

CS 466

Introduction to Bioinformatics

Lecture 9

Mohammed El-Kebir

October 1, 2018



Course Announcements

Instructor:

- Mohammed El-Kebir (melkebir)
- Office hours: Mondays, 3:15-4:15pm

TA:

- Anusri Pampari (pampari2)
- Office hours: Thursdays, 11:00-11:59am in SC 4105

Homework 2 due Oct. 5 by 11:59pm

Midterm on Oct. 10, 7-9pm, 1310 DCL

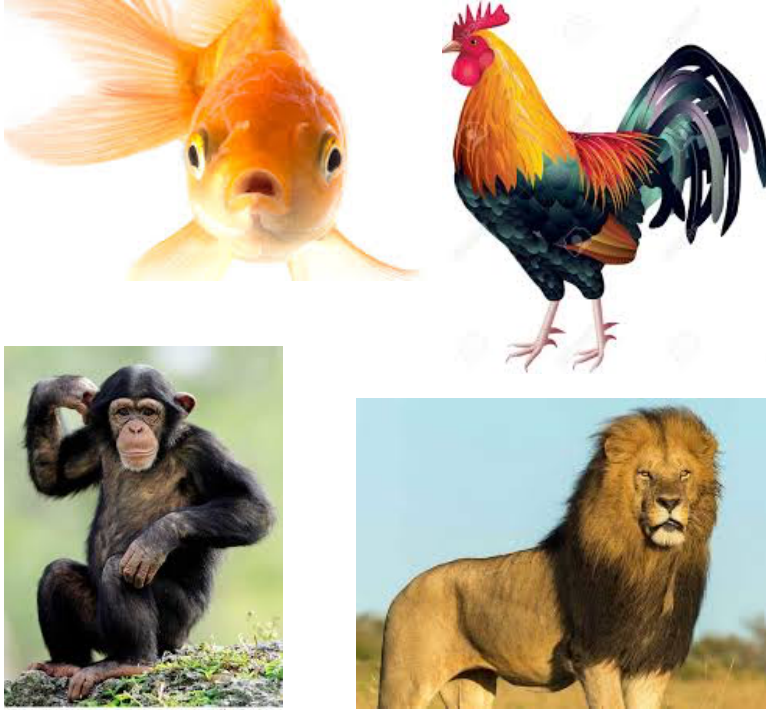
Outline

- Multiple sequence alignment
- Exact algorithm
- Sum-of-pairs (SP) score
- Carillo-Lipman
- Heuristic approaches

Reading:

- Material based on Chapter 14.6 in book “Algorithms on Strings, Trees and Sequences” by Dan Gusfield

Motivation



Simultaneous alignment of multiple (> 2) sequences enables inference of subtle similarities that are conserved in more than two species

```

      *           :           *           : : :
Q5E940_BOVIN  -----MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKOMQOIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE 76
RLA0_HUMAN   -----MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKOMQOIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE 76
RLA0_MOUSE   -----MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKOMQOIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE 76
RLA0_RAT      -----MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKOMQOIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE 76
RLA0_CHICK    -----MPREDRATWKSNYFMKIIQLLDDYPKCFVVGADNVGSKOMQOIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE 76
RLA0_RANSY    -----MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKOMQOIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--SALE 76
Q7ZUG3_BRARE  -----MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKOMQOIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE 76
RLA0 ICTPU    -----MPREDRATWKSNYFLKIIQLLNDYPKCFIVGADNVGSKOMQOIRMSLRGK-AIVLMGKNTMMRKAIRGHLENN--PALE 76
RLA0_DROME    -----MVRENKAAWKAQYFIKVVLEFDFPKCFIVGADNVGSKOMQOIRMSLRGL-AVVLMGKNTMMRKAIRGHLENN--PQLE 76
RLA0_DICDI    -----MSGAG-SKRKKLFIEKATKLFITYDKMIVAEADNVGSKOQKIRKSRIRGI-GAVLMGKNTMIRKVIKIRDLADSK--PELD 75
Q54LP0_DICDI  -----MSGAG-SKRKNVFIEKATKLFITYDKMIVAEADNVGSKOQKIRKSRIRGI-GAVLMGKNTMIRKVIKIRDLADSK--PELD 75
RLA0_PLAF8    -----MAKLSKQQKQMYIEKLSLIQQYSKILIVHVDNVGSKNOMASVRKSLRGK-ATILMGKNTIRIRTALKKNLQAV--PQIE 76
RLA0_SULAC    -----MIGLAVTTTKKIAKWKVDEVAELTEKLEKTHKTIIIANIEGFPADKLHEIRKKLRGK-ADIKVTKNNLFIKALKNAG-----YDTK 79
RLA0_SULTO    -----MRIMAVITQERKIAKWKIEEVKELEQLREYHTIIIIANIEGFPADKLHDIRKKMRGM-AEIKVTKNTLFGIAAKNAG-----LDVS 80
RLA0_SULSO    -----MKRLALALKQKVASWVLEEVKELTELKNSNTILIGNLEGFPADKLHEIRKKLRGK-ATIKVTKNTLFGIAAKNAG-----IDIE 80
RLA0_AERPE    MSVVS LVGQMYKREKPIPEWKTLMLELEELFSKHRVVFADLTGPTTFVVRVRKKLWKK-YPMVAVKRIILRAMKAAGLE---LDDN 86
RLA0_PYRAE    -MMLAIGKRRYVRTROYIPARKVKIVSEATELLQKYPYVFLFDLHGLSSRILHEYRYRLRY-GVIKIKPTLFKIAFTKVYGG---IPAE 85
RLA0_METAC    -----MAEERHTEHIPQWKDEIENIKELIQSHKVFVGMVIEGILATKQIRRDLDKDV-AVLKVSNTLTERALNQLG-----ETIP 78
RLA0_METMA    -----MAEERHTEHIPQWKDEIENIKELIQSHKVFVGMVIEGILATKQIRRDLDKDV-AVLKVSNTLTERALNQLG-----ESIP 78
RLA0_ARCFU    -----MAAVRGS---PPEYKVRAVEEIKRMISKPVAIVSFRNVPAGQMKIRREFRGK-AEIKVVKNTLLERALDALG-----GDYL 75
RLA0_METKA    MAVKAKGQPPSGYE PKVAEWKRREVKELELMDEYENVGLVDLEGIPAPQLOEIRAKLRERDTIIRMSRNTLMRIALEEKLDER--PELE 88
RLA0_METTH    -----MAHVAEWKKKEVQELHDLIKGYEVVGIANLADIPAROLOKMRQTLRDS-ALIRMSKKTLLISLALAKAGREL--ENVD 74
RLA0_METTL    -----MITAESEHKIAPWKIEEVNKLKELKNGQIIVALVDMMEVPAQLOEIRDKIR-GTMTLKMSRNTLIERAIKEVAEETGNPEFA 82
RLA0_METVA    -----MIDAKSEHKIAPWKIEEVNALKELLSANVIALIDMMEVPAVQLOEIRDKIR-DQMTLKMSRNTLIKRAVEEVAEETGNPEFA 82
RLA0_METJA    -----METKVKAHVAPWKIEEVKTLKGLIKSKPVAIVDMMDVPAPQLOEIRDKIR-DKVKLRMSRNTLIIRALKEAAEELNNPKLA 81
RLA0_PYRAB    -----MAHVAEWKKKEVEELANLIKSYPVVIALVDVSSMPAYPLSQMRRLIRENGGLLRVSNTLIELAIKKAQELGKPELE 77
RLA0_PYRHO    -----MAHVAEWKKKEVEELAKLIKSYPVVIALVDVSSMPAYPLSQMRRLIRENGGLLRVSNTLIELAIKKAQELGKPELE 77
RLA0_PYRFU    -----MAHVAEWKKKEVEELANLIKSYPVVIALVDVSSMPAYPLSQMRRLIRENGLLRVSNTLIELAIKKAQELGKPELE 77
RLA0_PYRKO    -----MAHVAEWKKKEVEELANLIKSYPVVIALVDVAGVPAYPLSKMRDKLR-GKALLRVSNTLIELAIKRAAQELGQPELE 76
RLA0_HALMA    -----MSAESERKTETIPEWKQEVDAIVEMIESYESVGVVNIAGIPSRQLODMRRDLHGT-AELRVSNTLLEALDDVD-----DGLE 79
RLA0_HALVO    -----MSESEVRQTEVIPQWKREVDLVDVDFIESYESVGVVGVAGIPSRQLODMRRDLHGS-AAVRVSNTLVNRAALDEVN-----DGFE 79
RLA0_HALSA    -----MSAEEQRTTEEVPEWKRQEVAVLDLLETYDSVGVVNVGTGIPSKQLODMRRGLHGQ-AALRMSRNTLLVRALEEAG-----DGLD 79
RLA0_THEAC    -----MKEVSQKKELVNEITQRIKASRSVAIVDTAGIRTRQIQDIRGKNRGK-INLKVIKKTLLFKALENLGD-----EKLS 72
RLA0_THEVO    -----MRKINPKKKEIVSELAQDITKSKAVAVDIKGVRTROMQDIRAKNRDK-VKIKVVKKTLLFKALDSIND-----EKLT 72
RLA0_PICTO    -----MTEPAQWKIDFVKNLENEINSRKVAIVS IKGLRNNEFQKIRNSIRDK-ARIKVSARLLRLAIENTGK-----NNIV 72
ruler 1.....10.....20.....30.....40.....50.....60.....70.....80.....90

```

“Pairwise alignment whispers ... multiple alignment shouts out loud”.
 Hubbard, Lesk, Tramontano, Nature Structural Biology 1996.

Multiple Sequence Alignment (MSA)

A **multiple sequence alignment** \mathcal{M} between k strings $\mathbf{v}_1, \dots, \mathbf{v}_k$ is a $k \times q$ matrix, where $q = \{\max\{|\mathbf{v}_i| : i \in [k]\}, \dots, \sum_{i=1}^k |\mathbf{v}_i|\}$ such that the i -th row contains the characters of \mathbf{v}_i in order with spaces '-' interspersed and no column contains k spaces

\mathbf{v}_1	A	T	-	G	C	G	-
\mathbf{v}_2	A	-	C	G	T	-	C
\mathbf{v}_3	A	T	C	A	C	-	A

Question: How to score a multiple sequence alignment?

Scoring a Multiple Sequence Alignment

\mathbf{v}_1	A	T	-	G	C	G	-
\mathbf{v}_2	A	-	C	G	T	-	C
\mathbf{v}_3	A	T	C	A	C	-	A

Question: How to score a multiple sequence alignment?

Pairwise scoring function:

$$\delta : (\Sigma \cup \{-\}) \times (\Sigma \cup \{-\}) \rightarrow \mathbb{R}$$

Scoring a Multiple Sequence Alignment

\mathbf{v}_1	A	T	-	G	C	G	-
\mathbf{v}_2	A	-	C	G	T	-	C
\mathbf{v}_3	A	T	C	A	C	-	A

Question: How to score a multiple sequence alignment?

Pairwise scoring function:

$$\delta : (\Sigma \cup \{-\}) \times (\Sigma \cup \{-\}) \rightarrow \mathbb{R}$$

k -wise scoring function:

$$\delta : (\Sigma \cup \{-\})^k \rightarrow \mathbb{R}$$

Outline

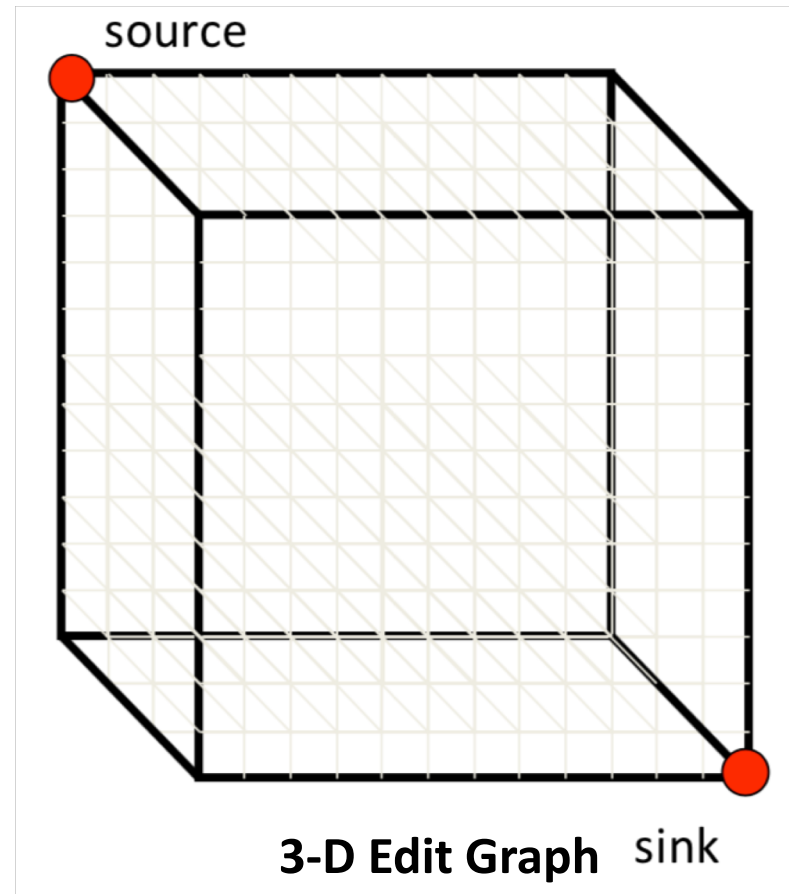
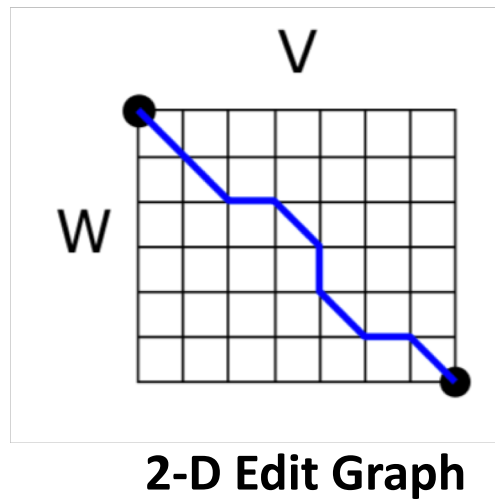
- Multiple sequence alignment
- Exact algorithm
- Sum-of-pairs (SP) score
- Carillo-Lipman
- Heuristic approaches

Reading:

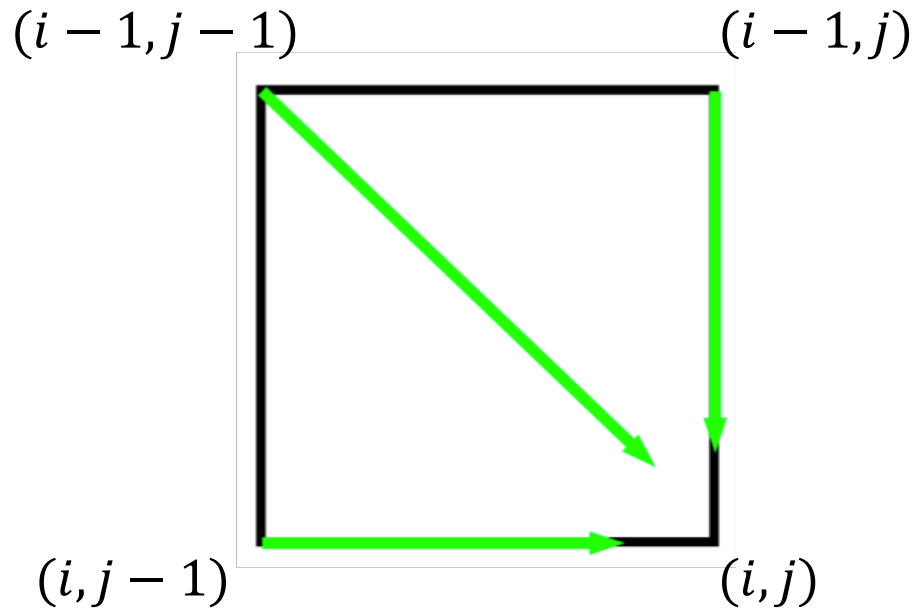
- Material based on Chapter 14.6 in book “Algorithms on Strings, Trees and Sequences” by Dan Gusfield

Aligning Three Sequences

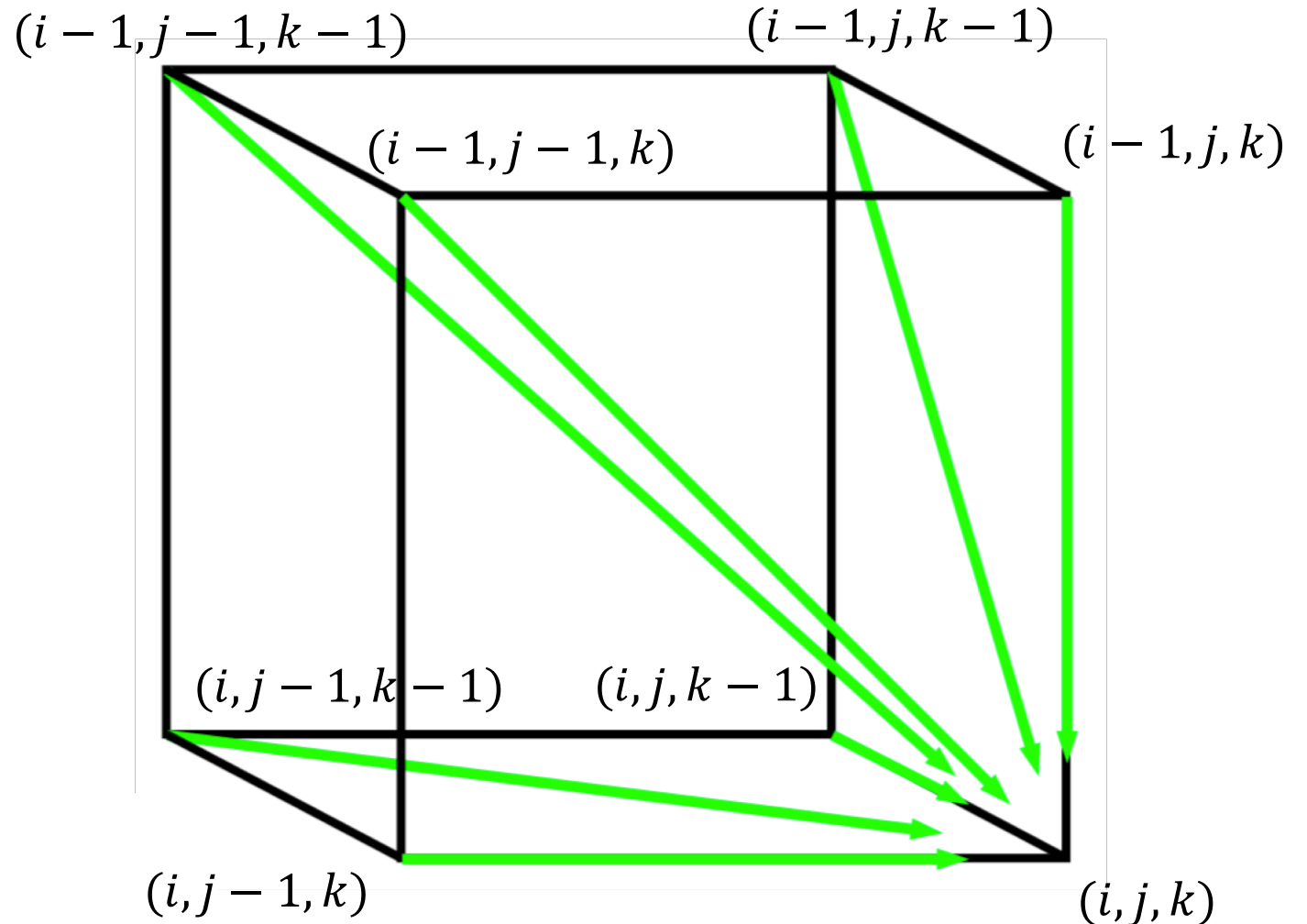
- Same strategy as pairwise edit distance
- Use 3-D cube, with each axis representing an input sequence
- Alignment is a path from source to sink



2-D vs 3-D Vertex Neighborhood



2-D Neighborhood
(3 edges)

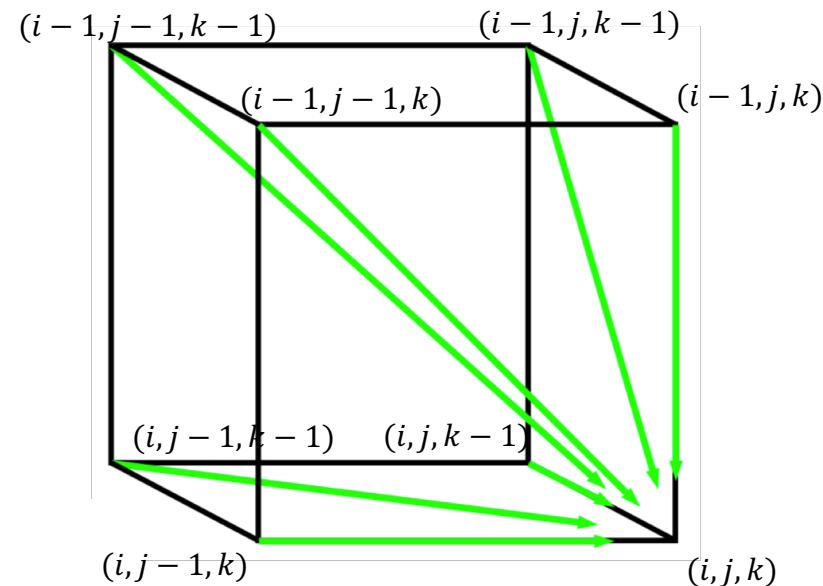


3-D Neighborhood
(7 edges)

3-D Sequence Alignment

$\delta(x, y, z)$ is an entry in 3-D scoring matrix

Given three sequences each of length n ,
running time: $O(n^3)$



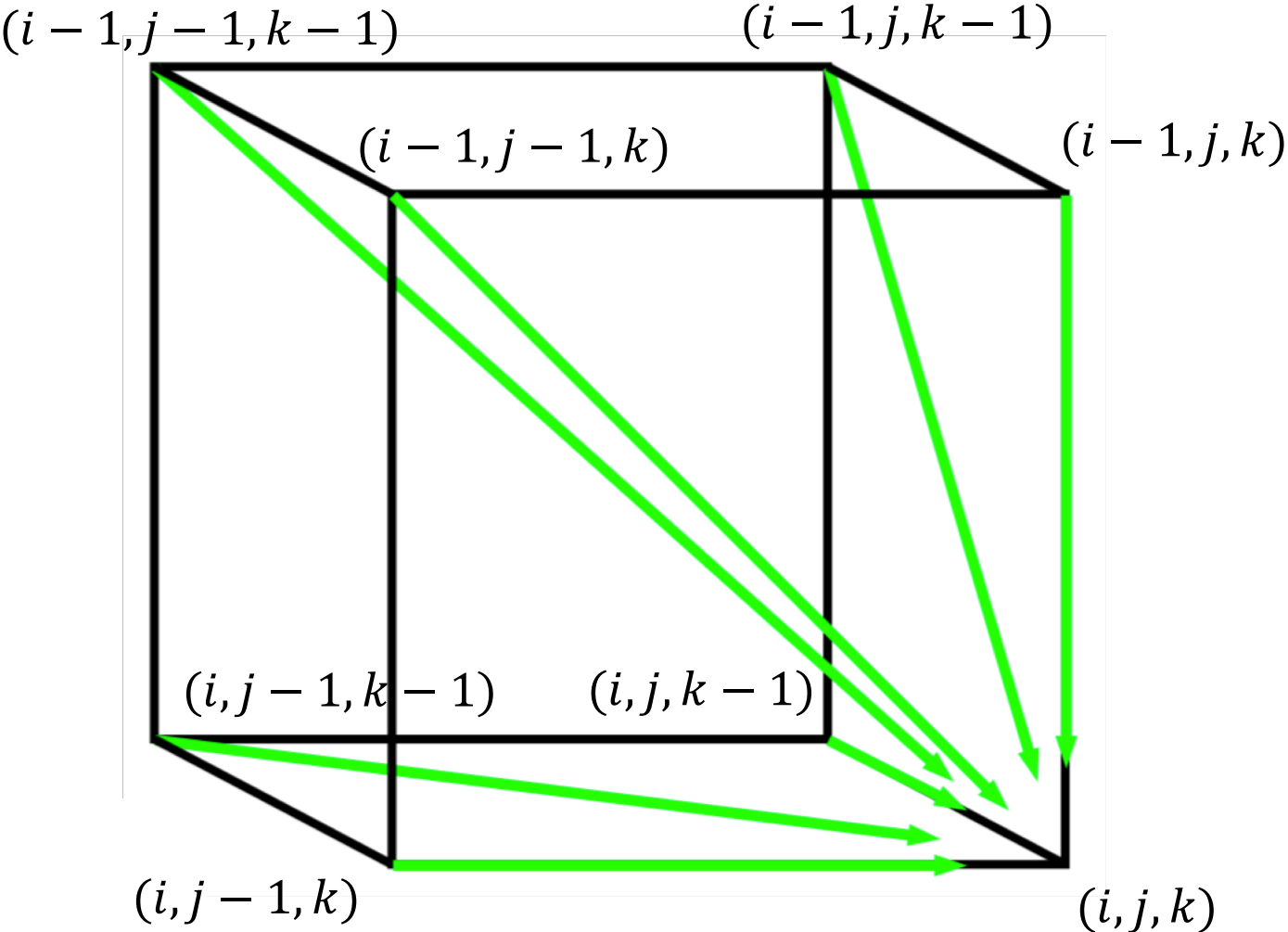
$$s[i, j, k] = \max \left\{ \begin{array}{l} s[i-1, j-1, k-1] + \delta(v_i, w_j, u_k), \\ s[i-1, j-1, k] + \delta(v_i, w_j, -), \\ s[i-1, j, k-1] + \delta(v_i, -, u_k), \\ s[i, j-1, k-1] + \delta(-, w_j, u_k), \\ s[i-1, j, k] + \delta(v_i, -, -), \\ s[i, j-1, k] + \delta(-, w_j, -), \\ s[i, j, k-1] + \delta(-, -, u_k), \end{array} \right.$$

no gaps

one gap

two gaps

3-D vs k -D Vertex Neighborhood



3-D Neighborhood
(7 edges)

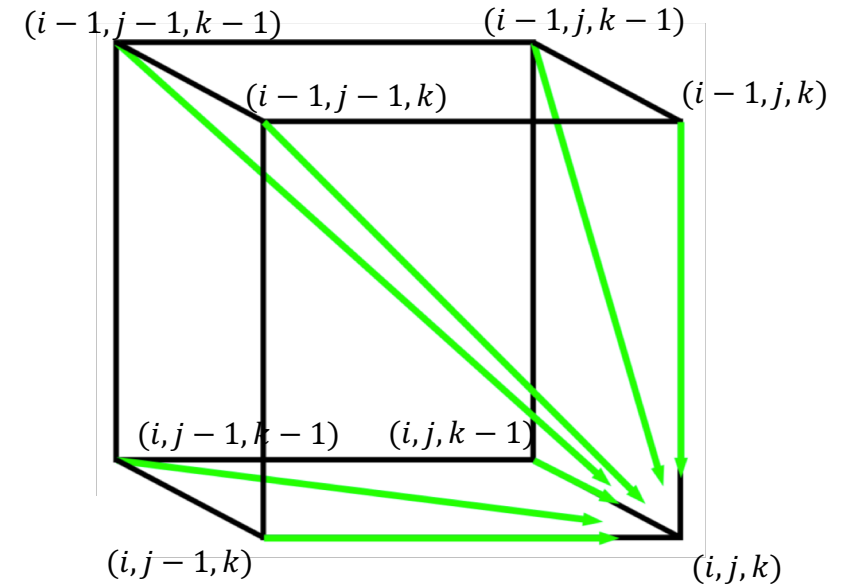
- $(i_1 - 1, i_2 - 1, \dots, i_{k-1} - 1, i_k - 1)$
- $(i_1 - 1, i_2 - 1, \dots, i_{k-1} - 1, i_k)$
- ...
- $(i_1, i_2 - 1, \dots, i_{k-1} - 1, i_k - 1)$
- ...
- $(i_1 - 1, i_2, \dots, i_{k-1}, i_k)$
- ...
- $(i_1, i_2, \dots, i_{k-1}, i_k - 1)$

k -D Neighborhood
($2^k - 1$ edges)

k -D Sequence Alignment

$\delta(x_1, \dots, x_k)$ is an entry in k -D scoring matrix

Given k sequences each of length n ,
running time: $O(2^k n^k)$



$$s[i_1, i_2, \dots, i_{k-1}, i_k] = \max \left\{ \begin{array}{l} s[i_1 - 1, i_2 - 1, \dots, i_{k-1} - 1, i_k - 1] + \delta(\mathbf{v}_1[i_1], \mathbf{v}_2[i_2], \dots, \mathbf{v}_{k-1}[i_{k-1}], \mathbf{v}_k[i_k]) \\ s[i_1 - 1, i_2 - 1, \dots, i_{k-1} - 1, i_k] + \delta(\mathbf{v}_1[i_1], \mathbf{v}_2[i_2], \dots, \mathbf{v}_{k-1}[i_{k-1}], -) \\ \vdots \\ s[i_1, i_2 - 1, \dots, i_{k-1} - 1, i_k - 1] + \delta(-, \mathbf{v}_2[i_2], \dots, \mathbf{v}_{k-1}[i_{k-1}], \mathbf{v}_k[i_k]) \\ \vdots \\ s[i_1 - 1, i_2, \dots, i_{k-1}, i_k] + \delta(\mathbf{v}_1[i_1], -, \dots, -, -) \\ \vdots \\ s[i_1, i_2, \dots, i_{k-1}, i_k - 1] + \delta(-, -, \dots, -, \mathbf{v}_k[i_k]) \end{array} \right. \begin{array}{l} \text{no gaps} \\ \text{one gap} \\ \\ \\ \text{\(k - 1\) gaps} \end{array}$$

Multiple Sequence Alignment – Running Time

Given 2 sequences each of length n ,
running time: $O(n^2)$

Given 3 sequences each of length n ,
running time: $O(n^3)$

Given k sequences each of length n ,
running time: $O(2^k n^k)$

Multiple Sequence Alignment – Running Time

Given 2 sequences each of length n ,
running time: $O(n^2)$

Given 3 sequences each of length n ,
running time: $O(n^3)$

Given k sequences each of length n ,
running time: $O(2^k n^k)$

Question: Can we align
 k sequences each of
length n in time
 $O(\text{poly}(n))$?

Multiple Sequence Alignment – Running Time

Given 2 sequences each of length n ,
running time: $O(n^2)$

Given 3 sequences each of length n ,
running time: $O(n^3)$

Given k sequences each of length n ,
running time: $O(2^k n^k)$

Question: Can we align
 k sequences each of
length n in time
 $O(\text{poly}(n))$?

Let's look at a more
wieldy scoring function

Outline

- Multiple sequence alignment
- Exact algorithm
- Sum-of-pairs (SP) score
- Carillo-Lipman
- Heuristic approaches

Reading:

- Material based on Chapter 14.6 in book “Algorithms on Strings, Trees and Sequences” by Dan Gusfield

Multiple Alignment Induces Pairwise Alignments

v_1	A	T	-	G	C	G	-
v_2	A	-	C	G	T	-	C
v_3	A	T	C	A	C	-	A

v_1	A	T	-	G	C	G	-
v_2	A	-	C	G	T	-	C

v_1	A	T	-	G	C	G	-
v_3	A	T	C	A	C	-	A

v_2	A	-	C	G	T	C
v_3	A	T	C	A	C	A

Resulting columns with -/- are removed

Sum-of-Pairs (SP) Score

v_1	A	T	-	G	C	G	-
v_2	A	-	C	G	T	-	C
v_3	A	T	C	A	C	-	A

$S(v_i, v_j)$ is score of induced pairwise alignment of sequences (v_i, v_j)

Multiple sequence alignment \mathcal{M}

v_1	A	T	-	G	C	G	-
v_2	A	-	C	G	T	-	C

v_1	A	T	-	G	C	G	-
v_3	A	T	C	A	C	-	A

v_2	A	-	C	G	T	C
v_3	A	T	C	A	C	A

$$\text{SP-score}(\mathcal{M}) = \sum_{i=1}^k \sum_{j=i+1}^k S(v_i, v_j)$$

Sum-of-Pairs (SP) Score – Example

\mathbf{v}_1	A	T	G	-	C
\mathbf{v}_2	A	-	G	-	C
\mathbf{v}_3	A	T	C	C	C

Multiple sequence alignment \mathcal{M}

Match score: 3
Mismatch score: 1
Gap score: $-\sigma$

Question: Calculate

$$\text{SP-score}(\mathcal{M}) = \sum_{i=1}^k \sum_{j=i+1}^k S(\mathbf{v}_i, \mathbf{v}_j)$$

Sum-of-Pairs (SP) Score – Example

v_1	A	T	G	-	C
v_2	A	-	G	-	C
v_3	A	T	C	C	C

Multiple sequence alignment \mathcal{M}

Question: Calculate

$$\text{SP-score}(\mathcal{M}) = \sum_{i=1}^k \sum_{j=i+1}^k S(\mathbf{v}_i, \mathbf{v}_j)$$

Match score: 3
Mismatch score: 1
Gap score: $-\sigma$

We can sum over scores for the columns, ignoring -/-

Multiple Sequence Alignment Problem w/ SP-Score

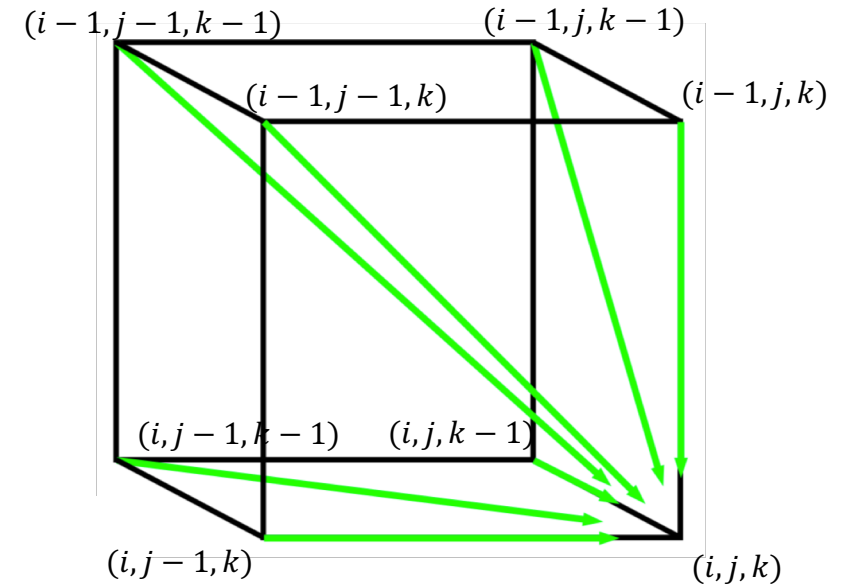
A **multiple sequence alignment** \mathcal{M} between k strings $\mathbf{v}_1, \dots, \mathbf{v}_k$ is a $k \times q$ matrix, where $q = \{\max\{|\mathbf{v}_i| : i \in [k]\}, \dots, \sum_{i=1}^k |\mathbf{v}_i|\}$ such that the i -th row contains the characters of \mathbf{v}_i in order with spaces '-' interspersed and no column contains k spaces

MSA-SP problem: Given strings $\mathbf{v}_1, \dots, \mathbf{v}_k$ find multiple sequence alignment \mathcal{M}^* with **minimum** value of $\text{SP-score}(\mathcal{M}^*) = \sum_{i=1}^k \sum_{j=i+1}^k S(\mathbf{v}_i, \mathbf{v}_j)$ where $S(\mathbf{v}_i, \mathbf{v}_j)$ is the score of the induced pairwise alignment of $(\mathbf{v}_i, \mathbf{v}_j)$ in \mathcal{M}^*

3-D MSA-SP

$\delta(x, y, z)$ is an entry in 3-D scoring matrix

Given three sequences each of length n ,
running time: $O(n^3)$

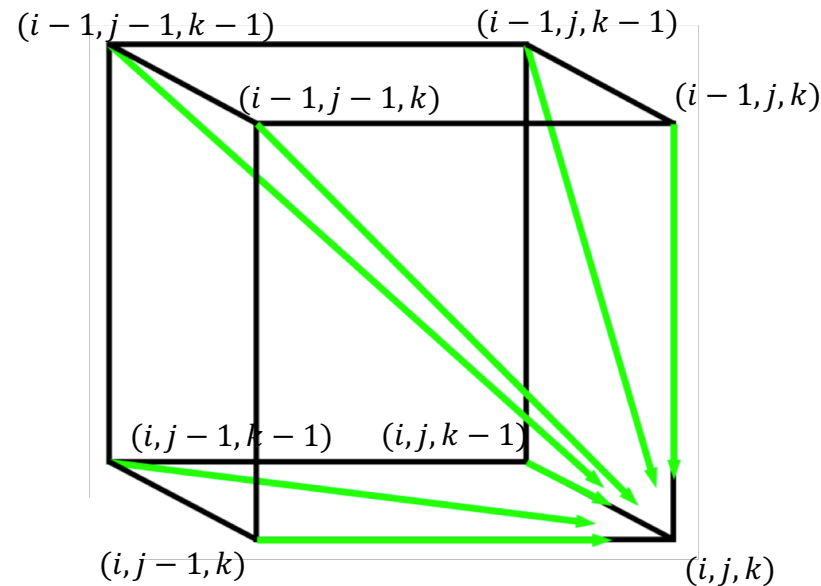


$$d[i_1, i_2, i_3] = \min \left\{ \begin{array}{l} d[i_1 - 1, i_2 - 1, i_3 - 1] + \delta(\mathbf{v}_1[i_1], \mathbf{v}_2[i_2]) + \delta(\mathbf{v}_1[i_1], \mathbf{v}_3[i_3]) + \delta(\mathbf{v}_2[i_2], \mathbf{v}_3[i_3]) \quad \text{no gaps} \\ d[i_1 - 1, i_2 - 1, i_3] + \delta(\mathbf{v}_1[i_1], \mathbf{v}_2[i_2]) + 2\sigma \\ d[i_1 - 1, i_2, i_3 - 1] + \delta(\mathbf{v}_1[i_1], \mathbf{v}_3[i_3]) + 2\sigma \\ d[i_1, i_2 - 1, i_3 - 1] + \delta(\mathbf{v}_2[i_2], \mathbf{v}_3[i_3]) + 2\sigma \quad \text{one gap} \\ d[i_1 - 1, i_2, i_3] + 2\sigma \\ d[i_1, i_2 - 1, i_3] + 2\sigma \\ d[i_1, i_2, i_3 - 1] + 2\sigma \quad \text{two gaps} \end{array} \right.$$

k -D MSA-SP

Computing SP-score in each case: $O(k^2)$ time

Given k sequences each of length n ,
running time: $O(k^2 2^k n^k)$



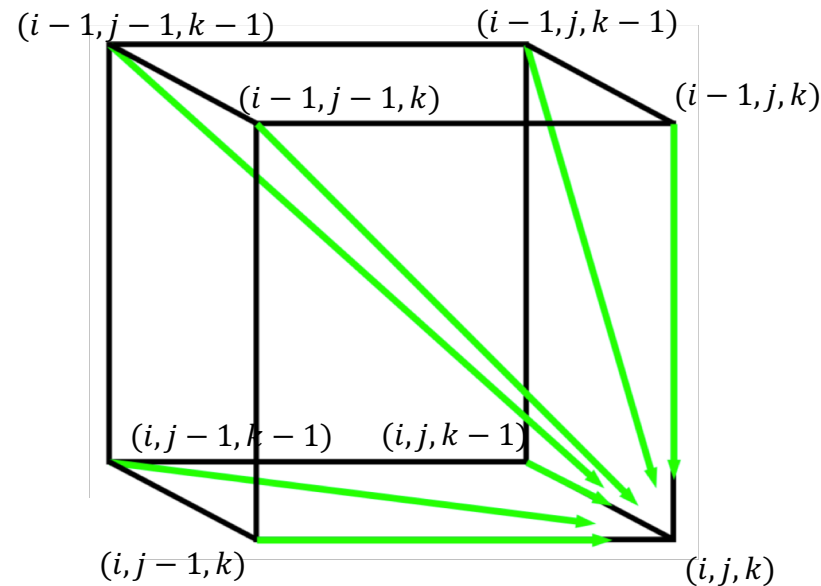
$$d[i_1, i_2, \dots, i_{k-1}, i_k] = \min \left\{ \begin{array}{l} s[i_1 - 1, i_2 - 1, \dots, i_{k-1} - 1, i_k - 1] + \sum_{p=1}^k \sum_{q=p+1}^k \delta(\mathbf{v}_p[i_p], \mathbf{v}_q[i_q]) \\ s[i_1 - 1, i_2 - 1, \dots, i_{k-1} - 1, i_k] + (k-1)\sigma + \sum_{p=1}^{k-1} \sum_{q=p+1}^{k-1} \delta(\mathbf{v}_p[i_p], \mathbf{v}_q[i_q]) \\ \vdots \\ s[i_1, i_2 - 1, \dots, i_{k-1} - 1, i_k - 1] + (k-1)\sigma + \sum_{p=2}^k \sum_{q=p+1}^k \delta(\mathbf{v}_p[i_p], \mathbf{v}_q[i_q]) \\ \vdots \\ s[i_1 - 1, i_2, \dots, i_{k-1}, i_k] + (k-1)\sigma \\ \vdots \\ s[i_1, i_2, \dots, i_{k-1}, i_k - 1] + (k-1)\sigma \end{array} \right.$$

} no gaps
} one gap
} $k-1$ gaps

k -D MSA-SP

Computing SP-score in each case: $O(k^2)$ time

Given k sequences each of length n ,
running time: $O(k^2 2^k n^k)$



$$d[i_1, i_2, \dots, i_{k-1}, i_k] = \min \left\{ \begin{array}{l} s[i_1 - 1, i_2 - 1, \dots, i_{k-1} - 1, i_k - 1] + \sum_{p=1}^k \sum_{q=p+1}^k \delta(\mathbf{v}_p[i_p], \mathbf{v}_q[i_q]) \\ s[i_1 - 1, i_2 - 1, \dots, i_{k-1} - 1, i_k] + (k-1)\sigma + \sum_{p=1}^{k-1} \sum_{q=p+1}^{k-1} \delta(\mathbf{v}_p[i_p], \mathbf{v}_q[i_q]) \\ \vdots \\ s[i_1, i_2 - 1, \dots, i_{k-1} - 1, i_k - 1] + (k-1)\sigma + \sum_{p=2}^k \sum_{q=p+1}^k \delta(\mathbf{v}_p[i_p], \mathbf{v}_q[i_q]) \\ \vdots \\ s[i_1 - 1, i_2, \dots, i_{k-1}, i_k] + (k-1)\sigma \\ \vdots \\ s[i_1, i_2, \dots, i_{k-1}, i_k - 1] + (k-1)\sigma \end{array} \right.$$

} no gaps
} one gap
} $k - 1$ gaps

Question: How many times gap penalty with 2 gaps?

Multiple Sequence Alignment Problem w/ SP-Score

MSA-SP problem: Given strings $\mathbf{v}_1, \dots, \mathbf{v}_k$ find multiple sequence alignment \mathcal{M}^* with **minimum** value of $\text{SP-score}(\mathcal{M}^*) = \sum_{i=1}^k \sum_{j=i+1}^k S(\mathbf{v}_i, \mathbf{v}_j)$ where $S(\mathbf{v}_i, \mathbf{v}_j)$ is the score of the induced pairwise alignment of $(\mathbf{v}_i, \mathbf{v}_j)$ in \mathcal{M}^*

Question: Can we align k sequences each of length n in time $O(\text{poly}(n))$?

Multiple Sequence Alignment Problem w/ SP-Score

MSA-SP problem: Given strings $\mathbf{v}_1, \dots, \mathbf{v}_k$ find multiple sequence alignment \mathcal{M}^* with **minimum** value of $\text{SP-score}(\mathcal{M}^*) = \sum_{i=1}^k \sum_{j=i+1}^k S(\mathbf{v}_i, \mathbf{v}_j)$ where $S(\mathbf{v}_i, \mathbf{v}_j)$ is the score of the induced pairwise alignment of $(\mathbf{v}_i, \mathbf{v}_j)$ in \mathcal{M}^*

Question: Can we align k sequences each of length n in time $O(\text{poly}(n))$?

No, MSA-SP is NP-hard.

[WANG, L., & JIANG, T. (2009). On the Complexity of Multiple Sequence Alignment. *Journal of Computational Biology*, 1(4), 337–348. <http://doi.org/10.1089/cmb.1994.1.337>]

Outline

- Multiple sequence alignment
- Exact algorithm
- Sum-of-pairs (SP) score
- Carillo-Lipman
- Heuristic approaches

Reading:

- Material based on Chapter 14.6 in book “Algorithms on Strings, Trees and Sequences” by Dan Gusfield

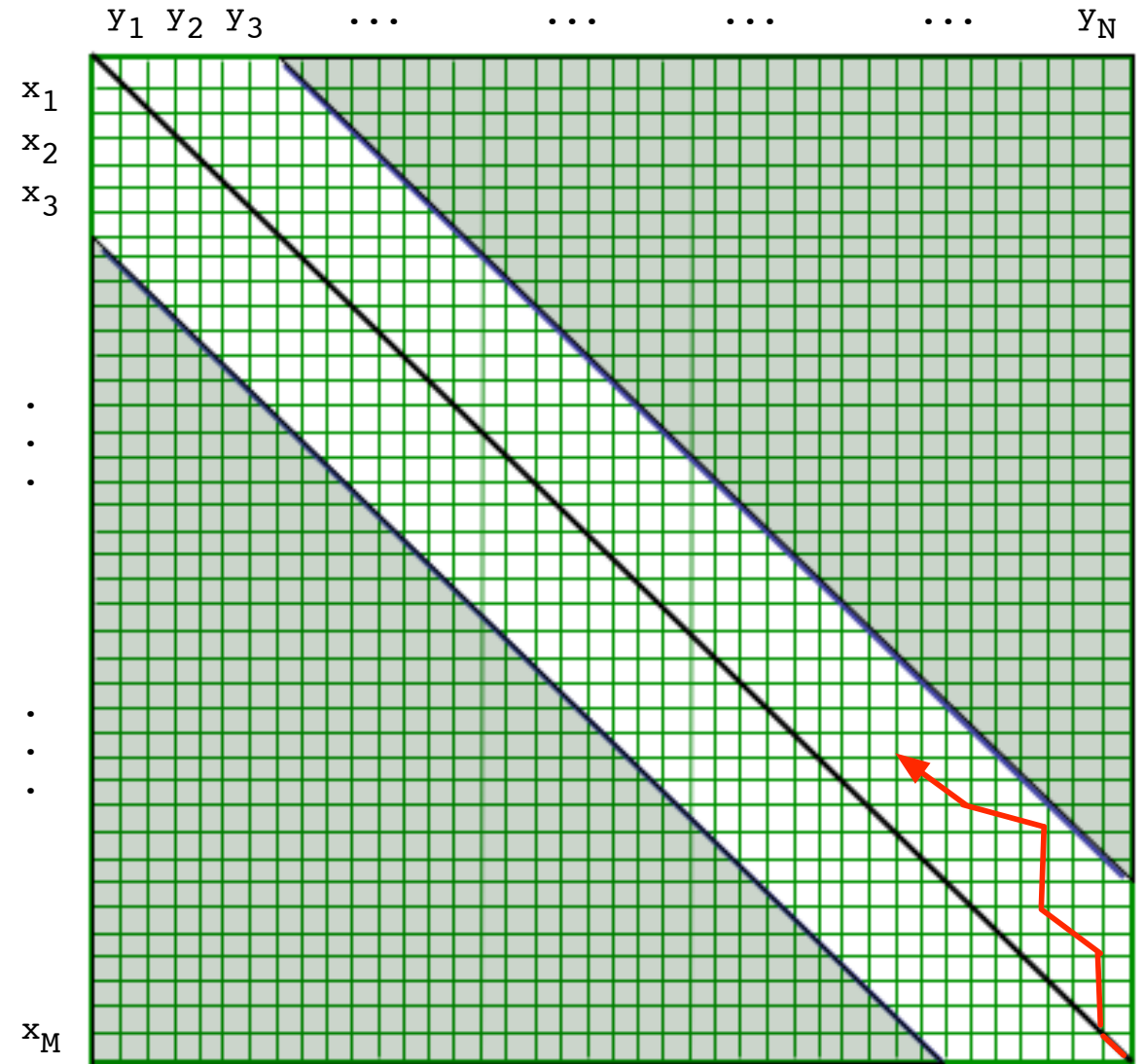
Recall: Banded Alignment

Alignment is a path from source $(0, 0)$ to target (m, n) in edit graph

Constraint path to band of width k around diagonal

Running time: $O(nk)$

Question: Alternative ways of constraining search space?

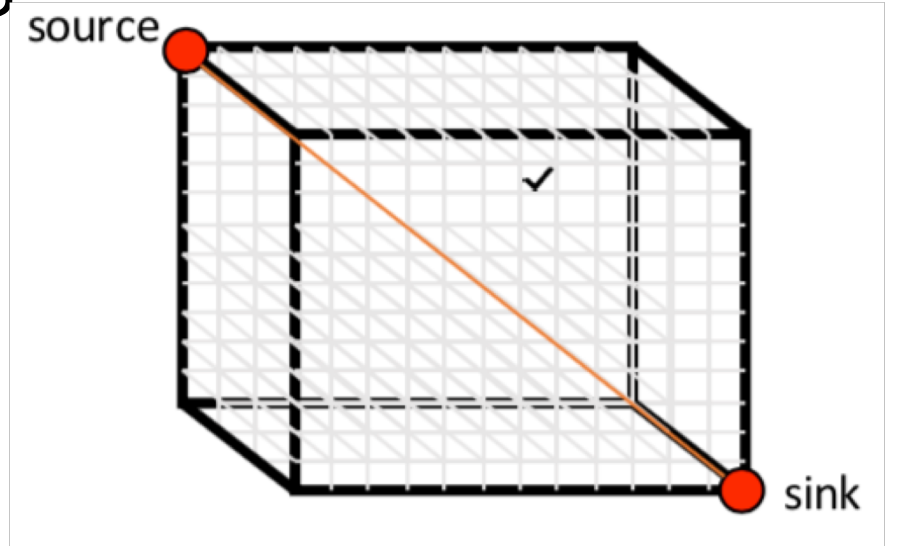


Constrain traceback to band of DP matrix (penalize big gaps)

Forward Dynamic Programming

Banded alignment: constraint path to polyhedron around diagonal

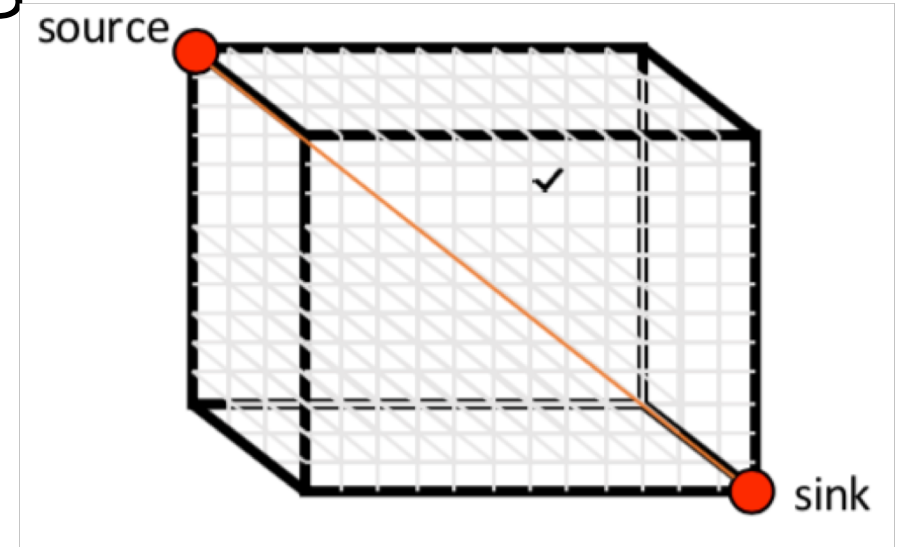
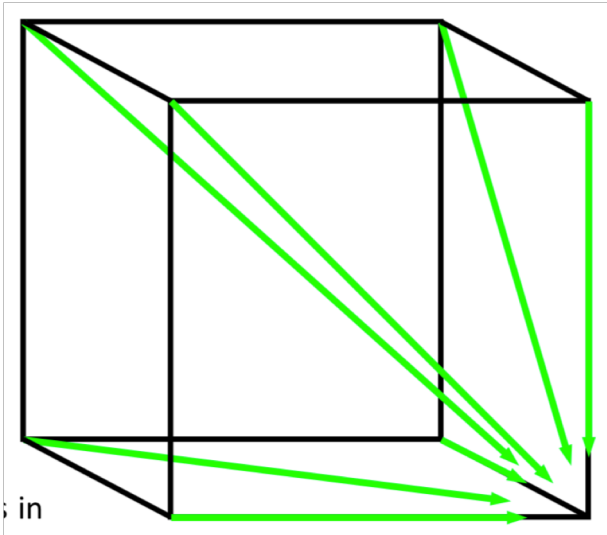
Alternatively: Stop computing when remaining alignment will be suboptimal



Forward Dynamic Programming

Banded alignment: constraint path to polyhedron around diagonal

Alternatively: Stop computing when remaining alignment will be suboptimal



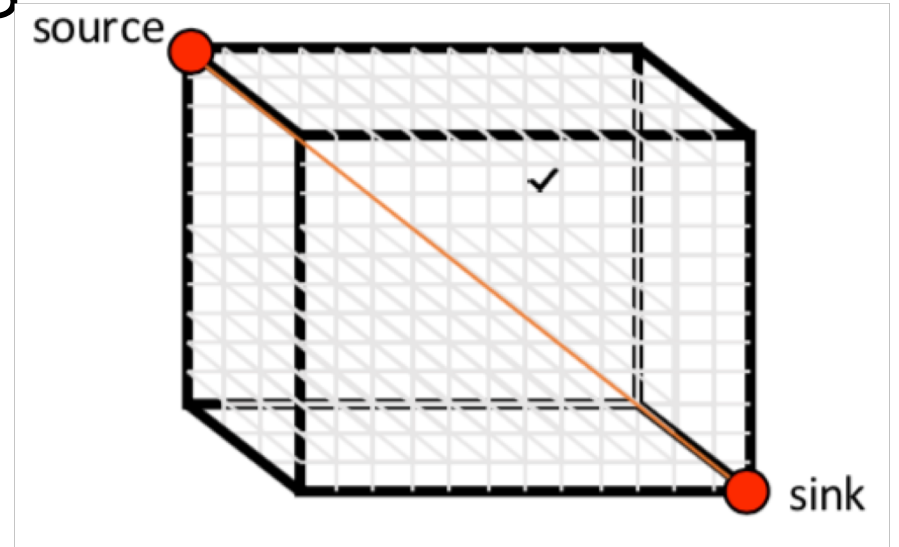
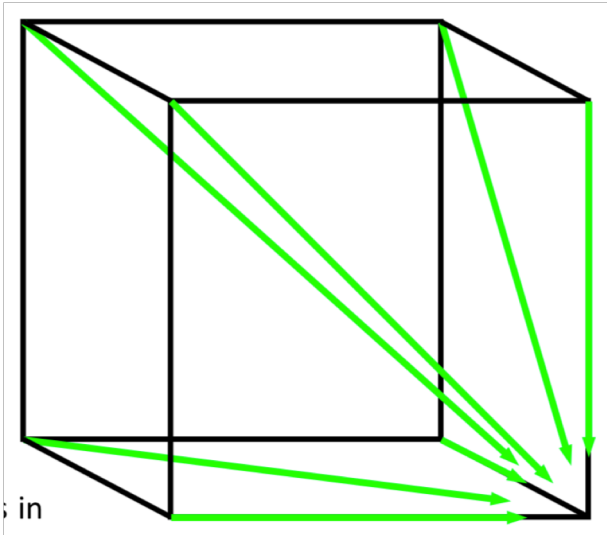
Forward dynamic programming – think of Dijkstra’s algorithm:

- Queue of unvisited vertices
- Maintain $p[i, j, k]$ shortest distance yet found from $(0,0,0)$ to (i, j, k) .
- For each directed edge (i, j, k) to (i', j', k') with cost w , set $p[i', j', k'] = \min\{p[i', j', k'], p[i, j, k] + w\}$

Forward Dynamic Programming

Banded alignment: constraint path to polyhedron around diagonal

Alternatively: Stop computing when remaining alignment will be suboptimal



Forward dynamic programming – think of Dijkstra’s algorithm:

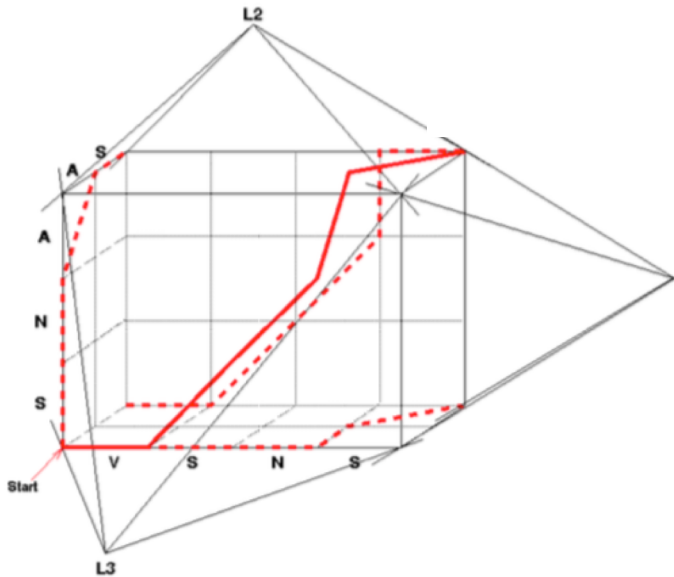
- Queue of unvisited vertices
- Maintain $p[i, j, k]$ shortest distance yet found from $(0,0,0)$ to (i, j, k) .
- For each directed edge (i, j, k) to (i', j', k') with cost w , set $p[i', j', k'] = \min\{p[i', j', k'], p[i, j, k] + w\}$

Question: Can we remove vertices from consideration based on alignment score of prefix?

Alignment Projection and SP-score

Sequences $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ each of length n

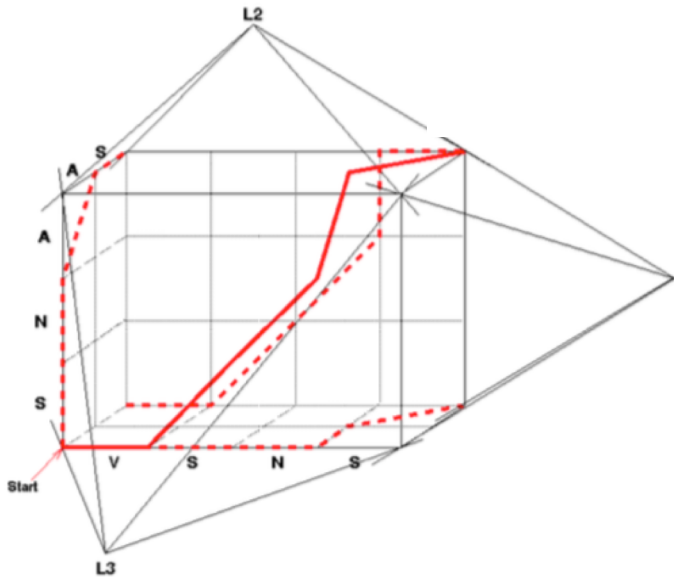
- $D(i, j, k)$ is min SP-cost of aligning $\mathbf{v}_1[1..i], \mathbf{v}_2[1..j], \mathbf{v}_3[1..k]$
- $d_{p,q}(i, j)$ is cost of induced alignment of $\mathbf{v}_p[1..i], \mathbf{v}_q[1..j]$
- $D_{p,q}(i, j)$ is min cost of aligning $\mathbf{v}_p[1..i], \mathbf{v}_q[1..j]$



Alignment Projection and SP-score

Sequences $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ each of length n

- $D(i, j, k)$ is min SP-cost of aligning $\mathbf{v}_1[1..i], \mathbf{v}_2[1..j], \mathbf{v}_3[1..k]$
- $d_{p,q}(i, j)$ is cost of induced alignment of $\mathbf{v}_p[1..i], \mathbf{v}_q[1..j]$
- $D_{p,q}(i, j)$ is min cost of aligning $\mathbf{v}_p[1..i], \mathbf{v}_q[1..j]$

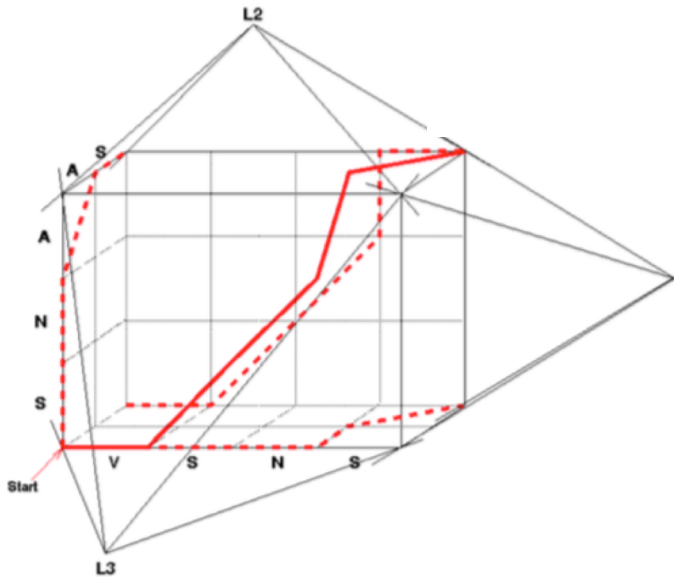


$$d_{p,q}(i, j) \geq D_{p,q}(i, j)$$

Alignment Projection and SP-score

Sequences $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ each of length n

- $D(i, j, k)$ is min SP-cost of aligning $\mathbf{v}_1[1..i], \mathbf{v}_2[1..j], \mathbf{v}_3[1..k]$
- $d_{p,q}(i, j)$ is cost of induced alignment of $\mathbf{v}_p[1..i], \mathbf{v}_q[1..j]$
- $D_{p,q}(i, j)$ is min cost of aligning $\mathbf{v}_p[1..i], \mathbf{v}_q[1..j]$

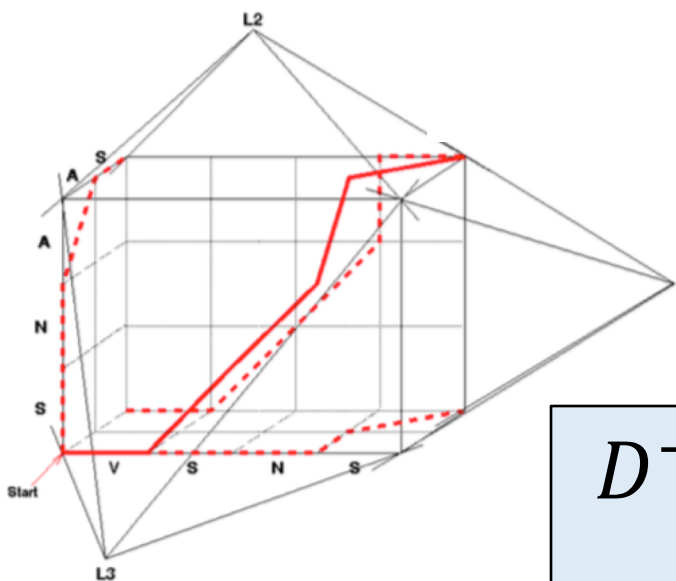


$$d_{p,q}(i, j) \geq D_{p,q}(i, j)$$

$$\begin{aligned} D(i, j, k) &= d_{1,2}(i, j) + d_{1,3}(i, k) + d_{2,3}(j, k) \\ &\geq D_{1,2}(i, j) + D_{1,3}(i, k) + D_{2,3}(j, k) \end{aligned}$$

Carillo-Lipman Method

- $D^+(i, j, k)$ is min SP-cost of alignment of **suffix** $\mathbf{v}_1[i..n], \mathbf{v}_2[j..n], \mathbf{v}_3[k..n]$
- $d_{p,q}^+(i, j)$ is cost of induced alignment of **suffix** $\mathbf{v}_p[i..n], \mathbf{v}_q[j..n]$
- $D_{p,q}^+(i, j)$ is min cost of alignment of **suffix** $\mathbf{v}_p[i..n], \mathbf{v}_q[j..n]$



$$d_{p,q}^+(i, j) \geq D_{p,q}^+(i, j)$$

$$\begin{aligned} D^+(i, j, k) &= d_{1,2}^+(i, j) + d_{1,3}^+(i, k) + d_{2,3}^+(j, k) \quad \square \\ &\geq D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k) \end{aligned}$$

Carillo-Lipman Method

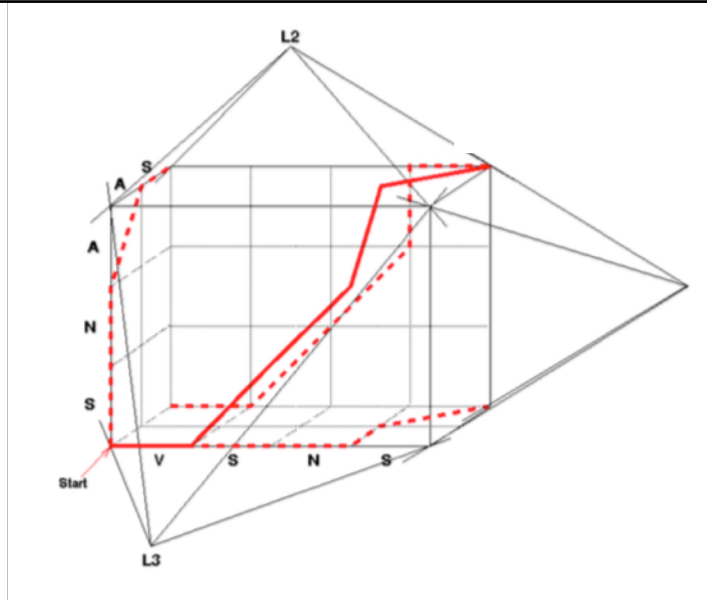
$$D^+(i, j, k) = d_{1,2}^+(i, j) + d_{1,3}^+(i, k) + d_{2,3}^+(j, k) \geq D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k)$$

$$D(i, j, k) + D^+(i, j, k) \geq D(i, j, k) + D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k)$$

Carillo-Lipman Method

$$D^+(i, j, k) = d_{1,2}^+(i, j) + d_{1,3}^+(i, k) + d_{2,3}^+(j, k) \geq D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k)$$

$$D(i, j, k) + D^+(i, j, k) \geq D(i, j, k) + D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k)$$

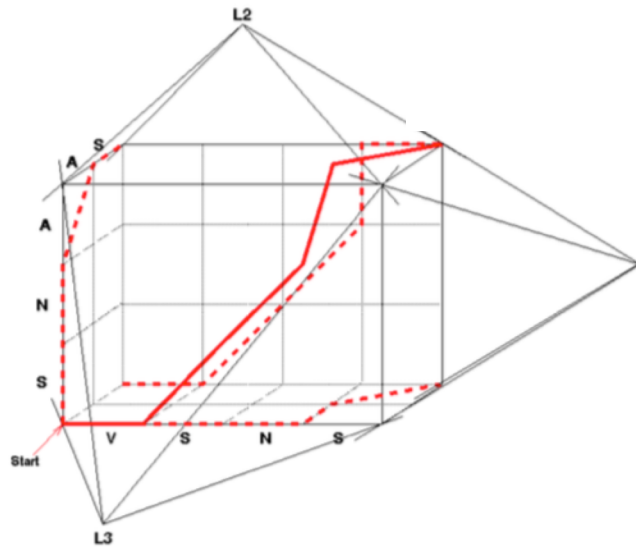


Question: What if we have an alignment with cost z ?

Carillo-Lipman Method

$$D^+(i, j, k) = d_{1,2}^+(i, j) + d_{1,3}^+(i, k) + d_{2,3}^+(j, k) \geq D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k)$$

$$D(i, j, k) + D^+(i, j, k) \geq D(i, j, k) + D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k)$$



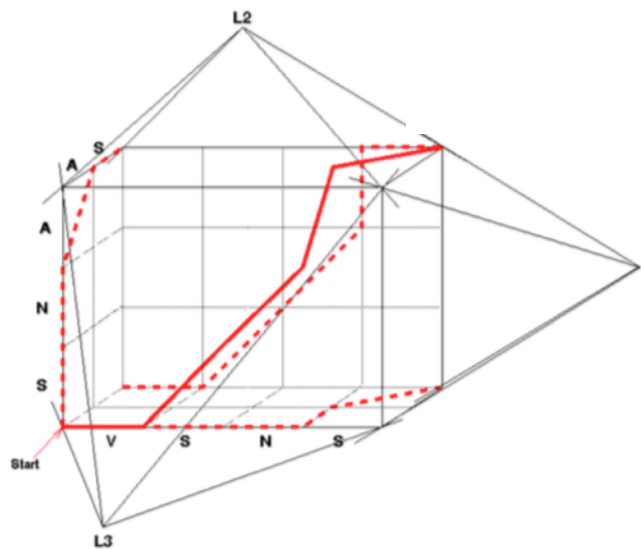
Question: What if we have an alignment with cost z ?

If $z < D(i, j, k) + D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k)$
then (i, j, k) not on optimal path => **Prune!**

Carillo-Lipman Method

$$D^+(i, j, k) = d_{1,2}^+(i, j) + d_{1,3}^+(i, k) + d_{2,3}^+(j, k) \geq D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k)$$

$$D(i, j, k) + D^+(i, j, k) \geq D(i, j, k) + D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k)$$



Question: What if we have an alignment with cost z ?

Question: How to find this alignment?

If $z < D(i, j, k) + D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k)$
then (i, j, k) not on optimal path => **Prune!**

Outline

- Multiple sequence alignment
- Exact algorithm
- Sum-of-pairs (SP) score
- Carillo-Lipman
- Heuristic approaches

Reading:

- Material based on Chapter 14.6 in book “Algorithms on Strings, Trees and Sequences” by Dan Gusfield

Inverse Problem: From Pairwise to Multiple Alignment

v_1	A	C	G	C	T	G	G	-	C
v_2	A	C	G	C	-	-	G	A	G

v_1	A	C	-	G	C	T	G	G	-	C
v_3	G	C	C	G	C	A	-	G	A	G

v_2	A	C	-	G	C	-	G	A	G
v_3	G	C	C	G	C	A	G	A	G

Question: Can we construct a multiple alignment that induces the above three pairwise alignments?

Inverse Problem: From Pairwise to Multiple Alignment

v_1	A	C	G	C	T	G	G	-	C
v_2	A	C	G	C	-	-	G	A	G

v_1	A	C	-	G	C	T	G	G	-	C
v_3	G	C	C	G	C	A	-	G	A	G

v_2	A	C	-	G	C	-	G	A	G
v_3	G	C	C	G	C	A	G	A	G

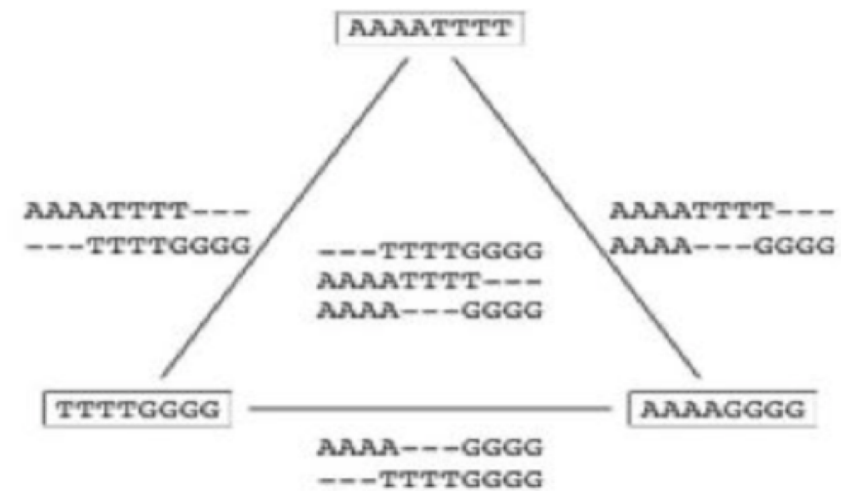
Question: Can we construct a multiple alignment that induces the above three pairwise alignments?

Not always!

Compatibility

Compatible: Pairwise alignments can be combined into multiple alignment

Incompatible: Pairwise alignments *cannot* be combined into multiple alignment



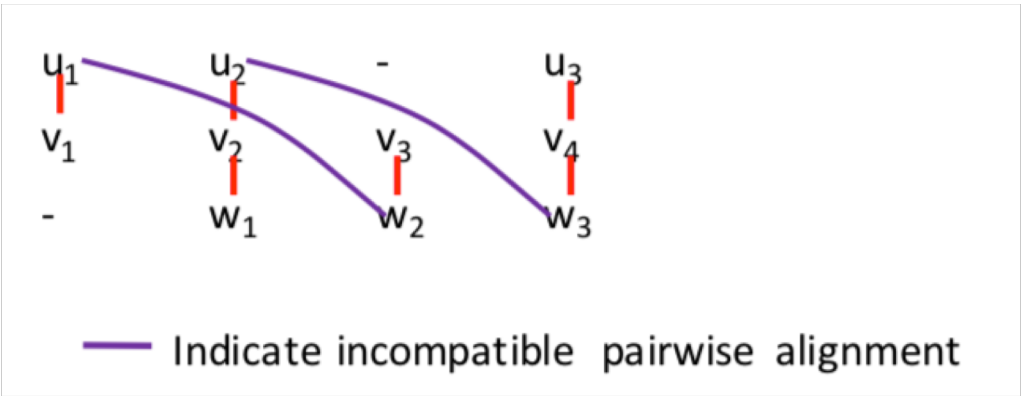
(a) Compatible pairwise alignments



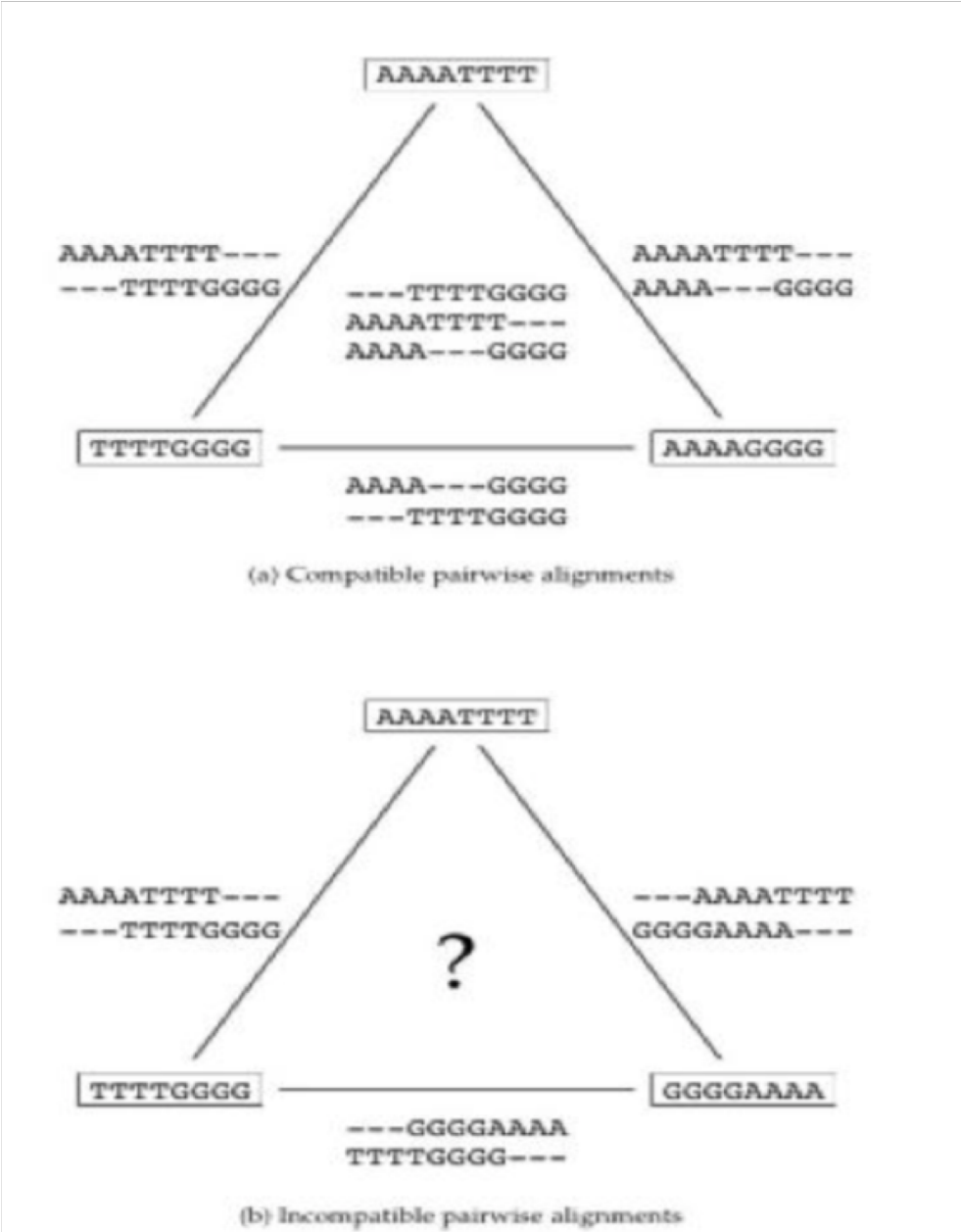
(b) Incompatible pairwise alignments

Compatibility

Compatible: Pairwise alignments can be combined into multiple alignment



Incompatible: Pairwise alignments *cannot* be combined into multiple alignment



From Compatible Pairwise to Multiple Alignment

Optimal multiple alignment



Easy

Pairwise alignments between *all* pairs of sequences, but they are *not* necessarily optimal

(Sub)optimal multiple alignment



Challenging

Good (or optimal) *compatible* pairwise alignments between all sequences

From Compatible Pairwise to Multiple Alignment

Iterative/progressive
multiple sequence alignment:
Merge pairwise alignments

(Sub)optimal multiple alignment

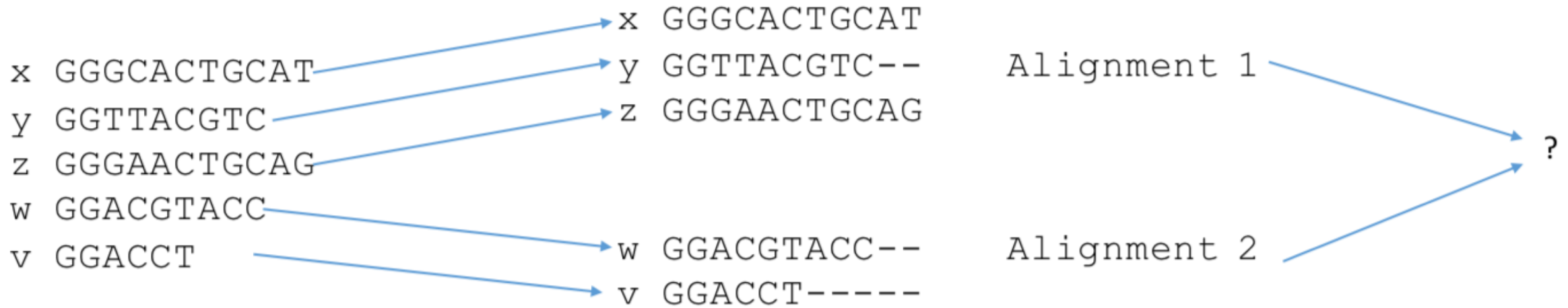


Challenging

Good (or optimal) *compatible*
pairwise alignments between all
sequences

Heuristic: Iterative/Progressive Alignment

Iteratively add strings (or alignments) to existing alignment(s).



Issues:

1. How to merge alignments?
2. What order to use in merging strings/alignments?

Heuristic Approach: Merge Pairwise Alignments

```
x GGGCACTGCAT
y GGTTACGTC-- Alignment 1
z GGGAACTGCAG

w GGACGTACC-- Alignment 2
v GGACCT-----
```

Question:
Can we align two
alignments?

Need a way to summarize
an alignment and score
merged alignments

Profile Representation of Multiple Alignment

	-	A	G	G	C	T	A	T	C	A	C	C	T	G
T	A	G	-	C	T	A	C	C	A	-	-	-	-	G
C	A	G	-	C	T	A	C	C	A	-	-	-	-	G
C	A	G	-	C	T	A	T	C	A	C	-	G	G	G
C	A	G	-	C	T	A	T	C	G	C	-	G	G	G
A		1				1			.8					
C	.6			1		.4	1		.6	.2				
G			1	.2					.2			.4	1	
T	.2				1	.6						.2		
-	.2		.8						.4	.8	.4			

A **profile** $P = [p_{i,j}]$ is a $(|\Sigma| + 1) \times l$ matrix, where $p_{i,j}$ is the frequency of i -th letter in j -th position of alignment

Profile Representation of Multiple Alignment

We know how to align sequence against sequence

-	A	G	G	C	T	A	T	C	A	C	C	T	G
T	A	G	-	C	T	A	C	C	A	-	-	-	G
C	A	G	-	C	T	A	C	C	A	-	-	-	G
C	A	G	-	C	T	A	T	C	A	C	-	G	G
C	A	G	-	C	T	A	T	C	G	C	-	G	G

A		1				1			.8				
C	.6			1			.4	1	.6	.2			
G		1	.2						.2			.4	1
T	.2				1	.6						.2	
-	.2		.8						.4	.8	.4		

Question: Can we align sequence against profile?

Question: Can we align profile against profile?

Aligning String to Profile

A **profile** $P = [p_{i,j}]$ is a $(|\Sigma| + 1) \times n$ matrix, where $p_{i,j}$ is the frequency of i -th letter in j -th position of alignment

Given: Sequences $\mathbf{v} = v_1, \dots, v_m$ and profile P with n columns

- $s[i, j]$ is optimal alignment of v_1, \dots, v_m and first j columns of P
- $\delta(x, y)$ is score for aligning characters x and y
- $\tau(x, j)$ is score for aligning character x and column j of P

Aligning String to Profile

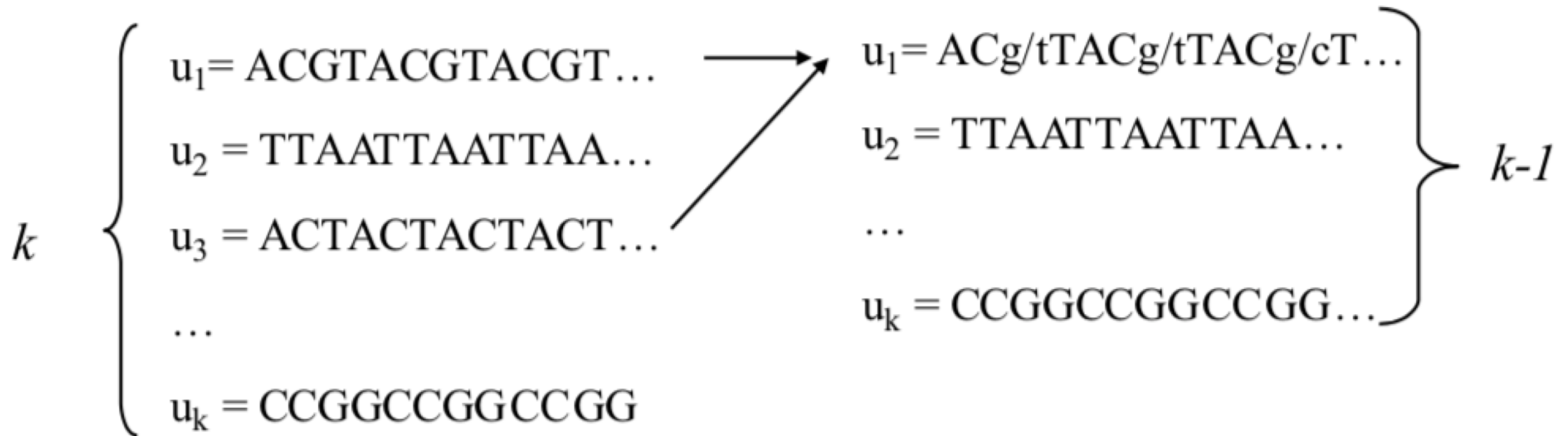
$$\tau(x, j) = \sum_{y \in \Sigma \cup \{-\}} p_{y,j} \cdot \delta(x, y)$$

$$s[i, j] = \max \begin{cases} 0, & \text{if } i = 0 \text{ and } j = 0, \\ s[i - 1, j] + \delta(v_i, -), & \text{if } i > 0, \quad \text{Insert space in profile} \\ s[i, j - 1] + \tau(-, j), & \text{if } j > 0, \quad \text{Insert space in string} \\ s[i - 1, j - 1] + \tau(v_i, j), & \text{if } i > 0 \text{ and } j > 0. \end{cases}$$

- $s[i, j]$ is optimal alignment of v_1, \dots, v_m and first j columns of P
- $\delta(x, y)$ is score for aligning characters x and y
- $\tau(x, j)$ is score for aligning character x and column j of P

Progressive Multiple Alignment: Greedy Algorithm

Choose most similar pair among k input strings, combine into a profile. This reduces the original problem to alignment of $k-1$ sequences to a profile. Repeat.



Example

Score of +1 for matches, -1 otherwise.

s2 GTCTGA
s4 GTCAGC (score = 2)

s1 GAT-TCA
s2 G-TCTGA (score = 1)

s1 GAT-TCA
s3 GATAT-T (score = 1)

s1 GATTCA--
s4 G-T-CAGC (score = 0)

s2 G-TCTGA
s3 GATAT-T (score = -1)

s3 GAT-ATT
s4 G-TCAGC (score = -1)

Example

Score of +1 for matches, -1 otherwise.

s2 GTCTGA
s4 GTCAGC (score = 2)

s1 GAT-TCA
s2 G-TCTGA (score = 1)

s1 GAT-TCA
s3 GATAT-T (score = 1)

s1 GATTCA--
s4 G-T-CAGC (score = 0)

s2 G-TCTGA
s3 GATAT-T (score = -1)

s3 GAT-ATT
s4 G-TCAGC (score = -1)

Question: Any theoretical guarantees on optimality?

No guarantees!

Summary

- Multiple sequence alignment
- Exact algorithm
- Sum-of-pairs (SP) score
- Carillo-Lipman
- Heuristic approaches

Homework 2 due Oct. 5 by 11:59pm

Midterm on Oct. 10, 7-9pm, 1310 DCL

Reading:

- Material based on Chapter 14.6 in book “Algorithms on Strings, Trees and Sequences” by Dan Gusfield