<th>AACTG</th>
<th>$</th>
<th>g</th>
<th>t</th>
<th>g</th>
<th>a</th>
<th>a</th>
<th>c</th>
<th>t</th>
<th>g</th>
<th>g</th>
<th>t_0</th>
</tr>
</thead>
<tbody>
<tr>
<td>a_0</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>c</td>
<td>t</td>
<td>g_0</td>
</tr>
<tr>
<td>a_1</td>
<td>a</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>a_0</td>
</tr>
<tr>
<td>a_2</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>a</td>
<td>a_1</td>
</tr>
<tr>
<td>c_0</td>
<td>t</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>g</td>
<td>t</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>a_2</td>
</tr>
<tr>
<td>g_0</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>g</td>
<td>t_1</td>
</tr>
<tr>
<td>g_1</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>g</td>
<td>t</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>t_2</td>
</tr>
<tr>
<td>g_2</td>
<td>t</td>
<td>$</td>
<td>g</td>
<td>t</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>t</td>
<td>g_1</td>
</tr>
<tr>
<td>g_3</td>
<td>t</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
</tr>
<tr>
<td>t_0</td>
<td>$</td>
<td>g</td>
<td>t</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>g_2</td>
</tr>
<tr>
<td>t_1</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>c</td>
<td>t</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>g_3</td>
</tr>
<tr>
<td>t_2</td>
<td>g</td>
<td>g</td>
<td>t</td>
<td>$</td>
<td>g</td>
<td>t</td>
<td>g</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>c_0</td>
</tr>
</tbody>
</table>
BWT is very, very compressible

whole human genome fully indexed and compressed (FM-index) in <3Gb (much smaller than a suffix array, and searching is just as fast)
now what?

once we find an alignment to our seed, we extend the alignment with some sort of dynamic programming.

What if there are multiple matches? will I find them??

What about paired ends?

What happened to those quality scores?

fragment = one read that comes from one end of a DNA segment

paired end = two reads that come from both ends of a DNA segment (usually returned in two separate files)
short read aligner output: SAM files

```
HWUSI-ES661_0016_FC629YPAAXX:2:1:1067:4986#TTAGGC  141  *  0  0  *
*  0  0  NTCTGATGTAAATCTCTTCTGAAACTGTGAAGGTAAGTTGTCATGC
BHQJLYYYVV^__[^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^][^]...

```
short read aligner output: SAM files

read name, flag, target_aln, position, mapping quality, CIGAR string, name of next read, position of next read, template length, aligned sequence, quality of aligned sequence

HWUSI-ES661_0016_FC629YPAAXX:2:1:1067:4986#TTAGGC
141  *       0       0       *       *       0       0
NTCTGATGTAATCTCTTCTGAAAATCTGTAACGAGGTGAAAGTTGCAAACTG
BHQJLYYYVV__[^^^][^^][^^][^^][^^][^^][^^][^^][^^][^^]
XM:i:0

HWUSI-ES661_0016_FC629YPAAXX:2:1:1067:10500#TTAGGC
99     B2112_final     14564   255     51M     =       14603   90
GAGGCCGACTACACGACGATCGACAAGGGCAACGGGACTTCGAGAC
babad^eeecdeeaeddeaacc^da^ddceae^\a^yl[Y`[T^dceccT
XA:i:1
MD:Z:14C23T12
NM:i:2

HWUSI-ES661_0016_FC629YPAAXX:2:1:1067:10500#TTAGGC
147     B2112_final     14603   255     51M     =       14564   -90
GACTTCGAGACGACGACCTCGTCTCCCACATCACCAACGACTCCAAA
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
XA:i:1
MD:Z:50C0
NM:i:1
sequence repeats in the human genome (aka multiple matches: a major problem)

<table>
<thead>
<tr>
<th>type</th>
<th>how many</th>
<th>total length</th>
<th>exact?</th>
</tr>
</thead>
<tbody>
<tr>
<td>telomeres (chromosome ends)</td>
<td>24<em>2</em>2</td>
<td>0.5Mbp</td>
<td>yes</td>
</tr>
<tr>
<td>centromeres (chromosome middles)</td>
<td>24*2</td>
<td>1-5Mbp</td>
<td>very similar</td>
</tr>
<tr>
<td>SINE transposons</td>
<td>~1M</td>
<td>350Mbp</td>
<td>nearly</td>
</tr>
<tr>
<td>LINE transposons</td>
<td>~500,000</td>
<td>700Mbp</td>
<td>no</td>
</tr>
<tr>
<td>tandem repeats (CACACA etc)</td>
<td>?</td>
<td>200Mbp</td>
<td>almost</td>
</tr>
<tr>
<td>gene families, orthologs, paralogs</td>
<td>?</td>
<td>?</td>
<td>no</td>
</tr>
</tbody>
</table>
problem with repeats

? fragment read matches equally well in many places

if part of a paired end read is unique, it may still be mappable
next up: nanopore reads!

so far, the algorithms for nanopore alignment are fairly similar to those for shorter read alignment, with tweaks to the implementation for scalability