Multiple Sequence Alignment for Morphology Induction

Tzvetan Tchoukalov
Christian Monson
Brian Roark

Multiple Sequence Alignment in Biology

prokaryote16S rRNA

Columns 1623-1703 out of 7683

```
---T--C---C-G--------------C----T-G---A-TA-G---AT---G-G-----G-CTC-GCG--T-CTG--A
------G---T-G--------------G----T-A---T-AA-G---AT---G-G-----A-CCC-GCG--T-TGG--A
------G---T-G--------------G----T-A---T-AG-G---AT---G-G-----A-CCC-GCG--T-CTG--A
------G--GC-G--------------G----T-G---A-AG-G---AT---G-A-----G-CCC-GCG--G-CCT--A
------C---C-G--------------G----T-A---G-AC-G---AT---G-G-----G-GAT-GCG--T-TCC--A
---T--C---C-G--------------C----T-T---T-GA-G---AT---G-G-----C-CTC-GCG--T-CCG--A
```

Multiple Sequence Alignment in Biology

prokaryote16S rRNA

Columns 1623-1703 out of 7683

```
---T--C---C-G--------------C----T-G---A-TA-G---AT---G-G-----G-CTC-GCG--T-CTG--A
------G---T-G--------------G----T-A---T-AA-G---AT---G-G-----A-CCC-GCG--T-TGG--A
------G---T-G--------------G----T-A---T-AG-G---AT---G-G-----A-CCC-GCG--T-CTG--A
------G--GC-G--------------G----T-G---A-AG-G---AT---G-A-----G-CCC-GCG--G-CCT--A
------C---C-G--------------G----T-A---G-AC-G---AT---G-G-----G-GAT-GCG--T-TCC--A
---T--C---C-G--------------C----T-T---T-GA-G---AT---G-G-----C-CTC-GCG--T-CCG--A
```

Sequences of symbols

Sequences are **related**

e.g. serve same function in different organisms

Why?

To identify conserved regions

To identify regions with similar physical structure

Multiple Sequence Alignment for Morphology

English Verbs

```
d - a n c - e s
d - a n c - e d
d - a n c - e
d - a n c i n g
r - u n n i n g
j - u m p i n g
j - u m p - e d
j - u m p - s
j - u m p - - -
l a u g h i n g
```

Sequences of symbols

Sequences are **related**

e.g. serve same function in different **words**

Why?

To learn morphological structure

Language Vs. Biology

Differences

	# of Sequences to Align	Length of Sequences	Symbol = Meaning
Language	Millions	10's	No
Biology	10's	Millions	Yes

Similarities

Both involve sequences

Size of Alphabet (less than 100)

What We Did

1. **Progressive alignment**
   To build a profile
2. **Leave-one-out realignment**
3. **Align words to the profile**
4. **Segment words**
   Based on alignment

Step 1) Progressive Alignment

A Profile

```
1 2 3 4 5 6 7 8
d - a n c - e s
d - a n c - e d
d - a n c - e
d - a n c i n g
r - u n n i n g
j - u m p i n g
j - u m p - e d
j - u m p - s
j - u m p - - -
l a u g h i n g
```

Step 1) Progressive Alignment

A Profile

1	2	3	4	5	6	7	8
d	-	a	n	c	-	e	s
d	-	a	n	c	-	e	d
d	-	a	n	c	-	e	
d	-	a	n	c	i	n	g
r	-	u	n	n	i	n	g
j	-	u	m	p	i	n	g
j	-	u	m	p	-	e	d
j	-	u	m	p	-	s	
j	-	u	m	p	-	-	-
l	a	u	g	h	i	n	g

	1	2	3	4	5	6	7	8
a	1	2	5	1	1	1	5	1
c	1	1	1	1	5	1	1	1
d	5	1	1	1	1	1	1	3
e	1	1	1	1	1	1	1	1
g	1	1	1	2	1	1	1	5
h	1	1	1	1	2	1	1	1
i	1	1	1	1	1	5	1	1
j	5	1	1	1	1	1	1	1
l	2	1	1	1	1	1	1	1
m	1	1	1	5	1	1	1	1
n	1	1	1	6	2	1	5	1
p	1	1	1	1	5	1	1	1
r	2	1	1	1	1	1	1	1
s	1	1	1	1	1	1	2	2
u	1	1	7	1	1	1	1	1
gap	1	A	1	1	1	7	2	4

Column Distributions

Step 1) Progressive Alignment

A Profile

1	2	3	4	5	6	7	8
d	-	a	n	c	-	e	s
d	-	a	n	c	-	e	d
d	-	a	n	c	-	e	
d	-	a	n	c	i	n	g
r	-	u	n	n	i	n	g
j	-	u	m	p	i	n	g
j	-	u	m	p	-	e	d
j	-	u	m	p	-	s	
j	-	u	m	p	-	-	-
l	a	u	g	h	i	n	g

	1	2	3	4	5	6	7	8
a	1	2	5	1	1	1	1	1
c	1	1	1	1	5	1	1	1
d	5	1	1	1	1	1	1	3
e	1	1	1	1	1	1	5	1
g	1	1	1	2	1	1	1	5
h	1	1	1	1	2	1	1	1
i	1	1	1	1	1	5	1	1
j	5	1	1	1	1	1	1	1
l	2	1	1	1	1	1	1	1
m	1	1	1	5	1	1	1	1
n	1	1	1	6	2	1	5	1
p	1	1	1	1	5	1	1	1
r	2	1	1	1	1	1	1	1
s	1	1	1	1	1	1	2	2
u	1	1	7	1	1	1	1	1
gap	1	A	1	1	1	7	2	4

Laplace Smoothing

Step 1) Progressive Alignment

A Profile

1	2	3	4	5	6	7	8
d	–	a	n	c	–	e	s
d	–	a	n	c	–	e	d
d	–	a	n	c	–	e	–
d	–	a	n	c	i	n	g
r	–	u	n	n	i	n	g
j	–	u	m	p	i	n	g
j	–	u	m	p	–	e	d
j	–	u	m	p	–	s	–
j	–	u	m	p	–	–	–
l	a	u	g	h	i	n	g

1. **Sort** words by frequency
2. **Using Levenshtein distance**
   In first n=1000 words
   Find most similar pair of words, W_1 and W_2
3. **Align** W_1 and W_2 (using Levenshtein)
   This is our Profile
4. **For *i*=3 to M=5000, 10000, …**
   Find word W_i, most similar to W_j, $j<i$
   Align W_i to profile

Step 1) Progressive Alignment

A Profile

```
1 2 3 4 5 6
d a n c e s
d a n c e d
d a n c e -

d a n c i n g
```

New W_i

Align W_i to Profile

Dynamic Programming

		d d d	a a a	n n n	c c c	e e e	d s -
	0.0						
d							
a							
n							
c							
i							
n							
g							

The Goal

Align W_i *to Profile*

Dynamic Programming

		d d d	a a a	n n n	c c c	e e e	d s -
	0.0 ←	4.4 ←	5.9 ←	7.4 ←	8.9 ←	10.4 ←	11.9
d	4.4						
a	5.9						
n	7.4						
c	8.8						
i	10.4						
n	11.9						
g	13.4						

Align W_i *to Profile*

Dynamic Programming

		d d d	a a a	n n n	c c c	e e e	d s -
	0.0 ←	4.4 ←	5.9 ←	7.4 ←	8.9 ←	10.4 ←	11.9
d	4.4 ↑	**1.6**					
a	5.9 ↑						
n	7.4 ↑						
c	8.8 ↑						
i	10.4 ↑						
n	11.9 ↑						
g	13.4						

Match

cost = -log P(character)

Align W_i *to Profile*

Dynamic Programming

		d d d	a a a	n n n	c c c	e e e	d s -
	0.0 ←	4.4 ←	5.9 ←	7.4 ←	8.9 ←	10.4 ←	11.9
d	4.4 ←	**8.8**					
a	5.9						
n	7.4						
c	8.8						
i	10.4						
n	11.9						
g	13.4						

Insert gap into new word

cost = -log P(gap)

Align W_i *to Profile*

Dynamic Programming

		d d d	a a a	n n n	c c c	e e e	d s -
	0.0 ←	4.4 ←	5.9 ←	7.4 ←	8.9 ←	10.4 ←	11.9
d	4.4 ↑	**8.8** ↑					
a	5.9 ↑						
n	7.4 ↑						
c	8.8 ↑						
i	10.4 ↑						
n	11.9 ↑						
g	13.4						

Insert gap into alignment profile

cost = -log P(unattested)

Align W_i *to Profile*

Dynamic Programming

		d d d	a a a	n n n	c c c	e e e	d s -
	0.0 ←	4.4 ←	5.9 ←	7.4 ←	8.9 ←	10.4 ←	11.9
d	4.4 ↑	1.6					
a	5.9 ↑						
n	7.4 ↑						
c	8.8 ↑						
i	10.4 ↑						
n	11.9 ↑						
g	13.4						

Match

cost = -log P(character)

Align W_i *to Profile*

Dynamic Programming

		d d d	a a a	n n n	c c c	e e e	d s -
	0.0	4.4	5.9	7.4	8.9	10.4	11.9
d	4.4	1.6	6.0	7.5	9.0	10.5	12.0
a	5.9	3.1	3.1	7.6	9.1	10.6	12.1
n	7.4	4.6	4.6	4.7	9.1	10.6	12.1
c	8.8	6.1	6.1	6.2	6.2	10.7	12.2
i	10.4	7.6	7.6	7.7	7.7	9.2	12.9
n	11.9	9.1	9.1	9.2	9.2	10.7	12.1
g	13.4	10.6	10.6	10.7	10.7	12.2	13.6

Align W_i *to Profile*

Dynamic Programming

		d d d	a a a	n n n	c c c	e e e	d s -
	0.0	4.4	5.9	7.4	8.9	10.4	11.9
d	4.4	1.6	6.0	7.5	9.0	10.5	12.0
a	5.9	3.1	3.1	7.6	9.1	10.6	12.1
n	7.4	4.6	4.6	4.7	9.1	10.6	12.1
c	8.8	6.1	6.1	6.2	6.2	10.7	12.2
i	10.4	7.6	7.6	7.7	7.7	9.2	12.9
n	11.9	9.1	9.1	9.2	9.2	10.7	12.1
g	13.4	10.6	10.6	10.7	10.7	12.2	13.6

Align W_i *to Profile*

Dynamic Programming

		d	a	n	c	e	d
		d	a	n	c	e	s
		d	a	n	c	e	-
	0.0	4.4	5.9	7.4	8		
d	4.4	1.6	6.0	7.5	9		
a	5.9	3.1	3.1	7.6	9		
n	7.4	4.6	4.6	4.7	9		
c	8.8	6.1	6.1	6.2	6.2	10.7	12.2
i	10.4	7.6	7.6	7.7	7.7	9.2	12.9
n	11.9	9.1	9.1	9.2	9.2	10.7	12.1
g	13.4	10.6	10.6	10.7	10.7	12.2	13.6

```
1 2 3 4 5 6 7
d a n c - e s
d a n c - e d
d a n c - e -
d a n c i n g
```

Steps 2 & 3

Step 2) Leave-one-out realignment

Improves the greedy alignment

Step 3) Align remaining words

Profile is frozen

Gaps inserted in word only

Step 4) Segmentation

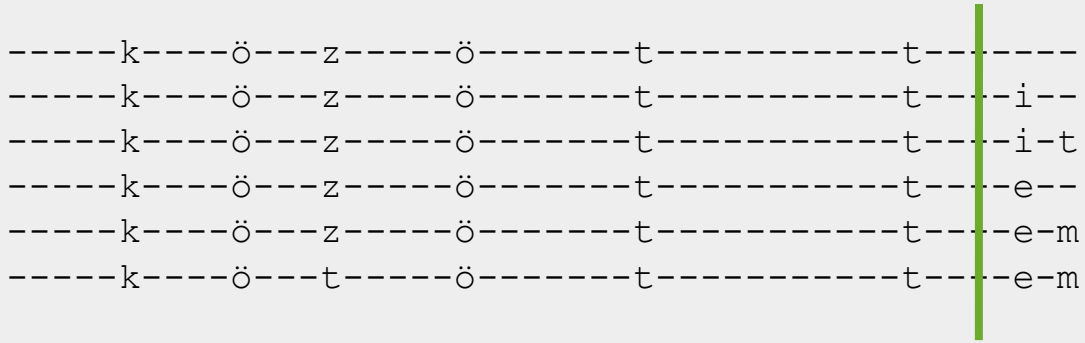
6 Hungarian words from a real alignment

```
-----k----ö---z-----ö-------t-----------t-------
-----k----ö---z-----ö-------t-----------t----i--
-----k----ö---z-----ö-------t-----------t----i-t
-----k----ö---z-----ö-------t-----------t----e--
-----k----ö---z-----ö-------t-----------t----e-m
-----k----ö---t-----ö-------t-----------t----e-m
```

Where are the morpheme boundaries?

Step 4) Segmentation

6 Hungarian words from a real alignment



Where are the morpheme boundaries?

Gaps do **not** correspond to morpheme boundaries

Biologists **don't** segment!!

Step 4) Segmentation

Mimic the ParaMor-Morfessor Union!

Take ParaMor-Morfessor Union as THE TRUTH

Greedy search

For each column, c, in profile

Segment all words at c

Score against Union system

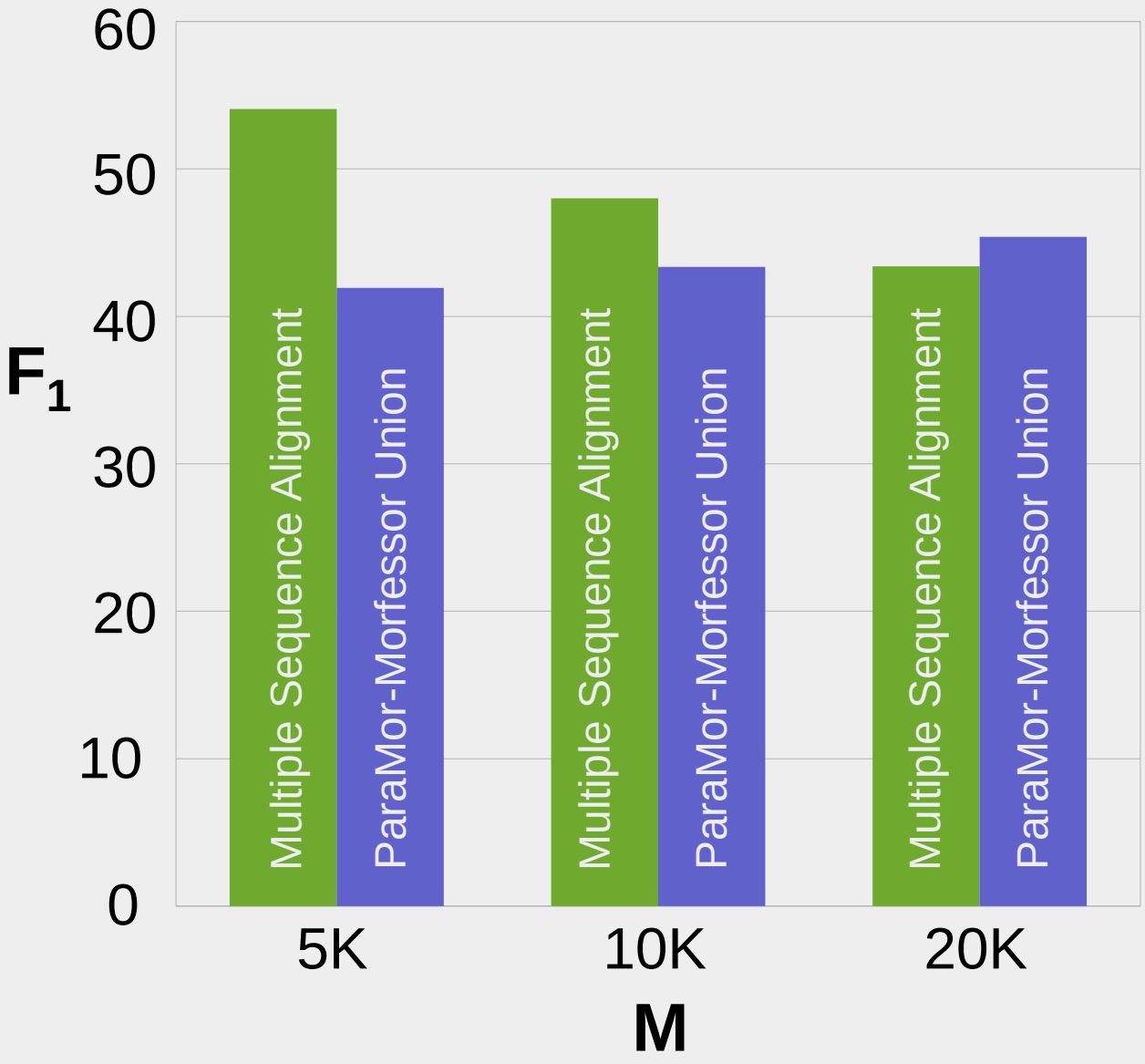
Keep the best scoring segmentation column

Repeat until no column improves score

Turkish Linguistic Competition Results

AUTHOR	METHOD	PREC.	REC.	F1
Monson et al.	ParaMor-Morfessor Mimic	48.07%	60.39%	53.53%
Monson et al.	ParaMor-Morfessor Union	47.25%	60.01%	52.88%
Monson et al.	ParaMorMimic	49.54%	54.77%	52.02%
Lavallée & Langlais	RALI-COF	48.43%	44.54%	46.40%
-	Morfessor CatMAP	79.38%	31.88%	45.49%
Spiegler et al.	PROMODES 2	35.36%	58.70%	44.14%
Spiegler et al.	PROMODES	32.22%	66.42%	43.39%
Bernhard	MorphoNet	61.75%	30.90%	41.19%
Can & Manandhar	2	41.39%	38.13%	39.70%
Spiegler et al.	PROMODES committee	55.30%	28.35%	37.48%
Golénia et al.	UNGRADE	46.67%	30.16%	36.64%
Tchoukalov et al.	**MetaMorph**	**39.14%**	**29.45%**	**33.61%**
Virpioja & Kohonen	Allomorfessor	85.89%	19.53%	31.82%
-	Morfessor Baseline	89.68%	17.78%	29.67%
Lavallée & Langlais	RALI-ANA	69.52%	12.85%	21.69%
-	letters	8.66%	99.13%	15.93%
Can & Manandhar	1	73.03%	8.89%	15.86%

Performance Before Profile is Frozen



Next Steps

Build many smaller alignments

- Focus on closely related words

Too many parameters

- Tie column parameters by region

New Segmentation algorithm

- Directly map gaps to morpheme boundaries
- This is an unsolved Problem

ευχαριστώ

(Thank You!)