Hello again!

- I’m a researcher in bioinformatics algorithms
- *de novo* assembly, big data alignment, k-mers, pangenomics. Well, week 1 stuff :)

@RayanChikhi on Twitter

http://rayan.chikhi.name
Course objectives

- **Enough background** to understand the alignment part of a biology article
- Increase confidence in using alignment tools
- Understand **why** alignment isn’t so straightforward
Course outline

- **Fundamentals**
- The many **flavors** and **tools** for pairwise DNA alignment
- **Multiple** sequence alignment
- Alignment to **databases**
- Into the unknown: profile and structure search
Questions to the audience

1. Have you ever run a sequence alignment software?
2. Was it willfully or as part of a pipeline?
3. Done multiple sequence alignment?
4. Know who/what Smith-Waterman is?
What’s an “alignment”? 

Given two (or more) sequences, determine how the residues best line up, to capture evolutionary relationships.
The many types of alignments

Pairwise (2 sequences)
The many types of alignments

Multiple sequences (>2)

<table>
<thead>
<tr>
<th>Seq ID</th>
<th>Sequence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q+S948</td>
<td>ROYTH</td>
<td>Multiple sequences (&gt;2)</td>
</tr>
<tr>
<td>RA9_HUMAN</td>
<td>MREDGKSYBEFLKIQGCTFTYDGKVRKCEQMEIAKAVYLVGKTRKEAIGYWN</td>
<td>Homo sapiens</td>
</tr>
<tr>
<td>RA9_MOUSE</td>
<td>MREDGKSYBEFLKIQGCTFTYDGKVRKCEQMEIAKAVYLVGKTRKEAIGYWN</td>
<td>Mus musculus</td>
</tr>
<tr>
<td>RA9_RAT</td>
<td>MREDGKSYBEFLKIQGCTFTYDGKVRKCEQMEIAKAVYLVGKTRKEAIGYWN</td>
<td>Rattus norvegicus</td>
</tr>
<tr>
<td>RA9_CROCK</td>
<td>MREDGKSYBEFLKIQGCTFTYDGKVRKCEQMEIAKAVYLVGKTRKEAIGYWN</td>
<td>Crocodylus porosus</td>
</tr>
<tr>
<td>RA9_SAMY</td>
<td>MREDGKSYBEFLKIQGCTFTYDGKVRKCEQMEIAKAVYLVGKTRKEAIGYWN</td>
<td>Salmo gairdneri</td>
</tr>
<tr>
<td>Q+S2033</td>
<td>BRAR</td>
<td>Multiple sequences (&gt;2)</td>
</tr>
<tr>
<td>RA9_ICPV0</td>
<td>MREDGKSYBEFLKIQGCTFTYDGKVRKCEQMEIAKAVYLVGKTRKEAIGYWN</td>
<td>Ictalurus punctatus</td>
</tr>
<tr>
<td>RA9_DROME</td>
<td>MREDGKSYBEFLKIQGCTFTYDGKVRKCEQMEIAKAVYLVGKTRKEAIGYWN</td>
<td>Danio rerio</td>
</tr>
<tr>
<td>RA9_DICD1</td>
<td>MREDGKSYBEFLKIQGCTFTYDGKVRKCEQMEIAKAVYLVGKTRKEAIGYWN</td>
<td>Danio rerio</td>
</tr>
<tr>
<td>Q+S4039</td>
<td>DICD1</td>
<td>Multiple sequences (&gt;2)</td>
</tr>
<tr>
<td>RA9_Play8</td>
<td>MREDGKSYBEFLKIQGCTFTYDGKVRKCEQMEIAKAVYLVGKTRKEAIGYWN</td>
<td>Drosophila melanogaster</td>
</tr>
</tbody>
</table>
The many types of alignments

1 sequence versus a database
The many types of alignments

1 sequence versus a profile
Why align?

One of the two pillars of sequence bioinformatics (with assembly).

Variant calling, RNA-seq quantification, taxonomic classification, etc..
How to do molecular biology

1. Sequences
2. Alignment
3. Tree, structure, function...
4. Publish

R.C. Edgar 2021,
https://www.youtube.com/watch?v=2HmjHStpu7I
What can be aligned? Many things:

- DNA vs DNA
- RNA vs RNA
- DNA vs RNA, DNA vs protein sequence, ..
- Protein sequence vs protein sequence
- Protein structure vs protein structure
Some vocabulary

**Query**: sequence to align

**Reference** (or **target**): sequence to align to

**Hit** (or **match** or **alignment**): part of query aligned to part of reference

**Homology**: shared ancestry

**Similarity, identity**: mathematical ways to detect homology

**String**: sequence

**Letter** (or **character** or **residue** or **monomer**): base pair or nucleotide or amino-acid
Pairwise DNA

General techniques
Global vs local

Global: must align all nucleotides, using insertions/deletions if necessary

Local: you’re allowed to skip beginning and/or end of either sequence
Alignment is based on scoring

What is a *good* alignment?
*One that* **minimizes** a penalty (or **maximizes** a score).

E.g. here a mismatch gives 1 penalty, a deletion gives 2 penalties:

<table>
<thead>
<tr>
<th>r:</th>
<th>TAC</th>
<th>r:</th>
<th>GAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>q:</td>
<td>TTC</td>
<td>q:</td>
<td>G-T</td>
</tr>
</tbody>
</table>

penalty=1  penalty=2
Example: (global alignment)

r: CAAGTTA
q: CAT-GGA

MMXDXXM

total penalty: 5

Is it the best we can do?

CAAGTTA

CATG-GA

MMXMDXXM

total penalty: 4 better!

(can also be aligned this way:)

(Here a mismatch gives 1 penalty, a deletion gives 2 penalties.)
CIGAR strings ("Concise Idiosyncratic Gapped Alignment Report")

A succession of M,X,I,D letters to represent an alignment.

- M = match
- I = insertion (gap in the target sequence)
- X = mismatch
- D = deletion (gap in the query sequence)

* some programs use M for both matches and mismatches \(_-(\)\_\) others use = instead of M

r: CAAGT TA
q: CAT-GGA

MMXDXXM (also written 2M1X1D2X1M), means: "to align the query to the target, do 2 matches, 1 mismatch, 1 deletion, 2 mismatches, 1 match".
Exercice 1

Write the CIGAR string for this alignment:

target:  GATCA-TGA

query:  G-CAACCA-
Solution

Write the CIGAR string for this alignment:

target: GATCA-TGA
query: G-CAACCA-

MDXXMIXXD

Quite high penalty alignment. It’s unlikely any tool would output it, as those two sequences are probably not evolutionarily related.
Is it possible to know the lowest possible penalty?

Yes, but you have to pay the price

(The price is a rather complex algorithm, that we’ll see next)
A special case: only mismatches

**Hamming** (= Manhattan) distance, A and B sequences of **same length**:  

*Minimum number of substitutions to turn sequence A into sequence B*

e.g.

ACTAGATG

CGTACATG
A special case: only mismatches

Hamming (= Manhattan) distance, A and B sequences of same length:

Minimum number of substitutions to turn sequence A into sequence B

e.g.

ACTAGATG

Hamming distance: 3

CGTACATG

Quick to calculate, just walk along both strings
A harder case: mismatches and indels

How to find **lowest penalty alignment** with **mismatches** AND **indels**?

*(We can no longer scan the seqs from left to right and decide on the fly.)*

To see this, consider aligning:

r: ACAG

q: AGACTG

**Novice level:**

ACAG--

AGACTG

penalty=2 X’s and 2 I’s

**Expert level:**

You must reach level 3 to unlock this content
Exercice 2

Find a good (=low penalty) global alignment for these two sequences:

ref: ACTAGATG
query: GTACAT

Give the CIGAR string

Given that:

- a mismatch (X) has 1 penalty,
- a deletion (D) has 2 penalty,
- a match (M) has no penalty

hint: no insertions
Solution

Find a good (=low penalty) global alignment for these two sequences:

ACTAGATG

-GTACAT-

DXMMXMMMD

total penalty = 6

a mismatch (X) has 1 penalty, a deletion (D) has 2 penalty, a match (M) has no penalty
Exercice

Just as a note, the best local alignment is:

ACTAGATG
GTACAT
XMMXMM

total penalty = 2

a mismatch (X) has 1 penalty,
a deletion (D) has 2 penalty,
a match (M) has no penalty
Penalties / scores

So far we’ve used penalties:

- a mismatch (X) has 1 penalty,
- a deletion (D) has 2 penalty,
- a match (M) has no penalty

We will now switch to scores:

- a mismatch (X) has -1 score,
- a deletion (D) has -2 scores,
- a match (M) has +1 score
Finding best alignments

Think about CIGAR strings, and imagine you’re ChatGPT.

Somebody gave you CATATGATGACAC to align.
CAGAGGGAATGCT

You output the CIGAR letters one by one. So far you’ve said:

MMXMXIMMIMIMMDMX

You are GPT5 so this is indeed the beginning of the best alignment:

CATAT-GA-TGACA...
CAGAGGGAATG-CT

What will be your next letter? If you have an incomplete CIGAR string just missing the last letter, then you have no choice for the last letter (M, X, D, or I? D here).
**The trick**

Optimal alignment $= \text{align(} \text{CATATGATGACAC, CAGAGGGAATGCT}\text{)}$ = Optimal alignment until the last CIGAR letter $+ \text{Last CIGAR letter}$

- $\text{align(} \text{CATATGATGACAC, CAGAGGGAATGCT}\text{)} = \text{align(} \text{CATATGATGACA, CAGAGGGAATGC}\text{)} + \text{CT (X)}$
- or $\text{align(} \text{CATATGATGACAC, CAGAGGGAATGCT}\text{)} = \text{align(} \text{CATATGATGACAC, CAGAGGGAATGC}\text{)} + \text{-T (I)}$
- or $\text{align(} \text{CATATGATGACAC, CAGAGGGAATGCT}\text{)} = \text{align(} \text{CATATGATGACAC, CAGAGGGAATGC}\text{)} + \text{C (D)}$
The trick

Optimal alignment =

Optimal alignment until the last CIGAR letter + Last CIGAR letter

align( CATATGATGACAC CAGAGGGAATGCT ) =

align( CATATGATGACA CAGAGGGAATGC ) + C T

CATAT-GA-TGACAC or CATAT-GA-TGACAC
CAGAGGGAATG-C or CAGAGGGAATG-C

score = -2
score = -2

score = -1

total score: -3

align( CATATGATGACA CAGAGGGAATGCT ) + C T

CATAT-GA-TGACAC or CATAT-GA-TGACAC
CAGAGGGAATG-C or CAGAGGGAATG-C

score = -1
score = -2

total score: -3
Finding the best alignment with mismatches+indels is possible, recursively.

But it takes effort.

There is a more direct way.
Needleman-Wunsch

- Start with a scoring scheme. Say, $M = +1$, $X = -1$, $I$ or $D = -2$.
- Write down a matrix of the two sequences to align.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*note to purists, I’m slightly simplifying presentation here, no epsilon rows*
Needleman-Wunsch

- Start with a scoring scheme. Say, $M = +1$, $X = -1$, $I$ or $D = -2$.
- Write down a matrix of the two sequences to align.
- We start with the top left, then we fill all neighboring cells.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each cell is the alignment score of [query up to this row] vs [reference up to this column]
Needleman-Wunsch

- Start with a scoring scheme. Say, $M = +1$, $X = -1$, $I$ or $D = -2$.
- Write down a matrix of the two sequences to align.
- We start with the top left, then we fill all neighboring cells

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Score: -1
Needleman-Wunsch

- Start with a scoring scheme. Say, \( M = +1, X = -1, I \) or \( D = -2 \).
- Write down a matrix of the two sequences to align.
- We start with the top left, then we fill all neighboring cells

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A-AT

MI
score: -1
Needleman-Wunsch

- Start with a scoring scheme. Say, \( M = +1, X = -1, I \ or \ D = -2 \).
- Write down a matrix of the two sequences to align.
- We start with the top left, then we fill all neighboring cells

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flash exercise!
Think hard about **what** to put here
Needleman-Wunsch

- Start with a scoring scheme. Say, M = +1, X = -1, I or D = -2.
- Write down a matrix of the two sequences to align.
- We start with the top left, then we fill all neighboring cells

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Three possibilities:
- **MX** -> score 0
- **MDI** -> score -3
- **MID** -> score -3

btw, MID is:  A-G  
AT-
**Needleman-Wunsch**

- Start with a scoring scheme. Say, $M = +1$, $X = -1$, I or D = -2.
- Write down a matrix of the two sequences to align.
- We start with the top left, then we fill all neighboring cells.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>-1</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
<td>0</td>
<td>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Insight:** each filled cell corresponds to the CIGAR string of the alignment so far.
Needleman-Wunsch

- Start with a scoring scheme. Say, $M = +1$, $X = -1$, $I$ or $D = -2$.
- Write down a matrix of the two sequences to align.
- We start with the top left, then we fill all neighboring cells

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>-1</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
<td>0</td>
<td>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Needleman-Wunsch

- Start with a scoring scheme. Say, M = +1, X = -1, I or D = -2.
- Write down a matrix of the two sequences to align.
- We start with the top left, then we fill all neighboring cells

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>-1</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-4</td>
<td>-6</td>
</tr>
<tr>
<td>C</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>C</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Needleman-Wunsch

- Start with a scoring scheme. Say, M = +1, X = -1, I or D = -2.
- Write down a matrix of the two sequences to align.
- We start with the top left, then we fill all neighboring cells

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>-1</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-4</td>
<td>-6</td>
</tr>
<tr>
<td>C</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>C</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Then the alignment is the CIGAR string at the **bottom right** cell. It traces back to the top left cell:

MDMMX

AGTCA

A → TCC
Exercice 3 (hard): fill this matrix

- Scoring function: $M = +1$, $X = -1$, $I$ or $D = -2$.
- Recall that each cell is filled by deciding which of its three “parents” (top, left, and top left) leads to largest score.

<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-1</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
<td>-7</td>
</tr>
<tr>
<td>T</td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Recall:

In general, bottom_right = $\max(\text{top_left} + M \text{ or } X, \text{bottom_left} + D, \text{top_right} + I)$

Three possibilities:
- $M\times$ -> score 0
- $MD\times$ -> score -3
- $MI\times$ -> score -3
Solution

- Scoring function. Say, M = +1, X = -1, I or D = -2.

<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-1</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
<td>-7</td>
</tr>
<tr>
<td>T</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-4</td>
<td>-6</td>
</tr>
<tr>
<td>C</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>-1</td>
<td>-3</td>
</tr>
<tr>
<td>C</td>
<td>-7</td>
<td>-6</td>
<td>-5</td>
<td>-2</td>
<td>-2</td>
</tr>
</tbody>
</table>

score: -2

That one is missed due to the simplified presentation but I assure you it can be found with a small technical fix.
Coffee break?
"Dynamic programming"?

...The 1950s were not good years for mathematical research. [the] Secretary of Defense ... had a pathological fear and hatred of the word, research...

I decided therefore to use the word, "programming".

I wanted to get across the idea that this was dynamic, this was multistage... I thought, let's ... take a word that has an absolutely precise meaning, namely dynamic... it's impossible to use the word, dynamic, in a pejorative sense. Try thinking of some combination that will possibly give it a pejorative meaning. It's impossible.

Thus, I thought dynamic programming was a good name. It was something not even a Congressman could object to."


https://twitter.com/padfoot_c4/status/1292518916120600579
Smith-Waterman

Same as Needleman-Wunsch, but make it local.

<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

1. Allow gaps at beginning
2. Find the highest scoring cell
3. Trace it back to a zero

Here: TC aligned to TC (.. how surprising)
Limits of Smith-Waterman: Equally good alignments

query: AAAGAGATAT

aligns with same score to and

reference: ...TCATAAACAGATATGA...CCAAAGAGATTGATA...

Most tools will either report a fixed number of equally good alignments, or just one arbitrarily with a warning (‘low mapping quality’). Either way, beware.
Why can’t we Smith-Waterman everything?

It requires \((n \times m)\) operations, where \(n\) and \(m\) are the sequence lengths.

When \(n \sim m\), it’s \(n^2\) operations:
Approximate alignment

Also called “heuristic”.

BLAST, minimap2, bowtie2, BWA, DIAMOND, .. everything.

Pranay Pathole @PPathole · 3/6/20
Algorithm - when programmers don't want to explain what they did.

Heuristic - when programmers can't explain what they did.

Machine Learning - when programmers don't know what they did.
Be BLAST!

Can you visually find where this sequence (locally) aligns to?

query : CAAAATGA

reference:
ACATGATGATGATGACATGATGATGATGAGTACATGGGAGTATGATGATGATATG
ATGATGATATGATGACACAAAAATGAGTGACACAGGCCACAATGATGATTAGGTTCCCTTTTTGAAAGTTGATGATGAGGGTTAACCTTATGATATAGATGATG
Be BLAST!

Can you visually find where this sequence (locally) aligns to?

query : CAAAATGA

reference:
ACATGTGATGATGACATGATGATGATGAGTACATGGGAAGTATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGATGACAAACAAAAATGAGTGACACAGGCCCAACAATGATGATTAGG
GTTCCCTTTTTTGAAAGTTGATGATGAGGGTTAAACCTTATGATATAGATGATGATG

How about now?
How BLAST works

**Seeds**: short sequences found in both the query and the reference.

1) **Find** seeds using a table
2) **Align** with SW-like method around seeds

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Found in ref at position(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAAAAA</td>
<td>10, 65, 147, ...</td>
</tr>
<tr>
<td>AAAAC</td>
<td>80</td>
</tr>
<tr>
<td>....</td>
<td></td>
</tr>
<tr>
<td>CTTAA</td>
<td>none</td>
</tr>
<tr>
<td>....</td>
<td></td>
</tr>
<tr>
<td>CCCCCC</td>
<td>49, 101</td>
</tr>
</tbody>
</table>
Some DNA scoring schemes

- **Edit Distance:**
  - Match = +1
  - Mismatch = -1
  - Indel = -1

- **BLAST (megablast):**
  - Match = +1
  - Mismatch = -2
  - Indel = -2.5

- **Minimap2:**
  - Match = +2
  - Mismatch = -4
  - Gap open = -4 (‘affine gap penalty’)
  - Gap extend = -2
WFA ("WaveFront Alignment")

Not enough time / instructor skill to teach that today.
But for now:

- Smith-Waterman, but faster for high-identity pairs
- Uses a special scoring system (M=0, gap open/extend)
- Resolves a 30 year conjecture on the speed of affine gap alignment
BLAST’s E-value

E-value = number of hits one can “expect” to see by chance on a database this size.

Always raise an eyebrow if your E-value is $\geq 0.01$.

Common thresholds: $< 0.01$, or $< 1e-5$

If we have time: search for this random seq GAGATGCTGGCCACGAGCTAAATTAAG
Pairwise DNA

Long sequences versus long sequences
Tools

- BLAT
- Exonerate
- LASTZ
- MUMmer
- minimap2
BLAT

Close but not quite BLAST.

Differences:

1) Sequence-vs-genome (BLAT), instead of sequence-vs-database (BLAST)
2) Only find hits with $\geq 95\%$ identity, over $\geq 40$ bases
3) Faster than BLAST, integrated into UCSC Genome Browser

https://genome.ucsc.edu/FAQ/FAQblat.html
Dotplots

Tools: LASTZ, D-Genies, yass, MUMmer
Reciproqual best hits

A strange technique for e.g. finding orthologs.

If:

1) top alignment of gene A in species X is gene B in species Y and

2) top alignment of gene B in species Y is gene A in species X then genes A and B are RBH.
ANI (average nucleotide identity)

A strange identity metric, used to compare two bacterial genomes:

1. Extract many 1 Kbp fragments from query
2. ANI = mean identity of the reciprocal best hits

(from FastANI: https://www.nature.com/articles/s41467-018-07641-9)

Fast method: skani

https://twitter.com/jim_elevator/status/161683599031611394
Minimap2 parameters to keep an eye on

- `a` (SAM) or `c` (PAF) to really align,

- `x[mode]` controls mapping modes:
  - `map-pb/map-ont` - PacBio CLR/Nanopore vs reference mapping
  - `map-hifi` - PacBio HiFi reads vs reference mapping
  - `ava-pb/ava-ont` - PacBio/Nanopore read overlap
  - `asm5/asm10/asm20` - asm-to-ref mapping, for ~0.1/1/5% seq div
  - `splice/splice:hq` - long-read/Pacbio-CCS spliced alignment
  - `sr` - genomic short-read mapping
Pairwise DNA

Short sequences versus short sequences

Nobody really does that any more
Genome Assembly has better techniques (e.g. de Bruijn graphs)
Pointers

minimap (then miniasm)

StarCode https://academic.oup.com/bioinformatics/article/31/12/1913/213875

SlideSort https://github.com/iskana/SlideSort

PAF file format
Pairwise DNA

Long sequences versus short sequences
a.k.a read mapping
Short read mapping, in principle

AACACTGTCTGCCTCAGGAGTTAAATCTTTACA−GGATGA reference
AACACTGTCTGCTT read1
TCTG−TTCAGGAGTT read2
CTGCTTCAGGAGTT read3
GGGAGTTAAATCTTT read4
GAGTTAAAT read5

Adapted from https://wikis.univ-lille.fr/bilille/_media/ngs2023_dnaseq_03_cours_dna_mapping.pdf
Wait.. is this **local** alignment or **global** alignment?

Neither. It’s **glocal**.
Why is it difficult? Need to find a home for every read

Problem: Half of the human genome is comprised of repeats

(first bit of human chromosome 1)
Output format

SAM, BAM formats

Will be discussed in the file formats session
Tools

- Bowtie2
- BWA-MEM
- Strobealign
- minimap2

Which one to choose? *It does not matter much.* They all have their perks:

**Bowtie2, BWA-MEM**: battle-tested, well-documented

**minimap2**: faster, but cannot map \( \leq 100 \) bp reads

**Strobealign**: ultra fast, newer
How to search for a short sequence (say, \texttt{mi}) inside a longer reference (say, \texttt{evomics})? Having all the suffixes of the reference, in sorted order, would help:

\begin{itemize}
  \item cs
  \item evomics
  \item ics
  \item mics <- can be found in 1 step by binary search
  \item omics
  \item vomics
\end{itemize}

(Here it is also easy to scan the whole reference but imagine if it was a million letters long)
The Burrows-Wheeler transform considers **sorted rotations:**

- csevomi
- evomics
- icsevom
- micsevo
- omicsev
- vomicse

And remembers only the last column.
The trick is that all prefixes can be reconstructed from only the last column.

```
.....i   i.....  c.....  c.....i  ic.....
.....s  rotate  s.....  sort  e.....  write lastcol  e.....s  rotate  se.....  keep going
.....m  ->  m.....  ->  i.....  ->  i.....m  ->  mi.....  ->  ...
.....o   o.....  m.....  m.....o  om.....
.....v   v.....  o.....  o.....v  vo.....
.....e   e.....  v.....  v.....e  ev.....
```

And when searching for a short read, only short prefixes need to be “reconstructed” this way.
**FM-Index and Burrows-Wheeler, a 20,000-feet view**

**Suffix tree**: all suffixes inside a tree, older technique

**Burrows-Wheeler transform**: last column of sorted rotations of reference

**FM-index**: set of tricks to quickly search inside the Burrows-Wheeler transform without reconstructing prefixes
How does Bowtie2 work?

Specializes in aligning Illumina reads to genomes.

1) Find seeds using FM-index, typically 20 nt length, up to 1 mismatch
2) Prioritizes seeds to further align
3) Extend seeds using SW-like algorithm

(that’s it)
Minimizers

Minimap2 and strobealign use minimizers as seeds, then SW extension.

**Minimizers**: slide a window over the reference, and pick the (lexicographically) smallest seed within that window. Do that for all windows.

<table>
<thead>
<tr>
<th>Seed</th>
<th>Found at position(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAAAAA</td>
<td>10, 65, 147, ...</td>
</tr>
<tr>
<td>AAAAC</td>
<td>80</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>AAAGG</td>
<td>none</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>TAAAAA</td>
<td>49, 101</td>
</tr>
</tbody>
</table>

reference: CTAAAAAGGTCA..
2nd window: TAAAAAAGG
seed: AAAAA
       AAAAG
       AAAGG
Chains

Useful component of minimap2 (taken from whole-genome alignment methods).

-> Before aligning, look for long enough co-linear chains of close seeds.
Paired reads

In some cases, Illumina sequencers output pairs of reads.

Just pay attention to:

- Orientation (forward-reverse is most common)
- Format: interleaved in one file, or two separate files
Mapping quality

...is your best friend, to avoid errors downstreams.

**Mapq:** how confidently each read is mapped (in log probability).

Grab only highly-confident alignments: `samtools view -q 60 [file.bam]`

Grab all alignments except trash ones: `samtools view -q 1 [file.bam]`

"I'm feeling lucky": `samtools view [file.bam]`
“Mapping” vs “Alignment”

In my view:

- **Mapping**: output where each read maps. That’s it.
- **Alignment**: do that, but also output how all bases line up (CIGAR).

“minimap2” vs “minimap2 -c” (or -a)
Visualization of alignments

Reference and BAM need to be indexed, use samtools
RNA read alignment is very similar to DNA, except:

- Split mapping (on genomes) due to splicing
- Ambiguity (on transcriptomes) due to many isoforms

Tools:

- Kallisto, Salmon
- STAR, HiSAT2
Long read mapping

Similar in spirit to short read mapping, but different tools.

PacBio CLR / ONT:

- Minimap2
- Variants of minimap2 for ~ 2-5x speed gain (mm2-fast, BLEND, ..)

PacBio HiFi:

- Minimap2
- Winnowmap2 (better accuracy)
- Mapquik (30x faster mapping, but no alignment)
Pairwise protein
What changes compared to pairwise DNA?

- Different alphabet, shorter sequences
- Some AA substitutions are more likely than others
  - BLOSUM

Applications:

- Low-homology search (high evolutionary distances)
Some words of caution

"Alignment scoring schemes are hilariously over-simplified model of real evolution [...] treat all alignments with large pinch of salt [...] dynamic programming is ‘exact’ only to an ivory-tower computer scientist”

- Robert Edgar (computer scientist)

There is no such thing as “the alignment” between two protein sequences.
Tools

MMseqs2
DIAMOND2
BLASTp
How mmseqs2 work: mmseqs search

Seed finding

Seed chaining

https://github.com/soedinglab/mmseqs2/wiki#description-of-workflows
How DIAMOND work:

“[..] A simple exact match criterion determines which seeds are passed on to the extension phase, in which a Smith-Waterman alignment is computed.”

https://www.nature.com/articles/nmeth3176
Multiple, protein
What it looks like

Input: $n$ sequences

ACATGA
ACGTG
CATTA

Output: aligned sequences, with indels

ACATGA
ACGTG-
-CATTA
In practice..

Colors = equivalent residues (structurally or functionally)
Why do multiple alignment?

- Comparative genomics
- Phylogeny
- Protein structure prediction
- RNA structure and function
- ...
How is a MSA scored?

“Sum-of-pairs” (SP) score:

1) Fix a scoring scheme, e.g. match=1, mismatch=-1, indel=-2.
2) For each column, for all pairs of residues, compute score
3) Sum scores across columns

Column: 123456

ACATGA
ACG-G-
-CAGTA

For column 4: score(T,-) + score(T,G) + score(-,G) = -2 + -1 + -2 = -5.
For column 5: score(G,G) + score(G,T) + score(G,T) = 1 + -1 + -1 = -1.
Optimal MSA

Remember Needleman-Wunsch?

Same, but with more possibilities.

So, best avoided.
Progressive MSA

Progressive alignment

At each node, align columns in left- and right-child MSAs to build a combined MSA.

https://www.youtube.com/watch?v=2HmjHStpu7I
MSA is on another level of difficulty

Challenging alignment

Alternative MSAs of same sequences

Which one is correct / better?

Hard / impossible to decide, even with structures

https://www.youtube.com/watch?v=2HmjHStpu7I
Tools

- MUSCLE
- ClustalW
- T-Coffee
- MAFFT
- ...

Multiple, DNA
What changes compared to protein MSA?

- Wayyy longer sequences
- Duplications, inversions, and translocations wreak linearity
Tools

- SibeliaZ
- Cactus

State of the art: human genome graphs, look for pangenomics papers.

e.g. HPRC: [https://www.nature.com/articles/s41586-023-05896-x](https://www.nature.com/articles/s41586-023-05896-x), CPC [https://www.nature.com/articles/s41586-023-06173-7](https://www.nature.com/articles/s41586-023-06173-7)
1 nucl sequence versus a database
Tools

BLASTn
MetaGraph, Pebblescout
Kraken

See the Big Data lecture!
“The nucleotide collection consists of GenBank+EMBL+DDBJ+PDB+RefSeq sequences, but excludes EST, STS, GSS, WGS, TSA”

[..] “The database is non-redundant.”

125 GB compressed

Limits of BLAST

- Can’t search all known genomes, only those in the BLAST database
- Under 85% identity, alignments tend to be missed
1 sequence versus a profile

PSSMs, HMMs

Is there enough time to present this?!
Position Specific Scoring Matrices (PSSM) and Hidden Markov Models (HMM)

Not quite alignment, but:

“Does this sequence belong to a particular family?”
PSSM

Way to represent families of sequences, with no gaps.

1) Construct MSA
2) Determine frequency per column
HMM

Hidden Markov Models generalize PSSMs with gaps.

Motivation: when pairwise fails

HMM of a PSSM:

Profile HMM:

HBA_HUMAN ... VGA--HAGEY...
HBB_HUMAN ... V-----NVDEV...
MYG_PHYCA ... VEA--DVAGH...
GLB3_CHITP ... VKG-------D...
GLB5_PETMA ... VYS--TYETS...
LGB2_LUPLU ... FNA--NIPKH...
GLB1_GLYDI ... IAGADNGAGV...

*** *****

http://www.mcb111.org/w06/durbin_book.pdf
Tools

HMMer

MMseqs profile

HHblits
Input: 2 PDB structures

Output: aligned residues, and a TM-score
(> 0.5 = same fold)
Personal take

As databases of genomes grow, alignment will both become easier and harder.

Solved:

● Human read alignment (DNA, RNA)
● High-identity to current genome databases
● Small-data HMMs

Unsolved:

● Genome-scale MSA
● Ancient DNA
● Large MSAs
● Big-data HMMs
● Sequences to peta-scale databases
What we’ve seen

- **Pairwise DNA alignment**
  - CIGAR strings
  - Scoring
  - Needleman-Wunsch
  - Smith-Waterman
  - BLAST
  - BLAT, minimap2

- **Short read mapping**
  - Burrows-Wheeler transform
  - Minimizers
  - Bowtie2, BWA, minimap2, Strobealign

- **Pairwise protein alignment**
  - Diamond, mmseqs2

- **MSA**

- **HMMs**
Thank you for your attention!