

Bioinformatik von RNA- und Proteinstrukturen

Inhaltsverzeichnis

1	Formale Sprachen	1
1.1	formale Grammatik G	1
1.2	Klassifikation von formalen Sprachen	1
1.3	Hidden Markov Model	3
2	Einleitung	5
2.1	RNA	5
2.2	R/DNA-Sekundärstruktur	6
2.3	Strukturabbildungen	7
3	Strukturvorhersage	10
3.1	Nussinov	10
3.2	Erweiterung des Nussinov-Algorithmus	11
3.3	Zuker-Algorithmus	11
3.3.1	Turner-Modell (Nearest-Neighbor-Modell)	11
3.3.2	Freie Energien	12
3.3.3	Der Zuker-Algorithmus (1981)	13
3.3.4	Backtracking von Zuker	16
3.4	Suboptimale Strukturen	17
3.4.1	Zuker-Suboptimals (1989)	17
3.4.2	Wuchty-Algorithmus	17
3.5	McCaskill	20
3.6	stochastisches Backtracking	25
4	Verbesserung von Strukturvorhersagen	27
4.1	Energieparameter	27
4.2	Dangling ends	27
4.3	Training der Energieparameter	27
4.4	Constraint Folding	28
5	Konsensusstrukturvorhersage - Komparative Analyse	29
5.1	RNAaliFold- zuerst alignen, dann falten	29
5.2	Sankoff-Algorithmus - gleichzeitiges Alignen und Falten	30
5.3	TREEforester - zuerst falten, dann alignen	32
5.4	Lokales Falten	33
6	RNA-RNA-Interaktionen (RNA-Interferenz)	33
6.1	RNA miteinander falten und konkatenieren	35
6.2	RNAplex	35
7	Neutrale Netzwerke von RNA-Strukturen (Peter Schuster)	37
7.1	Shape-Abstraktion (R. Giegerich)	37
7.2	Faltungskinetik mit Energielandschaften	37

7.2.1	Metropolis-Monte-Carlo	39
7.2.2	Barrier-Trees	39
7.2.3	Baumbau mit Flooding-Algorithmus	40
7.2.4	Direkte Pfade	40
7.2.5	Cotranskriptional Folding	41
8	weitere Bindungsarten, erlaubte Basenpaare	42
9	Proteine	43
10	Sekundärstrukturelemente	44
10.1	Chou-Fasman (Sekundärstrukturvorhersage von Proteinen) . .	44
11	(Protein-) Strukturvorhersage (3D)	47
11.1	Strukturaufklärung	47
11.2	Qualität der Strukturvorhersage	47
11.3	Problem der Strukturvorhersage (Levinthal-Paradoxon)	47
11.4	Protein-Domains (Domänen)	48
11.5	Zwei Typen von Vorhersagen	49
11.5.1	Ab-initio-Vorhersage	49
11.5.2	Template based methods	52

1 Formale Sprachen

Formale Sprache¹ L über Alphabet Σ

$L \subseteq \Sigma^*$

mit Σ^* = Kleensche Hülle² von Σ

$$\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$$

$\Sigma^0 = \{\varepsilon\}, \Sigma^1 = \Sigma, \Sigma^2 = \Sigma \times \Sigma$

$\varepsilon \rightarrow$ leeres Wort (leere Menge)

Beispiel: $\Sigma = \{a\}, \Sigma^* = \{\varepsilon, a, aa, aaa, \dots\}, L = \{a, aa, aaaa, \dots\}$

1.1 formale Grammatik G

$G = (N, \Sigma, P, S)$ mit

- N = Nichtterminale
- Σ = Alphabet
- P = Produktionsregeln
- S = Startsymbol ($\in N$)

$P \subseteq (N \cup \Sigma)^* / N(N \cup \Sigma)^* \rightarrow (N \cup \Sigma)^*$

Beispiel:

$G = (\{S\}, \{a\}, \{S \rightarrow aaS, S \rightarrow a\}, S)$

führt zu: $S \rightarrow aaS \rightarrow aaa$

1.2 Klassifikation von formalen Sprachen

durch die Chomsky-Hierarchie³:

- Typ 0 = rekursiv auszählbar ($\alpha N \beta \rightarrow \gamma$)
- Typ 1 = kontext-sensitiv ($\alpha N \beta \rightarrow \alpha \gamma \beta$)
- Typ 2 = kontext-frei, $N \rightarrow (N \cup \Sigma)^* \rightarrow$ stochistisch kontextfreie Grammatik (SCFG) \rightarrow Dynamics Programming

¹https://de.wikipedia.org/wiki/Formale_Sprache

²https://de.wikipedia.org/wiki/Kleenesche_und_positive_H%C3%BClle

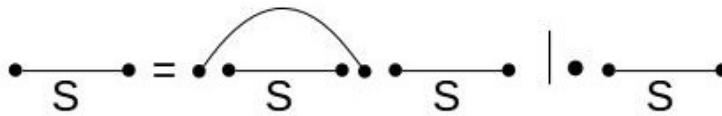
³<https://de.wikipedia.org/wiki/Chomsky-Hierarchie>

- Typ 3 = regular ($N \rightarrow \Sigma|\Sigma N$) \rightarrow dann immer Hidden Markov Model (HMM) modellierbar

bei Alignments: $\boxed{S} \longrightarrow \boxed{S} \begin{smallmatrix} \vdots \\ \vdots \end{smallmatrix} \mid \boxed{S} \begin{smallmatrix} - \\ - \end{smallmatrix} \mid \boxed{S} \begin{smallmatrix} \cdot \\ \cdot \end{smallmatrix} \mid \varepsilon$

Erweiterung mit Wahrscheinlichkeit: $G=(N, \Sigma, P, S, \Omega)$
mit Ω = Wahrscheinlichkeit für Produktionsregeln

jetzt auf RNA-Vorhersagen:

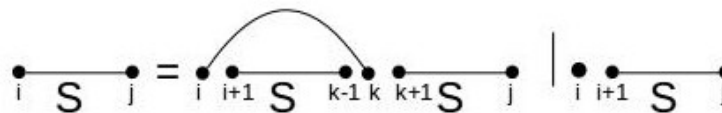


scoring scheme: Bewertung von $\sigma(\curvearrowright) = 1, (\sigma(\dashrightarrow)), \sigma(\cdot) = 0$
scoring function:

- max Basepairs: + (Summe),
- Anzahl der Strukturen: \cdot (Multiplikation)

choice function:

- max Basepairs: max,
- Anzahl der Strukturen: + (Summe)



$$S_{ij} = \begin{cases} S_{i+1,j} + \sigma(\cdot) \\ S_{i+1,k-1} + S_{k+1,j} + \sigma(\curvearrowright) \end{cases}$$

1.3 Hidden Markov Model



M: Match, I: Insertion, D: Deletion

Grammatik:

- $M \rightarrow M_{A_A} | \dots | I | D$
- $I \rightarrow I_{A_{--}} | \dots | D | M$
- $D \rightarrow D_{--A} | \dots | M | I$

Beispiel:



Faltungsgrammatik

$S \rightarrow (S)S|.S|\epsilon$

Nichtterminale = S, Alphabet = {(,), .}

Beispiel in Baumdarstellung:



weiteres Beispiel: Sankoff, Kombination von zwei Grammatiken (Alignment und Faltung)

Alignmentgrammatik

$$S \rightarrow .S|_S|\varepsilon$$

$$G = (N = \{S\}, \Sigma = \{., _\}, P = \{S \rightarrow .S|_S|\varepsilon\}, S)$$

$$\text{Alignment: } G^2 = G \times G = (N \times N, \Sigma \times \Sigma, P^2, (S, S))$$

$$P^2 = P \times P = \begin{pmatrix} S \\ S \end{pmatrix}$$

2 Einleitung

Struktur: Form \rightarrow Funktion

Funktion folgt Form, Form folgt Sequenz

Proteine, RNA, DNA: Sequenzen

4 Strukturlevels:

- primäre Struktur (Sequenz): 1 Dimension
- sekundäre Struktur (grobe Annäherung an Struktur): 2 Dimensionen
- tertiäre Struktur (räumliche Struktur): 3 Dimensionen
- quartäre Struktur (räumliche Anordnung von interagierenden Strukturen): 4 Dimensionen

Behandlung hauptsächlich 2D

2.1 RNA

⁴ Funktion:

- Informationsträger
- Regulator/Katalysator
- Theorie der RNA-World

- Nicht-Messenger-RNA: ncRNA (nc - non-coding)

- Aufbau: Zucker-Phosphat-Rückgrat
- Basen:
 - Purine: Adenin, Guanin
 - Pyrimidine: Cytosin, Uracil
- Paarung: A-U, G-C
- RNA einzelsträngige A-Helix (DNA: doppelsträngige B-Helix)

⁴<https://de.wikipedia.org/wiki/Ribonukleins%C3%A4ure>

2.2 R/DNA-Sekundärstruktur

Definition: Liste von Basenpaaren, sodass gilt (theoretische Regeln):

- erlaubte Basenpaarungen:
 - Watson-Crick: AU, UA, GC, CG
 - Wobble: GU, UG
- zwischen miteinander paarenden Basen müssen mindestens 3 Basen stehen $if(i, j) \in B \rightarrow i < j - 3$
 Beispiel Paarung A und U:

A	U	<u>A</u>	U	A	U	A	<u>U</u>
			1	2	3	4	
- keine Tripletts (Multipletts): eine Base paart maximal mit einer anderen $if(i, j); (i, k) \in B \rightarrow j = k$
- keine pseudo-Knoten: Basen kreuzen sich nicht $if(i, j); (k, l) \in B \rightarrow i < j < k < l$ und $i < k < l < j$

Motivation zu Regeln: jedes Basenpaar teilt das Molekül in 2 Teile (innen und außen), die miteinander nicht interagieren (vor allem Regel 3 + 4)

physikalische Eigenschaften:

1. Großteil des stabilisierenden Energie für RNA-Struktur kommt aus der Sekundärstruktur
2. Sekundärstruktur bildet sich zeitlich vor Tertiärstruktur aus

Experimenteller Nachweis 3D, 4D:

- Röntgenkristallographie: Kristall benötigt → oft schwierig
- nuclear magnet resonanz (nmr): stark konzentrierte Lösung benötigt, nur Distanzen zwischen Atomen ermittelbar

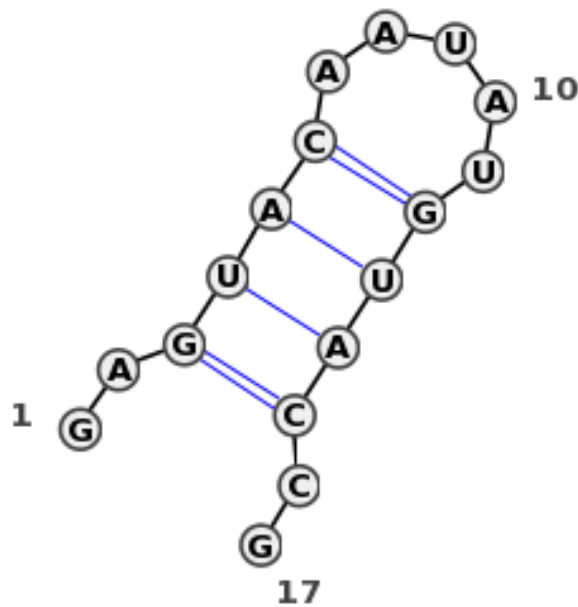
für 2D: Methoden, die bevorzugt einzelsträngige oder doppelsträngige Strukturen schneiden

2.3 Strukturabbildungen

1. Strukturplot:

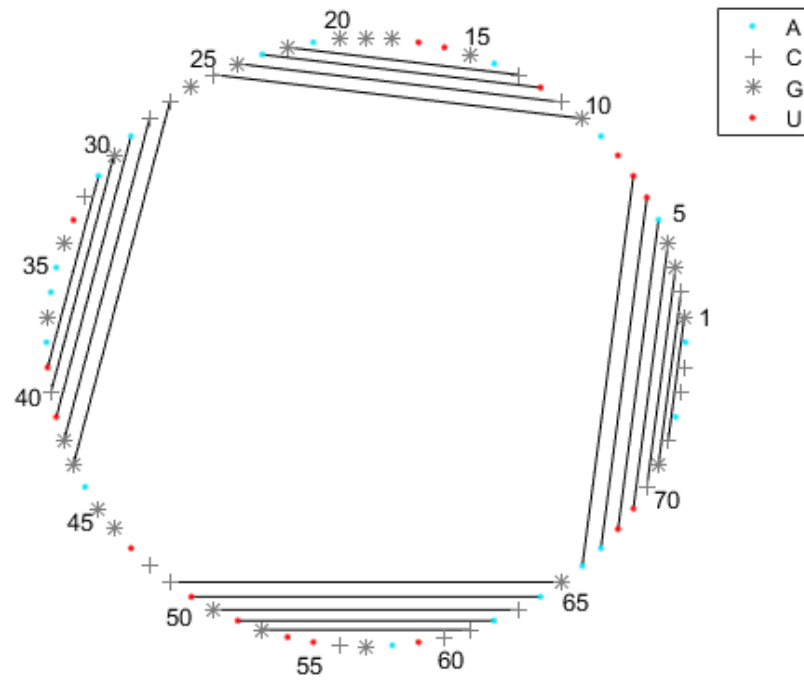


2. Dot-Bracket:

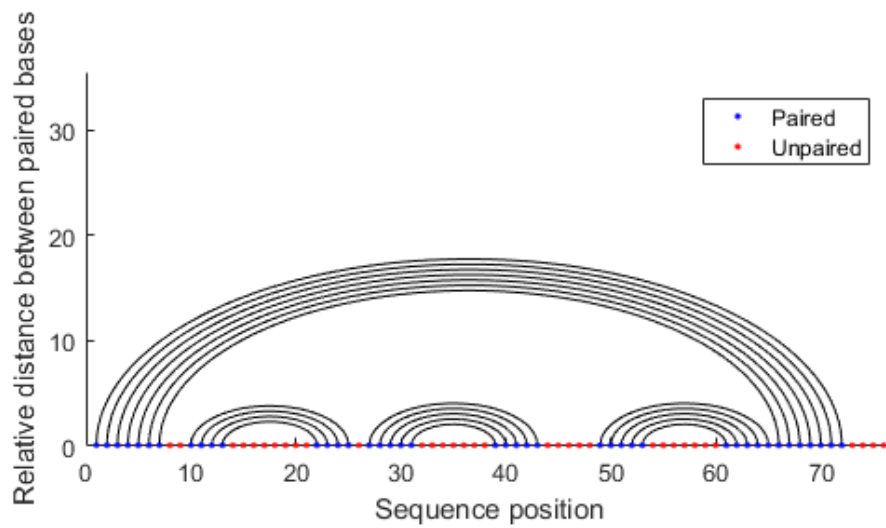


Seq:	GAGUACAAUAUGUACCG
Str:	..(((.....)))).

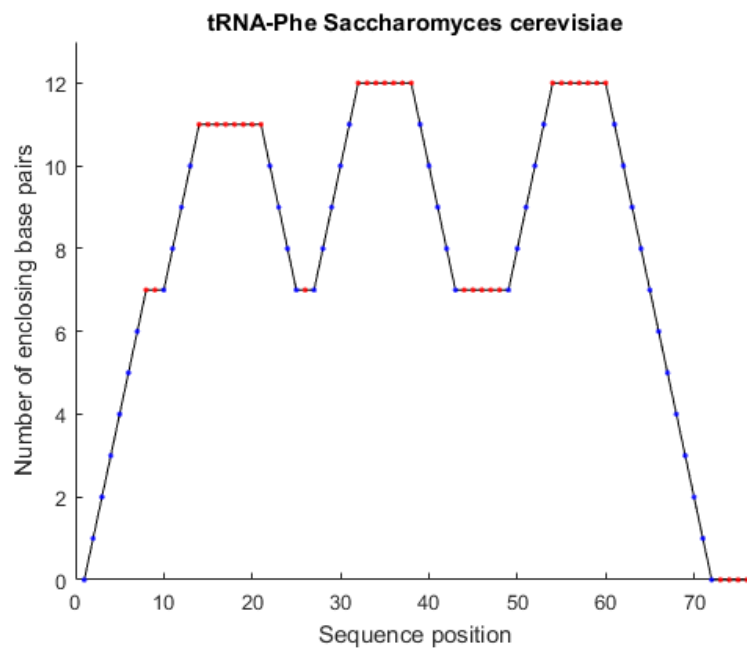
3. Zirkulärplot:



4. Bogenplot:



5. Mountainplot:



6. Dotplot:



3 Strukturvorhersage

- durch Aufteilung kann Dynamics Programming verwendet werden
- Beginn: einzelne Basen → keine Struktur

3.1 Nussinov

- von Ruth Nussinov (1978)
- Versuch Struktur mit der maximalen Anzahl der Basenpaare zu finden (Grundlage ist Sequenz)

Dynamic Programming

- **Initialisierung:**

- $N(i, i) = 0$
- $N(i, j) = 0$ if $i < j \leq i + 3$ (siehe Regel 2)
- $N(j + 1, j) = 0$

- **Brechung:**

$$N(i, j) = \max \begin{cases} N(i + 1, j) \text{ (ungepaart)} \\ \max_{i+3 < k \leq j} N(i + 1, k - 1) + N(k + 1, j) + F(i, k) \end{cases}$$

$$\text{mit } F(i, k) = \begin{cases} 1 \text{ if } i, k \in \{AU, GC, GU\} \\ -\infty \text{ else} \end{cases}$$

Basenpaarung mit i und k teilt Sequenz in inneren und äußeren Teil:



→ höchste Punktzahl wahrscheinlichste Sekundärstruktur

Ressourcenbedarf:

- Speicher: $O(n^2)$
- Prozessor: $O(n^3)$

3.2 Erweiterung des Nussinov-Algorithmus

Der ursprüngliche Nussinov-Algorithmus bestimmt die maximal mögliche Anzahl an Basenpaaren, die folgende Abwandlung bestimmt wie viele unterschiedliche Strukturen sind insgesamt möglich sind.

Initialisierung:

$$Z(i, j) = 1 \text{ if } i < j \leq i + 3$$

$$Z(j + 1, j) = 1$$

Berechnung:

$$Z(i, j) = + \begin{cases} Z(i + 1, j) & (i \text{ ungepaart}) \\ \sum_{i+3 < k \leq j} Z(i + 1, k - 1) * Z(k + 1, j) * 1(i \text{ gepaart}) \end{cases}$$

- Anzahl der Strukturen wächst exponentiell
- Ansatz zu einfach
- schlechte Vorhersage (in Wahrheit Energieabhängig)

Was stabilisiert RNA-Struktur? Interaktionen der aromatischen Ringe, die beim **Stapeln** auftreten

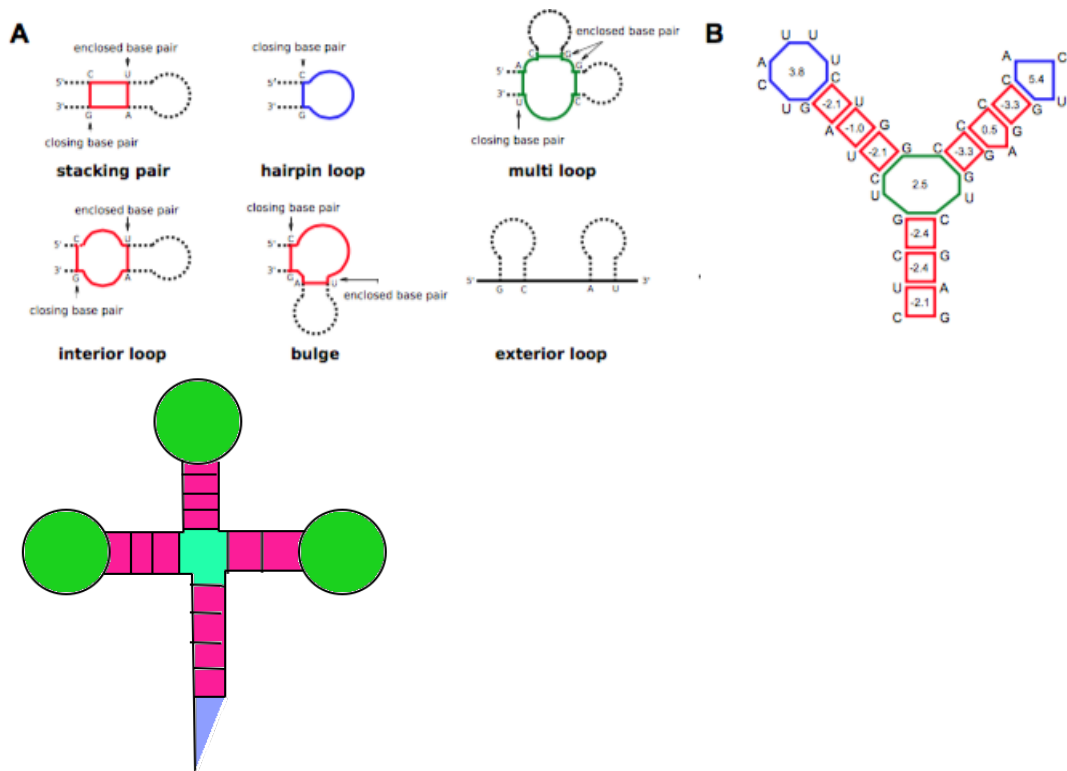
3.3 Zuker-Algorithmus








In der Praxis ist der Nussinov-Algorithmus unzureichend, da eine Maximierung von Basenpaaren nicht zwangsläufig mit einer Maximierung von Stabilität einhergeht. RNA's werden vor allem durch Interaktionen zwischen den Aromaten ihrer Basen stabilisiert. Das bedeutet, dass möglichst lange Abschnitte von gepaarten Basen besonders stabil sind. Der Nussinov-Algorithmus maximiert lediglich die Anzahl der Basenpaare, berücksichtigt jedoch nicht wie die Basenpaare im Molekül verteilt sind.

Der Zuker-Algorithmus berechnet die optimale Sekundärstruktur einer RNA-Sequenz mit der minimalen freien Energie indem er thermodynamische Daten verwendet. Er geht davon aus, dass nicht die Struktur mit der maximalen Anzahl an Basenpaaren die stabilste Struktur darstellt, sondern jene mit der geringsten freien Energie. Dazu wird das RNA-Molekül zunächst wie folgt nach dem Turner-Modell in einzelne Schleifen zerlegt.

3.3.1 Turner-Modell (Nearest-Neighbor-Modell)

Das Turner-Modell beschreibt die verschiedenen Untereinheiten/Schleifen eines RNA-Moleküls wie folgt:



- 0 Basenpaare: exterior loop 
- 1 Basenpaar: hair-pin loop 
- 2 Basenpaare: interior loop 
- 0 Basen ungepaart: stack 
- auf einer Seite eine Base ungepaart: buldge 
- auf beiden Seiten eine Base ungepaart 
- >2 Basenpaare: multi-loop 

3.3.2 Freie Energien

Damit der Zuker-Algorithmus freie Energie minimieren kann, müssen für die verschiedenen Basenkombinationen in den verschiedenen Loop-Arten Energiewerte gegeben sein. Diese wurden in aufwendigen Schmelztemperaturmessungen experimentell bestimmt. Es gibt Werte für alle möglichen Stack-Arten und für alle Buldges und interior-Loops bis zu 2,3 Basenpaaren.

Die Energie eines Interior-Loops ist abhängig von:

- den schließenden Basenpaaren
- den Basen, die den schließenden Basenpaaren benachbart sind
- der Anzahl der ungepaarten Basen
- der Assymetrie

Die Energie eines Hairpin-Loops ist abhängig von:

- Typ der schließenden Basenpaaren
- den Basen, die den schließenden Basenpaaren direkt benachbart sind
- der Anzahl der ungepaarten Basen im Loop

Die Energie eines Multi-Loops ist abhängig von:

- Anzahl der schließenden Basenpaaren

3.3.3 Der Zuker-Algorithmus (1981)

- verwendet das Turner-Modell
- minimiert freie Energie: **minimum free energy** (mfe)
- Freie Energien sind additiv. Die Energie einer RNA-Struktur ist die Summe der freien Energien der einzelnen Schleifen.
- Festlegung: freie Energie der offenen Kette = 0

Initialisierung:

$$F(i, j) = 0 \text{ if } i < j \leq i + 3$$

$$F(j + 1, j) = 0$$

Berechnung:

$$F(i, j) = \min \begin{cases} F(i+1, j) \\ \min_{i < k \leq j} \{C(i, k) + F(k+1, j)\} \end{cases}$$

$$C(i, j) = \min \begin{cases} \mathcal{H}(i, j) \\ \min_{i < k < l < j} \{ \mathcal{I}(i, j, k, l) + C(k, l) \} \\ \min_{i < u < j} \{ a + M(i+1, u) + M^1(u+1, j-1) \} \end{cases}$$

$$M(i, j) = \min \begin{cases} M(i, j-1) + c \\ \min_{i < k < j} \{ (k-i) \cdot c + b + C(k, j) \} \\ \min_{i < k < j} \{ M(i, k-1) + b + C(k, j) \} \end{cases}$$

$$M^1(i, j) = \min \begin{cases} M^1(i, j-1) + c \\ b + C(i, j) \end{cases}$$

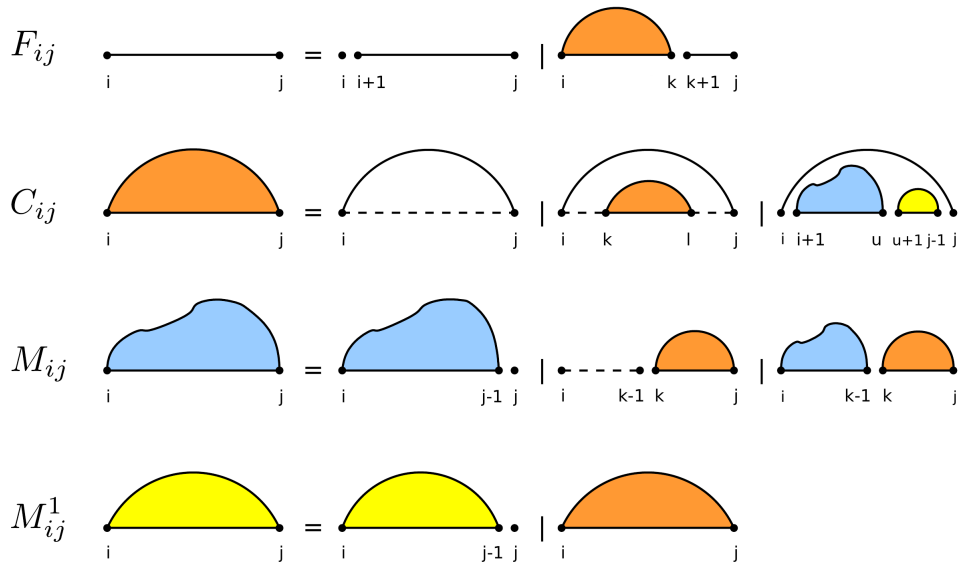
\mathcal{H} und \mathcal{I} sind gegebene Energietabellen für Hairpin und Interior-Loops

a - Strafe für Schließen eines Multi-Loops

b - Strafe für inneres Basenpaar in einem Multi-Loop

c - Strafe für ungepaarte Base in einem Multi-Loop

Grafische Darstellung des Zuker-Algorithmus



Komplexität

Speicher: $\mathcal{O}(n^2)$

Prozessor: $\mathcal{O}(n^4)$

Um die Laufzeit einzuschränken wird die Anzahl der ungepaarten Basen im Interior-Loop $j - l + k - i < \text{Schwellenwert} \approx 30$.

3.3.4 Backtracking von Zuker

Stapel unaufgeklärter Strukturen: i, j und Typ

push $F(1, n)$ on stack

Wiederhole, solange Stack nicht leer ist:

pop

if $F(i, j) = F(i + 1, j)$:

→ **push** $F(i + 1, j)$ on stack

else:

Suche k , mit $i < k \leq j$, sodass $F(i, j) = C(i, k) + F(k + 1, j)$:

→ **push** $C(i, k)$ and $F(k + 1, j)$ on stack

if $C(i, j) = \mathcal{H}(i, j)$:

→ i, j als Basenpaar markieren

else if $\exists k, l$, mit $i < k < l < j$, sodass $C(i, j) = \mathcal{I}(i, j, k, l) + C(k, l)$:

→ i, j als Basenpaar markieren

→ **push** $C(k, l)$ on stack

else:

Suche u , mit $i < u < j$, sodass $C(i, j) = a + M(i + 1, u) + M^1(u + 1, j - 1)$

→ i, j als Basenpaar markieren

→ **push** $M(i + 1, u)$ and $M^1(u + 1, j - 1)$ on stack

if $M(i, j) = M(i, j - 1) + c$:

→ **push** $M(i, j - 1)$ on stack

else if $\exists k$, mit $i < k < j$, sodass $(k - i) \cdot c + b + C(k, j)$:

→ **push** $C(k, j)$ on stack

else:

Suche k , mit $i < k < j$, sodass $M(i, j) = M(i, k - 1) + b + C(k, j)$:

→ **push** $M(i, k - 1)$ and $C(k, j)$ on stack

if $M^1(i, j) = M^1(i, j - 1) + c$:

→ **push** $M^1(i, j - 1)$ on stack

else:

→ **push** $C(i, j)$ on stack

3.4 Suboptimale Strukturen

3.4.1 Zuker-Suboptimals (1989)

Berechnet suboptimale Strukturen, indem für jedes *mögliche* Basenpaar die beste Struktur, insgesamt quadratisch viele Strukturen, berechnet wird. Dies wird durch ein Verdoppeln der Input-Sequenz erreicht. Die beiden Sequenzen werden dann wie folgt aneinander gehängt:

$$1, \dots, n, n+1, \dots, 2n$$

$$\text{Optimale Energie für } BP(i, j) = C(i, j) + C(j, n+i)$$

Wird der Zuker-Algorithmus auf diese verdoppelte Sequenz durchgeführt erhält man Ergebnisse... **Wer kann's erklären?**

3.4.2 Wuchty-Algorithmus

1. **Forward-Algorithmus** kann Nussinov- oder Zuker-Algorithmus sein
2. **Backtracking:**
Stapel enthält mehr Information als bei Zuker: i, j , Typ, Struktur und $\Delta\epsilon$ (Energie, die noch verfügbar ist)

Wuchty-Backtrack von Nussinov-Matrizen Beim Wuchty-Backtracking von Nussinov-Matrizen werden, im Gegensatz zum "normalen" Backtrack, *Refinements* aufgehoben und weiterverfolgt, die eine höhere Anzahl von Basenpaaren haben als vorher über einen Schwellenwert als Minimum festgelegt.

$$P_{S'} \geq P_{\max} - \Delta_{\text{Schwellenwert}} \quad (1)$$

Erfüllt nun das Refinement S mit seiner maximalen Anzahl Basenpaare P (2) das Kriterium für gewünschte Suboptimale Strukturen (1), wird S' für weitere *Refinements* auf den übergeordneten Stack R geschoben.

$$P_{S'} = |\mathcal{P}| + 1 + P_{i+1, k-1} + P_{k+1, j} + \sum_{[a, b] \in \sigma} P_{a, b} \quad (2)$$

Refinement Beim optimalen Nussinov Backtracking wird ein Tupel $(S, \text{bedeutet partielle Struktur})$ aus einem Stack mit unaufgeklärten Sequenzintervallen (σ) und eine Menge von Basenpaaren (\mathcal{P}) , $S = (\sigma, \mathcal{P})$ bearbeitet. Am Ende des Backtrackings ist σ leer und \mathcal{P} möglichst groß. *Refinement* bezeichnet nun den Schritt, ein Sequenzintervall $[i, j]$ vom Stack σ zu nehmen, in ein kleineres $([i+1, j])$, bzw. zwei kleinere Intervalle $([i+1, k-1], [k+1, j])$ aufzuteilen, diese jeweils wieder auf den Stack zu schieben, \mathcal{P} gegebenenfalls um ein Basenpaar zu erweitern $(\mathcal{P} \cup \{i \cdot k\})$ und damit zusammengefasst S zu S' zu "refinieren".

Wuchty-Backtrack von MFE-Matrizen Die partielle Struktur \mathcal{S} wird hier mit der totalen freien Energie aller Loops (E_{LS}) aus der partiellen Struktur erweitert zu $\mathcal{S} = (\sigma, \mathcal{P}, E_{LS})$. Außerdem wird der Stack σ erweitert um den Typ der Matrix (F, C, M, M^1), in der weitergearbeitet werden soll (z.B. $\sigma = \{[i, j]_M\}$).

$$E_{\mathcal{S}'} \leq E_{min} + \delta \quad (3)$$

Wenn zum Beispiel (Zucker, Fall 1) $F_{i+1,j} + E_{LS} + \sum_{[k,l] \in \sigma} E_{[k,l]} \leq E_{min} + \delta$ gilt, also das vorher gewählte Kriterium für suboptimale Faltungen (3) erfüllt ist, wird $\mathcal{S}' = (\{[i+1, j]_F\} \cup \sigma; \mathcal{P}; E_{LS})$ auf den Stack R geschoben.

Thermodynamik von Molekülen

viele Zustände möglich

Die Boltzmann-Statistik besagt, dass die Wahrscheinlichkeit p einen Zustand der Energie E mit einem Teilchen besetzt zu finden, proportional ist zum Boltzmann-Faktor:

- *Boltzmannfaktor* $= e^{-\frac{E}{k_B T}}$, mit Energie E , Boltzmannkonstante k_B und der absoluten Temperatur T
- $p(E) \propto e^{-\frac{E}{k_B T}}$, $\beta = \frac{1}{k_B T}$
- für eine Struktur S und eine Energie $E(S)$ gilt:
- $p(S) \propto e^{-\beta E(S)}$
- $\sum p(S) = 1 \rightarrow$ Summe aller Wahrscheinlichkeiten aller Strukturen
-

$$p(S) = \frac{e^{-\beta E(S)}}{\sum_{s'=s1}^{sn} e^{-\beta E(S)}} \quad (4)$$

wobei

$$\sum_{s'=s1}^{sn} e^{-\beta E(S)} = Z \quad (5)$$

Z=Zustandssumme (partition function)

- D.h., dass eine Wahrscheinlichkeit für jede Eigenschaft x (z.B. spezifisches Basenpaar) meines Moleküls wie folgt berechnet werden kann:

$$p(x) = \frac{\sum_x e^{-\beta E(x)}}{Z} \quad (6)$$

wobei die Summierung über alle Strukturen läuft, die diese Eigenschaft x besitzen

Pseudocode Wuchty-Algorithmus

- gesucht: Strukturen mit mind. x Basenpaaren
- $x = \max - \Delta$
z.B. $\Delta = 1 \rightarrow x = 3$... wir suchen alle Strukturen mit mind. 3 Basenpaaren
- als Datenstruktur wird ein Stack verwendet
 - push (puts element on the stack)
 - pop (takes top element from the stack)
- Jeder Eintrag D auf dem Stack enthält:
 - I - Intervall
 - BP - Anzahl an Basenpaaren
 - S - Struktur

1. **Forward-Algorithmus (hier Nussinov) durchführen** \rightarrow outputs $N(i, j)$

2. **Backtrack:**

```
push([1, n], 0, 0)
while (stack not empty)
    • Nimm Intervall  $a$  aus  $I$ 
    • Berechne die Anzahl an möglichen Basenpaaren in den anderen Intervallen
    •  $Z = BP + \sum_{b \neq a \in I} N[b] \rightarrow$  in  $a$  mind.  $x - Z$  Basenpaare
    • if( $N(i, j - 1) \geq x - Z$ )
        –  $a' = [i, j - 1]$ 
        – push ( $[a', b], BP, S$ )
    • else output
    • for( $i \leq k < j - 3$ )
        – if( $N(k + 1, j - 1) + N(i, k - 1) + 1 \geq x - Z$ )
             $c = k + 1, j - 1$ 
             $d = (i, k - 1)$ 
             $BP' = BP + 1$ 
             $S' = S \cup k, j$ 
            push( $c, d, BP', S'$ )
        – else output
```

Das in der Vorlesung berechnete Beispiel wird hier nicht aufgeführt.

3.5 McCaskill

Summe über alle Strukturen

$$Z = \sum_S e^{-\beta \cdot E_s} \quad (7)$$

mit $\beta = \frac{1}{R \cdot T}$, E_s = Energie der Struktur S (alle Strukturen),
 R = Gaskonstante = $k \cdot N_A$, T = Temperatur, k = Boltzmann-Konstante, N_A = Avogadro-Konstante

Dynamic Programming Ansatz:

- Energie von einer Base = 1 $\rightarrow e^0 = 1$

$$Z(i, j) = \underbrace{Z(i+1, j)}_{\text{ungepaart}} + \underbrace{\sum_{i < k \leq j} Z^B(i, k) \cdot Z(k+1, j)}_{\text{gepaart}} \quad (8)$$

mit Z^B = Basenpaar zwischen i, k

$$Z^B(i, j) = \underbrace{\mathcal{H}(i, j)}_{\text{Hairpin}} + \underbrace{\sum_{i < k < l < j} \mathcal{I}(i, j, k, l) \cdot Z^B(k, l)}_{\text{Interior loop}} + \underbrace{\sum_{i < k < j} e^{-\beta \cdot a(i, j)} \cdot Z^M(i+1, u) \cdot Z^{M1}(u+1, j-1)}_{\text{Multiloop}} \quad (9)$$

$$\mathcal{H} = e^{-\beta \cdot \text{Hairpin}(i, j)}$$

$$\mathcal{I} = e^{-\beta \cdot \text{Interior}(i, j)}$$

mit

a = Energie fürs Schließen des Multiloops

Hairpin = Energie für Hairpin

Interior = Energie für Interior loop

Initialisierung:

$$Z^M = 0; Z^{M1} = 0; Z^B = 0$$

$Z^M(i, j)$ = ungepaart + genau 1 Basenpaar + Erweiterung um 1 Basenpaar, also mindestens 2

$$Z^M(i, j) = Z^M(i, j-1) \cdot e^{-\beta \cdot c} + \sum_{i < k < j} e^{-\beta \cdot ((k-i) \cdot c)} \cdot e^{-\beta \cdot b} \cdot Z^B(k, j) + \sum_{i < k < j} Z^M(i, k-1) \cdot e^{-\beta \cdot b} \cdot Z^B(k, j) \quad (10)$$

mit

b = Energie für Basenpaar innerhalb eines Multiloops

c = Energie für ungepaarte Base innerhalb eines Multiloops

→ da Z^M mit 0 initialisiert, kann zunächst Z^B als einzelnes Basenpaar "aufgeschrieben" werden, dann erst Erweiterung möglich

$$Z^{M1}(i, j) = Z^{M1}(i, j - 1) \cdot e^{-\beta \cdot c} + e^{-\beta \cdot b} \cdot Z^B(i, j)$$

Ergebnis= 4 Matrizen: $Z, Z^B, Z^M, Z^{M1} \rightarrow$ Backtracking

Wahrscheinlichkeit für Basenpaare:

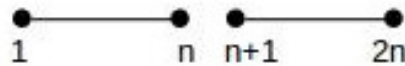
$$p(i, j) = \frac{Z^{(i,j)}}{Z} \quad (11)$$

mit $Z^{(i,j)}$ gegeben i, j

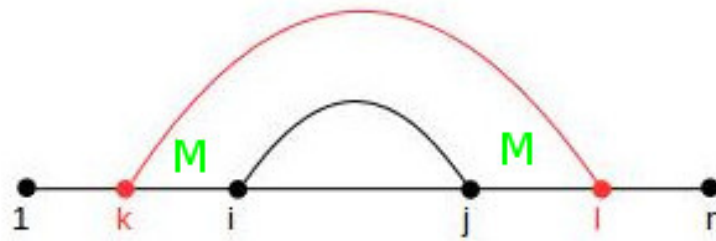
$$Z^{(i,j)} = \underbrace{Z^B(i, j)}_{\text{innen, schon bekannt}} \cdot \underbrace{\widehat{Z}^B(i, j)}_{\text{außen, noch unbekannt}} \quad (12)$$

mit \widehat{Z}^B : wie beim suboptimalen Falten (Zucker suboptimals) → Verdoppelung der Sequenzen

$$\widehat{Z}^B(i, j) = Z^B(j, n + i) \quad (13)$$



teuer!



Achtung: Gleichung über mehrere Zeilen!

$\widehat{Z}^B(i, j) = Z(1, i-1) \cdot Z(j+1, n) +$ (i, j) nicht von einem anderen Basenpaar eingeschlossen

$\sum_{k < i < j < l} \mathcal{I}(k, l, i, j) \cdot \widehat{Z}^B(k, l) +$ (i, j) von einem externen Basenpaar (k, l) eingeschlossen. Zusammen bilden sie einen Interior loop (kann kein Hairpin sein, da Basenpaar (i, j) innen vorhanden).

$\sum_{k < i < j < l} \widehat{Z}^B(k, l) \cdot e^{-\beta \cdot a} \cdot e^{-\beta \cdot b} \cdot \{$ (k, l) schließt Multiloop ein. Strafenergie für Basenpaarung in Multiloops b kann gleich hinzugefügt werden, da wir ja sicher (i, j) paaren.

$Z^M(k+1, i-1) \cdot Z^M(j+1, l-1) +$ (i, j) schließt einen mittleren Multiloop-Teil (linke und rechte Multiloop-Dekompositionen zwischen $(k+1, i-1)$ und $(j+1, l-1)$).

$Z^M(k+1, i-1) \cdot e^{-\beta \cdot ((l-j-1) \cdot c)} +$ (i, j) schließt den linken Multiloop-Teil (Multiloop-Dekomposition zwischen $k+1$ und $i-1$).

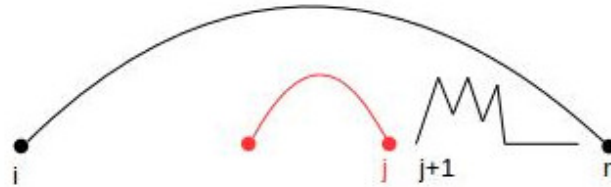
$Z^M(j+1, l-1) \cdot e^{-\beta \cdot ((j-k-1) \cdot c)} \}$ (i, j) schließt den rechten Multiloop-Teil (Multiloop-Dekomposition zwischen $j+1$ und $l-1$).

Berechnung dauert $O(n^4)$ da $k < i < j < l \rightarrow$ sehr teuer

\Rightarrow Verbesserung durch Multiloop-Berechnung

Definition:

$$Z^A(i, j) = \sum_{j \leq k \leq n (=Ende)} \widehat{Z}^B(i, k) \cdot Z^M(j, k-1)$$



Ermittelt alle Zustandssummen, bei denen i mit n Multiloop schließt und innerhalb ein weiterer Multiloop existiert

$$Z^{A'}(i, j) = \sum_{j \leq k \leq n} \widehat{Z}^B(i, k) \cdot e^{-\beta \cdot (k-j)}$$

wie oben aber ohne inneren Multiloop

⇒ **damit:**

$$\begin{aligned} \widehat{Z}^B(i, j) = & \dots + \sum_{1 \leq k < i} Z^A(k, j+1) \cdot (Z^M(k+1, i-1)) + \text{rechts und links Multiloop (gepaart)} \\ & \sum_{1 \leq k < i} Z^A(k, j+1) \cdot e^{-\beta(k-i-1) \cdot c} + \text{nur rechts Multiloop (ungepaart)} \\ & \sum_{1 \leq k < i} Z^{A'}(k, j+1) \cdot (Z^M(k+1, i-1)) \text{ nur links Multiloop} \end{aligned}$$

⇒ damit nur noch $O(n^3)$

⇒ Anwendung der Berechneten Daten in Dotplot

- Fläche der Zellen repräsentiert Wahrscheinlichkeit für jedes Basenpaar
- Quadrat in Dotplot mit $\sqrt{p(i, j)}$ Seitenlänge
- in unterer Dreiecksmatrix mfe-Struktur (minimal free energy aus Zuker)

Weitere Visualisierungsmöglichkeiten:

- Centroid: die Struktur, die von allen am wenigsten entfernt ist; alle Basenpaare, die mindestens $p \geq 0,5$ haben
- MEA-Struktur (maximum expected accuracy): z.B. Nussinov mit scores proportional zu $p(i, j)$

- Strukturplot mit Färbung proportional zur Basenpaarwahrscheinlichkeit oder Nicht-Basenpaare; $p^n(i) = (1 - \sum_{i \neq j} p(i, j))$; p^n =ungepaart
- Positional entropy: $S(i) = \sum_{i \neq j} p(i, j) \ln(p(i, j))$
- in Mountainplot: Z als Score der beschreibt, in 5' oder 3' Richtung gepaart zu sein: $Z(i) = \sum_{i < j} p(i, j) - \sum_{j < i} p(j, i)$

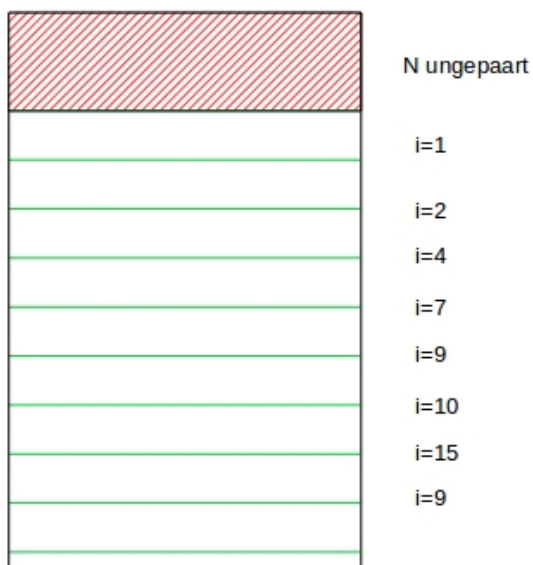
3.6 stochastisches Backtracking

Strukturen mit Wahrscheinlichkeit, mit der sie im Ensemble vorhanden sind
 → millionenfache Durchläufe

Ergebnis: Set von Strukturen, die das Ensemble abbilden - genaue Berechnung
 (siehe Punkt vorher) oft nicht möglich da zu teuer, daher stochastischer Ansatz

1. Forward Recursion der Zustandssummen Z, Z^B, Z^M, Z^{M1}
2. Backtracking

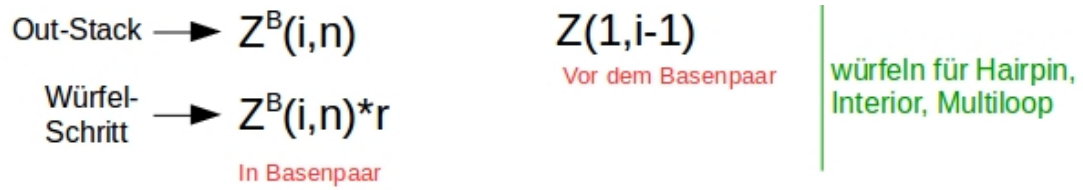
$$Z(1, n) = \underbrace{Z(1, n-1)}_{\text{nungepaart}} + \underbrace{\sum_{1 \leq i < n} Z^B(i, n) \cdot Z(1, i-1)}_{\text{ngepaart}(=xi)} \quad (14)$$



nicht alle i da nicht immer Paarungen möglich

- stochastisch: Zufallszahl zwischen $[0,1] = r$
- Grenze = $Z(1, n) \cdot r$
- aufsummieren bis $\sum_{0 \leq i < n} x_i > Z \cdot r$ mit $i =$ Bindungspartner

Backtracking von:



$$\begin{aligned}
 Z^B(i, j) = & \underbrace{\mathcal{H}(i, j)}_{\text{Hairpin}} + \underbrace{\sum_{i < k < l < j} \overbrace{\mathcal{I}(i, j, k, l)}^{\text{Energie}} \cdot \overbrace{Z^B(k, l)}^{\text{aufStack}}}_{\text{Interior}} + \underbrace{\sum_{i < u < j} \overbrace{Z^M(i+1, u)}^{\text{aufStack}} \cdot \overbrace{Z^{M1}(u+1, j-1)}^{\text{aufStack}}}_{\text{Multiloop}} \\
 & (15)
 \end{aligned}$$

- schnell für einzelne Strukturen (linear)
- bietet die Möglichkeit Strukturen auf einzelne Eigenschaften zu testen
- findet suboptimale Strukturen (wie Zuker und Wuchty)

4 Verbesserung von Strukturvorhersagen

Im folgenden sind mehrere Faktoren aufgeführt, die zur Optimierung der Strukturvorhersage bzw. die Vollständigkeit der Strukturvorhersage zeigen. Weiterhin kommt die komparative Analyse von RNA-Strukturen mit hinein, die aufgrund der evolutionär wichtigen Struktur-Funktions-Beziehung wichtige Aussagen über Strukturen geben können.

4.1 Energieparameter

Das Turner-Modell führte zur Erzeugung von fast 100 Parametern, die zur Ermittlung der thermodynamischen Energie eines RNA-Moleküls relevant ist zu bestimmen.

Andronescu et al. (2007)⁵ entwickelten eine "Constraint Generation"-Methode, bei der die Schätzung der Energieparameter auf Grundlage von bekannten thermodynamischen Datenbanken errechnet werden.

Hierbei werden zunächst die Energiewerte zur Berechnung der besten Struktur genutzt und durch ein Optimierungsproblem über mehrere Iterationsschritte verbessert, so dass die Energie minimal wird.

4.2 Dangling ends

- die ungepaarten Basen am Ende eines Stacks, werden als Dangling Ends bezeichnet.
- aufgrund ihre Instabilität als Einzelsträngige Teilsequenz sind diese Bereiche recht flexibel und können auf benachbarte Basen(-paare) umspringen um einen stabileren Zustand zu erreichen.
- 3'-Dangling Ends sind stabiler als 5'-Dangling Ends
- Dangling Ends werden als Ursache für coaxial Stacking diskutiert (Verdrehen und Rotation bestimmter Stacks zu anderen in einer Struktur)
- in der Berechnung werden diese Dangling Ends ebenfalls mit einem Strafbetrag verrechnet (Dangling-End-Parameter im Turner-Parameter-Modell)

4.3 Training der Energieparameter

Andronescu:

- Set von bekannten Strukturen aus Datenbank

⁵Andronescu,M; Condon,A; Hoos, H; Mathews, D; Murphy, KP(2007): Efficient parameter estimation for RNA secondary structure prediction. Bioinformatics. 2007;23(13):19-28.

- Turner-Parameter (thermodynamische Parameter)
- Strukturvorhersage: perturbieren (verändern, stören) der Parameter, versuche den Abstand zwischen vorhergesagten und richtigen Strukturen zu minimieren
- update Parameter

Problem: Training nur auf bekannten Daten, für unbekannte Strukturen nicht immer besser

4.4 Constraint Folding

Man unterscheidet zwei Typen von Constraints (Beschränkungen):

Hard Constraints Es darf keine Basenpaarung einer anderen widersprechen. Entweder eine Base i ist ungepaart oder ist nach links oder nach rechts mit einer anderen verbunden.

Soft Constraints Ergeben sich aus den Energiescorematrizen der Strukturvorhersage mit McCaskill (oder Zuker) und zeigen die Wahrscheinlichkeit einer bestimmten positionsabhängigen Basenpaarung an

Experimentell kann durch Methoden, wie PARS (Hochdurchsatz-Sequenzierung) bestimmt werden an welchen Positionen Basenpaarung vorliegen. Hierfür wird ein bioenzymatischer Verdau durchgeführt bei dem entweder doppelsträngige Stellen oder einzelsträngige Stellen zersetzt werden.

RNA → Struktur (Funktion) → evolutionär konserviert (Verwandschaft)

- es ist möglich Struktur und damit Funktion zu konservieren auch wenn die Sequenz verändert wird

- konsistente Mutation: $G-C \rightarrow G-U$
- kompensatorische Mutation: Mutation an C zieht Mutation an G mit sich

⇒ Ziel: Zusätzliche Informationsebene durch auffinden von konsistenten und Kompensatorischen Mutationen

⇒ benötigt Set von n Sequenzen, die die gleiche Struktur haben

nun drei Möglichkeiten: → prinzipiell: Alignments mit Sekundärstrukturvorhersage kombinieren

1. zuerst Strukturvorhersage, dann Aligement von Strukturen (Tree Alignment)
2. simultan alignen und Strukturen vorhersagen (Sankoff ($O(n^6)$))
3. zuerst alignen und dann Strukturen der Alignments vorhersagen (RNA Alifold)

5 Konsensusstrukturvorhersage - Komparative Analyse

In der Evolutionsbiologie geht man von einer Struktur-Funktions-Beziehung konservierter Merkmale aus. Das bedeutet: Funktional wichtige Sequenzbereiche sind in der DNA konserviert.

Für die RNA-Strukturvorhersage ist jedoch zu beachten, dass auch sich unterscheidende Sequenzen strukturell ähneln können und somit deren Sekundärstruktur evolutionär konserviert ist.

Betrachtet man nun einen Datensatz von RNA-Sequenzen, kann über Alignment der Sequenzen und/oder der Sekundärstrukturen die Ähnlichkeit dieser Sequenzen betrachtet werden. Es ist möglich, dass RNA-Moleküle mit ähnlicher bis gleicher Struktur miteinander verwandt sind.

→ komparative Analyse: Es gibt drei Hauptmethoden, die zur Konsensusstrukturvorhersage genutzt werden:

- Funktionsvorhersage
- Zuordnung von RNA-Klassen
- Motivsuche

5.1 RNAaliFold- zuerst alignen, dann falten

→ multiples Sequenzalignment (z.B. Needleman-Wunsch)

→ generalisierte Bestimmung einer RNA-Struktur

- minimiere die mittlere Energie

Für RNAaliFold ist ein multiples Alignment mit K Sequenzen der Länge m gegeben. Ziel ist es, eine RNA-Struktur der Konsensussequenz zu finden, deren minimierte Energie sich aus der Summe aller freien Energien der K Sequenzen und dem Konservierungsgrad gleichbleibender Sequenzen zusammensetzt.

- Unterscheidung von Mutationen

konsistente Mutationen (GC → GU)

und kompensatorische Mutationen (GC → UA)

Berechnung:

$$C(i, j) = \max \begin{cases} \mathcal{H}(i, j) \\ \text{Interior loop} \\ \text{Multiloop} \end{cases} \quad (16)$$

→ Rechne den Term, der die Konservierung beschreibt zu $C(i,j)$ dazu
 → Bestrafung von Verletzung der Komplementarität

$$\gamma(i, j) = \sum_{s_1, s_2 \in A} \begin{cases} h(s_{1i}, s_{2i}) + h(s_{1j}, s_{2j}) & \text{if } s_{1i}, s_{1j} \in Bp \text{ and } s_{2i}, s_{2j} \in Bp \\ 0 & \text{else} \end{cases} \quad (17)$$

$$\delta(i, j) = \sum_s \begin{cases} 0 & \text{if } (s_i, s_j) \in Bp \\ 0,5 & \text{if } s_i, s_j \in \{-\} \\ 1 & \text{else} \end{cases} \quad (18)$$

$$C(i, j) = x \cdot \gamma(i, j) + y \cdot \delta(i, j) + \text{mean}[C(i, j)] \quad (19)$$

A: Multiples Sequenzalignment (MSA)

s_1, s_2 : Sequenzen

$h(i, j)$: Hamming-Distanz zwischen i und j

x, y : Skalierungsfaktoren

$-$: Gap im Alignment

Es können so auf Wissen basierte Matrizen mit bekannten konservierten RNA-Strukturen erzeugt werden → Ribosum: knowledge-based Score, für die Wahrscheinlichkeit in einem Basenpaar (i,j) für die Basen s_{1i}, s_{1j} und s_{2i}, s_{2j}

→ Komplexität: $\Omega(n^3m)$

Problem: Es werden nur Konsensussequenzen gefalten. Somit ist nur Allgemein eine Faltung für alle Sequenzen vorliegend.

Somit nur sinnvoll, wenn zum einen ein Alignment möglich ist und wenn die Sequenzen sehr ähnlich zueinander sind.

5.2 Sankoff-Algorithmus - gleichzeitiges Alignen und Falten

Programm: locarna

Annahme: Zwei Sequenzen sind sich ähnlich, wenn ihre RNA-Struktur einen stark äquivalenten Shape besitzen ($\text{len}(A) = \text{len}(B)$ und $\text{pair}(A) = \text{pair}(B)$).

Idee: Finde für äquivalente Strukturen die minimale Editierdistanz bzw. MFE

Vorgehen: Sankoff = Zuker + Needleman-Wunsch

- 1 Gegeben sind zwei Sequenzen A und B
- 2 Finde äquivalente Strukturen in Sequenzen

- 3 Erzeuge ein zu Strukturen kompatibles Alignment beider Sequenzen
- 4 Editiere Alignment und Faltungen um minimalen Score $\min(E_A + E_B + \text{Distanz}_A B)$ zu finden
- 5 Consensussequenz und Sequenz C gegeben \rightarrow gehe zu 2

Distanzbestimmung (Needleman-Wunsch):

$$A(i, j; k, l) = \min \begin{cases} A(i+1, j; k+1, l) + \sigma \\ A(i+1, j; k, l) + \sigma_{\text{gap}} \\ A(i, j; k+1, l) + \sigma_{\text{gap}} \end{cases} \quad (20)$$

Initialisierung:

$$A(i, i; k, k) = \begin{cases} \sigma \text{ if } a_i \neq b_k \\ 0 \text{ else} \end{cases} \quad (21)$$

$$C(i, i; k, k) = \infty \quad (22)$$

$$M(i, i; k, l) = M(i, j; k, k) = \infty \quad (23)$$

Freie Energie der Sequenzen von i bis j, bzw. von k bis l:

$$F(i, j; k, l) = \min \begin{cases} F(i+1, j; k+1, l) + A(i, i+1; k, k+1) \text{ (i, k ungepaart)} \\ \min_{u,v} \begin{cases} C(i, u; k, v) + F(u+1, j; v+1, l) \\ + A(u, u+1; v, v+1) + A(i, i; k, k) \end{cases} \text{ (i, k gepaart mit k, v)} \end{cases} \quad (24)$$

eingeschlossene Energie zwischen dem Basenpaar (i,j) bzw. (k,l)

$$C(i, j; k, l) = \min \begin{cases} H(i, j) + H(k, l) + A(i, j; k, l) \\ \min_{i < u < v < j \quad k < x < y < l} \begin{cases} I(i, j; k, l; u, v; x, y) + C(u, v; x, y) \\ + A(i, u; k, x) + A(v, j; y, l) \end{cases} \\ \min_{i < u < j \quad k < x < l} \begin{cases} M(i+1, u; k+1, x) + M^1(u+1, j-1; x+1, l-1) \\ + A(u, u+1; x, x+1) + A(i, i+1; k, k+1) \\ + A(j-1, j; l-1, l) \end{cases} \end{cases} \quad (25)$$

Energie von Multiloops:

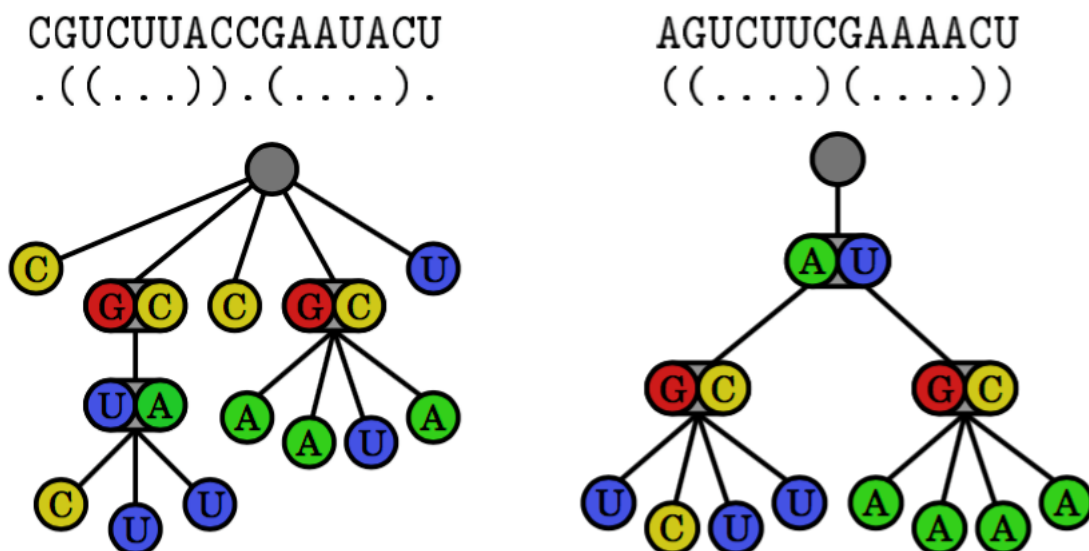
$$M(i, j; k, l) = \min \begin{cases} M(i, j-1; k, l-1) + A(j, j-1; l, l-1) \\ \min_{i < u < j \quad k < x < l} \begin{cases} C(u, j; x, l) + A(u, j; x, l) \\ \min_{i < u < j \quad k < x < l} \begin{cases} M(i, u-1; k, x-1) + C(u, j-1; x, l-1) + A(u-1, u; x-1, x) \end{cases} \end{cases} \end{cases} \quad (26)$$

$$M^1(i, j; k, l) = \min \begin{cases} M^1(i, j-1; k, l-1) + A(j-1, j; l-1, l) \\ C(i, j; k, l) \end{cases} \quad (27)$$

→ **Komplexität:** $\Omega(\text{Zeit}) = \Omega(n^6)$; $\Omega(\text{Memory}) = \Omega(n^4)$ →

5.3 TREEforester - zuerst falten, dann alignen

- Wie alignt man gefaltete RNA-Strukturen?
Nach welchen Kriterien wurde die scheinbar beste Faltung gewählt? → Tree-Editing (Bestrafung von strukturellen Mismatches)
- Sind sie überhaupt kompatibel nach ihrer Einschränkung? (Liegen Überkreuzungen vor und sind die Grundvoraussetzungen der Struktur gleich, z.B. keine Pseudoknots) → Tree-Alignment
- Im Falle von sich nicht kreuzenden RNA-Strukturen werden Bäume als vergleichende Datenstruktur erzeugt. Ein RNA-Baum ist ein geordneter Baum dessen Knoten einzelne Basen oder Basenpaare darstellen
- Insgesamt können somit alle Bäume zu einem RNA-Strukturen-Wald zusammengefasst werden (RNAforester).



Tree-Editing: wandle Baum A in Baum B um

- Basen umbenennen
- Basen löschen/hinzufügen
- Basenpaare umbenennen

- Basenpaare hinzufügen/löschen

Tree-Alignment: Erzeugung eines common Super-Trees

5.4 Lokales Falten

Vorhersagequalität nimmt mit Moleküllänge ab

Viele Moleküle haben keine globale Struktur aufgrund der Interaktionen in der Zelle

$$C'(i, j) = \begin{cases} C(i, j) & \text{if } j - i < x \\ \infty & \text{else} \end{cases} \quad (28)$$

→ Sliding-Window-Approach

→ lokales Backtracking mit

$$\text{RNALFold} = \begin{cases} F(n, n) & \text{if } F(i, n) < F(i + 1, n) \\ \text{NOTHING} & \text{else} \end{cases} \quad (29)$$

- Dinukleotidshuffling
 - klärt Frage, wie stabil die Struktur gegenüber ähnlichen Strukturen ist
 - Annahme: ähnliche Sequenzen sollten gleichen Dinukleotidinhalt haben

6 RNA-RNA-Interaktionen (RNA-Interferenz)

Durch das Falten von zwei RNA-Molekülen kommt es zu sogenannten RNA-RNA-Interaktionen oder RNA-Interferenzen (kurz: RNAi).

RNAi ermöglicht eine spezifische Steuerung von Wechselwirkungen.

Basenpaarregelungen und Stacking-Gesetzmäßigkeit gelten sowohl intra- als auch intermolekular

RNAi dient der Gerüstbildung und Erzeugung von RNA-Enzymen:

- Spliceosomen
- snoRNA/rRNA
- bakterielle sRNA (inhibieren virale mRNA in Weiterverarbeitung)
- miRNA (inhibieren virale mRNA in Weiterverarbeitung)

Hierbei gibt es zwei Typen von kurzen mRNA-Sequenzen, die helfen virale mRNA in ihrer Weiterverarbeitung zu inhibieren (miRNA und sRNA)

Zur Vorhersage der RNA-RNA-Interaktionsstruktur können verschiedene Abstraktionsstufen genutzt werden. Das vollständige Energiemodell (sequentielle Vorhersage: zwei Moleküle werden als ein Molekül betrachtet und gefaltet) ist sehr komplex mit $O(n^6)$.

- beliebig viele Interaktionsstellen möglich
- Intermolekulare Basenpaare innerhalb von intramolekularen Loops
→ zum Beispiel Kissing Hairpins
- keine intramolekularen Pseudo-Knoten
- keine überschneidenden intermolekularen Basenpaare
- intermolekulare Bp = externe Bp
- intramolekulare Bp = interne Bp
- ancestrale Bp = interne Bp, die externe Bp einschließen
- Eltern-Bp = ancestrale Bp mit minimaler Distanz
- subsummierende Bp = ancestrale Bp von jeweiliger Sequenz A,B, wobei A alle externen Bp, wie B auch einschließt

Dadurch können geschlossene Strukturen definiert werden:

- ein einzelnes externes Basenpaar
- die äußeren externen Basenpaare + Eltern-Bp
 - äquivalente Eltern-Bp
 - Elter A subsumiert Eltern B
 - Elter B subsumiert Eltern A

→ Damit kann die RNA-RNA-Interaktion in feste Bereiche vollständig zerlegt werden, die entweder geschlossene Strukturen sind oder aus rein intramolekulare Sequenzen bestehen (Reidys, Stadler, 2009)

Die Komplexität kann auf $O(n^3)$ durch Zusammenfassen von Termen reduziert werden. Hierbei werden beim Forward- und beim Backward-McCaskill-Algorithmus zusätzliche Matrizen eingespeichert. Die Backward-Matrix dient der Bestimmung der Zugänglichkeit der interagierenden Moleküle.

6.1 RNA miteinander falten und konkatenieren

Im Allgemeinen geht man wie folgt vor, um solche RNAi zu bestimmen:

- Ermittle die gemeinsame Struktur von zwei RNA-Molekülen
- Finde die Bindestellen von kleineren RNA-Fragmenten
- RNA miteinander falten (ohne Pseudoknoten(**) herzustellen)

Festlegung: Der Loop mit Konkatenationsstelle ist der externe Loop.

(**) Pseudoknoten sind sich überkreuzende Basenpaare und kommen auch in Natur vor (z.B. Kissing Hairpins, H-Typ). Der Ausschluss ermöglicht eine polynomiale Berechnung der Struktur

→ simple Pseudoknoten können mit dem RNAPKplex aus dem Vienna RNA-package gelöst werden $O(n^3)$

Zur Vereinfachung werden Sequenzabschnitte vorhergesagt, die regulatorische Relevanz haben. Die Sequenzen werden vereinfacht und dann gescannt. Die Vorhersage beruht auf zwei Termen:

$$\Delta G_{\text{Bindung}} = \Delta G_{\text{Opening}} + \Delta G_{\text{Interaktion}}$$

Bei der Ermittlung intermolekularer Helices werden intramolekulare Interaktionen nicht berücksichtigt (Verringerung der Anzahl an Interior Loops). Die Energiebeträge der Teilsequenzen werden bestimmt und gespeichert. Wahlweise kann mit diesem Verfahren entweder die minimum free Energy oder die Partition-Funktion bestimmt werden.

Besonderheit: Das erste intermolekulare Basenpaar erhält statt Hairpin-Energie eine Entropie-Strafe.

Um Rechenzeit zu sparen kann die maximale Länge der Interaktionsstelle mit einer Maximallänge beschränkt werden ($O(n^2 m^2)$). Die minimalen Interaktionsenergien der sequentiellen Teilabschnitte werden abgespeichert. Die gemeinsame Bestimmung von mfe und Zustandssumme benötigt $O((n + m)^3)$

Die Wahrscheinlichkeit einer Dimerbildung zweier RNA-Moleküle ist jedoch konzentrationsabhängig.

Anmerkung: Die Betrachtung von mehr als zwei Molekülen ist möglich, aber deutlich rechen- und speicherintensiver, da die Zahl der Rekombinationen stark ansteigt. Ein nutzbarer Algorithmus ist von Dirks et al.

6.2 RNAPlex

RNAPlex ist ein Alignemnt-ähnlicher Ansatz zur Untersuchung von RNAi

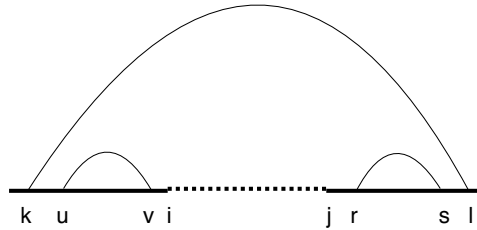
Die Interaktionsenergie kann hier schneller bestimmt werden ($O(nmL^2)$). Da die Energie von Interiorloops nicht logarithmisch betrachtet wird sondern in einer linearen Regression (Vernachlässigung von Assymetrien, somit aber auch Vernachlässigung der RNA-Struktur an sich).

- Weiterhin ist die Bindewahrscheinlichkeit davon abhängig, wie gut zugänglich das Target ist \rightarrow Zugänglichkeit von Base i: $z_i = 1 - \sum_{i \neq j} p(i, j)$
- Wahrscheinlichkeit, dass (i,j) ungepaart ist (entspricht der Öffnungs-Energie: $p = \frac{Z((i, j) \text{ ungepaart})}{Z(1, n)}$
- Struktur-Graphik für Formeln:

$$Z^n(i, j) = Z(1, i-1) + Z(j+1, n) + \sum_{k < i < j < l} Z^B(k, l) * \frac{Z^B(k, l)}{Z^{Bu}(k, l)} \quad (30)$$

$$Z^{Bu}(k, l) = \begin{aligned} & H(k, l) + \sum_{k < u < v < i} I(k, l; u, v) * Z^B(u, v) + \sum_{j < r < s < l} I(r, s; k, l) * Z^B(r, s) \\ & + M(k+1; i-1) * M(j+1, l-1) + M^2(k+1, i-1) + M^2(j+1, l-1) \end{aligned} \quad (31)$$

Die Energiewerte der Interior-Loops und 1-Bulge-Loops werden aus einer Standardtabelle mit Matthews-Parametern ausgelesen.



7 Neutrale Netzwerke von RNA-Strukturen (Peter Schuster)

Diese Methode erlaubt Aussagen über die Evolvierbarkeit von Strukturen zu formulieren. Es gibt konservierte Strukturen, welche stabil gegen Mutation sind. Für neutrale Netze ist festgelegt, dass aus einem neutralen Netz alle weiteren neutralen Netze mit einer Mutation erreicht werden können.

7.1 Shape-Abstraktion (R. Giegerich)

Simplifizierung einer Sequenz um diese vergleichbar mit anderen Sequenzen zu machen. Erzeugung sogenannter suboptimaler Shapes:

- ((((((...)))....((((.....))))).(((((...)))))) (Ausgangszustand)
- [[.].[.].[.].]
- [[.].[.].]
- [[]][[]] (stärkstes sinnvolles Abstraktionslevel)

SHREP = SHape REPresentatives: Als Ergebnis wird die Wahrscheinlichkeit der besten Struktur der Shapes ermittelt.

7.2 Faltungskinetik mit Energielandschaften

Kinetische Überlegungen im Falten von RNA:

- Moleküle in biologischen Systemen liegen meistens nicht im thermodynamischen Gleichgewicht
→ Entstehung von kinetischen Fallen == minimum free Energy (tiefe lokale Minima in der Energielandschaft)
- kinetische Fallen können zum Beispiel durch RNAi (cotranskriptionales Falten) erzeugt werden
- sogenannte RNA-Schalter wechseln von einem lokalen Minimum in ein anderes und falten somit absichtlich von einem in den anderen Zustand
→ metastabile RNA-Faltungszustände (erzeugen zusätzlichen Regulationsfaktor)
- freie Energie pro Struktur
- Wahrscheinlichkeit für bestimmte freie Energie ΔG abhängig von der Zeit

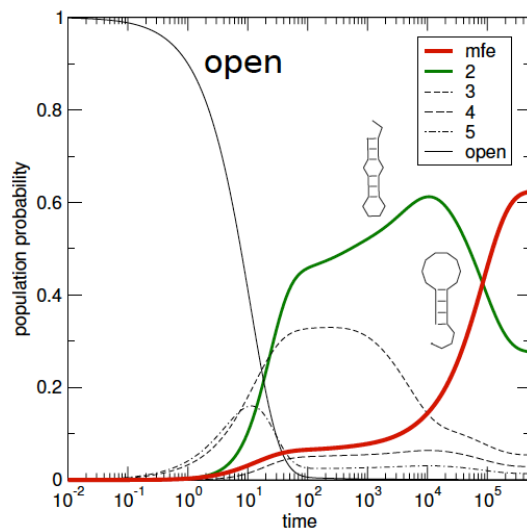


Abbildung A



Abbildung B

Abbildung A zeigt die Wahrscheinlichkeit eine Struktur einer RNA-Sequenz zu bestimmter Rechenzeit zu ermitteln. Hierbei zeigen die verschiedenen Kurven verschiedene Bereiche der Energielandschaft. Rot entspricht der minimalen freien Energie.

Abbildung B zeigt eine Energielandschaft, also die energetischen Niveaus einer RNA-Sequenz in Abhängigkeit ihrer Strukturzustände.

Erlaubte Schritte sind (Move-Set):

- öffnen von Basenpaaren
- schließen von Basenpaaren
- Verschiebung von Basenpaarteilen

7.2.1 Metropolis-Monte-Carlo

Die Anzahl zu betrachtender Zustände ist zumeist viel zu groß, weswegen mit stochastischen Methoden, wie Monte-Carlo und oder Markov-Prozessen gearbeitet werden muss.

Ein Schritt in Richtung niedriger Energie ist leichter als in Richtung höherer Energie.

Nachteil: Methode funktioniert nur für kleinere Probleme zufriedenstellend → grobkörniger Ansatz nötig: Erzeugen einer Übergangsmatrix k (exponentielles Wachstum) oder helixbasierte Move-Sets

Im ersten Schritt wird ein Zustandswechsel $i \rightarrow j$ vorgeschlagen. Überprüfe, ob **Zustand** besser ist → Metropolis-Regel:

$$p_{\text{Akzeptanz}}(i, j) = \begin{cases} e^{-\frac{G_j - G_i}{RT}} & \text{if } G_j > G_i \\ 1 & \text{else} \end{cases} \quad (32)$$

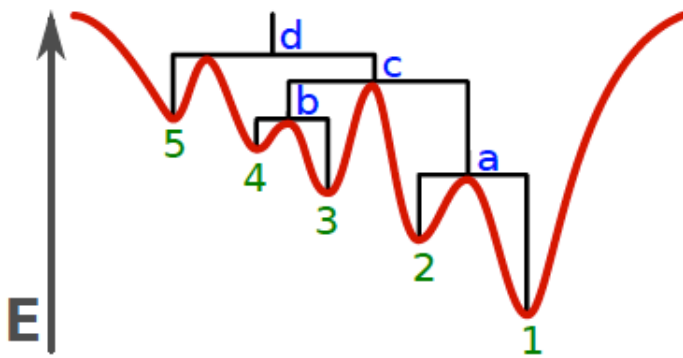
alternativ Kawasaki-Regel:

$$K_{ij} = e^{-\frac{G_j - G_i}{2RT}} \quad (33)$$

Voraussetzung: i und j sind benachbart (?wie auch immer das bei Zuständen gemeint ist?)

7.2.2 Barrier-Trees

- 1 Berechnung aller suboptimalen Strukturen des Energiebandes (→ Wuchty-Algorithmus)
- 2 Bauen eines Baums aus allen suboptimalen Strukturen
- 3 Faltungspfad zwischen Strukturzuständen i und j → Sequenz von erlaubten Schritten, die aus Anfangsstruktur i die Endstruktur j erzeugt
- 4 Wahl des optimalen Faltungspfads: Faltungspfad mit minimalen Maximum der Energie der Strukturen, die besucht werden



Die Abbildung zeigt einen Barriertree, der aus den Minima und den Sattelpunkten der Energielandschaft erzeugt wurde.

7.2.3 Baumbau mit Flooding-Algorithmus

Erzeugung einer Strukturliste, die nach Energiewerten aufsteigend sortiert ist. Diese wird dann als Hash eingespeichert.

besuchte Struktur → Niveaunummer

- Beginn bei Struktur 0 mit Niveaunummer 0
- Besuchen aller Nachbarn → Überprüfe im Hash
 - a gibt es keinen Nachbarn im Hash → neues Minimum im Hash
 - b haben alle die selbe Niveaunummer → alle gehören einem Hash an
 - c Nachbarn in zwei Niveaustufen → Sattelpunkt an der Stelle wo beide Niveaus zusammenkommen
- Berechnung der Wahrscheinlichkeit nach Arrhenius:

$$p(x, y) = A * e^{-\frac{E_{sx} - E_{sy}}{RT}} \quad (34)$$

7.2.4 Direkte Pfade

Die Nutzung direkter Pfade stellt eine Alternative zum Flooding-Algorithmus dar.

Gegeben sind zwei Sets von Basenpaaren (A,B)

a → b: erlaube nur Verschiebungen, die entweder Basenpaare A außer B wegnehmen oder ein BP aus B außer A hinzufügen.

Heuristiken zur Bestimmung von guten direkten Pfaden:

- 1 Morgan-Higgs-Heuristik: Wahl des besten Schritts
- 2 Find-Path-Heuristik: Generiere alle möglichen Schritte und wähle davon die fünf besten aus

7.2.5 Cotranskriptional Folding

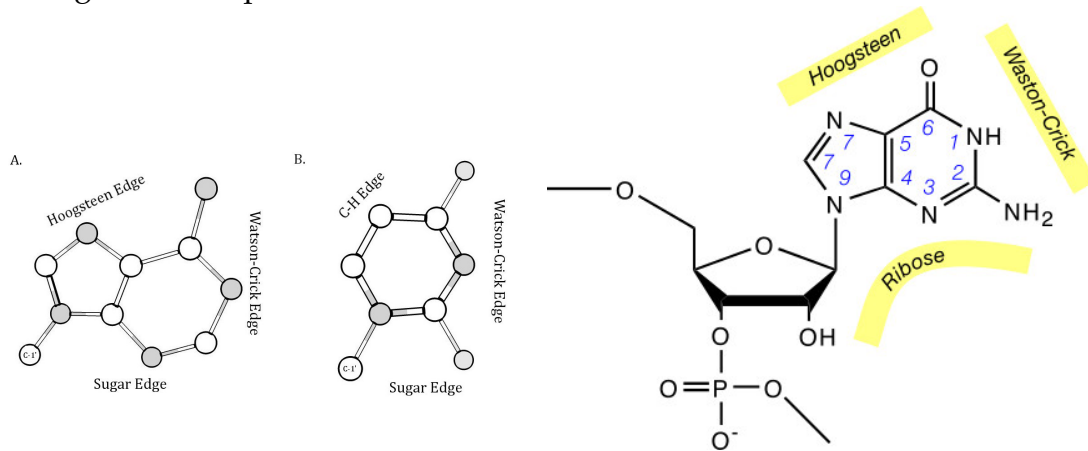
Programm: KinWalker

Barmap-Ansatz:

- 1 Zufälliges Wählen eines Barrier-Tree
- 2 Simulieren des Barriertrees auf einen neuen Tree und damit die Energielandschaft abändern
- 3 Mappe die Niveaus von Barrier-Tree I zu denen von Barrier-Tree II
- 4 Erneut zu Punkt 2 und mit Barrier-Tree II weitersimulieren

8 weitere Bindungsarten, erlaubte Basenpaare

- Hoogsteen base pair⁶



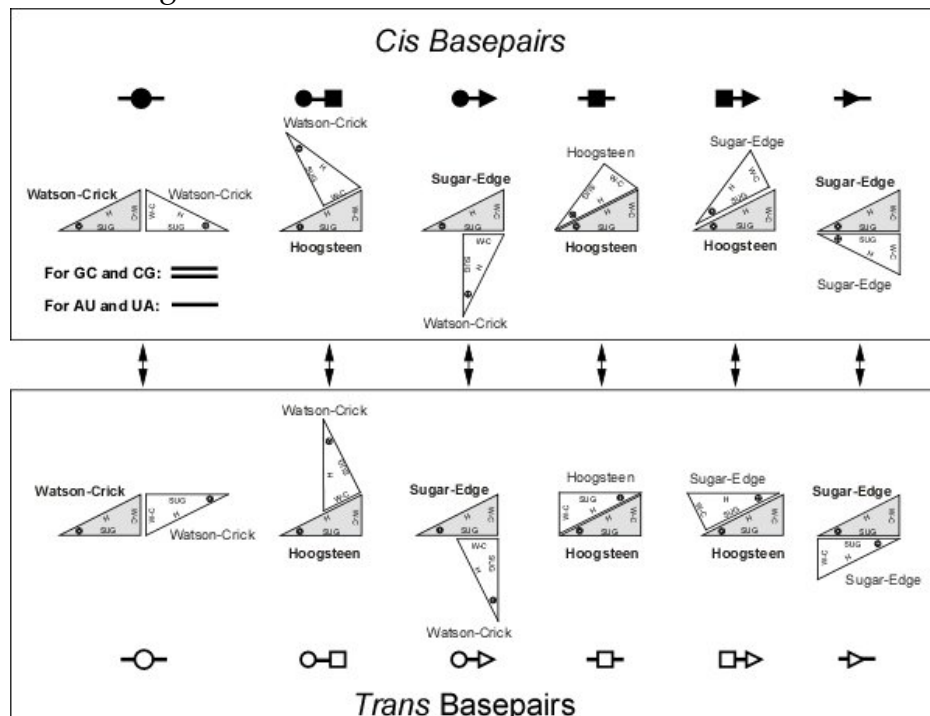
Jede Base kann mit jeder ihrer Kanten zu jeder Kante jeder Base ein Basenpaar bilden.

Non-Standard Basepairs:

Struktur motive: Pattern von Standard basepairs führt zu speziellen 3D-Struktur (Kink-Turn)

Bifurcations (tripletts meistens) $12 * 12 * 2$ mögliche Basenpaare: Warum 288?

Darstellung:

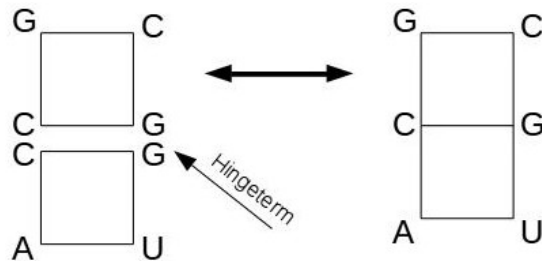


⁶https://en.wikipedia.org/wiki/Hoogsteen_base_pair

Isoelektrische Basenpaare

Änderung eines isoelektrischen Basenpaars gegen ein anderes ändert nichts an der Struktur

Listen von isoelektrischen Basenpaaren erstellt von Leontis und Westhof



Programme: MC-Fold, RNAWolf

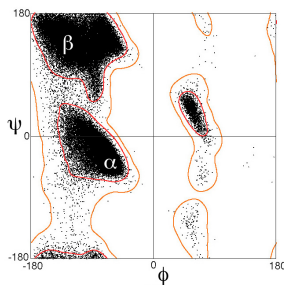
9 Proteine

- 20 Aminosäuren
- drei positiv geladene Aminosäuren (basisch): Arg (R), His (H), Lys (K)
- zwei negativ geladene Aminosäuren (sauer): Asp (D), Glutaminsäure (E)
- sehr unterschiedlich in den Seitenketten
- Verbindung durch Peptidbindung

Frage: Wie rotieren Aminosäuren, die durch eine Peptidbindung verbunden sind, im Raum?

Stichworte: Cis, Torsionswinkel

Ramachandran Plot:⁷
allgemeines Beispiel:



⁷https://en.wikipedia.org/wiki/Ramachandran_plot

10 Sekundärstrukturelemente

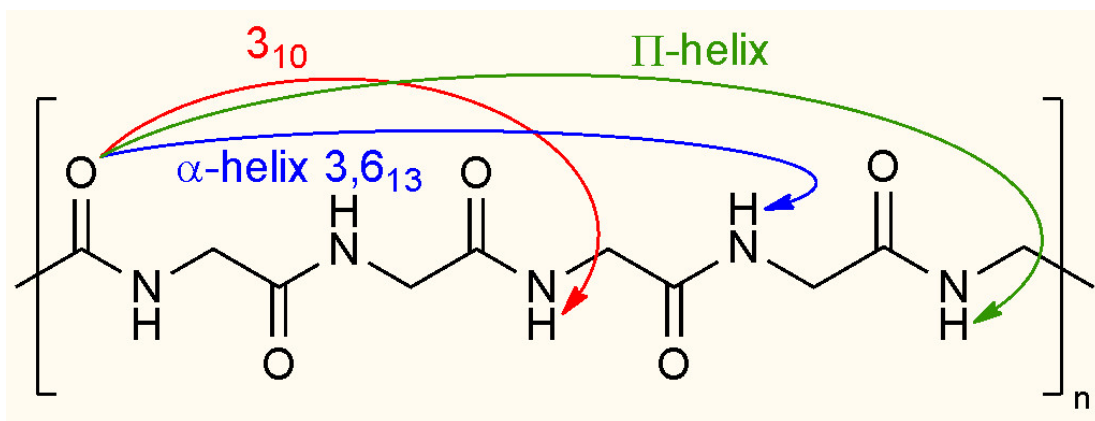
- Unterscheidung in drei Haupttypen⁸

Proteine:

- Helix α -Helix (häufigstes)
 - coiled-coil-Struktur: Helix umgeben mit einer Helix
 - Transmembranhelices: 20 - 30 Aminosäuren, hydrophob, gehen durch die Zellmembran durch
- Extended-Faltblatt: mindestens zwei Faltblätter immer zusammen, da diese sich gegenseitig stabilisieren
 - parallel, antiparallel
- Turn (drehen der Backbonerichtung)
- Coil (Rest)

drei Helixe: Unterscheidung, was und wie viel zwischen den Wasserstoffbrückenbindungen steht⁹

- α – Helix: 3,6,13-Helix (Helix zwischen 3. und 6. Atom, dazwischen liegen 13 Atome)
- π – Helix: 4,1,16



10.1 Chou-Fasman (Sekundärstrukturvorhersage von Proteinen)

- ca. 50% Genauigkeit

⁸<https://de.wikipedia.org/wiki/Sekund%C3%A4rstruktur>

⁹https://en.wikipedia.org/wiki/Protein_secondary_structure

- 3 Tabellen mit Scores für α (Helix), β (Faltblatt) und t (Turn) für alle Aminosäuren
 - z.B. gut für Helix: Glu (1,51), Met Ala, Leu
 - schlecht für Helix: Pro, Gly (0,57)
 - gut für Faltblatt: Val (1,7), Ile (1,6)
 - schlecht für Faltblatt: Asp, Glu (0,37), Pro (0,55)
- Unabhängig voneinander α, β, t bewerten:
 - nucleation: 4 von 6 Aminosäuren haben $S_{(\alpha)} > 1,03$
Erweitern nach links und rechts, bis Durchschnitt der letzten 4 AS $S_{(\alpha)} < 1$ haben
 - β : 3 von 5 Aminosäuren sollen $S_{(\beta)} \geq 1$ haben, letzten 4AS $S_{(\beta)} < 1$
- Turn: $score(t) = S_{(t)}(x1) \cdot S_{(t)}(x2) \cdot S_{(t)}(x3) \cdot S_{(t)}(x4)$

Weiterentwicklung:

- nicht nur eine Aminosäure sondern gesamte Umgebung anschauen

GOR-Algorithmus:¹⁰

- bis zu 70% genau - es gibt GOR1 bis GOR5, unterschiedliche Berechnungen

- drei Matritzen mit Scores
20 x 17 Matritze ($\alpha, \beta, turn$)
Beispiel für α : waagerecht: -8 bis +8, senkrecht alle Aminosäuren
- Score aus Summierung über Matriceinträge, dann ähnliche wie Chou-Fasman

Beispiel: ACCTYRARRGHSTFYSW

für R $S_{\alpha} = S^{\alpha}(-8, A) + S^{\alpha}(-7, C) + \dots + S^{\alpha}(8, W)$

- das für alle Sekundärstrukturelemente

weiterer Algorithmus: SPIDER2

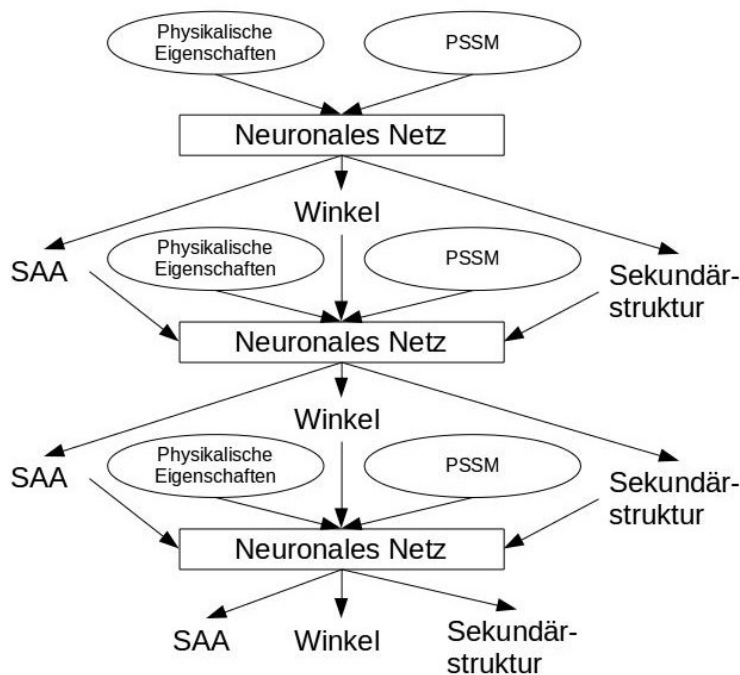
- ca. 80% genau
- Winkel zwischen Aminosäuren berechnen
- Surface Accesible Area
- Sekundärstrukturen

¹⁰https://en.wikipedia.org/wiki/GOR_method

Physikalische Eigenschaften von Aminosäuren:

- sterischer Parameter (graph shape index: dünnes oder dickes Molekül)
- Hydrophobizität
- Polarisierbarkeit
- Isoelektrischen Punkt
- Helix Wahrscheinlichkeit
- Volumen
- Falblattwahrscheinlichkeit
- zusätzlich mit psi-Blast: PSSM ermitteln (kein Ergebnis für Struktur sondern nur für Sequenz!)

dann alle diese Parameter in neuronales Netz stecken:



weitere Möglichkeit: Meta Server

- ruft mehrere Algorithmen auf
- höhere Wahrscheinlichkeit durch vergleichen der Ergebnisse (z.B. majority vote)

11 (Protein-) Strukturvorhersage (3D)

11.1 Strukturaufklärung

- Röntgen-Kristallographie
- NMR

11.2 Qualität der Strukturvorhersage

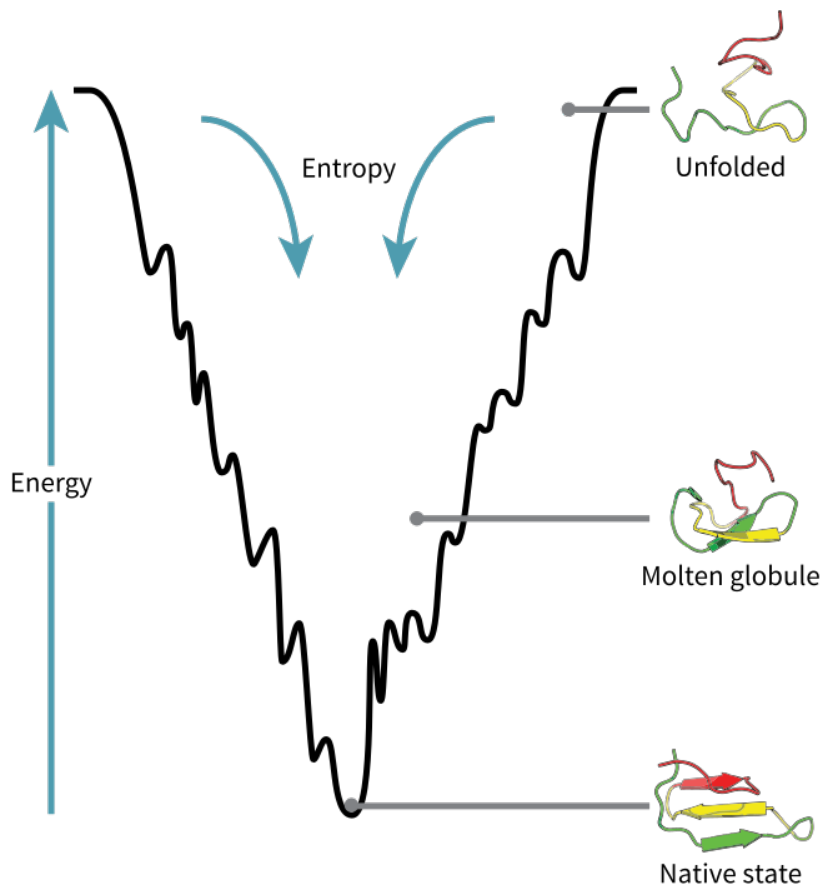
- RMSD (root mean square deviation): mittlerer Abstand in Å (10^{-10} m)

1-2 Å RMSD ist ein (sehr) guter Wert.

11.3 Problem der Strukturvorhersage (Levinthal-Paradoxon)

Eine Polypeptidkette von 100 Residuen hat 99 Peptidbindungen und daher 198 verschiedene Φ - und Ψ -Winkel. Wenn nun jeder dieser Winkel drei stabile Konformationen einnehmen kann, so ergibt sich eine Anzahl verschiedener Proteinstrukturen von 3^{198} . Wenn die Peptidkette bei der Faltung zum Protein nacheinander jeden dieser Winkel ausprobieren würde, bräuchte es länger als der Alter des Universums, um korrekt zu Falten.¹¹

¹¹https://en.wikipedia.org/wiki/Levinthal_paradox/



In der Natur falten Proteine aber im Bereich von Millisekunden. Wie ist das zu erklären?

Lokale Interaktionen führen den Faltungsprozess und schränken die Möglichkeiten ein. Experimente zeigen die resultierenden Intermediates und Transition states. Struktur und Faltung sind also *sequenzkodiert*.

11.4 Protein-Domains (Domänen)

- Protein-Untereinheit
- Falten unabhängig vom Rest des Proteinstrukturen
- Meistens funktionelle Untereinheit
- ca. 2700 Familien
- ca. 120000 Proteine (pdb)
- ca. 2/3 sind Multidomain-Proteine
- ca. 1224 Folds
 - Folds (SCOPE-Datenbank) Structural classification of proteins

- All α
- All β
- α/β , abwechselnde $\alpha/\beta \Rightarrow$ parallele β -Faltblätter
- $\alpha + \beta$, getrennte $\alpha, \beta \Rightarrow$ antiparallele β -Faltblätter
- Multidomain ($\alpha + \beta$)
- Andere (coiled coil, membrane, cell-surface)

11.5 Zwei Typen von Vorhersagen

- Ab initio
- Template based
 - Homology based
 - Threading

11.5.1 Ab-initio-Vorhersage

- Suche Strukturvorschläge
- Bewerten der Strukturen
 - physikalisch
 - knowledge-based $\log(\frac{\text{observed}}{\text{expected}})$

Physikalisch Molecular force field

E_{Bindung}

- Bindungen
 - Abstand
 - Winkel α (Bindung)
- Winkel ϕ (Torsion)

$E_{\text{ungebunden}}$

- Ladungen
- Dipol

$$E = E_{\text{Bindung}} + E_{\text{Nicht-Bindung}}$$

$$E_{\text{Bindung}} = \sum_{\alpha} k(\alpha - \alpha_0)^2$$

$$+ \sum_{\text{Bindungen}} k(r - r_0)^2$$

$$+ \sum_{\phi(\text{Torsion})} \frac{V_n}{2}(1 - \cos(n\phi - \gamma))$$

$$\text{Alternative für Bindungspotential (Morse-Potential¹²) } \sum_{\text{Bindungen}} D_e * (1 - e^{-a(r-r_0)^2})$$

k Kraftkonstante

α Bindungswinkel

r Bindungslänge

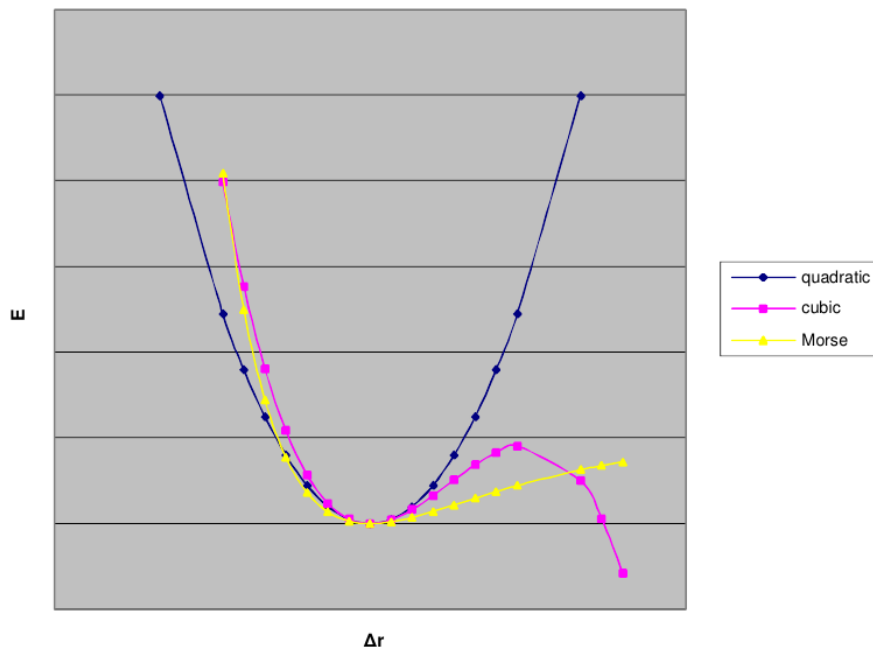
r_0 Bindungslänge mit der geringsten potentiellen Energie

D_e Dissoziationsenergie

$$a = (0.5 * \frac{k}{D_e})^{1/2} \text{ "Steifigkeits-"Konstante}$$

V_n Barrier height

γ Phasenverschiebung



$$E_{\text{Nicht-Bindung}} =$$

$$\sum_{i,j \in \text{Atome}} \frac{P_i P_j}{\epsilon r_{i,j}} \text{ Ladung: Coulomb-Terme}$$

$$+ \sum_{\text{Paar}} \frac{c}{r^{12}} - \frac{c}{r^6} \text{ Dipol: Van-der-Waals-Kräfte, Lennard-Jones-Potential 12, 6}$$

cut-off-radius

¹²https://en.wikipedia.org/wiki/Morse_potential

Lösungsmittel

- implicit solvent
- explicit solvent

Spezialterme: H-Terme, Π -Interaktionen

Knowledge based

- coarse-graining
- c_α als Beschreibung der AS
- Alle Backbone-Atome
- Alle Backbone-Atome + repräsentativ die Sk (center of mass)
- Alle Atome

Einfache Potentiale

- Abstand der Aminosäuren
- Nachbarschaft der Aminosäuren

QUARK

- Backbone atomweises Paar-Potential
- Sk-Schwerpunkt
- Excluded volume
- H-Bindungen
- Surface accessible area
- Torsionswinkel im Backbone
- Distanzen von Fragmenten
- Gyrationradius
- Relative Position der Strukturen (Faltblatt/ Helices)
 - $\beta\alpha\beta$ -linkshändig
 - $\beta\alpha\beta$ -packing
 - α -packing

– β -packing

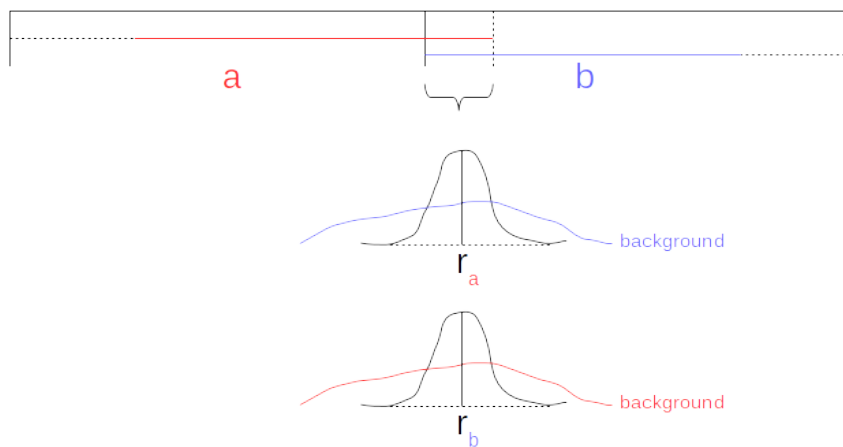
Konformation \Rightarrow dafür Minimum

- Steepest descent
- Conjugate gradient
- Newton-Verfahren
- Monte-Carlo-Verfahren
- Simulated annealing

11.5.2 Template based methods

Homology based

- Sequenzalignment zu den Sequenzen der bekannten Strukturen
- Alignment der Sequenz zur Struktur des Kandidaten
- Bauen einer Struktur aus dem Alignment
- Bewerten der Struktur



- Verbinden der Distanzpotentiale (gewichtet, multiplikativ)

CASP (critical assessment of structure prediction)

Threading

1. Sequenz-Struktur-Alignment zur Identifizierung der Kandidaten

$$\begin{pmatrix} \alpha \\ 0.5 \\ P \\ \text{groß} \end{pmatrix} \begin{pmatrix} \alpha \\ 0.3 \\ P \\ \text{klein} \end{pmatrix}$$

2. Bauen einer Struktur aus Alignment
3. Bewerten der Struktur