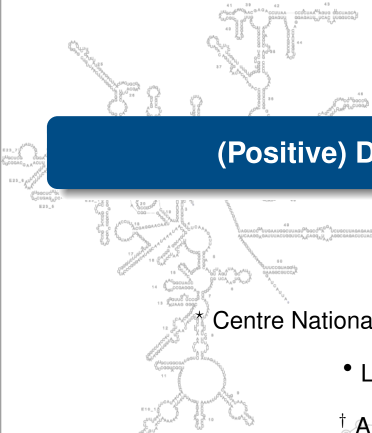
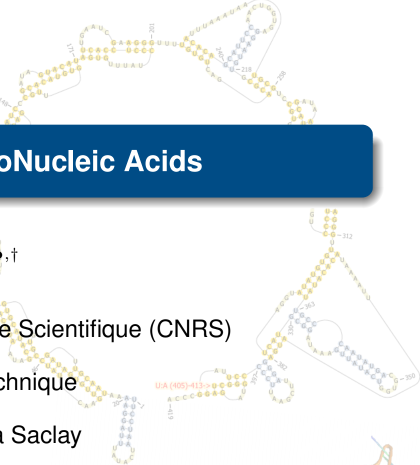




(Positive) Design of RiboNucleic Acids

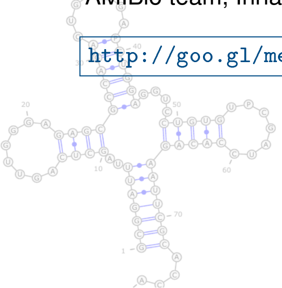
Yann Ponty[⋆,•,†]

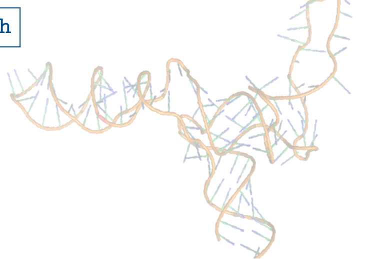
[⋆] Centre National de la Recherche Scientifique (CNRS)

[•] LIX, Ecole Polytechnique

[†] AMIBio team, Inria Saclay

http://goo.gl/mejsFh





Yann Ponty

AmirKabir
University of Technology

Computational Biology
Research Center

PhD in Computer Science, Université Paris-Sud (France)

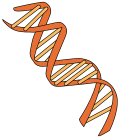
- CNRS research scientist
- Faculty at LIX, Computer Science department of Ecole Polytechnique

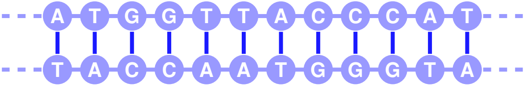
- Group leader – AMIBio team (Ecole Polytechnique and Inria Saclay)
- Postdoc experience in RNA Computational Biology (Boston, Paris) and Discrete Mathematics (Paris)
- Extended sabbatical at Simon Fraser University (Vancouver, Canada)

Fundamental *dogma* of molecular biology

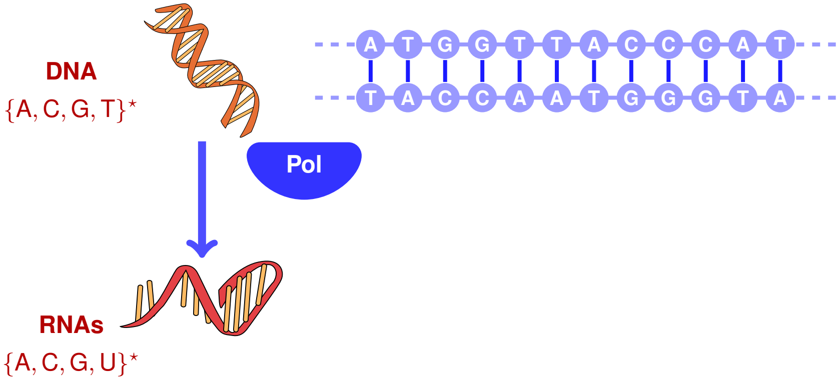
DNA

$\{A, C, G, T\}^\star$





Fundamental *dogma* of molecular biology



DNA

$\{A, C, G, T\}^\star$

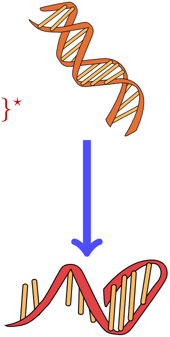
RNAs

$\{A, C, G, U\}^\star$

Fundamental *dogma* of molecular biology

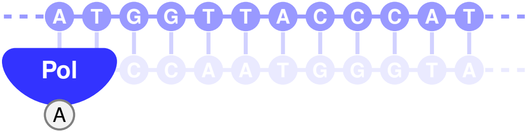
DNA

$\{A, C, G, T\}^\star$



RNAs

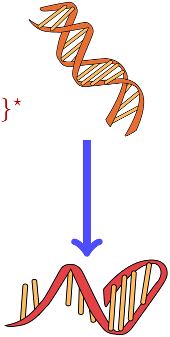
$\{A, C, G, U\}^\star$



Fundamental *dogma* of molecular biology

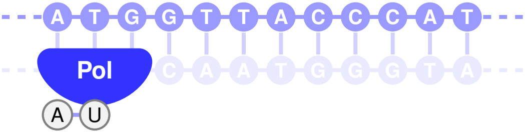
DNA

$\{A, C, G, T\}^\star$



RNAs

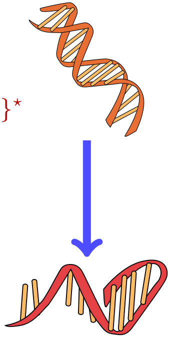
$\{A, C, G, U\}^\star$



Fundamental *dogma* of molecular biology

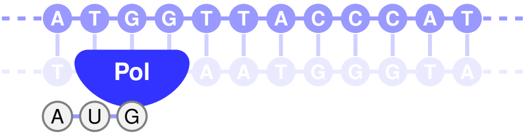
DNA

$\{A, C, G, T\}^\star$



RNAs

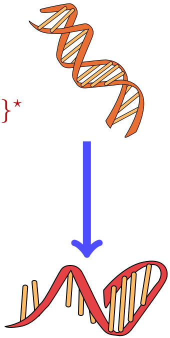
$\{A, C, G, U\}^\star$



Fundamental *dogma* of molecular biology

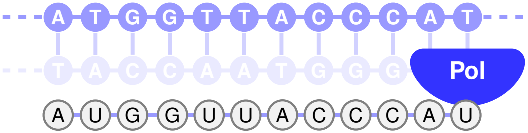
DNA

$\{A, C, G, T\}^\star$



RNAs

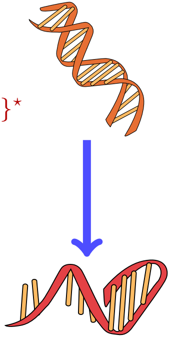
$\{A, C, G, U\}^\star$

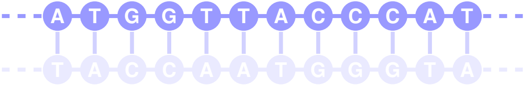


Fundamental *dogma* of molecular biology

DNA

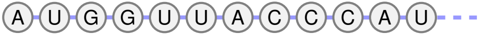
$\{A, C, G, T\}^*$



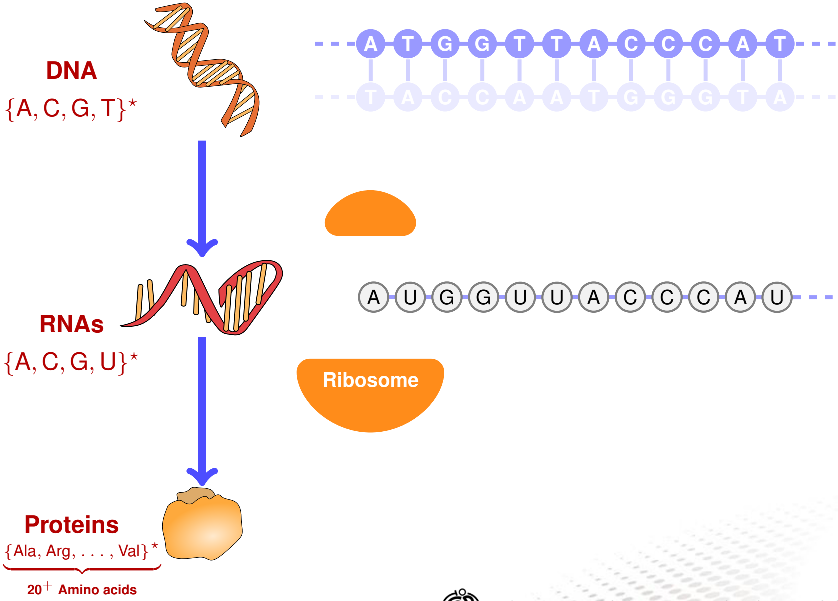


RNAs

$\{A, C, G, U\}^*$



Fundamental *dogma* of molecular biology



DNA

$\{A, C, G, T\}^\star$

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

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

RNAs

$\{A, C, G, U\}^\star$

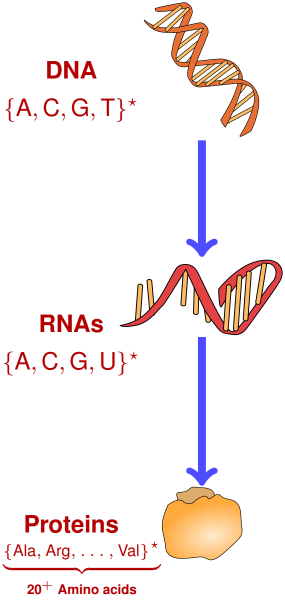
A U G G U U A C C C A U

Ribosome

Proteins

$\underbrace{\{\text{Ala}, \text{Arg}, \ldots, \text{Val}\}^\star}_{20^+ \text{ Amino acids}}$

Fundamental *dogma* of molecular biology



DNA

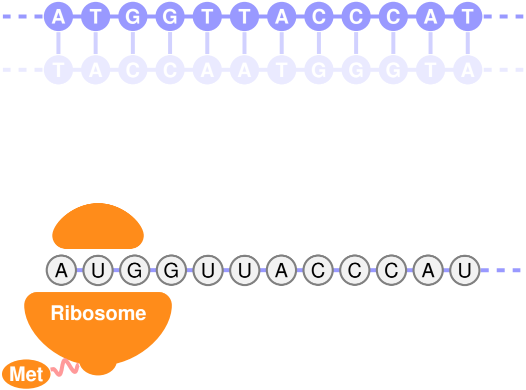
$\{A, C, G, T\}^\star$

RNAs

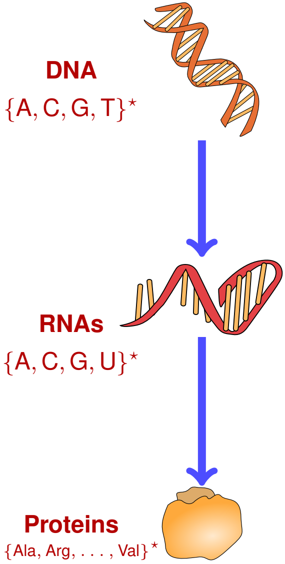
$\{A, C, G, U\}^\star$

Proteins

$\underbrace{\{\text{Ala}, \text{Arg}, \ldots, \text{Val}\}^\star}_{20^+ \text{ Amino acids}}$



Fundamental *dogma* of molecular biology



DNA

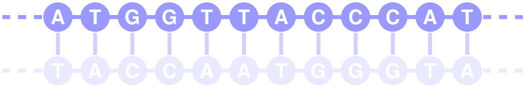
$\{A, C, G, T\}^\star$

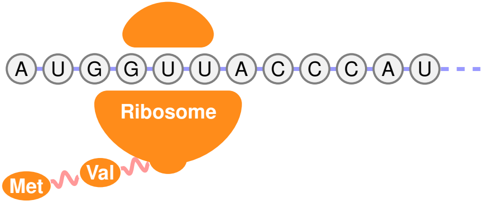
RNAs

$\{A, C, G, U\}^\star$

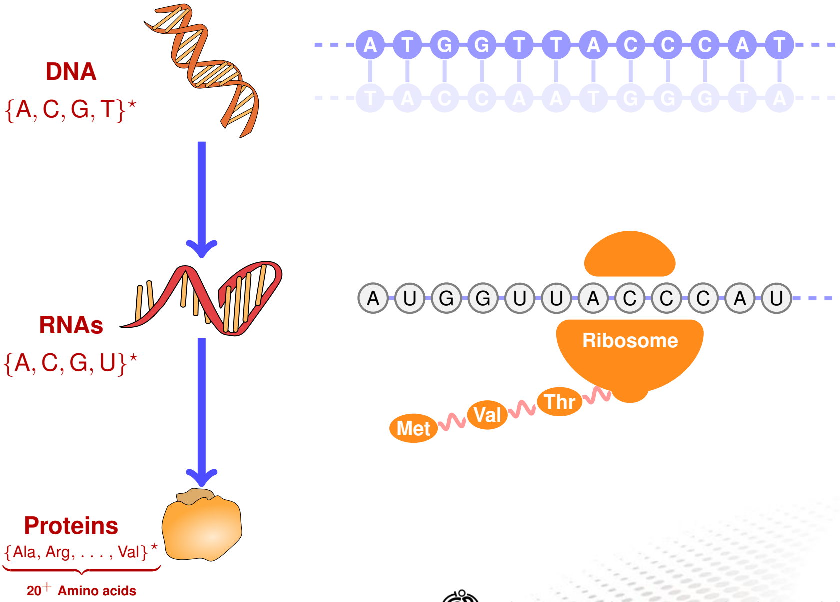
Proteins

$$\underbrace{\{\text{Ala}, \text{Arg}, \ldots, \text{Val}\}^\star}_{20^+\ \textbf{Amino acids}}$$





Fundamental *dogma* of molecular biology



DNA

$\{A, C, G, T\}^\star$

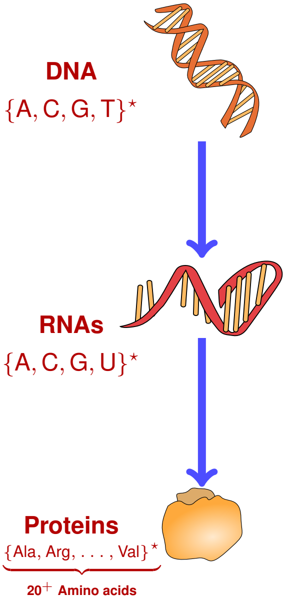
RNAs

$\{A, C, G, U\}^\star$

Proteins

$\underbrace{\{\mathrm{Ala}, \mathrm{Arg}, \ldots, \mathrm{Val}\}^\star}_{20^+ \text{ Amino acids}}$

Fundamental *dogma* of molecular biology



DNA

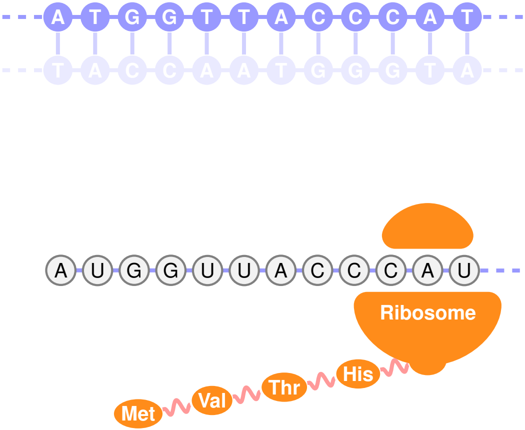
$\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}^\star$

RNAs

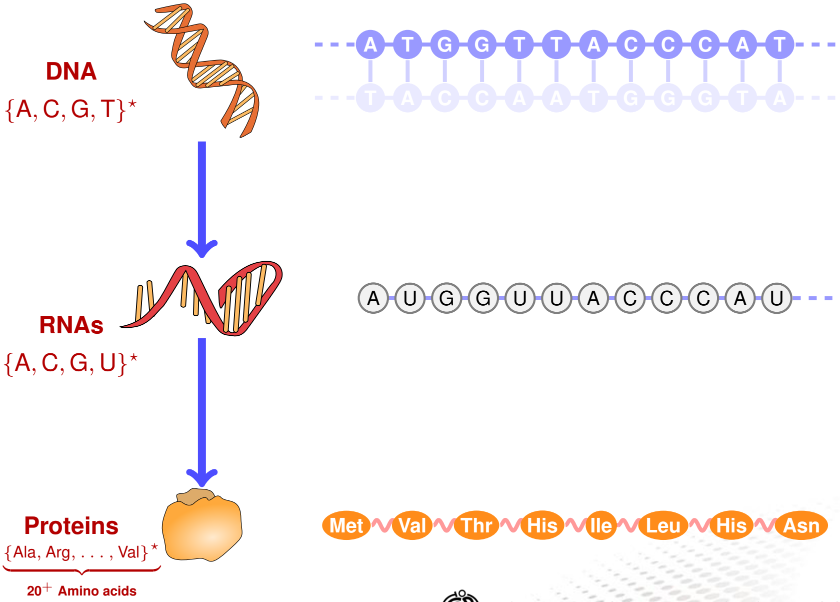
$\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}^\star$

Proteins

$\underbrace{\{\mathrm{Ala}, \mathrm{Arg}, \ldots, \mathrm{Val}\}^\star}_{20^+ \text{ Amino acids}}$



Fundamental *dogma* of molecular biology



DNA

$\{A, C, G, T\}^\star$

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

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

RNAs

$\{A, C, G, U\}^\star$

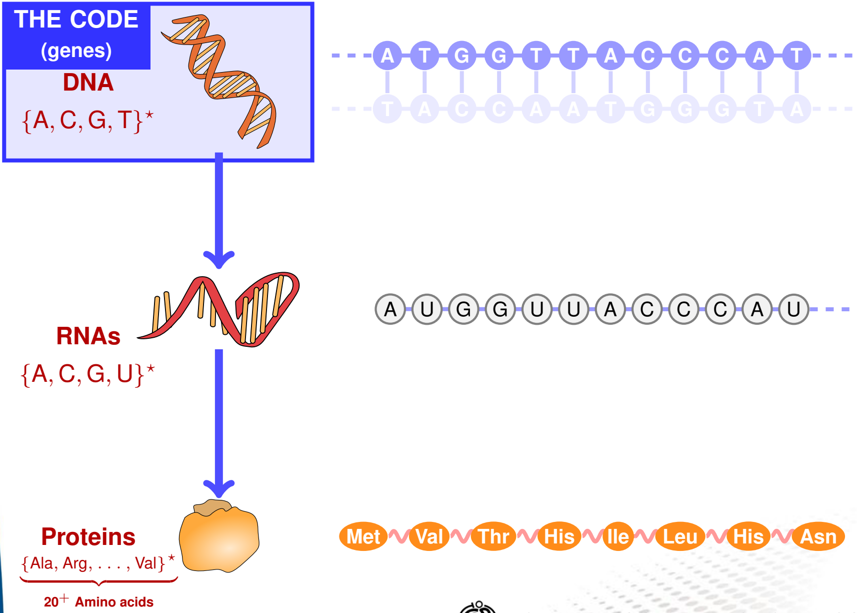
A U G G U U A C C C A U

Proteins

$\underbrace{\{\mathrm{Ala}, \mathrm{Arg}, \ldots, \mathrm{Val}\}^\star}_{20^+ \text{ Amino acids}}$

Met Val Thr His Ile Leu His Asn

Fundamental *dogma* of molecular biology



THE CODE (genes)

DNA

$\{A, C, G, T\}^*$

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

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

RNAs

$\{A, C, G, U\}^*$

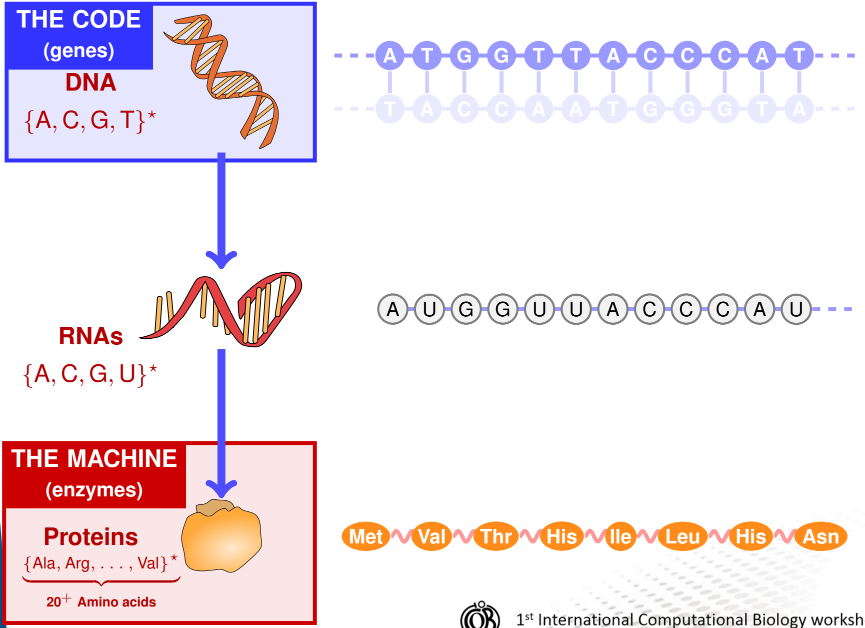
A U G G U U A C C C A U

Proteins

$\underbrace{\{Ala, Arg, \ldots, Val\}^*}_{20^+ \text{ Amino acids}}$

Met Val Thr His Ile Leu His Asn

Fundamental *dogma* of molecular biology



THE CODE (genes)

DNA

$\{A, C, G, T\}^\star$

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

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

RNAs

$\{A, C, G, U\}^\star$

A U G G U U A C C C A U ---

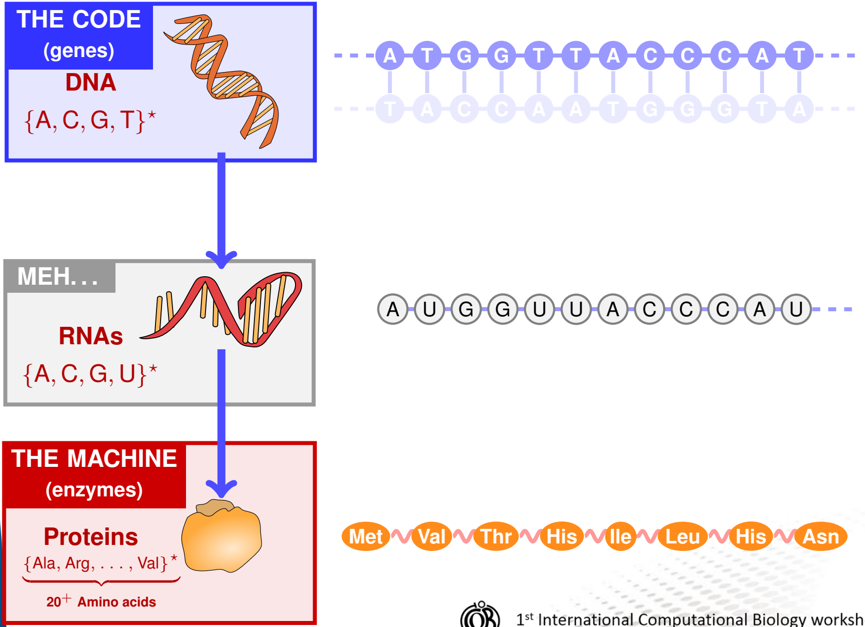
THE MACHINE (enzymes)

Proteins

$\underbrace{\{\text{Ala}, \text{Arg}, \ldots, \text{Val}\}^\star}_{20^{+}\ \text{Amino acids}}$

Met Val Thr His Ile Leu His Asn

Fundamental *dogma* of molecular biology



THE CODE
(genes)

DNA

$\{A, C, G, T\}^\star$

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

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

MEH. . .

RNAs

$\{A, C, G, U\}^\star$

A U G G U U A C C C A U

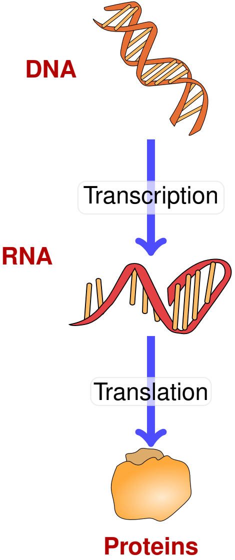
THE MACHINE
(enzymes)

Proteins

$\underbrace{\{\text{Ala, Arg, . . . , Val}\}^\star}_{\textbf{20}^+\ \textbf{Amino acids}}$

Met Val Thr His Ile Leu His Asn

Fundamental *dogma* of molecular biology



DNA

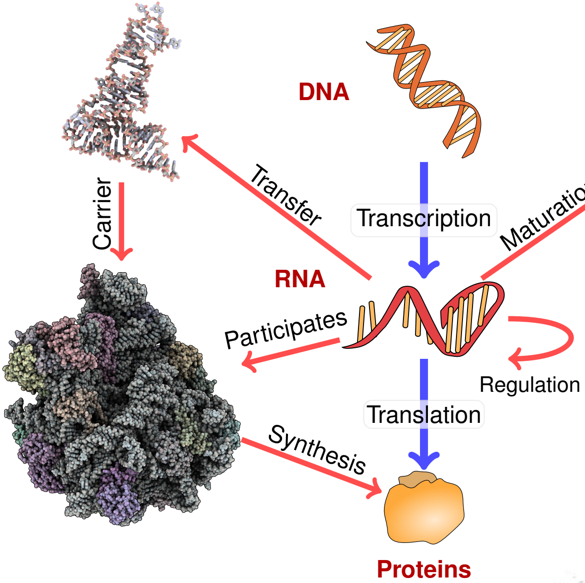
Transcription

RNA

Translation

Proteins

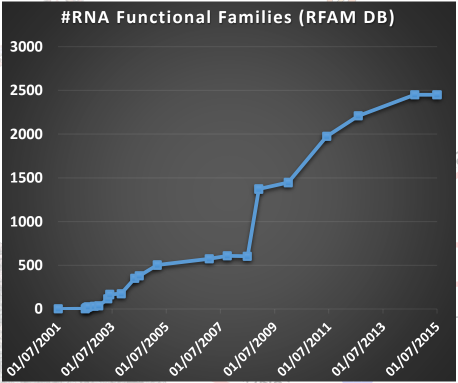
Fundamental *dogma* of molecular biology (v2.0)



RNA functions

- Messenger
- Translation
- Regulation
- Enzyme
- Catalytic
- . . .

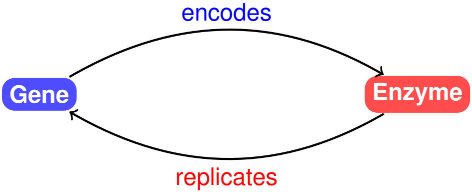
Fundamental *dogma* of molecular biology



RNA functions

- Messenger
- Translation
- Regulation
- Enzyme
- Catalytic
- . . .

RNA world: Resolving the *chicken vs egg* paradox at the origin of life...

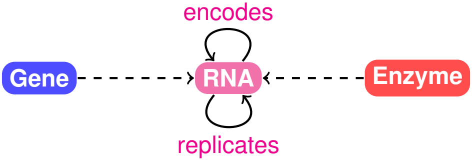


A gene big enough to specify an enzyme would be too big to replicate accurately without the aid of an enzyme of the very kind that it is trying to specify. So the system *apparently cannot get started*.

[· · ·] This is the RNA World. To see how plausible it is, we need to look at why proteins are good at being enzymes but bad at being replicators; at why DNA is good at replicating but bad at being an enzyme; and finally why *RNA might just be good enough at both roles to break out of the Catch-22.*

R. Dawkins. *The Ancestor's Tale: A Pilgrimage to the Dawn of Evolution*

RNA world: Resolving the *chicken vs egg* paradox at the origin of life. . .



A gene big enough to specify an enzyme would be too big to replicate accurately without the aid of an enzyme of the very kind that it is trying to specify. So the system *apparently cannot get started*.

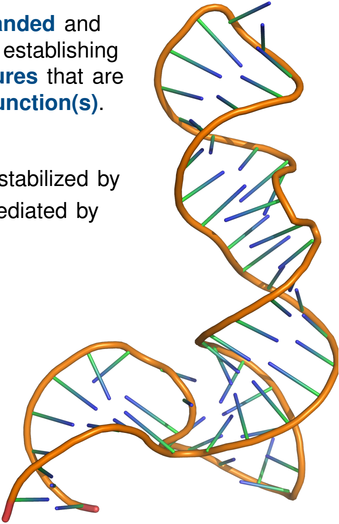
[· · ·] This is the RNA World. To see how plausible it is, we need to look at why proteins are good at being enzymes but bad at being replicators; at why DNA is good at replicating but bad at being an enzyme; and finally why *RNA might just be good enough at both roles to break out of the Catch-22.*

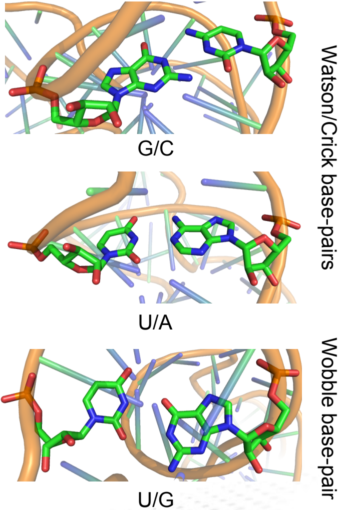
R. Dawkins. *The Ancestor's Tale: A Pilgrimage to the Dawn of Evolution*

RNA folding

RNA are **single-stranded** and **fold** on themselves, establishing **complex 3D structures** that are **essential to their function(s)**.

RNA structures are stabilized by **base-pairs**, each mediated by **hydrogen bonds**.





Canonical base-pairs

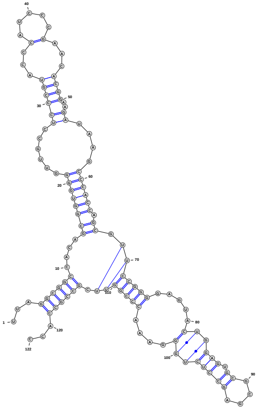
RNA Design

RNA = Linear Polymer = Sequence in $\{A, C, G, U\}^\star$

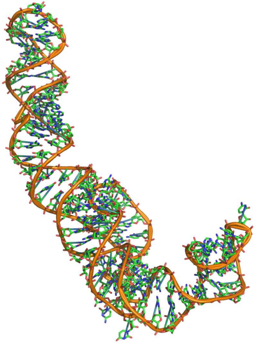
```
UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA

CC
```

Primary Structure



Secondary Structure



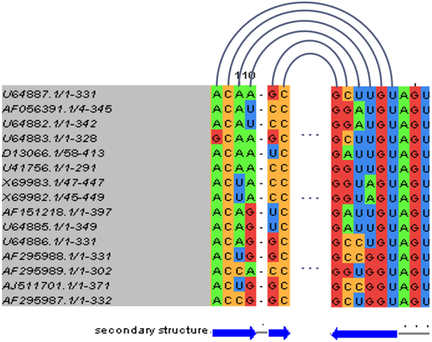
Structure Tertiaire

5s rRNA (PDBID: 1K73:B)

Evolution of RNAs

Homologous genes = **Functionally** equivalent, within or across organisms
Usually well-captured by **sequence similarity** in proteins, binding sites. . .

Problem: Many classes of non-(protein) coding RNAs (ncRNAs) poorly conserved at the sequence level **but** adopt a **conserved structure!**



```
U64887.1/1-331     ACAA-GC      GCUUGUAGU
AF056391.1/4-345   ACAU-CC      GGAUGUAGU
U64882.1/1-342     ACAU-CC      GGAUGUAGU
U64883.1/1-328     GCAA-GC ...  GCUUGUAGU
D13066.1/58-413    ACAA-UC      GAUUGUAGU
U41756.1/1-291     ACAA-CC      GGUUGUAGU
X69983.1/47-447    ACUA-CC      GGUAGUAGU
X69982.1/45-449    ACUA-CC ...  GGUAGUAGU
AF151218.1/1-397   ACAG-UC      GAUUGUAGU
U64885.1/1-349     ACAG-UC      GAUUGUAGU
U64886.1/1-331     ACAG-GC      GCCUGUAGU
AF295988.1/1-331   ACUG-GC      GCCGGUAGU
AF295989.1/1-302   ACCA-CC ...  GGUGGUAGU
AJ511701.1/1-371   ACUG-GC      GCCGGUAGU
AF295987.1/1-332   ACCG-GC      GCUGGUAGU
```

RFAM Bacterial RNAse-P class B Alignment

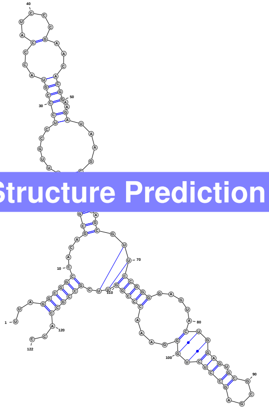
RNA Design

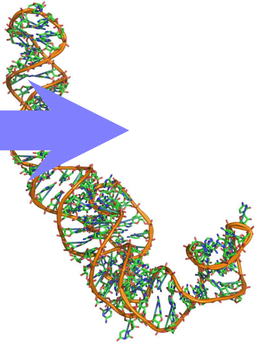
RNA = Linear Polymer = Sequence in $\{A, C, G, U\}^\star$

```
UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA

CC
```

Structure Prediction





Primary Structure

Secondary Structure

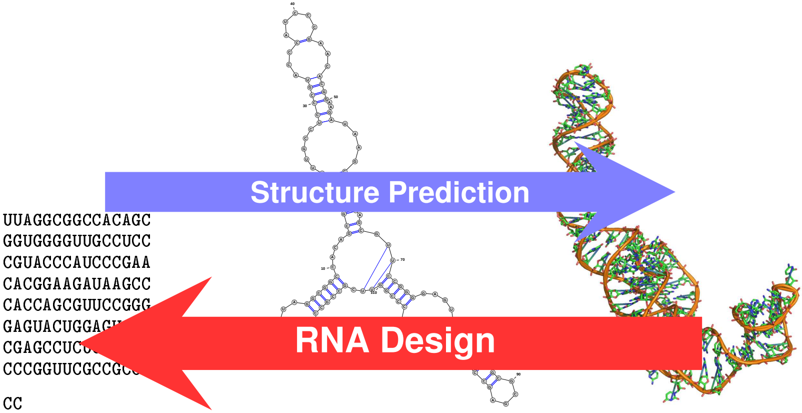
Structure Tertiaire

5s rRNA (PDBID: 1K73:B)

1st International Computational Biology workshop

RNA Design

RNA = Linear Polymer = Sequence in $\{A, C, G, U\}^\star$



UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGU
CGAGCCUCC
CCCGGUUCGCCGCC

CC

Primary Structure — Secondary Structure — Structure Tertiaire

5s rRNA (PDBID: 1K73:B)

Why we design RNAs

- To create building blocks for synthetic systems
  Rationally-designed RNAs increase orthogonality
- To assess the significance of observed phenomenon
  Random models should include every established characters. . .
  . . . including adoption of a single structure
- To test/push our understanding of how RNA folds
  Misfolding RNAs reveal gaps in our energy models and descriptors for the conformational spaces
- To help search for homologous sequences
  Incomplete covariance models hindered by limited training sets
  Design can be used to generalize existing alignments
- To fuel RNA-based therapeutics
  Sequence-based (siRNA, synthetic genes), but structure matters
- To perform controlled experiments

Why we design RNAs

- To create building blocks for synthetic systems
  Rationally-designed RNAs increase orthogonality
- To assess the significance of observed phenomenon
  Random models should include every established characters. . .
  . . . including adoption of a single structure
- To test/push our understanding of how RNA folds
  Misfolding RNAs reveal gaps in our energy models and descriptors for the conformational spaces
- To help search for homologous sequences
  Incomplete covariance models hindered by limited training sets
  Design can be used to generalize existing alignments
- To fuel RNA-based therapeutics
  Sequence-based (siRNA, synthetic genes), but structure matters
- To perform controlled experiments

Why we design RNAs

- To create building blocks for synthetic systems
  Rationally-designed RNAs increase orthogonality

- To assess the significance of observed phenomenon
  Random models should include every established characters. . .
  . . . including adoption of a single structure

- To test/push our understanding of how RNA folds
  Misfolding RNAs reveal gaps in our energy models and descriptors for the conformational spaces

- To help search for homologous sequences
  Incomplete covariance models hindered by limited training sets
  Design can be used to generalize existing alignments

- To fuel RNA-based therapeutics
  Sequence-based (siRNA, synthetic genes), but structure matters

- To perform controlled experiments

Why we design RNAs

- To create building blocks for synthetic systems
  Rationally-designed RNAs increase orthogonality

- To assess the significance of observed phenomenon
  Random models should include every established characters. . .
  . . . including adoption of a single structure

- To test/push our understanding of how RNA folds
  Misfolding RNAs reveal gaps in our energy models and descriptors for the conformational spaces

- To help search for homologous sequences
  Incomplete covariance models hindered by limited training sets
  Design can be used to generalize existing alignments

- To fuel RNA-based therapeutics
  Sequence-based (siRNA, synthetic genes), but structure matters

- To perform controlled experiments

Why we design RNAs

- To create building blocks for synthetic systems
  Rationally-designed RNAs increase orthogonality
- To assess the significance of observed phenomenon
  Random models should include every established characters. . .
  . . . including adoption of a single structure
- To test/push our understanding of how RNA folds
  Misfolding RNAs reveal gaps in our energy models and descriptors for the conformational spaces
- To help search for homologous sequences
  Incomplete covariance models hindered by limited training sets
  Design can be used to generalize existing alignments
- To fuel RNA-based therapeutics
  Sequence-based (siRNA, synthetic genes), but structure matters
- To perform controlled experiments

Why we design RNAs

- To create building blocks for synthetic systems
  Rationally-designed RNAs increase orthogonality
- To assess the significance of observed phenomenon
  Random models should include every established characters. . .
  . . . including adoption of a single structure
- To test/push our understanding of how RNA folds
  Misfolding RNAs reveal gaps in our energy models and descriptors for the conformational spaces
- To help search for homologous sequences
  Incomplete covariance models hindered by limited training sets
  Design can be used to generalize existing alignments
- To fuel RNA-based therapeutics
  Sequence-based (siRNA, synthetic genes), but structure matters
- To perform controlled experiments

Controlled experiments through RNA design

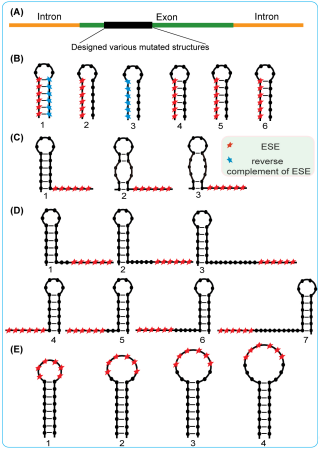
Motivation: Quantifying the impact of structure S on efficacy of a single Exon Splicing Enhancers (ESE):

- Presence of ESE motif E;
- Different structures $S_1, S_2 \ldots$;
- Avoid library of ($\sim$1500!) documented ESEs motifs.

Objectives. Design RNA which:

1. Folds into a prescribed structure;
2. Features/avoids motifs.
3. Control GC%, Boltz. prob.. . . .

Structural context of ESE motif in transcript was shown to affect its functionality. [Liu *et al*, FEBS Lett. 2010]


(A)
Intron
Exon
Intron
Designed various mutated structures
(B)
1
2
3
4
5
6
(C)
1
2
3
ESE
reverse complement of ESE
(D)
1
2
3
4
5
6
7
(E)
1
2
3
4

Design objectives

Positive structural design

Optimize **affinity** of designs towards target structure(s)

Examples: Most stable sequence for given fold. . .

Negative structural design

Limit affinity of designs towards **alternative structures**

Examples: Lowest free-energy, High Boltzmann probability/Low entropy. . .

Additional constraints:

- **Forbid** motif list to appear **anywhere** in design
- **Force** motif list to appear **each at least once**
- **Limit** available alternatives at certain positions
- **Control** overall composition (GC-content)

Outline

- I. Single Structure Design (IncaRNAtion)
- II. Constrained Design using Formal Languages
- III. Multiple Structures

I. Inverse Folding

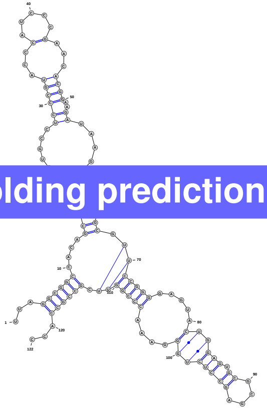
Designing a given structure

RNA sequence and structure(s)

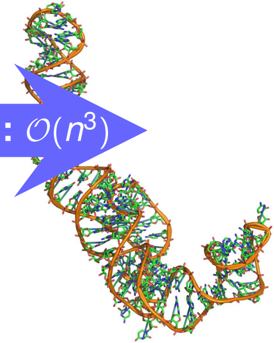
RNA = Linear Polymer = Sequence in $\{A, C, G, U\}^\star$

UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA

CC



MFE folding prediction: $\mathcal{O}(n^3)$



Primary Structure

Secondary Structure

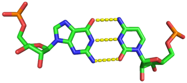
Tertiary Structure

5s rRNA (PDBID: 1K73:B)

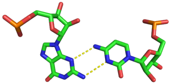
Crossing interactions

Excluded from the secondary structure:

- **Non-canonical base-pairs:**
  Any base-pair **other than** {(A-U), (C-G), (G-U)}
  OR interacting in a non-standard way (WC/WC-Cis) **[Leontis Westhof, RNA 2001]**.

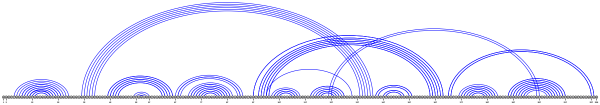


Canonical CG base-pair (WC/WC-Cis)



Non-canonical base-pair (Sugar/WC-Trans)

- **(Pseudo?)knots:** Crossing sets of nested stable base-pairs

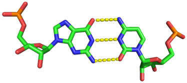


Group I Ribozyme (PDBID: 1Y0Q:A)

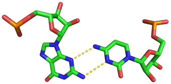
Crossing interactions

Excluded from the secondary structure:

- **Non-canonical base-pairs:**
  Any base-pair **other than** {(A-U), (C-G), (G-U)}
  OR interacting in a non-standard way (WC/WC-Cis) **[Leontis Westhof, RNA 2001]**.

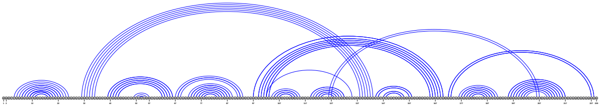


Canonical CG base-pair (WC/WC-Cis)



Non-canonical base-pair (Sugar/WC-Trans)

- **(Pseudo?)knots:** Crossing sets of nested stable base-pairs

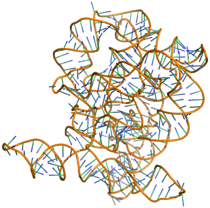


Group I Ribozyme (PDBID: 1Y0Q:A)

Crossing interactions

Excluded from the secondary structure:

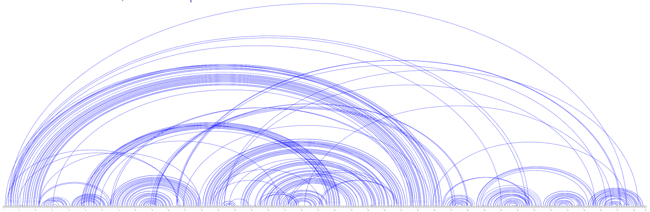
- **Non-canonical base-pairs:**
  Any
  OR
  200



Crossing interactions **do exist**!

Example: Group II Intron (PDB ID: 3IGI)

But are **hard** to predict
[Lyngsoe-ICALP'04]
[Sheikh Backofen Ponty, CPM'12]

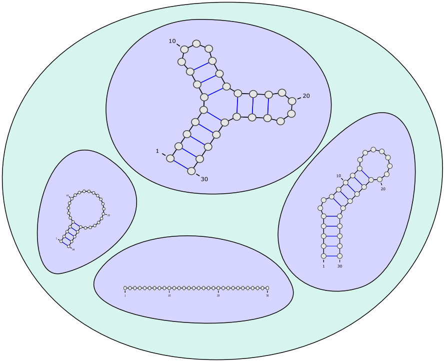


- **(Ps**

Thermodynamics vs Kinetics

Paradigms for RNA structure prediction

- **1978–1990s** Most probable structure = Minimal Free-Energy (MFE)
- **1990s–2010s** Functional structure(s) = Boltzmann ensemble (partition function)
- **2010s–????** Embracing the kinetics of RNA folding



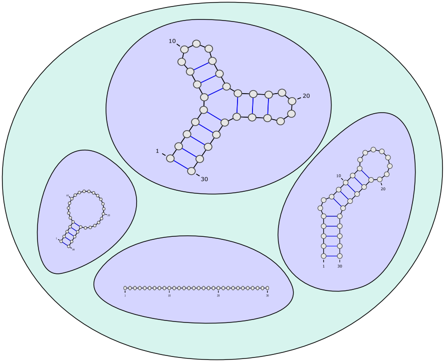
$T \to \infty$

mRNA half-life: ∼7h
(Mouse [Sharova2009])

Thermodynamics vs Kinetics

Paradigms for RNA structure prediction

- **1978–1990s** Most probable structure = Minimal Free-Energy (MFE)
- **1990s–2010s** Functional structure(s) = Boltzmann ensemble (partition function)
- **2010s–????** Embracing the kinetics of RNA folding



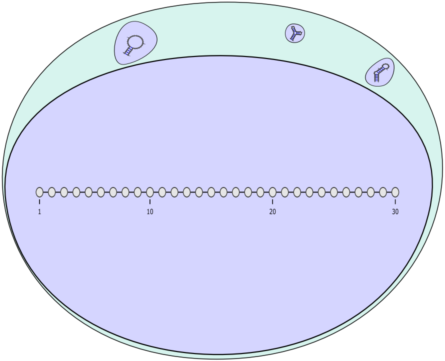
$T \to \infty$

mRNA half-life: $\sim$7h (Mouse [Sharova2009])

Thermodynamics vs Kinetics

Paradigms for RNA structure prediction

- **1978–1990s** Most probable structure = Minimal Free-Energy (MFE)
- **1990s–2010s** Functional structure(s) = Boltzmann ensemble (partition function)
- **2010s–????** Embracing the kinetics of RNA folding



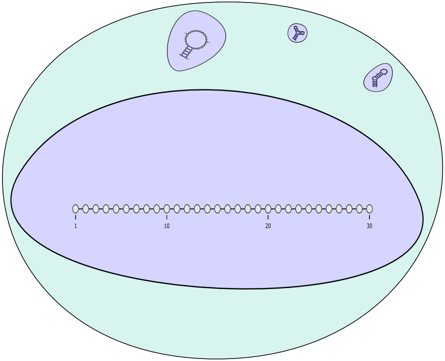
$T = 0\text{h}$

mRNA half-life: $\sim$7h
(Mouse [Sharova2009])

Thermodynamics vs Kinetics

Paradigms for RNA structure prediction

- **1978–1990s** Most probable structure = Minimal Free-Energy (MFE)
- **1990s–2010s** Functional structure(s) = Boltzmann ensemble (partition function)
- **2010s–????** Embracing the kinetics of RNA folding



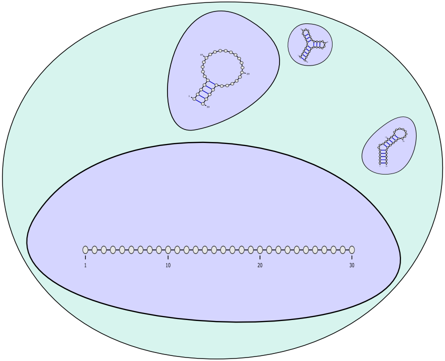
$T = 1\text{h}$

mRNA half-life: $\sim$7h (Mouse [Sharova2009])

Thermodynamics vs Kinetics

Paradigms for RNA structure prediction

- **1978–1990s** Most probable structure = Minimal Free-Energy (MFE)
- **1990s–2010s** Functional structure(s) = Boltzmann ensemble (partition function)
- **2010s–????** Embracing the kinetics of RNA folding



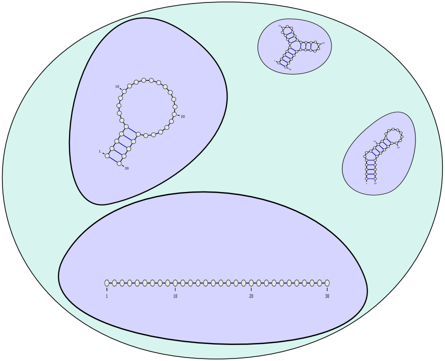
$T = 2\text{h}$

mRNA half-life: $\sim$7h (Mouse [Sharova2009])

Thermodynamics vs Kinetics

Paradigms for RNA structure prediction

- **1978–1990s** Most probable structure = Minimal Free-Energy (MFE)
- **1990s–2010s** Functional structure(s) = Boltzmann ensemble (partition function)
- **2010s–????** Embracing the kinetics of RNA folding



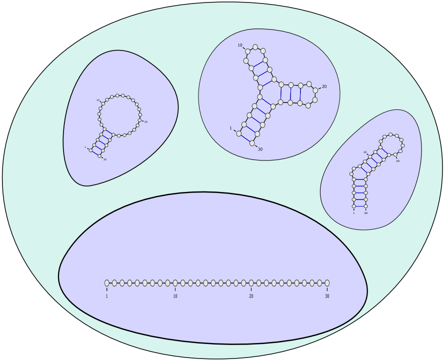
$T = 5\text{h}$

mRNA half-life: ~7h (Mouse [Sharova2009])

Thermodynamics vs Kinetics

Paradigms for RNA structure prediction

- **1978–1990s** Most probable structure = Minimal Free-Energy (MFE)
- **1990s–2010s** Functional structure(s) = Boltzmann ensemble (partition function)
- **2010s–????** Embracing the kinetics of RNA folding



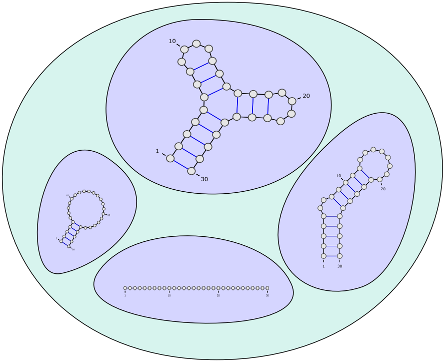
$T = 10\text{h}$

mRNA half-life: ~7h (Mouse [Sharova2009])

Thermodynamics vs Kinetics

Paradigms for RNA structure prediction

- **1978–1990s** Most probable structure = Minimal Free-Energy (MFE)
- **1990s–2010s** Functional structure(s) = Boltzmann ensemble (partition function)
- **2010s–????** Embracing the kinetics of RNA folding



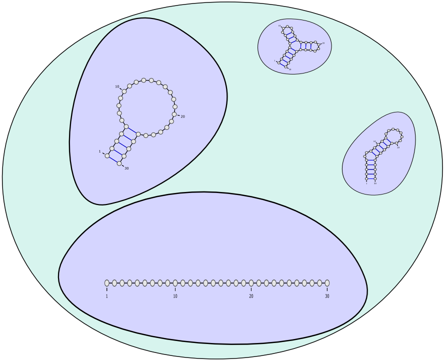
$T \to \infty$

mRNA half-life: $\sim$7h
(Mouse [Sharova2009])

Thermodynamics vs Kinetics

Paradigms for RNA structure prediction

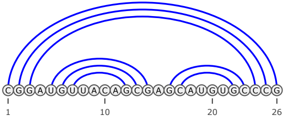
- **1978–1990s** Most probable structure = Minimal Free-Energy (MFE)
- **1990s–2010s** Functional structure(s) = Boltzmann ensemble (partition function)
- **2010s–????** Embracing the kinetics of RNA folding

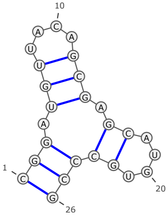


$T = 10\text{h}$

mRNA half-life: $\sim$7h
(Mouse [Sharova2009])

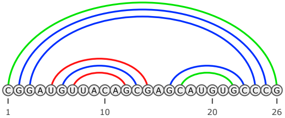
Problem statement

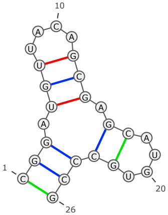




- **RNA structure S:** Non-crossing base-pairs for positions in sequence w
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacks, Loops. . .)
- **Energy model:**
  Motif $\rightarrow$ Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in S

Problem statement



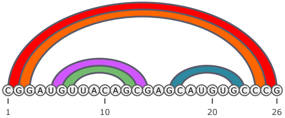


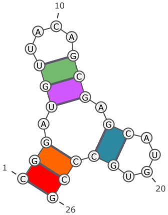
- **RNA structure *S*:** Non-crossing base-pairs for positions in sequence *w*
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacks, Loops. . .)
- **Energy model:**

  Motif $\rightarrow$ Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$

  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in *S*

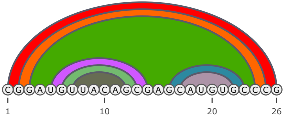
Problem statement

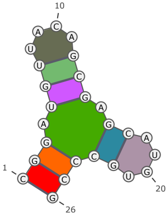




- **RNA structure S:** Non-crossing base-pairs for positions in sequence w
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacks, Loops. . .)
- **Energy model:**
  Motif → Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in S

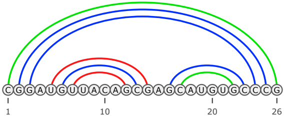
Problem statement

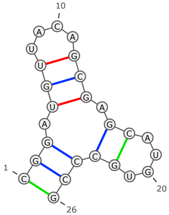




- **RNA structure *S*:** Non-crossing base-pairs for positions in sequence *w*
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacks, Loops. . .)
- **Energy model:**
  Motif $\to$ Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in S

Problem statement

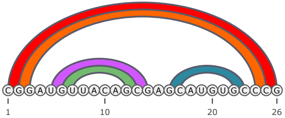


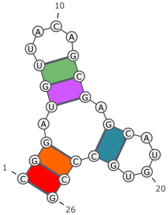


- **RNA structure** ***S*****:** Non-crossing base-pairs for positions in sequence *w*
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacks, Loops. . .)
- **Energy model:**
  Motif → Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^{-} \cup \{+\infty\}$
  Free-Energy $E_w(S)$**:** Sum over (independently contributing) motifs in *S*

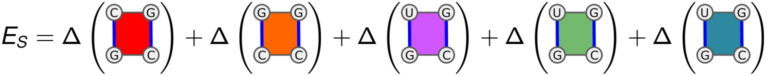
$$E_S = 2 \cdot \Delta\begin{pmatrix} \text{U} \\ | \\ \text{G} \end{pmatrix} + 4 \cdot \Delta\begin{pmatrix} \text{G} \\ | \\ \text{C} \end{pmatrix} + 2 \cdot \Delta\begin{pmatrix} \text{C} \\ | \\ \text{G} \end{pmatrix}$$

Problem statement

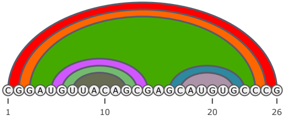


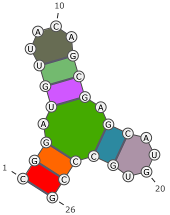


- **RNA structure** ***S*****:** Non-crossing base-pairs for positions in sequence *w*
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacks, Loops. . .)
- **Energy model:**
  Motif → Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  Free-Energy $E_w(S)$**:** Sum over (independently contributing) motifs in *S*

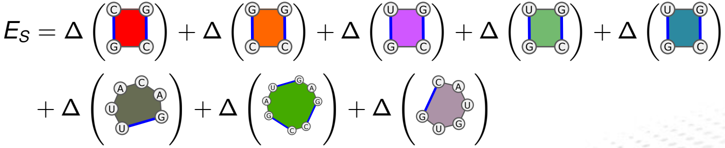


Problem statement

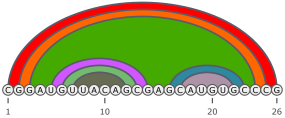


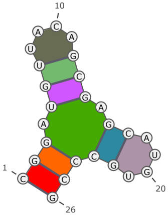


- **RNA structure *S*:** Non-crossing base-pairs for positions in sequence *w*
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacks, Loops...)
- **Energy model:**
  Motif → Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in *S*



Problem statement





- **RNA structure** S**:** Non-crossing base-pairs for positions in sequence w
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacks, Loops. . .)
- **Energy model:**
  Motif → Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^{-} \cup \{+\infty\}$
  Free-Energy $E_w(S)$**:** Sum over (independently contributing) motifs in S

Definition (MFE-PREDICT(E) problem)

Input: RNA sequence $w \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}^*$
Output: Secondary struct. S^* with Minimal Free-Energy (MFE) $E_w(S^*)$

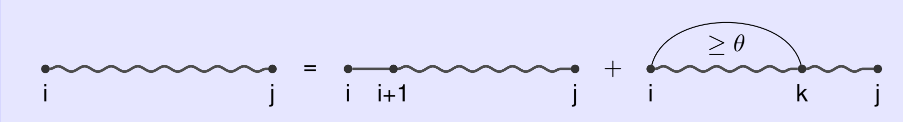
Problem solved **exactly** in $O(n^3)$ time.

[Nussinov Jacobson, PNAS 1980] [Zuker Stiegler, NAR 1981]. . . .

Dynamic programming (DP) for RNA folding

Theorem ([Nussinov and Jacobson(1980)])

Max #base-pairs/min energy structure computed in $\mathcal{O}(n^3)/\mathcal{O}(n^2)$ *time/memory*



$E_{i,k}$: Free-energy contribution of base-pair (i,k). $(-1/+\infty$ or $\Delta G(s_i \overset{?}{\equiv} s_k))$

$\boldsymbol{N}_{i,j}$: Max #base-pairs over interval $[i,j]$

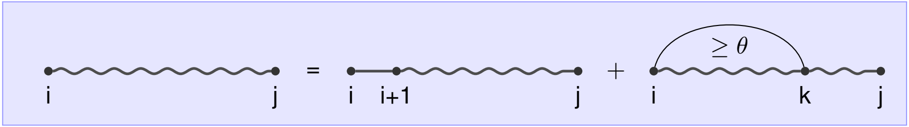
$$\boldsymbol{N}_{i,t} = 0, \quad \forall t \in [i, i+\theta]$$

$$\boldsymbol{N}_{i,j} = \min \begin{cases} \boldsymbol{N}_{i+1,j} & \{i \text{ unpaired}\} \\ \min\limits_{k=i+\theta+1}^{j} E_{i,k} + \boldsymbol{N}_{i+1,k-1} + \boldsymbol{N}_{k+1,j} & \{i \text{ paired to } k\} \end{cases}$$

Dynamic programming (DP) for RNA folding

Theorem ([Nussinov and Jacobson(1980)])

Max #base-pairs/min energy structure computed in $\mathcal{O}(n^3)/\mathcal{O}(n^2)$ time/memory



$E_{i,k}$: Free-energy contribution of base-pair (i,k). $(-1/+\infty$ or $\Delta G(s_i \overset{?}{\equiv} s_k))$

$C_{i,j}$: Number of secondary structures compatible with interval $[i,j]$

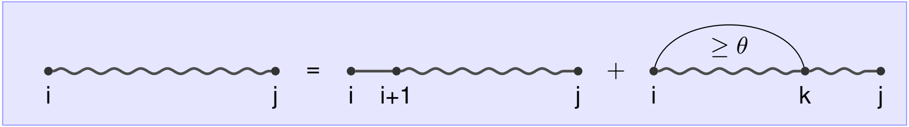
$$C_{i,t} = 1, \quad \forall t \in [i, i+\theta]$$

$$C_{i,j} = \sum \begin{cases} C_{i+1,j} & \{i \text{ unpaired}\} \\ \sum_{k=i+\theta+1}^{j} \mathbb{1}_{\text{comp.}(i,k)} \times C_{i+1,k-1} \times C_{k+1,j} & \{i \text{ paired to } k\} \end{cases}$$

Dynamic programming (DP) for RNA folding

Theorem ([Nussinov and Jacobson(1980)])

Max #base-pairs/min energy structure computed in $\mathcal{O}(n^3)/\mathcal{O}(n^2)$ time/memory



$E_{i,k}$: Free-energy contribution of base-pair (i,k). $(-1/+\infty$ or $\Delta G(s_i \overset{?}{\equiv} s_k))$

$\mathcal{Z}_{i,j} = \sum_{\substack{S \text{ comp.}\\ \text{with } w_{[i,j]}}} e^{\frac{-E_W(S)}{RT}}$ = Partition function for compatible structs within $[i,j]$

$$\begin{aligned}\mathcal{Z}_{i,t} &= \mathbf{1}, \quad \forall t \in [i, i+\theta]\\ \mathcal{Z}_{i,j} &= \sum \begin{cases} \mathcal{Z}_{i+1,j} & \{i \text{ unpaired}\}\\ \sum_{k=i+\theta+1}^{j} e^{\frac{-E_{i,k}}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} & \{i \text{ paired to } k\} \end{cases}\end{aligned}$$

Dynamic programming (DP) for RNA folding

Many extensions:

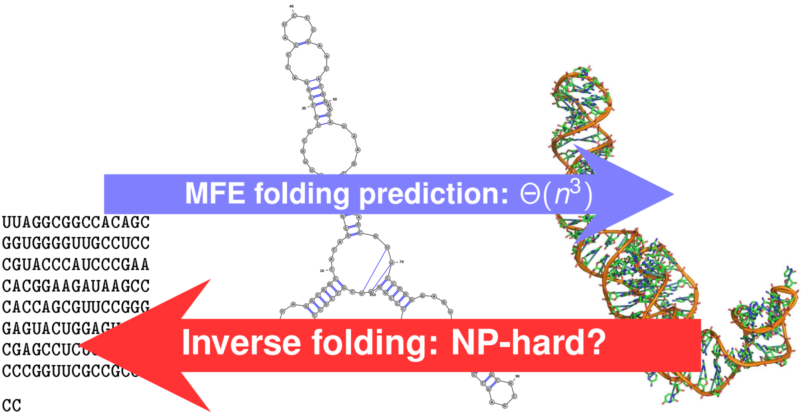
- Nearest-neighbor/Turner energy model [Zuker1981]
- Comparative folding [Sankoff1985]
- Equilibrium base-pairing probabilities [McCaskill1990]
- Moments of additive features [Miklos2005,Ponty2011]
- Δ kcal.mol^{-1} suboptimal structures of MFE [Wuchty1999]
- Basic crossing structures [Rivas1999]. . .
- Exact sampling in Boltzmann distr. [Ding2003,Ponty2008]
- Moments of additive features [Miklos2005,Ponty2011]
- Maximum expected accuracy structure [Do2006]
- Distance-classified partitioning of Boltzmann ens. [E.Freyhult2007a]

Made possible by:

- **Completeness**/**Unambiguity** of decomposition
  $\exists$ energy-preserving bijection between **derivations of DP scheme** and **search space**
- Objective function **additive** with respect to DP scheme

RNA inverse folding

RNA = Linear Polymer = Sequence in $\{A, C, G, U\}^\star$



MFE folding prediction: $\Theta(n^3)$

```
UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAG
CGAGCCUC
CCCGGUUCGCCGC

CC
```

Inverse folding: NP-hard?

Primary Structure Secondary Structure Structure Tertiaire

5s rRNA (PDBID: 1K73:B)

RNA Inverse Folding

Definition (Inverse-Folding(E) problem)

Input: Secondary structure S + Energy distance $\Delta > 0$.
Output: RNA sequence $w \in \Sigma^\star$ such that:

$$\forall S' \in \mathcal{S}|w| \setminus \{S\} :\ E_{w,S'} \geq E w, S + \Delta$$

or $\varnothing$ if no such sequence exists.

Difficult problem: No **obvious** DP decomposition

- Existing algorithms: Heuristics or Exponential-time
- Complexity of problem unknown (despite **[Schnall Levin** *et al*, **ICML'08]**)
  Reason: Non locality, no theoretical frameworks, too many parameters. . .

RNA Inverse Folding

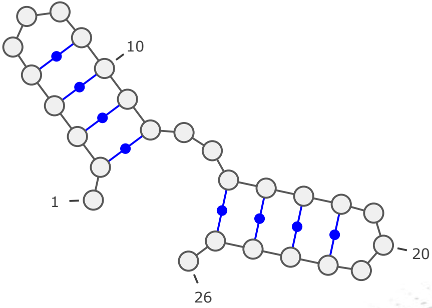
Definition (INVERSE-FOLDING(E) problem)

Input: Secondary structure S + Energy distance $\Delta > 0$.
Output: RNA sequence $w \in \Sigma^\star$ such that:

$$\forall S' \in \mathcal{S}|w| \setminus \{S\} :\; E_{w,S'} \geq Ew, S + \Delta$$

or $\varnothing$ if no such sequence exists.

Example:



RNA Inverse Folding

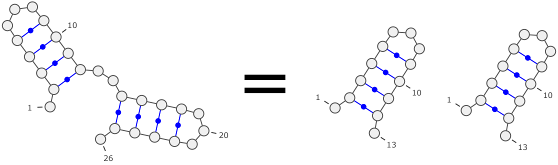
Definition (INVERSE-FOLDING(E) problem)

Input: Secondary structure S + Energy distance $\Delta > 0$.
Output: RNA sequence $w \in \Sigma^\star$ such that:

$$\forall S' \in \mathcal{S}|w| \setminus \{S\} : \; E_{w,S'} \geq Ew, S + \Delta$$

or $\varnothing$ if no such sequence exists.

Example:



RNA Inverse Folding

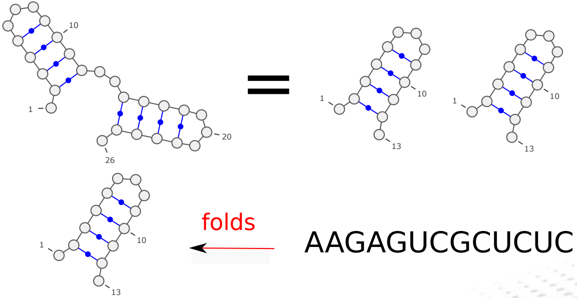
Definition (INVERSE-FOLDING(E) problem)

Input: Secondary structure S + Energy distance $\Delta > 0$.
Output: RNA sequence $w \in \Sigma^\star$ such that:

$$\forall S' \in \mathcal{S}|w| \setminus \{S\} : \; E_{w,S'} \geq Ew, S + \Delta$$

or $\varnothing$ if no such sequence exists.

Example:



folds

AAGAGUCGCUCUC

RNA Inverse Folding

Definition (INVERSE-FOLDING(E) problem)

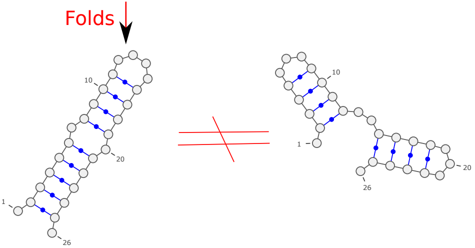
Input: Secondary structure S + Energy distance $\Delta > 0$.
Output: RNA sequence $w \in \Sigma^\star$ such that:

$$\forall S' \in \mathcal{S}|w| \setminus \{S\} : \; E_{w,S'} \geq Ew, S + \Delta$$

or $\varnothing$ if no such sequence exists.

Example:

AAGAGUCGCUCUCAAGAGUCGCUCUC



Existing approaches for negative design

Based on local search. . .

- RNAInverse - TBI Vienna
- Info-RNA - Backofen@Freiburg
- RNA-SSD - Condon@UBC
- NUPack - Pierce@Caltech

. . . bio-inspired algorithms. . .

- RNAFBinv - Barash@Ben Gurion
- FRNAKenstein - Hein@Oxford
- AntaRNA - Backofen@Freiburg
- ERD - Ganjtabesh@Tehran

. . . exact approaches. . .

- RNAIFold - Clote@Boston College
- CO4 - Will@Leipzig

Typical issues:

- Naive initialization strategies
- Poor coverage of sequence space:
  Local search remain *confined* near initial sequence
- GC-rich produced sequences

⇒ **Global sampling** **[Levin *et al*, NAR 12]**

Existing approaches for negative design

Based on local search. . .

- RNAInverse - TBI Vienna
- Info-RNA - Backofen@Freiburg
- RNA-SSD - Condon@UBC
- NUPack - Pierce@Caltech

. . . bio-inspired algorithms. . .

- RNAFBinv - Barash@Ben Gurion
- FRNAKenstein - Hein@Oxford
- AntaRNA - Backofen@Freiburg
- ERD - Ganjtabesh@Tehran

. . . exact approaches. . .

- RNAIFold - Clote@Boston College
- CO4 - Will@Leipzig

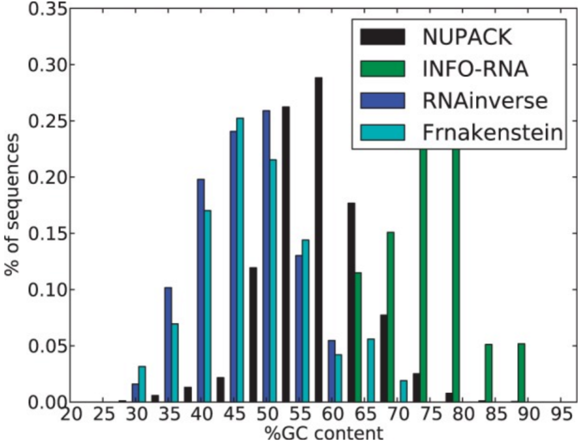
Typical issues:

- Naive initialization strategies
- Poor coverage of sequence space:

  Local search remain *confined* near initial sequence
- GC-rich produced sequences

⇒ **Global sampling [Levin *et al*, NAR 12]**

The case for a control of GC-content



High GC-content suspected to induce **kinetic traps**

Global sampling [Levin *et al*, NAR 12]

Target structure *S*

- **Boltzmann distribution** based on **affinity** towards *S*
- **Random generation** from **Boltzmann Distribution**
- **Fold** sampled sequences and **compare** to target

Boltzmann factor:

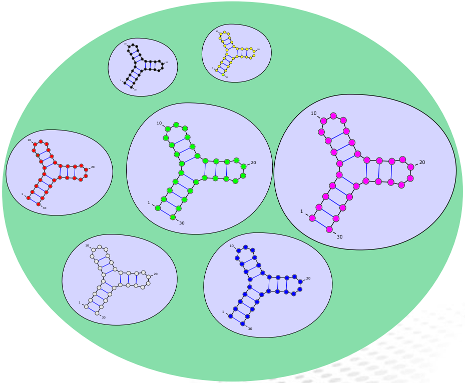
$$\mathcal{B}_w(S) := e^{\frac{-E_w(S)}{RT}}$$

Pseudo-Partition Function:

$$\mathcal{Z}(S) = \sum_{w\in\Sigma^*} \mathcal{B}_w(S)$$

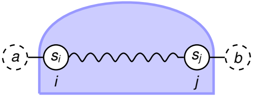
Boltzmann probability:

$$p(s) := \frac{\mathcal{B}_w(S)}{\mathcal{Z}}$$



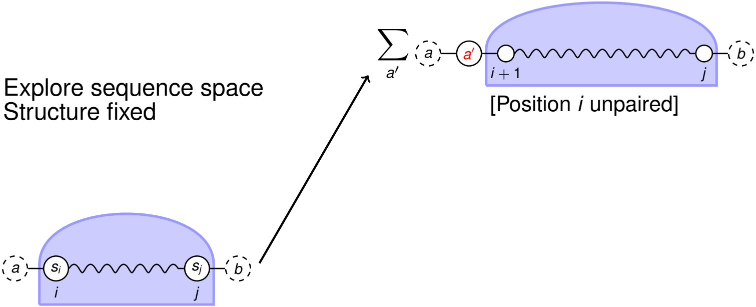
IncaRNAtion [Reinharz *et al*, Bioinformatics 2013]

Explore sequence space
Structure fixed



IncaRNAtion [Reinharz *et al*, Bioinformatics 2013]

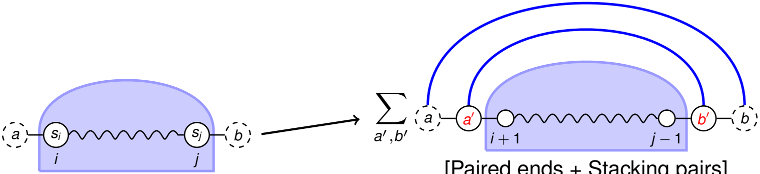
Explore sequence space
Structure fixed



[Position *i* unpaired]

IncaRNAtion [Reinharz *et al*, Bioinformatics 2013]

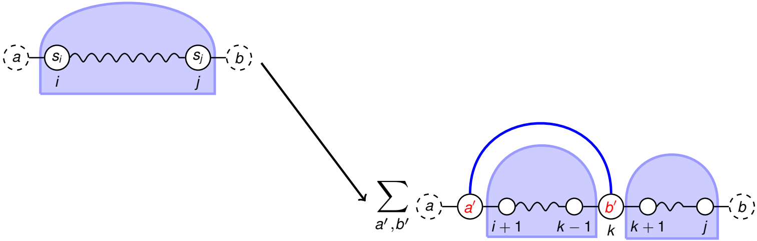
Explore sequence space
Structure fixed



[Paired ends + Stacking pairs]

IncaRNAtion [Reinharz *et al*, Bioinformatics 2013]

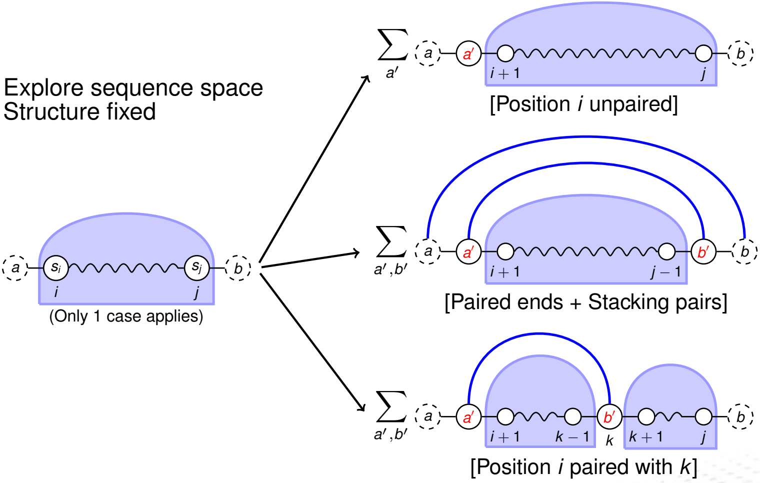
Explore sequence space
Structure fixed



[Position *i* paired with *k*]

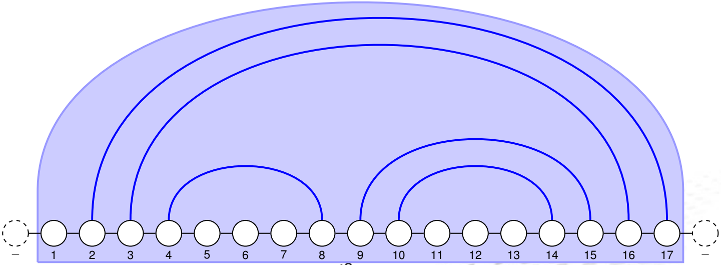
IncaRNAtion [Reinharz *et al*, Bioinformatics 2013]

Explore sequence space
Structure fixed



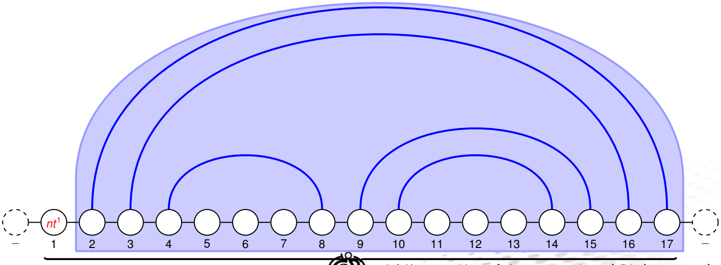
"Global" Stochastic Backtrack

Sequence:



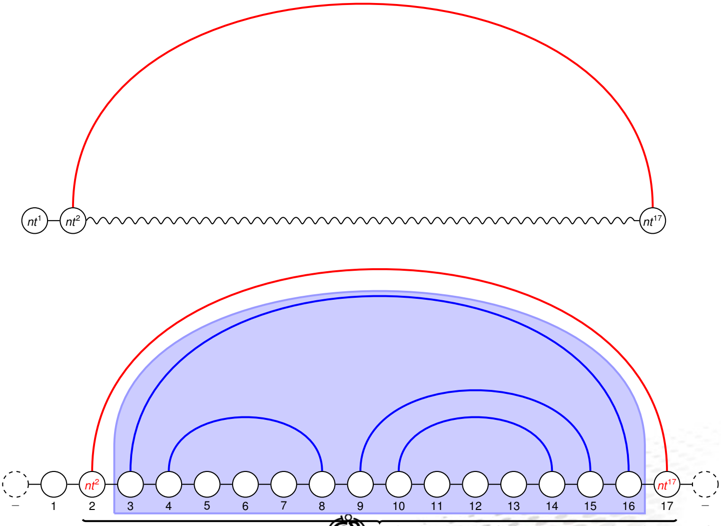
"Global" Stochastic Backtrack

Sequence: nt^1



"Global" Stochastic Backtrack

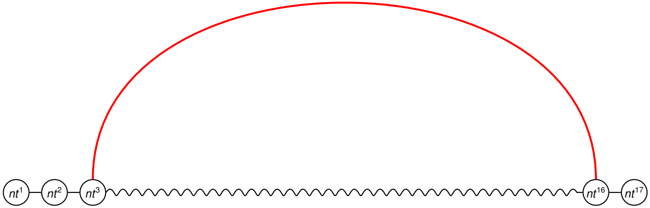
Sequence:

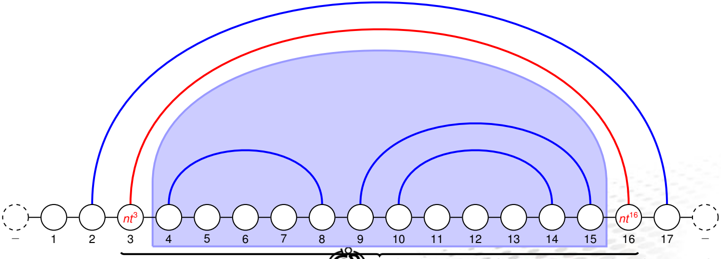


$\mathbb{Z}$ = Weight of $nt^2 \wedge nt^{17}$

"Global" Stochastic Backtrack

Sequence:

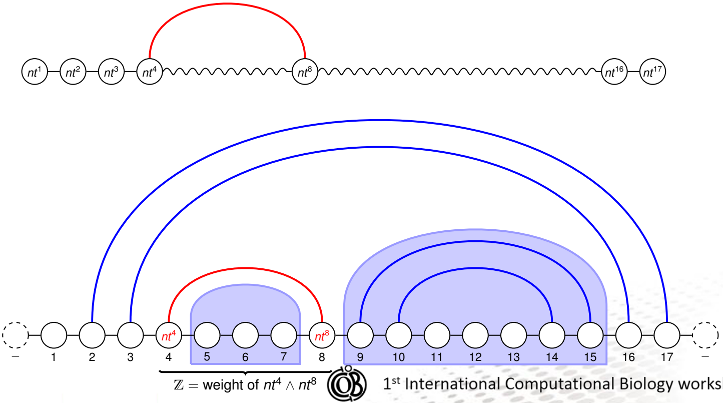




$\mathbb{Z} = \text{weight of } nt^3 \wedge nt^{16}$

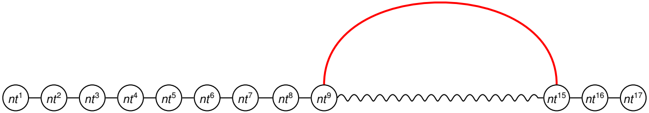
"Global" Stochastic Backtrack

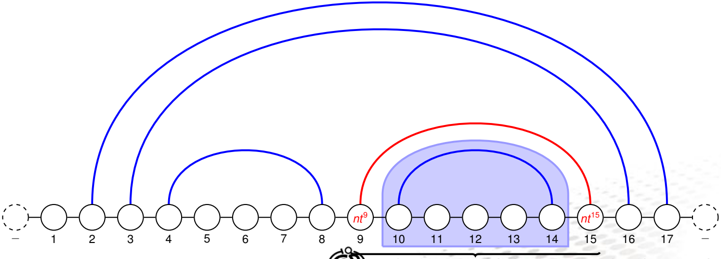
Sequence:



"Global" Stochastic Backtrack

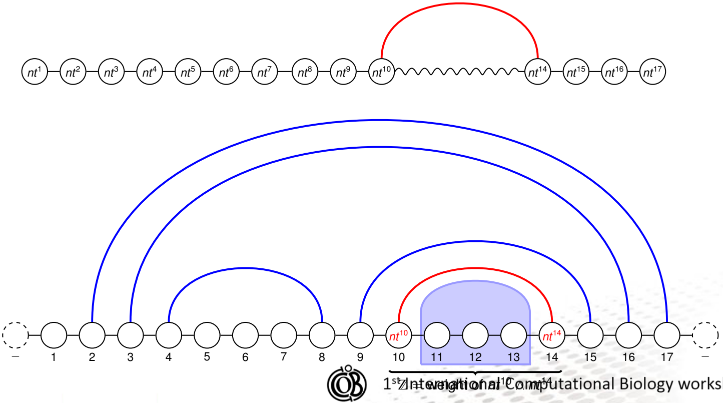
Sequence:





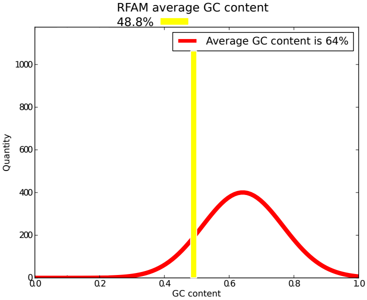
"Global" Stochastic Backtrack

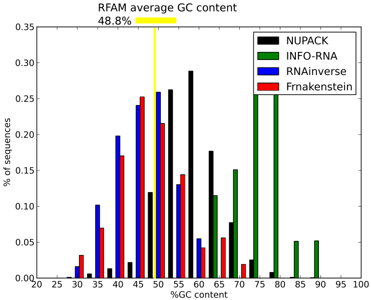
Sequence:



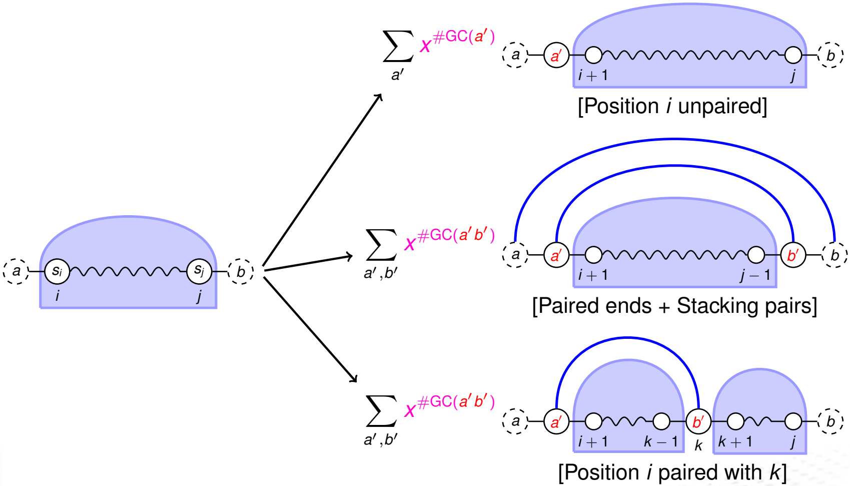
nt1 nt2 nt3 nt4 nt5 nt6 nt7 nt8 nt9 nt10 nt14 nt15 nt16 nt17
–
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
–
nt10
nt14
ℤ = weight nt10 ∧ nt14

GC-content bias





Weighted DP Recursions



$$\sum_{a'} x^{\#\mathrm{GC}(a')}$$

[Position *i* unpaired]

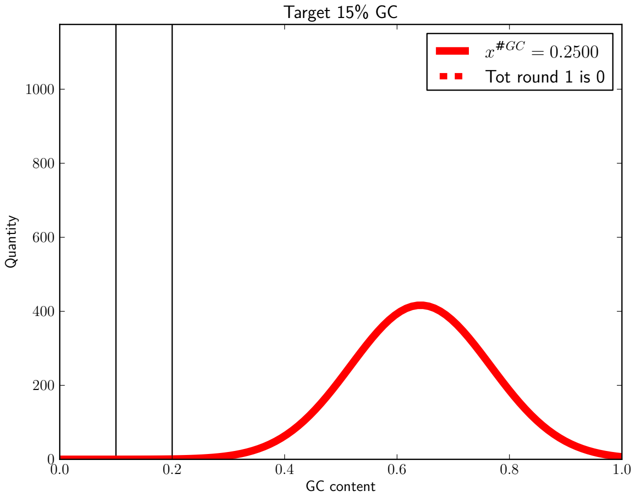
$$\sum_{a',b'} x^{\#\mathrm{GC}(a'b')}$$

[Paired ends + Stacking pairs]

$$\sum_{a',b'} x^{\#\mathrm{GC}(a'b')}$$

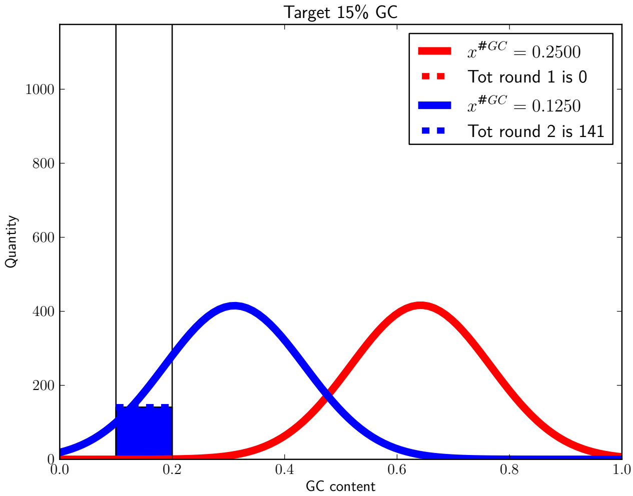
[Position *i* paired with *k*]

Incarnation NT distribution: Bissection scheme



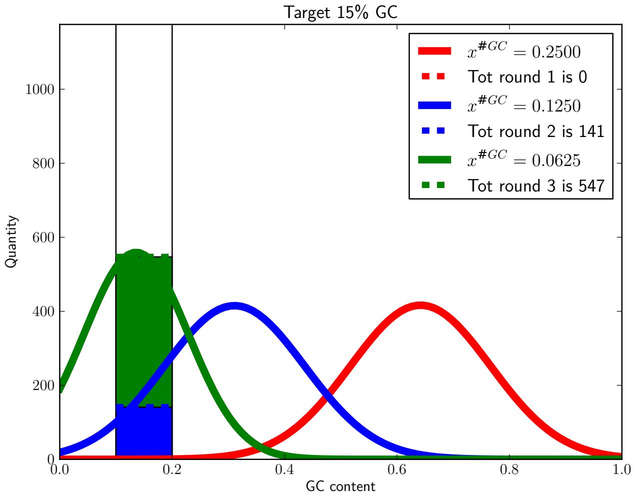
[Waldispühl and Ponty, RECOMB, 2011]

Incarnation NT distribution: Bissection scheme



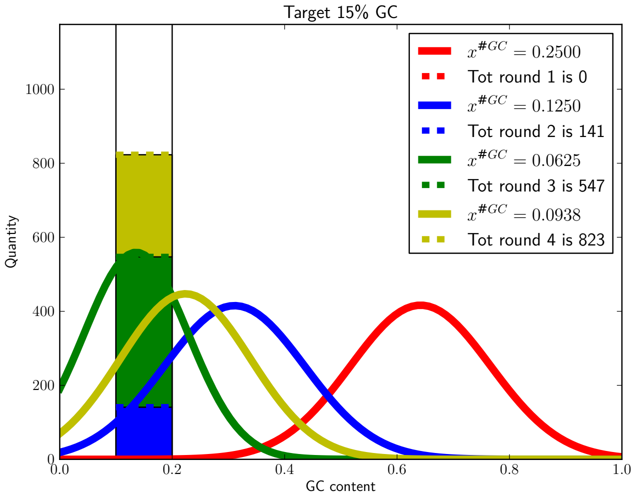
[Waldispühl and Ponty, RECOMB, 2011]

Incarnation NT distribution: Bissection scheme



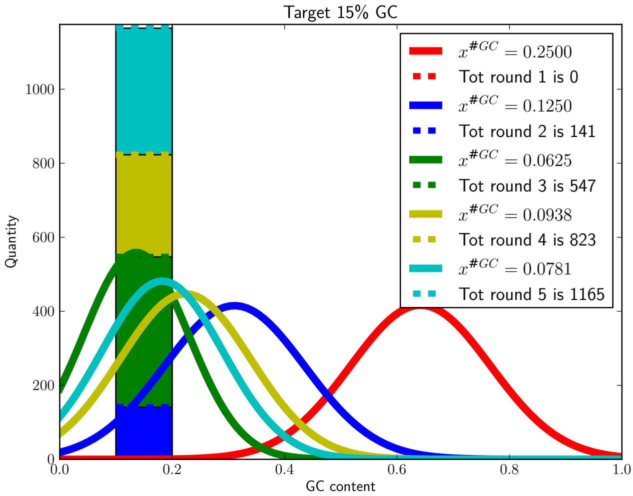
[Waldispühl and Ponty, RECOMB, 2011]

Incarnation NT distribution: Bissection scheme



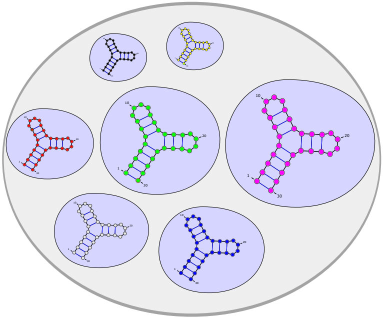
[Waldispühl and Ponty, RECOMB, 2011]

Incarnation NT distribution: Bissection scheme



[Waldispühl and Ponty, RECOMB, 2011]

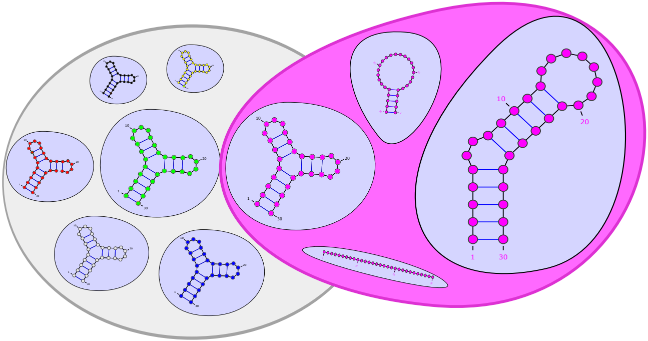
Limits of the approach



Heuristic: Strong affinity is neither sufficient, nor necessary, **but** . . .

- Strong **empirical** correlation affinity/success of design **[Levin et al, NAR 2012]**
- **Linear** time-complexity **[Reinharz Ponty Waldispühl, ISMB/ECCB'13]**
- **Composition** control **[Bodini Ponty, AofA'10] [Reinharz et al, ISMB/ECCB'13]**
- **Complementary** with local search approaches **[Reinharz et al, ISMB/ECCB'13]**

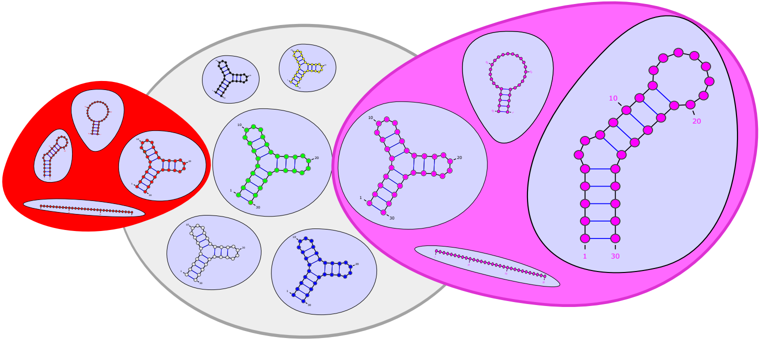
Limits of the approach



Heuristic: Strong affinity is **neither sufficient,** nor necessary, **but** . . .

- Strong **empirical** correlation affinity/success of design **[Levin et al, NAR 2012]**
- **Linear** time-complexity **[Reinharz Ponty Waldispühl, ISMB/ECCB'13]**
- **Composition** control **[Bodini Ponty, AofA'10] [Reinharz et al, ISMB/ECCB'13]**
- **Complementary** with local search approaches **[Reinharz et al, ISMB/ECCB'13]**

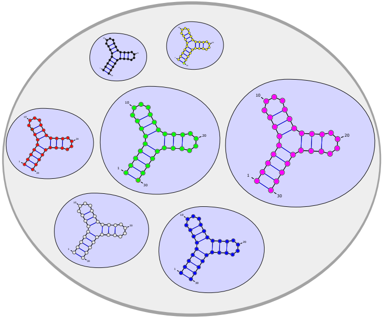
Limits of the approach



Heuristic: Strong affinity is **neither sufficient, nor necessary**, **but** . . .

- Strong **empirical** correlation affinity/success of design **[Levin et al, NAR 2012]**
- **Linear** time-complexity **[Reinharz Ponty Waldispühl, ISMB/ECCB'13]**
- **Composition** control **[Bodini Ponty, AofA'10] [Reinharz et al, ISMB/ECCB'13]**
- **Complementary** with local search approaches **[Reinharz et al, ISMB/ECCB'13]**

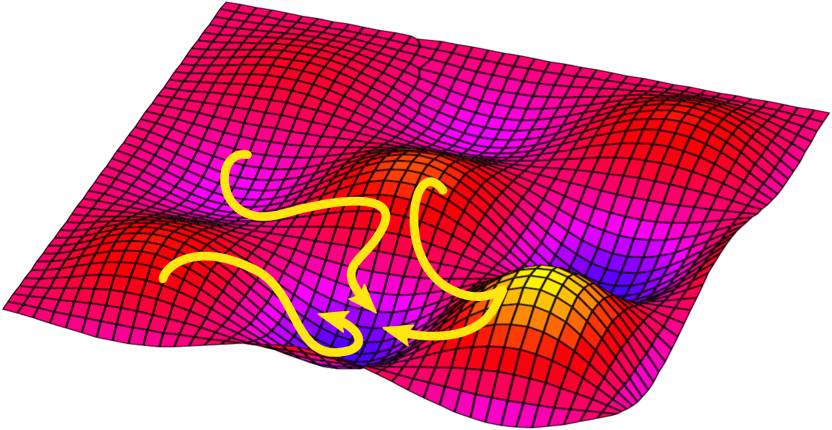
Limits of the approach



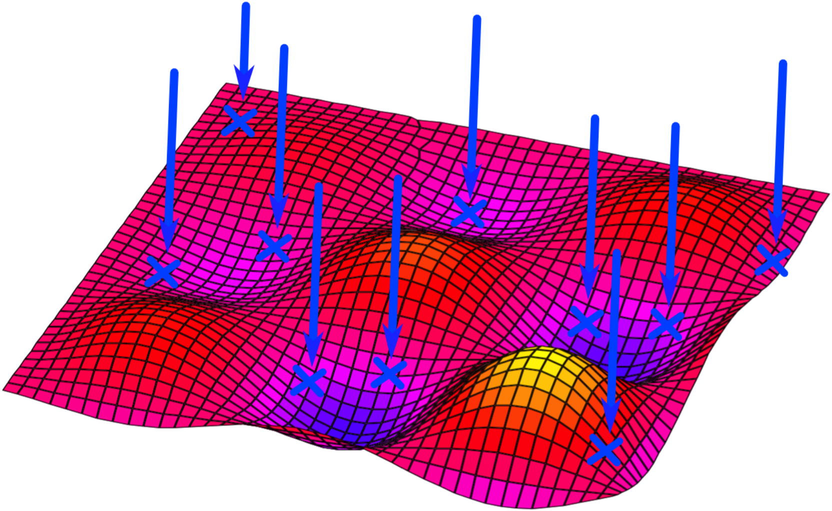
Heriustic: Strong affinity is **neither sufficient, nor necessary**, **but** . . .

- Strong **empirical** correlation affinity/success of design **[Levin et al, NAR 2012]**
- **Linear** time-complexity **[Reinharz Ponty Waldispühl, ISMB/ECCB'13]**
- **Composition** control **[Bodini Ponty, AofA'10] [Reinharz et al, ISMB/ECCB'13]**
- **Complementary** with local search approaches **[Reinharz et al, ISMB/ECCB'13]**

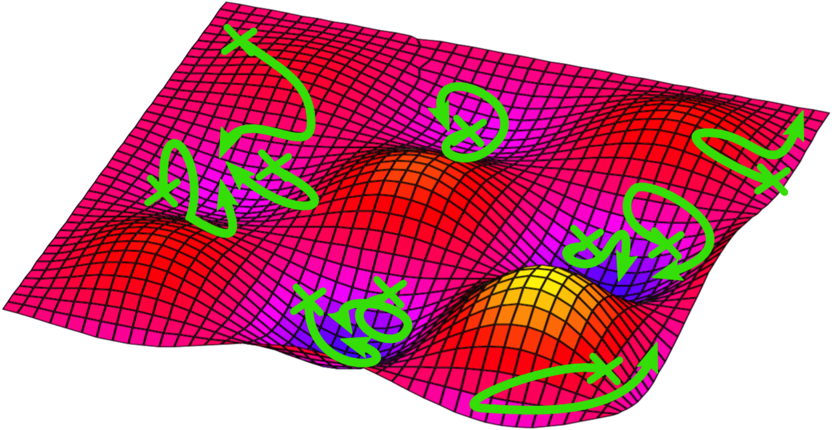
Local vs Global vs "Glocal"



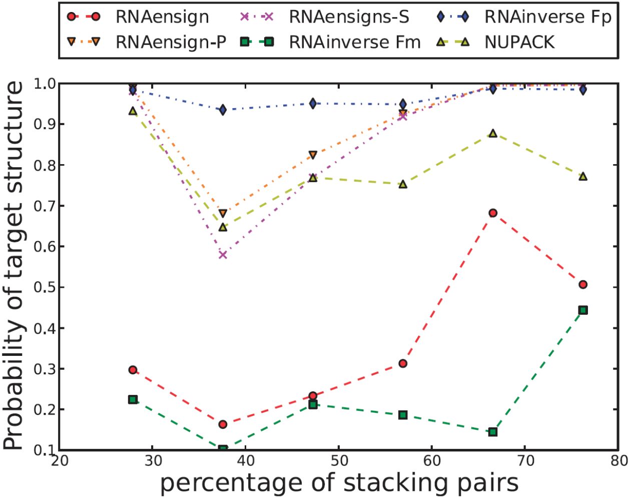
Local vs Global vs "Glocal"



Local vs Global vs "Glocal"



The success of glocal strategies



Sampling + Optimize creates highly probable design sequences

II. Constrained design

Avoiding/forcing motifs

Existing approaches for negative design

Based on local search. . .

- RNAInverse - TBI Vienna
- Info-RNA - Backofen@Freiburg
- RNA-SSD - Condon@UBC
- NUPack - Pierce@Caltech

. . . bio-inspired algorithms. . .

- RNAFBinv - Barash@Ben Gurion
- FRNAKenstein - Hein@Oxford
- AntaRNA - Backofen@Freiburg

. . . exact approaches. . .

- RNAIFold - Clote@Boston College
- CO4 - Will@Leipzig

Few algorithms support avoided/mandatory motifs. . .

. . . none guarantees *reasonable* runtime.

Typical reasons:

- Deep local minima (Rugged landscape)
- Mandatory motifs ⇒ Late deadends (Branch and Bound)
- Forbidden motifs ⇒ Search space disconnection (Local Search)

Existing approaches for negative design

Based on local search. . .

- RNAInverse - TBI Vienna
- Info-RNA - Backofen@Freiburg
- RNA-SSD - Condon@UBC
- NUPack - Pierce@Caltech

. . . bio-inspired algorithms. . .

- RNAFBinv - Barash@Ben Gurion
- FRNAKenstein - Hein@Oxford
- AntaRNA - Backofen@Freiburg

. . . exact approaches. . .

- RNAIFold - Clote@Boston College
- CO4 - Will@Leipzig

Few algorithms support avoided/mandatory motifs. . .

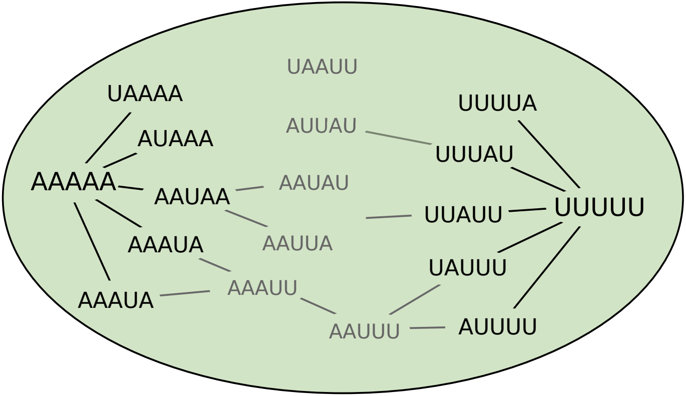
. . . none guarantees *reasonable* runtime.

Typical reasons:

- Deep local minima (Rugged landscape)
- Mandatory motifs ⇒ Late deadends (Branch and Bound)
- Forbidden motifs ⇒ Search space disconnection (Local Search)

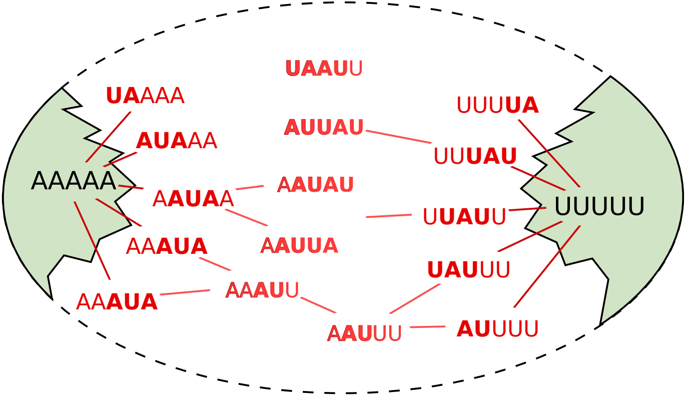
Problem with local approaches: An example

Simplified vocabulary $\{\mathsf{A}, \mathsf{U}\}$



Problem with local approaches: An example

Simplified vocabulary $\{A, U\}$ + **Forbidden motifs** $\mathcal{F} = \{AU,UA\}$



⇒ $\mathcal{F}$ may **disconnect** search space (holds for **any** move set!)

Idea

Use formal language constructs to constrain global sampling

Forced motifs
Avoided motifs → Regular language $\mathcal{L}_C \in$ Reg

Structure compatibility
+ Positional constraints → **Weighted** Context-Free Lang $\mathcal{L}_S \in$ CFL
+ Energy Model

Folklore theorem (constructive): Reg $\cap$ (**W**)CFL $\subseteq$ (**W**)CFL

Build weighted context-free grammar $\mathcal{G}$ for $\mathcal{L}_C \cap \mathcal{L}_S$
+ Random generation

$\Rightarrow$ **Global sampling under constraints**

Idea

Use formal language constructs to constrain global sampling

Forced motifs
Avoided motifs → Regular language $\mathcal{L}_C \in$ Reg

Structure compatibility
+ Positional constraints → **Weighted** Context-Free Lang $\mathcal{L}_S \in$ CFL
+ Energy Model

Folklore theorem (constructive): Reg $\cap$ (W)CFL $\subseteq$ (W)CFL

Build weighted context-free grammar $\mathcal{G}$ for $\mathcal{L}_C \cap \mathcal{L}_S$
+ Random generation

$\Rightarrow$ **Global sampling under constraints**

Idea

Use formal language constructs to constrain global sampling

Forced motifs
Avoided motifs
→ Regular language $\mathcal{L}_C \in$ Reg

Structure compatibility
+ Positional constraints
+ Energy Model
→ **Weighted** Context-Free Lang $\mathcal{L}_S \in$ CFL

Folklore theorem (constructive): Reg $\cap$ (**W**)CFL $\subseteq$ (**W**)CFL

Build weighted context-free grammar $\mathcal{G}$ for $\mathcal{L}_C \cap \mathcal{L}_S$
+ Random generation

$\Rightarrow$ **Global sampling under constraints**

Idea

Use formal language constructs to constrain global sampling

Forced motifs
Avoided motifs $\rightarrow$ Regular language $\mathcal{L}_C \in$ Reg

Structure compatibility
+ Positional constraints $\rightarrow$ **Weighted** Context-Free Lang $\mathcal{L}_S \in$ CFL
+ Energy Model

Folklore theorem (constructive): Reg $\cap$ (**W**)CFL $\subseteq$ (**W**)CFL

Build weighted context-free grammar $\mathcal{G}$ for $\mathcal{L}_C \cap \mathcal{L}_S$
+ Random generation

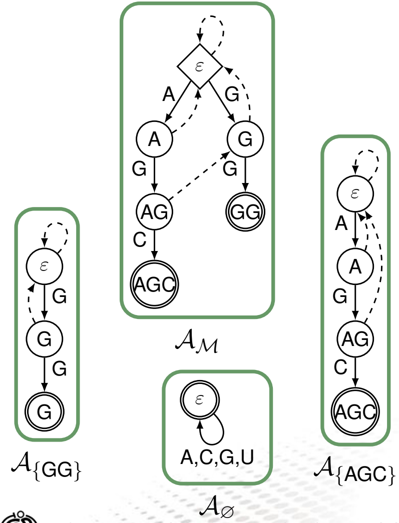
$\Rightarrow$ **Global sampling under constraints**

Building the Finite State Automaton

To force multiple words, **keep track** of generated words:

- Create disjunctive automata for each $\mathcal{M}' \subseteq \mathcal{M}$
- **Reroute** accepting states
- Accepting state = no **forced word** remaining (ε in $\mathcal{A}_{\varnothing}$)
- Forbidden words can be added to sub-automata

Example: $\mathcal{M} = \{\mathrm{AGC}, \mathrm{GG}\}$



ε
A
G
A
G
G
G
AG
GG
C
AGC
$\mathcal{A}_{\mathcal{M}}$
ε
G
G
G
G
$\mathcal{A}_{\{\mathrm{GG}\}}$
ε
A,C,G,U
$\mathcal{A}_{\varnothing}$
ε
A
A
G
AG
C
AGC
$\mathcal{A}_{\{\mathrm{AGC}\}}$

#States:

$$O\left(2^{|\mathcal{M}|} \cdot \left(\sum_i |f_i| + \sum_j |m_j|\right)\right)$$

Building the Finite State Automaton

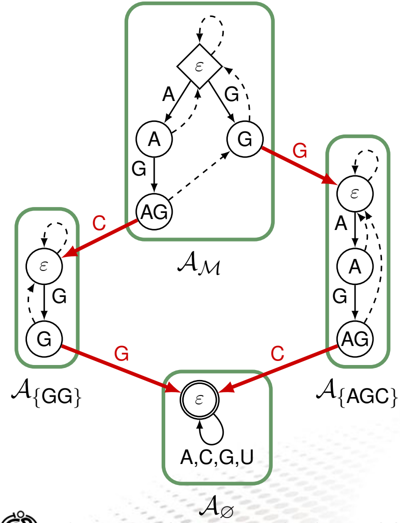
To force multiple words, **keep track** of generated words:

- Create disjunctive automata for each $\mathcal{M}' \subseteq \mathcal{M}$
- **Reroute** accepting states
- Accepting state = no **forced word** remaining (ε in $\mathcal{A}_\varnothing$)
- Forbidden words can be added to sub-automata

#States:

$$O\left(2^{|\mathcal{M}|} \cdot \left(\sum_i |f_i| + \sum_j |m_j|\right)\right)$$

Example: $\mathcal{M} = \{\text{AGC}, \text{GG}\}$



ε
A
G
A
G
G
AG
$\mathcal{A}_{\mathcal{M}}$
G
ε
A
A
G
AG
$\mathcal{A}_{\{\text{AGC}\}}$
C
ε
G
G
$\mathcal{A}_{\{\text{GG}\}}$
G
C
ε
A,C,G,U
$\mathcal{A}_\varnothing$

Building the Finite State Automaton

To force multiple words, **keep track** of generated words:

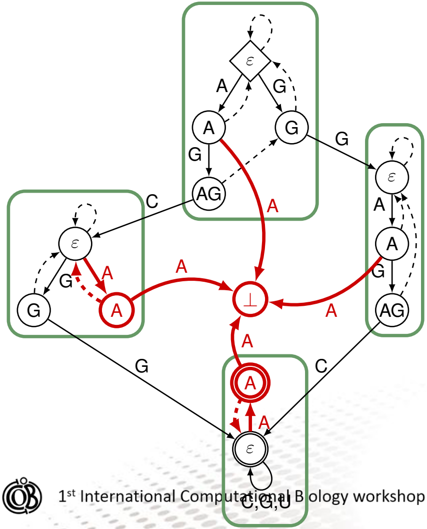
- Create disjunctive automata for each $\mathcal{M}' \subseteq \mathcal{M}$
- **Reroute** accepting states
- Accepting state = no **forced word** remaining (ε in $\mathcal{A}_\varnothing$)
- Forbidden words can be added to sub-automata

#States:

$$O\left(2^{|\mathcal{M}|} \cdot \left(\sum_i |f_i| + \sum_j |m_j|\right)\right)$$

Example:

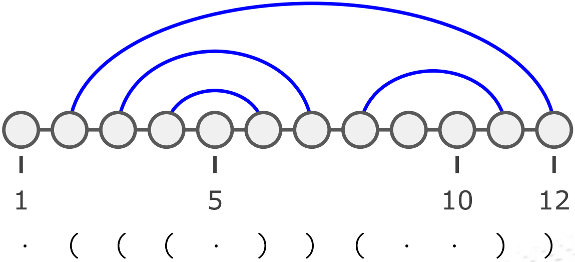
$\mathcal{M} = \{\text{AGC}, \text{GG}\}; \mathcal{F} = \{\text{AA}\}$



Building the grammar

Input: Secondary Structure *S* + Positional constraints

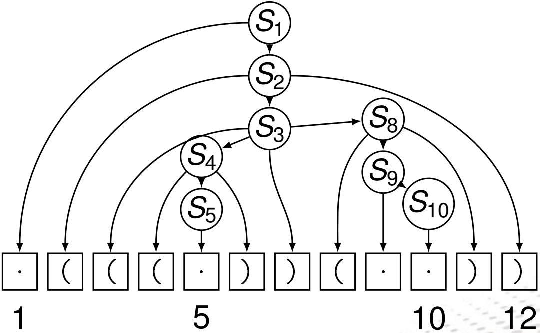
- A **Create Parse Tree** for secondary structure
- B **Translate Parse Tree** into single-word grammar
- C **Expand** grammar to instantiate compatible base/base-pairs
- D **Restrict** to bases/base-pairs allowed at each position



Building the grammar

Input: Secondary Structure S + Positional constraints

- **A** **Create Parse Tree** for secondary structure
- **B** **Translate Parse Tree** into single-word grammar
- **C** **Expand** grammar to instantiate compatible base/base-pairs
- **D** **Restrict** to bases/base-pairs allowed at each position



Building the grammar

Input: Secondary Structure S + Positional constraints

- **A** **Create Parse Tree** for secondary structure
- **B** **Translate Parse Tree** into single-word grammar
- **C** **Expand** grammar to instantiate compatible base/base-pairs
- **D** **Restrict** to bases/base-pairs allowed at each position

$$S_1 \rightarrow .\,S_2 \qquad S_2 \rightarrow (\,S_3\,) \qquad S_3 \rightarrow (\,S_4\,)\,S_8 \qquad S_4 \rightarrow (\,S_5\,)$$

$$S_5 \rightarrow . \qquad S_8 \rightarrow (\,S_9\,) \qquad S_9 \rightarrow .\,S_{10} \qquad S_{10} \rightarrow .$$

Building the grammar

Input: Secondary Structure *S* + Positional constraints

A **Create Parse Tree** for secondary structure

B **Translate Parse Tree** into single-word grammar

C **Expand** grammar to instantiate compatible base/base-pairs

D **Restrict** to bases/base-pairs allowed at each position

$$V_1 \to \mathrm{A}\,V_2 \mid \mathrm{C}\,V_2 \mid \mathrm{G}\,V_2 \mid \mathrm{U}\,V_2$$

$$V_2 \to \mathrm{A}\,V_3\,\mathrm{U} \mid \mathrm{C}\,V_3\,\mathrm{G} \mid \mathrm{G}\,V_3\,\mathrm{C} \mid \mathrm{G}\,V_3\,\mathrm{U} \mid \mathrm{U}\,V_3\,\mathrm{A} \mid \mathrm{U}\,V_3\,\mathrm{G}$$

$$V_3 \to \mathrm{A}\,V_4\,\mathrm{U}\,V_8 \mid \mathrm{C}\,V_4\,\mathrm{G}\,V_8 \mid \mathrm{G}\,V_4\,\mathrm{C}\,V_8 \mid \mathrm{G}\,V_4\,\mathrm{U}\,V_8 \mid \mathrm{U}\,V_4\,\mathrm{A}\,V_8 \mid \mathrm{U}\,V_4\,\mathrm{G}\,V_8$$

$$V_4 \to \mathrm{A}\,V_5\,\mathrm{U} \mid \mathrm{C}\,V_5\,\mathrm{G} \mid \mathrm{G}\,V_5\,\mathrm{C} \mid \mathrm{G}\,V_5\,\mathrm{U} \mid \mathrm{U}\,V_5\,\mathrm{A} \mid \mathrm{U}\,V_5\,\mathrm{G}$$

$$V_5 \to \mathrm{A} \mid \mathrm{C} \mid \mathrm{G} \mid \mathrm{U}$$

$$V_8 \to \mathrm{A}\,V_9\,\mathrm{U} \mid \mathrm{C}\,V_9\,\mathrm{G} \mid \mathrm{G}\,V_9\,\mathrm{C} \mid \mathrm{G}\,V_9\,\mathrm{U} \mid \mathrm{U}\,V_9\,\mathrm{A} \mid \mathrm{U}\,V_9\,\mathrm{G}$$

$$V_9 \to \mathrm{A}\,V_{10} \mid \mathrm{C}\,V_{10} \mid \mathrm{G}\,V_{10} \mid \mathrm{U}\,V_{10}$$

$$V_{10} \to \mathrm{A} \mid \mathrm{C} \mid \mathrm{G} \mid \mathrm{U}$$

Building the grammar

Input: Secondary Structure *S* **+ Positional constraints**

- **A Create Parse Tree** for secondary structure
- **B Translate Parse Tree** into single-word grammar
- **C Expand** grammar to instantiate compatible base/base-pairs
- **D Restrict** to bases/base-pairs allowed at each position

$$
\begin{aligned}
V_1 &\rightarrow \mathrm{A}\,V_2 \mid \mathrm{C}\,V_2 \mid \mathrm{G}\,V_2 \mid \mathrm{U}\,V_2 \\
V_2 &\rightarrow \mathrm{A}\,V_3\,\mathrm{U} \mid \cancel{\mathrm{C}\,V_3\,\mathrm{G}} \mid \cancel{\mathrm{G}\,V_3\,\mathrm{C}} \mid \cancel{\mathrm{G}\,V_3\,\mathrm{U}} \mid \mathrm{U}\,V_3\,\mathrm{A} \mid \cancel{\mathrm{U}\,V_3\,\mathrm{G}} \\
V_3 &\rightarrow \mathrm{A}\,V_4\,\mathrm{U}\,V_8 \mid \cancel{\mathrm{C}\,V_4\,\mathrm{G}\,V_8} \mid \cancel{\mathrm{G}\,V_4\,\mathrm{C}\,V_8} \mid \cancel{\mathrm{G}\,V_4\,\mathrm{U}\,V_8} \mid \mathrm{U}\,V_4\,\mathrm{A}\,V_8 \mid \cancel{\mathrm{U}\,V_4\,\mathrm{G}\,V_8} \\
V_4 &\rightarrow \mathrm{A}\,V_5\,\mathrm{U} \mid \cancel{\mathrm{C}\,V_5\,\mathrm{G}} \mid \cancel{\mathrm{G}\,V_5\,\mathrm{C}} \mid \cancel{\mathrm{G}\,V_5\,\mathrm{U}} \mid \mathrm{U}\,V_5\,\mathrm{A} \mid \cancel{\mathrm{U}\,V_5\,\mathrm{G}} \\
V_5 &\rightarrow \mathrm{A} \mid \mathrm{C} \mid \mathrm{G} \mid \mathrm{U} \\
V_8 &\rightarrow \mathrm{A}\,V_9\,\mathrm{U} \mid \cancel{\mathrm{C}\,V_9\,\mathrm{G}} \mid \cancel{\mathrm{G}\,V_9\,\mathrm{C}} \mid \cancel{\mathrm{G}\,V_9\,\mathrm{U}} \mid \mathrm{U}\,V_9\,\mathrm{A} \mid \cancel{\mathrm{U}\,V_9\,\mathrm{G}} \\
V_9 &\rightarrow \mathrm{A}\,V_{10} \mid \mathrm{C}\,V_{10} \mid \mathrm{G}\,V_{10} \mid \mathrm{U}\,V_{10} \\
V_{10} &\rightarrow \mathrm{A} \mid \mathrm{C} \mid \mathrm{G} \mid \mathrm{U}
\end{aligned}
$$

Random generation

Combine CFG and aut. → CFG (Multiplying $\#$Rules by $|Q|^3$)

GenRGenS [Ponty Termier Denise, Bioinformatics 2006]:

- Precomputes #words for each non-terminal
- Random Generation w.r.t. **weighted distribution**

Energy models:

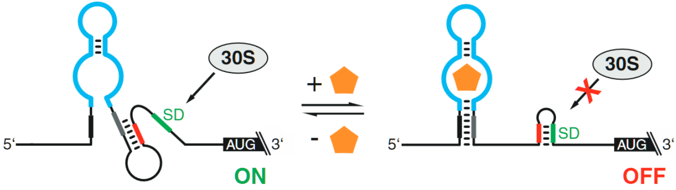
- **Uniform distribution**
- **Nussinov energy model**
- **Stacking-pairs model (Turner 2004)**
  Based on refined, yet similar, grammar

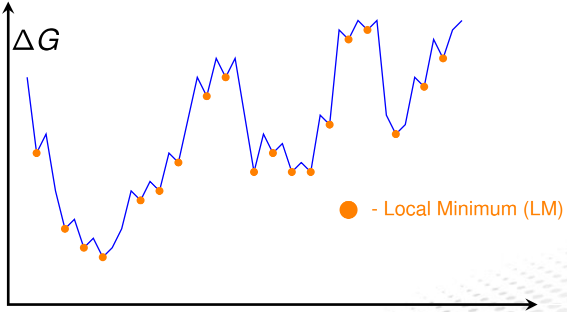
Overall complexity: $|S| \cdot 2^{3|\mathcal{M}|} \cdot \left(\sum_i |f_i| + \sum_j |m_j|\right)^3$

- **Linear** on $|S|$
- **Exponential** on $|\mathcal{M}|$, but **NP-Hard** problem

III. Positive design for multiple structures

Motivation: Kinetics and riboswitches





Design objectives

Positive structural design

Optimize affinity of designed sequences towards target structure
Or simply ensure their compatibility with one or **several structures**

Examples: Most stable sequence for given fold. . .

Negative structural design

Limit affinity of designed sequences towards **alternative structures**

Examples: Lowest free-energy, High Boltzmann probability/Low entropy. . .

Additional constraints:

- **Forbid** motif list to appear **anywhere** in design
- **Force** motif list to appear **each at least once**
- **Limit** available alternatives at certain positions
- **Control** overall composition (GC-content)

Design objectives

Positive structural design

Optimize affinity of designed sequences towards target structure

Or simply ensure their compatibility with one or **several structures**

Examples: Most stable sequence for given fold. . .

Negative structural design

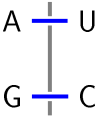
Limit affinity of designed sequences towards **alternative structures**

Examples: Lowest free-energy, High Boltzmann probability/Low entropy. . .

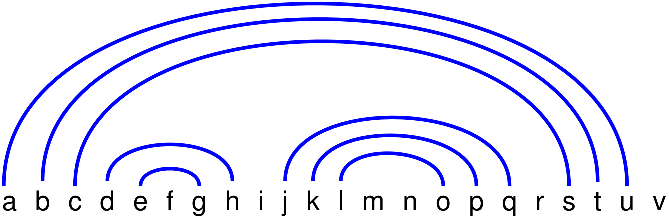
Additional constraints:

- **Forbid** motif list to appear **anywhere** in design
- **Force** motif list to appear **each at least once**
- **Limit** available alternatives at certain positions
- **Control** overall composition (GC-content)

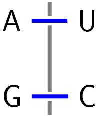
Counting compatible RNAs: Watson-Crick + Single structure



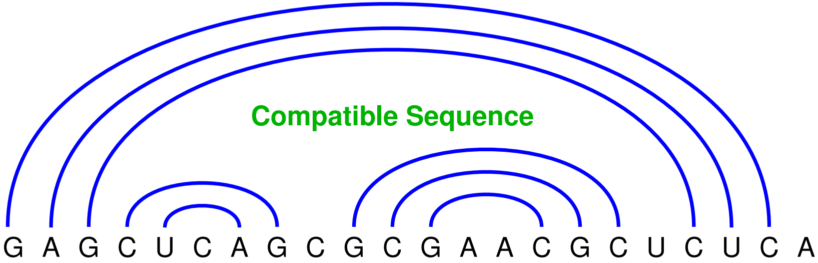
Compatible Base Pairs = Only **Watson-Crick** base pairs



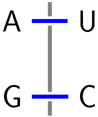
Counting compatible RNAs: Watson-Crick + Single structure



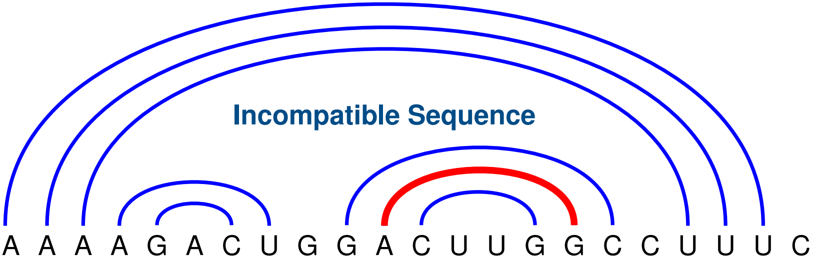
Compatible Base Pairs = Only **Watson-Crick** base pairs



Counting compatible RNAs: Watson-Crick + Single structure

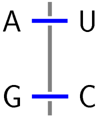


Compatible Base Pairs = Only **Watson-Crick** base pairs

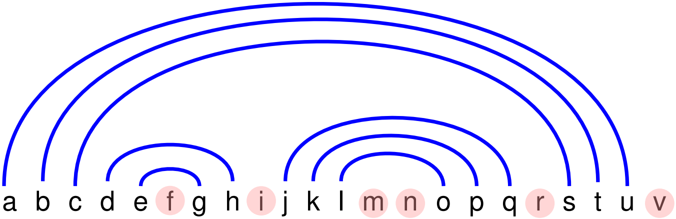


Question: How many **Compatible** sequences?

Counting compatible RNAs: Watson-Crick + Single structure

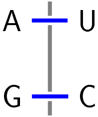


Compatible Base Pairs = Only **Watson-Crick** base pairs

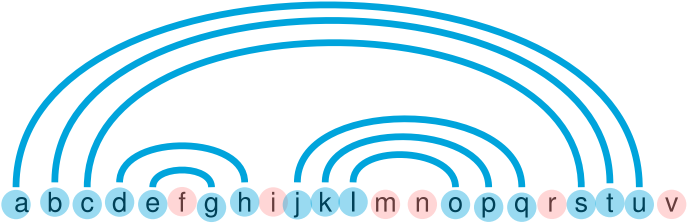


Question: How many **Compatible** sequences?

Counting compatible RNAs: Watson-Crick + Single structure

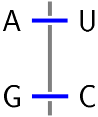


Compatible Base Pairs = Only **Watson-Crick** base pairs

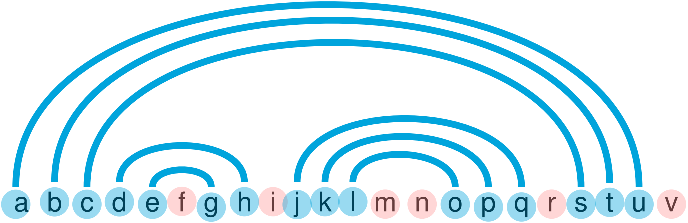


Question: How many **Compatible** sequences?

Counting compatible RNAs: Watson-Crick + Single structure



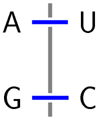
Compatible Base Pairs = Only **Watson-Crick** base pairs



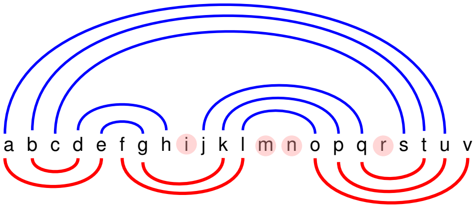
Question: How many **Compatible** sequences?

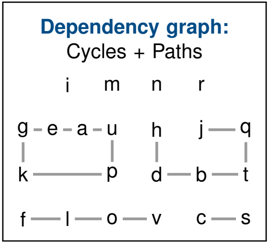
Answer: $4^{\#\text{BPs}} \times 4^{\#\text{Unpaired}} \rightarrow 268\,435\,456$

Counting compatible RNAs: Watson-Crick + Two structures



Compatible Base Pairs = Only **Watson-Crick** base pairs





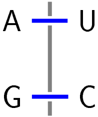
Dependency graph:

Cycles + Paths

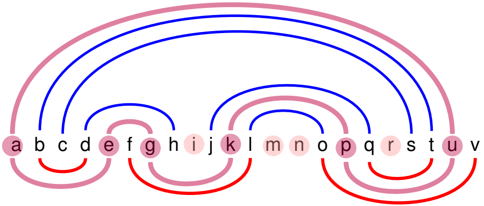
Question: How many **Compatible** sequences?

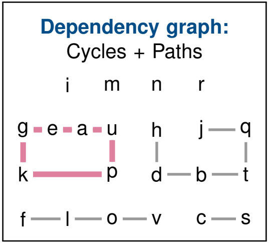
Answer: $\neq \varnothing$! (both base-pairs and dependency graphs **bipartite**)

Counting compatible RNAs: Watson-Crick + Two structures



Compatible Base Pairs = Only **Watson-Crick** base pairs

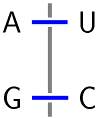



Dependency graph:
Cycles + Paths

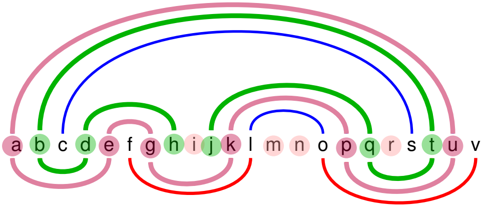
Question: How many **Compatible** sequences?

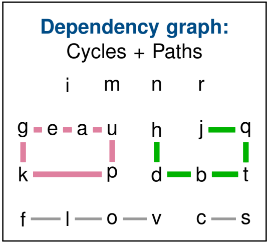
Answer: $\neq \varnothing$! (both base-pairs and dependency graphs **bipartite**)

Counting compatible RNAs: Watson-Crick + Two structures



Compatible Base Pairs = Only **Watson-Crick** base pairs

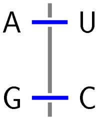



Dependency graph:
Cycles + Paths

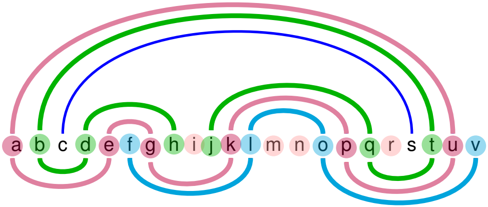
Question: How many **Compatible** sequences?

Answer: $\neq \varnothing$! (both base-pairs and dependency graphs **bipartite**)

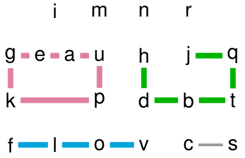
Counting compatible RNAs: Watson-Crick + Two structures



Compatible Base Pairs = Only **Watson-Crick** base pairs



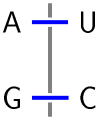
Dependency graph:
Cycles + Paths



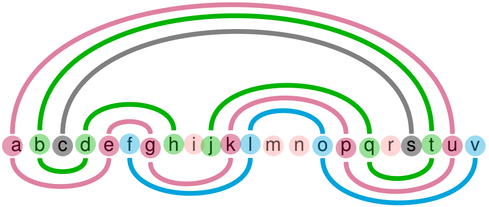
Question: How many **Compatible** sequences?

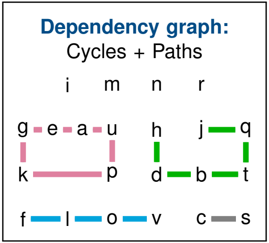
Answer: $\neq \varnothing$! (both base-pairs and dependency graphs **bipartite**)

Counting compatible RNAs: Watson-Crick + Two structures



Compatible Base Pairs = Only **Watson-Crick** base pairs



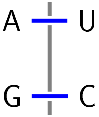

Dependency graph:
Cycles + Paths

Question: How many **Compatible** sequences?

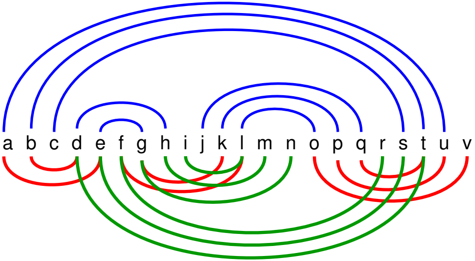
Answer: $\neq \varnothing$! (both base-pairs and dependency graphs **bipartite**)

$4^{\#\text{CCs}} \rightarrow 65\,536$

Counting compatible RNAs: Watson-Crick + > 2 structs

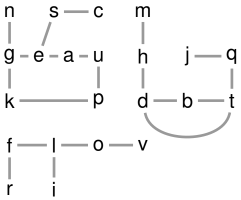


Compatible Base Pairs = Only **Watson-Crick** base pairs



Dependency graph:

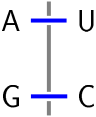
Cycles, Paths, Trees. . .



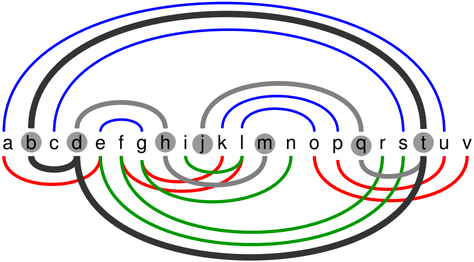
Question: How many **Compatible** sequences?

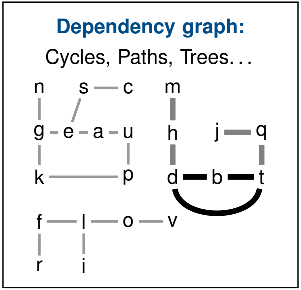
Answer: Non-bipartite → $\varnothing$; Bipartite → $4^{\#\text{CCs}} = 64$

Counting compatible RNAs: Watson-Crick + > 2 structs



Compatible Base Pairs = Only **Watson-Crick** base pairs





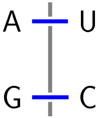
Dependency graph:

Cycles, Paths, Trees...

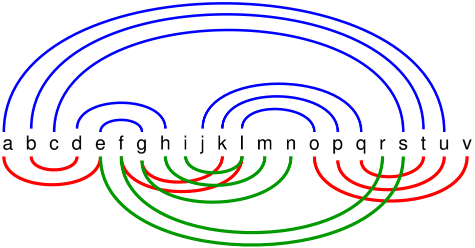
Question: How many **Compatible** sequences?

Answer: Non-bipartite → $\varnothing$; Bipartite → $4^{\#CCs} = 64$

Counting compatible RNAs: Watson-Crick + >2 structs

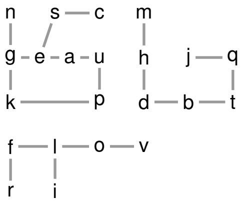


Compatible Base Pairs = Only **Watson-Crick** base pairs



Dependency graph:

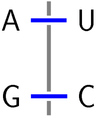
Cycles, Paths, Trees...



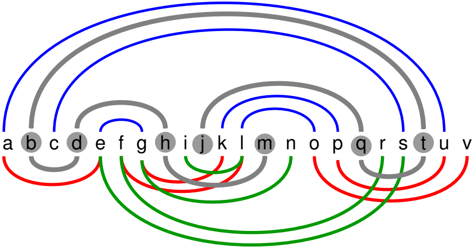
Question: How many **Compatible** sequences?

Answer: Non-bipartite → ∅; Bipartite → $4^{\#CCs} = 64$

Counting compatible RNAs: Watson-Crick + > 2 structs

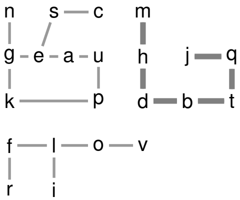


Compatible Base Pairs = Only **Watson-Crick** base pairs



Dependency graph:

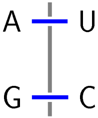
Cycles, Paths, Trees. . .



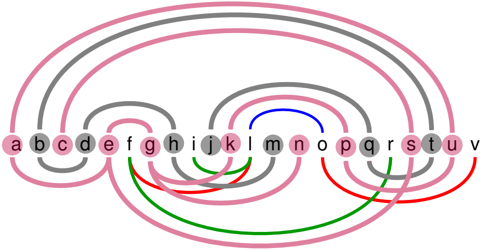
Question: How many **Compatible** sequences?

Answer: Non-bipartite → $\varnothing$; Bipartite → $4^{\#\text{CCs}} = 64$

Counting compatible RNAs: Watson-Crick + > 2 structs

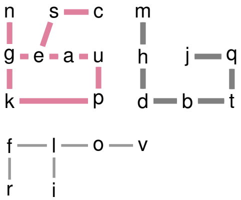


Compatible Base Pairs = Only **Watson-Crick** base pairs



Dependency graph:

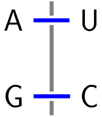
Cycles, Paths, Trees. . .



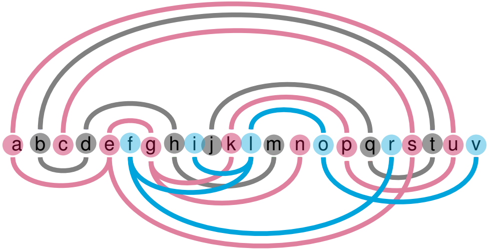
Question: How many **Compatible** sequences?

Answer: Non-bipartite → $\varnothing$; Bipartite → $4^{\#\text{CCs}} = 64$

Counting compatible RNAs: Watson-Crick + > 2 structs

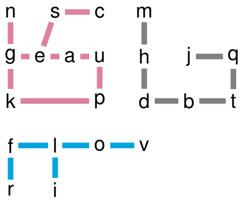


Compatible Base Pairs = Only **Watson-Crick** base pairs



Dependency graph:

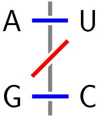
Cycles, Paths, Trees. . .



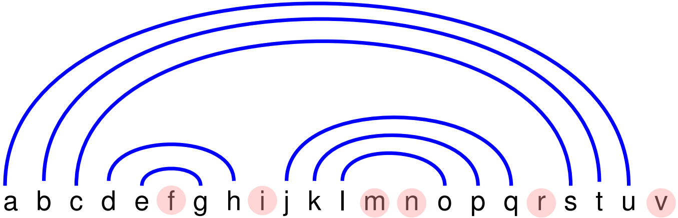
Question: How many **Compatible** sequences?

Answer: Non-bipartite → ∅; Bipartite → $4^{\#\text{CCs}} = 64$

Counting compatible RNAs: WC/Wobble + Single struct.



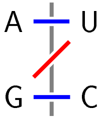
Compatible Base Pairs = Include **Wobble** base pairs



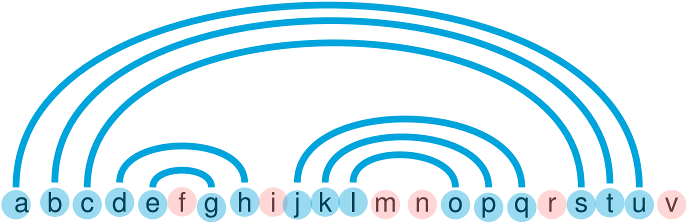
Question: How many **Compatible** sequences?

Answer: $4^{\#\text{Unpaired}} \times 6^{\#\text{BPs}} \rightarrow 6\,879\,707\,136$

Counting compatible RNAs: WC/Wobble + Single struct.



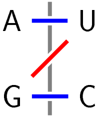
Compatible Base Pairs = Include **Wobble** base pairs



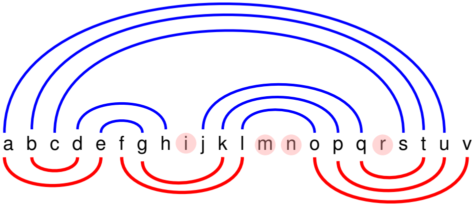
Question: How many **Compatible** sequences?

Answer: $4^{\#\text{Unpaired}} \times 6^{\#\text{BPs}} \rightarrow 6\,879\,707\,136$

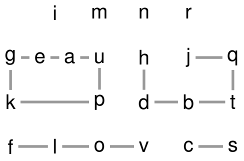
Counting compatible RNAs: WC/Wobble + Two structures



Compatible Base Pairs = Include **Wobble** base pairs



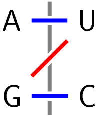
Dependency graph:
Cycles + Paths



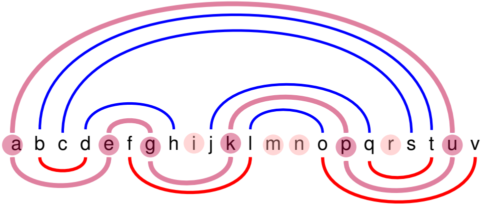
Question: How many **Compatible** sequences?

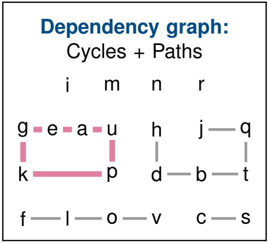
Answer: $\neq \varnothing$! (base-pairs and dependency graphs **always bipartite**)

Counting compatible RNAs: WC/Wobble + Two structures



Compatible Base Pairs = Include **Wobble** base pairs



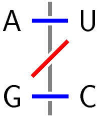


Dependency graph:
Cycles + Paths

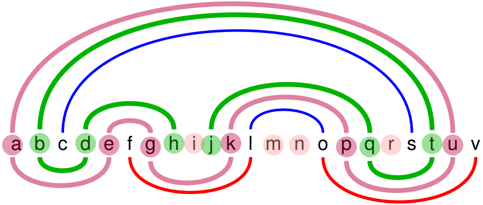
Question: How many **Compatible** sequences?

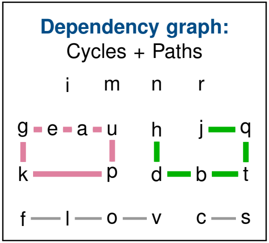
Answer: $\neq \varnothing$! (base-pairs and dependency graphs **always bipartite**)

Counting compatible RNAs: WC/Wobble + Two structures



Compatible Base Pairs = Include **Wobble** base pairs

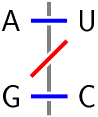




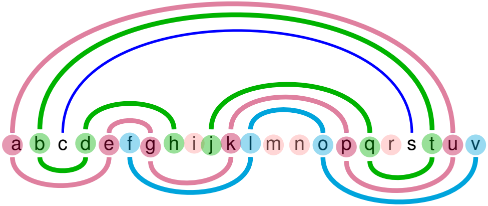
Question: How many **Compatible** sequences?

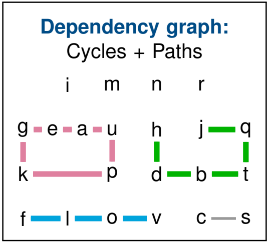
Answer: $\neq \varnothing$! (base-pairs and dependency graphs **always bipartite**)

Counting compatible RNAs: WC/Wobble + Two structures



Compatible Base Pairs = Include **Wobble** base pairs



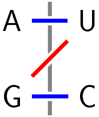

Dependency graph:
Cycles + Paths

Question: How many **Compatible** sequences?

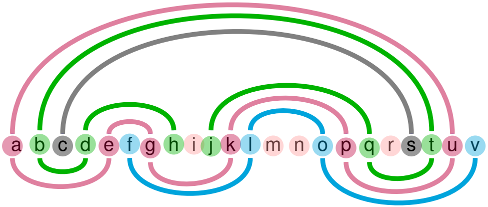
Answer: $\neq \varnothing$! (base-pairs and dependency graphs **always bipartite**)

1st International Computational Biology workshop

Counting compatible RNAs: WC/Wobble + Two structures

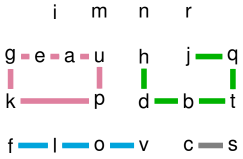


Compatible Base Pairs = Include **Wobble** base pairs



Dependency graph:

Cycles + Paths



Question: How many **Compatible** sequences?

Answer: $\neq \varnothing$! (base-pairs and dependency graphs **always bipartite**)

$$\#\text{Designs}(G) = \prod_{c\in CC(G)} \#\text{Designs}(cc)$$

Counting compatible designs for paths and cycles

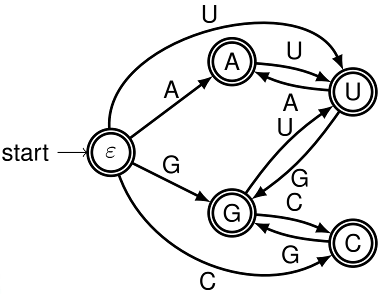
Theorem (#Compatible designs for paths and cycles)

The numbers of compatible designs for paths and cycles of length n are:

$$p(n) = 2\,\mathcal{F}_{n+2} \qquad \textit{and} \qquad c(n) = 2\,\mathcal{F}_n + 4\,\mathcal{F}_{n-1}$$

where $\mathcal{F}_n$: n^{th} ***Fibonacci number***, $\mathcal{F}_0 = 0$, $\mathcal{F}_1 = 1$ *and* $\mathcal{F}_n = \mathcal{F}_{n-1} + \mathcal{F}_{n-2}$.

For paths: A simple DFA generates compatible sequences



Remark: A $\leftrightarrow$ C/G $\leftrightarrow$ U symmetry

Counting compatible designs for paths and cycles

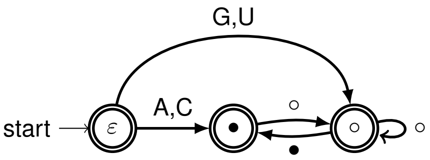
Theorem (#Compatible designs for paths and cycles)

The numbers of compatible designs for paths and cycles of length n are:

$$p(n) = 2\,\mathcal{F}_{n+2} \qquad \text{and} \qquad c(n) = 2\,\mathcal{F}_n + 4\,\mathcal{F}_{n-1}$$

*where $\mathcal{F}_n$: n^{th} **Fibonacci number**, $\mathcal{F}_0 = 0$, $\mathcal{F}_1 = 1$ and $\mathcal{F}_n = \mathcal{F}_{n-1} + \mathcal{F}_{n-2}$.*

For paths: A simple DFA generates compatible sequences



Remark: A ↔ C/G ↔ U symmetry

Counting compatible designs for paths and cycles

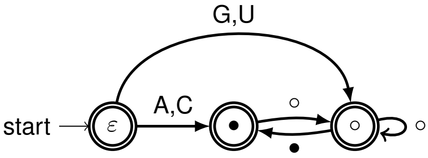
Theorem (#Compatible designs for paths and cycles)

The numbers of compatible designs for paths and cycles of length n are:

$$p(n) = 2\,\mathcal{F}_{n+2} \qquad \textit{and} \qquad c(n) = 2\,\mathcal{F}_n + 4\,\mathcal{F}_{n-1}$$

where $\mathcal{F}_n$: n^{th} ***Fibonacci number****, $\mathcal{F}_0 = 0$, $\mathcal{F}_1 = 1$ and $\mathcal{F}_n = \mathcal{F}_{n-1} + \mathcal{F}_{n-2}$.*

For paths: A simple DFA generates compatible sequences



Remark: A ↔ C/G ↔ U symmetry

$$m_{\bullet}(n) = m_{\circ}(n-1)$$

Counting compatible designs for paths and cycles

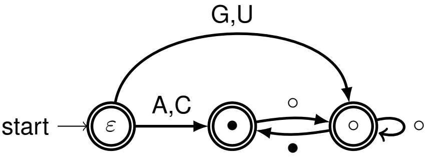
Theorem (#Compatible designs for paths and cycles)

The numbers of compatible designs for paths and cycles of length n are:

$$p(n) = 2\,\mathcal{F}_{n+2} \qquad \text{and} \qquad c(n) = 2\,\mathcal{F}_n + 4\,\mathcal{F}_{n-1}$$

where $\mathcal{F}_n$: n^{th} ***Fibonacci number****, $\mathcal{F}_0 = 0$, $\mathcal{F}_1 = 1$ and $\mathcal{F}_n = \mathcal{F}_{n-1} + \mathcal{F}_{n-2}$.*

For paths: A simple DFA generates compatible sequences



Remark: A ↔ C/G ↔ U symmetry

$$\begin{aligned} m_\bullet(n) &= m_\circ(n-1) \\ m_\circ(n) &= m_\circ(n-1) + m_\bullet(n-1) \\ &= m_\circ(n-1) + m_\circ(n-2) \\ &= \mathcal{F}(n+2) \end{aligned}$$

Counting compatible designs for paths and cycles

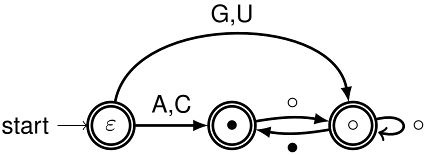
Theorem (#Compatible designs for paths and cycles)

The numbers of compatible designs for paths and cycles of length n are:

$$p(n) = 2\,\mathcal{F}_{n+2} \qquad \textit{and} \qquad c(n) = 2\,\mathcal{F}_n + 4\,\mathcal{F}_{n-1}$$

where $\mathcal{F}_n$: n^{th} ***Fibonacci number***, $\mathcal{F}_0 = 0$, $\mathcal{F}_1 = 1$ *and* $\mathcal{F}_n = \mathcal{F}_{n-1} + \mathcal{F}_{n-2}$.

For paths: A simple DFA generates compatible sequences



Remark: A ↔ C/G ↔ U symmetry

$$\begin{aligned} m_\bullet(n) &= m_\circ(n-1) \\ m_\circ(n) &= m_\circ(n-1) + m_\bullet(n-1) \\ &= m_\circ(n-1) + m_\circ(n-2) \\ &= \mathcal{F}(n+2) \end{aligned}$$

(Since $m_\circ(0) = 1$ and $m_\circ(1) = 2$)

Counting compatible designs for paths and cycles

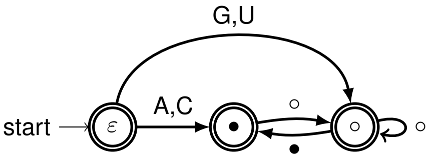
Theorem (#Compatible designs for paths and cycles)

The numbers of compatible designs for paths and cycles of length n are:

$$p(n) = 2\,\mathcal{F}_{n+2} \qquad \text{and} \qquad c(n) = 2\,\mathcal{F}_n + 4\,\mathcal{F}_{n-1}$$

where $\mathcal{F}_n$: n^{th} ***Fibonacci number***, $\mathcal{F}_0 = 0$, $\mathcal{F}_1 = 1$ *and* $\mathcal{F}_n = \mathcal{F}_{n-1} + \mathcal{F}_{n-2}$.

For paths: A simple DFA generates compatible sequences



Remark: A ↔ C/G ↔ U symmetry

$$\begin{aligned}
m_\bullet(n) &= m_\circ(n-1)\\
m_\circ(n) &= m_\circ(n-1) + m_\bullet(n-1)\\
&= m_\circ(n-1) + m_\circ(n-2)\\
&= \mathcal{F}(n+2)
\end{aligned}$$

(Since $m_\circ(0) = 1$ and $m_\circ(1) = 2$)

$$p(n) := m_\varepsilon(n) = 2\,m_\bullet(n-1) + 2\,m_\circ(n-1) = 2(\mathcal{F}(n) + \mathcal{F}(n+1)) =, \mathcal{F}(n+2)$$

Counting compatible designs for paths and cycles

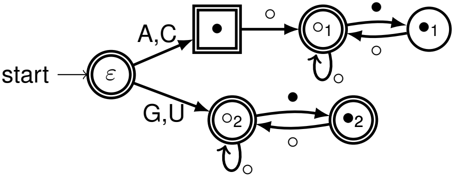
Theorem (#Compatible designs for paths and cycles)

The numbers of compatible designs for paths and cycles of length n are:

$$p(n) = 2\,\mathcal{F}_{n+2} \qquad \textit{and} \qquad c(n) = 2\,\mathcal{F}_{n} + 4\,\mathcal{F}_{n-1}$$

where $\mathcal{F}_n$: n^{th} ***Fibonacci number***, $\mathcal{F}_0 = 0$, $\mathcal{F}_1 = 1$ *and* $\mathcal{F}_n = \mathcal{F}_{n-1} + \mathcal{F}_{n-2}$.

For cycle: A *barely more involved* DFA generates compatible sequences



Remark: A ↔ C/G ↔ U symmetry

$$m_{\circ_2}(n) = \mathcal{F}(n+2)$$
$$m_{\circ_1}(n) = \mathcal{F}(n+1)$$

(Since $m_{\circ_1}(0) = 1$ and $m_{\circ_1}(1) = 1$)

$$\begin{aligned} c(n) := m_{\varepsilon}(n) &= 2\,m_{\circ_1}(n-2) + 2\,m_{\circ_2}(n-1) \\ &= 2(\mathcal{F}(n-1) + \mathcal{F}(n+1)) = 2\,\mathcal{F}(n) + 4\,\mathcal{F}(n-1) \end{aligned}$$

Counting compatible designs for paths and cycles

Theorem (#Compatible designs for paths and cycles)

The numbers of compatible designs for paths and cycles of length n are:

$$p(n) = 2\,\mathcal{F}_{n+2} \qquad \text{and} \qquad c(n) = 2\,\mathcal{F}_n + 4\,\mathcal{F}_{n-1}$$

where $\mathcal{F}_n$: n^{th} ***Fibonacci number****, $\mathcal{F}_0 = 0$, $\mathcal{F}_1 = 1$ and $\mathcal{F}_n = \mathcal{F}_{n-1} + \mathcal{F}_{n-2}$.*

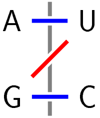
Theorem (#Compatible designs for general 2-structures graphs)

G: dependency graph associated with 2 RNA structures (max deg=2).
The number $\#\text{Designs}(G)$ of compatible designs for G is given by

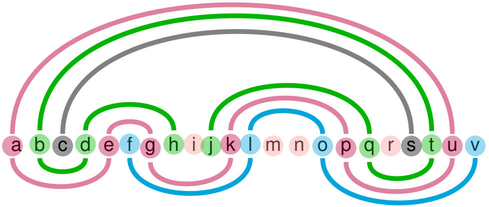
$$\#\text{Designs}(G) = \prod_{p\in\mathcal{P}(G)} 2\,\mathcal{F}_{|p|+2} \times \prod_{c\in\mathcal{C}(G)} \left(2\,\mathcal{F}_{|c|} + 4\,\mathcal{F}_{|c|-1}\right)$$

where G decomposes into paths $\mathcal{P}(G)$ and cycles $\mathcal{C}(G)$.

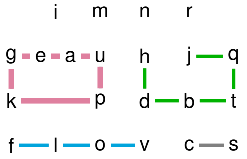
Counting compatible sequences: WC/Wobble + Two structures



Compatible Base Pairs = Include **Wobble** base pairs



Dependency graph:
Cycles + Paths

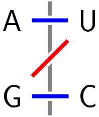


Question: How many **Compatible** sequences?

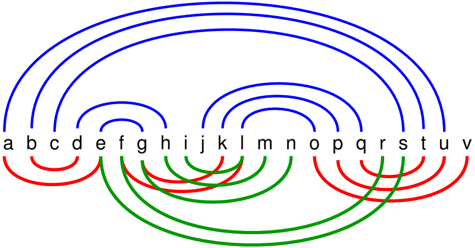
Answer: $\neq \varnothing$! (base-pairs and dependency graphs **always bipartite**)

$$\#\text{Designs}(G) = \prod_{c\in CC(G)} \#\text{Designs}(cc) = 2\,322\,432$$

Counting compatible sequences: Watson-Crick + >2 structures

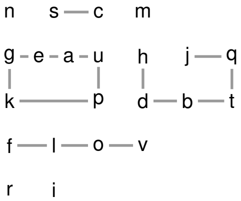


Compatible Base Pairs = Include **Wobble** base pairs



Dependency graph:

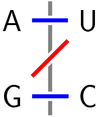
Cycles, Paths, Trees. . .



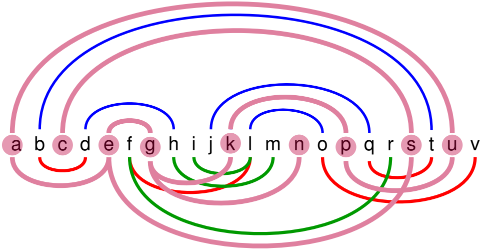
Question: How many **Compatible** sequences?

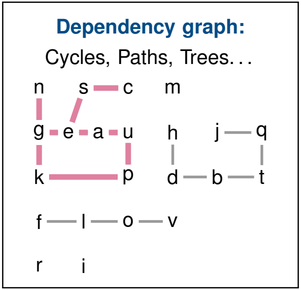
Answer: Non-bipartite → ∅; Bipartite →

Counting compatible sequences: Watson-Crick + >2 structures



Compatible Base Pairs = Include **Wobble** base pairs



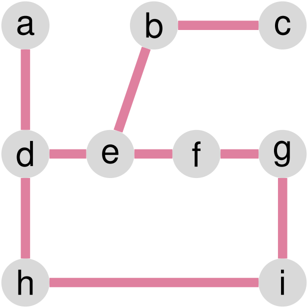

Dependency graph:

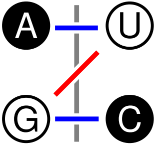
Cycles, Paths, Trees. . .

Question: How many **Compatible** sequences?

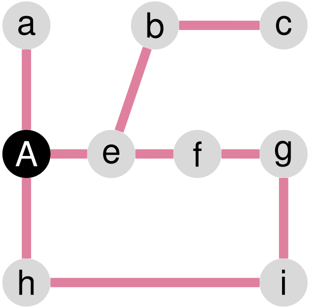
Answer: Non-bipartite → $\varnothing$; Bipartite → $\prod_{cc \in CC(G)} 2 \times \#IS(cc)$

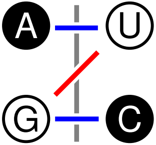
Bijection between Independent Sets and Valid Designs



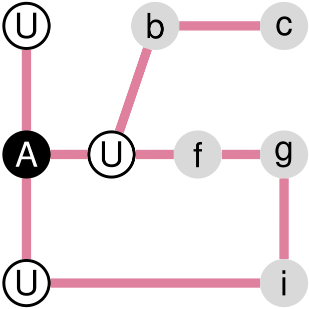


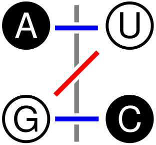
Bijection between Independent Sets and Valid Designs



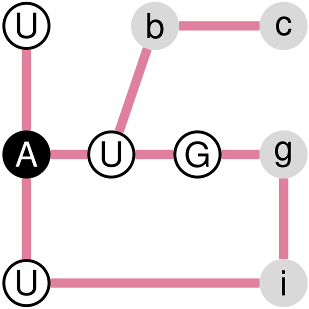


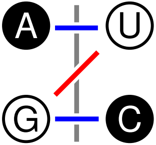
Bijection between Independent Sets and Valid Designs



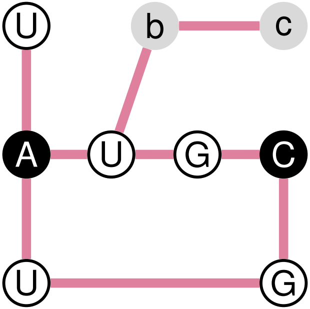


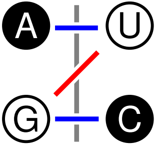
Bijection between Independent Sets and Valid Designs



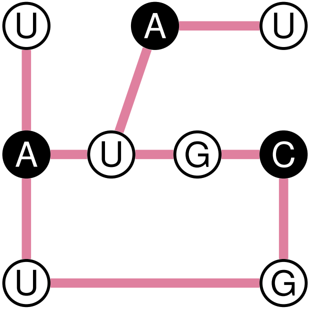


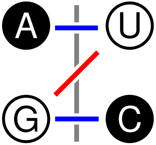
Bijection between Independent Sets and Valid Designs



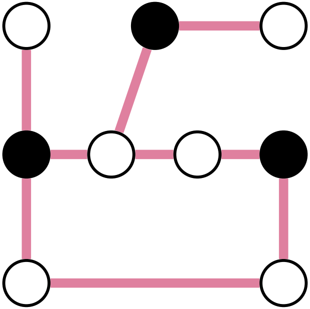


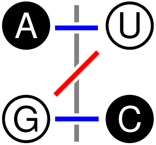
Bijection between Independent Sets and Valid Designs





Bijection between Independent Sets and Valid Designs



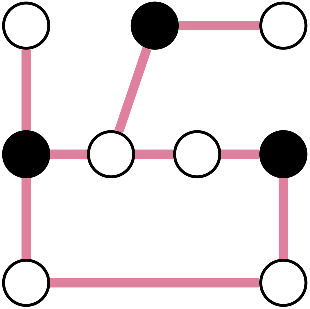


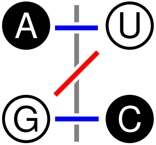
Remark: No adjacent **black letters** in compatible designs

Up to trivial symmetry* (*e.g.* top-left position $\in \{\text{G}, \text{A}\}$):

$$\text{Designs}^\star(\text{cc}) \subseteq \text{IndependentSets}(\text{cc})$$

Bijection between Independent Sets and Valid Designs





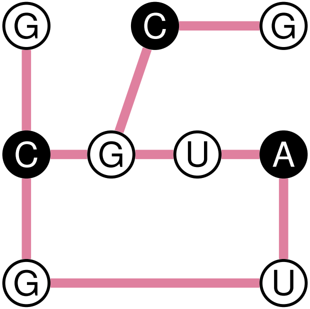
Remark: No adjacent **black letters** in compatible designs

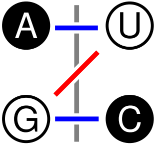
Up to trivial symmetry* (*e.g.* top-left position $\in \{\mathrm{G}, \mathrm{A}\}$):

$$\text{Designs}^\star(\text{cc}) \subseteq \text{IndependentSets}(\text{cc})$$

Also, IS (black) + $\nwarrow$ vert. $\in \{\mathrm{G}, \mathrm{A}\} \Rightarrow$ **Unique** compatible design

Bijection between Independent Sets and Valid Designs





Remark: No adjacent **black letters** in compatible designs

Up to trivial symmetry* (*e.g.* top-left position $\in \{G, A\}$):

$$\text{Designs}^\star(\text{cc}) \subseteq \text{IndependentSets}(\text{cc})$$

Also, IS (black) + $\nwarrow$ vert. $\in \{G, A\} \Rightarrow$ **Unique** compatible design

$\Rightarrow$ Bijection between Designs*(cc) and IndependentSets(cc).

Valid designs and independent sets

Theorem (#Valid design for bipartite connected dependency graphs)

Let G be a ***bipartite connected*** *dependency graph, one has:*

$$\#\text{Designs}(G) = 2 \times \#\text{Designs}^\star(G) = 2 \times \#\text{IS}(G)$$

For a **bipartite** dependency graph G we get:

$$\#Designs(G) = \prod_{cc \in CC(G)} 2 \times \#IS(cc) = 2^{|CC(G)|} \times \#IS(G)$$

But $\#IS(G)$ is #P-hard on bipartite graphs **[Bubbley&Dyer'01]**
(+ Any G is a dependency graph)

Algorithm $\mathcal{A} \in P$ for $\#Designs(G) \rightarrow$ Algorithm $\mathcal{A}' \in P$ for $\#BIS\ldots$

Theorem

#Designs is #P-hard.

No polynomial algorithm for #Designs(G) unless $\#P = FP$ ($\Rightarrow P = NP$)

Valid designs and independent sets

Theorem (#Valid design for bipartite connected dependency graphs)

*Let G be a **bipartite connected** dependency graph, one has:*

$$\#\text{Designs}(G) = 2 \times \#\text{Designs}^{\star}(G) = 2 \times \#\text{IS}(G)$$

For a **bipartite** dependency graph G we get:

$$\#Designs(G) = \prod_{cc \in CC(G)} 2 \times \#IS(cc) = 2^{|CC(G)|} \times \#IS(G)$$

But $\#IS(G)$ is #P-hard on bipartite graphs **[Bubbley&Dyer'01]**
(+ Any G is a dependency graph)

Algorithm $\mathcal{A} \in P$ for $\#Designs(G) \rightarrow$ Algorithm $\mathcal{A}' \in P$ for $\#BIS\ldots$

Theorem

#Designs is #P-hard.

No polynomial algorithm for #Designs(G) unless $\#P = FP$ ($\Rightarrow P = NP$)

Valid designs and independent sets

Theorem (#Valid design for bipartite connected dependency graphs)

*Let G be a **bipartite connected** dependency graph, one has:*

$$\#\text{Designs}(\text{G}) = 2 \times \#\text{Designs}^{\star}(\text{G}) = 2 \times \#\text{IS}(\text{G})$$

For a **bipartite** dependency graph G we get:

$$\#Designs(G) = \prod_{cc \in CC(G)} 2 \times \#IS(cc) = 2^{|CC(G)|} \times \#IS(G)$$

But $\#IS(G)$ is #P-hard on bipartite graphs **[Bubbley&Dyer'01]**
(+ Any G is a dependency graph)

Algorithm $\mathcal{A} \in P$ for $\#Designs(G) \rightarrow$ Algorithm $\mathcal{A}' \in P$ for $\#BIS\ldots$

Theorem

#Designs is #P-hard.

No polynomial algorithm for #Designs(G) unless $\#P = FP$ ($\Rightarrow P = NP$)

Valid designs and independent sets

Theorem (#Valid design for bipartite connected dependency graphs)

Let G be a ***bipartite connected*** *dependency graph, one has:*

$$\#\text{Designs(G)} = 2 \times \#\text{Designs}^{\star}\text{(G)} = 2 \times \#\text{IS(G)}$$

For a **bipartite** dependency graph G we get:

$$\#Designs(G) = \prod_{cc \in CC(G)} 2 \times \#IS(cc) = 2^{|CC(G)|} \times \#IS(G)$$

But $\#IS(G)$ is $\#$P-hard on bipartite graphs **[Bubbley&Dyer'01]**
(+ Any G is a dependency graph)

Algorithm $\mathcal{A} \in P$ for $\#Designs(G) \rightarrow$ Algorithm $\mathcal{A}' \in P$ for $\#BIS\ldots$

Theorem

#Designs is #P-hard.

No polynomial algorithm for #Designs(G) unless $\#P = FP$ ($\Rightarrow P = NP$)

Consequences

Corollary (#Approximability for ≤ 5 structures) [Weitz'06]

For any G built from ≤ 5 pseudoknotted structures, #Design(G) can be approximated within **any ratio** in **polynomial time** (PTAS)

Corollary (#BIS hardness for > 5 struct.) [Cai, Galanis, Goldberg, Jerrum, McQuillan'16]

Beyond 5 **pseudoknotted** structures, approximating #Design becomes **as hard as** approximating #BIS without any constraint.

Why pseudoknotted? Because any bipartite graph of max degree Δ can be **decomposed** into Δ matchings **in polynomial time** (Vizing's theorem).

Lastly, connection between **counting** and **sampling** [Jerrum, Valiant, Vazirani'86].

Conjecture (#BIS hardness of sampling)

Generating comp. sequences **(almost) uniformly** for general input is **#BIS-hard**.

Consequences

Corollary (#Approximability for ≤ 5 structures) [Weitz'06]

For any G built from ≤ 5 pseudoknotted structures, #Design(G) can be approximated within **any ratio** in **polynomial time** (PTAS)

Corollary (#BIS hardness for > 5 struct.) [Cai, Galanis, Goldberg, Jerrum, McQuillan'16]

Beyond 5 **pseudoknotted** structures, approximating #Design becomes **as hard as** approximating #BIS without any constraint.

Why pseudoknotted? Because any bipartite graph of max degree Δ can be **decomposed** into Δ matchings **in polynomial time** (Vizing's theorem).

Lastly, connection between **counting** and **sampling** [Jerrum, Valiant, Vazirani'86].

Conjecture (#BIS hardness of sampling)

Generating comp. sequences **(almost) uniformly** for general input is **#BIS-hard**.

Consequences

Corollary (#Approximability for ≤ 5 structures) [Weitz'06]

For any G built from ≤ 5 pseudoknotted structures, #Design(G) can be approximated within **any ratio** in **polynomial time** (PTAS)

Corollary (#BIS hardness for > 5 struct.) [Cai, Galanis, Goldberg, Jerrum, McQuillan'16]

Beyond 5 **pseudoknotted** structures, approximating #Design becomes **as hard as** approximating #BIS without any constraint.

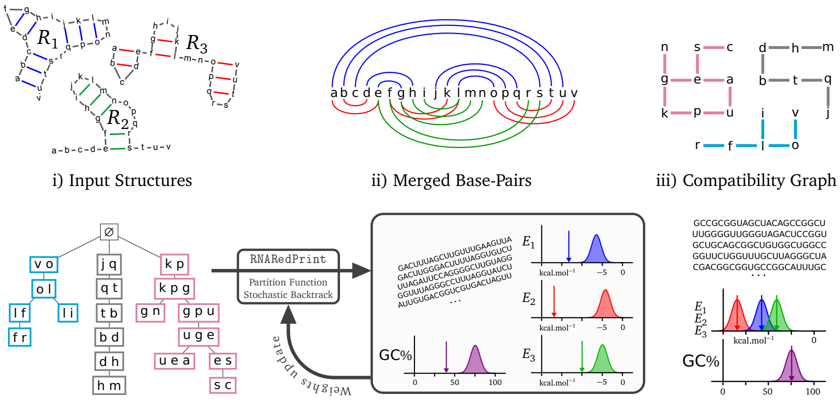
Why pseudoknotted? Because any bipartite graph of max degree Δ can be **decomposed** into Δ matchings **in polynomial time** (Vizing's theorem).

Lastly, connection between **counting** and **sampling** **[Jerrum, Valiant, Vazirani'86]**.

Conjecture (#BIS hardness of sampling)

Generating comp. sequences **(almost) uniformly** for general input is **#BIS-hard**.

Perspectives: FPT and Boltzmann sampling algorithms



i) Input Structures

ii) Merged Base-Pairs

iii) Compatibility Graph

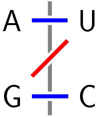
iv) Tree Decomposition

v) Weight Optimization (Adaptive Sampling)

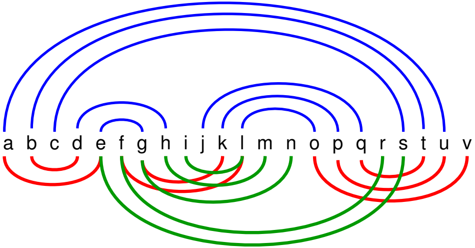
vi) Final Designs

- **FPT algorithm** for counting based on **tree decomposition**
- Multidimensional Boltzmann sampling to control energies, GC. . .

Counting compatible sequences: Watson-Crick + >2 structures

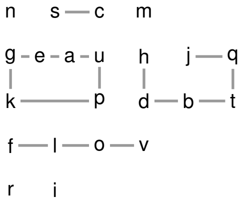


Compatible Base Pairs = Include **Wobble** base pairs



Dependency graph:

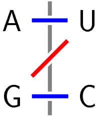
Cycles, Paths, Trees...



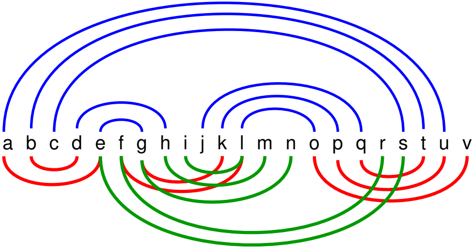
Question: How many **Compatible** sequences?

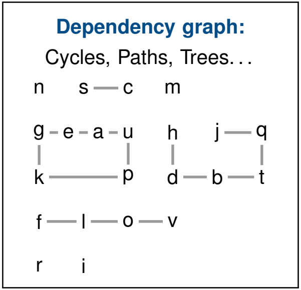
Answer: Bipartite →

Counting compatible sequences: Watson-Crick + >2 structures



Compatible Base Pairs = Include **Wobble** base pairs





Dependency graph:

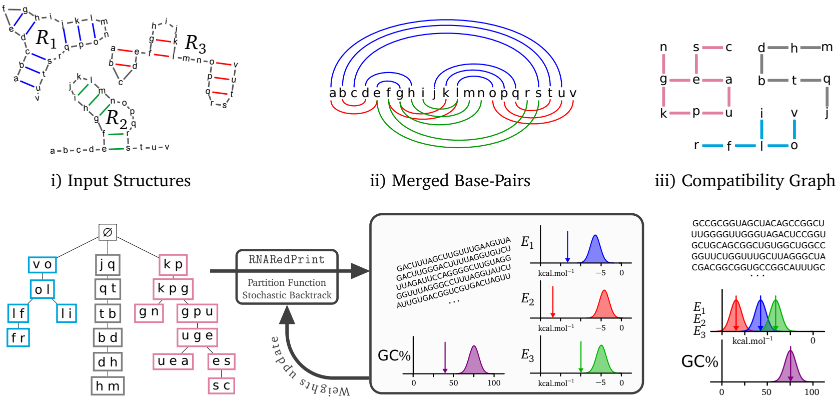
Cycles, Paths, Trees. . .

Question: How many **Compatible** sequences?

Answer: Bipartite →

$$\prod_{cc\in CC(G)} 2 \times \#IS(cc) = 496\,672$$

Perspectives: FPT and Boltzmann sampling algorithms



i) Input Structures

ii) Merged Base-Pairs

iii) Compatibility Graph

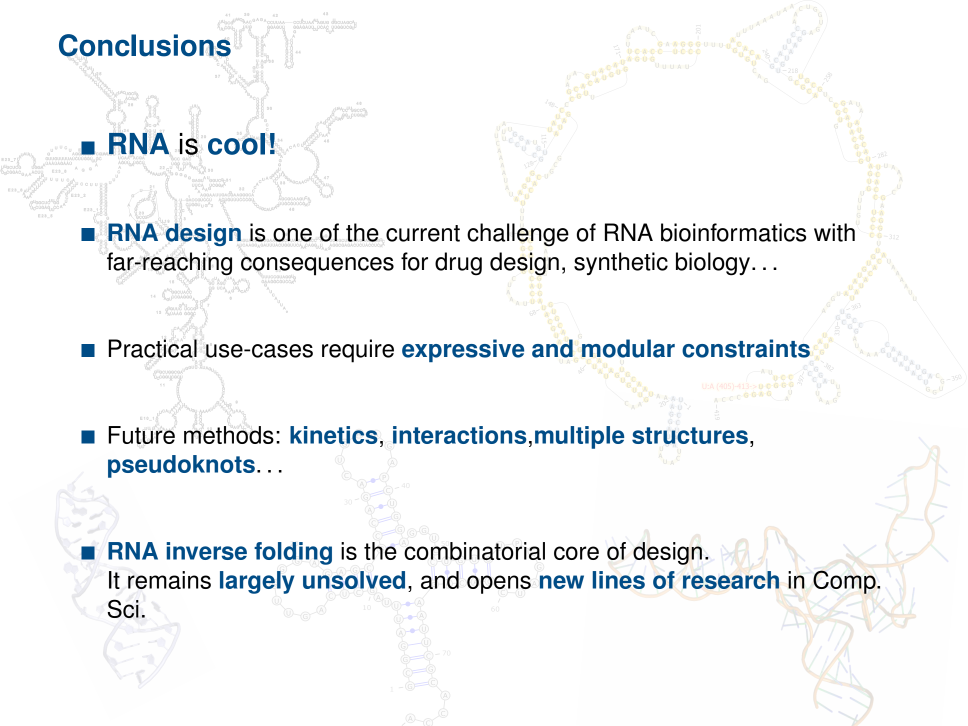
iv) Tree Decomposition

v) Weight Optimization (Adaptive Sampling)

vi) Final Designs

- **FPT algorithm** for counting based on **tree decomposition**
- **Multidimensional Boltzmann sampling** to control energies, GC. . .

Conclusions



- **RNA** is **cool!**
- **RNA design** is one of the current challenge of RNA bioinformatics with far-reaching consequences for drug design, synthetic biology. . .
- Practical use-cases require **expressive and modular constraints**
- Future methods: **kinetics**, **interactions**,**multiple structures**, **pseudoknots**. . .
- **RNA inverse folding** is the combinatorial core of design.
  It remains **largely unsolved**, and opens **new lines of research** in Comp. Sci.

Collaborators

University McGill

Vladimir Reinharz
Jérôme Waldispühl

MIT

Bonnie Berger
Srinivas Devadas
Alex Levin
Mieszko Lis
Charles O'Donnell

LRI – Univ. Paris Sud

Alain Denise
Vincent Le Gallic

Wuhan University

Yi Zhang
Yu Zhou

LIGM – Marne la Vallée

Stéphane Vialette

LIX – Ecole Polytechnique

Alice Héliou
Mireille Regnier

Simon Fraser University

Jozef Hales
Jan Manuch (UBC)
Ladislav Stacho

Cédric Chauve
Julien Courtiel

TBI Vienna

Ronnie Lorenz
Andrea Tanzer

Agence Nationale de la Recherche ANR

Inria INVENTEURS DU MONDE NUMÉRIQUE

cnrs dépasser les frontières

Thanks!

Poster submission & **Registration** open soon. . .
(+ ISCB travel fellowships for students)

References

R. Nussinov and A.B. Jacobson.

Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proc Natl Acad Sci U S A*, 77:6903–13, 1980.