



RNA Faltung - II

W1-High-throughput Genomics, FU Berlin
OWL RNA Bioinformatics, MPI Molgen Berlin
05.12.16

Nussinov Nachteile



Die Maximierung der Anzahl von Bp führt unter Umständen nicht zu biologisch relevante Strukturen

- Stapeln von Basenpaaren sind nicht berücksichtigt. Dies beeinflusst die Struktur und Stabilität der Helices



- Größen von internen Schleifen sind nicht berücksichtigt



instabil



stabil



instabil



- ❑ realistischer Ansatz: Basiert auf Thermodynamik und Statistischer Mechanik
- ❑ Die Stabilität einer RNA-Struktur stimmt mit der thermodynamischen Stabilität überein.
- ❑ Die Minimum Free Energy (MFE) ist quantifiziert als die Menge der freien Energie, welche durch die Bildung von Basenpaaren freigesetzt wurde

Zucker Algorithmus: die Idee



- ❑ **Vorhersage der Sekundärstruktur mit der Thermodynamik**
- ❑ **Energie Minimierung:** Die richtige Struktur ist die **mit der niedrigsten freier Energie im Equilibrium**
- ❑ Man kann die Beiträge für die freie Energie von einzelnen Schleifen messen
Eine komplette Liste von energetischen Beiträgen stellt sich so dar:
 - Freie Energie von Basenpaaren (stabilisierend)
 - Freie Energie von Basen Stapeln (stabilisierend)
 - Freie Energie von End Schleifen (destabilisierend)
 - Freie Energie von Inneren Schleifen (destabilisierend)
 - Freie Energie von Bulges (destabilisierend)

Minimierung der Energie



Freie Energie des bp (x_i, x_j) führt zu besseren Ergebnissen

$e(x_i, x_j)$ ist die Menge der freien Energie für „die Basenpaar (x_i, x_j) “

Die Energiewerte für C-G, A-U and G-U Basenpaare wurden experimentell bei 37 Grad gemessen.

Nur wenige Änderungen an dem Nussinov Algorithmus würden ihn in ein **Minimierungs Problem** konvertieren



Wie?
An der Tafel



- Angemessene Schätzungen für freie Energie der Basenpaare C-G, A-U und G-U bei 37C sin -12 kJ/mol, -8 kJ/mol und -4 kJ/mol
- Eine einfache Definition von $e(i, j)$ wäre, beispielsweise, diese:

$$e(i, j) = \begin{cases} -12, & \text{if } \{i, j\} = \{C, G\} \\ -8 & \text{if } \{i, j\} = \{A, U\} \\ -4, & \text{if } \{i, j\} = \{G, U\} \end{cases}$$

- Die Gesamtenergie $E(s, P)$ einer Sequenz s , die in die Sekundärstruktur P faltet ist dann die Summe der Basenpaarbeiträge:

$$E(s, P) = \sum_{(i, j) \in P} e(s_i, s_j)$$

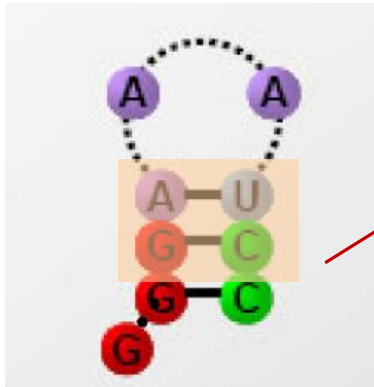


- Leider führt diese Formulierung des Nussinov Algorithmus nicht immer zu guten Strukturen
- Die kumulative Wirkung von Stapeln und Schleifen würde dabei nicht beachtet
- Es berücksichtigt nicht die destabilisierende Wirkung von Schleifen
- Wir brauchen einen besseren Ansatz (aber wir wollen die Idee der Energie Minimierung halten - es ist aus thermodynamischer Sicht sinnvoll)
 - **Wir brauchen eine bessere Energiefunktion (Energie score)!!**

Sequenzabhängige freie Energie



Nächster Nachbar Regel -> Dann müssen wir eine Menge von Regeln definieren, die die Sequenzabhängigkeit beachten



Was ist die freie Energie des GC Basenpaars, wenn AU das Vorherige Basenpaar ist?

Energie wird nur von der vorherigen Basenpaar beeinflusst (nicht von Basenpaaren weiter unten).

Der Gesamtenergie = Summe über Stabilität verschiedener Motive / Schleifen
Die Energie wurde experimentell geschätzt aus kleinen synthetischen RNAs

Beispiel Werte: GC GC GC GC
AU GC CG UA
-2.3 -2.9 -3.4 -2.1



Zuker algorithmus: Vorhersage der Sekundärstruktur mit der Thermodynamik (proposed by Zuker and Stiegler in 1981)

Zuker Algorithmus: die Idee



Man kann die Beiträge für die freie Energie von einzelnen Schleifen messen
Eine komplette Liste von energetischen Beiträgen stellt sich so dar:

- Freie Energie von Basenpaaren (stabilisierend)
- Freie Energie von Basen Stapeln (stabilisierend)
- Freie Energie von End Schleifen (destabilisierend)
- Freie Energie von Inneren Schleifen (destabilisierend)
- Freie Energie von Bulges (destabilisierend)

Kernidee: die Struktur in Loops und gestapelte Basenpaare zerlegen,
(anstelle von einzelnen Basenpaaren)

Freie Energien sind additiv. Die Energie einer RNA-Struktur ist die Summe aus den einzelnen

Wichtiger Unterschied zu Nussinov: *Es werden Energien aus Schleifen und nicht von einzelnen Basenpaare gerechnet.*

K-loop-abhängige Energien



Nur gestapelte Basenpaare liefern einen negativen Beitrag zur Energie

Wir bezeichnen die Energie der gestapelten Paare (i, j) und (k, l) in der ‚stem loop‘ durch (i, j) geschlossen als $e_s(i, j)$.

e_s	A/U	C/G	G/C	U/A	G/U	U/G
A/U	-0.9	-1.8	-2.3	-1.1	-1.1	-0.8
C/G	-1.7	-2.9	-3.4	-2.3	-2.1	-1.4
G/C	-2.1	-2.0	-2.9	-1.8	-1.9	-1.2
U/A	-0.9	-1.7	-2.1	-0.9	-1.0	-0.5
G/U	-0.5	-1.2	-1.4	-0.8	-0.4	-0.2
U/G	-1.0	-1.9	-2.1	-1.1	-1.5	-0.4

Freienergie in kcal/mol bei 37C

K-loop-abhängige Energien



Alle k-Schleifen tragen positive Energien zur gesamten freien Energie bei.

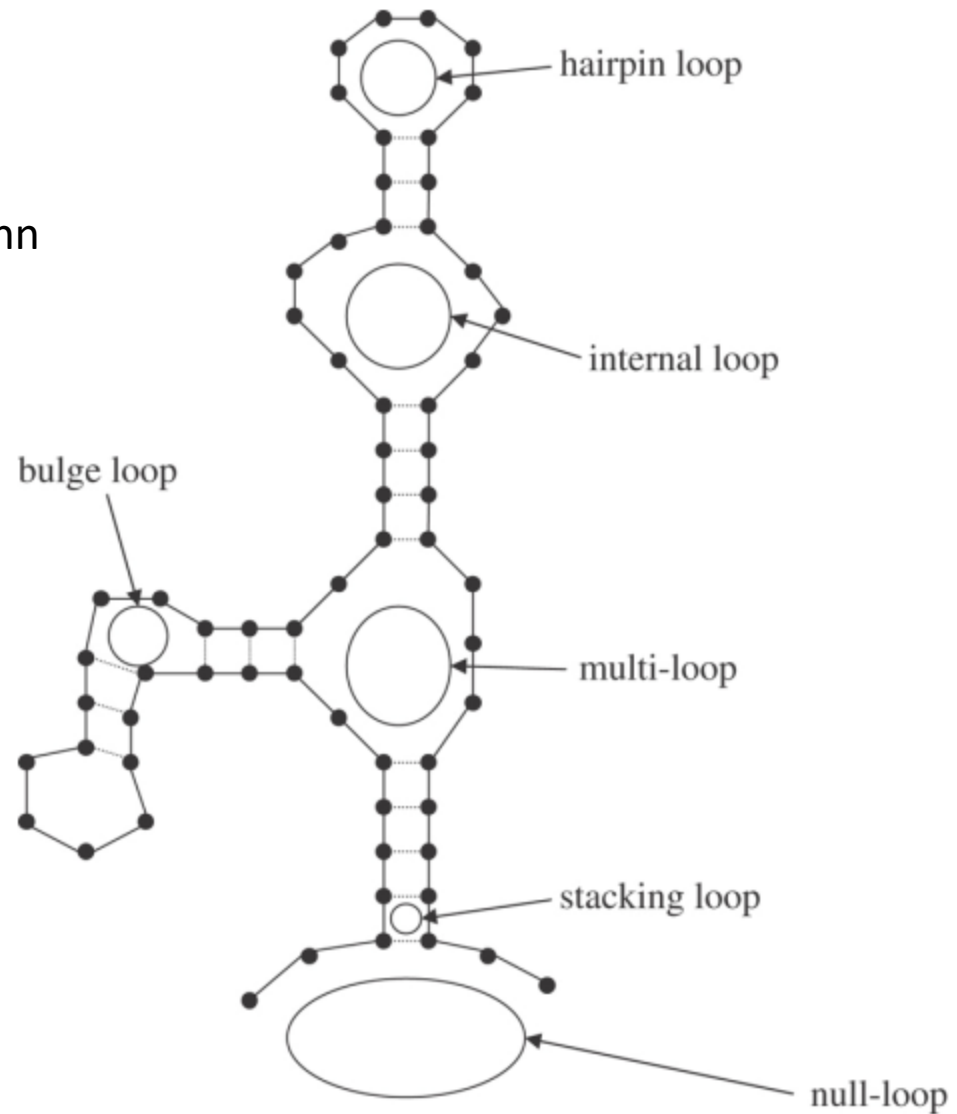
k	interior loop	bulge	hairpin
1	-	3.9	-
2	4.1	3.1	-
3	5.1	3.5	4.1
4	4.9	4.2	4.9
5	5.3	4.8	4.4
6	6.3	5.5	5.3
...			

Freienergie in kcal/mol bei 37C

K-loop Decomposition



- Zucker Algorithmus ist auf k-loop Dekomposition basiert.
- Es basiert auf die Sache, dass Ich kann eine complexe Struktur in seiner Grundsekundaerstrukturelemente zerlegen!
- Wir werden eine formale Definition von diese Strukturelemente geben.



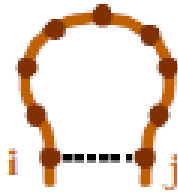
Bemerkung (1)



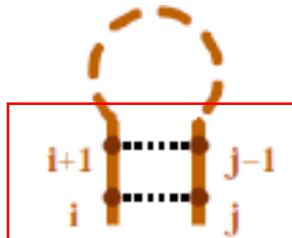
- Normalerweise müssen hairpin loops mindestens 3 Nukleotiden beinhalten -> für jede hairpin loop (i,j) von P gilt die Bedingung:
 $i < j-3$
- Jede Sekundärstrukturelement ist durch **ihre Abschlussbasenpaar** definiert.

Sei N eine Sequenz und P eine Struktur für N

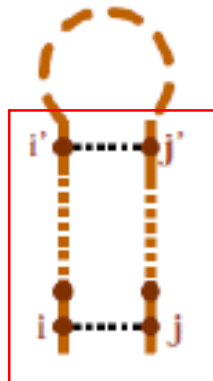
- Basenpaar (i,j) in P ist ein **hairpin loop**
k-loop wird 1-loop genannt falls es durch ein Basenpaar geschlossen wird



- Stacking loop: Basenpaar (i,j) in P ist ein **stacking**
k loop wird 2-loop genannt falls es durch zwei Basenpaare geschlossen wird



- Basenpaaren (i,j) und (i',j') in P formen ein **internal loop**
k-loop wird 2-loop genannt weil es wurde durch zwei Basenpaare geschlossen
 - es gibt keine Basenpaare zwischen (i,j) und (i',j')

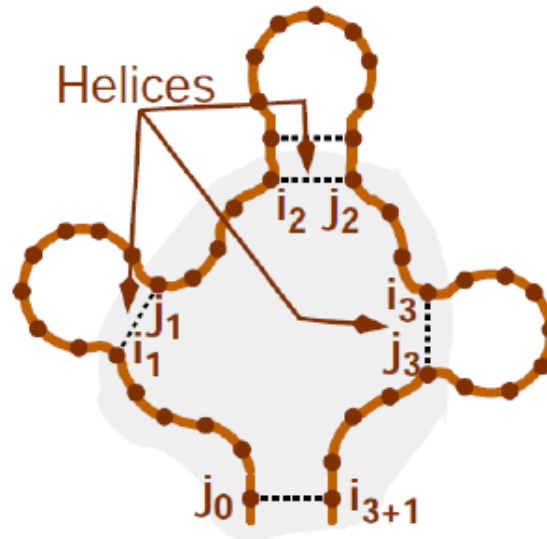


Ein internal loop wird **left (right) bulge genannt**, falls $j-j' > 1$ oder $i'-i > 1$ (aber nicht beide). **Noch $k=2$, 2-loop.**

Ein k -multiloop besteht aus mehreren Basenpaaren $(i_1, j_1) \dots (i_k, j_k)$ mit einem Schlussbasenpaar (j_0, i_{k+1}) . Es wurde durch mehr als zwei Basenpaare geschlossen.

pairs ($k \geq 3$, **k-loop**)

- $(i_1, j_1) \dots (i_k, j_k)$ werden helices von den multiloop genannt



Multiloop oder Multi-branched Loop

Bemerkung (2)



Bei einer Sequenz s , jedes Sekundärstruktur P auf s teilt die Menge $\{1.....L\}$ in k-loops $S_0, S_1, ...S_m$ auf.

Jeder k-loop hat eine Energie $e(s_i)$ Und die Energie aus einer Struktur P ist

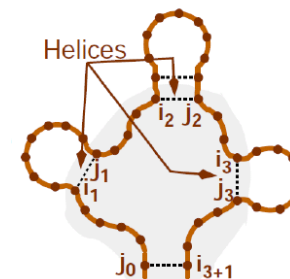
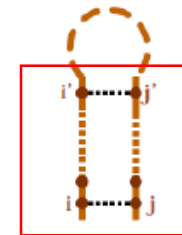
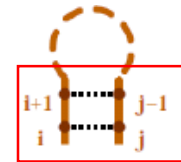
$$E(s, P) = \sum_{k=0}^m e(s_i)$$

Energie ist eine Funktion der k-loops statt eine Funktion der Basenpaare!

Jede $e(s_i)$ wird aus den Nachbarn Basen (nach den Nächster Nachbar Regeln) und den Energiewerten der Schleifen unterschiedlicher Länge berechnet werden

Die Energie der gesamten Sekundärstruktur besteht aus den folgenden Beiträge:

- **Hairpin loop** durch (i, j) geschlossen
 $e_h(i, j)$
- **Stacked base pair** (i, j)
 $e_s(i, j, i - 1, j - 1)$
- **Internal loop** or buldge (i, j, i', j')
 $e_{bl}(i, j, i', j')$
- **K-multiloop**
 $e_{ml}(j_0, i_1, j_1, \dots, i_k, j_k, i_{k+1})$
Berechnung für alle
möglichen k-loops





Input: Eine Sequenz S

Output: Ein Satz von Basenpaaren P , die die sekundäre Struktur mit Minimale Freie Energie von s beschreibt

- Der Zuker Algorithmus findet eine minimale Freie Energie Sekundärstruktur S mit eine Energiefunktion für k -Schleifen (durch dynamische Programmierung)
- **Im Gegensatz zum Nussinovs Algorithmus ist die Rekursion nun auf k -Schleifen zentriert Auf , nicht auf Basenpaaren!**
- Die Rekursion ist etwas komplizierter, es benutzt jetzt zwei DP Matrizen, V und W

Zucker algorithmus – Rekursionschritt



Für alle i, j mit $1 \leq i < j \leq L$ (*Sequenzlänge*):

Minimale Freie Energie

für die Struktur die von Subsequenz s_{ij}
geformt wurde

$$W(i, j) = \min \left\{ \begin{array}{l} W(i + 1, j) \\ W(i, j - 1) \\ V(i, j) \\ \min_{i < j < k} (W(i, k) + W(k + 1, j)) \end{array} \right.$$

Wir berechnen die bekannten Fälle:

1. i ist ungepaart
2. j ist ungepaart
3. i und j sind miteinander gepaart (und schließen ein k-loop)
4. i und j sind gepaart aber nicht miteinander

Die minimale Faltungsenergie $W_{\min}$ ist gegeben durch $W(1, L)$

Zucker algorithmus – Rekursionschritt



Für alle i, j mit $1 \leq i < j \leq L$ (*Sequenzlänge*):

$$W(i, j) = \min \left\{ \begin{array}{l} W(i + 1, j) \\ W(i, j - 1) \\ \boxed{V(i, j)} \rightarrow \\ \min_{i < j < k} (W(i, k) + W(k + 1, j)) \end{array} \right.$$

Minimale Freie Energie
für die Struktur die von Subsequenz s_{ij}
geformt wurde, in der i und j gepaart sind
 $V(i, j) = \min\{E_1, E_2, E_3, E_4\}$

Wir berechnen die bekannten Fälle:

1. i ist ungepaart
2. j ist ungepaart
3. i und j sind miteinander gepaart (und schließen ein k-loop)
4. i und j sind gepaart aber nicht miteinander

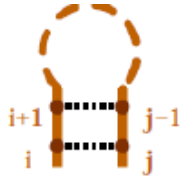
Die minimale Faltungsennergie $W_{\min}$ ist gegeben durch $W(1, L)$

Zucker: loop Zerlegung, rekursive Berechnung von $V(i,j)$



Hairpin loop

Fall 1, $E_1 = e_h(i, j)$



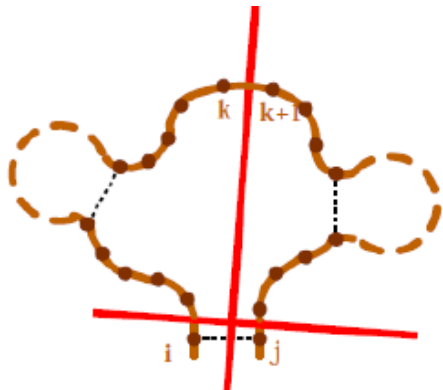
Stapeln

Fall 2, $E_2 = e_s(i, j) +$



Innere Schleife

Fall 3, $E_3 = e_{bl}(i, j) +$



+



Fall 4, $E_4 = e_{ml}$



Gibt es viele Möglichkeiten,
um ein Multilop zu teilen

Zucker: loop Zerlegung, rekursive Berechnung von $V(i,j)$



- Zusammenfassend, $V(i,j)$ ist die minimale freie Energie, die in vier Wegen berechnet werden kann:

$$V(i,j) = \min \{ \mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3, \mathbf{E}_4 \}$$

$$V(i,j) = \min \begin{cases} e_h(i,j) & \text{Hairpin} \\ e_s(i,j) & \text{Stem} \\ V_{BI}(i,j) & \text{Internal loop} \\ \mathbf{V_M(i,j)} & \text{mehr kompliziert..} \end{cases}$$

$\mathbf{E}_4$

Der Multiloop muss mindestens einmal geteilt werden, sonst ist es ein einfacher loop

Idee: schneide Teile des Multilops bis nur einzelnen hairpin loops übrig sind! $\mathbf{E}_4$ wird auch rekursiv berechnet!

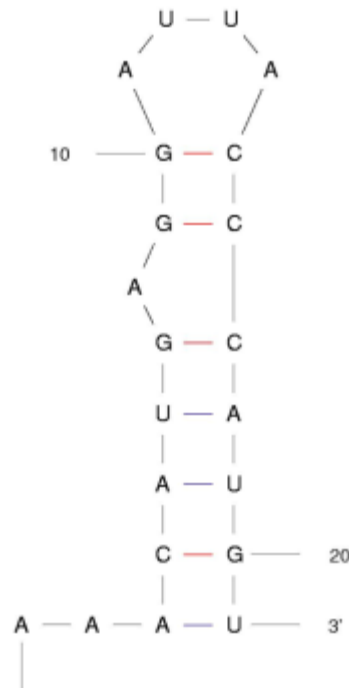


An der Tafel!

Beispiel - MFOLD web server



- Michael Zuker bietet einen Web-Server, der verwendet werden kann, um RNA-Sequenzen zu falten. Beispiel: **AAACAUGAGGAUUACCCAUGU**
- Es gibt eine Strukturdarstellung und thermodynamische Details aus



Loop Free-Energy Decomposition

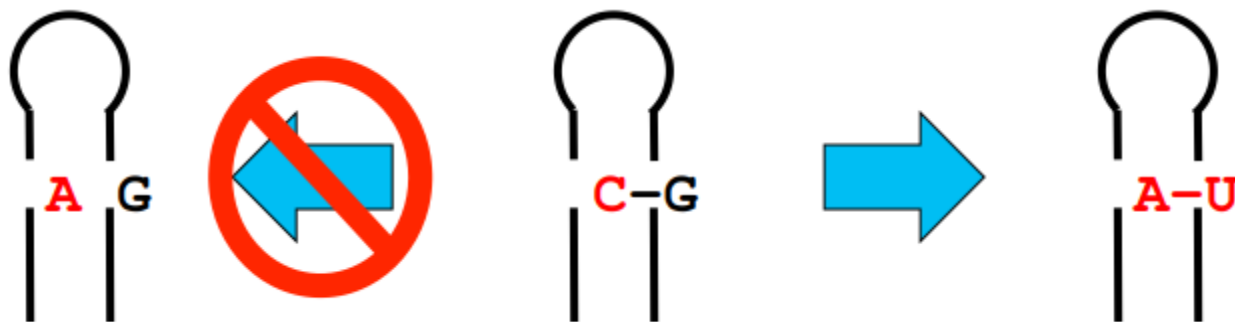
Structure 1
08Apr21-03-47-23
 $\Delta G = -5.60$

Structural element	δG	Information
External loop	0.20	2 ss bases & 1 closing helices.
Stack	-2.20	External closing pair is A ³ -U ²¹
Stack	-2.10	External closing pair is C ⁴ -G ²⁰
Stack	-1.10	External closing pair is A ⁵ -U ¹⁹
Stack	-2.10	External closing pair is U ⁶ -A ¹⁸
Helix	-7.50	5 base pairs.
Bulge loop	0.50	External closing pair is G ⁷ -C ¹⁷
Stack	-3.30	External closing pair is G ⁹ -C ¹⁶
Helix	-3.30	2 base pairs.
Hairpin loop	4.50	Closing pair is G ¹⁰ -C ¹⁵

Faltung durch Komparative Analyse



- Die Nussinov und Zucker Algorithmen berechnen die Beste RNA Faltung durch die Optimierung einer bestimmten objectiven Funktion.
- Obwohl Energieminimierungstechniken attraktiv sind, zu Zeit funktionieren fast alle die RNA-Sekundärstrukturvorhersage Methoden mit **komparativer Analyse**. (looking at sequences of related structures) -> **RNA-Struktur kann hoch konserviert sein, auch wann Sequenzähnlichkeit vollständig weg ist!**
- Der Mechanismus, der die Struktur erhält, heißt **compensatory mutations**.



Faltung durch Komparative Analyse



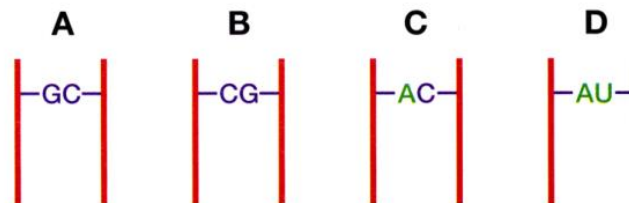
- Allerdings, brauchen komparative Methoden viele unterschiedliche Sequenzen als Input und hochwertigen multiplen Alignments gut zu funktionieren.

Ein erfolgreicher Ansatz basiert auf den Vergleich der Sequenz-Struktur von RNAs.

I. Sequence alignment

```
seq 1.  ---G-----C---
seq 2.  ---C-----G---
seq 3.  ---A-----C---
seq 4.  ---A-----T---
```

II. Structural alignment

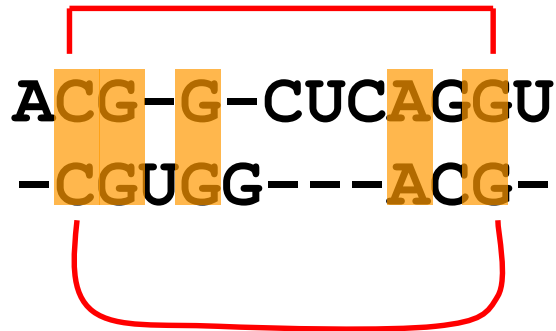


Eine Analyse der Sequenzkovarianz in verwandten RNAs helfen, um Positionen zu identifizieren, die Basenpaare bilden -> wir müssen die kovariante Spalten finden!

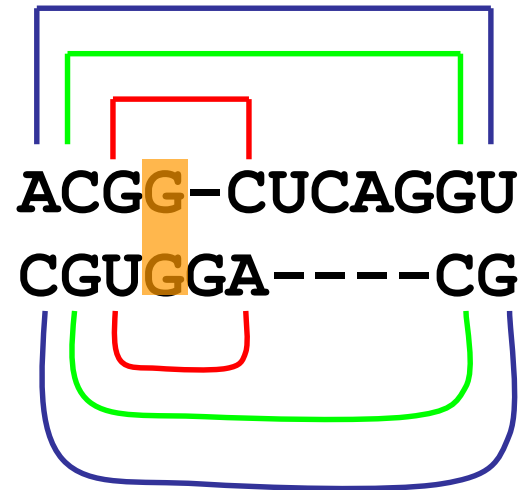
Faltung durch Komparative Analyse



Z.b. Hier zwei RNA Alignments: eines bewahrt nicht die sekundäre Struktur, während das andere diese bewahrt (und damit bewahrt die Interaktionen)



Alignment mit
hohem Sequenz Score
-> Sekundärstruktur
nicht erhalten



Alignment mit
hohem Interaktions Score
-> Sequenz nicht erhalten

Faltung durch Komparative Analyse



Die **Schlüsselidee** besteht darin, die **Interaktionen** zu identifizieren (das ist der Watson-Crick korrelierten Positionen) in einem Multiple Alignment und Sie verwenden um die Sekundärstruktur zu vorhersagen
Mutual information content (**gemeinsamer Informationsgehalt**)

Das ist ein eingegebene MSA. Welche sind die konservierte Interaktionen (die keine Sequenzkonservierung erhalten)?

Seq1	GCCUUCGGGC
Seq2	GACUUCGGUC
Seq3	GGCUUCGGCC
Seq4	GACUUUGGUC

Faltung durch Komparative Analyse



Seq1	G C CUUCGG G C
Seq2	G A CUUCGG U C
Seq3	G G CUUCGG C C
Seq4	G A CUUUGG U C

- Die Korrelation zwei Positionen kann als **Mutual Information** berechnet werden:
- „Wenn Ihr mir sagt, die Identität der Position i, wie viel kann ich über die Identität der Position j lernen? “
- **Wir berechnen Mutual Information** um kovariante Positionen in einem MSA zu lokalisieren

Mutual Information



- Die Korrelation zwei Positionen kann als **Mutual Information** berechnet werden:
- „Wenn Ihr mir sagt, die Identität der Position i, wie viel kann ich über die Identität der Position j lernen? “
- Wir können an Basenfrequenzen in eine Alignment schauen
- Zuerst, für jede Spalte i der Alignment, die frequenz $f_i(x)$ für basis x berechnet wird
- Zweitens, sind die 16 Zusammenfrequenzen zweier Nukleotide, x in Spalte i und y in Spalte j berechnet werden
- Für jedes Paar von Spalten berechnen wir das Verhältnis

$$\frac{f_{i,j}(x,y)}{f_i(x)f_i(y)}$$

- Falls die Basen frequenzen zweier Spalten unabhängig voreinander sind, dann ist der Ratio gleich 1



- Um die **Mutual Information** von zwei Spalten i and j zu berechnen, wird der Logarithmus für alle Basenkombinationen summiert:

$$M_{ij} = \sum_{x,y} f_{ij}(x,y) \log_2 \frac{f_{ij}(x,y)}{f_i(x)f_j(y)}$$

- $f_i(x)$: Frequenz jedes Base x in $\{A,C,G,U\}$ in Spalte i des Alignments
- $f_j(y)$: Frequenz jedes Base y in $\{A,C,G,U\}$ in Spalte j des Alignments
- $f_{ij}(x,y)$: zusammengeführte Frequenz zweier nukleotide x in spalte i und y in Spalte j

Mutual Information



- Um die **Mutual Information** von zwei Spalten i and j zu berechnen, wird der Logarithmus für alle Basenkombinationen summiert:

$$M_{ij} = \sum_{x,y} f_{ij}(x,y) \log_2 \frac{f_{ij}(x,y)}{f_i(x)f_j(y)}$$

- Bei **perfekter Korrelation zweier Spalten**, liegt der **MI content bei seinem maximum bei 2!** Null wann zwei Spalte völlig unabhängig sind
- **Wenn eine der beiden Stellen völlig konserviert ist, ist die MI 0**, weil es keine Kovariation gibt!

Mutual Information - Beispiel



- Berechnen die **Mutual Information**:

$$M_{ij} = \sum_{x,y} f_{ij}(x,y) \log_2 \frac{f_{ij}(x,y)}{f_i(x)f_j(y)}$$

1	2	3	4	5	6
C	G	C	G	A	U
C	G	G	C	C	G
C	G	C	G	G	C
C	G	G	C	U	A

$$M_{1,2} = ?$$

$$M_{3,4} = ?$$

$$M_{5,6} = ?$$

MI in dynamische Programmierung Algorithmen



Wie können wir MI in dynamische Programmierung Algorithmen enthalten?
Zum Beispiel im Zuker Algorithmus ..

$$W(i, j) = \min \left\{ \begin{array}{l} W(i + 1, j) \\ W(i, j - 1) \\ V(i, j) - \boxed{MI(i, j)} \\ \min_{i < j < k} (W(i, k) + W(k + 1, j)) \end{array} \right.$$



- R. Dubin, S.Eddy, A.Krogh und G. Mitchinson, Biological sequence Analysis, Cambridge, 1998
- **M. Zuker** and P. Stiegler: Optimal computer folding of large RNA sequences using thermodynamics
Nature Biotechnology, Vol 22, Num 11, pages 1457-1458, 2004
- Rune Lyngso, Lecture Notes on RNA Secondary Structure Prediction, 2010