A Combinatorial Framework for Designing (Pseudoknotted) RNA Algorithms

Yann Ponty Cédric Saule

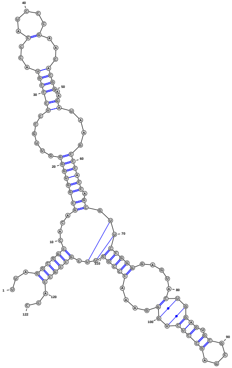
CNRS – INRIA AMIB – École Polytechnique – Palaiseau – France

Institute for Research in Immunology and Cancer – Montreal – Canada

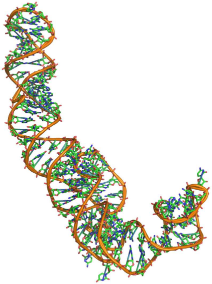
September 5, 2011

UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC

Primary structure



Secondary structure (Matching)



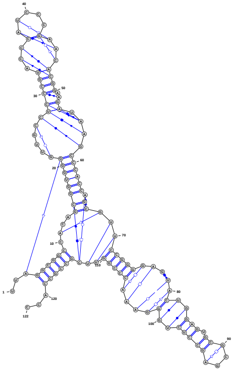
Tertiary structure

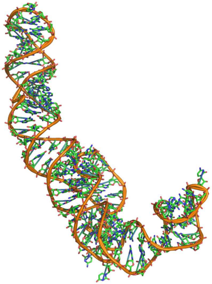
Source: 5s rRNA (PDBID: 1K73:B)

Bottom-up approach to molecular biology

Understand and predict how RNAs fold to decypher their function(s).

UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC





Primary structure

Secondary^{+} structure (Matching)

Tertiary structure

Source: 5s rRNA (PDBID: 1K73:B)

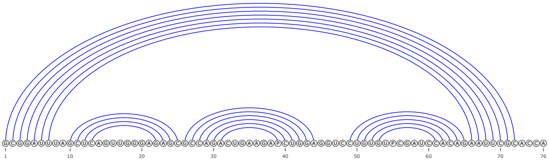
Bottom-up approach to molecular biology

Understand and predict how RNAs fold to decypher their function(s).

Input: RNA sequence ω

Definition (Minimum Free-Energy (MFE) Folding Problem)

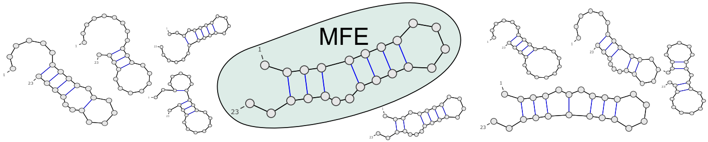
Find a partial matching s^* of positions from ω that min(max)-imizes a free-energy function E_{ω,s^*} within some restricted class of matching.



Secondary Structure (Non-crossing) + Additive energies: Easy!
Optimal substructure $\Rightarrow$ Dynamic Programming (DP)

- (Weighted) base-pairs maximization: $\Theta(n^3)$
  [Nussinov and Jacobson, 1980]
- Nearest-neighbor model: $\Theta(n^4)/\Theta(n^3)$
  [Zuker and Stiegler, 1981]

Boltzmann ensemble and partition function



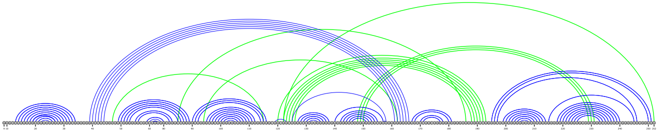
- Energy functions are not ideally accurate
- MFE structure might be isolated

$\Rightarrow$ Recent thermodynamic studies postulate a Boltzmann equilibrium, i.e. admissible conformations exist in a probability distribution [McCaskill, 1990]

$$\mathbb{P}(s) = \frac{e^{\frac{-E_s}{RT}}}{\mathcal{Z}} \quad \text{where} \quad \mathcal{Z} = \sum_{s' \in \mathcal{S}} e^{\frac{-E_s}{RT}} \quad \text{(Partition function)}$$

Observables can be derived, such that the base-pairing prob. [McCaskill, 1990], centroid-structure [Ding and Lawrence, 2003], likelihood of multi-stable RNAs [Voss *et al.*, 2004], confidence in prediction [Mathews, 2004], moments of the free-energy distribution [Miklós *et al.*, 2005]. . .

Pseudoknots



Any matching (crossing): Harder for realistic energy models

- BP maximization: $O(n^3)$ (Max. Weighted Matching)
  [Tabaska *et al.*, 1998]
- Nearest-neighbor: NP-complete
  [Akutsu, 2000, Lyngsø and Pedersen, 2000]

In practice:

- Heuristics/local search
- Restricted conformational spaces solved exactly (DP) in polynomial time
  [Rivas and Eddy, 1999, Lyngsø and Pedersen, 2000, Dirks and Pierce, 2003, Reeder and Giegerich, 2004, Cao and Chen, 2006, Cao and Chen, 2009, Chen *et al.*, 2009, Cao and Chen, 2009, Huang *et al.*, 2009, Theis *et al.*, 2010, Reidys *et al.*, 2011].

Very few of them allow for a transposition to ensemble based approach!

Developing new algorithms: Motivation

Folding RNAs including pseudoknots remains a challenge:

- Incorporate incoming thermodynamic parameters
- Capture complex topological aspects
- Optimize expressivity/computational complexity tradeoff
- Address ensemble-related questions
- Tackle other problems (RNA-RNA interaction)

Yet developing new DP algorithms is tedious and error-prone:

- Lack of modularity
- Tedious proofs for unambiguity/correctness
- Hard to connect DP equation (product) to decomposition (source)

CS geek: Underlying object to define meta-algorithms/proofs?

State-of-the-art

Existing abstractions for Dynamic Programming algorithms:

- Giegerich *et (many!) al*: Algebraic Dynamic Programming
  - Folding = Parsing: Based on grammar/algebra pair
  - \+ Automated (efficient!) code generation
  - \+ Partially automated (heuristic) ambiguity check
  - \+ High level: No indices...
  - \- *Hacky* for context-dependent features (Addressed by Bellmann's GAP?)
- Lefebvre *et al*: Multi-tape attributed grammars
- Roytberg and Finkelstein: Forward hypergraphs

State-of-the-art

Existing abstractions for Dynamic Programming algorithms:

- Giegerich *et (many!) al*: Algebraic Dynamic Programming
- Lefebvre *et al*: Multi-tape attributed grammars
  - Context-free grammar, coupled with attributes system
  - \+ Mixed algebras
  - \- Synchronized multiple tapes for PKs
    $\Rightarrow$ Performance degradation...
- Roytberg and Finkelstein: Forward hypergraphs

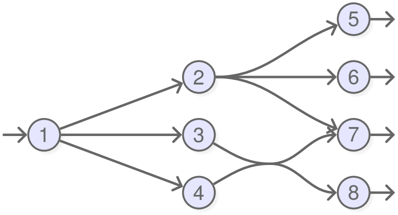
Existing abstractions for Dynamic Programming algorithms:

- Giegerich *et (many!) al*: Algebraic Dynamic Programming
- Lefebvre *et al*: Multi-tape attributed grammars
- Roytberg and Finkelstein: Forward hypergraphs
    - Conformations bijectively associated with hyperpaths ($\sim$ Traces)
    - \+ Highly expressive
    - \- Low-level: Explicit indices manipulation

Main Contribution

Considering families of hypergraphs as combinatorial classes will

- Simplify algorithms
- Ease proving their correctness
- Help develop new applications



Hypergaphs generalize directed graphs to arcs of arbitrary in/out degrees.

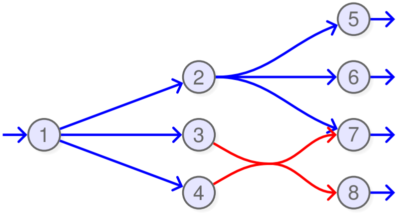
Definition (Hypergraph)

A directed hypergaph $\mathcal{H}$ is a couple (V, E) such that:

- V is a set of vertices
- E is a set of hyperarcs $e = (t(e) \to h(e))$ such that $t(e), h(e) \subset E$

Forward hypergraphs, or F-graphs, are hypergraphs whose arcs have ingoing degree exactly 1.

Hypergraphs as decompositions



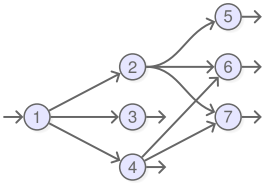
Hypergaphs generalize directed graphs to arcs of arbitrary in/out degrees.

Definition (Hypergraph)

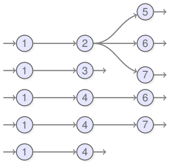
A directed hypergaph $\mathcal{H}$ is a couple (V, E) such that:

- V is a set of vertices
- E is a set of hyperarcs $e = (t(e) \to h(e))$ such that $t(e), h(e) \subset E$

Forward hypergraphs, or F-graphs, are hypergraphs whose arcs have ingoing degree exactly 1.



F-graph



All F-paths starting from vertex 1

Definition (F-path)

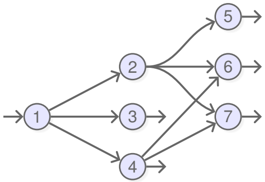
A F-path is a tree having root $s \in V$, whose children are F-paths built from the outgoing vertices of some arc $e = (s \to \mathbf{t}) \in E$.

Remark: Vertices of out degree 0 ($\mathbf{t} = \varnothing$) provide an elegant terminal case.

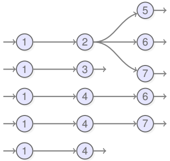
F-graph is independent iff each F-path sees at most once each arc.

A numerical feature fonction $\alpha : E \to \mathbb{R}$ assigns a value to each arc:

- Weight of a path is **the product** of its arcs' values
- Score of a path is **the sum** of its arcs' values



F-graph



All F-paths starting from vertex 1

Definition (F-path)

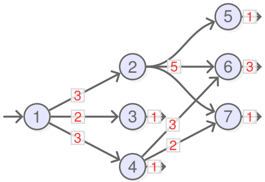
A F-path is a tree having root $s \in V$, whose children are F-paths built from the outgoing vertices of some arc $e = (s \to \mathbf{t}) \in E$.

Remark: Vertices of out degree 0 ($\mathbf{t} = \varnothing$) provide an elegant terminal case.

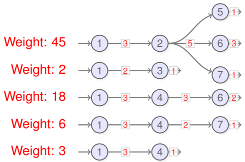
F-graph is independent iff each F-path sees at most once each arc.

A numerical feature fonction $\alpha : E \to \mathbb{R}$ assigns a value to each arc:

- Weight of a path is **the product** of its arcs' values
- Score of a path is **the sum** of its arcs' values



F-graph



All F-paths starting from vertex 1

Definition (F-path)

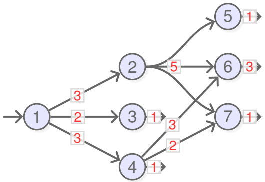
A F-path is a tree having root $s \in V$, whose children are F-paths built from the outgoing vertices of some arc $e = (s \rightarrow \mathbf{t}) \in E$.

Remark: Vertices of out degree 0 ($\mathbf{t} = \varnothing$) provide an elegant terminal case.

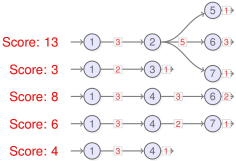
F-graph is independent iff each F-path sees at most once each arc.

A numerical feature fonction $\alpha : E \rightarrow \mathbb{R}$ assigns a value to each arc:

- Weight of a path is **the product** of its arcs' values
- Score of a path is **the sum** of its arcs' values



F-graph



All F-paths starting from vertex 1

Definition (F-path)

A F-path is a tree having root $s \in V$, whose children are F-paths built from the outgoing vertices of some arc $e = (s \to \mathbf{t}) \in E$.

Remark: Vertices of out degree 0 ($\mathbf{t} = \varnothing$) provide an elegant terminal case.

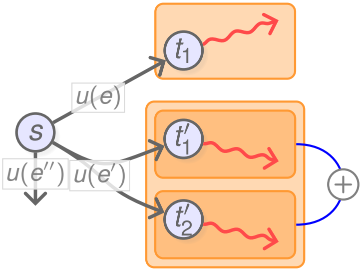
F-graph is independent iff each F-path sees at most once each arc.

A numerical feature fonction $\alpha : E \to \mathbb{R}$ assigns a value to each arc:

- Weight of a path is **the product** of its arcs' values
- Score of a path is **the sum** of its arcs' values

$\mathcal{H} = (v_0, V, E, \alpha)$: acyclic F-graph v_0: Init. node α: feature function

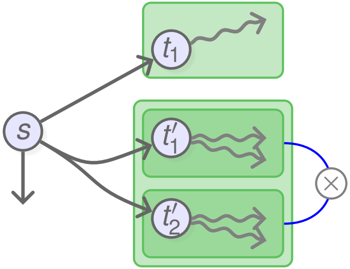
$$m_s = \min_{p \in \mathcal{P}_s} Score(s)$$
$$= \min_{e=(s \to \mathbf{t})} \left(u(e) + \sum_{u \in \mathbf{t}} m_u \right)$$



Problem	Recurrence	Time/space (DP)
Min. score	$m_s = \min_{e=(s \to \mathbf{t})} \left(u(e) + \sum_{u \in \mathbf{t}} m_u \right)$	$\Theta(\|E\| + \|V\|)/\Theta(\|V\|)$
Num. paths	$n_s = \sum_{(s \to \mathbf{t})} \prod_{u \in \mathbf{t}} n_u$	$\Theta(\|E\| + \|V\|)/\Theta(\|V\|)$
Total weight	$w_s = \sum_{e=(s \to \mathbf{t})} u(e) \cdot \prod_{s' \in \mathbf{t}} w_{s'}$	$\Theta(\|E\| + \|V\|)/\Theta(\|V\|)$

$\mathcal{H} = (v_0, V, E, \alpha)$: acyclic F-graph v_0: Init. node α: feature function

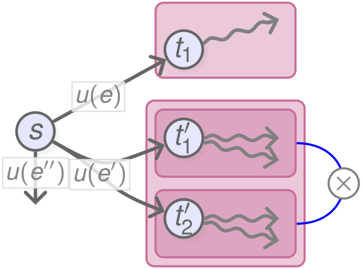
$$n_s = |\mathcal{P}_s| = \sum_{p \in \mathcal{P}_s} 1$$
$$= \sum_{(s \to \mathbf{t})} \prod_{u \in \mathbf{t}} n_u$$



Problem	Recurrence	Time/space (DP)
Min. score	$m_s = \min_{e=(s \to \mathbf{t})} \left(u(e) + \sum_{u \in \mathbf{t}} m_u \right)$	$\Theta(\|E\| + \|V\|)/\Theta(\|V\|)$
Num. paths	$n_s = \sum_{(s \to \mathbf{t})} \prod_{u \in \mathbf{t}} n_u$	$\Theta(\|E\| + \|V\|)/\Theta(\|V\|)$
Total weight	$w_s = \sum_{e=(s \to \mathbf{t})} u(e) \cdot \prod_{s' \in \mathbf{t}} w_{s'}$	$\Theta(\|E\| + \|V\|)/\Theta(\|V\|)$

$\mathcal{H} = (v_0, V, E, \alpha)$: acyclic F-graph v_0: Init. node α: feature function

$$w_s = \sum_{p \in \mathcal{P}_s} Weight(s) = \sum_{e=(s \to \mathbf{t})} u(e) \cdot \prod_{s' \in \mathbf{t}} w_{s'}$$



Problem	Recurrence	Time/space (DP)
Min. score	$m_s = \min_{e=(s \to \mathbf{t})} \left(u(e) + \sum_{u \in \mathbf{t}} m_u \right)$	$\Theta(\lvert E\rvert + \lvert V\rvert)/\Theta(\lvert V\rvert)$
Num. paths	$n_s = \sum_{(s \to \mathbf{t})} \prod_{u \in \mathbf{t}} n_u$	$\Theta(\lvert E\rvert + \lvert V\rvert)/\Theta(\lvert V\rvert)$
Total weight	$w_s = \sum_{e=(s \to \mathbf{t})} u(e) \cdot \prod_{s' \in \mathbf{t}} w_{s'}$	$\Theta(\lvert E\rvert + \lvert V\rvert)/\Theta(\lvert V\rvert)$

Problem	Recurrence	Time/space (DP)
Min. score	$m_s = \min_{e=(s\to\mathbf{t})} \left(u(e) + \sum_{u\in\mathbf{t}} m_u \right)$	$\Theta(\|E\|+\|V\|)/\Theta(\|V\|)$
Num. paths	$n_s = \sum_{(s\to\mathbf{t})} \prod_{u\in\mathbf{t}} n_u$	$\Theta(\|E\|+\|V\|)/\Theta(\|V\|)$
Total weight	$w_s = \sum_{e=(s\to\mathbf{t})} u(e)\cdot \prod_{s'\in\mathbf{t}} w_{s'}$	$\Theta(\|E\|+\|V\|)/\Theta(\|V\|)$

Assume a weighted (Boltzmann) probability distribution on F-paths $\mathcal{P}$:

$$\mathbb{P}(p) = \prod_{e\in p} \alpha(e)/w_{v_0}$$

Problem	Algorithm	Time/space (DP)
Random gen.	Compute w_s; Starting with $s \leftarrow v_0$, pick an arc $s \to (t_1, t_2, \dots)$ w.p. $\prod w_{t_i}/w_s$ and recurse on each t_i.	$\Theta(\|E\|+\|V\|)/\Theta(\|V\|)$
Arcs prob.	$p_e = \dfrac{b_{t(e)}\cdot \prod_{s'\in\mathbf{h}(e)} w_{s'}}{w_{v_0}}$ $b_s = (1+) \sum_{s'\to(t_1\cdots s\cdots)} \alpha(e')b_{s'} \prod_{t_i} w_{t_i}$	$\Theta(\|E\|+\|V\|+\sum\|\mathbf{h}(e)\|^2)$ $\Theta(\|V\|)$

Problem	Recurrence	Time/space (DP)
Min. score	$m_s = \min_{e=(s\to\mathbf{t})} \left(u(e) + \sum_{u\in\mathbf{t}} m_u \right)$	$\Theta(\lvert E\rvert + \lvert V\rvert)/\Theta(\lvert V\rvert)$
Num. paths	$n_s = \sum_{(s\to\mathbf{t})} \prod_{u\in\mathbf{t}} n_u$	$\Theta(\lvert E\rvert + \lvert V\rvert)/\Theta(\lvert V\rvert)$
Total weight	$w_s = \sum_{e=(s\to\mathbf{t})} u(e) \cdot \prod_{s'\in\mathbf{t}} w_{s'}$	$\Theta(\lvert E\rvert + \lvert V\rvert)/\Theta(\lvert V\rvert)$

Assume a weighted (Boltzmann) probability distribution on F-paths $\mathcal{P}$:

$$\mathbb{P}(p) = \prod_{e\in p} \alpha(e)/w_{v_0}$$

Problem	Algorithm	Time/space (DP)
Random gen.	Compute w_s; Starting with $s \leftarrow v_0$, pick an arc $s \to (t_1, t_2, \ldots)$ w.p. $\prod w_{t_i}/w_s$ and recurse on each t_i.	$\Theta(\lvert E\rvert + \lvert V\rvert)/\Theta(\lvert V\rvert)$
Arcs prob.	$p_e = \dfrac{b_{t(e)} \cdot \prod_{s'\in\mathbf{h}(e)} w_{s'}}{w_{v_0}}$ $b_s = (1+) \sum_{s'\to(t_1\cdots s\cdots)} \alpha(e')b_{s'} \prod_{t_i} w_{t_i}$	$\Theta(\lvert E\rvert + \lvert V\rvert + \sum \lvert\mathbf{h}(e)\rvert^2)$ $\Theta(\lvert V\rvert)$

Given an RNA sequence ω and an energy function E, assume one has:

- Acyclic hypergraph $\mathcal{H}$, whose F-paths are in bijection with the targeted (pseudoknotted) conformations
- Feature function α which returns, for each F-path p, the energy $E_{\omega,s}$ of its conformation s_p.

Application		Hypergraph Algorithm	Arguments
MFE folding	$\Leftrightarrow$	Minimum score	$(\mathcal{H}, \alpha)$
Partition function	$\Leftrightarrow$	Total weight	$(\mathcal{H}, e^{-\alpha/RT})$
Statistical sampling	$\Leftrightarrow$	Random generation	$(\mathcal{H}, e^{-\alpha/RT})$
BP probabilities (dot-plot)	$\Leftrightarrow$	Arcs prob.	$(\mathcal{H}, e^{-\alpha/RT})$

Message #1

Dynamic programming equations for ensemble applications are by-products of an underlying combinatorial decomposition ($\Rightarrow$ Family of hypergraphs).

How to design such hypergraphs/energy function?
You do it yourself! But combinatorics can help…

Given an RNA sequence ω and an energy function E, assume one has:

- Acyclic hypergraph $\mathcal{H}$, whose F-paths are in bijection with the targeted (pseudoknotted) conformations
- Feature function α which returns, for each F-path p, the energy $E_{\omega,s}$ of its conformation s_p.

Application		Hypergraph Algorithm	Arguments
MFE folding	$\Leftrightarrow$	Minimum score	$(\mathcal{H}, \alpha)$
Partition function	$\Leftrightarrow$	Total weight	$(\mathcal{H}, e^{-\alpha/RT})$
Statistical sampling	$\Leftrightarrow$	Random generation	$(\mathcal{H}, e^{-\alpha/RT})$
BP probabilities (dot-plot)	$\Leftrightarrow$	Arcs prob.	$(\mathcal{H}, e^{-\alpha/RT})$

Message #1

Dynamic programming equations for ensemble applications are by-products of an underlying combinatorial decomposition ($\Rightarrow$ Family of hypergraphs).

How to design such hypergraphs/energy function?
You do it yourself! But combinatorics can help...

Given an RNA sequence ω and an energy function E, assume one has:

- Acyclic hypergraph $\mathcal{H}$, whose F-paths are in bijection with the targeted (pseudoknotted) conformations
- Feature function α which returns, for each F-path p, the energy $E_{\omega,s}$ of its conformation s_p.

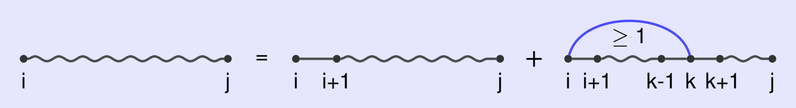
Application		Hypergraph Algorithm	Arguments
MFE folding	$\Leftrightarrow$	Minimum score	$(\mathcal{H}, \alpha)$
Partition function	$\Leftrightarrow$	Total weight	$(\mathcal{H}, e^{-\alpha/RT})$
Statistical sampling	$\Leftrightarrow$	Random generation	$(\mathcal{H}, e^{-\alpha/RT})$
BP probabilities (dot-plot)	$\Leftrightarrow$	Arcs prob.	$(\mathcal{H}, e^{-\alpha/RT})$

Message #1

Dynamic programming equations for ensemble applications are by-products of an underlying combinatorial decomposition ($\Rightarrow$ Family of hypergraphs).

How to design such hypergraphs/energy function?
You do it yourself! But combinatorics can help...

Decomposition:



As an Hypergraph: $\Theta(n^2)$ vertices, $\Theta(n^3)$ arcs ($n = |\omega|$).

$1,|\omega|$

Initial vertex

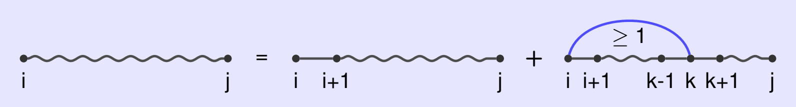
i,i-1 $\xrightarrow{0}$

Terminal vertices

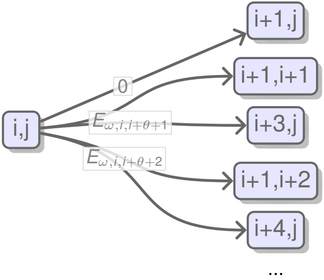
$$E_{\omega,i,j} = \begin{cases} 1 & \text{If } \omega_i \text{ base-pairs with } \omega_j \\ +\infty & \text{Otherwise} \end{cases}$$

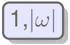
Energy function

Decomposition:

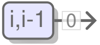


As an Hypergraph: $\Theta(n^2)$ vertices, $\Theta(n^3)$ arcs ($n = |\omega|$).





Initial vertex

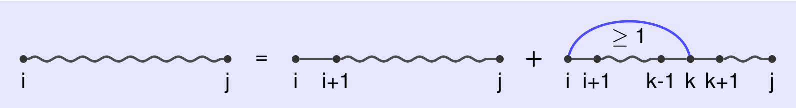


Terminal vertices

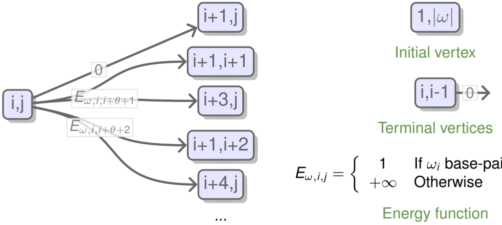
$$E_{\omega,i,j} = \begin{cases} 1 & \text{If } \omega_i \text{ base-pairs with } \omega_j \\ +\infty & \text{Otherwise} \end{cases}$$

Energy function

Decomposition:



As an Hypergraph: $\Theta(n^2)$ vertices, $\Theta(n^3)$ arcs ($n = |\omega|$).



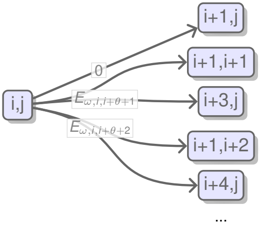
$$E_{\omega,i,j} = \begin{cases} 1 & \text{If } \omega_i \text{ base-pairs with } \omega_j \\ +\infty & \text{Otherwise} \end{cases}$$

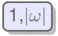
Energy function

Before applying generic ensemble algorithms, one needs to prove:

- Bijection between F-paths and secondary structures of length n. Halve the burden through generating functions (cf proceedings).
- Free-energy is reflected by weight/score.

As an Hypergraph: $\Theta(n^2)$ vertices, $\Theta(n^3)$ arcs ($n = |\omega|$).





Initial vertex

Terminal vertices

$$E_{\omega,i,j} = \begin{cases} 1 & \text{If } \omega_i \text{ base-pairs with } \omega_j \\ +\infty & \text{Otherwise} \end{cases}$$

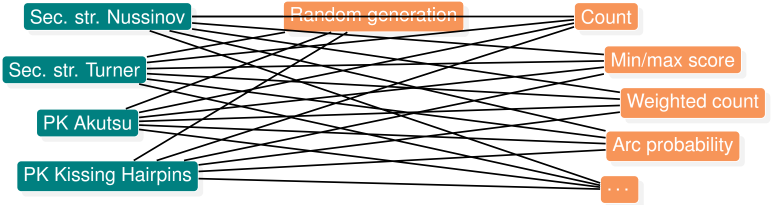
Energy function

Application	Algorithm	Feature fun.	Time/Space
Energy minimization	Minimal weight	E	$O(n^3)/O(n^2)$
Partition function	Weighted count	$e^{\frac{-E}{RT}}$	$O(n^3)/O(n^2)$
BP prob.	Arc-traversal prob.	$e^{\frac{-E}{RT}}$	$O(n^3)/O(n^2)$
Stat. sampling (k-samples)	Random gen.	$e^{\frac{-E}{RT}}$	$O(n^3 + kn\log n)$ $O(n^2)$

Half time summary

Message #2 (cf ADP)

Applications of DP could (and should) be detached from the equation, and be expressed at an abstract – combinatorial – level.



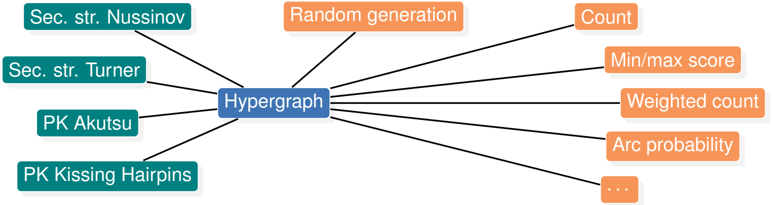
Credits: Roytberg and Finkelstein for Hypergraph DP in Bioinformatics, L. Hwang for algebraic hypergraph DP, R. Giegerich for ADP...

Let us extend applications of DP...

Half time summary

Message #2 (cf ADP)

Applications of DP could (and should) be detached from the equation, and be expressed at an abstract – combinatorial – level.

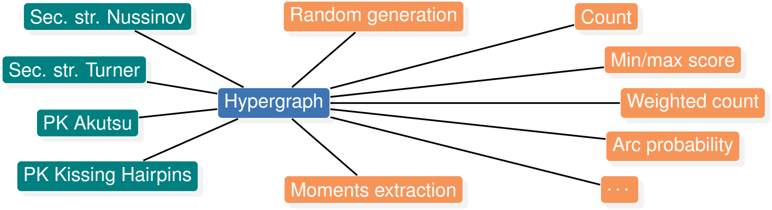


Credits: Roytberg and Finkelstein for Hypergraph DP in Bioinformatics, L. Hwang for algebraic hypergraph DP, R. Giegerich for ADP...

Let us extend applications of DP...

Message #2 (cf ADP)

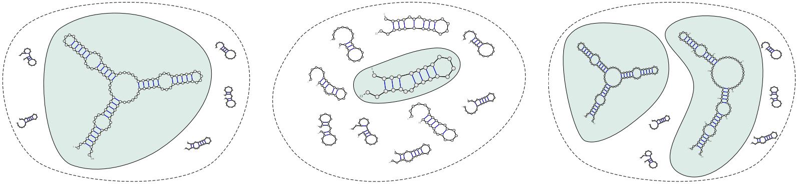
Applications of DP could (and should) be detached from the equation, and be expressed at an abstract – combinatorial – level.



Credits: Roytberg and Finkelstein for Hypergraph DP in Bioinformatics, L. Hwang for algebraic hypergraph DP, R. Giegerich for ADP...

Let us extend applications of DP...

Distribution of solutions



Functional folding? | Ill-defined folding: mRNA? | Bistable RNA Kinetics?

Toward ab-initio signatures for ncRNAs, one may have to go beyond average behavior of additive features (#Hairpins, #Multiple Loops, Free-Energy...).

Input:

- An acyclic F-graph $\mathcal{H}$, defining the set of F-paths (trees) in $\mathcal{H}$.
- Weight function $w : E \to \mathbb{R}$, defining a probability distribution.
- Additive feature functions $\alpha_1, \ldots, \alpha_k : E \to \mathbb{R}$.

What can be said about the (joint) distribution of features?
$\Rightarrow$ (Generalized) moments.

Generalized moments

Definition (Generalized moments)

$$\mathbb{E}[\alpha_1^{m_1}\alpha_2^{m_2}\cdots\alpha_k^{m_k}] = \sum_{p\in\mathcal{P}_s}\frac{\pi(p)}{w_s}\prod_{i=1}^{k}\alpha_i(p)^{m_i}$$

Remark: Single feature + $m_1 = 1 \Rightarrow$ Expectation in the weighted distribution.

Theorem (Generalized moments extraction (Generalizes Miklos *et al* 2005))

Generalized moments can be computed as $\mathbb{E}[\alpha_1^{m_1}\cdots\alpha_k^{m_k}] = c_s^{\mathbf{m}}/w_s$, *where*

$$c_s^{\mathbf{m}} = \sum_{e=(s\to\mathbf{t})}\pi(e)\cdot\sum_{\substack{\mathbf{m}',\left(\mathbf{m}''_1,\cdots,\mathbf{m}''_{|t|}\right)\\ s.\ t.\ \mathbf{m}'+\sum_j \mathbf{m}''_j=\mathbf{m}}}\prod_{i=1}^{k}\binom{m_i}{m'_i,\, m''_{1,i},\cdots,\, m''_{|t|,i}}\cdot\alpha_i(e)^{m'_i}\cdot\prod_{i=1}^{|t|}c_{t_i}^{\mathbf{m}''_i}$$

Time: $\mathcal{O}\left((|E|+|V|)\cdot k\cdot t^+\cdot\prod_{i=1}^{k} m_i^{t^++1}\right)$ Memory: $\Theta\left(|V|\cdot\prod_{i=1}^{k} m_i\right)$
(t^+ =max. out-degree).

Definition (Generalized moments)

$$\mathbb{E}[\alpha_1^{m_1}\alpha_2^{m_2}\cdots\alpha_k^{m_k}] = \sum_{p\in\mathcal{P}_s}\frac{\pi(p)}{w_s}\prod_{i=1}^{k}\alpha_i(p)^{m_i}$$

Remark: Single feature + $m_2 = 2$ characterizes Standard Deviation.

Theorem (Generalized moments extraction (Generalizes Miklos *et al* 2005))

Generalized moments can be computed as $\mathbb{E}[\alpha_1^{m_1}\cdots\alpha_k^{m_k}] = c_s^{\mathbf{m}}/w_s$*, where*

$$c_s^{\mathbf{m}} = \sum_{e=(s\to\mathbf{t})}\pi(e)\cdot\sum_{\substack{\mathbf{m}',\left(\mathbf{m}''_1,\cdots,\mathbf{m}''_{|t|}\right)\\ s.\ t.\ \mathbf{m}'+\sum_j\mathbf{m}''_j=\mathbf{m}}}\prod_{i=1}^{k}\binom{m_i}{m'_i,m''_{1,i},\cdots,m''_{|t|,i}}\cdot\alpha_i(e)^{m'_i}\cdot\prod_{i=1}^{|t|}c_{t_i}^{\mathbf{m}''_i}$$

Time: $\mathcal{O}\left((|E|+|V|)\cdot k\cdot t^+\cdot\prod_{i=1}^k m_i^{t^++1}\right)$ Memory: $\Theta\left(|V|\cdot\prod_{i=1}^k m_i\right)$
(t^+ =max. out-degree).

Definition (Generalized moments)

$$\mathbb{E}[\alpha_1^{m_1}\alpha_2^{m_2}\cdots\alpha_k^{m_k}] = \sum_{p\in\mathcal{P}_s}\frac{\pi(p)}{w_s}\prod_{i=1}^{k}\alpha_i(p)^{m_i}$$

Remark: Two features + $m_1 = m_2 = 1$ computes Pearson product-moment corr. coeff..

Theorem (Generalized moments extraction (Generalizes Miklos *et al* 2005))

Generalized moments can be computed as $\mathbb{E}[\alpha_1^{m_1}\cdots\alpha_k^{m_k}] = c_s^{\mathbf{m}}/w_s$, *where*

$$c_s^{\mathbf{m}} = \sum_{e=(s\to\mathbf{t})}\pi(e)\cdot\sum_{\substack{\mathbf{m}',\left(\mathbf{m}''_1,\cdots,\mathbf{m}''_{|t|}\right)\\ s.\ t.\ \mathbf{m}'+\sum_j\mathbf{m}''_j=\mathbf{m}}}\prod_{i=1}^{k}\binom{m_i}{m'_i,m''_{1,i},\cdots,m''_{|t|,i}}\cdot\alpha_i(e)^{m'_i}\cdot\prod_{i=1}^{|t|}c_{t_i}^{\mathbf{m}''_i}$$

Time: $\mathcal{O}\left((|E|+|V|)\cdot k\cdot t^+\cdot\prod_{i=1}^{k}m_i^{t^++1}\right)$ Memory: $\Theta\left(|V|\cdot\prod_{i=1}^{k}m_i\right)$ (t^+ =max. out-degree).

Definition (Generalized moments)

$$\mathbb{E}[\alpha_1^{m_1}\alpha_2^{m_2}\cdots\alpha_k^{m_k}] = \sum_{p\in\mathcal{P}_s}\frac{\pi(p)}{w_s}\prod_{i=1}^{k}\alpha_i(p)^{m_i}$$

Remark: Two features + $m_1 = m_2 = 1$ computes Pearson product-moment corr. coeff..

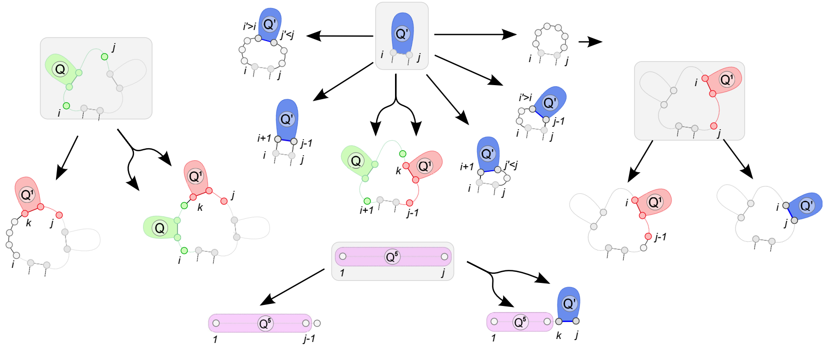
Theorem (Generalized moments extraction (Generalizes Miklos *et al* 2005))

Generalized moments can be computed as $\mathbb{E}[\alpha_1^{m_1}\cdots\alpha_k^{m_k}] = c_s^{\mathbf{m}}/w_s$, *where*

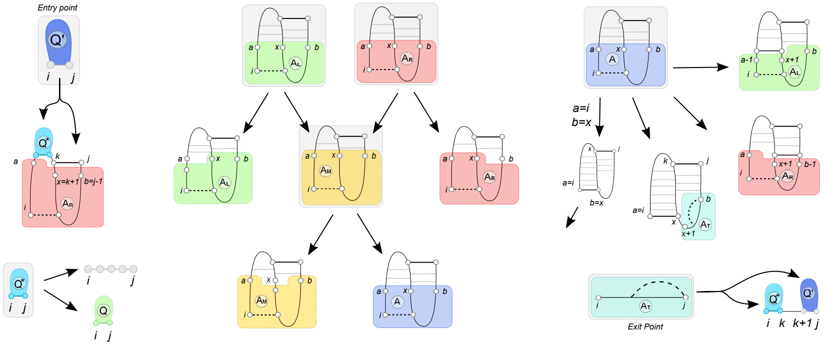
$$c_s^{\mathbf{m}} = \sum_{e=(s\to\mathbf{t})}\pi(e)\cdot\sum_{\substack{\mathbf{m}',\left(\mathbf{m}''_1,\cdots,\mathbf{m}''_{|t|}\right)\\ s.\ t.\ \mathbf{m}'+\sum_j\mathbf{m}''_j=\mathbf{m}}}\prod_{i=1}^{k}\binom{m_i}{m'_i, m''_{1,i},\cdots, m''_{|t|,i}}\cdot\alpha_i(e)^{m'_i}\cdot\prod_{i=1}^{|t|}c_{t_i}^{\mathbf{m}''_i}$$

Time: $\mathcal{O}\left((|E|+|V|)\cdot k\cdot t^{+}\cdot\prod_{i=1}^{k} m_i^{t^{+}+1}\right)$ Memory: $\Theta\left(|V|\cdot\prod_{i=1}^{k} m_i\right)$
(t^{+} =max. out-degree).

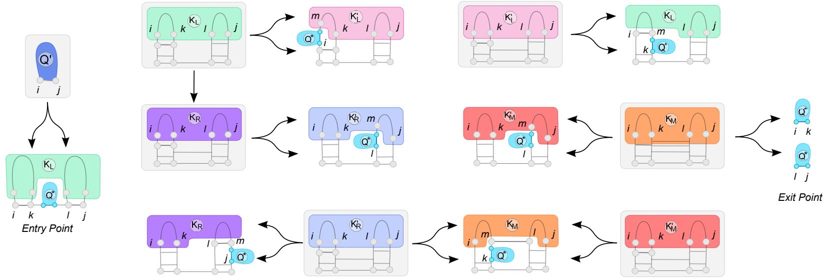
Mfold/Unafold decomposition



Application	Algorithm	Weight fun.	Time/Space	Ref.
Energy minimization	Minimal weight	$\pi_{\mathcal{T}}$	$O(n^{3(4)})/O(n^2)$	[Zuker and Stiegler, 1981]
Partition function	Weighted count	$e^{-\frac{\pi_{\mathcal{T}}}{RT}}$	$O(n^{3(4)})/O(n^2)$	[McCaskill, 1990]
Base-pairing probabilities	Arc-traversal prob.	$e^{-\frac{\pi_{\mathcal{T}}}{RT}}$	$O(n^{3(4)})/O(n^2)$	[McCaskill, 1990]
Statistical sampling (k-samples)	Random gen.	$e^{-\frac{\pi_{\mathcal{T}}}{RT}}$	$O(n^{3(4)} + kn\log n)/O(n^2)$	[Ding and Lawrence, 2003, Ponty, 2008]
Moments of energy (Mean, Var.)	Moments extraction	$e^{-\frac{\pi_{\mathcal{T}}}{RT}}$	$O(n^{3(4)})/O(n^2)$	[Miklós *et al.*, 2005]
m-th moment of additive features	Moments extraction	$e^{-\frac{\pi_{\mathcal{T}}}{RT}}$	$O(m^3 \cdot n^{3(4)})/O(m \cdot n^2)$	–
Correlations of additive features	Moments extraction	$e^{-\frac{\pi_{\mathcal{T}}}{RT}}$	$O(n^{3(4)})/O(n^2)$	–



Application	Algorithm	Weight fun.	Time/Space	Ref.
Energy minimization	Minimal weight	π_{bp}	$O(n^4)/O(n^4)$	[Akutsu, 2000]
Partition function	Weighted count	$e^{\frac{-\pi_{bp}}{RT}}$	$O(n^4)/O(n^4)$	$\Theta(n^6)$[Cao and Chen, 2009]
Base-pairing probabilities	Arc-traversal prob.	$e^{\frac{-\pi_{bp}}{RT}}$	$O(n^4)/O(n^4)$	–
Statistical sampling (k-samples)	Random gen.	$e^{\frac{-\pi_{bp}}{RT}}$	$O(n^4 + kn\log n)/O(n^4)$	–
Moments of energy (Mean, Var.)	Moments extraction	$e^{\frac{-\pi_{bp}}{RT}}$	$O(n^4)/O(n^4)$	–
m-th moment of additive features	Moments extraction	$e^{\frac{-\pi_{bp}}{RT}}$	$O(m^3 \cdot n^4)/O(m \cdot n^4)$	–



Application	Algorithm	Weight fun.	Time/Memory	Ref.
Energy minimization	Minimal weight	$\pi_{\mathcal{T}}$	$O(n^5)/O(n^4)$	[Chen *et al.*, 2009]
Partition function	Weighted count	$e^{\frac{-\pi_{\mathcal{T}}}{RT}}$	$O(n^5)O(n^4)$	–
Base-pairing probabilities	Arc-traversal prob.	$e^{\frac{-\pi_{\mathcal{T}}}{RT}}$	$O(n^5)/O(n^4)$	–
Statistical sampling (k-samples)	Random gen.	$e^{\frac{-\pi_{\mathcal{T}}}{RT}}$	$O(n^5 + k \cdot n \log n)/O(n^4)$	–
Moments of energy (Mean, Var.)	Moments extraction	$e^{\frac{-\pi_{\mathcal{T}}}{RT}}$	$O(n^5)/O(n^4)$	–
m-th moment of additive features	Moments extraction	$e^{\frac{-\pi_{\mathcal{T}}}{RT}}$	$O(m^3 \cdot n^5)/O(m \cdot n^4)$	–

Conclusion/perspectives

- Implementation issues: Table design, avoid memory consumption, compilation to low-level language. . .
- Generate hypergraph from more abstract description (CFG, Möhl's *split-types*, Nebel's algebraic descriptors)
- Novel sequence-only features $\Rightarrow$ Thermodynamic signatures for ncRNAs, Riboswitches, Pseudoknotted RNAs classifier. . . ?
- Formulate generic optimizations: Sparsification, four-russians. . .
- Extensions: Predicting interactions, RNA design, Simultaneous folding and alignment

Thanks to

Agence Nationale de la Recherche ANR

References I

Tatsuya Akutsu.
Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots.
Discrete Appl. Math., 104(1-3):45–62, 2000.

S. Cao and S. J. Chen.
Predicting RNA pseudoknot folding thermodynamics.
Nucleic Acids Res, 34(9):2634–2652, 2006.

S. Cao and S-J Chen.
Predicting structured and stabilities for H-type pseudoknots with interhelix loop.
RNA, 15:696–706, 2009.

Ho-Lin Chen, Anne Condon, and Hosna Jabbari.
An O(n(5)) algorithm for MFE prediction of kissing hairpins and 4-chains in nucleic acids.
Journal of Computational Biology, 16(6):803–815, 2009.

Y. Ding and E. Lawrence.
A statistical sampling algorithm for RNA secondary structure prediction.
Nucleic Acids Res, 31(24):7280–7301, 2003.

R.M. Dirks and N.A. Pierce.
A partition function algorithm for nucleic acid secondary structure including pseudoknots.
J Comput Chem, 24:1664–1677, 2003.

Fenix W D Huang, Wade W J Peng, and Christian M Reidys.
Folding 3-noncrossing rna pseudoknot structures.
J Comput Biol, 16(11):1549–1575, Nov 2009.

R. B. Lyngsø and C. N. S. Pedersen.
RNA pseudoknot prediction in energy-based models.
Journal of Computational Biology, 7(3-4):409–427, 2000.

References II

D. H. Mathews.
Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization.
RNA, 10(8):1178–1190, 2004.

J.S. McCaskill.
The equilibrium partition function and base pair binding probabilities for RNA secondary structure.
Biopolymers, 29:1105–1119, 1990.

István Miklós, Irmtraud M Meyer, and Borbála Nagy.
Moments of the boltzmann distribution for RNA secondary structures.
Bull Math Biol, 67(5):1031–1047, Sep 2005.

R. Nussinov and A. B. Jacobson.
Fast algorithm for predicting the secondary structure of single stranded RNA.
Proc. Natl. Acad. Sci. U. S. A., 77(11):6309–6313, 1980.

Y. Ponty.
Efficient sampling of RNA secondary structures from the boltzmann ensemble of low-energy: The boustrophedon method.
J Math Biol, 56(1-2):107–127, Jan 2008.

J. Reeder and R. Giegerich.
Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics.
BMC Bioinformatics, 5:104, 2004.

Christian M Reidys, Fenix W D Huang, Jørgen E Andersen, Robert C Penner, Peter F Stadler, and Markus E Nebel.
Topology and prediction of rna pseudoknots.
Bioinformatics, 27(8):1076–1085, Apr 2011.

E. Rivas and S.R. Eddy.
A dynamic programming algorithm for RNA structure prediction including pseudoknots.
J Mol Biol, 285:2053–2068, 1999.

References III

J. E. Tabaska, R. B. Cary, H. N. Gabow, and G. D. Stormo.
An rna folding method capable of identifying pseudoknots and base triples.
Bioinformatics, 14(8):691–699, 1998.

Corinna Theis, Stefan Janssen, and Robert Giegerich.
Prediction of rna secondary structure including kissing hairpin motifs.
In *Proceedings of WABI 2010*, pages 52–64, 2010.

Björn Voss, Carsten Meyer, and Robert Giegerich.
Evaluating the predictability of conformational switching in rna.
Bioinformatics, 20(10):1573–1582, Jul 2004.

M. Zuker and P. Stiegler.
Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information.
Nucleic Acids Res, 9:133–148, 1981.