CS 364 COMPUTATIONAL BIOLOGY

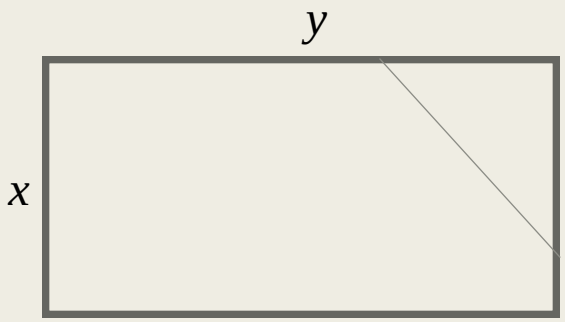
Sara Mathieson
Haverford College

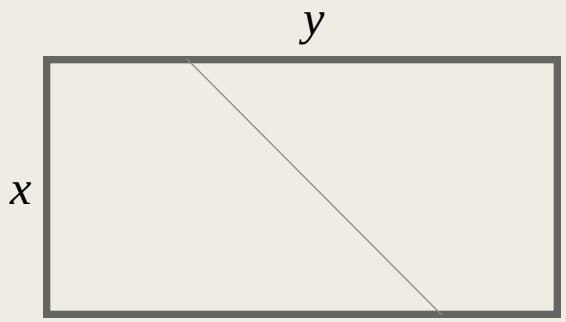
Outline

- Pairwise alignment variations
- Multiple sequence alignment
- Introduction to phylogenetics

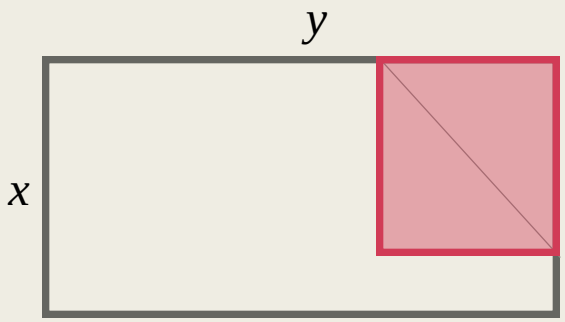
Local alignment variations

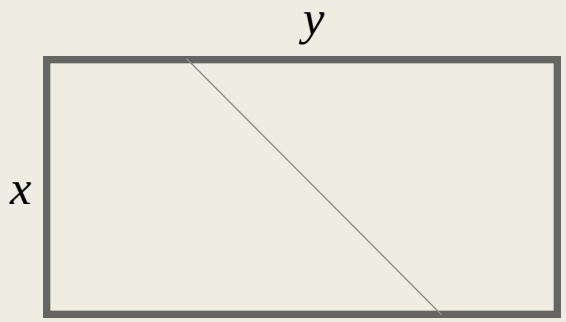
Handout 7: what portion of x aligns to what portion of y?



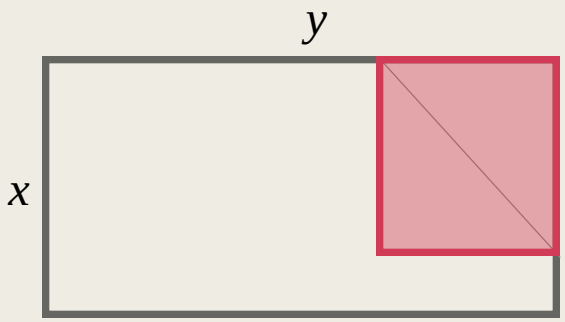


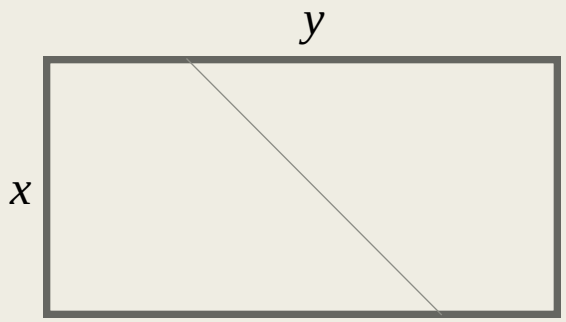
Handout 7: what portion of x aligns to what portion of y?



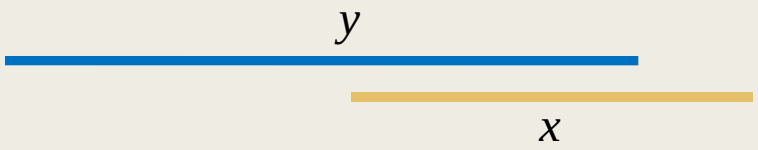


Handout 7: what portion of x aligns to what portion of y?

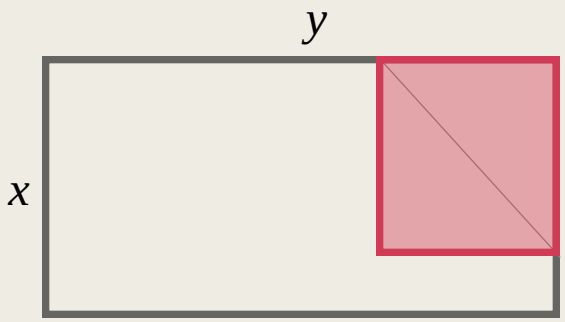


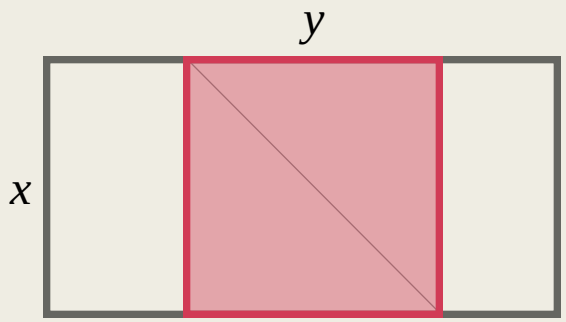


Beginning of *x* with end of *y*

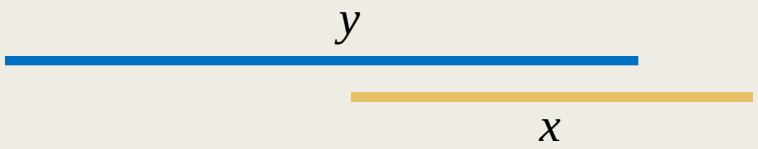


Handout 7: what portion of x aligns to what portion of y?

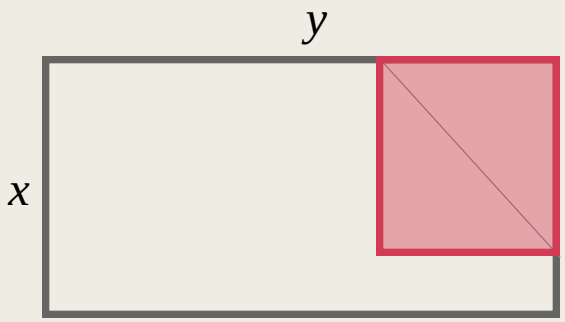




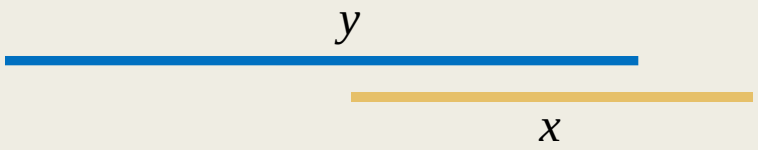
Beginning of *x* with end of *y*

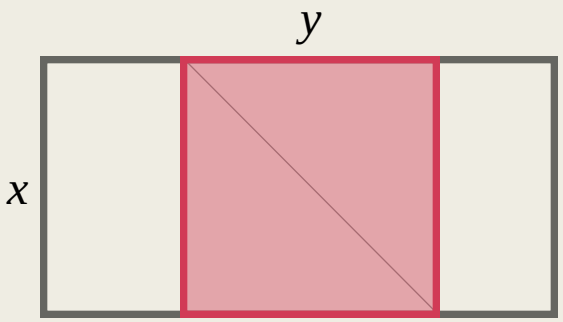


Handout 7: what portion of x aligns to what portion of y?

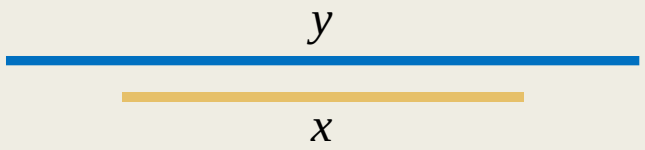


Beginning of *x* with end of *y*

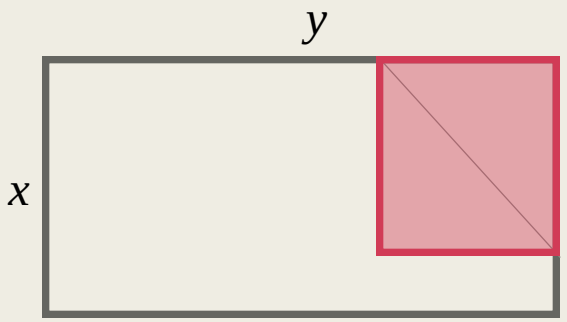




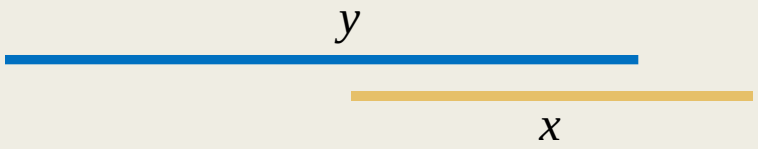
x aligns with the middle of *y*

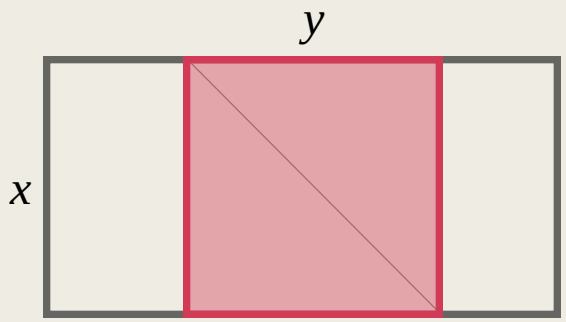


Handout 7: what portion of x aligns to what portion of y?

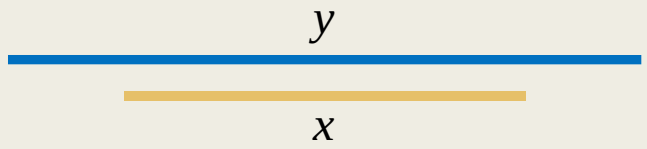


Beginning of *x* with end of *y*



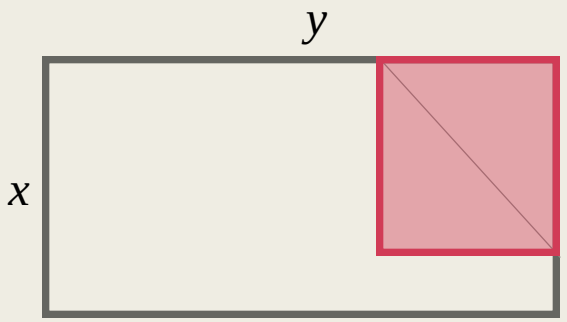


x aligns with the middle of *y*

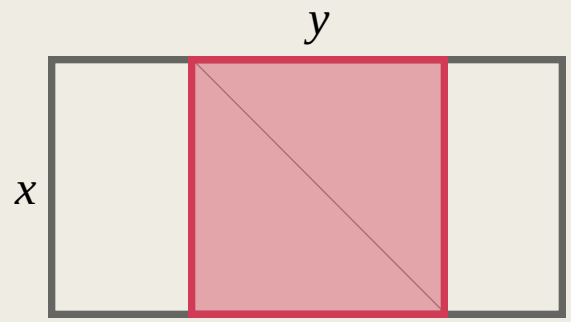


DP algorithm modifications:

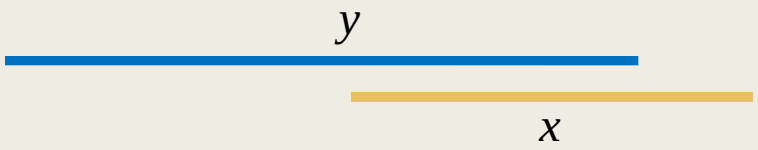
Handout 7: what portion of x aligns to what portion of y?

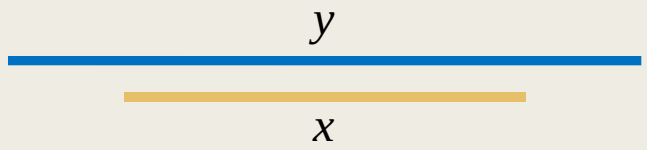


Beginning of *x* with end of *y*



x aligns with the middle of *y*

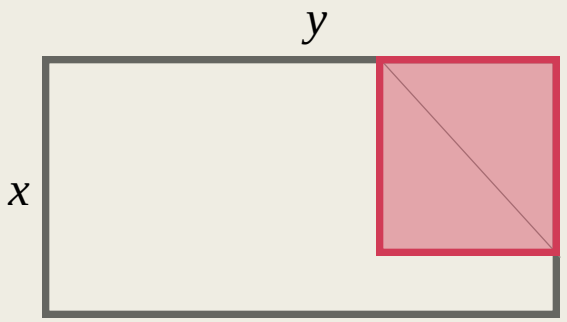




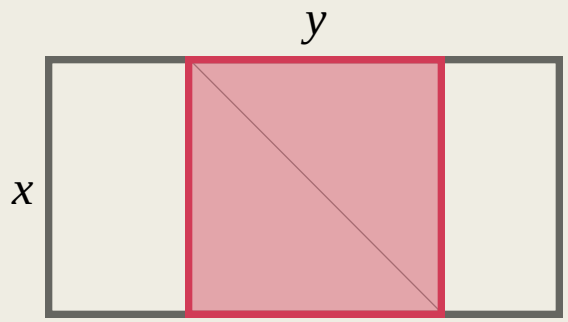
DP algorithm modifications:

1) Initialization: *0^{th} row and 0^{th} column with 0's to not penalize leading/trailing gaps*

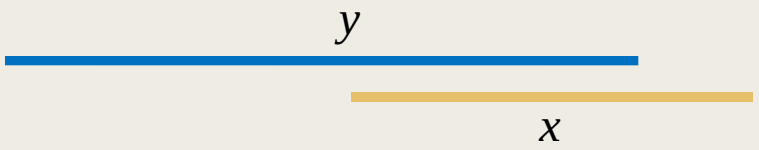
Handout 7: what portion of x aligns to what portion of y?

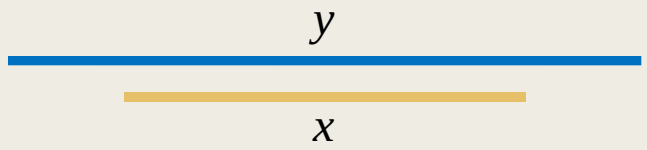


Beginning of *x* with end of *y*



x aligns with the middle of *y*

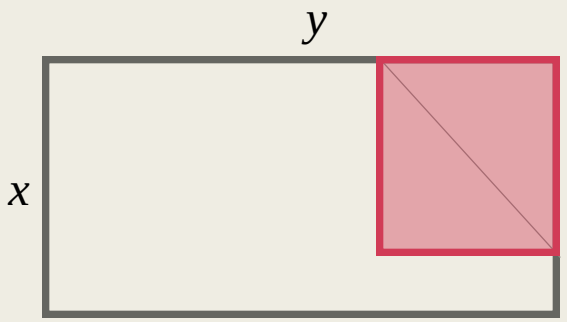




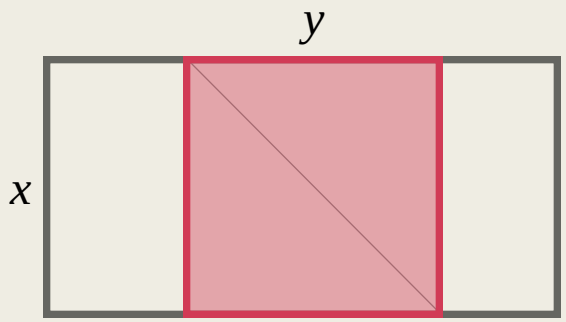
DP algorithm modifications:

1) Initialization: *0^{th} row and 0^{th} column with 0's to not penalize leading/trailing gaps*
2) Recursion: *to fill in the rest of the table, use global alignment (i.e. don't restart at 0)*

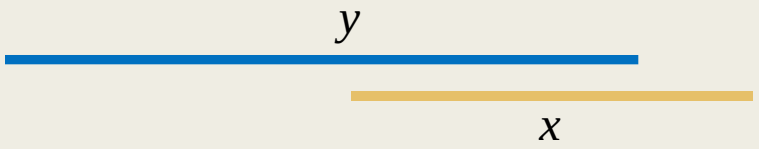
Handout 7: what portion of x aligns to what portion of y?

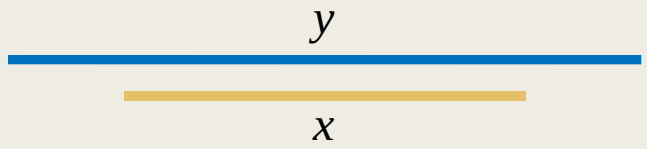


Beginning of *x* with end of *y*



x aligns with the middle of *y*





DP algorithm modifications:

1) Initialization: *0^{th} row and 0^{th} column with 0's to not penalize leading/trailing gaps*
2) Recursion: *to fill in the rest of the table, use global alignment (i.e. don't restart at 0)*
3) Traceback: *start at the maximum value along the last row or last column*

Approximate local alignment: BLAST

I sequenced something – what is it?

Want to compare with all the sequences in published databases

Smith-Waterman far too slow ($O(n^2)$ time and space)

So in practice , often use heuristic (approximate) methods

BLAST (Basic Local Alignment Search Tool)
https://blast.ncbi.nlm.nih.gov/Blast.cgi

"Basic local alignment search tool." Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. *J. Mol. Biol.* **215**, 403–410 (1990). 12th most cited scientific paper of all time (as of 2014).

Approximate local alignment: BLAST

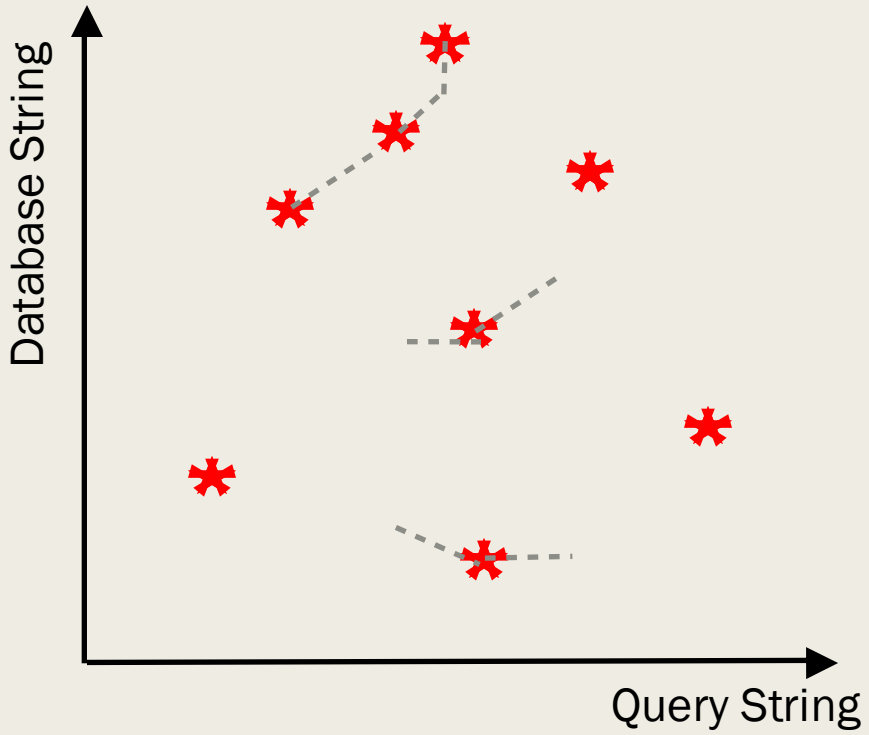
1 Find potential matches (“seeds”)

Scoring matrix aware k-mer search

Query string: ...RP**PEG**IAAXXPAPX...

Database string: ...G**PIG**ISID**PEG**PX...

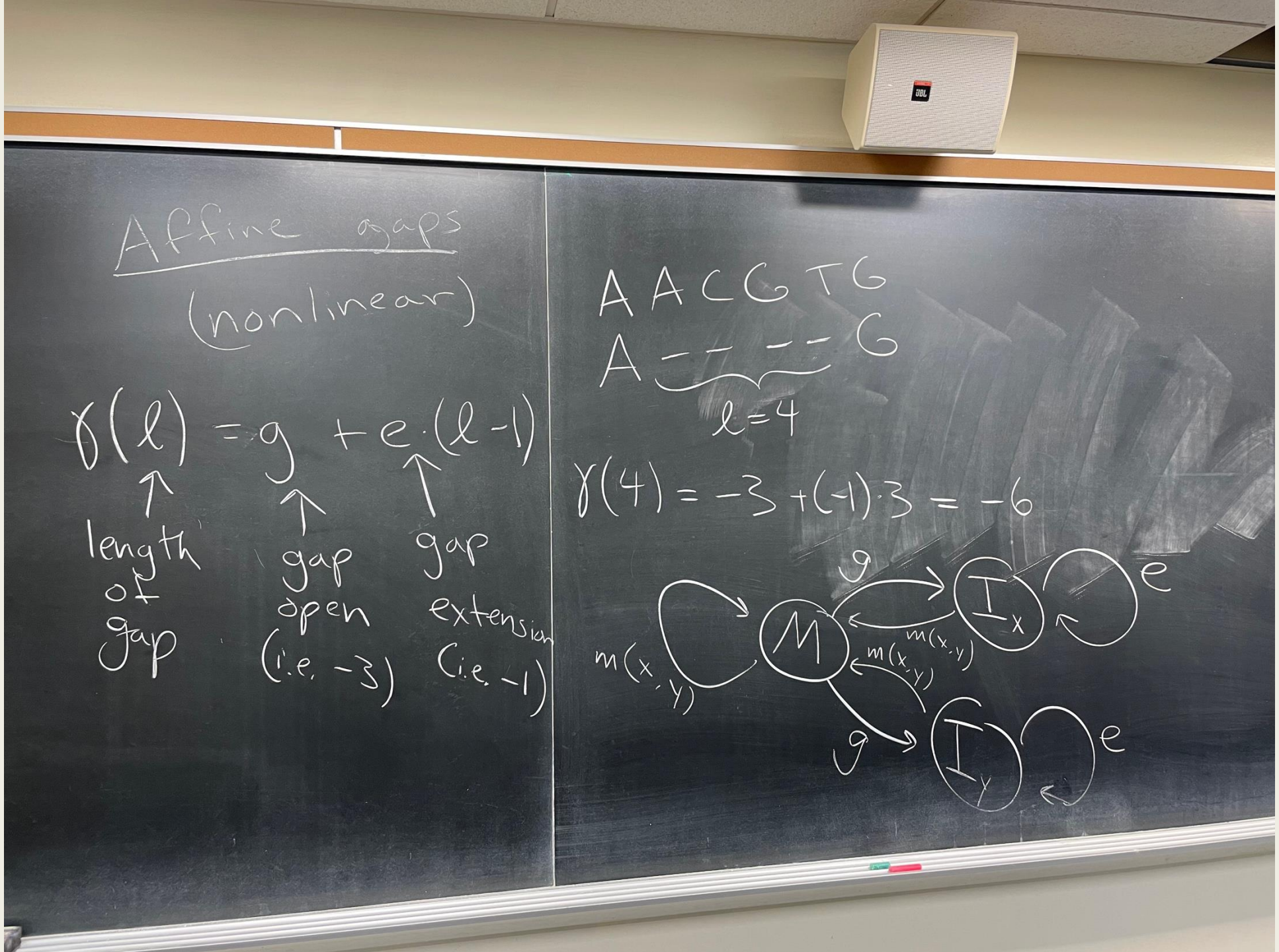
2 Extend matches around seed and join up high-scoring pairs



Not guaranteed to find the optimal alignment
But in practice it is very good!

Report the “E-value”: Expected number of matches we would see by chance

Affine (non-linear) gaps


Affine gaps
(nonlinear)
$\gamma(\ell) = g + e \cdot (\ell - 1)$
length of gap
gap open (i.e. −3)
gap extension (i.e. −1)
AACGTG
A---G
$\ell = 4$
$\gamma(4) = -3 + (-1) \cdot 3 = -6$
M
I_x
I_y
g
e
m(x,y)

Multiple Sequence Alignment

From pairwise to multiple alignment

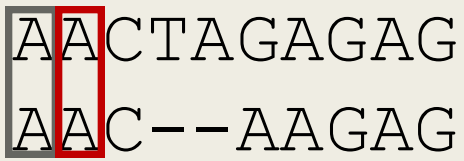
```
AACTAGAGAG
AAC--AAGAG
```

K=2 sequences

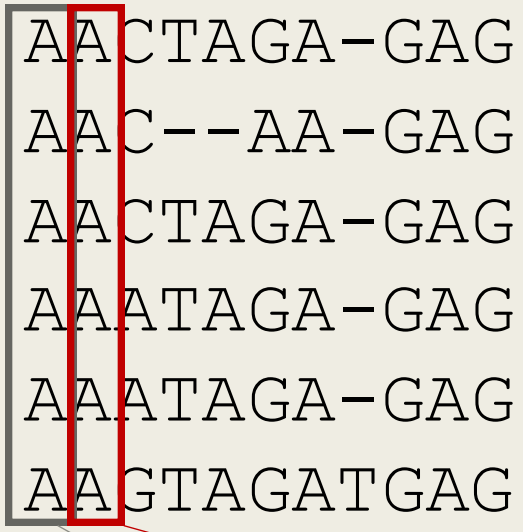
```
AACTAGA-GAG
AAC--AA-GAG
AACTAGA-GAG
AAATAGA-GAG
AAATAGA-GAG
AAGTAGATGAG
```

K=6 sequences

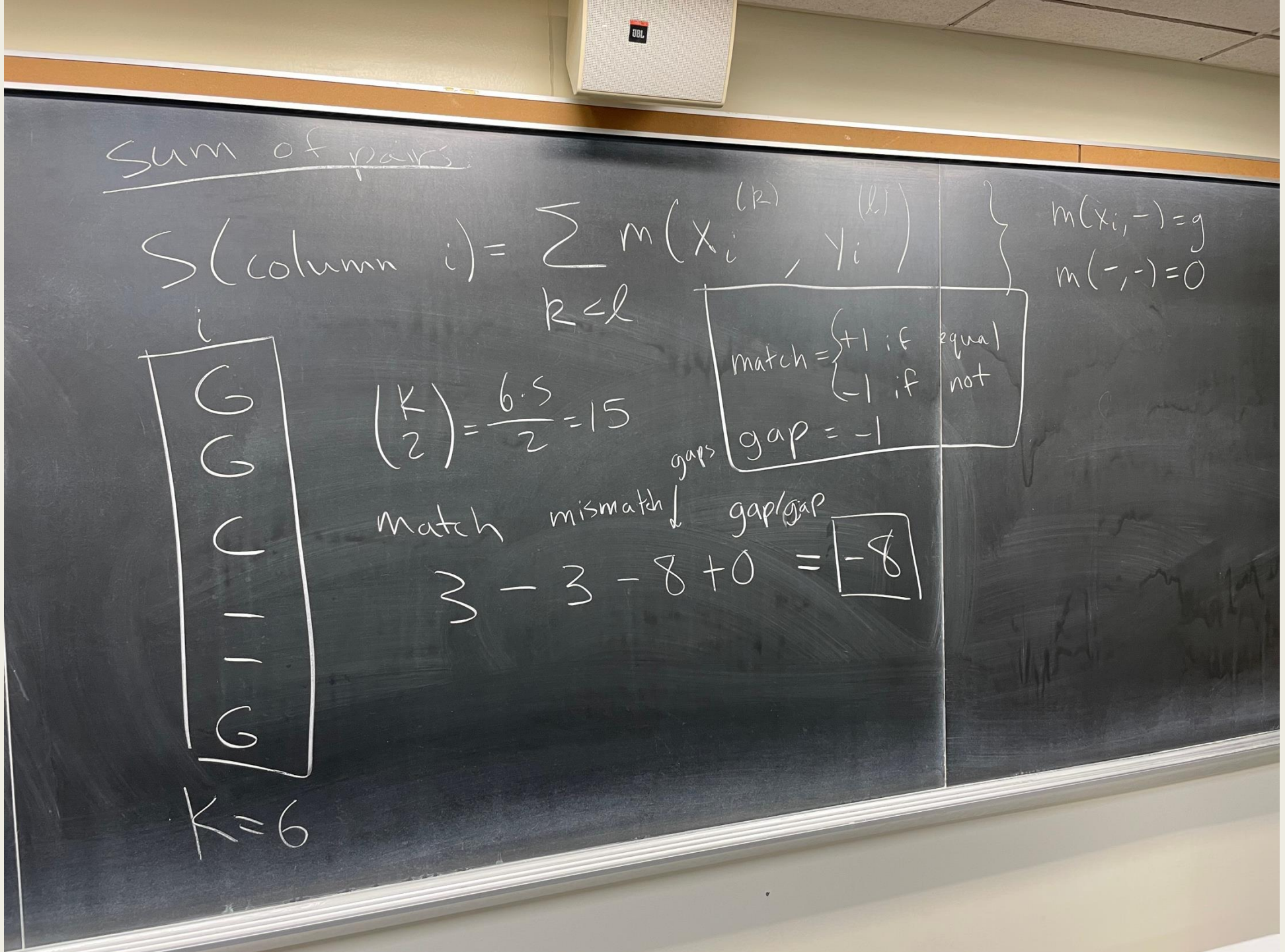
How to score multiple alignments



Score = m(A,A)+m(A,A)+….

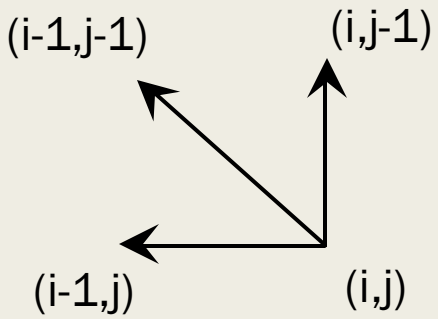


Score = S(column 1)+S(column 2)+….


Sum of pairs
$S(\text{column } i) = \sum_{k<l} m(x_i^{(k)}, y_i^{(l)})$
$m(x_i, -) = g$
$m(-, -) = 0$
match = +1 if equal
−1 if not
gap = −1
i
G
G
C
−
−
G
K=6
$\binom{K}{2} = \frac{6 \cdot 5}{2} = 15$
match mismatch gaps gap/gap
$3 - 3 - 8 + 0 = -8$

K-dimensional dynamic programming

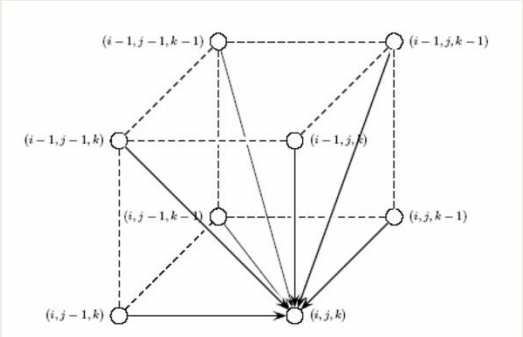
K=2



sequences a=$a_1a_2a_3$... b=$b_1b_2b_3$...

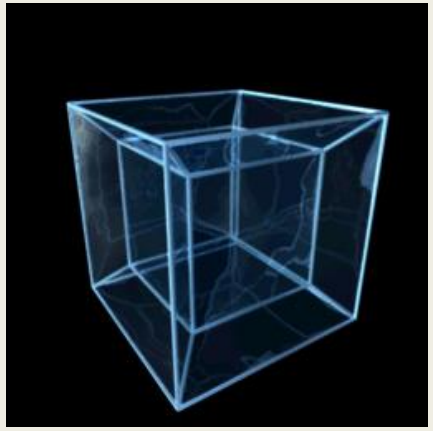
$$S(i,j) = max \begin{cases} S(i-1,j-1) + S(a_i, b_j) \\ S(i-1,j) + S(a_i, -) \\ S(i,j-1) + S(-, b_j) \end{cases}$$

K=3

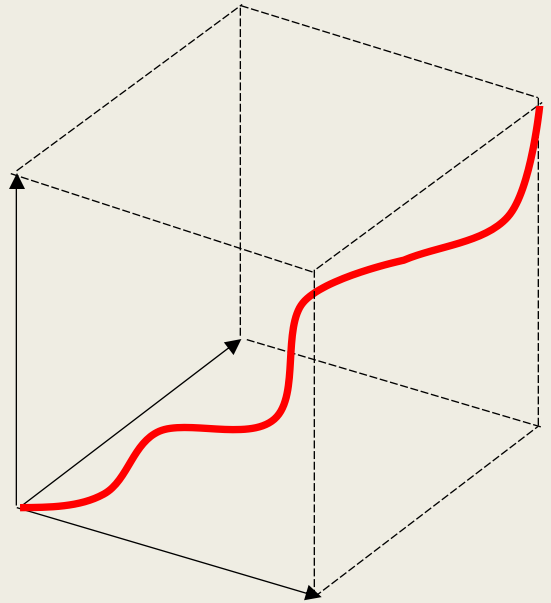


$$S(i,j,k) = max \begin{cases} S(i-1,j-1,k-1) + S(a_i, b_j, c_k) \\ S(i,j-1,k-1) + S(-, b_j, c_k) \\ S(i-1,j,k-1) + S(a_i, -, c_k) \\ S(i-1,j-1,k) + S(a_i, b_j, -) \\ S(i,j,k-1) + S(-, -, c_k) \\ S(i,j-1,k) + S(-, b_j, -) \\ S(i-1,j,k) + S(a_i, -, -) \end{cases}$$

K-dimensional dynamic programming



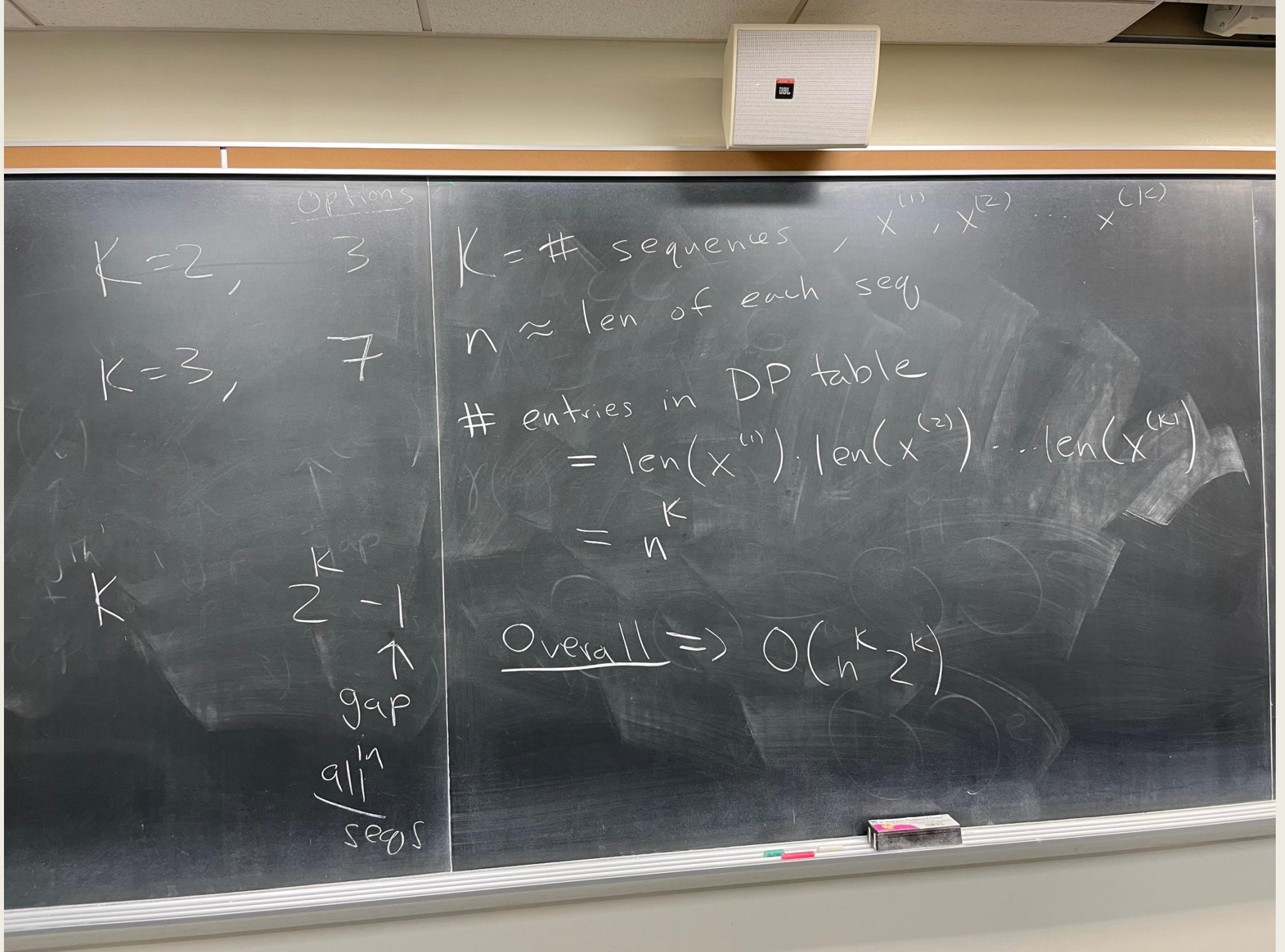
$$S(i,j,k,\dots) = max \begin{cases} S(i-1,j-1,k-1,\dots) + S(a_i,b_j,c_k,\dots) \\ S(i,j-1,k-1,\dots) + S(-,b_j,c_k,\dots) \\ \vdots \\ S(i,j,k,\dots) + S(-,-,-,\dots) \\ S(i,j,k,\dots) + S(-,-,-,\dots) \end{cases}$$



2^K-1 choices for each cell

Number of cells = $\prod_i$ length(sequence$_i$)
= n^K if all length n

So algorithm is $O(2^K n^K)$


Options
K=2, 3
K=3, 7
K
$2^K - 1$
gap
gaps
K = # sequences, $x^{(1)}, x^{(2)} \cdots x^{(K)}$
n ≈ len of each seq
entries in DP table
$= \text{len}(x^{(1)}) \cdot \text{len}(x^{(2)}) \cdots \text{len}(x^{(K)})$
$= n^K$
Overall $\Rightarrow O(n^K 2^K)$

K-dimensional dynamic programming

Assume you have some sequences that are n=50 characters long, and that pairwise alignment of K=2 such sequences takes one second on your computer.

Then alignment of K=4 sequences takes $0.0001*(2n)^K = 10^4$s ~3 hours

Suppose you had unlimited memory and you were willing to wait until the sun burns out and turns into a red dwarf in 5 billion years, how many sequences could you align?

Progressive alignment

Iterative alignment

Idea: align sequences to each other in some order, so we only ever have to solve pairwise problems

```
X1: AACTAGA-GAG
X2: ATC--AA-GAG
```

```
X1: AACTAGA-GAG
X2: ATC--AA-GAG
X3: AACTAGA-GAG
```

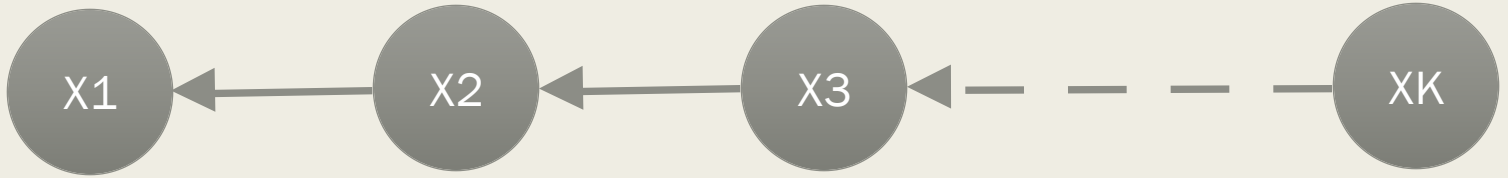
…

```
AACTAGA-GAG
ATC--AA-GAG
AACTAGA-GAG
AAATAGA-GAG
AAATAGA-GAG
AAGTAGATGAG
```

1) Align X2 to X1

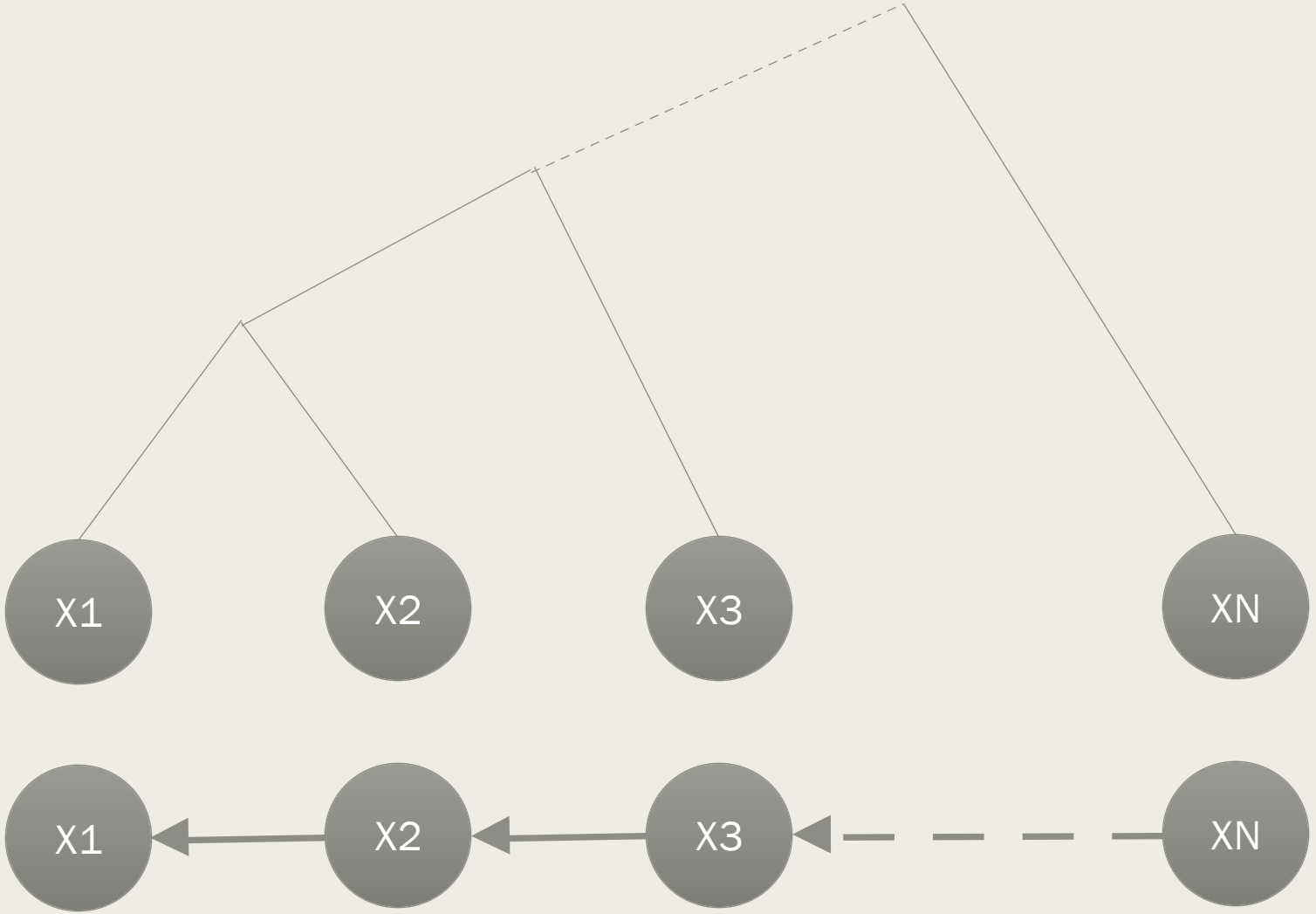
2) Align X3 to the alignment of X2 and X3*

K) etc…



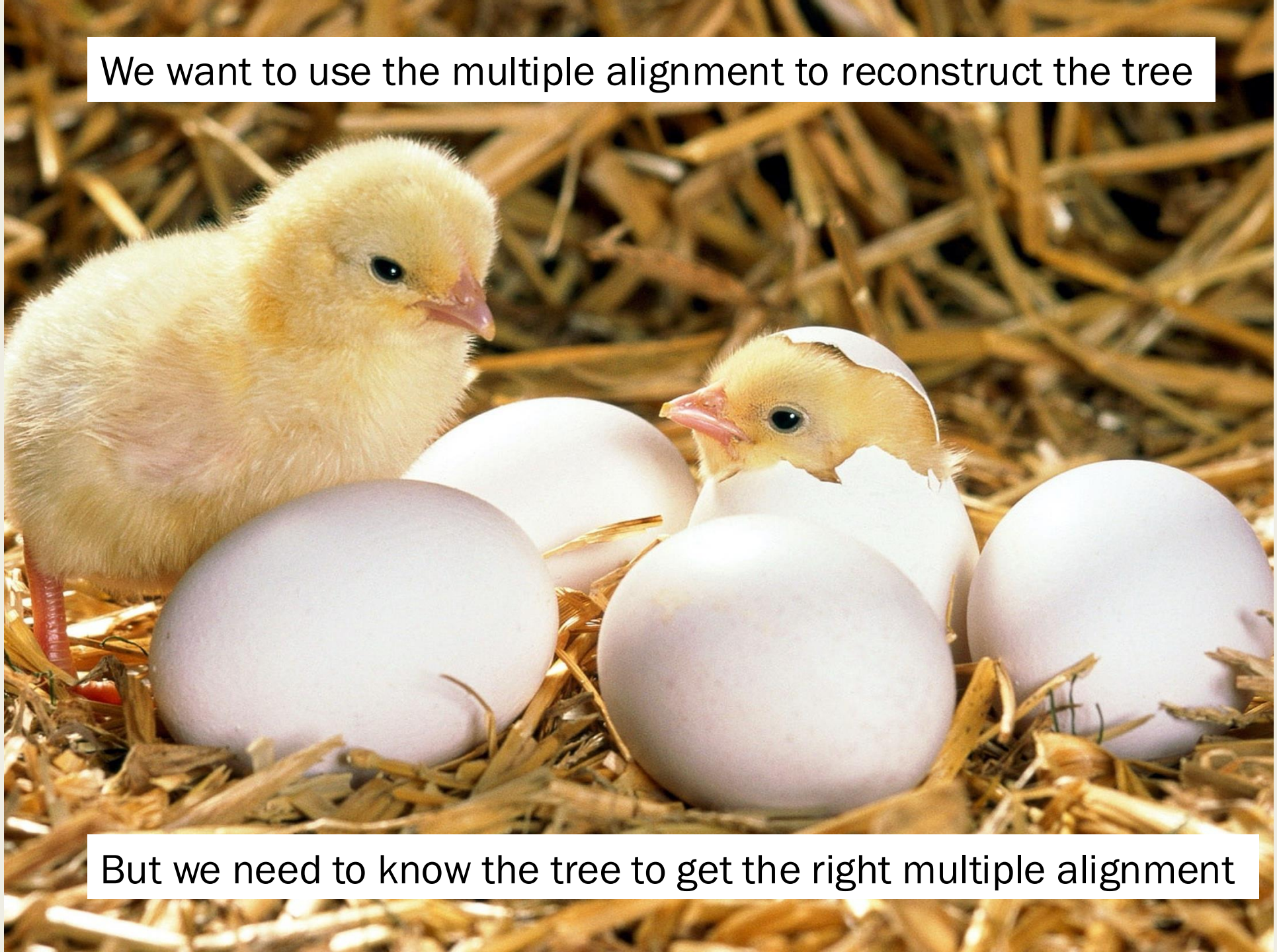
*How to align a sequence to an alignment?

Progressive alignment



Key idea: Align most similar sequences first

We want to use the multiple alignment to reconstruct the tree



But we need to know the tree to get the right multiple alignment

Clustal-W: multiple sequence alignment algorithm

- 75,000 citations (2024)

JOURNAL ARTICLE

CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice

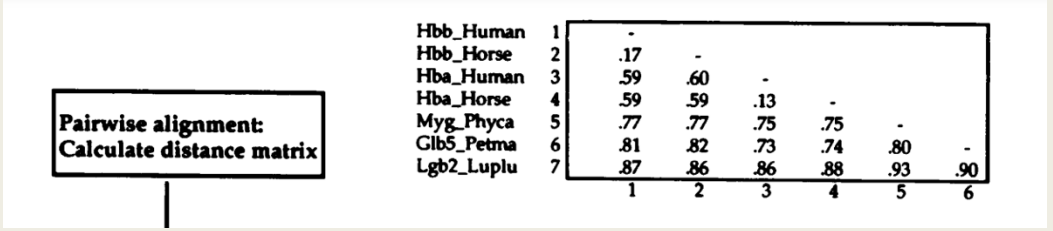
Julie D. Thompson, Desmond G. Higgins, Toby J. Gibson ✉ Author Notes

Nucleic Acids Research, Volume 22, Issue 22, 11 November 1994, Pages 4673–4680,
https://doi.org/10.1093/nar/22.22.4673
Published: 11 November 1994 **Article history** ▾

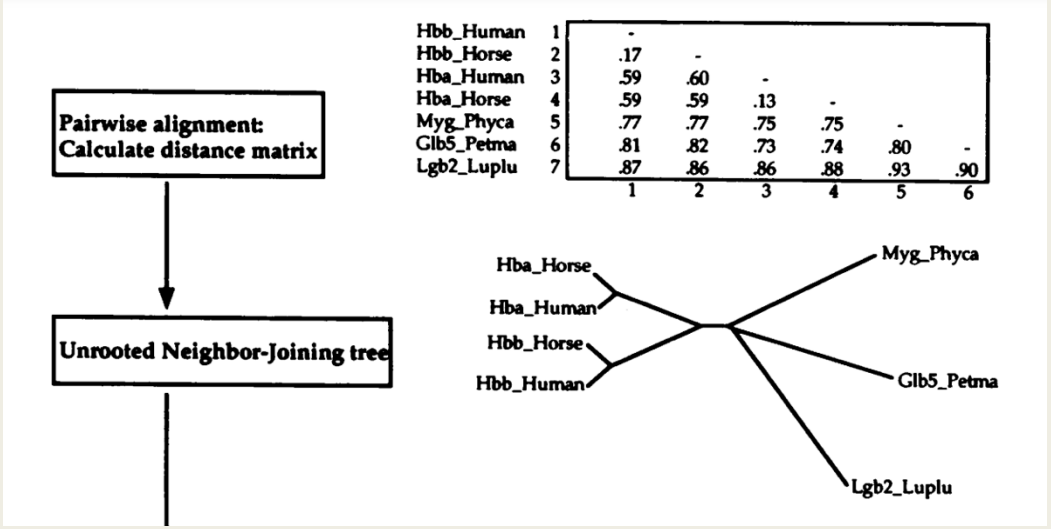
Clustal-W: multiple sequence alignment algorithm

- Compute all possible pairwise alignments, use to create a pairwise distance matrix between the strings
- Use the pairwise distance matrix to create a UPGMA or NJ tree graph (next week!)
- Use the tree graph to iteratively merge pairwise alignments of nodes in the tree (which may be leaves or ancestral nodes). Start with most similar nodes first

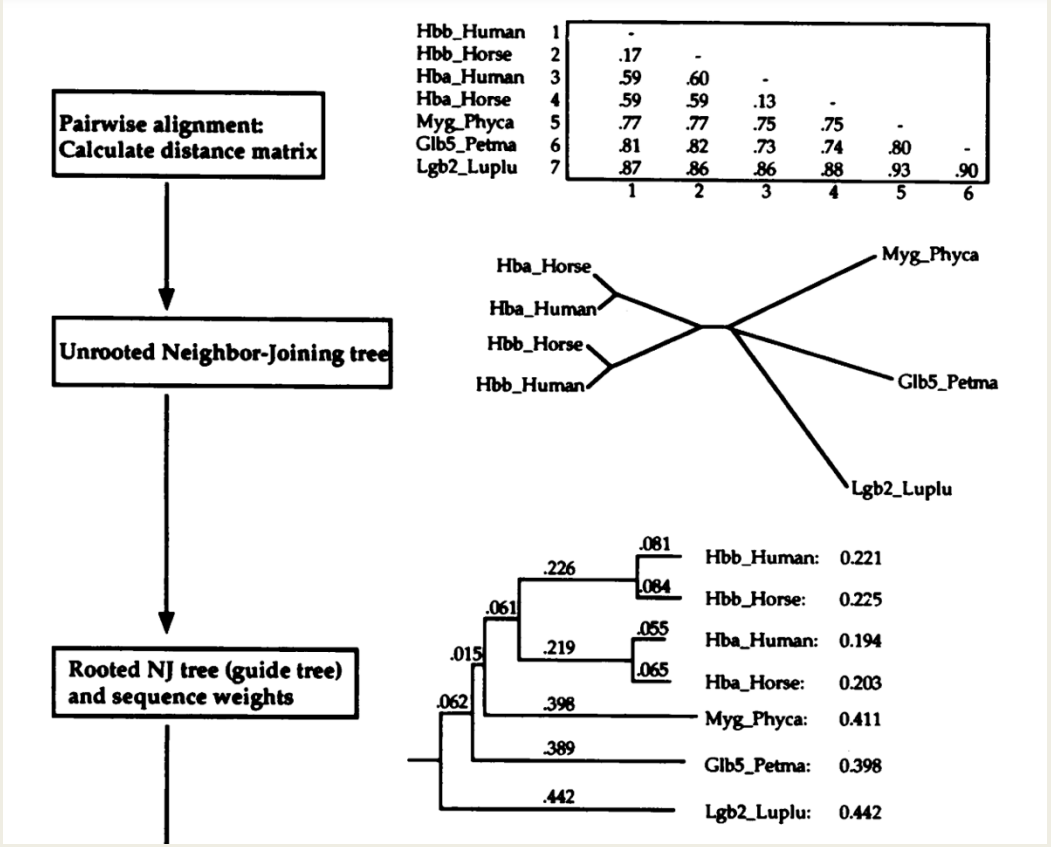
ClustaI-W workflow



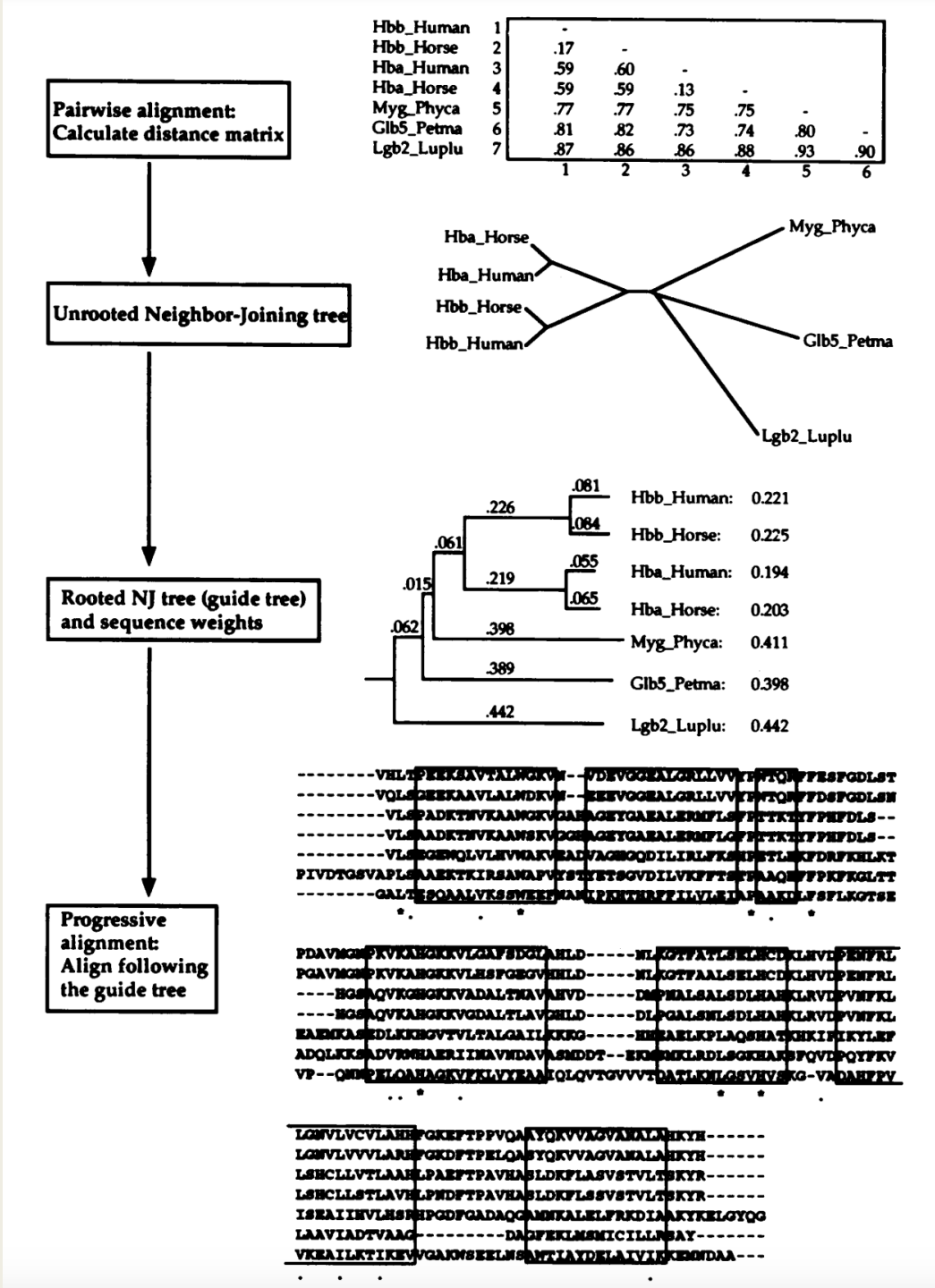
ClustaI-W workflow

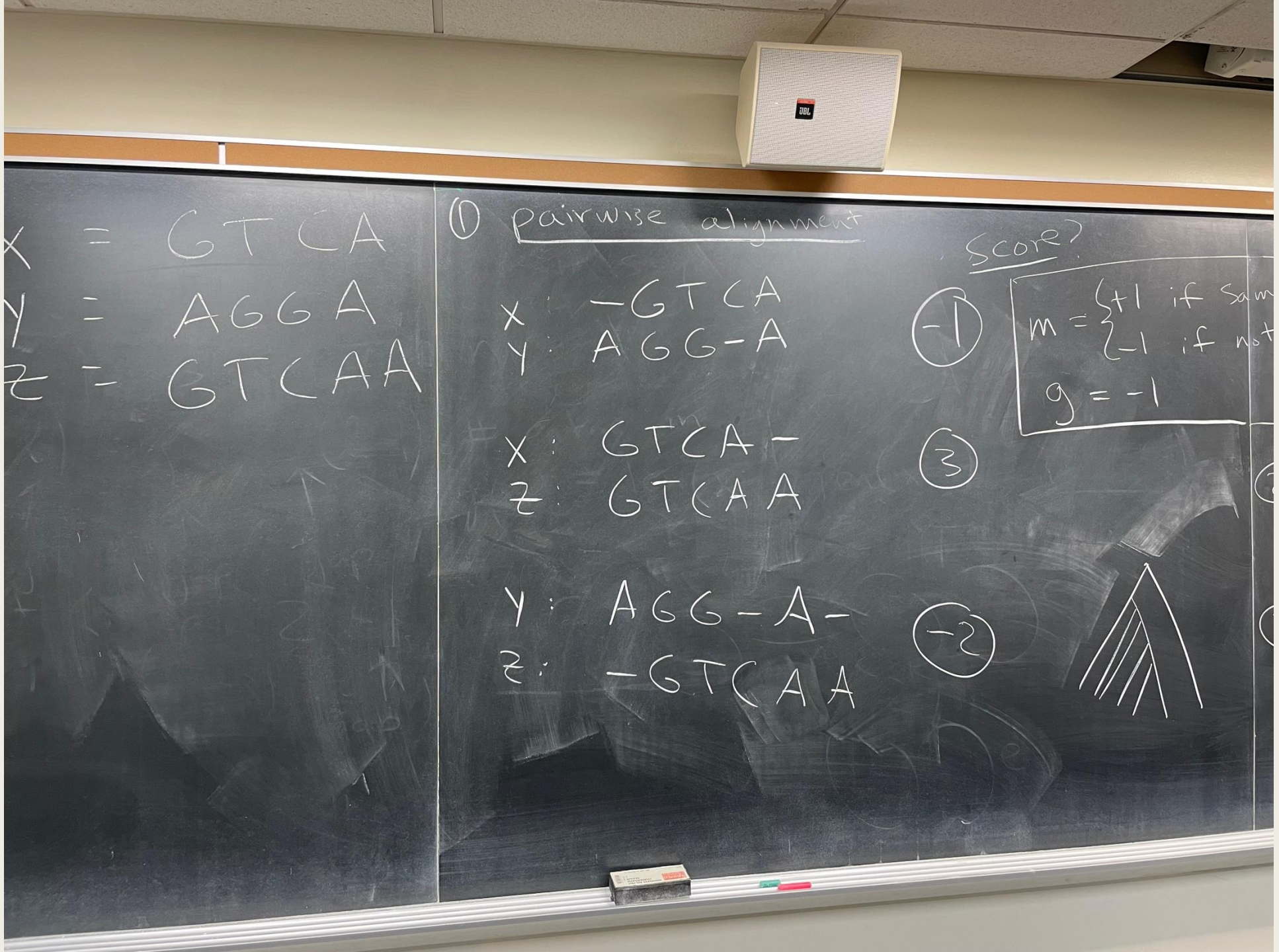


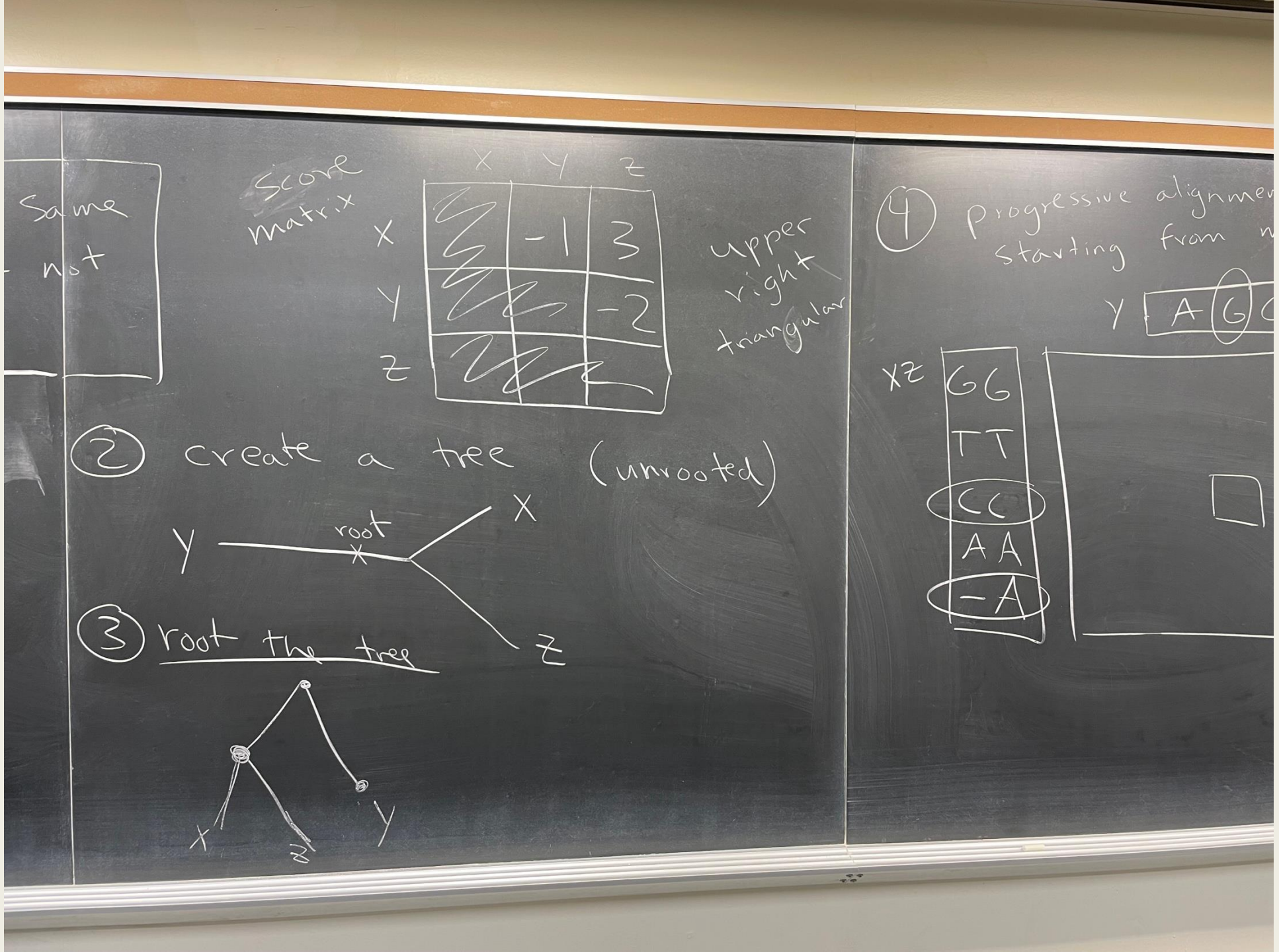
ClustaI-W workflow

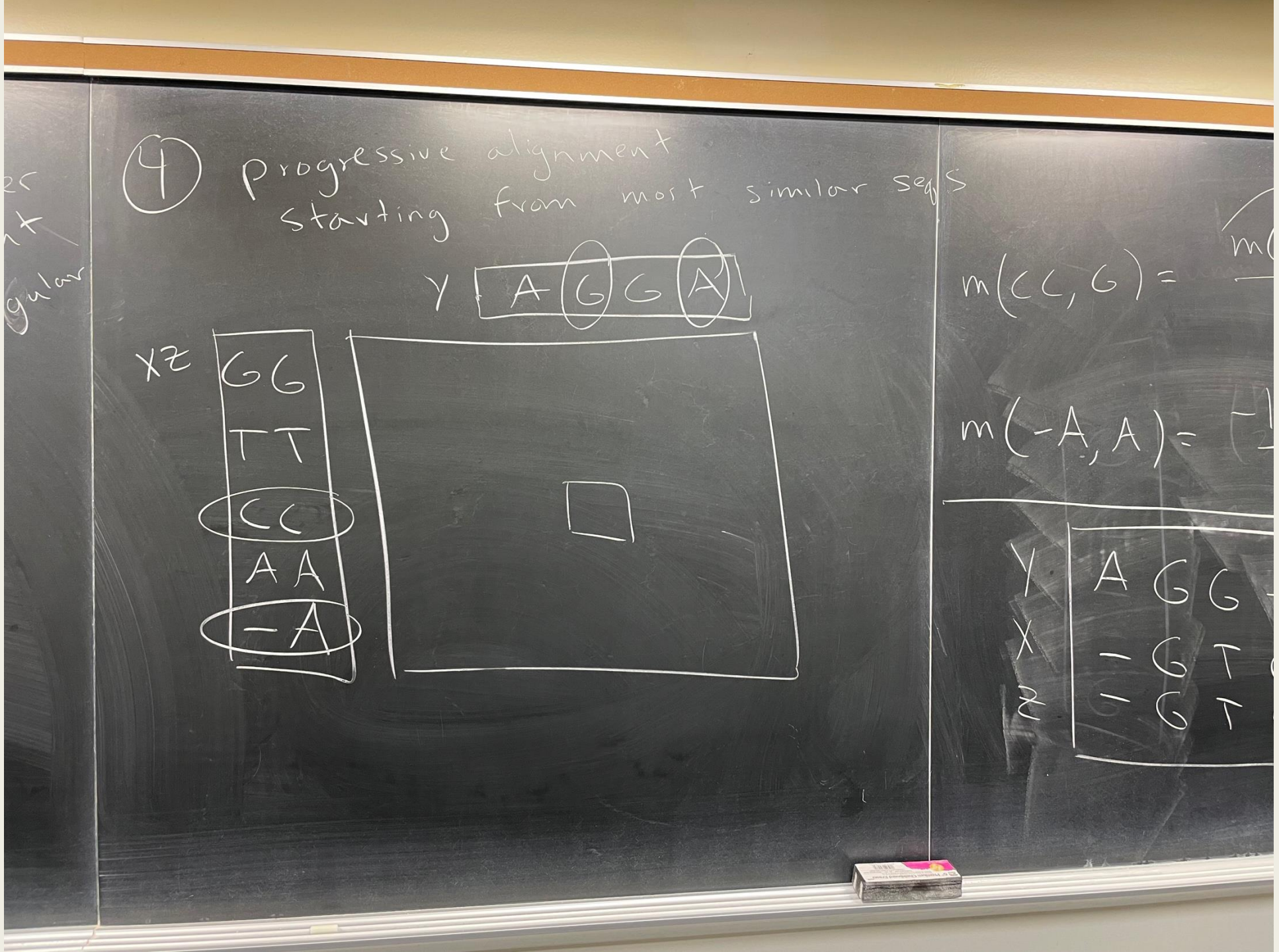


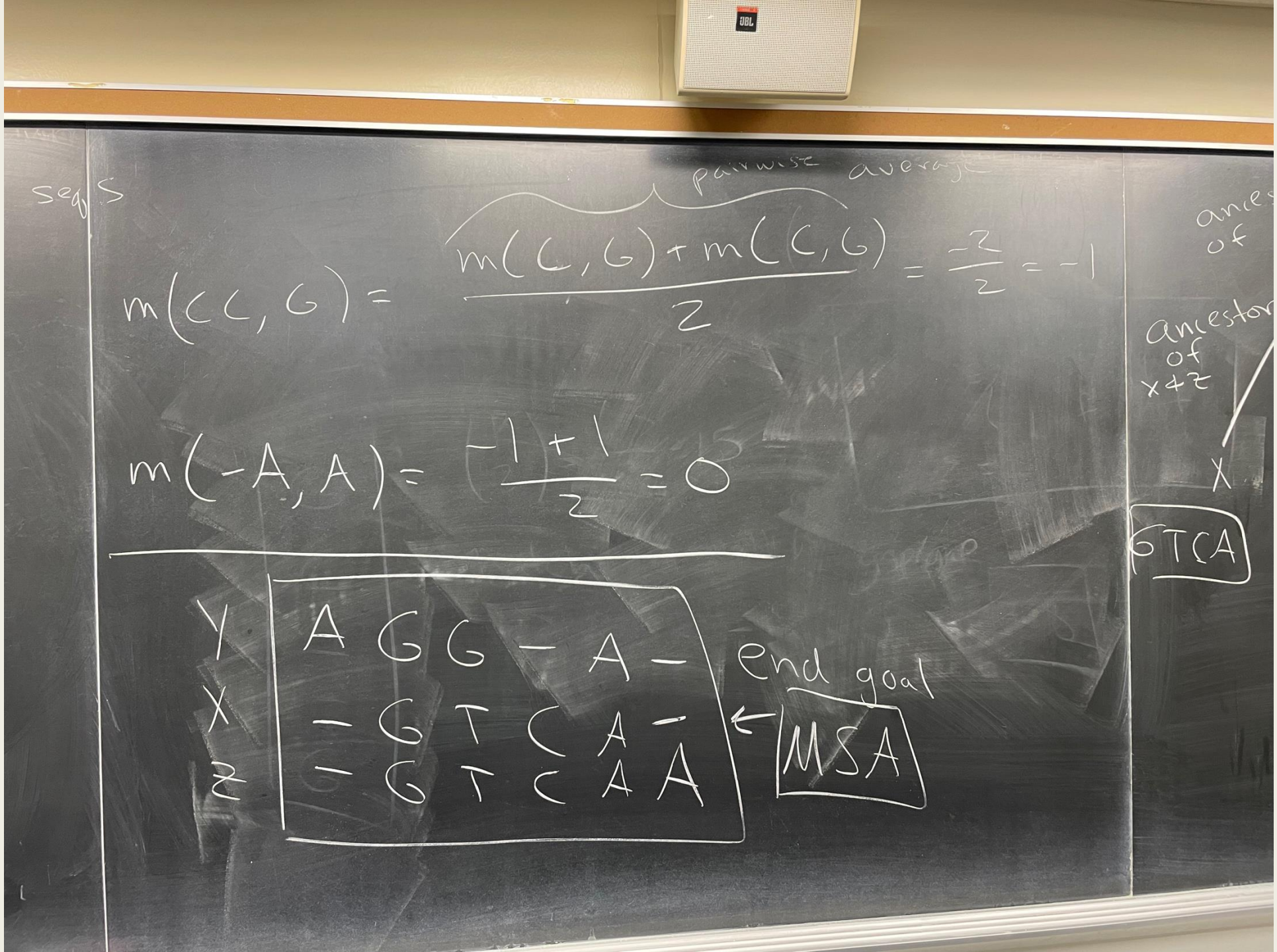
ClustaI-W workflow

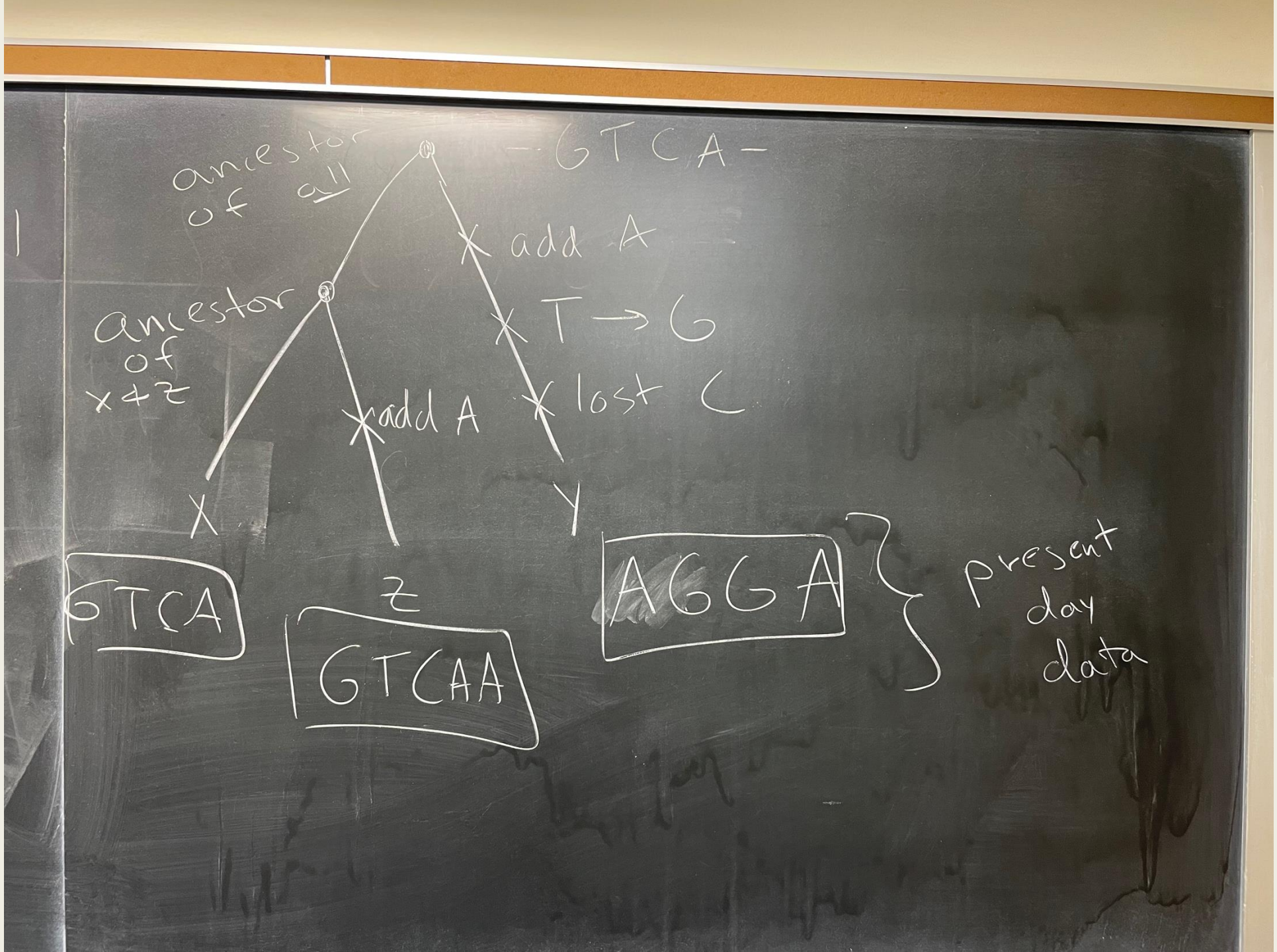



x = GTCA
y = AGGA
z = GTCAA
① pairwise alignment
x: -GTCA
y: AGG-A
-1
x: GTCA-
z: GTCAA
3
y: AGG-A-
z: -GTCAA
-2
score?
m = {+1 if Sa
{-1 if not
g = -1


same
not
score matrix
X Y Z
X
Y
Z
-1
3
-2
upper right triangular
① progressive alignmen
starting from
Y A G
XZ
G G
T T
C C
A A
– A
② create a tree (unrooted)
root
Y
X
Z
③ root the tree
X
Z
Y

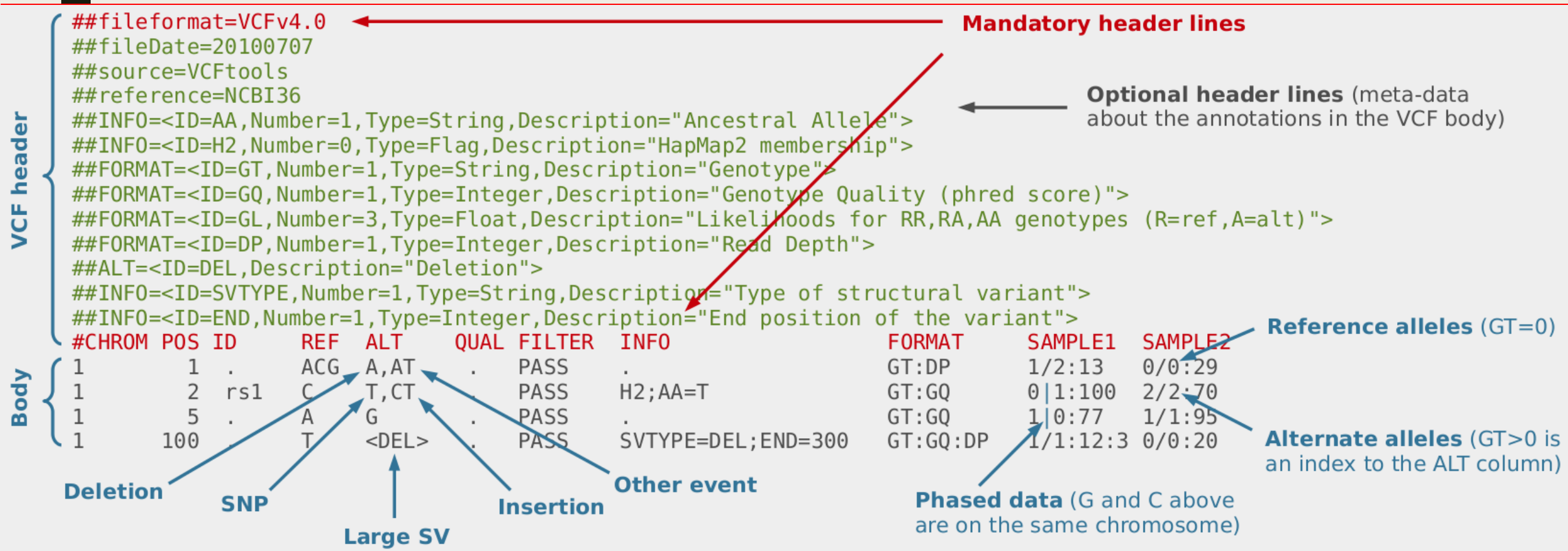

④ progressive alignment
starting from most similar seqs
Y: A G G A
XZ
G G
T T
C C
A A
- A
m(C, G) =
m(-A, A) =
Y
X
Z
A G G
- G
- G


seqs
pairwise average
$m(C,G) = \frac{m(C,G)+m(C,G)}{2} = \frac{-2}{2} = -1$
$m(-A,A) = \frac{-1+1}{2} = 0$
Y A G G – A –
X – G T C A –
Z – G T C A A
end goal
MSA
ancestor of X
GTCA


ancestor of all
-GTCA-
add A
ancestor of x≠z
T→G
add A
lost C
X
Y
Z
GTCA
GTCAA
AGGA
present day data

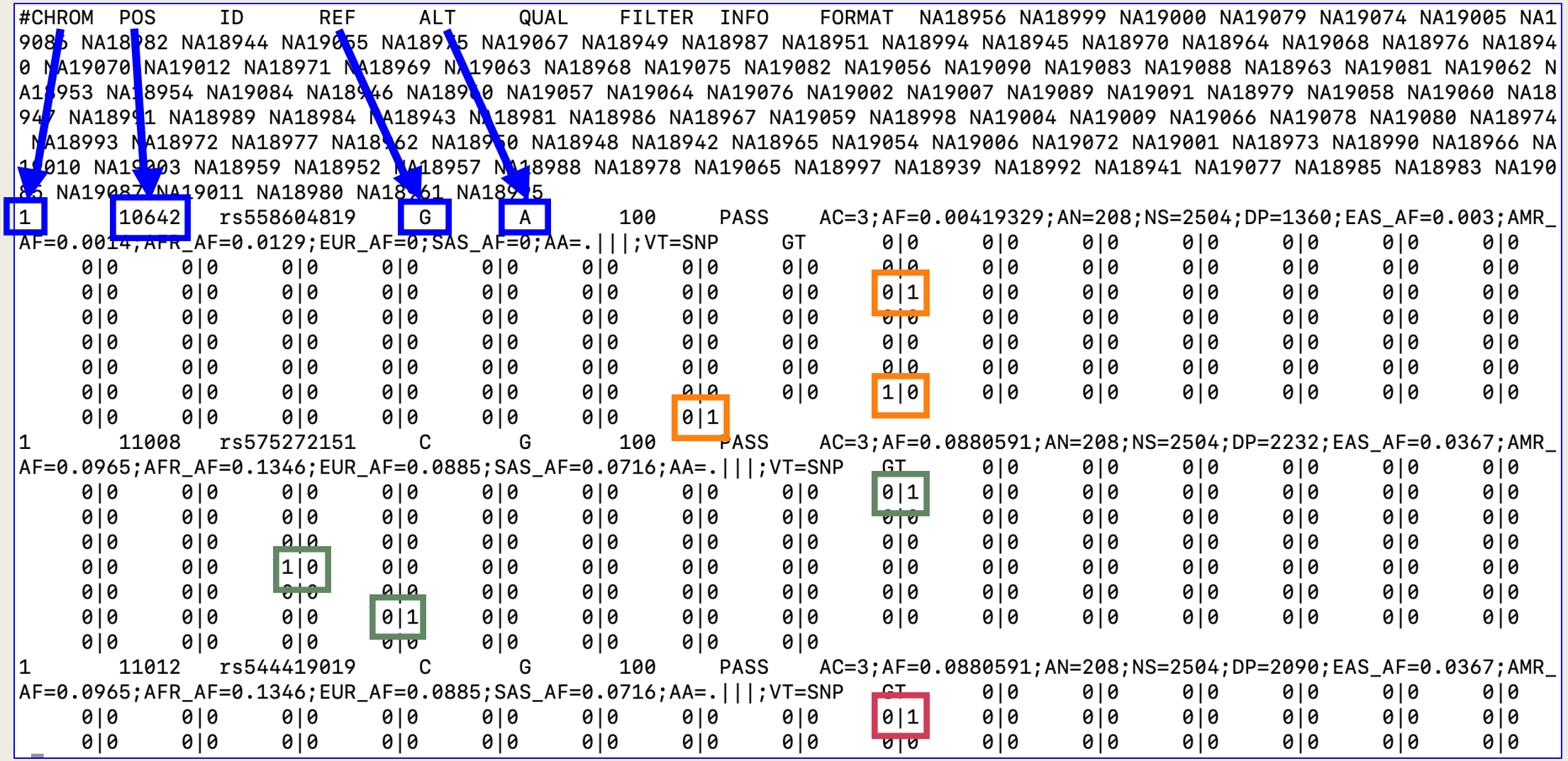
Begin: using variation to reconstruct evolutionary events

VCF file format

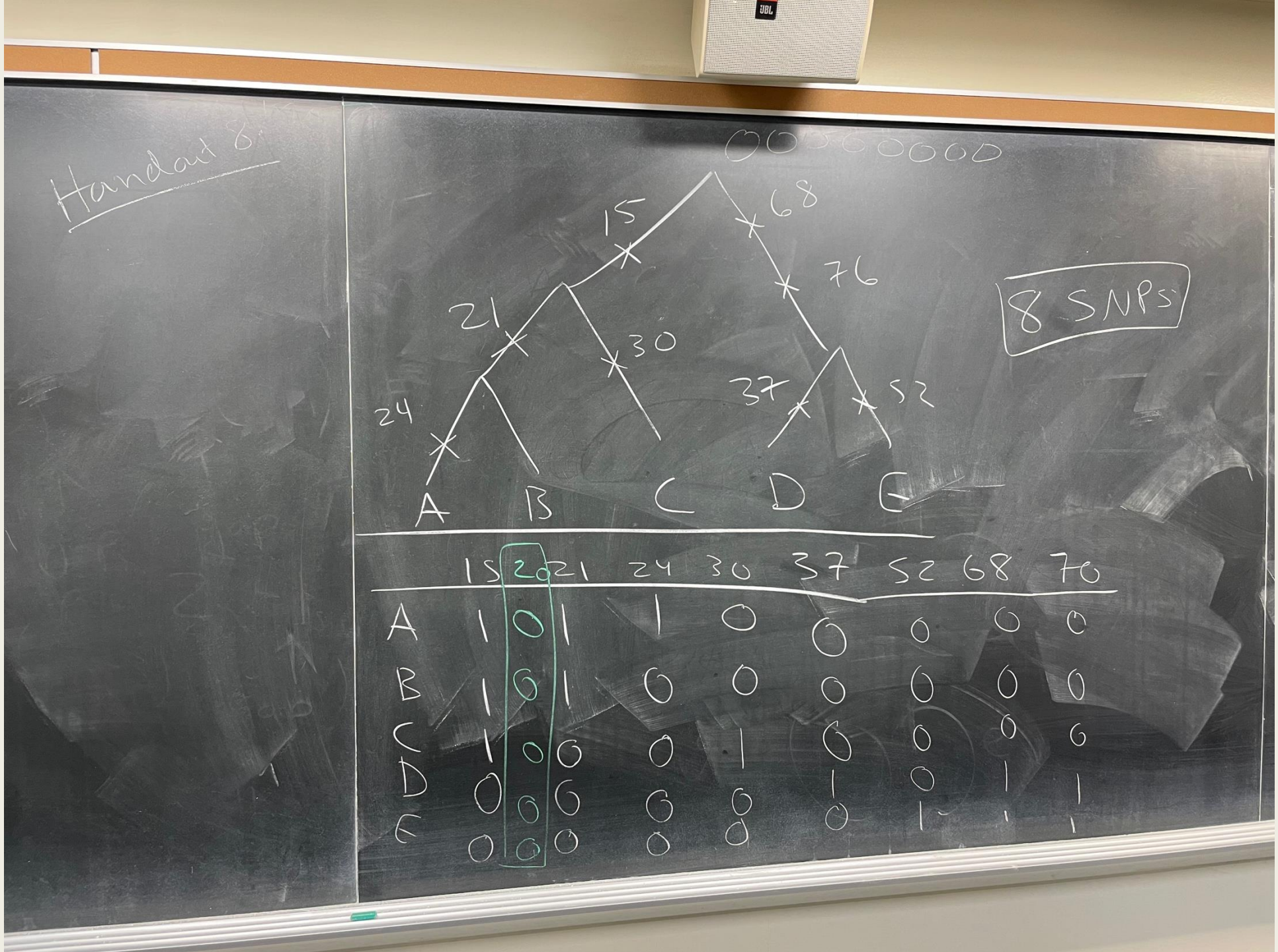


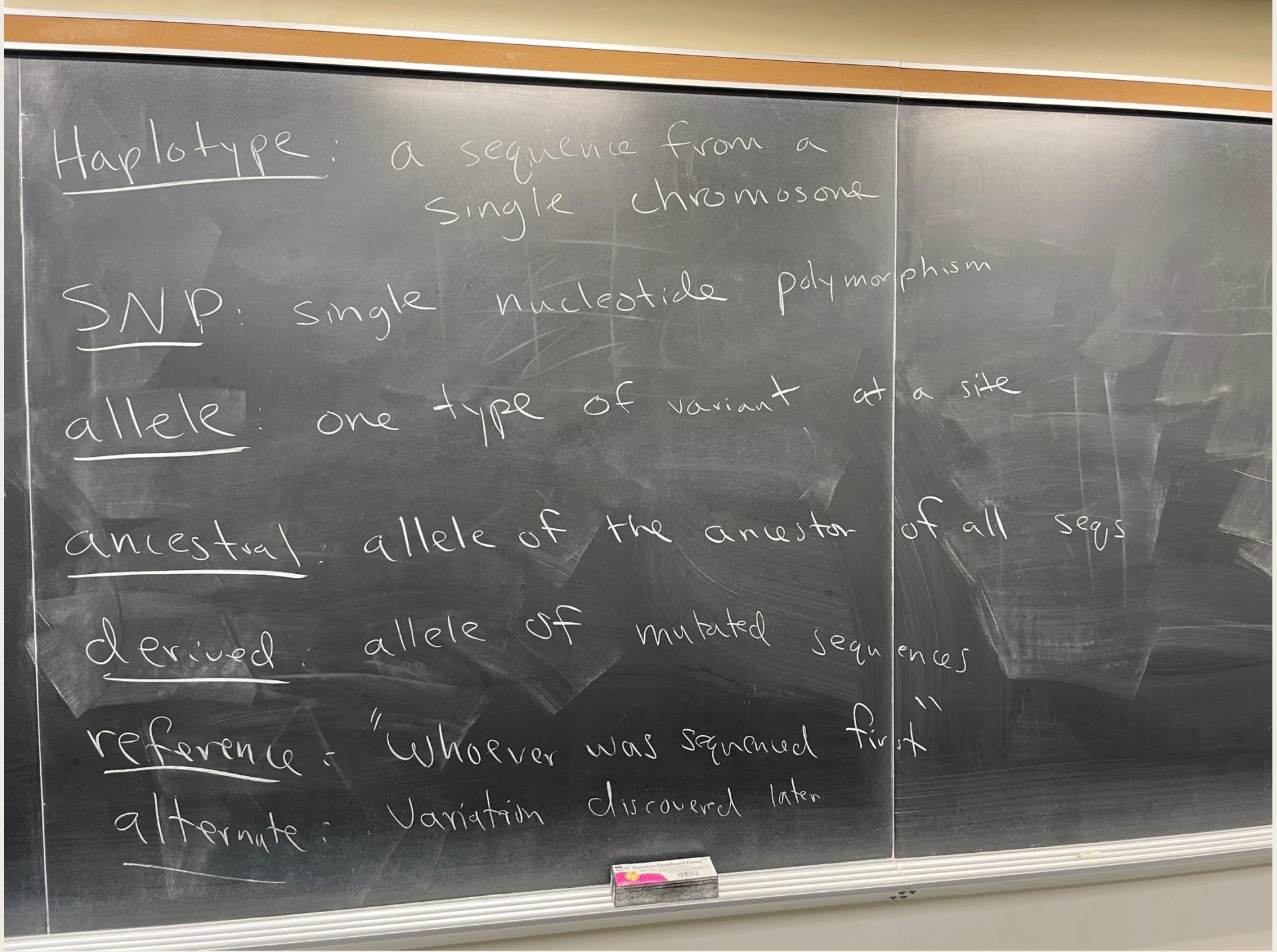
Credit: Dan Bolster (EBI)

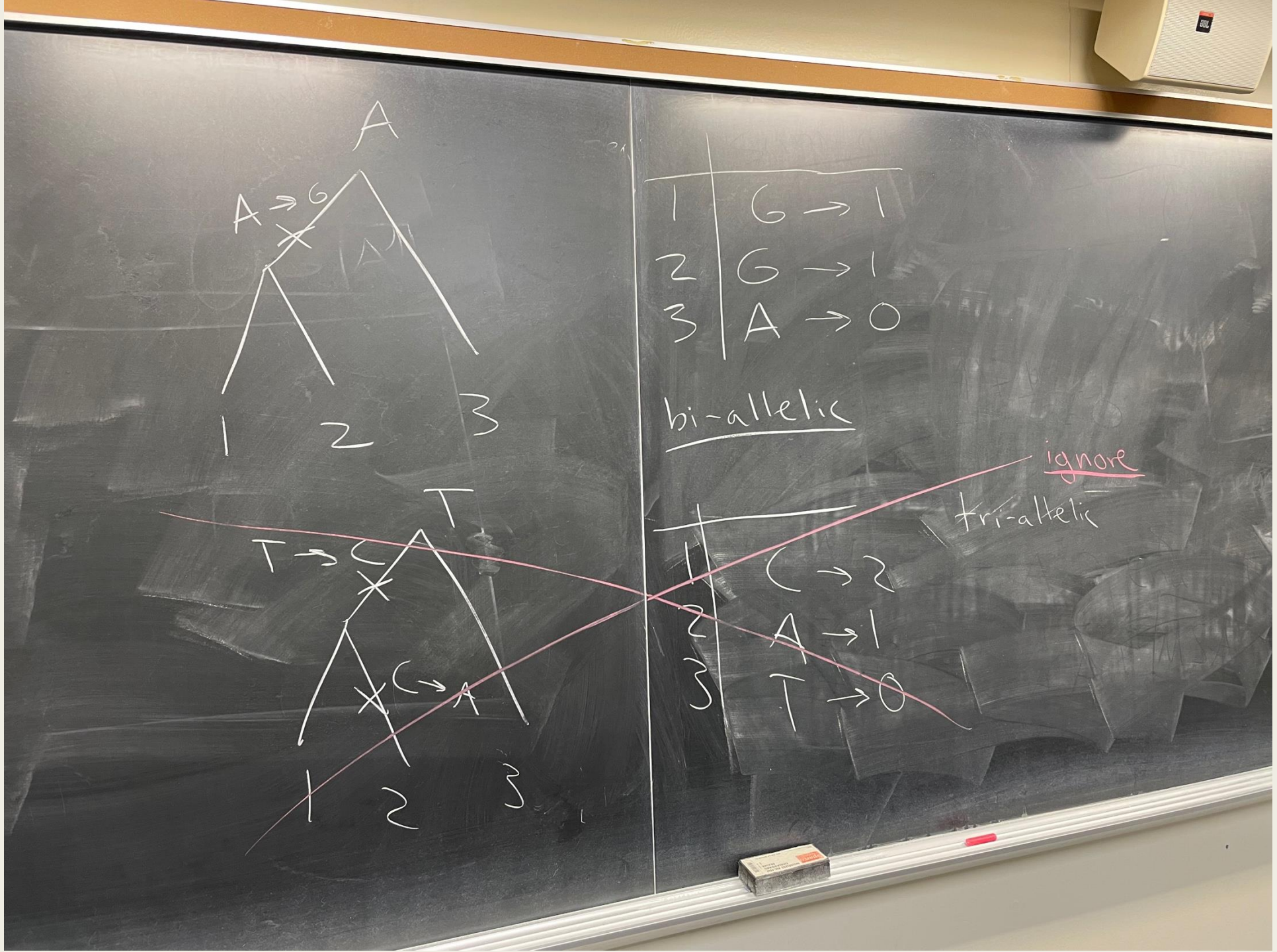
VCF file format



1000 Genomes Project (JPT population)


Handout 8
15
68
21
76
30
24
37
52
A
B
C
D
E
8 SNPs
15 20 21 24 30 37 52 68 70
A 1 0 1 1 0 0 0 0 0
B 1 0 1 0 0 0 0 0 0
C 1 0 0 0 1 0 0 0 0
D 0 0 0 0 0 0 0 1 1
E 0 0 0 0 0 0 1 1 1


Haplotype: a sequence from a single chromosome
SNP: single nucleotide polymorphism
allele: one type of variant at a site
ancestral: allele of the ancestor of all seqs
derived: allele of mutated sequences
reference: "whoever was sequenced first"
alternate: variation discovered later


A
A→G
1
2
3
1 G→1
2 G→1
3 A→0
bi-allelic
ignore
tri-allelic
T
T→C
C→A
1
2
3
1 C→2
2 A→1
3 T→0