

# Bioinformatik von RNA- und Proteinstrukturen

# Inhaltsverzeichnis

<b>1</b>	<b>Formale Sprachen</b>	<b>1</b>
1.1	formale Grammatik G . . . . .	1
1.2	Klassifikation von formalen Sprachen . . . . .	1
1.3	Hidden Markov Model . . . . .	3
<b>2</b>	<b>Einleitung</b>	<b>5</b>
2.1	RNA . . . . .	5
2.2	R/DNA-Sekundärstruktur . . . . .	6
2.3	Strukturabbildungen . . . . .	7
<b>3</b>	<b>Strukturvorhersage</b>	<b>10</b>
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 . . . . .	13
3.4	Wuchty-Backtrack mit Nussinov-Algorithmus als Forward-Algorithmus	16
3.5	Turner-Modell (Nearest-Neighbor-Modell) . . . . .	17
3.6	Zuker-Algorithmus . . . . .	17
3.6.1	suboptimales Falten . . . . .	17
3.7	Wuchty-Algorithmus . . . . .	17
3.7.1	Wuchty-Backtracking . . . . .	17
3.8	McCaskill . . . . .	17
3.9	stochastisches Backtracking . . . . .	17
3.10	Strukturvorhersagen verbessern . . . . .	17
3.11	Konsensusstrukturvorhersagen . . . . .	17
3.12	Wie kann RNA evolvieren? . . . . .	17
3.12.1	Neutrale Netzwerke . . . . .	17
3.12.2	SHAPE-Abstraktion . . . . .	17
3.12.3	Energielandschaften . . . . .	17
3.12.4	Faltungskinetik . . . . .	17
3.12.5	Barriers Trees . . . . .	17
3.12.6	Flooding-Algorithmus . . . . .	17
3.12.7	Co-transcriptional folding . . . . .	17
<b>4</b>	<b>weitere Bindungsarten, erlaubte Basenpaare</b>	<b>18</b>
<b>5</b>	<b>Proteine</b>	<b>19</b>
<b>6</b>	<b>Sekundärstrukturelemente</b>	<b>20</b>
6.1	Chou-Fasman (Sekundärstrukturvorhersage von Proteinen) . . . .	20

# 1 Formale Sprachen

Formale Sprache<sup>1</sup> L über Alphabet  $\Sigma$

$L \subseteq \Sigma^*$

mit  $\Sigma^*$  = Kleensche Hülle<sup>2</sup> von  $\Sigma$

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

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

$\varepsilon \rightarrow$  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
- $\Sigma$  = 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 Comsky-Hierarchie<sup>3</sup>:

- 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

---

<sup>1</sup>[https://de.wikipedia.org/wiki/Formale\\_Sprache](https://de.wikipedia.org/wiki/Formale_Sprache)

<sup>2</sup>[https://de.wikipedia.org/wiki/Kleenesche\\_und\\_positive\\_H%C3%BClle](https://de.wikipedia.org/wiki/Kleenesche_und_positive_H%C3%BClle)

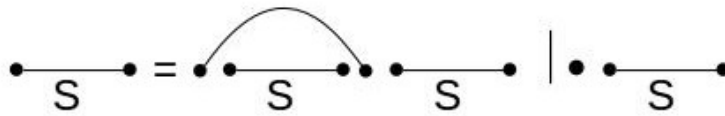
<sup>3</sup><https://de.wikipedia.org/wiki/Chomsky-Hierarchie>

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

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

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

jetzt auf RNA-Vorhersagen:

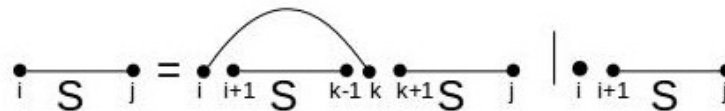


scoring scheme: Bewertung von  $\sigma(\curvearrowright) = 1$ ,  $(\sigma(\text{---}))$ ,  $\sigma(\bullet) = 0$   
scoring function:

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

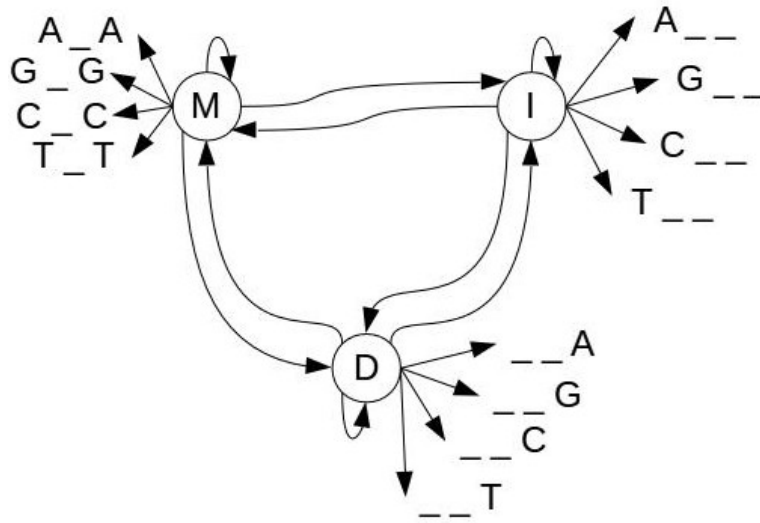
choice function:

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



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

### 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:

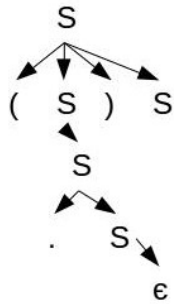


Faltungsgrammatik

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

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

<sup>4</sup> Funktion:

- Informationsträger
- Regulator/Katalysator
- Theorieder 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)

---

<sup>4</sup><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 Triplets (Multiplets): 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:

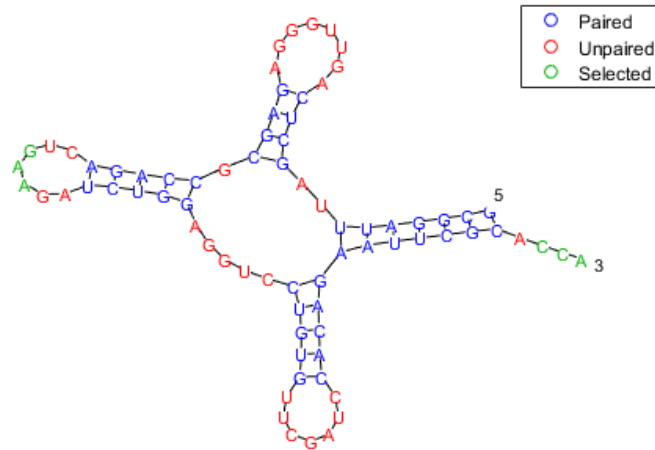
- Röntgenkristallographie: Kristall benötigt → oft schwierig
- nuclear magnet resonanz (nmr): stark konzentrierte Lösung benötigt, nur Distanzen zwischen Atomen ermittelbar

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

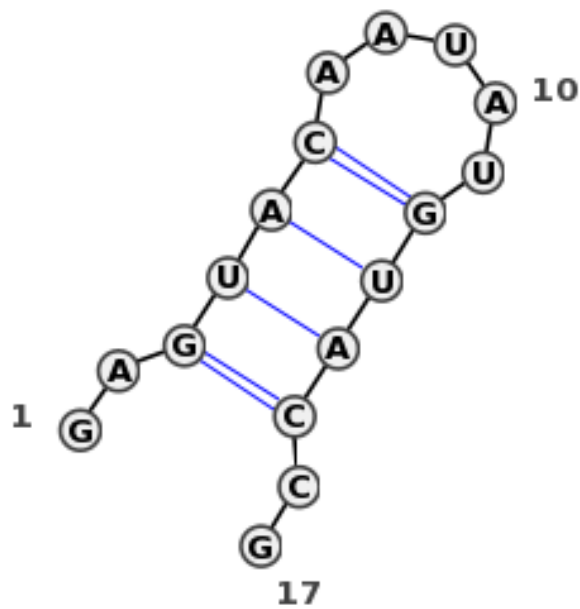


## 2.3 Strukturabbildungen

1. Strukturplot:

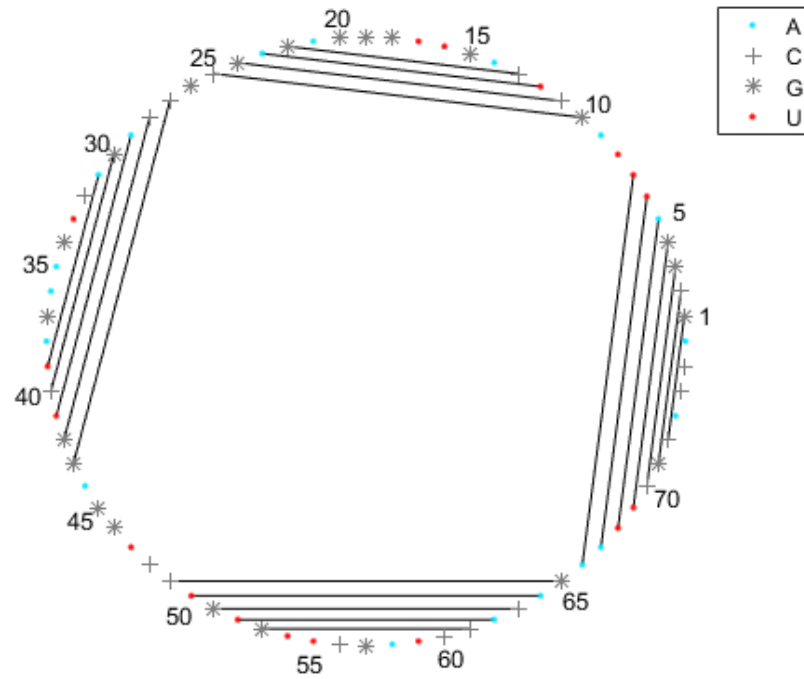


2. Dot-Bracket:

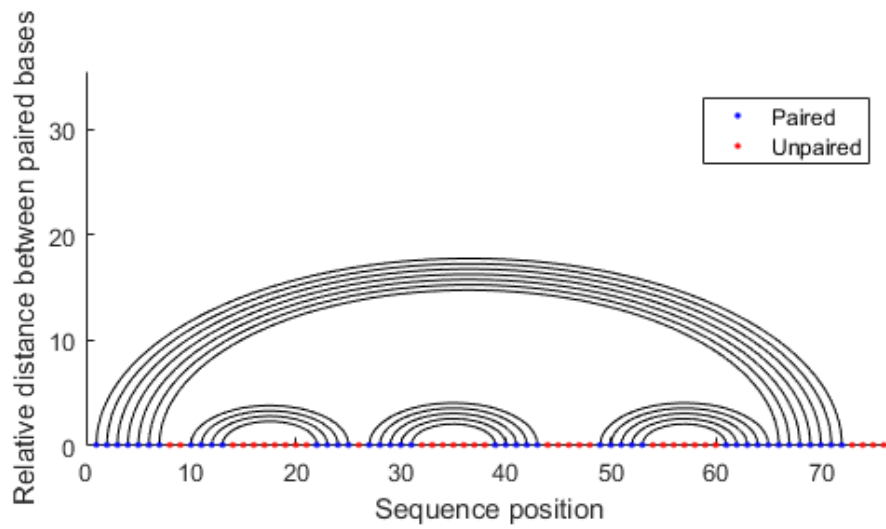


Seq:	GAGUACAAUAUGUACCG
Str:	..(((.....)))...

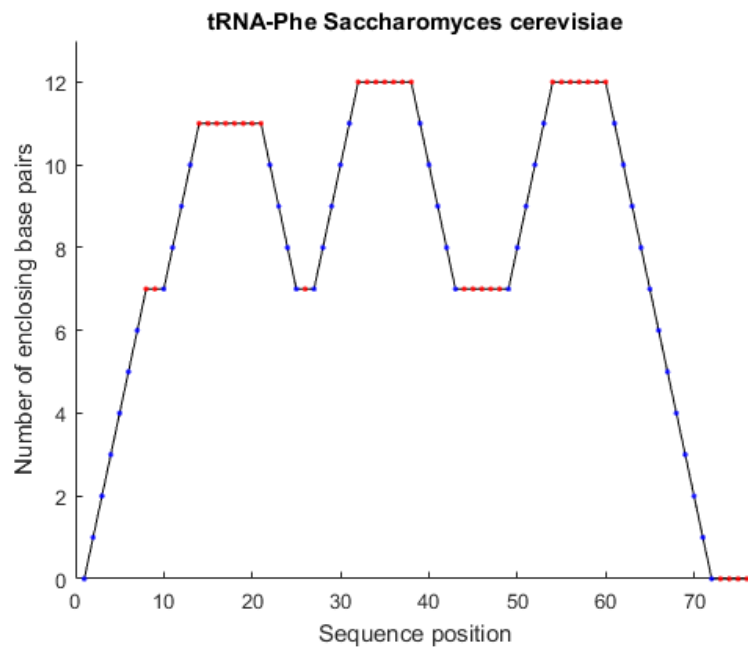
3. Zirkulärplot:



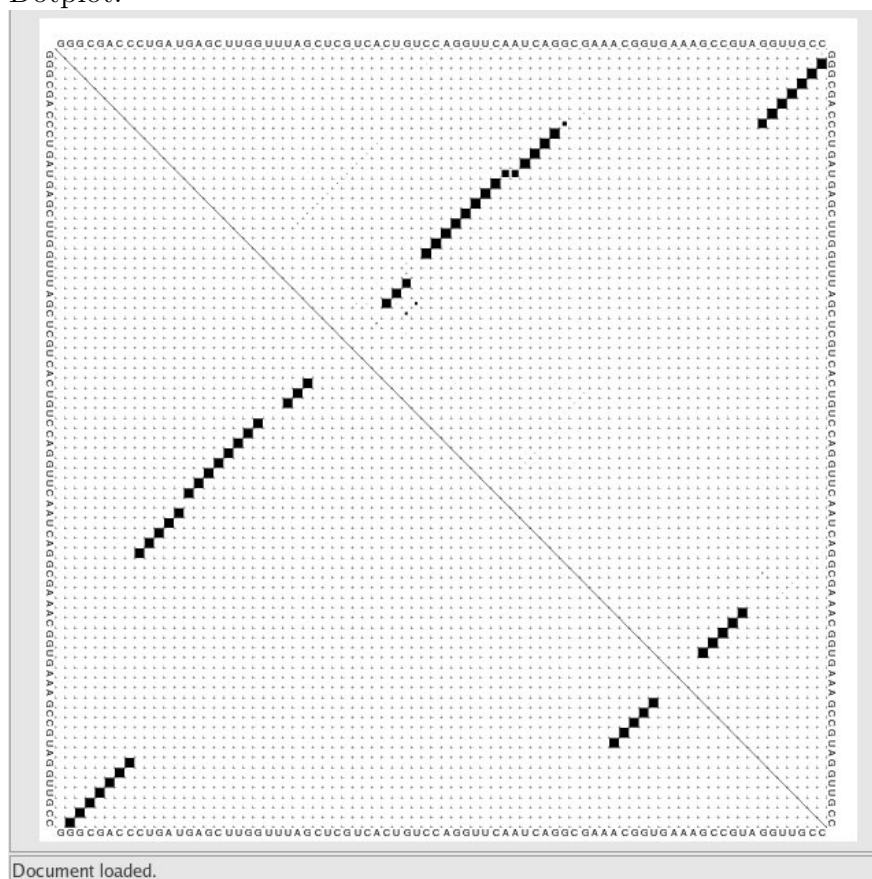
4. Bogenplot:



5. Mountainplot:



6. Dotplot:



### 3 Strukturvorhersage

- durch Aufteilung kann Dynamics Programming verwendet werden
- Beginn: einzelne Basen  $\rightarrow$  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:**

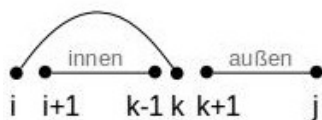
- $N(i, i) = 0$
- $N(i, j) = 0$  if  $i < j \leq i + 3$  (siehe Regel 2)
- $N(j + 1, j) = 0$

- **Berechnung:**

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

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

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



$\rightarrow$  höchste Punktzahl wahrscheinlichste Sekundärstruktur

Ressourcenbedarf:

- Speicher:  $\mathcal{O}(n^2)$
- Prozessor:  $\mathcal{O}(n^3)$

### 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:**

$$Z(i, j) = + \left\{ \begin{array}{l} Z(i + 1, j) \text{ (ungepaart)} \\ \sum_{i+3 < k \leq j} Z(i + 1, k - 1) * Z(k + 1, j) * 1 \end{array} \right.$$

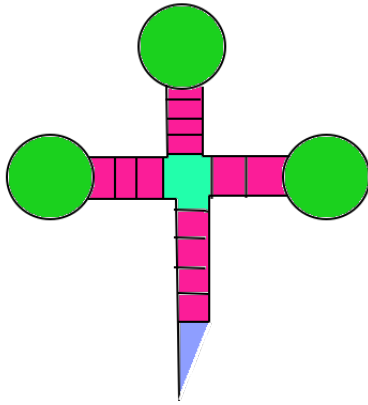
### 3.3 Zuker-Algorithmus




In der Praxis ist der Nussinov-Algorithmus unzureichend, da eine Maximierung von Basenpaaren nicht zwangsläufig mit einer Maximierung von Stabilität einhergeht. RNA's werden vorallem 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

- 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:**

$$F(i, j) = \min \begin{cases} F(i + 1, j) \text{ (iungepaart)} \\ \min_{i+3 < k \leq j} C(i, k) + F(k + 1, j) \end{cases}$$

$$C(i, j) = \min \begin{cases} H(i, j) \\ \min_{i < k < l < j} I(i, j, kl) + C(k, l) \\ \min_{i < u < l-5} M(i + 1, u - 1) + M^1(u, j - 1) + a \end{cases}$$

$$M(i, j) = \min \begin{cases} M(i+1, j) + c \\ \min_{i < k \leq j} C(i, k) + b + (j - k) * c \\ \min_{i < k < j-5} C(i, k) + b + M(k+1, j) \end{cases}$$

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

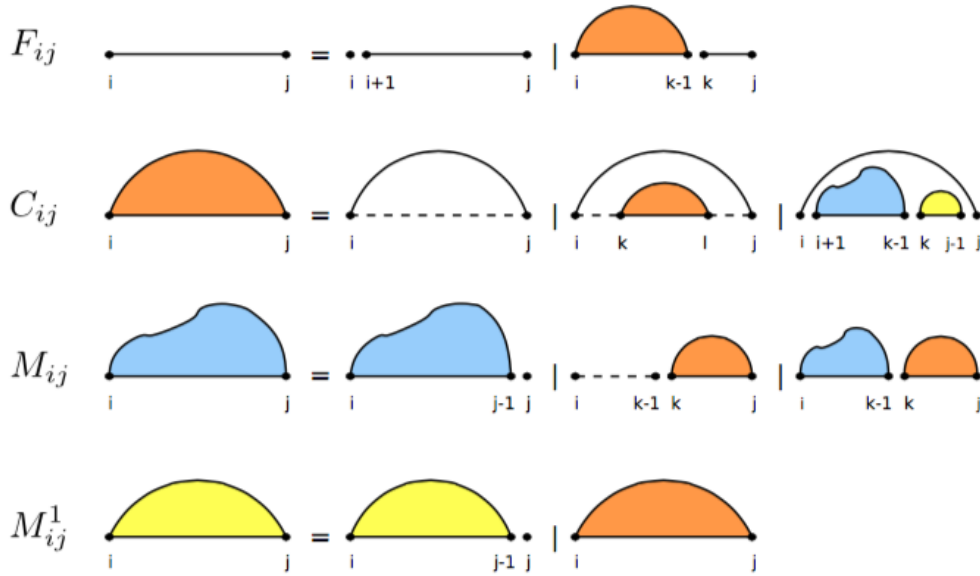
$H$  und  $I$  sind gegebene Energietabellen

$a$  - 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  $j - l + k - i < \text{Schwellenwert} \approx 30$ .



### 3.4 Wuchty-Backtrack mit Nussinov-Algorithmus als Forward-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 durchführen**  $\rightarrow$  outputs  $N(i, j)$

2. **Backtrack:**

```

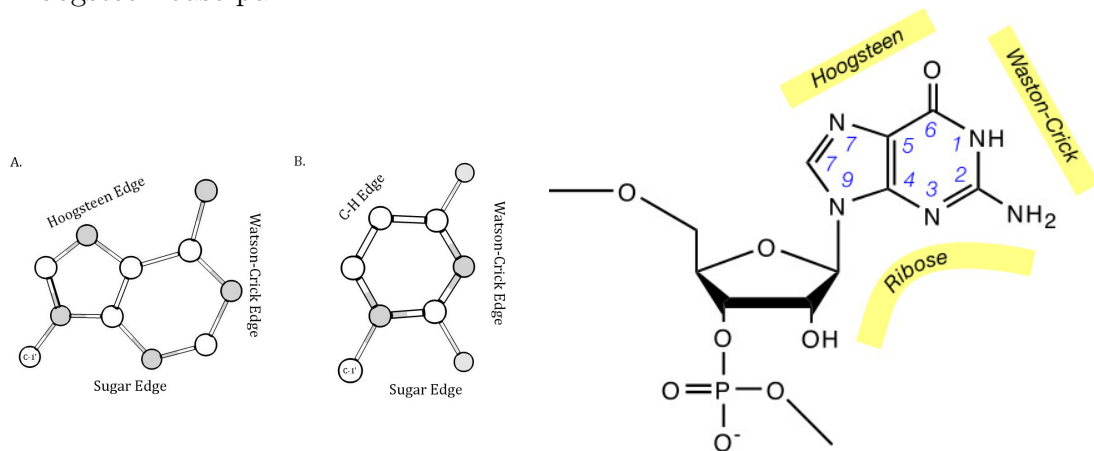
push([1, n], 0, 0)
while (stack not empty)
  • Nimm Intervall  $a$  aus  $I$ 
  • Berechne die Anzahl an möglichen Basenpaaren in den anderen Intervallen
  •  $Z = BP + \sum_{b \neq a \in I} N[b] \rightarrow$  in  $a$  mind.  $x - Z$  Basenpaare
  •  $if(N(i, j - 1) \geq x - Z)$ 
    –  $a' = [i, j - 1]$ 
    – 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 Turner-Modell (Nearest-Neighbor-Modell)
- 3.6 Zuker-Algorithmus
  - 3.6.1 suboptimales Falten
- 3.7 Wuchty-Algorithmus
  - 3.7.1 Wuchty-Backtracking
- 3.8 McCaskill
- 3.9 stochastisches Backtracking
- 3.10 Strukturvorhersagen verbessern
- 3.11 Konsensusstrukturvorhersagen
- 3.12 Wie kann RNA evolvieren?
  - 3.12.1 Neutrale Netzwerke
  - 3.12.2 SHAPE-Abstraktion
  - 3.12.3 Energielandschaften
  - 3.12.4 Faltungskinetik
  - 3.12.5 Barriers Trees
  - 3.12.6 Flooding-Algorithmus
  - 3.12.7 Co-transcriptional folding

## 4 weitere Bindungsarten, erlaubte Basenpaare

- Hoogsteen base pair<sup>5</sup>



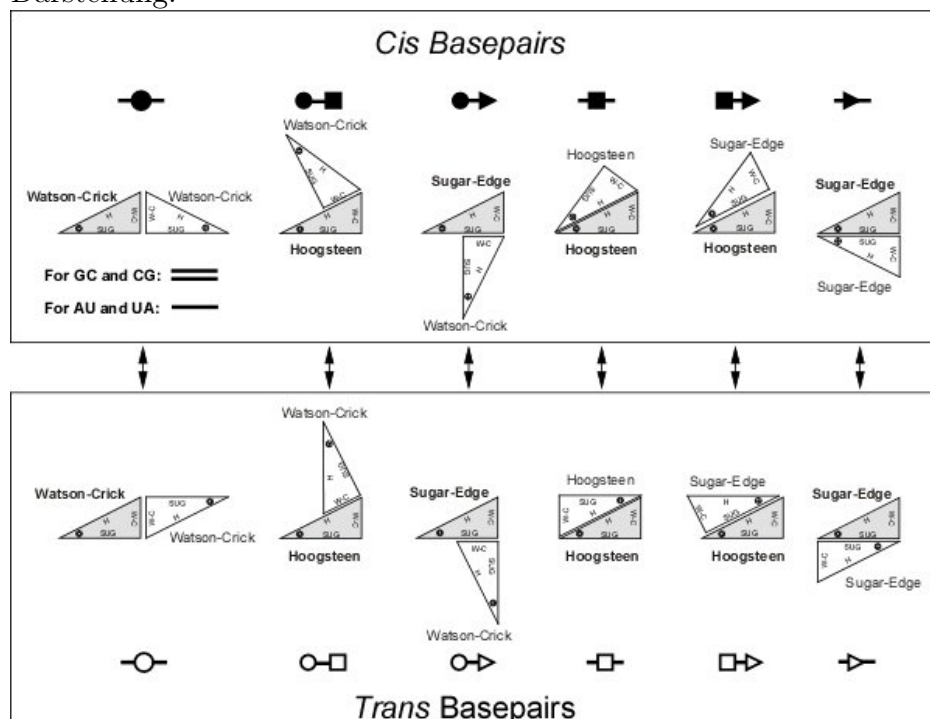
Jede Base kann mit jeder ihrer Kanten zu jeder Kante jeder Base ein Basenpaar bilden.

Non-Standard Basepairs:

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

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

Darstellung:

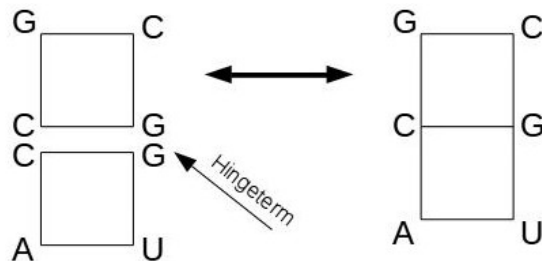


<sup>5</sup>[https://en.wikipedia.org/wiki/Hoogsteen\\_base\\_pair](https://en.wikipedia.org/wiki/Hoogsteen_base_pair)

### 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

## 5 Proteine

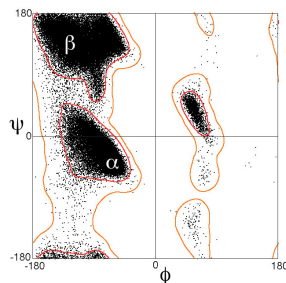
- 20 Aminosäuren
- drei positiv geladene Aminosäuren (basisch): Arg (R), His (H), Lys (K)
- zwei negativ geladene Aminosäuren (sauer): Asp (D), Glutaminsäure (E)
- sehr unterschiedlich in den Seitenketten
- Verbindung durch Peptidbindung

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

Stichworte: Cis, Torsionswinkel

Ramachandran Plot:<sup>6</sup>

allgemeines Beispiel:



<sup>6</sup>[https://en.wikipedia.org/wiki/Ramachandran\\_plot](https://en.wikipedia.org/wiki/Ramachandran_plot)

## 6 Sekundärstrukturelemente

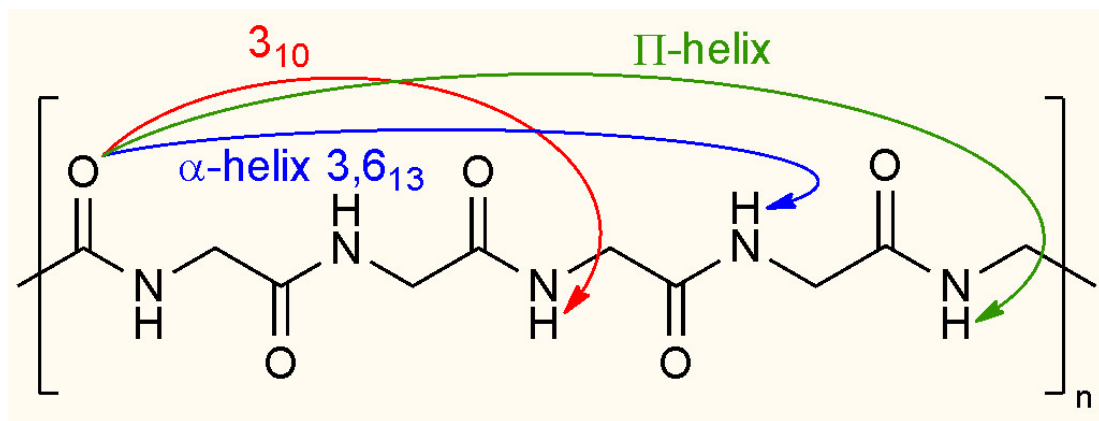
- Unterscheidung in drei Haupttypen<sup>7</sup>

Proteine:

- Helix  $\alpha$ -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 steht<sup>8</sup>

- $\alpha$  – Helix: 3,6,13-Helix (Helix zwischen 3. und 6. Atom, dazwischen liegen 13 Atome)
- $\pi$  – Helix: 4,1,16



### 6.1 Chou-Fasman (Sekundärstrukturvorhersage von Proteinen)

- ca. 50% Genauigkeit

<sup>7</sup><https://de.wikipedia.org/wiki/Sekund%C3%A4rstruktur>

<sup>8</sup>[https://en.wikipedia.org/wiki/Protein\\_secondary\\_structure](https://en.wikipedia.org/wiki/Protein_secondary_structure)

- 3 Tabellen mit Scores für  $\alpha$  (Helix),  $\beta$  (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  $\alpha, \beta, t$  bewerten:
  - nucleation: 4 von 6 Aminosäuren haben  $S_{(\alpha)} \geq 1,03$   
Erweitern nach links und rechts, bis Durchschnitt der letzten 4 AS  $S_{(\alpha)} \geq 1$  haben
  - $\beta$ : 3 von 5 Aminosäuren sollen  $S_{(\beta)} \geq 1$  haben, letzten 4AS  $S_{(\beta)} \geq 1$
- 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:<sup>9</sup>

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

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

Beispiel: ACCTYRARRRGHSTFYSW

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

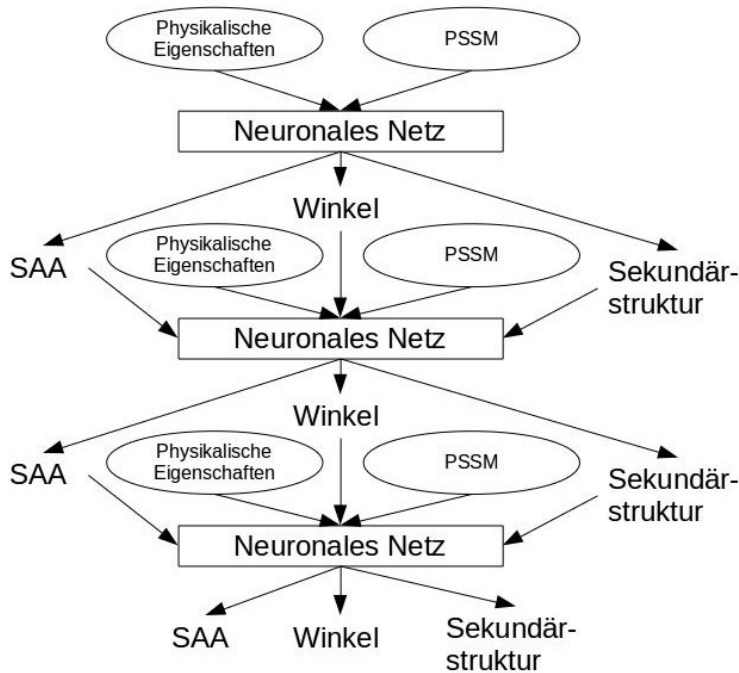
Physikalische Eigenschaften von Aminosäuren:

---

<sup>9</sup>[https://en.wikipedia.org/wiki/GOR\\_method](https://en.wikipedia.org/wiki/GOR_method)

- 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)