

Sample Complexity of Algorithm Configuration for Sequence Alignment

Travis Dick

Nina Balcan



Dan DeBlasio



Carl Kingsford



Tuomas Sandholm



Ellen Vitercik



Sequence alignment

Goal: Line up pairs of strings (*DNA, RNA, protein, ...*)

Uncover functional, structural, or evolutionary relationships

$S_1 =$ GRTCPKPDDLPFSTVVPLKTFYEPGEEITYSCKPGYVSRGGMRKFICPLTGLWPINTLKCTP
 $S_2 =$ EVKCPFPSRPDNGFVNYPKPTLYYKDKATFGCHDGYSLDGPEEIECTKLGNSAMPSCKA

GRTCP---KPDDLPFSTVVPLKTFYEPGEEITYSCKPGYVSRGGMRKFICPLTGLWPINTLKCTP
EVKCPFPSRPDN-GFVNYPKPTLYYK-DKATFGCHDGY-SLDGPEEIECTKLGNS-AMPSCKA

Sequence alignment algorithms

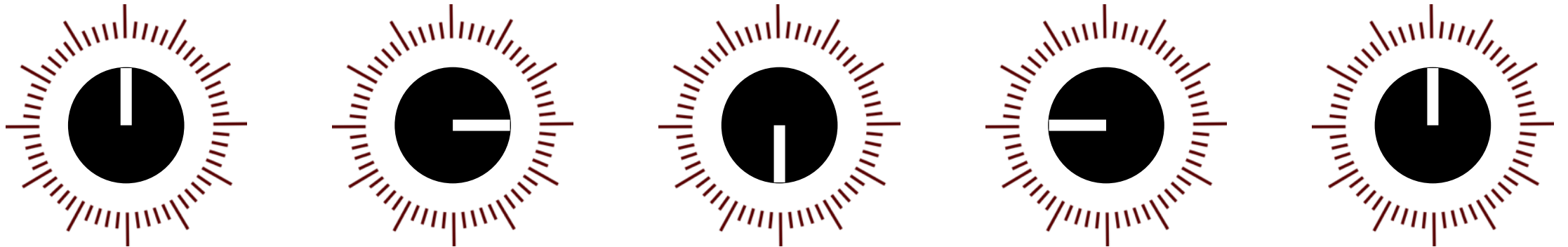
Typically optimize for alignment *features*:

Number of matching characters, number of gaps, ...

[Needleman and Wunsch '70; Gotoh '82]

Standard algos solve for alignment maximizing weighted sum

How to tune the feature weights?



Sequence alignment algorithms

Can sometimes access **ground-truth alignment**

Requires extensive manual alignments

Given set of application's "typical" alignment problems,
together with ground-truth alignments,
can we **learn** parameters that recover ground truth?



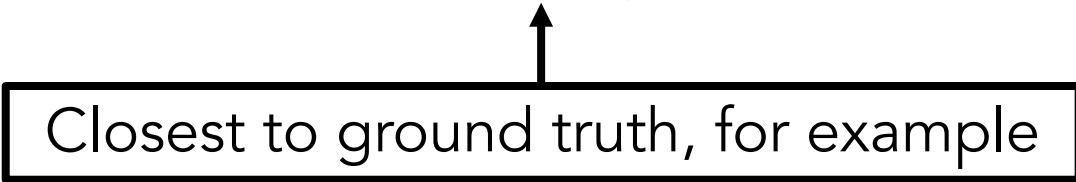
Model

1. Fix a parameterized alignment optimization function
2. Receive sample problems from unknown distribution

Sequence S_1 Sequence S_m
Sequence S'_1 Sequence S'_m
★ Alignment ★ Alignment

3. Find parameter values with best performance over samples

Closest to ground truth, for example



Model

1. Fix a parameterized alignment optimization function
2. Receive sample problems from unknown distribution

Sequence S_1		Sequence S_m
Sequence S'_1	...	Sequence S'_m
★ Alignment		★ Alignment

3. Find parameter values with best performance over samples

Model studied from empirical perspective

Kim and Kececioglu '07; Xu, Hutter, Hoos, Leyton-Brown '08; Dai, Khalil, Zhang, Dilkina, Song '17 ...

Model

1. Fix a parameterized alignment optimization function
2. Receive sample problems from unknown distribution

Sequence S_1		Sequence S_m
Sequence S'_1	...	Sequence S'_m
★ Alignment		★ Alignment

3. Find parameter values with best performance over samples

Model studied from theoretical perspective

Gupta and Roughgarden '16; Kleinberg, Leyton-Brown, Lucier '17; Weisz, György, Szepesvári '18 ...

Questions

Focus of this talk:

Will those parameters have high performance in expectation?

Sequence S_1 ✓
Sequence S'_1 ✓ ...
★ Alignment

Sequence S_m ✓
Sequence S'_m ✓
★ Alignment

Sequence S ?
Sequence S' ?

Focus of prior work [e.g., Kim and Kececioglu '07]:

Algorithmically, how to find good parameters over training set

Model

\mathcal{D} : Distribution over sequence pairs (S, S')

\mathbb{R}^d : Set of parameters

For any sequence pair (S, S') :

$u_{\rho}(S, S')$ = utility of using params $\rho \in \mathbb{R}^d$ to align S, S'

Similarity between algorithm's output & ground truth

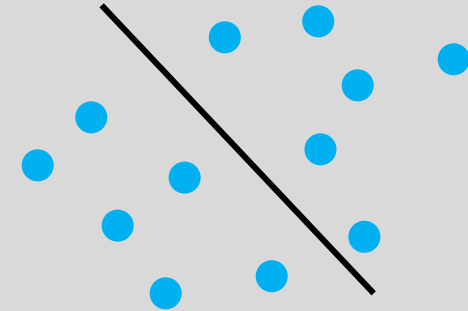
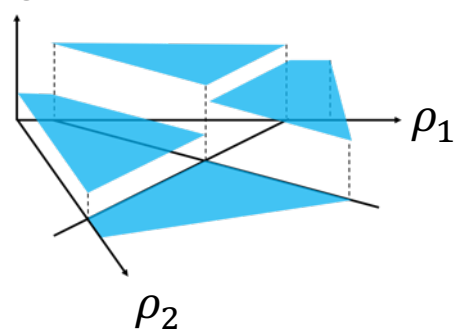
Generalization: Given samples $(S_1, S'_1), \dots, (S_m, S'_m) \sim \mathcal{D}$,

for any $\rho \in \mathbb{R}^d$, $\left| \frac{1}{m} \sum_{i=1}^m u_{\rho}(S_i, S'_i) - \mathbb{E}_{(S, S') \sim \mathcal{D}} [u_{\rho}(S, S')] \right| \leq ?$

Primary challenge:

Algorithmic performance is volatile function of parameters

Similarity to ground truth



For well-understood functions in machine learning:

Close connection between function parameters and value

Outline

1. Pairwise sequence alignment algorithms
2. Sample complexity for pairwise alignment
3. Multiple-sequence alignment algorithms
4. Sample complexity for multiple-sequence alignments
5. Additional applications

Pairwise sequence alignment

Input: Two sequences $S, S' \in \Sigma^n$

Alignment: Sequences $\tau, \tau' \in (\Sigma \cup \{-\})^*$ such that:

Deleting "-" yields S from τ and S' from τ'

$S = A C T G$
 $S' = G T C A$

Gap
↓
 $\tau = A \quad \underline{- \quad -} \quad C \quad T \quad G$
 $\tau' = - \quad G \quad T \quad C \quad A \quad -$
↑ ↑ ↑
 Match Mismatch
Insertion/deletion (*indel*)

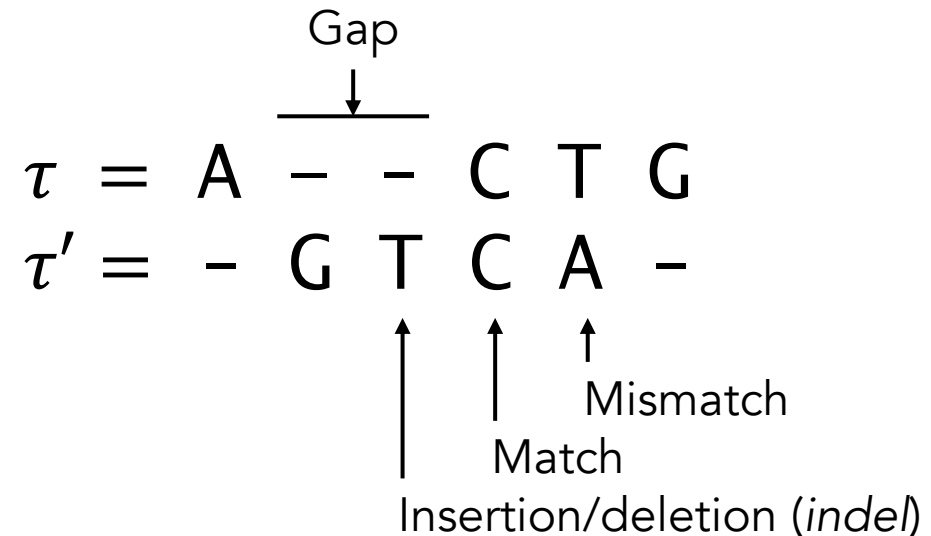
Pairwise sequence alignment algorithms

Standard algorithm with parameters $\rho_1, \rho_2, \rho_3 \geq 0$:

Use dynamic programming to find alignment A maximizing:

$$(\# \text{ matches}) - \rho_1 \cdot (\# \text{ mismatches}) - \rho_2 \cdot (\# \text{ indels}) - \rho_3 \cdot (\# \text{ gaps})$$

$S = A C T G$
 $S' = G T C A$



Pairwise sequence alignment algorithms

More generally, given parameters $\boldsymbol{\rho} \in \mathbb{R}^d$:

Use dynamic programming to find alignment A maximizing:

$$\rho_1 \cdot f_1(A) + \dots + \rho_d \cdot f_d(A)$$

$f_1(A), \dots, f_d(A)$ features of alignment A (e.g., # matches, ...)

Pairwise sequence alignment algorithms

-GRTCPKPDDLPFSTVVP-LKTFYEPGEEITYSCKPGYVSRGGMRKFICPLTGLWPINTLKCTP
E-VKCPFPSRPDNGFVNYPKPTLYYKDKATFGCHDGYSLDGP-EEIECTKLGNEWSAMPSC-KA

Ground-truth alignment

Pairwise sequence alignment algorithms

-G**RTCP**KPDDLPFSTVVP-LKTFYE**PG**EEITYSCKPGYVSRGGMRKFICPLTGLWPINTLKCTP
E-V**KCP**FPSRPDNGFVNYP**AKPTLYYK**DKATFGCHDGYSLDGP-EEIECTKLG**NS**AMPSC-**KA**

Ground-truth alignment

G**RTCP**---KPDDLPFSTVV**PLKTFYE**PGEEITYSCKPGYVSRGGMRKFICPLTGLWPINTLKCTP
E**VKCP**FPSRPDN-GFVNYP**AKPTLYYK**-DKATFGCHDGY-SLDGP**EEIECTKLG**NS-AMPSC**KA**

Alignment by algorithm with poorly-tuned parameters

Pairwise sequence alignment algorithms

-GRTCPKPDDL PFSTVVP-LKTFYEPGEEITYSCKPGYVSRGGMRKFICPLTGLWPINTLKCTP
E-VKCPFPSRPDNGFVNYP AKPTLYYKDKATFGCHDGYSLDGP-EEIECTKLGNSAMPSC-KA

Ground-truth alignment

GRTCP---KPDDL PFSTVVPLKTFYEPGEEITYSCKPGYVSRGGMRKFICPLTGLWPINTLKCTP
EVKCPFPSRPDN-GFVNYP AKPTLYYK-DKATFGCHDGY-SLDGPEEIECTKLGNS-AMPSCKA

Alignment by algorithm with poorly-tuned parameters

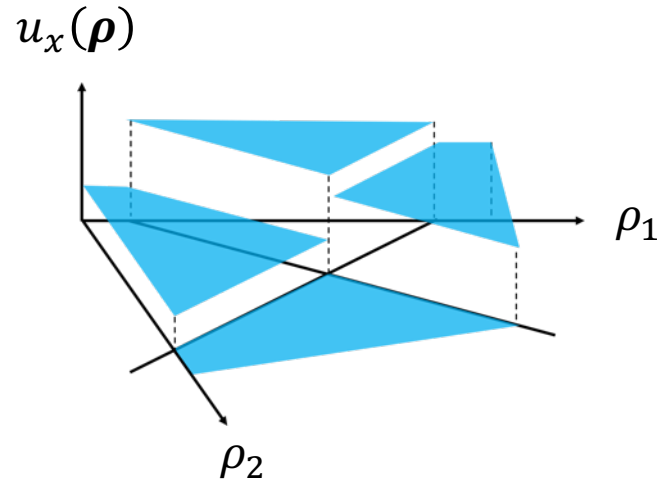
GRTCPKPDDL PFSTV-VPLKTFYEPGEEITYSCKPGYVSRGGMRKFICPLTGLWPINTLKCTP
EVKCPFPSRPDNGFVNYP AKPTLYYKDKATFGCHDGY-SLDGPEEIECTKLGNSA-MPSCKA

Alignment by algorithm with well-tuned parameters

Outline

1. Pairwise sequence alignment algorithms
2. Sample complexity for pairwise alignment
3. Multiple-sequence alignment algorithms
4. Sample complexity for multiple-sequence alignments
5. Additional applications

Piecewise-constant utility functions



$$x = (S, S')$$

Theorem

If for any problem x , the func $\rho \mapsto u_\rho(x)$ is piecewise constant and boundaries between pieces defined by k hyperplanes:

Pseudo-dimension of $\{u_\rho \mid \rho \in \mathbb{R}^d\}$ is $O(d \log k)$

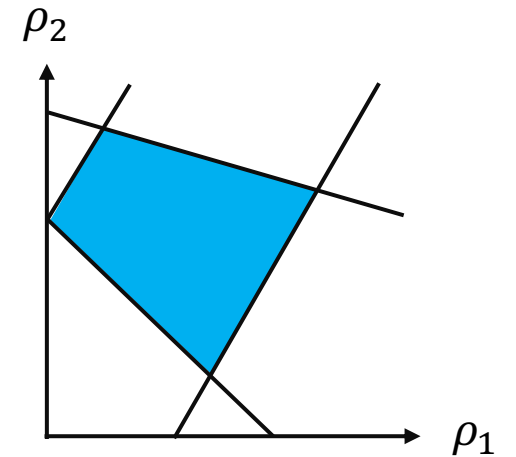
An optimal ρ on $O\left(\frac{d \log k}{\epsilon^2}\right)$ samples is ϵ -optimal on \mathcal{D} .

Need to show piecewise constant utilities and bound $\log(k)$

Key structural property

Lemma:

- For any sequence pair $S, S' \in \Sigma^n$, there exists partition of \mathbb{R}^d such that:
For any region R , across all $\boldsymbol{\rho} \in R$, algorithm's output is invariant
- Partition induced by $\binom{\text{total \# alignments}}{2}$ hyperplanes



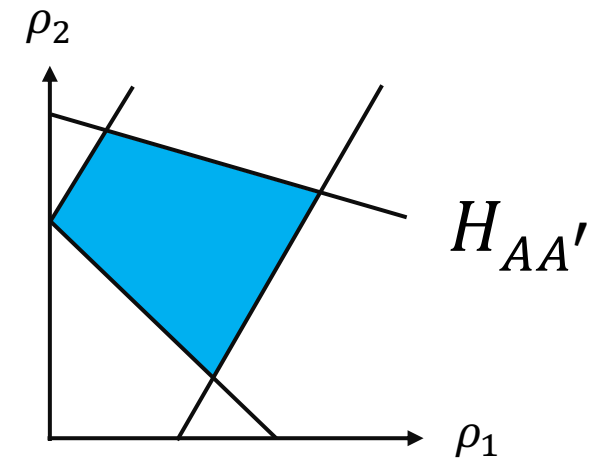
Key structural property

Lemma:

- For any sequence pair $S, S' \in \Sigma^n$, there exists partition of \mathbb{R}^d such that:
For any region R , across all $\boldsymbol{\rho} \in R$, algorithm's output is invariant
- Partition induced by $\binom{\text{total \# alignments}}{2}$ hyperplanes

Proof:

- For any pair of alignments A, A' , prefer A over A' when $\sum_i \rho_i \cdot f_i(A) > \sum_i \rho_i \cdot f_i(A')$.
- Preference for A vs A' determined by hyperplane $H_{AA'}$.
- Let $\mathcal{H} = \{H_{AA'} \mid A, A' \text{ alignments}\}$.
- On any region R in $\mathbb{R}^d \setminus \mathcal{H}$, alignment ordering fixed.
- If DP solver breaks ties reasonably, output constant.



Key structural property

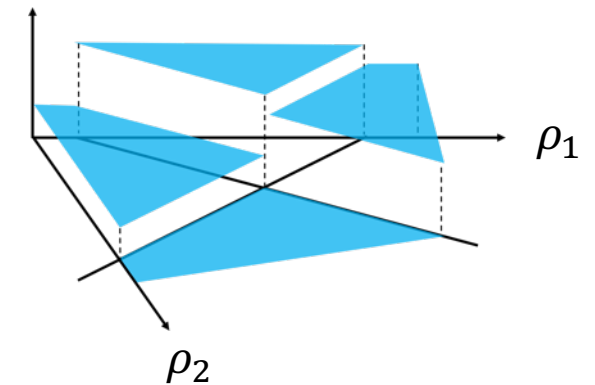
Lemma:

- For any sequence pair $S, S' \in \Sigma^n$, there exists partition of \mathbb{R}^d such that:
For any region R , across all $\boldsymbol{\rho} \in R$, algorithm's output is invariant
- Partition induced by $\binom{\text{total \# alignments}}{2}$ hyperplanes

Corollary:

- For fixed S, S' , algorithm's utility is piecewise-constant function of $\boldsymbol{\rho}$

Similarity to ground truth



Key structural property

Lemma:

- For any sequence pair $S, S' \in \Sigma^n$, there exists partition of \mathbb{R}^d such that:
For any region R , across all $\rho \in R$, algorithm's output is invariant
- Partition induced by $\binom{\text{total \# alignments}}{2}$ hyperplanes

Total # alignments when $|S|, |S'| \leq n$ at most $2^n n^{2n+1}$

Generalization for pairwise alignment

For any sequence pair (S, S') :

$u_{\boldsymbol{\rho}}(S, S')$ = utility of using params $\boldsymbol{\rho} \in \mathbb{R}^d$ to align S, S'

Similarity between algorithm's output & ground truth

Theorem

Pseudo-dimension of $\{u_{\boldsymbol{\rho}} \mid \boldsymbol{\rho} \in \mathbb{R}^d\}$ is $\tilde{O}(dn)$ where $n = \max |S|$

Proof: Pseudo-dimension is $O(d \log(k))$ where $k = O(2^n n^{2n+1})$

Corollary

Optimal $\boldsymbol{\rho}$ on sample of size $\tilde{O}\left(\frac{dn}{\epsilon^2}\right)$ is ϵ -optimal for \mathcal{D} w.h.p.

Improvement for a special case

Special case widely used in practice:

Given parameters $\rho_1, \rho_2, \rho_3 \geq 0$, find alignment maximizing:

$$(\# \text{ matches}) - \rho_1 \cdot (\# \text{ mismatches}) - \rho_2 \cdot (\# \text{ indels}) - \rho_3 \cdot (\# \text{ gaps})$$

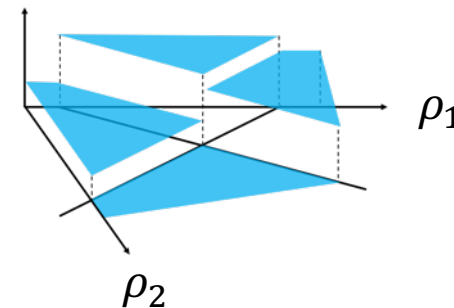
Theorem

[Gusfield, Balasubramanian, Naor '94; Fernández-Baca, Seppäläinen, Slutzki '04]

- For any sequence pair S, S' , there exists partition of \mathbb{R}^3 such that:
 - For any region R , across all $\boldsymbol{\rho} \in R$, algorithm's output is invariant
- Partition induced by $O(n^6)$ hyperplanes

Improvement from $\approx n^n$ to n^6

Improvement for a special case



Given parameters $\rho_1, \rho_2, \rho_3 \geq 0$, find alignment maximizing:

$$(\# \text{ matches}) - \rho_1 \cdot (\# \text{ mismatches}) - \rho_2 \cdot (\# \text{ indels}) - \rho_3 \cdot (\# \text{ gaps})$$

Theorem

Pseudo-dim of $\{u_\rho \mid \rho \in \mathbb{R}^3\}$ is $O(\log n)$ where $n = \max |S|$
vs $\tilde{O}(dn)$

Corollary

- Optimal ρ on sample of size $\tilde{O}\left(\frac{\log n}{\epsilon^2}\right)$ is ϵ -optimal for \mathcal{D} w.h.p.
vs $\tilde{O}\left(\frac{dn}{\epsilon^2}\right)$

Outline

1. Pairwise sequence alignment algorithms
2. Sample complexity for pairwise alignment
3. Multiple-sequence alignment algorithms
4. Sample complexity for multiple-sequence alignments
5. Additional applications

Multiple sequence alignment

```
Q5E940_BOVIN -----MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKOMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE
RLA0_HUMAN -----MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKOMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE
RLA0_MOUSE -----MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKOMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE
RLA0_RAT -----MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKOMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE
RLA0_CHICK -----MPREDRATWKSNYFMKIIQLLDDYPKCFVVGADNVGSKOMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE
RLA0_RANSY -----MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKOMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--SALE
Q7ZUG3_BRARE -----MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKOMQTIIRLSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE
RLA0 ICTPU -----MPREDRATWKSNYFLKIIQLLNDYPKCFIVGADNVGSKOMQTIIRLSLRGK-AIVLMGKNTMMRKAIRGHLENN--PALE
RLA0_DROME -----MVRENKAAWKAQYFIKVVVLELDFEFPKCFIVGADNVGSKOMQNIIRTSLRGL-AVVLMGKNTMMRKAIRGHLENN--PQLE
RLA0_DICDI -----MSGAG-SKRKKLFIEKATKLFTTYDKMIVAEADFGVSSQLQKIRKSIRGI-GAVLMGKKTMRKRVIRDLADSK--PELD
Q54LP0_DICDI -----MSGAG-SKRKNVFIEKATKLFTTYDKMIVAEADFGVSSQLQKIRKSIRGI-GAVLMGKKTMRKRVIRDLADSK--PELD
RLA0_PLAF8 -----MAKLSKQKKQMYIEKLSLIQQYSKILIVHVDNVGSNQMASVRKSLRGK-ATIILMGKNTRIRRTALKKNLQAV--PQIE
RLA0_SULAC -----MIGLAVTTTKKIAKWKVDEVAELTEKTKTHKTIIIANIEGFPADKLHEIRKKLRGK-ADIKVTKNNLFNIALKNAG-----YDTK
RLA0_SULTO -----MRIMAVITQERKIAKWKIEEVKELEOKLREYHTIIIANIEGFPADKLHDIRKKMRGM-AEIKVTKNTLFGIAAKNAG-----LDVS
RLA0_SULSO -----MKRLALALKQRKVASWVLEEVKELTELKNSNTILIGNLEGFPADKLHEIRKKLRGK-ATIKVTKNTLFLKIAAKNAG-----IDIE
RLA0_AERPE MSVVS LVGQMYKREKPIPEWKTLMLELEELFSKHRVVLFAADLTGTPTFVVQRVRKKLWKK-YPMVAKKRIILRAMKAAGLE---LDDN
RLA0_PYRAE -MMLAIGKRRYVTRQYPARKVKIVSEATELLQKYPYVFLFDLHGLSSRILHEYRYRLRRY-GVIKIIKPTLFLKIAFTKVYGG---IPAE
RLA0_METAC -----MAEERHTEHIPQWKDEIENIKELIQSHKVFVGMVIEGILATKMQKIRRDLDKV-AVLKVSNTLTERALNQLG-----ETIP
RLA0_METMA -----MAEERHTEHIPQWKDEIENIKELIQSHKVFVGMVRIEGILATKIQKIRRDLDKV-AVLKVSNTLTERALNQLG-----ESIP
RLA0_ARCFU -----MAAVRGS---PPEYKVRAVEEIKRMISSKPVVAIVSFRNVPAGOMQKIRREFRGK-AEIKVKNTLLERALDALG-----GDYL
RLA0_METKA MAVKAKGQPPSGYEPKVAEWRREVKELKELMDEYENVGLVDLEGIPAPQLQEIRAKLRERDTIIRMSRNTLMRIALEEKLDER--PELE
RLA0_METH -----MAHVAEWKKKEVQELHDLIKGYEVVGIANLADIPARQLQKMRQTLRDS-ALIRMSKKTLLISLAEKAGREL--ENVD
```

Multiple sequence alignment

Input: Collection of sequences $S_1, \dots, S_N \in \Sigma^n$

Alignment: Sequences $\tau_1, \dots, \tau_N \in (\Sigma \cup \{-\})^*$ such that:
Deleting “-” from τ_i yields S_i .

$S_1 =$ A C T G

$S_2 =$ G T C A

$S_3 =$ C T T A

$\tau_1 =$ A - - C T G

$\tau_2 =$ - G T C A -

$\tau_3 =$ C - T T A -

Multiple sequence alignment algorithms

Given parameters $\boldsymbol{\rho} \in \mathbb{R}^d$:

Find alignment A maximizing: $\rho_1 \cdot f_1(A) + \dots + \rho_d \cdot f_d(A)$

$f_1(A), \dots, f_d(A)$ features of alignment A (e.g., # matches, ...)

Dynamic programming table has n^N entries – exp. running time!

Finding $\min_A \rho_1 \cdot f_1(A) + \dots + \rho_d \cdot f_d(A)$ is NP-complete!

[Wang and Jiang, 1994, Kececioğlu and Starrett, 2004]

In practice, use heuristic algorithms

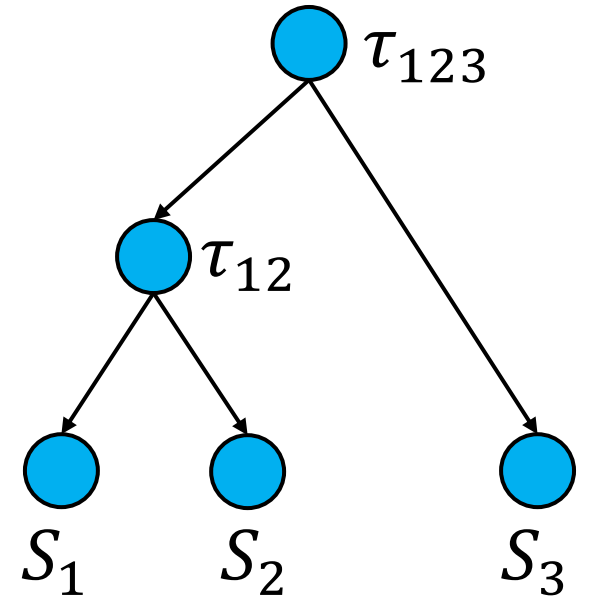
Progressive multiple sequence alignment

Given a binary *guide tree* over sequences
e.g. obtained by clustering sequences

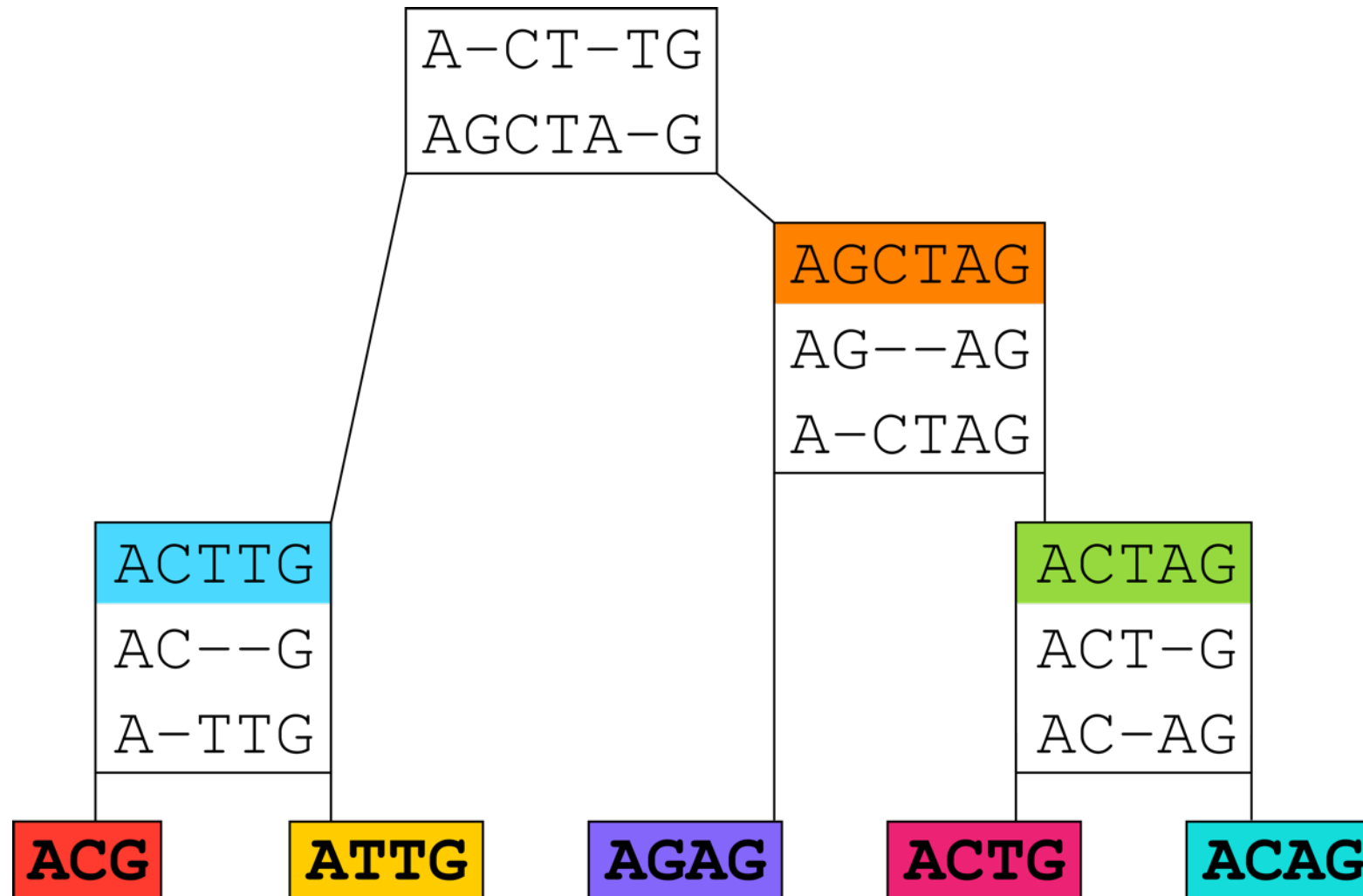
Use pairwise algo to align children of each node
Find pairwise alignments minimizing $\sum_i \rho_i \cdot f_i(A)$

Output alignment at the root node

Algorithm parameters: ρ_1, \dots, ρ_d



Progressive multiple sequence alignment



Outline

1. Pairwise sequence alignment algorithms
2. Sample complexity for pairwise alignment
3. Multiple-sequence alignment algorithms
4. Sample complexity for multiple-sequence alignments
5. Additional applications

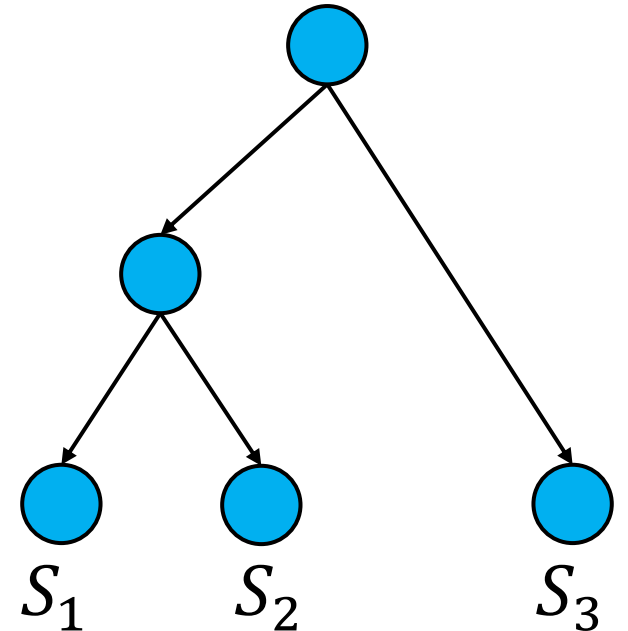
Key structural property

Lemma:

- For any sequences $S_1, \dots, S_N \in \Sigma^n$, there exists partition of \mathbb{R}^d such that:
 - For any region R , across all $\rho \in R$, algorithm's output is invariant
- Partition induced by k hyperplanes with $\log(k) = \tilde{O}(d^{\eta+1}nN)$
 $\eta =$ bound on depth of guide trees

Idea:

- Solve pairwise alignment at each node.
- Collect the hyperplanes from each node!
- Complication: prob. at internal node depends on children alignment.
- Include hyperplanes for every possible problem faced at each node.



Pseudo-dim of multi-sequence alignment

Theorem

Pseudo-dimension of $\{u_{\rho} \mid \rho \in \mathbb{R}^d\}$ is $\tilde{O}(d^{\eta+2}nN)$

n = number of problems

N = number of sequences per problem

d = number of alignment features

η = bound on guide-tree depth.

Corollary

Optimal ρ on sample of size $\tilde{O}\left(\frac{d^{\eta+2}nN}{\epsilon^2}\right)$ is ϵ -optimal for \mathcal{D} w.h.p.

If guide trees roughly balanced, then $\eta = O(\log(n))$.

Outline

1. Pairwise sequence alignment algorithms
2. Sample complexity for pairwise alignment
3. Multiple-sequence alignment algorithms
4. Sample complexity for multiple-sequence alignments
5. Additional applications

RNA folding

RNA assembled as a chain of bases

Denoted as sequence in $\{A, U, C, G\}^*$

Often found as single strand folded into itself

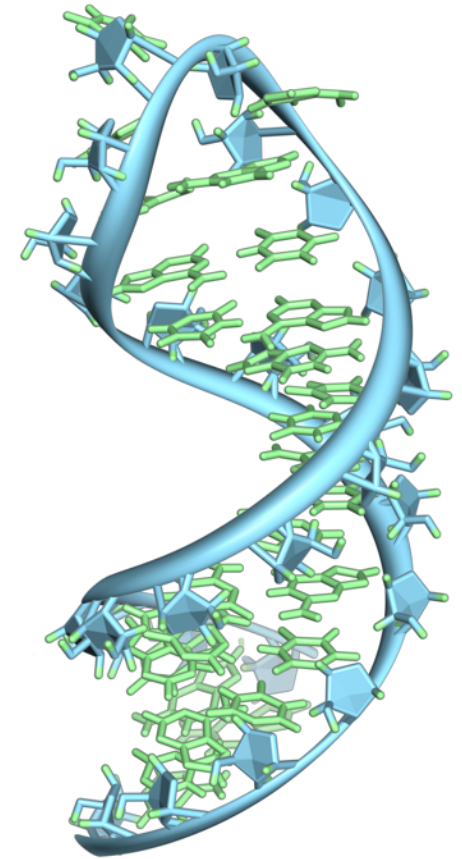
Non-adjacent bases physically bound together

Given unfolded RNA strand:

Infer how would naturally fold

Sheds light on function

We provide sample complexity guarantees for inferring RNA folding



Predicting TADs

Linear DNA of genome wraps into 3D structures
Influence genome function

Topologically associating domains (TADs):
Contiguous segments of genome
that fold into compact regions



We provide sample complexity guarantees for predicting TADs

Conclusion

- Goal: Learn parameters for sequence alignment to recover ground truth alignments
- Sample complexity for pairwise alignment.
- Sample complexity for progressive multi-sequence alignment
- Mentioned other computational biology applications