A geometric combinatorial approach to RNA folding

Qijun He (Clemson University)

Joint work with:
Christine Heitsch (Georgia Tech)
Svetlana Poznanovikj (Clemson)
Andrew Gainer-Dewar (UConn Health Center)
Elizabeth Drellich (North Texas)
Heather Harrington (Oxford)

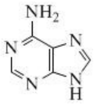
Mathematics Research Communities (MRC) 2014
Snowbird, Utah

Virginia Bioinformatics Institute
July 9, 2015

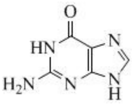
What is RNA?

RNA (*Ribonucleic acid*) are biological molecules built from strings of nucleotides.

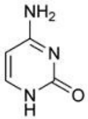
adenine (**A**), *guanine* (**G**), *cytosine* (**C**), *thymine* (**T**), *uracil* (**U**)



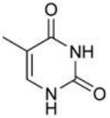
adenine



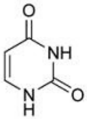
guanine



cytosine



thymine



uracil

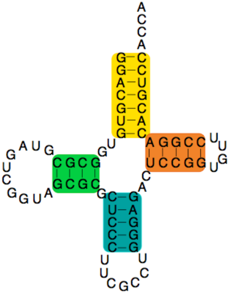
RNA strands consist of **A**, **C**, **G**, and **U**.

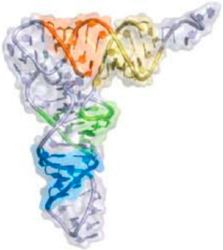
Combinatorially, an RNA strand is a length-n sequence, over the alphabet $\{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{U}\}$.

RNA sequences via base pairings

Primary sequence ⟶ Secondary structure ⟶ 3D molecule

GGGCGUAUGGCG
CGUAGUCGGUAG
CGCGCUCCCUUC
GCCUGGGAGACU
CCGGUGUUCCGG
ACACGUCCACCA

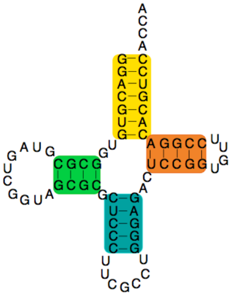




RNA sequences via base pairings

Primary sequence ⟶ Secondary structure ⟶ 3D molecule

GGGCGUAUGGCG
CGUAGUCGGUAG
CGCGCUCCCUUC
GCCUGGGAGACU
CCGGUGUUCCGG
ACACGUCCACCA

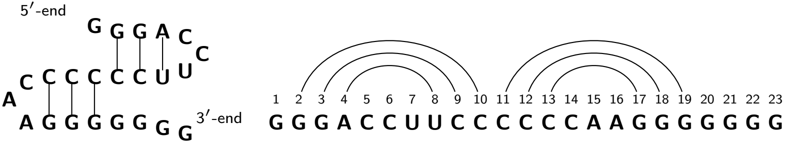


RNA secondary structures balance energetically favorable helices (consecutive base pairs) against destabilizing loops (single-stranded bases).

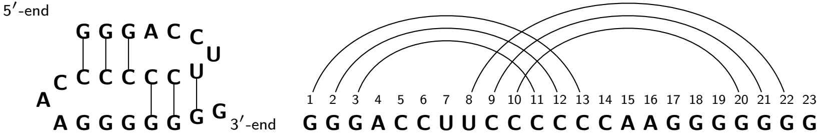
Secondary structure & pseudoknots

Here are two folds of the same RNA strand, and the corresponding arc diagrams.

The first is a secondary structure and the second is a pseudoknot.



5′-end
3′-end
G G G A C C U U C C C C C A A G G G G G G
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
G G A C C U U C C C C C C A A G G G G G G G



5′-end
3′-end
G G G A C C U U C C C C C A A G G G G G G
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
G G A C C U U C C C C C C A A G G G G G G G

Secondary structure prediction

The problem

For a given sequence, there are many possible secondary structures into which it can fold. *What is the most 'likely' one?*

Secondary structure prediction

The problem

For a given sequence, there are many possible secondary structures into which it can fold. *What is the most 'likely' one?*

Minimal free energy (mfe) model

The optimal secondary structure minimizes the free energy, ΔG.

Example energy model

Given an RNA sequence $S = b_1 b_2 \cdots b_n$, let

$$\delta g(i,j) = \begin{cases} -3 & \{b_i, b_j\} = \{\mathbf{C}, \mathbf{G}\} \text{ and } i \leq j-4 \\ -2 & \{b_i, b_j\} = \{\mathbf{A}, \mathbf{U}\} \text{ and } i \leq j-4 \\ -1 & \{b_i, b_j\} = \{\mathbf{G}, \mathbf{U}\} \text{ and } i \leq j-4 \\ 0 & \text{otherwise.} \end{cases}$$

be the free energy of the potential bond between b_i and b_j. Find the structure that minimizes ΔG, the sum of the energies of the base pairs.

Secondary structure prediction

The problem

For a given sequence, there are many possible secondary structures into which it can fold. *What is the most 'likely' one?*

Minimal free energy (mfe) model

The optimal secondary structure minimizes the free energy, ΔG.

Example energy model

Given an RNA sequence $S = b_1 b_2 \cdots b_n$, let

$$\delta g(i,j) = \begin{cases} -3 & \{b_i, b_j\} = \{\mathbf{C}, \mathbf{G}\} \text{ and } i \le j-4 \\ -2 & \{b_i, b_j\} = \{\mathbf{A}, \mathbf{U}\} \text{ and } i \le j-4 \\ -1 & \{b_i, b_j\} = \{\mathbf{G}, \mathbf{U}\} \text{ and } i \le j-4 \\ 0 & \text{otherwise.} \end{cases}$$

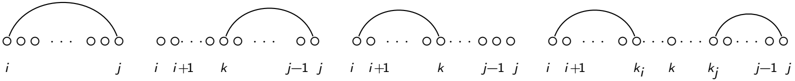
be the free energy of the potential bond between b_i and b_j. Find the structure that minimizes ΔG, the sum of the energies of the base pairs.

This can be done using dynamic programming (DP) to recurse on the substructures.

MFE folding: toy example

There are 4 ways to recurse on the substructure $S_{i,j} = b_i b_{i+1} \cdots b_{j-1} b_j$.

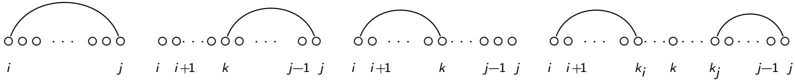
These correspond to the following 4 cases about how bases b_i and b_j can bond:



MFE folding: toy example

There are 4 ways to recurse on the substructure $S_{i,j} = b_i b_{i+1} \cdots b_{j-1} b_j$.

These correspond to the following 4 cases about how bases b_i and b_j can bond:



Thus the optimal energy score $\Delta G(i,j)$ of subsequence $S_{i,j}$ is given by:

$$\Delta G(i,j) = \min \begin{cases} \Delta G(i+1, j-1) + \delta g(i,j) \\ \Delta G(i+1, j) \\ \Delta G(i, j-1) \\ \min\limits_{i<k<j} \Delta G(i,k) + \Delta G(k+1, j). \end{cases}$$

Our final goal is to compute $S_{1,n}$.

A toy example: $S =$ **GGGACCUUCC**

i \ j →	G	G	G	A	C	C	U	U	C	C
G	0	0	0	0	−3					
G		0	0	0	0	−3				
G			0	0	0	0	−1			
A				0	0	0	0	−2		
C					0	0	0	0	0	
C						0	0	0	0	0
U							0	0	0	0
U								0	0	0
C									0	0
C										0

	G	G	G	A	C	C	U	U	C	C
G	0	0	0	0	−3	−3	−4	−4	−6	**−8**
G		0	0	0	0	−3	−3	−3	−5	−8
G			0	0	0	0	−1	−2	−5	−5
A				0	0	0	0	−2	−2	−2
C					0	0	0	0	0	0
C						0	0	0	0	0
U							0	0	0	0
U								0	0	0
C									0	0
C										0

Figure: Recording the optimal scores in a table during a DP routine.

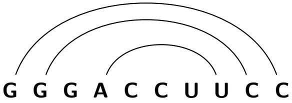
A toy example: $S =$ **GGGACCUUCC**

j

i	G	G	G	A	C	C	U	U	C	C
G	0	0	0	0	−3					
G		0	0	0	0	−3				
G			0	0	0	0	−1			
A				0	0	0	0	−2		
C					0	0	0	0	0	
C						0	0	0	0	0
U							0	0	0	0
U								0	0	0
C									0	0
C										0

	G	G	G	A	C	C	U	U	C	C
G	0	0	0	0	−3	−3	−4	−4	−6	**−8**
G		0	0	0	0	−3	−3	−3	−5	−8
G			0	0	0	0	−1	−2	−5	−5
A				0	0	0	0	−2	−2	−2
C					0	0	0	0	0	0
C						0	0	0	0	0
U							0	0	0	0
U								0	0	0
C									0	0
C										0

Figure: Recording the optimal scores in a table during a DP routine.



$\Delta G(S) = -8$

MFE folding: toy example

We can use the language from OR to describe RNA folding folding problems.

In our toy example of $S = $ **GGGACCUUCC**, the problem can be rephrased as:

$$\min \Delta G = -3k_1 - 2k_2 - k_3,$$

such that,

MFE folding: toy example

We can use the language from OR to describe RNA folding folding problems.

In our toy example of $S =$ **GGGACCUUCC**, the problem can be rephrased as:

$$\min \Delta G = -3k_1 - 2k_2 - k_3,$$

such that,

there exist a 'valid' structure T on S with

- k_1 **CG** pairs,
- k_2 **AU** pairs,
- k_3 **GU** pairs.

MFE folding: toy example

We can use the language from OR to describe RNA folding folding problems.

In our toy example of $S =$ **GGGACCUUCC**, the problem can be rephrased as:

$$\min \Delta G = -3k_1 - 2k_2 - k_3,$$

such that,

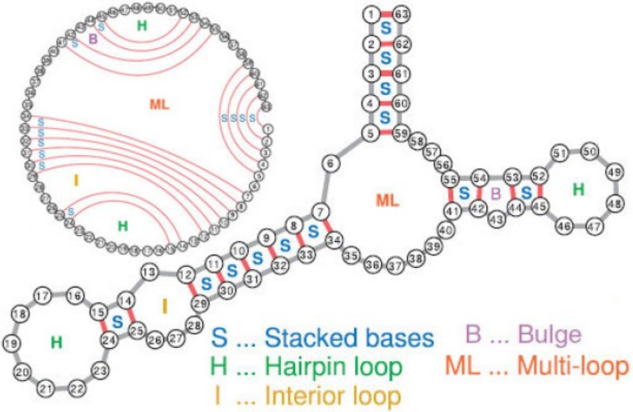
there exist a 'valid' structure T on S with

- k_1 **CG** pairs,
- k_2 **AU** pairs,
- k_3 **GU** pairs.

We know little about the *feasible region* of this optimization problem, but we know it is finite.

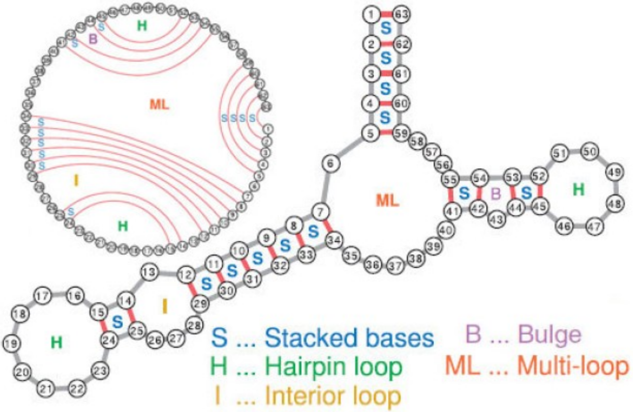
NNTM

Nearest neighbor thermodynamic model: linear objective function with over 7,000 parameters (Turner99 database)!



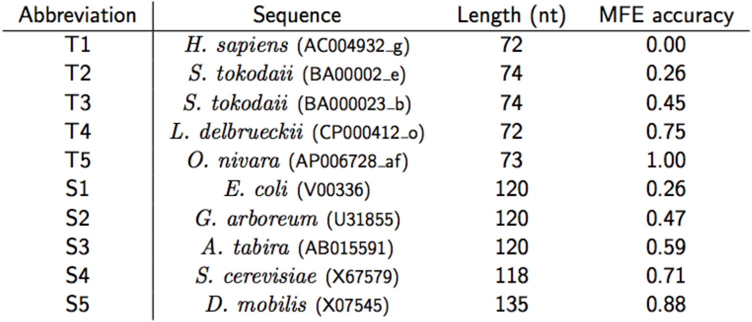
NNTM

Nearest neighbor thermodynamic model: linear objective function with over 7,000 parameters (Turner99 database)!



NNTM

DP solves discrete optimization efficiently, but quality of free energy approximation by the NNTM objective function varies widely.



Abbreviation	Sequence	Length (nt)	MFE accuracy
T1	*H. sapiens* (AC004932_g)	72	0.00
T2	*S. tokodaii* (BA00002_e)	74	0.26
T3	*S. tokodaii* (BA000023_b)	74	0.45
T4	*L. delbrueckii* (CP000412_o)	72	0.75
T5	*O. nivara* (AP006728_af)	73	1.00
S1	*E. coli* (V00336)	120	0.26
S2	*G. arboreum* (U31855)	120	0.47
S3	*A. tabira* (AB015591)	120	0.59
S4	*S. cerevisiae* (X67579)	118	0.71
S5	*D. mobilis* (X07545)	135	0.88

NNTM

DP solves discrete optimization efficiently, but quality of free energy approximation by the NNTM objective function varies widely.

Abbreviation	Sequence	Length (nt)	MFE accuracy
T1	*H. sapiens* (AC004932_g)	72	0.00
T2	*S. tokodaii* (BA00002_e)	74	0.26
T3	*S. tokodaii* (BA000023_b)	74	0.45
T4	*L. delbrueckii* (CP000412_o)	72	0.75
T5	*O. nivara* (AP006728_af)	73	1.00
S1	*E. coli* (V00336)	120	0.26
S2	*G. arboreum* (U31855)	120	0.47
S3	*A. tabira* (AB015591)	120	0.59
S4	*S. cerevisiae* (X67579)	118	0.71
S5	*D. mobilis* (X07545)	135	0.88

What might go wrong?

Multibranch loop parameters

For computational efficiency, only 3 parameters are used to govern multibranch loops.

Multibranch loop parameters

For computational efficiency, only 3 parameters are used to govern multibranch loops. Even worse: they are almost purely 'made up'.

Multibranch loop parameters

For computational efficiency, only 3 parameters are used to govern multibranch loops. Even worse: they are almost purely 'made up'.

Multibranch loop parameters

- a: energy penalty for a multibranch loop.
- b: energy for each unpaired nucleotide in a multibranch loop.
- c: energy for each branching helix in a multibranch loop.

Multibranch loop parameters

For computational efficiency, only 3 parameters are used to govern multibranch loops. Even worse: they are almost purely ‘made up’.

Multibranch loop parameters

- a: energy penalty for a multibranch loop.
- b: energy for each unpaired nucleotide in a multibranch loop.
- c: energy for each branching helix in a multibranch loop.

Multibranch loops encode the topological information, or the shape of the secondary structure.

Multibranch loop parameters

For computational efficiency, only 3 parameters are used to govern multibranch loops. Even worse: they are almost purely ‘made up’.

Multibranch loop parameters

- a: energy penalty for a multibranch loop.
- b: energy for each unpaired nucleotide in a multibranch loop.
- c: energy for each branching helix in a multibranch loop.

Multibranch loops encode the topological information, or the shape of the secondary structure.

Question

How do multibranch loop parameters (a, b, c) affect the optimal structure?

Dimension reduction

For a given structure T, we can write its free energy as:

$$\Delta G(T) = ax_T + by_T + cz_T + w_T,$$

where

- x_T: number of multibranch loops in T,
- y_T: number of unpaired nucleotides in multibranch loops in T,
- z_T: number of branching helices in multibranch loops in T,
- w_T: energy of the remaining structures using Turner99 parameters.

Dimension reduction

For a given structure T, we can write its free energy as:

$$\Delta G(T) = ax_T + by_T + cz_T + w_T,$$

where

- x_T: number of multibranch loops in T,
- y_T: number of unpaired nucleotides in multibranch loops in T,
- z_T: number of branching helices in multibranch loops in T,
- w_T: energy of the remaining structures using Turner99 parameters.

The *profile space* is (x_T, y_T, z_T, w_T), and we introduce a dummy variable d:

$$\Delta G(T) = ax_T + by_T + cz_T + {\color{red}d}w_T.$$

Dimension reduction

For a given structure T, we can write its free energy as:

$$\Delta G(T) = ax_T + by_T + cz_T + w_T,$$

where

- x_T: number of multibranch loops in T,
- y_T: number of unpaired nucleotides in multibranch loops in T,
- z_T: number of branching helices in multibranch loops in T,
- w_T: energy of the remaining structures using Turner99 parameters.

The *profile space* is (x_T, y_T, z_T, w_T), and we introduce a dummy variable d:

$$\Delta G(T) = ax_T + by_T + cz_T + {\color{red}d}w_T.$$

Question

How do multibranch loop parameters (a, b, c, d) affect the optimal structure?

Multibranch loop parameters

We want to run parametric analysis on some specific parameters, but we don't know how 'wrong' they are.

Answer?

We can construct the convex hull (a polytope) of the feasible region.

Multibranch loop parameters

We want to run parametric analysis on some specific parameters, but we don't know how 'wrong' they are.

Answer?

We can construct the convex hull (a polytope) of the feasible region.

The optimal value should exist on an extreme point of the feasible region (a vertex of this polytope).

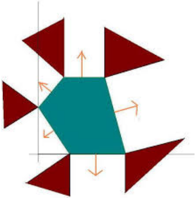
Multibranch loop parameters

We want to run parametric analysis on some specific parameters, but we don't know how 'wrong' they are.

Answer?

We can construct the convex hull (a polytope) of the feasible region.

The optimal value should exist on an extreme point of the feasible region (a vertex of this polytope).



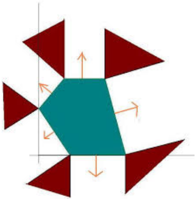
Multibranch loop parameters

We want to run parametric analysis on some specific parameters, but we don't know how 'wrong' they are.

Answer?

We can construct the convex hull (a polytope) of the feasible region.

The optimal value should exist on an extreme point of the feasible region (a vertex of this polytope).



The real problem

How to compute the convex hull of an implicit finite feasible region?

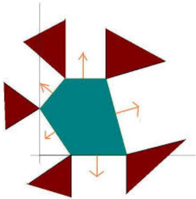
Multibranch loop parameters

We want to run parametric analysis on some specific parameters, but we don't know how 'wrong' they are.

Answer?

We can construct the convex hull (a polytope) of the feasible region.

The optimal value should exist on an extreme point of the feasible region (a vertex of this polytope).



The real problem

How to compute the convex hull of an implicit finite feasible region?

We know little about the feasible region, but we do know how to solve optimization problem over it!

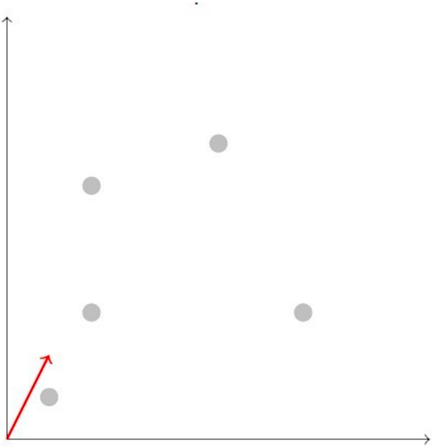
Beneath-beyond method

Big idea: given an implicit finite feasible region, we can build up the convex hull of the feasible region incrementally, by systematically solving different objective vectors.

Beneath-beyond method

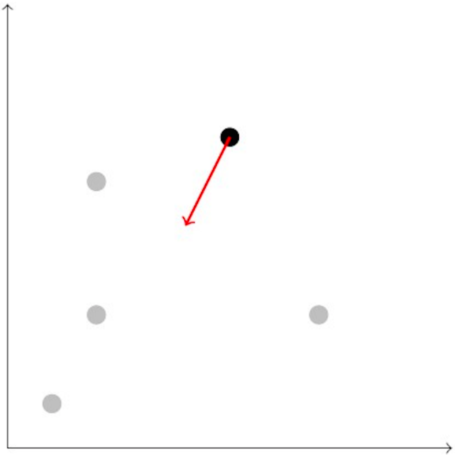
Big idea: given an implicit finite feasible region, we can build up the convex hull of the feasible region incrementally, by systematically solving different objective vectors.

Step 1: find the affine hull of the feasible region.
Start with a random objective vector v, solve for x that optimize $v \cdot x$.



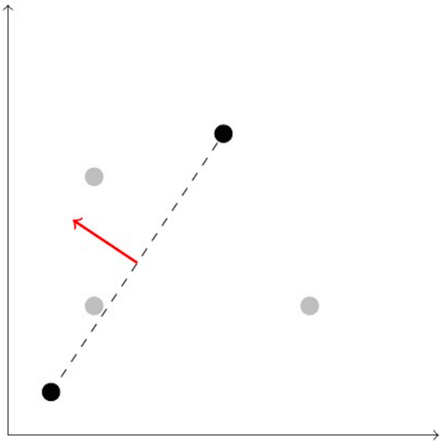
Beneath-beyond method

Step 1: find the affine hull of the feasible region.
Solve for x that optimize $-v \cdot x$.



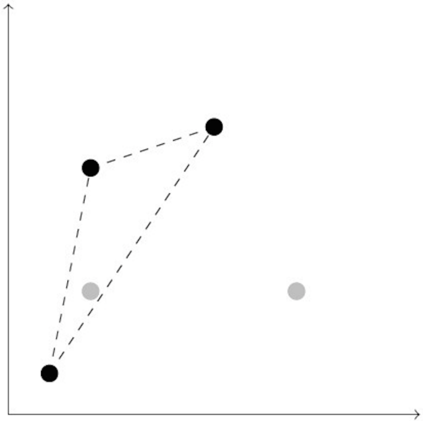
Beneath-beyond method

Step 1: find the affine hull of the feasible region.
Compute the affine hull of the existing vertices. If it is full dimension, go to Step 2. Else, generate a vector orthogonal to the current affine hull and compute the optimization problem w.r.t that vector.



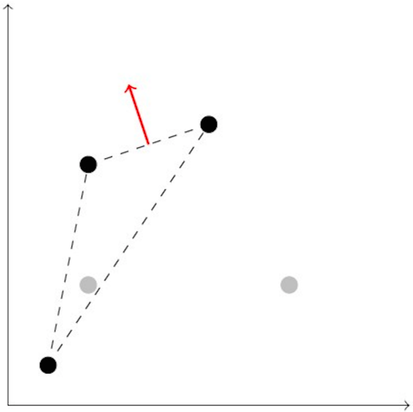
Beneath-beyond method

Step 2: build the polytope incrementally.
Once we 'use up' all the dimensions, compute the convex hull of the the existing vertices. Label each face of the convex hull as temporary.



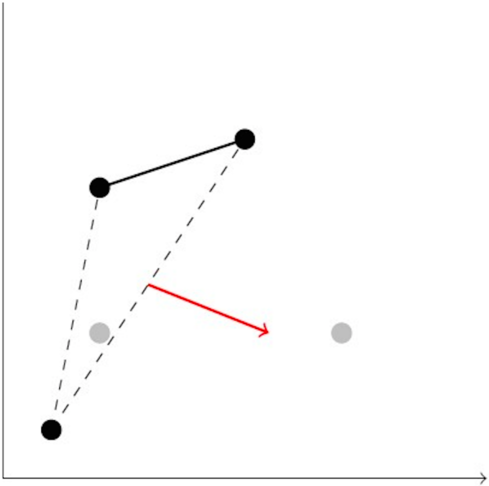
Beneath-beyond method

Step 2: build the polytope incrementally.
Pick a temporary face, use its outer normal vector as objective function and solve the corresponding optimization problem.



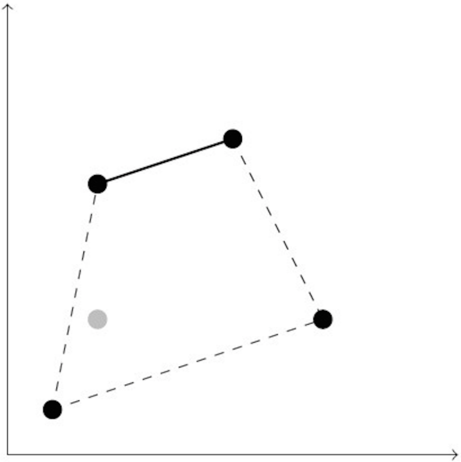
Beneath-beyond method

Step 2: build the polytope incrementally.
If no new vertex outside of the face is found, that face becomes confirmed and the process is restarted with a temporary face.



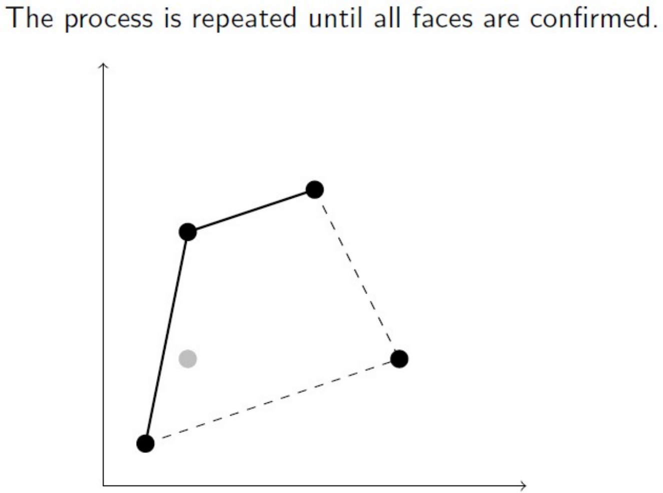
Beneath-beyond method

Step 2: build the polytope incrementally.
If there is a new vertex outside, compute the new convex hull and label the newly added faces as temporary.



Beneath-beyond method

Step 2: build the polytope incrementally.

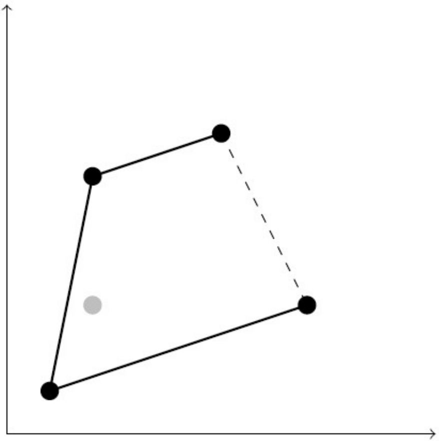


The process is repeated until all faces are confirmed.

Beneath-beyond method

Step 2: build the polytope incrementally.

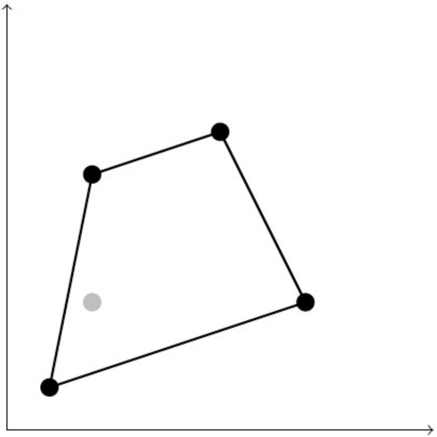
The process is repeated until all faces are confirmed.



Beneath-beyond method

Step 2: build the polytope incrementally.

The process is repeated until all faces are confirmed.



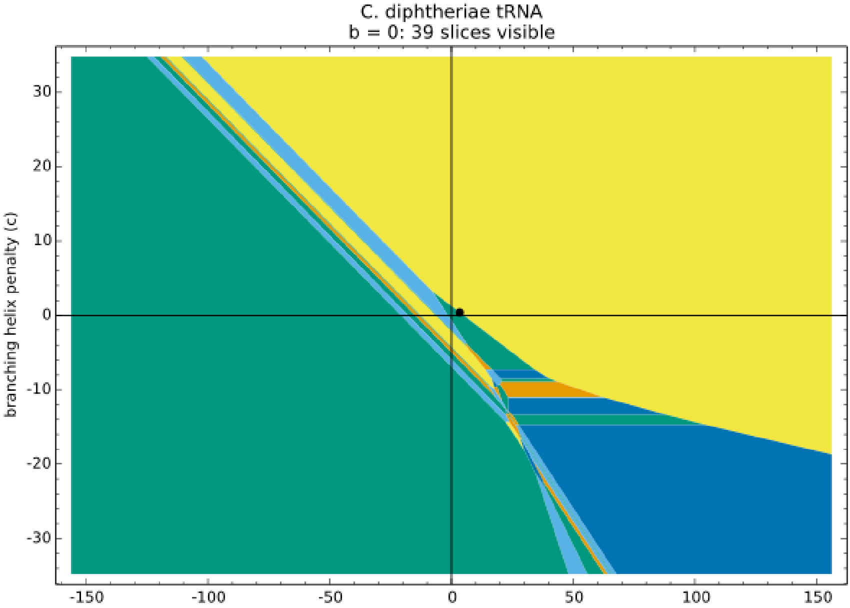
Complexity : $O(V + F)$ objective vectors.

Overview of the actual algorithm

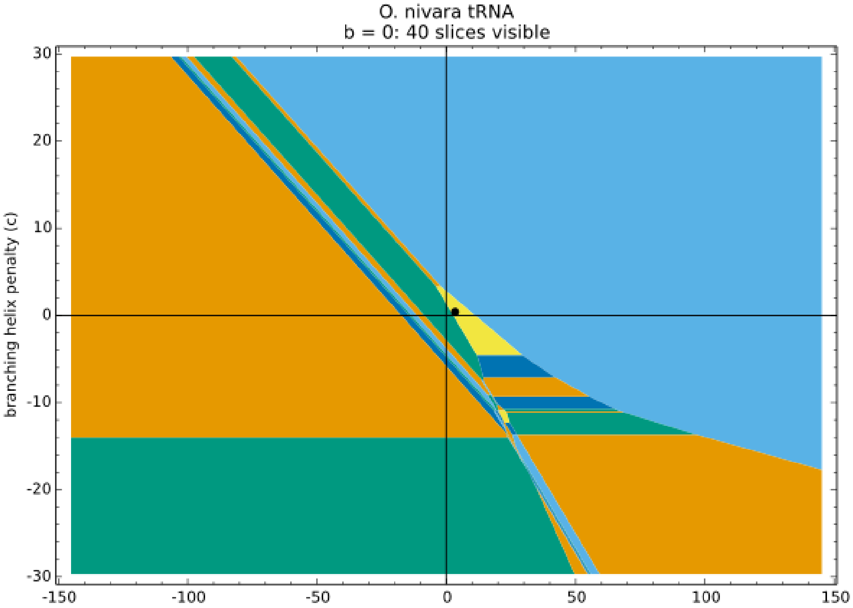
1. For each objective vector (a', b', c', d'), go to the parameter set, set $a = a'$, $b = b'$, $c = c'$ and multiply *all* other parameters by d'. Then compute the MFE structure w.r.t this new parameter set.
2. For each structure we obtained, compute its profile and generate a corresponding vertex.

All the computations are done in rational field, so there won't be any rounding error.

Example


C. diphtheriae tRNA
b = 0: 39 slices visible
branching helix penalty (c)
30
20
10
0
-10
-20
-30
-150
-100
-50
0
50
100
150

Example


O. nivara tRNA
b = 0: 40 slices visible
branching helix penalty (c)
30
20
10
0
-10
-20
-30
-150
-100
-50
0
50
100
150

Future work

- Further develop our software to make it more stable and convenient.
- Run sensitivity analysis on our current multibranch loop parameters.
- Improve prediction to known structures by modifying multibranch loop parameters.
- Predict unknown structures with desired parameters.

pmfe: https://github.com/AMS-MRC-disc-math-bio/pmfe

Future work

- Further develop our software to make it more stable and convenient.
- Run sensitivity analysis on our current multibranch loop parameters.
- Improve prediction to known structures by modifying multibranch loop parameters.
- Predict unknown structures with desired parameters.

pmfe: https://github.com/AMS-MRC-disc-math-bio/pmfe

Thank you!

References

- Colin N Dewey, Peter M Huggins, Kevin Woods, Bernd Sturmfels, and Lior Pachter.
  Parametric alignment of drosophila genomes.
  PLoS Computational Biology, 2(6):e73, 2006.

- Valerie Hower and Christine E Heitsch.
  Parametric analysis of rna branching configurations.
  Bulletin of mathematical biology, 73(4):754–776, 2011.

- Lior Pachter and Bernd Sturmfels.
  Algebraic statistics for computational biology, volume 13.
  Cambridge University Press, 2005.

- M Shel Swenson, Joshua Anderson, Andrew Ash, Prashant Gaurav, Zsuzsanna Sükösd, David A Bader, Stephen C Harvey, and Christine E Heitsch.
  Gtfold: Enabling parallel rna secondary structure prediction on multi-core desktops.
  BMC research notes, 5(1):341, 2012.

- Douglas H Turner and David H Mathews.
  Nndb: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure.
  Nucleic acids research, page gkp892, 2009.

- Michael Zuker.
  Rna folding prediction: The continued need for interaction between biologists and mathematicians.