Aus dem Institut für Neuro- und Bioinformatik der Universität zu Lübeck

Direktor: Prof. Dr. rer. nat. Thomas Martinetz

Proteinfaltung und Alignment von Strukturen

Diplomarbeit
im Rahmen des Informatik-Hauptstudiums, überarbeitete Version

Vorgelegt von

Gerrit Leder

Ausgegeben von

Prof. Dr. Thomas Martinetz
Institut für Neuro- und Bioinformatik

Betreut von

Prof. Dr. Thomas Martinetz
Institut für Neuro- und Bioinformatik

Lübeck, Juli 2005 und Remseck, Juli 2008

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur angefertigt habe.

(Gerrit Leder)
Remseck, den 10. April 2017

Inhaltsverzeichnis

Abbildungsverzeichnis

Danksagung

Diese Diplomarbeit ist entstanden aus zwei Konferenzartikeln (s. [CGP+98] und [GIP99]), die für mich Grundlage von zwei Seminarvorträgen waren. In ausführlicher Form werden beide Artikel in der Dissertation von Goldman [Gol00] wiedergegeben.

Ich möchte allen danken, die zur Entstehung der verwendeten Artikel und den beiden Seminaren beigetragen haben. Insbesondere möchte ich Herrn Gerhard Buntrock, der mir mit immer währender Beratung kritisch und impulsgebend zur Seite gestanden hat, für seine Unterstützung danken. Weiterhin danke ich Frau Gisela Schiffko und Frau Angela Herbertz für das Korrekturlesen und Herrn Klaus-Dieter Leder für die technischen Voraussetzungen für die Arbeit.

1 Einleitung

1.1 Historischer Rückblick

Im Jahre 1953 haben Watson und Crick (nach Müller [Mül77]) die DNA-Struktur (‘desoxyribonucleic acid’ oder Desoxyribonukleinsure) entdeckt. Diese kodiert direkt alle in lebendigen Körpern vorhandenen Proteine. Sie ist somit ein Beispiel für die Informationsverarbeitung in der belebten Natur, die wir im nächsten Absatz beschreiben werden. Wir geben im Folgenden den Eintrag des Lexikons [Mül77] unter dem Begriff “Nukleinsäuren” sinngemäß wieder:

Nukleinsuren sind hochmolekulare Verbindungen, die bei Hydrolyse in stickstoffhaltige Basen (Adenin “A”, Guanin “G”, Thymin “T”, Urazil “U” und Cytosin “C”), Phosphorsäure und Zucker zerfallen. Für die räumliche Struktur der DNA haben Watson und Crick das “Helix-Modell” vorgeschlagen, das experimentell überprüft wurde und folgendermaßen beschrieben wird: “Zwei Polynukleotidfäden sind schraubenförmig umeinander gewunden und stehen durch Wasserstoffbrücken zwischen ihren Basen in Beziehung. Jede Base des einen Fadens bestimmt den Basenpartner im anderen und umgekehrt. Solche Basenpaare sind Adenin (A) und Thymin (T), Guanin (G) und Cytosin (C). DNA ist

in der Zelle in Chromosomen enthalten; aus DNA bestehen Gene. über Millionen von Zellgenerationen werden diese unverändert weitergegeben, weil die DNA die Fähigkeit zur identischen Reduplikation besitzt." Der letzte Punkt bezeichnet die Teilung einer Basenpaarfolge in der Zelle und Verdopplung des ursprünglichen Moleküls, der so genannten "Matrize", mit dem Vorrat an freien Nukleotiden im Zell-Plasma.

Als weitere Beispiele für informatische Methoden in der Biologie führen wir Aminosäure- und DNA-Sequenzen an, die in der Informatik als Zeichenketten betrachtet werden und im Gebiet der "formalen Sprachen" mit einem endlichen Alphabet dargestellt werden können.

1.2 Themenüberblick

Die Proteinfaltung wird in der Literatur auch Vorhersage von Proteinstrukturen genannt (nach Lengauer [Len96]). Die *de novo*[1] Proteinfaltung ist bis heute nicht möglich. Es gibt jedoch für die Proteinfaltung Approximationsalgorithmen von Hart und Istrail [HI94].

Ziel dieser Arbeit ist es, zu zeigen, wie wir im H-P-Modell (Hydrophob-Polar-Modell) Aminosäuresequenzen bezgülich einer Bewertungsfunktion optimal falten können. Das H-P-Modell zeichnet sich dadurch aus, dass es in der Natur die stärksten Kräfte beim Vorgang der Proteinfaltung widerspiegelt und ein stimmiges Modell mit anschaulichen Ergebnisssen in der Theorie darstellt. Dabei repräsentieren wir mit 0 und 1 die hydrophilen bzw. hydrophoben Aminosäuren.

Vom H-P-Modell kommen wir direkt zu einem weiteren Modell für das Strukturalignment, indem wir von der 0-1 Sequenzinformation zu Knoten eines Graphen abstrahieren. Hierfür werden wir mit Hilfe der Graphentheorie Modelle bilden. Dazu definieren wir ungerichtete Graphen in den Präliminarien, die wir in dieser Arbeit ausschließlich verwenden werden. Dann werden wir einen Contact-Map-Graphen definieren, den wir auch 'self-avoiding walk' nennen werden.

[1] Vorhersage, ohne dass die Proteinsequenz verwandt ist mit in ihrer Struktur bekannten Proteinen

Anhand des einfachen Modells der Aminosäuresequenzen betrachten wir zuerst also eine noch einfachere Darstellung der Abfolge der H-P-Eigenschaften jeder einzelnen Aminosäure. Es wird angenommen, dass die H-P-Kräfte in der Aminosäuresequenz die Hauptursache dafür sind, dass sich entfernte Aminosäuren aneinander anlagern. Diese H-P-Krfte, die zur Anlagerung führen, werden *nicht lokale* Ursache für die Proteinfaltung genannt. Dieses kann man sich so vorstellen, dass sich alle in einem Protein vorhandenen hydrophoben Aminsäuren in wässriger Lösung im Inneren des Proteins verklumpen. Dabei bleibt die Abfolge in der Aminosäuresequenz erhalten, und es ist Energie notwendig, um vom nativen Zustand des Proteins zum ursprünglichen, sequentiell angeordneten Zustand zurück zu kehren.

Darber hinaus zielt Hayes [Hay98b] auf alle Sequenzen ab, die *eindeutig* optimal gefaltet werden können. Das sind optimal gefaltete Sequenzen, die genau eine solche Faltung besitzen. Unter den vielen Instanzen für das Proteinfaltungsproblem im H-P-Modell untersucht Hayes die Sonderfälle, deren optimale Lösung gleichzeitig eine eindeutige Lösung ist.

Die Grundbausteine aller Proteine, egal wie komplex ihre Struktur ist, bestehen laut Biet [Bie04] aus nur 20 von der DNA kodierten Aminosäuren, die in Bachmann [Bac76] ausführlich beschrieben werden. Wir führen Aminosäuresequenzen als abstraktes und stark vereinfachendes Modell für Proteine (dreidimensionale Aminosäureketten oder Eiweiß) ein. Aminosäuresequenzen entstehen aus den kodierenden Abschnitten, die auch als Gene bezeichnet werden, der DNA. Das Sequenzieren des gesamten menschlichen Genoms wurde von der ‘Human Genome Organization’ (HUGO) 2000/2001 erreicht. Die gesamte DNA eines Genoms und ihre Repräsentation durch Aminosäuren wird die “erste Hälfte des genetischen Codes” genannt und ist durch HUGO offengelegt worden.

Wie bereits gesagt, ist die Schlussfolgerung von einer aus einem Gen entstandenen Aminosäuresequenz auf das im lebendigen Körper gebildete Protein und seine dreidimensionale Struktur bis heute unmöglich. Das Problem, die Proteinfaltung vorherzusagen, für unbekannte Proteine in der Größenordnung wie sie in der Natur vorkommen, ist bisher nicht lösbar. Als Längenbeispiel für ein natürliches Protein dient uns Hämoglobin mit 574 Aminosäuren in zwei α-Ketten der Länge 141 und zwei β-Ketten der Länge 146 (nach Lystad [Lys04]). Die von Crescenzi et al. [CGP$^+$98] so genannte “zweite Hälfte des genetischen Codes”

ist unbekannt. Dies entspricht der o. g. Aussage, dass eine Schlussfolgerung noch nicht möglich ist.

Von der Proteinfaltung in dem H-P-Modell kommen wir zum Alignment[2] zweier dreidimensionaler Strukturen; dies ist im Modell dargestellt durch die Betrachtung zweier so genannter Contact-Map-Graphen. Hierbei ist zu beachten, dass die dreidimensionale Struktur eindeutig durch den aus ihr entstandenen Contact-Map-Graphen bestimmt wird (vgl. Vendrusculo et al. [VK97]).

1.3 Verwandte Themen in der Literatur

Eine allgemeine Einführung in die Bioinformatik wird in Merkl et al. [MW03] gegeben. Und für eine Diskussion über die Eigenschaften von Ähnlichkeitsmaßen und Beispiele anderer in der Literatur vorhandener Maße ('root-mean-square-distance', ...) siehe Goldman [Gol00]. Beim Vergleich sowohl von DNA-Sequenzen als auch von Aminosäuresequenzen findet man in der Literatur andere Verfahren, wie z. B. "Probabilistisches Alignment".

1.4 Eigene Ergebnisse

Die Betrachtungen in Kapitel 5 und Abschnitt 5.2 über Anordnungen von Strukturen ("Faltung") im zweidimensionalen H-P-Modell und die von ihnen erzeugten Strukturen des in Kapitel 4 eingeführten Graphmodells sind eigene Ergebnisse. Dasselbe gilt für die nichtdeterministischen Polynomialzeitalgorithmen in Abschnitt 3.3 und im Beweis zu Satz 1.

2 Präliminarien

Als Hilfsmittel der Informatik benötigen wir folgende Theorien und Begriffe:

[2] Maß für die Ähnlichkeit

2.1 Graphentheorie

Nach Diestel [Die96] definieren wir die folgenden Begriffe der Graphentheorie:

Ein *Graph* ist ein Paar $G = (V, E)$ disjunkter Mengen. Die Elemente von E sind 2-elementige, ungeordnete Teilmengen von V. Die Elemente von V nennt man die *Ecken*, oder *Knoten*, des Graphen G, die Elemente von E seine *Kanten*. Jede Kante $x \in E$ notieren wir folgendermaßen: $x = \{u, v\}$, mit $u, v \in V$. Bildlich kann man G darstellen, indem man seine Ecken als Punkte zeichnet und zwei dieser Punkte immer dann durch eine Linie verbindet, wenn die entsprechenden beiden Ecken eine Kante bilden.

2.2 Sprachen

Die NP-Vollständigkeitstheorie ist zentraler Bestandteil der Theoretischen Informatik. Sie trifft Komplexitätsaussagen über *Sprachen* und *Entscheidungsprobleme*.

Definition 1 *Ein* Alphabet Σ *ist eine endliche Menge von Zeichen, z. B.* $\Sigma = \{0, 1\}$.

Definition 2 Σ^* *ist die Menge aller endlichen Zeichenketten über einem Alphabet und mit* λ *sei die leere Zeichenkette, die keine Zeichen enthält, bezeichnet.*

Beispiel 1 *Sei* $\Sigma = \{0, 1\}$*, dann ist* $\Sigma^* = \{\lambda, 0, 1, 00, 01, 10, 11, 000, \ldots\}$.

Definition 3 *Ein* Wort w *ist eine endliche Konkatenation* $w = a_1 \ldots a_n$ *von Zeichen mit* $n \in \mathbb{N}$ *und* $a_1, \ldots, a_n \in \Sigma$*, dem Alphabet. Oder* w *entspricht dem leeren Wort* $\lambda \in \Sigma^*$.

Definition 4 *Eine* Sprache L *ist eine Teilmenge* $L \subseteq \Sigma^*$ *von Wörtern über einem Alphabet.*

Beispiel 2 *DNA-Sprachen sind Mengen von DNA-Makromolekülen, bei uns als Wörter bezeichnet, mit bestimmten Eigenschaften über dem Alphabet* $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$. *Alle möglichen DNA-Sequenzen werden mit* $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}^*$ *dargestellt.*

2.2.1 Entscheidungsprobleme

Nach Wegener [Weg96] haben Entscheidungsprobleme stets die folgenden Eigenschaften:

- *Entscheidungsprobleme* sind Funktionen mit dem Bildbereich $\{0, 1\}$.
- Programme haben immer die Ausgabe 0 oder 1, wenn sie die Lösung für Entscheidungsprobleme berechnen.
- Zu jeder Eingabe aus einer Definitionsmenge gehört entweder die Ausgabe 1, die "ja" bedeutet, oder die Ausgabe 0, die "nein" bedeutet.
- Wir sagen, eine Eingabe wird *akzeptiert*, wenn die Ausgabe 1 ist, oder sie wird *nicht akzeptiert*, wenn die Ausgabe 0 ist.
- Alle Eingaben, für die ein Entscheidungsproblem die Ausgabe 1 liefert, definieren eine Sprache.

2.2.2 Optimierungsprobleme

Allgemein unterscheiden wir bei Optimierungsproblemen drei Varianten:

- Bei der *Optimierungsvariante* besteht die Aufgabe in der Berechnung einer Lösung, deren Wert bei Maximierungsproblemen maximal bzw. bei Minimierungsproblemen minimal ist.
- Bei der *Zahlvariante* soll der Wert einer optimalen Lösung berechnet werden.
- Bei der *Entscheidungsvariante* soll für ein Limit entschieden werden, ob der Wert einer optimalen Lösung bei Maximierungsproblemen oberhalb bzw. bei Minimierungsproblemen unterhalb des Limits liegt.

Beispiel 3 *In der Optimierungsvariante ist eine* Clique *in einem Graphen $G = (V, E)$ eine Knotenmenge V, zwischen deren Knoten alle möglichen Kanten in E existieren. Der* Wert *einer Clique ist die Zahl ihrer Knoten $|V|$. Eine Clique der Größe oder mit dem Wert k heißt* k-Clique. *Das* Cliquenproblem *CLIQUE (k) ist ein Maximierungsproblem.*

2.2.3 CLIQUE (Entscheidungsvariante)

Das Cliquenproblem in der Entscheidungsvariante lautet wie folgt (s. Definition in Reischuk [Rei99]):

Definition 5 *Wir definieren* CLIQUE *wie folgt: Gegeben sei ein Graph G und ein Wert k:*

$$\textit{CLIQUE} := \{(G, k) \mid G \textit{ besitzt eine } k\textit{-Clique}\}$$

Mit der Definition 5 haben wir ein Beispiel für das Cliquenproblem in der Entscheidungsvariante kennen gelernt.

2.3 Nichtdeterministischer Algorithmus und die Klasse NP

Wegener [Weg93] definiert das Verfahren des *nichtdeterministischen Algorithmus* wie folgt: Die Entscheidungsvariante des von uns betrachteten Optimierungsproblems CLIQUE hat eine Eigenschaft. “Wenn wir zufällig (durch ein Orakel ähnlich mysteriös wie das von Delphi, durch glückliches Raten oder wie auch immer) eine Lösung vorliegen haben, können wir sehr einfach und effizient verifizieren, dass die Lösung tatschlich den Anforderungen entspricht.

CLIQUE: Für eine Knotenmenge V' kann effizient überprüft werden, ob G eine Clique auf V' enthält und ob V' mindestens k Knoten enthält.”

Weiterhin entwirft Wegener ein dem Ratemechanismus zugehöriges Rechnermodell für Entscheidungsprobleme. Es heißt *nichtdeterministische Turingmaschine (NTM)* und ist definiert analog zur *deterministischen Turingmaschine (DTM)*.

Wegener definiert die folgenden beiden Begriffe:

Definition 6 *Es sei eine NTM M, die eine Sprache L akzeptiert, gegeben. Die Rechenzeit für eine Eingabe w ist, falls $w \in L$, gleich der Anzahl der Rechenschritte auf einem kürzesten akzeptierenden Rechenweg, und 0, falls $w \notin L$. Die* worst case Rechenzeit*, Notation $t_M(n)$, ist das Maximum der Rechenzeiten für alle Eingaben w der Länge n.*

Definition 7 *NP (nichtdeterministisch polynomiell) ist die Klasse der Entscheidungsprobleme, für die es eine nichtdeterministische Turingmaschine M gibt, deren worst case Rechenzeit polynomiell beschränkt ist.*

2.4 MAXSNP- und NP-Vollstndigkeit

MAXSNP- und *NP-vollständig* sind die Komplexitätsklassen für vergleichbar schwere Optimierungs- bzw. Entscheidungsprobleme. Zu den Definitionen und als Einführung in die Komplexitätstheorie siehe Papadimitriou [Pap94] und Wegener [Weg96].

Für die NP-Vollständigkeit wiederholen wir hier die entsprechenden Definitionen aus Wegener [Weg93]:

Definition 8 *Es seien L_1 und L_2 Sprachen über Σ_1 und Σ_2. Dann heißt L_1* polynomiell auf L_2 reduzierbar*, Notation $L_1 \leq_p L_2$, wenn es eine von einer DTM in polynomieller Zeit berechenbare Funktion $f : \Sigma_1^* \to \Sigma_2^*$ gibt, sodass für alle $w \in \Sigma_1^*$ gilt:*

$$w \in L_1 \Leftrightarrow f(w) \in L_2.$$

Definition 9
1. *Eine Sprache L heißt* NP-vollständig*, wenn $L \in NP$ ist und für alle $L' \in NP$ gilt: $L' \leq_p L$.*
2. *Eine Sprache L heißt* NP-hart*, wenn für alle $L' \in NP$ gilt: $L' \leq_p L$.*

Diese Definition ergibt, dass das Beispiel CLIQUE aus Abschnitt 2.2.3 NP-vollständig ist. Ein Beweis hierzu steht in Wegener [Weg93]. Jetzt sehen wir uns die Beziehung zwischen CLIQUE und CLIQUE (k) an:

Satz *CLIQUE $\leq_p$ CLIQUE (k)*

Beweis: z. z. es existiert ein polynomiell berechenbares f mit:

$$x \in CLIQUE \Leftrightarrow f(x) \in CLIQUE(k)$$

Seien $G' = G = (V, E), k, k' \in \mathbb{N}$ und $f' = f$ folgendermaßen:

1. Berechne CLIQUE für die Eingabe (G', k'), wenn das Ergebnis 0 ist, dann ist $x \notin CLIQUE$ und $f(x) := (\emptyset, \emptyset) \notin CLIQUE(k)$. Sonst ist das Ergebnis 1, und dann, für $e \in E$:

2. Berechne CLIQUE für Eingabe $((V, E/e), k')$, wenn das Ergebnis 0 ist, dann ist $x \notin CLIQUE$ und $f(x) := (\emptyset, \emptyset) \notin CLIQUE(k)$. Sonst ist das Ergebnis 1, und dann berechne 1. mit $(G', k') := ((V, E/e), k')$ und $f'(x) := (V, E/e)$

   Das Ergebnis ist entweder:

$$f(x) = f'(x) \in CLIQUE(k) \text{ oder } f(x) = (\emptyset, \emptyset) \notin CLIQUE(k).$$

Damit ist gezeigt worden, dass CLIQUE nicht schwieriger ist als CLIQUE (k).

3 Proteinfaltung

Der Übergang von der Aminosäuresequenz zum fertig gefalteten Protein wird die Proteinfaltung genannt. Unser Ziel in diesem Abschnitt ist es, dieses mathematisch zu modellieren und die Komplexität der angegeben Algorithmen abzuschätzen.

Die Mehrzahl der Aminosäuresequenzen kann keine biologische Bedeutung haben, weil sie nicht eindeutig dreidimensional gefaltet werden kann. Der Vorgang der Proteinfaltung findet in der Natur selbstständig statt.

3.1 Biologische Fragestellung

Es gibt die These, dass die *ab initio*[3] Vorhersage der Proteinfaltung möglich ist. Im Folgenden gehen wir davon aus, dass es so ist.

Von der natürlichen Proteinfaltung wird angenommen, dass sie den Zustand minimaler Energie findet. Es kann nicht ausgeschlossen werden, dass nur bestimmte lokale Minima erreicht werden. Die optimale Faltung ist das Ergeb-

[3] Vorhersage unter ausschließlicher Zuhilfenahme der Aminosäuresequenz

nis der Proteinfaltung und deren Vorhersage ist unser Ziel. Optimal gefaltete Sequenzen, die genau eine optimale Faltung besitzen, nennen wir “eindeutig” optimal gefaltet. In der Regel sind alle in der Biologie auftauchenden Aminosäureketten eindeutig optimal gefaltet. Denn nur so können sie im lebendigen Körper eine eindeutige Funktion übernehmen.

Die Vorhersage zur Proteinfaltung kann grundsätzlich nur in einem Modell stattfinden. Wir möchten zuerst ein Modell für die Struktur von Proteinen vorstellen: Das H-P-Modell, wobei H für hydrophob und P für polar steht. Es wird heute angenommen, dass nicht-lokale Gründe für die Faltung von Aminosäuresequenzen zu ihrer dreidimensionalen Struktur, dem so genannten nativen Zustand des Proteins, ausschlaggebend sind. Im H-P-Modell stellen wir Monomere resp. Aminosäureketten durch Sequenzen von ‘0’, nämlich die polaren oder hydrophilen Aminosäuren, bzw. ‘1’, nämlich die hydrophoben Aminosäuren, dar. Für in der Sequenz nicht benachbarte hydrophobe Monomere, die sich im Raum bzw. in der Ebene nebeneinander anordnen, werden wir eine Punktbewertung (Verluste) definieren. Das Problem, die Verluste im Modell für benachbarte hydrophobe Monomere, die keine Sequenznachbarn sind, zu minimieren, entspricht der realen Proteinfaltung.

Es gibt mathematische Beweise, dass das Proteinfaltungsproblem im zweidimensionalen und auch im dreidimensionalen H-P-Modell NP-vollständig ist. Auf das Proteinfaltungsproblem im zweidimensionalen Modell gehen wir in Abschnitt 3.3 näher ein. Ein Beweis für das dreidimensionale Modell wird in der Literatur von Berger et al. [BL98] geführt. Auch dort suchen wir, wie allgemein bei NP-vollständigen Problemen, eine Lösung in einem exponentiell großen Lösungsraum.

Das implizierte Paradoxon[4] ist, dass die Schwierigkeit der Vorhersage der Proteinfaltung in der Natur ständig bei der realen Proteinfaltung innerhalb weniger Sekunden oder Minuten gelöst wird.

Das menschliche Erbgut besteht aus DNA/DNS (Desoxyribonukleinsäure). DNA-Sequenzen bilden eine Doppelhelixstruktur. DNA wird mit Hilfe von verschiedenen Formen von RNA/RNS (Ribonukleinsäure) in Aminosäuresequenzen übersetzt. Wie bereits eingangs in diesem Kapitel erwähnt, nennt man den

[4] lt. Duden: scheinbar falsche Aussage, die aber auf eine höhere Wahrheit hinweist

Übergang von der Aminosäuresequenz zum fertigen Protein die Proteinfaltung. Das gefaltete Protein wird als sein ‘native state’ bezeichnet, und es hat eine eindeutige dreidimensionale Struktur. Eine eindeutige dreidimensionale Struktur haben viele Polymere wie DNA-, RNA- und, wie bemerkt, Aminosäuresequenzen resp. Proteine.

3.2 Ein mathematisches Hydrophob-Polar-Modell

Das H-P-Modell dient grundsätzlich zum Modellieren der Proteinfaltung im dreidimensionalen Raum. Verlassen wir die dritte Dimension und beschränken uns weiter auf konstante Abstände sowie rechtwinklige Anordnungen der Aminosäuren, so können wir uns hierfür ein Perlenmodell denken:

Mit einer Ebene, in der Mulden angeordnet sind, in die schwarze und weiße Perlen gelegt werden können, visualisieren wir das mathematische H-P-Modell. Die Perlen sind an einer Schnur befestigt, sodass Nachbarn immer nur über- oder nebeneinander angeordnet werden können (sowohl diagonale Nachbarschaften von zwei an der Schnur aufeinanderfolgenden Perlen als auch mehr als eine Kugel pro Mulde sind nicht möglich). Die Perlenkette repräsentiert eine Aminosäuresequenz mit hydrophoben (schwarze Kugeln, in Kapitel 1 bezeichnet als Eins) und hydrophilen (weiße Kugeln, in Kapitel 1 bezeichnet als Null) Sequenzelementen. Vgl. Abbildung 1.

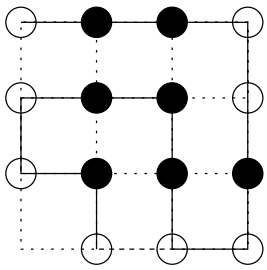


Abbildung 1: Anordnung einer Perlenkette im Gitter (Sequenznachbarn sind verbunden)

Wir beginnen nun mit der Definition des mathematischen H-P-Modells. Crescenzi et al. [CGP+98] definieren die folgenden Begriffe:

Ein *zweidimensionales Gitter* ist ein Graph $(\mathbb{Z}^2, L)$ mit der Knotenmenge $\mathbb{Z}^2 = \mathbb{Z} \times \mathbb{Z}$ (Punkte im Gitter) und den Kanten, die zwischen allen horizontal oder vertikal benachbarten Punkten im Gitter verlaufen und zur Kantenmenge $L = \{((x, y), (x', y')) : |x - x'| + |y - y'| = 1\}$ gehören.

Für eine Menge $S = \{s_1, s_2, \ldots, s_m\}$ mit $m \in \mathbb{N}$ von 0-1-Zeichenketten $s_1, s_2, \ldots, s_m \in \{0, 1\}^*$ ist eine *Faltung* dieser 0-1-Zeichenketten die folgende Abbildung f von S in das zweidimensionale Gitter:

$$f : \{(i, j) | 1 \leq i \leq m, 1 \leq j \leq |s_i|\} \to \mathbb{Z}^2$$

und es gilt für $\forall i, 1 \leq i \leq m : \forall j, 1 \leq j \leq |s_i| - 1 :$

$$(f(i, j), f(i, j + 1)) \in L.$$

Dabei werden Nachbarn in der 0-1-Zeichenkette s_i gefaltet zu *Gitternachbarn*.

Es folgt die *Bewertung* E einer Faltung der 0-1-Zeichenketten in S:

Gitternachbarn $((x, y), (x', y')) \in L$ sind ein *Verlust*, falls

1. sie keine Nachbarn in derselben 0-1-Zeichenkette sind und

2. genau einer der beiden Punkte eine 1 ist.

Das *Vielkettenproblem (VKP)* besteht darin, für eine Menge S von 0-1-Zeichenketten und eine Zahl E eine Faltung der 0-1-Zeichenketten in S mit E oder weniger Verlusten zu finden.

Das Proteinfaltungsproblem im zweidimensionalen H-P-Modell entspricht dem VKP, das auf eine 0-1-Zeichenkette ($|S| = 1$) beschränkt wird. Dieses Problem nennen wir *Einkettenproblem (1-KP)*. Das Beispiel in Abbildung 2 zeigt die zweidimensionale Einbettung einer 0-1-Zeichenkette mit $E = 4$ Verlusten.

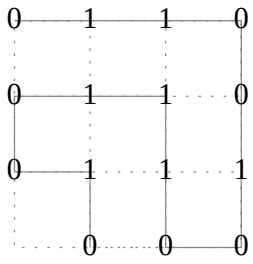


Abbildung 2: Anordnung einer 0-1-Zeichenkette im Gitter (Sequenznachbarn sind verbunden)

Das Muldenmodell diente der Veranschaulichung dieser Zeichenkette, wobei entsprechend unserer Ausführung Null durch die weißen und Eins durch die schwarzen Perlen ersetzt werden können.

3.3 Das Vielketten- und das Einkettenproblem sind NP-vollständig

In Crescenzi et al. [CGP+98] wird die NP-Härte des Vielketten- und des Einkettenproblems bewiesen. Wir wollen hier der Vollständigkeit halber eine kleine Beobachtung ergänzen, die dazu führt, dass das Vielketten- und das Einkettenproblem NP-vollständig sind. Dazu geben wir einen nichtdeterministischen Polynomialzeitalgorithmus für das 1-KP an:

Seien $\Sigma = \{0, 1\}$, $\Sigma' = \{\mathtt{N}, \mathtt{S}, \mathtt{W}, \mathtt{E}\}$, $s_i = S \in \{0, 1\}^n$, $n \in \mathbb{N}$ und $i = 1, \ldots, n$, dann

- rate nichtdeterministisch $x_1, \ldots, x_{n-1} \in \Sigma'$ (Richtungsänderungen der Sequenznachbarn)
- $f : S \to \mathbb{Z} \times \mathbb{Z}$, sei o. B. d. A. $f(s_1) = (0, 0)$. Wenn $f(s_i) = (x, y)$, dann

$$f(s_{i+1}) = \begin{cases} (x, y-1) & \text{, falls } x_i = \mathtt{S} \\ (x, y+1) & \text{, falls } x_i = \mathtt{N} \\ (x-1, y) & \text{, falls } x_i = \mathtt{W} \\ (x+1, y) & \text{, falls } x_i = \mathtt{E} \end{cases}$$

- verifiziere, dass f injektiv ist $(t \in O(n))$
- Berechne die Verluste für Nachbarfelder für $\forall s_i = 1$, sodass folgt $E \leq 2n + 2$ $(t \in O(n))$

Die Laufzeitangaben für die Zeit t in Klammern führen dazu, dass der Algorithmus linear ist. Analog verfahren wir beim Vielkettenproblem.

4 Alignment von Strukturen

Ebenso wichtig wie die Vorhersage der Proteinstruktur ist der Vergleich zweier gegebener Strukturen. Im Gegensatz zum Sequenzalignment, das mit der Sequenzinformation der Aminosäurekette arbeitet, benötigen wir diese beim Strukturalignment nicht, sondern wir modellieren Aminosäure Monomere als Knoten eines Graphen. Unser Ziel ist es, mit den folgenden Definitionen die Grundlagen für das Kapitel 5 zu legen.

4.1 Contact-Map-Graph

Nach Goldman [Gol00] entstehen Contact-Map-Graphen in der experimentellen Untersuchung der Abstände von Aminosäuren realer Proteine. Kontakt-Kanten entstehen zwischen Sequenzelementen, die einen gewissen Abstand unterschreiten.

Definition 10 *Eine* 'contact map' (n, E) *mit* $n \in \mathbb{N}$ *ist ein Graph* $G = (V, E)$ *mit:*

1. $V = \{1, 2, \ldots, n\}$ *— einer Menge von Knoten, die linear geordnet sind,*
2. E *— einer Menge von Kanten zwischen den Knoten,*
3. $\{i, i + 1\} \notin E$*, für* $i \in \{1, 2, \ldots, n - 1\}$ *— wobei benachbarte Knoten nicht in der Kantenmenge enthalten sind.*

Die Abbildung 3 zeigt den Contact-Map-Graphen zur Sequenz aus dem vorherigen Beispiel. Die Knotenmenge ist jetzt linear geordnet, und darüber befinden sich die Kontakt-Kanten, die wir mit gestrichelten Bogen darstellen.

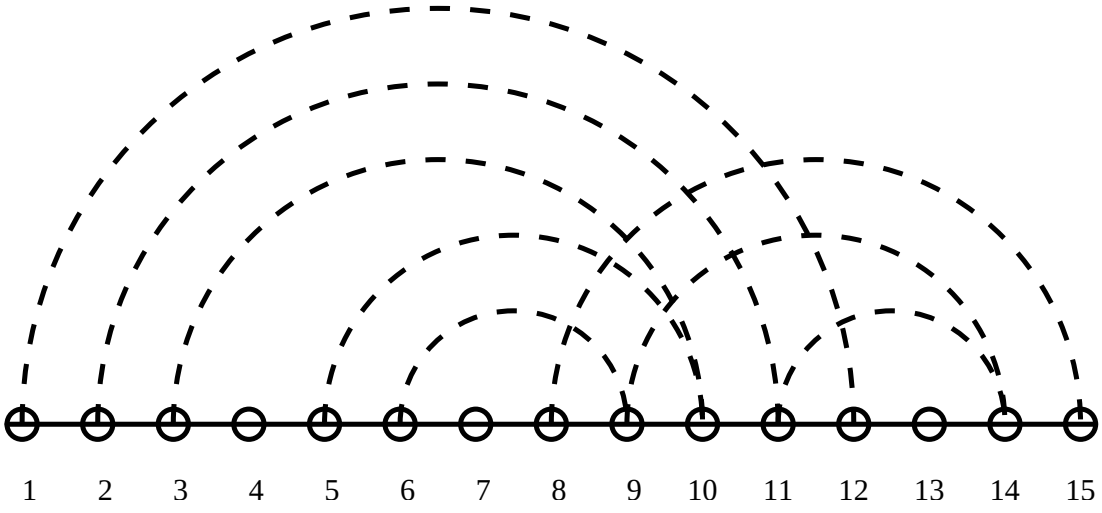


Abbildung 3: Contact-Map-Graph

Wir geben jetzt mit eigenen Worten wieder, wie Goldman [Gol00] die unten verwendete Kantennotation erklärt: Wir werden die Notation $[i, j]$ verwenden für eine Kante zwischen Paaren von Knoten $i, j \in \{1, \ldots, n\}$, mit dem Verständnis, dass $i < j$ ist. Diese Kantennotation soll ein Intervall suggerieren. Es wird manchmal nützlich sein, Contact-Map-Kanten als Intervalle zu repräsentieren.

Definition 11 *Das* 'Contact map overlap problem' (CMOP) *ist wie folgt definiert: Gegeben sind zwei 'contact maps'* (n, E) *und* (m, E'). *Finde Teilmengen mit* $|S| = |S'|$, *wobei gilt:*

$$S \subseteq \{1, \ldots, n\}, S' \subseteq \{1, \ldots, m\},$$

sodass die folgende Kardinalität c *größtmglich ist und* f *eine ordnungserhaltende (streng monoton steigende) Bijektion zwischen* S *und* S' *ist.*

$$c = |\{[u, v] \in E : u, v \in S, [f(u), f(v)] \in E'\}|$$

Die Abbildung 4 zeigt zwei weitere Beispiele für Contact-Map-Graphen. Die gesuchten Knoten der Teilmenge S werden in der Zeichnung mit Pfeilen auf die gesuchten Knoten der Teilmenge S' abgebildet. Die Knotenmengen sind wieder linear geordnet und darüber befinden sich die Kontakt-Kanten, wobei Kanten, die in beiden Graphen von aufeinander abgebildeten Knoten ausgehen, dick eingezeichnet sind. Die Anzahl der so überlappenden Kanten wird als 'contact overlap' bezeichnet und in Abbildung 4 ist $c = 4$.

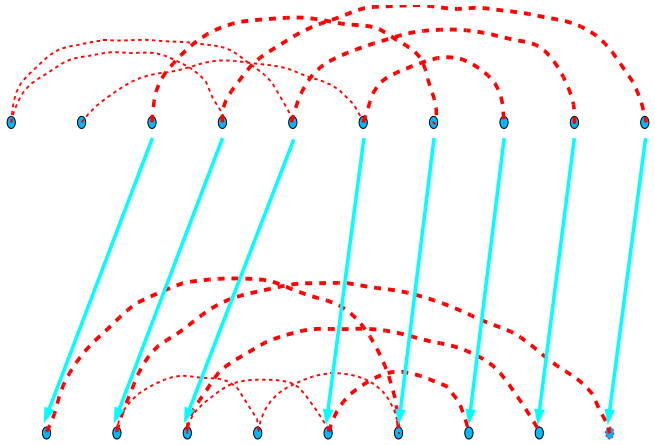


Abbildung 4: 'Contact map overlap problem' mit $c = 4$, die oberen Knoten sind S und die Unteren S'

In Goldman et al. [GIP99] steht der Satz:

Satz *Das 'contact map overlap problem' für Graphen mit maximalem Grad eins ist MAXSNP-vollständig.*

Ein Beweis wurde nicht angegeben. Wir sind der Ansicht, dass wir den Satz nicht aus Goldman [Gol00] folgern können.

4.2 Walk im zweidimensionalen Gitter

Wir modellieren, wie in Kapitel 3.2 erwähnt, eine Perlenschnur in einer Ebene voll Mulden. Eine Einbettung der Perlenschnur werden wir ‘walk’ oder synonym ‘Faltung’ nennen.

Definition 12 *Sei f eine injektive Abbildung von $\{1, 2, \ldots, n\}$ nach $\mathbb{Z}^2$ und es gelte: $||f(i) - f(i+1)||_2 = 1$, für $i = 1, \ldots, n-1$.*

Die Abbildung f heißt ‘walk’ *im zweidimensionalen Gitter. Synonym verwenden wir für den Begriff ‘walk’* Faltung.

4.3 Self-Avoiding Walk

Mit jedem solchen ‘walk’ f wird seine ‘contact map’ verknüpft: $G_f = (\{1, 2, \ldots, n\}, E)$, wobei:

$$E = \{[i, j] : |i - j| > 1, ||f(i) - f(j)||_2 = 1\}$$

Der Contact-Map-Graph heißt *‘self-avoiding walk’* und hat maximalen Grad zwei (ausgenommen die Knoten 1 und n, die Grad drei haben können).

Satz 1 *Das ‘contact map overlap problem’ ist NP-vollständig, sogar wenn beide ‘contact maps’ ‘self-avoiding walks’ sind.*

Beweis: In Goldman [Gol00] wird die NP-Härte bewiesen. Wie in Kapitel 3.3 geben wir auch hier einen nichtdeterministischen Polynomialzeitalgorithmus für das CMOP an:

Gegeben zwei Contact-Map-Graphen $G = (n, E)$ und $G' = (m, E')$ mit;

$$\{a_1 = 1, \ldots, a_n = n\}, \{b_1 = 1, \ldots, b_m = m\}$$

- rate nichtdeterministisch $x_1, \ldots, x_n \in \{0, 1\}$ und $y_1, \ldots, y_m \in \{0, 1\}$ (Teilmengen S und S' der Knotenmengen n und m)
- es existiert genau eine streng monoton steigende Bijektion f:

$$f(a_i) = b_j \wedge x_i = y_j = 1, (1 \leq i \leq n, 1 \leq j \leq m)$$

- verifiziere, dass $|S| = |S'|$:

$$|\{x_i | x_i = 1\}| = |\{y_j | y_j = 1\}|, (1 \leq i \leq n, 1 \leq j \leq m)$$

- verifiziere, dass für $x_l = x_k = 1, (1 \leq l, k \leq n)$:

$$[a_k, a_l] \in E \Leftrightarrow [f(a_k), f(a_l)] \in E'$$

  (benachbarte Knoten in G werden auf benachbarte Knoten in G' abgebildet)

- Zeitaufwand: $t = O(n + m) + O(|E| + |E'|) = O(n + m)$
  Dabei können die Größen der Kantenmengen abgeschätzt werden durch die Zahl der Knoten, weil die Graphen konstanten Grad haben. Die Zeit t ist somit polynomiell in der Größe der Graphen G und G'.

4.4 Strukturalignment vs. Sequenzalignment

Das *'Protein Threading Problem' (PTP)* hat als Eingabe eine Aminosäuresequenz über dem Alphabet Σ der 20 verschiedenen Aminosäuren, eine dreidimensionale Proteinstruktur in Form eines Contact-Map-Graphen G' inklusive Sequenzinformation und eine Bewertungsfunktion. Die Aufgabe besteht darin, die gegebene Aminosäuresequenz mit dem Contact-Map-Graphen bestmöglich zu alignieren.

Im Folgenden vergleiche Abbildung 5. Nimmt man als Bewertungsfunktion die Verlustefunktion E und transformiert man das Alphabet Σ in das Alphabet Σ' des H-P-Modells ($\Sigma' = \{0, 1\}$), dann entspricht das PTP dem Lösen der folgenden Probleme (s. Abbildung 5):

- löse das 1-KP mit Eingabe über dem Alphabet Σ' und den Kanten E,
- generiere aus der 2D-/3D-Struktur den Contact-Map-Graphen G,

- löse das CMOP mit Eingabe G und G'.

Das PTP ist MAXSNP-hart nach Akutsu und Miyano [AM97]. Ein dem PTP entsprechendes Problem ist NP-hart nach Lathrop [Lat94].

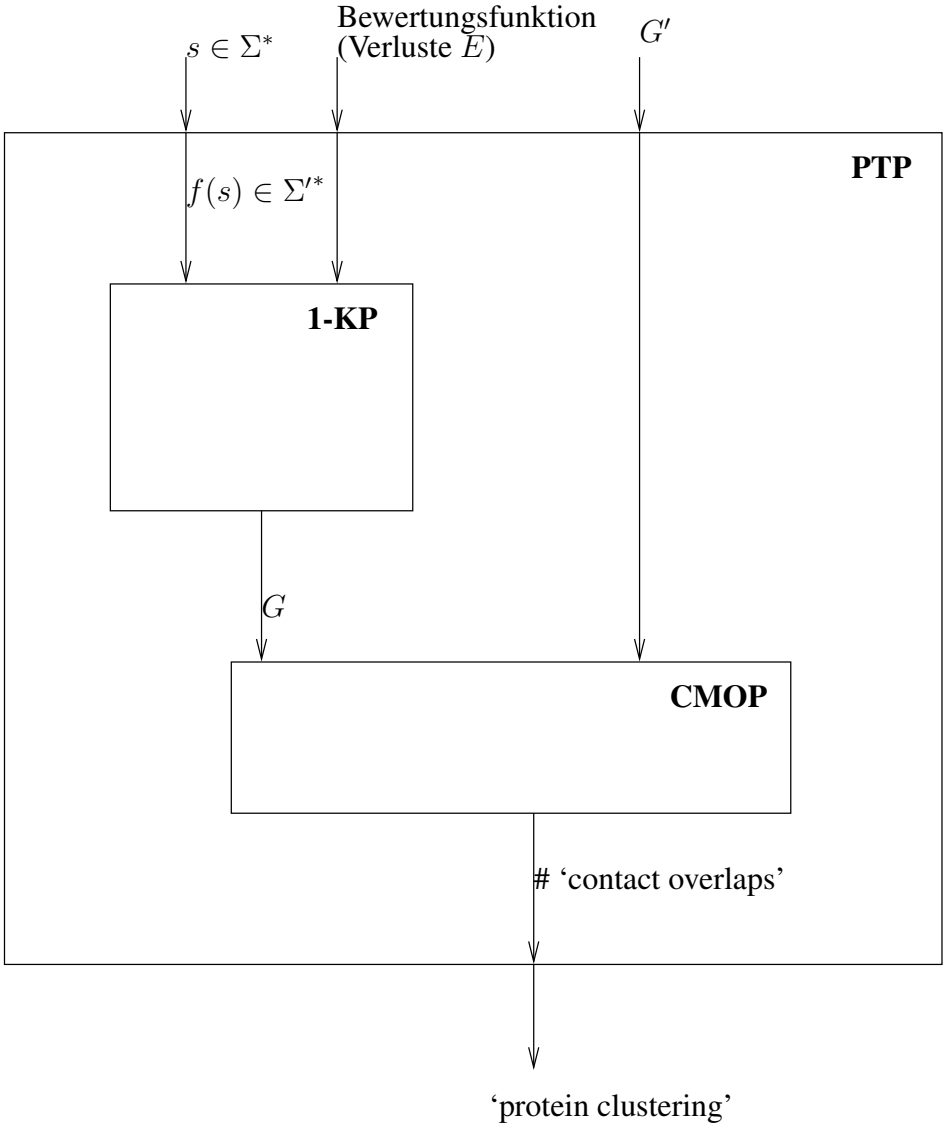


Abbildung 5: 'Protein Clustering', wie in den Abschnitten 4.4 und 4.5 erklärt

Anstatt zwei Strukturen beim Strukturalignment oder eine Sequenz mit einer Struktur beim PTP zu vergleichen, können wir Sequenzalignments mit zwei Aminosäureketten betrachten. Mit den Algorithmen für das Sequenzalignment

können wir Sequenzähnlichkeiten finden, die dann auch zu ähnlichen Strukturen und Funktionen führen. Umgekehrt lassen sich bei ähnlichen Strukturen und Funktionen keine Aussagen ber Ähnlichkeiten in der Sequenz machen.

4.5 Anwendung des Strukturalignments in der Phylogenie

Chotia [Cho92] beschreibt die Möglichkeit, dass es 1000 verschiedene Familien oder 'protein cluster' gibt. Die Mitglieder eines 'protein cluster's könnten von einem gemeinsamen Vorfahren abstammen, was in der Phylogenie bestimmt wird, und sich zu ähnlichen Strukturen falten; in der Regel haben sie dann auch ähnliche Sequenzen und Funktionen. Während Phylogenie bisher auf Vergleich von Sequenzen beruht, was Sequenzalignment genannt wird, bieten Contact-Map-Graphen die Möglichkeit des Vergleichens von Strukturen, was Strukturalignment genannt wird. Dies geschieht unter der Annahme, dass Proteine mit ähnlichen Funktionen gemeinsame Vorfahren haben und dass die Struktur von der Funktion bestimmt wird ('form follows function').

5 Strukturen im H-P- und Graphmodell

Beispiele von H-P-Sequenzen setzen wir in Beziehung mit ihren 'walk's und 'self-avoiding walk's. Diese werden wir als Strukturen genauer untersuchen. Im Folgenden stellen wir Beispiele für Anordnungen von Sequenzen im zweidimensionalen H-P-Modell vor, die wir Spiral- und Faltsequenzen nennen werden. Wir finden diese Sequenzen als Strukturen in natürlichen Proteinen wieder mit den Namen α-Helix bzw. β-Faltblatt.

Ein kombinatorisch interessantes Ergebnis ist, dass eine Sequenz der Form $1 - \ldots - 1$ und der Länge $n = m^2$ sowohl als Spiral- als auch als Faltsequenz mit minimalen Verlusten $E = 4m$ darstellbar ist. Hierbei nimmt die Sequenz die Form eines Quadrates mit Seitenlnge m an, und das Verhältnis der Oberfläche $4m$ ist minimal im Vergleich zur Quadratflche m^2. Diese nicht eindeutige Faltung im zweidimensionalen H-P-Modell widerspricht der Annahme, dass α-Helix bzw. β-Faltblatt (Sekundrstruktur eines Proteins) nötige und eindeutige Zustände innerhalb des gesamten Vorganges der Proteinfaltung darstellen. An-

dererseits ist bekannt, dass nur reale Proteine vorkommen, die eine eindeutige Tertiärstruktur ausbilden. Die beispielhaften H-P-Sequenzen sind daher nicht mit natürlich vorkommenden Proteinen zu verwechseln.

5.1 Spiral- und Faltsequenz

Eine *Spiralsequenz* ist eine Sequenz der Form $1-\ldots-1-0-\ldots-0$, deren Elemente in einer Richtung um das platzierte Startelement 1 herumgelegt werden. Die Abbildung 6 zeigt ein Beispiel für eine dieser möglichen Spiralsequenzen.

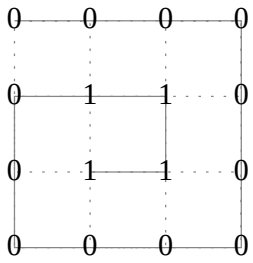


Abbildung 6: Spiralfrmige Anordnung einer Zeichenkette der Form $1-\ldots-1-0-\ldots-0$ mit $E = 7$ Verlusten

Aus dieser so definierten Struktur der Spiralsequenz ergeben sich direkt Formeln für:

- die Anzahl der Einsen $o(n)$ (‘one’) in einer Spiralsequenz,
- die Anzahl der Nullen $z(n)$ (‘zero’) am Ende einer Spiralsequenz,
- die Anzahl der sich ergebenden Verluste $E(n)$ dieser Spiralsequenz und
- die Innenfläche $In(n) := o(n)$ (entspricht der Anzahl der Einsen)
- das Verhältnis $r(n)$ (‘relation’) der Verluste $E(n)$ zu der Anzahl der Einsen der Innenfläche $In(n)$.

Crescenzi et al. [CGP+98] erwähnen, dass das Verhältnis $r(n)$ dem Verhältnis der Oberfläche zum Inhalt von realen Proteinen nahe kommt.

Es folgt der Quellcode von Maple[5] für die Berechnungen der einzelnen Formeln

[5] ©Maplesoft, a division of Waterloo Maple Inc. 2003. All rights reserved.

mit graphischer Veranschaulichung.

Der Quellcode benutzt die folgenden typographischen Konventionen, sinngemäß aus dem Maple Handbuch wiedergegeben:

- `courier` font — Maple Kommando, Paketname und Optionsname
- *italics* — Ausgaben von Maple

Ein *Semikolon* (;) bezeichnet das Ende von jeder Berechnung. Funktionen definieren wir durch die Maple *Pfeilnotation* (->). Diese Notation erlaubt uns, Funktionen zu evaluieren, wenn sie in einem Maple Ausdruck ('expression') erscheinen. Wir können einfache Graphiken einer Funktion erzeugen, indem wir das `plot` Kommando verwenden.

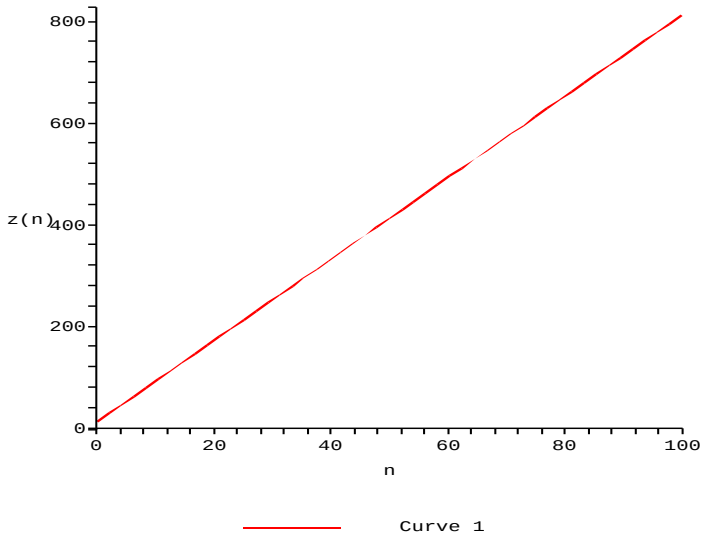
```
> rsolve(o(n)=o(n-1)+(8*n)+4,o(0)=4,o(n));
```

$$\{o(n) = -4 + 8\,(n+1)\,(1/2\,n+1) - 4\,n\}$$

```
> z:=n->(8*n)+12;
```

$$z := n \mapsto 8\,n + 12$$

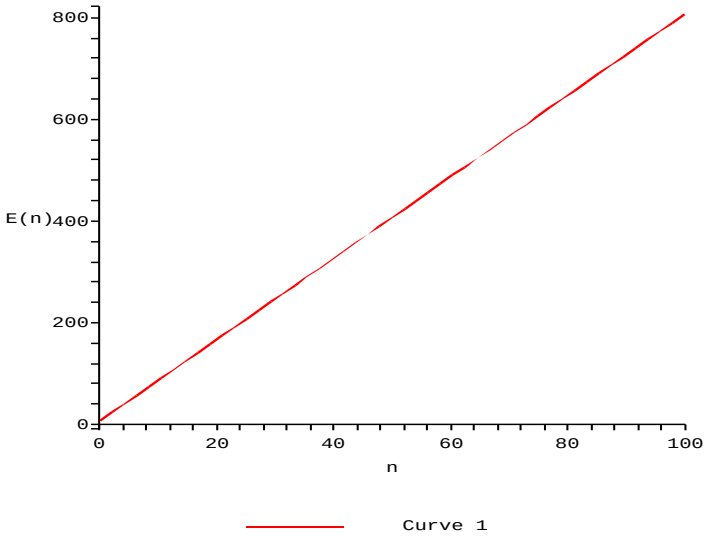
```
> plot (z(n),n=0..100,labels=["n","z(n)"]);
```




```
> E:=n->8*n+7;
```

$$n \mapsto 8\,n + 7 := n \mapsto 8\,n + 7$$

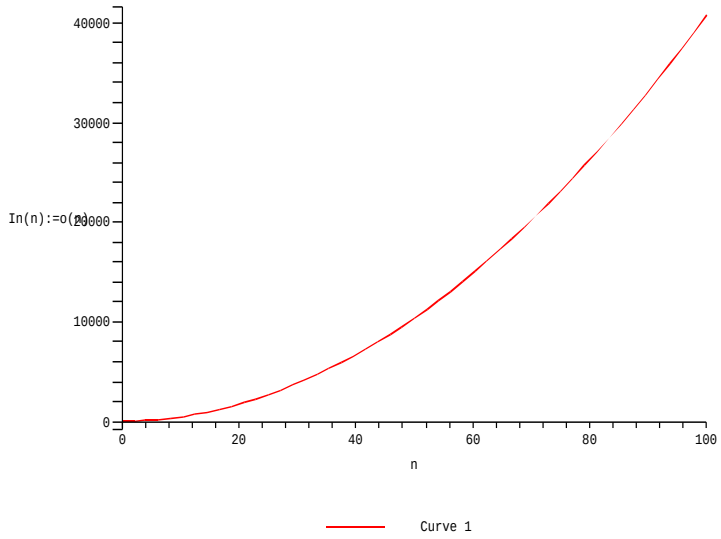
```
> plot (E(n), n=0..100,labels=["n","E(n)"]);
```




```
> In:=n->-4+8*(n+1)*((1/2)*n+1)-4*n;
```

$$n \mapsto -4 + 8\,(n+1)\,(1/2\,n+1) - 4\,n := n \mapsto -4 + 8\,(n+1)\,(1/2\,n+1) - 4\,n$$

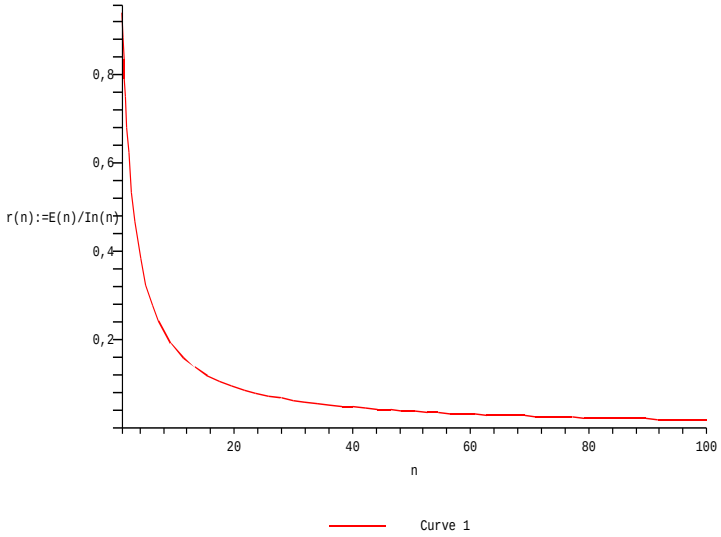
```
> plot(In(n),n=0..100,labels=["n","In(n):=o(n)"]);
```



```
> r:=n->E(n)/In(n);
```

$$n \mapsto \frac{8\,n+7}{-4+8\,(n+1)(1/2\,n+1)-4\,n} := n \mapsto \frac{8\,n+7}{-4+8\,(n+1)(1/2\,n+1)-4\,n}$$

```
> plot(r(n),n=1..100,labels=["n","r(n):=E(n)/In(n)"]);
```



Eine *Faltsequenz* ist eine Sequenz der Form $1 - \ldots - 1$ und der Länge m^2 $(m \in \mathbb{N})$, deren gerade Lagen, die sich in der Richtung abwechseln, die Form eines Quadrates annehmen (vgl. Abbildung 7). Die Abbildung 7 zeigt ein Beispiel für eine dieser möglichen Faltsequenzen.

Aus der so definierten Struktur der Faltsequenz ergeben sich direkt Formeln für:

- die Anzahl der Einsen $o(m) = m^2$ ('one') in einer Faltsequenz,
- die Anzahl der sich ergebenden Verluste $E(m) = 4m$ dieser Faltsequenz.

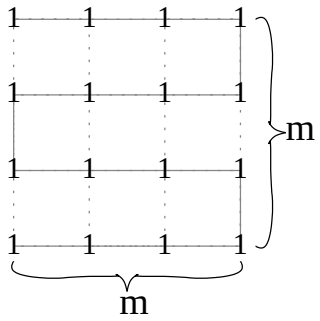


Abbildung 7: Gefaltete Anordnung einer Zeichenkette der Form $1 - \ldots - 1$ mit $m = 4$ und $E(m) = 16$ Verlusten

An dieser Stelle bemerken wir, dass die Sequenz der Form $1 - \ldots - 1$ und der Länge m^2 keine eindeutige gefaltete Anordnung mit minimalen Verlusten hat. Z. B. hat für $m = 4$ die Spiralsequenz mit $o(1) = 16$ Einsen auch genau $E = 8 * 1 + 8 = 16$ Verluste.

Im Folgenden geben wir eine geschlossene, an reguläre Ausdrücke angelehnte Form für eine so genannte *gefaltete* Struktur mit $E = 0$ Verlusten an. Hierbei heißt $(1)^i$ die aus i 1en bestehende Zeichen(-folge) und $(1)^*$ sind die keinmal oder beliebig, aber endlich, oft wiederholte Zeichen(-folge) aus 1en. Die Formel der Sequenz mit $i \in \mathbb{N}_0$ ist:

$$(01001)(1001)^i(10010)((011)(11)^i(110))^*(01001)(1001)^i(10010) \qquad (1)$$

Die Anordnung dieser Sequenz, wiedergegeben in Formel (1) mit $i = 0$, ist in der Ebene in Abbildung 8 zu sehen.

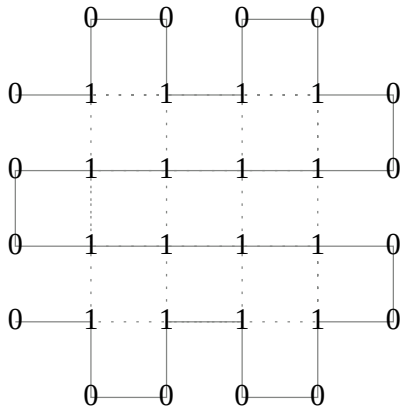


Abbildung 8: Gefaltete Anordnung einer Zeichenkette mit $E = 0$ Verlusten für $i = 0$

In Abbildung 9 sehen wir die generische Eigenschaft der gefalteten Struktur, die zu der Formel (1) gehört.

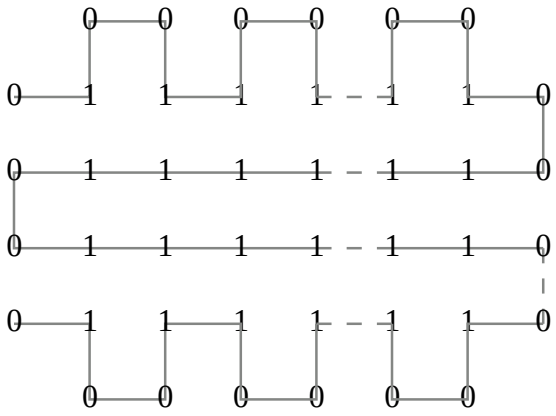


Abbildung 9: Gefaltete Anordnungen der generischen Zeichenkette mit $E = 0$ Verlusten

Im Folgenden geben wir eine geschlossene, an reguläre Ausdrücke angelehnte Form für eine so genannte *spiralförmige* Struktur mit $E = 1$ Verlust an. Die Formel der Sequenz mit $m \in \mathbb{N}$ und m gerade ist:

$$(1)^{m^2-4m+4}(1001)^{(m-2)/2}(10)(0110)^{(m-2)/2}(010)(0110)^{(m-2)/2}(010)(0110)^{(m-2)/2}(010) \tag{2}$$

Die Anordnung dieser Sequenz (s. Formel (2)) in der Ebene ist in Abbildung 10, für $m = 2$, und Abbildung 11, für $m = 4$, zu sehen.

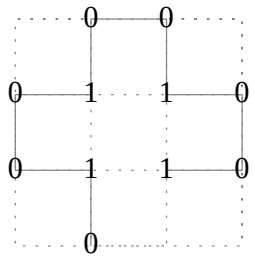


Abbildung 10: Spiralförmige Anordnung einer Zeichenkette mit $E = 1$ Verlust für $m = 2$

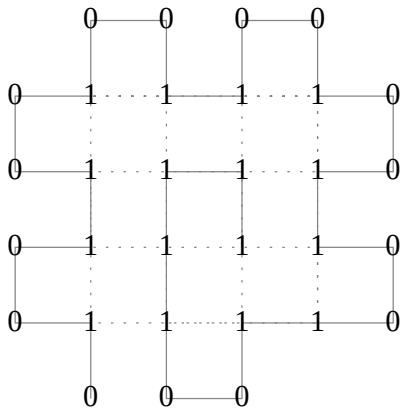


Abbildung 11: Spiralförmige Anordnung einer Zeichenkette mit $E = 1$ Verlust für $m = 4$

Bei der Graphik in Abbildung 11 handelt es sich um eine möglicherweise nicht eindeutige, aber optimale Faltung. Im Kern der Graphik befinden sich vier Einsen, die fortlaufend im Uhrzeigersinn miteinander verbunden sind. Diese Umlaufrichtung könnte auch gegen den Uhrzeigersinn laufen. Hierbei bleiben Sequenz und minimale Verluste erhalten. Wieviele minimale Lösungen die Se-

quenz besitzt, ist dem Autor nicht bekannt. Auf Eindeutigkeit von Faltungen gehen wir im folgenden Abschnitt ein.

5.2 Eindeutig optimale Faltung

Zur Präsentation eines eindeutig optimal gefalteten Beispiels in diesem Abschnitt benötigen wir folgenden Satz:

Satz 2 n $(n \in \mathbb{N})$ *horizontal benachbarte Einsen haben genau* $E = 2n + 2$ *Verluste*

Beweis: Die Verluste treten an der umgebenden Oberfläche der Einsen auf. Oberhalb und unterhalb befinden sich jeweils n leere Felder. Rechts und links befindet sich jeweils ein leeres Feld. Daraus ergeben sich $E = 2n + 2$ Verluste.

Zur Veranschaulichung der eindeutig optimalen Faltung folgen zwei Graphiken. In Abbildung 12 sehen wir alle vier Möglichkeiten, vier benachbarte Einsen so mit Nullen zu einer Sequenz zu verbinden, dass mindestens zwei Verluste entstehen. Die Formel für die Verluste ist $2n + 2 - 2n$ (n: Anzahl der benachbarten Einsen). Hierbei handelt es sich um die Formel aus Satz 2, von der die Grade aller Knoten $2n$ abgezogen werden.

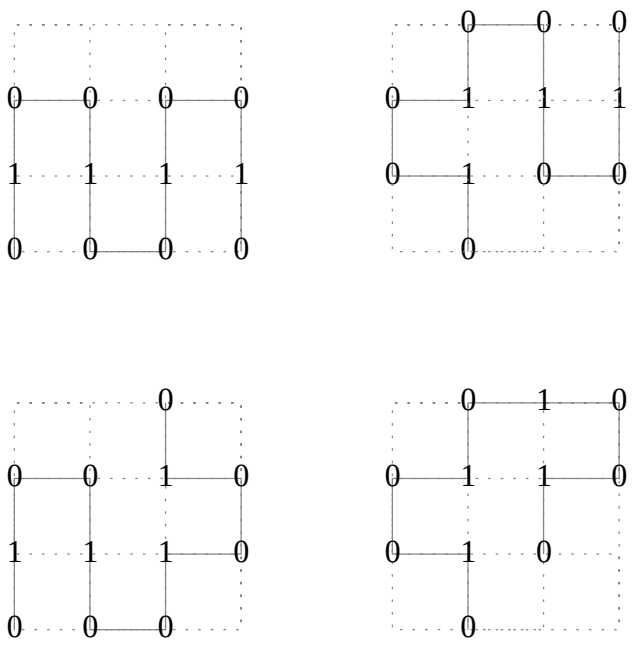


Abbildung 12: Nicht optimale Faltung mit $E = 2$ Verlusten mit vier benachbarten Einsen

Im Folgenden vergleiche Abbildung 13. Aus den vier Möglichkeiten in Abbildung 12 ergibt sich die fünfte Möglichkeit, vier benachbarte Einsen ohne entstehende Verluste zu verbinden, welches die eindeutig optimale Lösung ist.

Für vier quadratisch angeordnete Einsen gilt: Die Anzahl der Verluste ist mit $n = 4$:

$$E = 4\sqrt{n} = 8$$

Durch acht Kanten an den Stellen, an denen die Verluste auftreten, und acht Nullen an den Kantenenden werden die Verluste minimiert ($E = 0$). Zuletzt werden die Nullen durch drei Kanten zur eindeutig optimal gefalteten Sequenz $0-1-0-0-1-0-0-1-0-0-1-0$ verbunden.

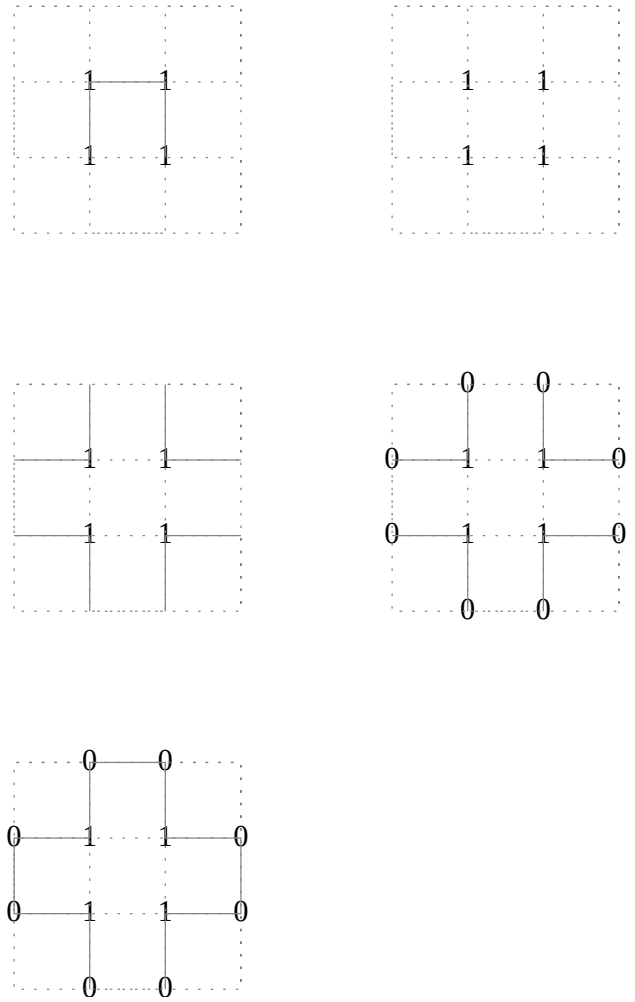


Abbildung 13: Beweis der eindeutig optimalen Faltung mit $E = 0$ Verlusten

Aichholzer et al. [ABD+03] untersuchen sämtliche Strukturen im H-P-Modell bis zu einer bestimmten Länge der Sequenz auf ihre Eindeutigkeit. Sie geben an, ein Programm geschrieben zu haben, das diese Strukturen berechnet.

5.3 Queue, Stack und 2-Stack

Im vorherigen Abschnitt haben wir bestimmte Sequenzen im zweidimensionalen H-P-Modell angegeben. Jetzt werden wir die von den Sequenzen erzeugten Strukturen im Graphmodell betrachten.

Komplexittstheoretisch interessant sind die Ergebnisse von Goldman [Gol00]

für die Spezialfälle des CMOP für alle drei folgenden Modellstrukturen. Die Ergebnisse beziehen sich auf *'self-avoiding walks'* für das zweidimensionale H-P-Modell. Die betrachteten Modellstrukturen sind daher Graphen vom Grad zwei, bis auf den ersten und letzten Knoten, die vom Grad drei sein können. Die polynomiellen Laufzeiten, die von Goldman gezeigt wurden, gelten sogar für Graphen mit konstantem Grad, was *'contact maps'* von realen Proteinen entspricht.

In den folgenden Definitionen 13 bis 15 werden die hier verwendeten Graphen und Bezeichnungen eingeführt.

Die Definitionen der verwendeten Graphen und Bezeichnungen *'queue'*, Contact-Map-Graph mit überlappenden Kanten, *'stack'*, Contact-Map-Graph mit verschachtelten Kanten, und *'2-stack'*, Contact-Map-Graph mit zwei *'stack's*, folgen in diesem Abschnitt. Die *'staircase'*, Contact-Map-Graph mit kompliziert überlappenden Kanten, wird in Kapitel 6 definiert.

Es folgen die komplexitätstheoretischen Ergebnisse:

- für mindestens einen *'stack'* und eine *'contact map'* ist das CMOP in P,
- für mindestens eine *'staircase'* und eine *'contact map'* ist das CMOP in P und
- für zwei *'2-stack's* ist das CMOP NP-vollständig, was aus dem Beweis zu Satz 1 folgt.

Für zwei *'queue's* vermuten wir die NP-Vollständigkeit des CMOPs.

Definition 13 *Eine* 'queue' *ist eine 'contact map'* (n, E)*, sodass für* $1 \leq i < j \leq n, 1 \leq k < l \leq n$ *und* $[i, j], [k, l] \in E$ *gilt:*

$[i, j]$ *und* $[k, l]$ *sind nicht ineinander enthalten, außer sie haben einen gemeinsamen Knoten* $i = k$ *oder* $j = l$*.*

Zur graphischen Verdeutlichung dieser Definition folgt in Abbildung 14 ein Beispiel für einen Contact-Map-Graphen 'queue'. Die 'queue' zeichnet sich dadurch aus, dass sich Enden der Contact-Map-Kanten überlappen können.

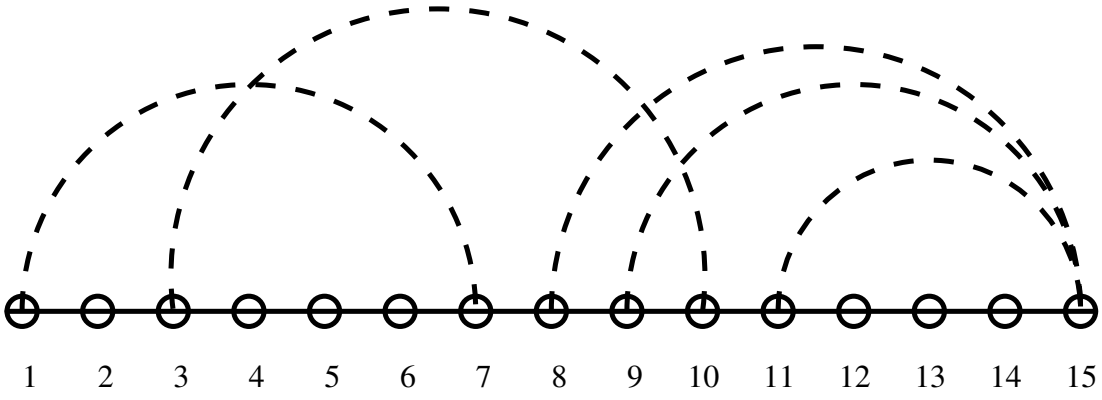


Abbildung 14: Contact-Map-Graph ‘queue’

Der zur Spiralsequenz in Abbildung 6 gehörige Contact-Map-Graph ist eine ‘queue’.

Definition 14 *Ein* ‘stack’ *ist eine ‘contact map’* (n, E)*, sodass für* $1 \leq i < j \leq n, 1 \leq k < l \leq n$ *und* $[i, j], [k, l] \in E$ *gilt:*

$[i, j]$ *und* $[k, l]$

- *sind ineinander enthalten,*
- *sind disjunkt oder*
- *haben einen gemeinsamen Knoten.*

Ein Beispiel für einen Contact-Map-Graphen ‘stack’ in Abbildung 15 zeigt, dass Contact-Map-Kanten nicht überlappen, sondern ineinander verschachtelt sind.

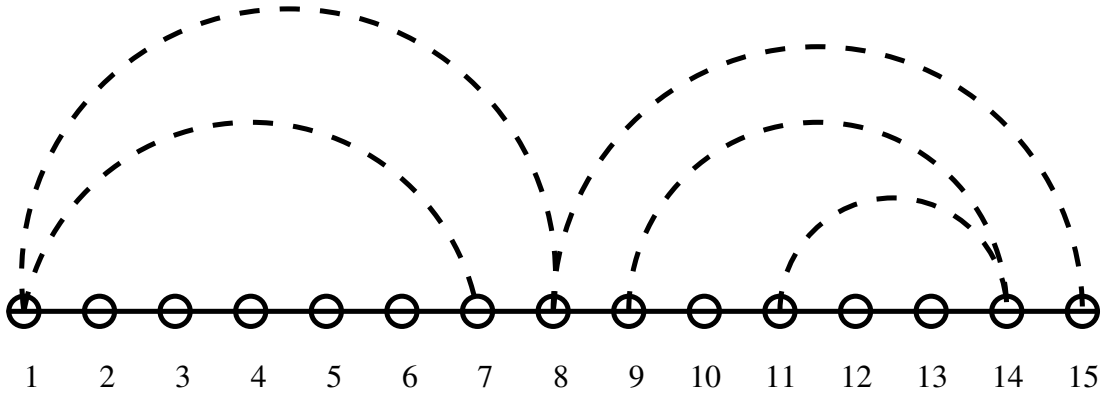


Abbildung 15: Contact-Map-Graph ‘stack’

Es folgt eine Definition, die sowohl Goldman et al. [GIP99], als auch Goldman [Gol00] auslassen:

Definition 15 *Ein* ‘2-stack’ *ist eine ‘contact map’* (n, E)*, die sich in der Ebene so zeichnen lässt, dass die Kanten in* E *oberhalb und unterhalb der linear geordneten Knotenmenge* n *jeweils einen ‘stack’ bilden.*

Wie wir in Abbildung 16 sehen, kann man den Contact-Map-Graphen ‘2-stack’ so in der Ebene zeichnen, dass sich ober- und unterhalb des Graphen jeweils ein ‘stack’ befindet.

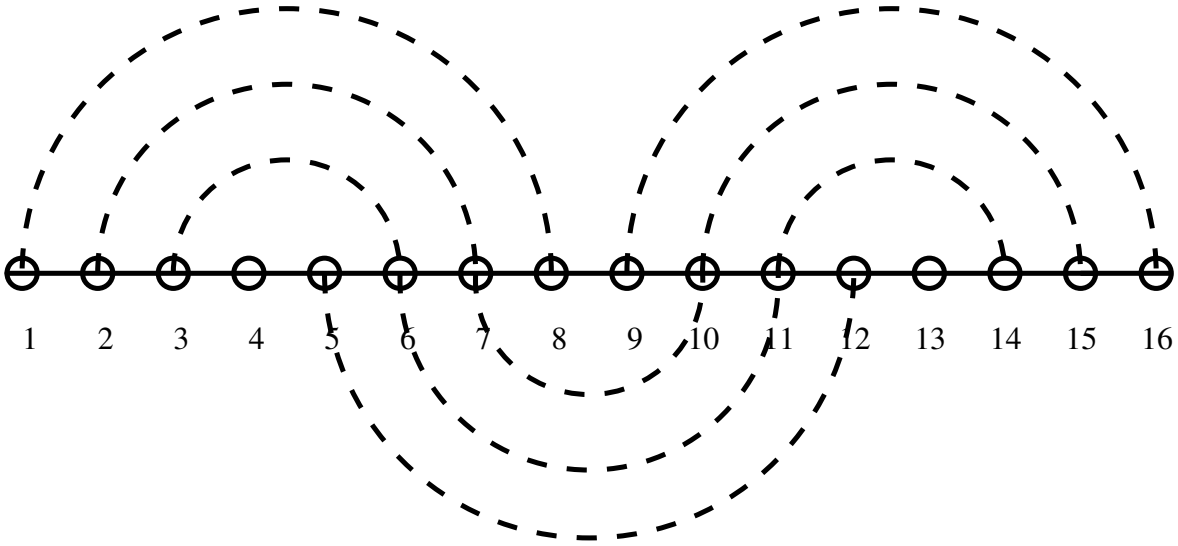


Abbildung 16: Contact-Map-Graph ‘2-stack’ zur Faltsequenz aus Abbildung 7

Der zur Faltsequenz aus Abbildung 7 gehörige Contact-Map-Graph ist der ‘2-stack’ aus Abbildung 16.

Die Beispiele in Abbildung 17 und Abbildung 18 verdeutlichen noch einmal die in sich geschachtelten Contact-Map-Kanten eines ‘2-stack’ und ‘stack’.

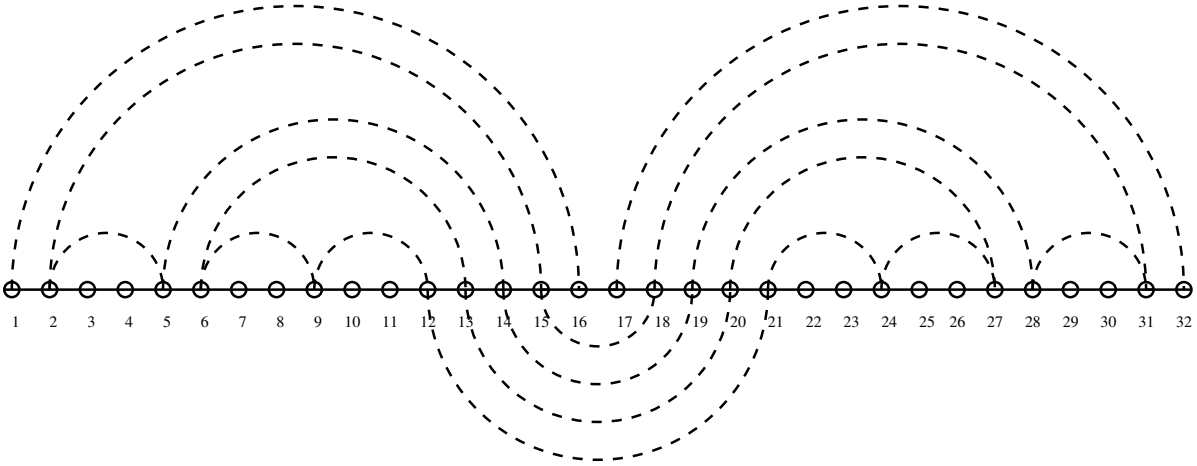


Abbildung 17: Contact-Map-Graph ‘2-stack’ zum ‘walk’ aus Abbildung 8

Der zum ‘walk’ aus Abbildung 8 geöhrige Contact-Map-Graph ist der ‘2-stack’ aus Abbildung 17.

Dieser Contact-Map-Graph ist ein ‘2-stack’, weil in der Zeichnung 17 die Kanten oberhalb und unterhalb der Knoten je einen ‘stack’ bilden.

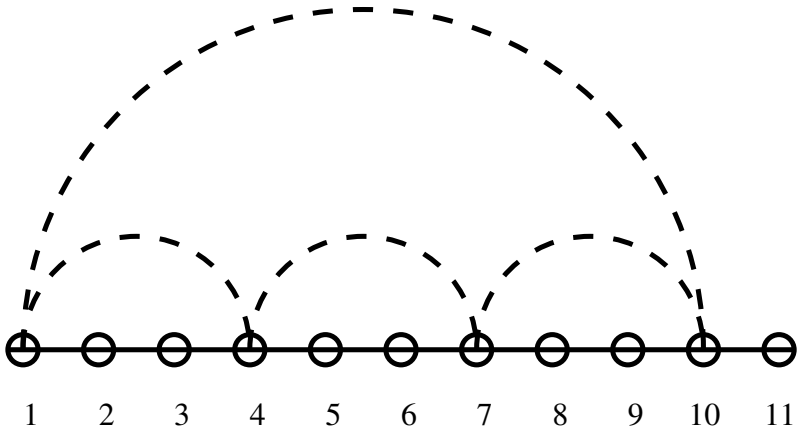


Abbildung 18: Contact-Map-Graph ‘stack’ zum ‘walk’ aus Abbildung 10

Der zum ‘walk’ aus Abbildung 10 gehörige Contact-Map-Graph ‘stack’ ist in Abbildung 18 wiedergegeben. Zu beachten ist hier, dass für die Spiralsequenz aus Abbildung 10 und die Faltsequenz aus Abbildung 13 die Modellstruktur ein ‘stack’ ist, der keine ‘queue’ ist.

Die genaue Definition von ‘stack’ und ‘2-stack’ ist in Heath et al. [HI92] gegeben. Dort wird für Graphen ein ‘book embedding’ beschrieben.

Am Beispiel der Abbildung 19 sehen wir einen komplizierten Contact-Map-Graphen, der zum ‘walk’ aus Abbildung 11 gehört.

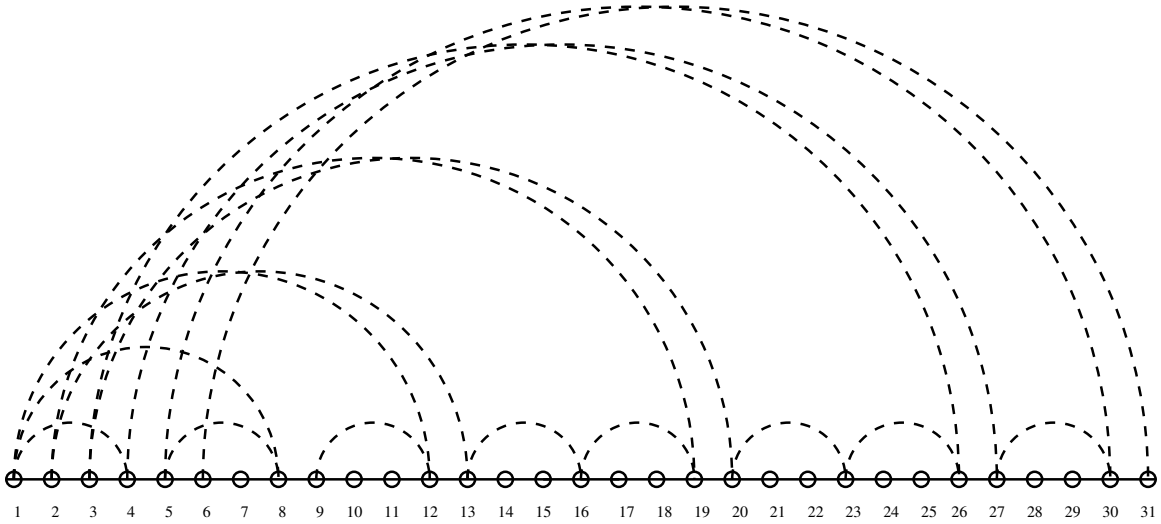


Abbildung 19: Contact-Map-Graph zum ‘walk’ aus Abbildung 11

5.4 Die Schnittmenge von Queue und Stack

In der Abbildung 20 sehen wir Beispiele für Strukturen, deren Modellstrukturen sowohl der ‘stack’ als auch die ‘queue’ sind.

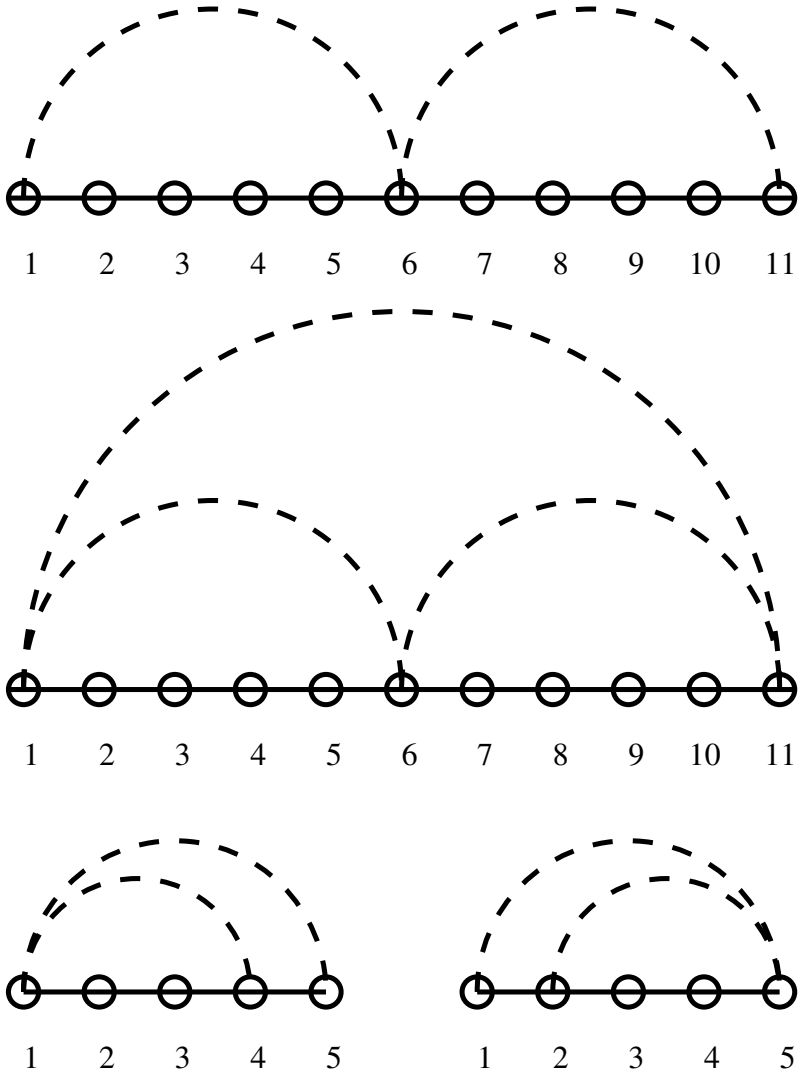


Abbildung 20: Contact-Map-Graphen, die sowohl ‘stack’s als auch ‘queue’s sind

5.5 Queue vs. 2-Stack

In der Abbildung 21 sehen wir eine Struktur, deren Modellstrukturen sowohl der ‘2-stack’ als auch die ‘queue’ sind. Wobei wir den ‘2-stack’ erhalten, wenn wir die Kante von Punkt drei bis Punkt neun nach unten klappen.

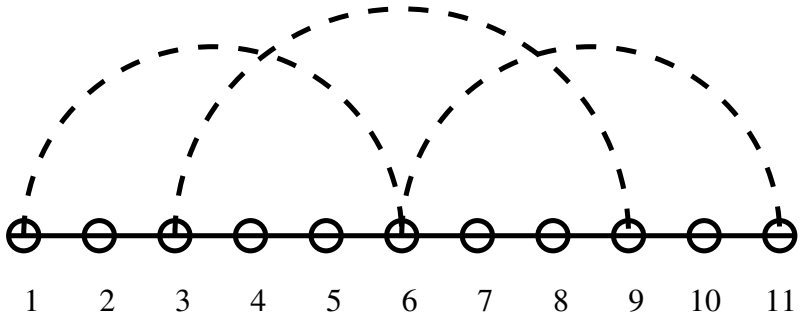


Abbildung 21: Contact-Map-Graph, der sowohl ein ‘2-stack’ als auch eine ‘queue’ ist

In der Abbildung 22 sehen wir eine Struktur, deren Modellstruktur die ‘queue’ ist, aber nicht der ‘2-stack’.

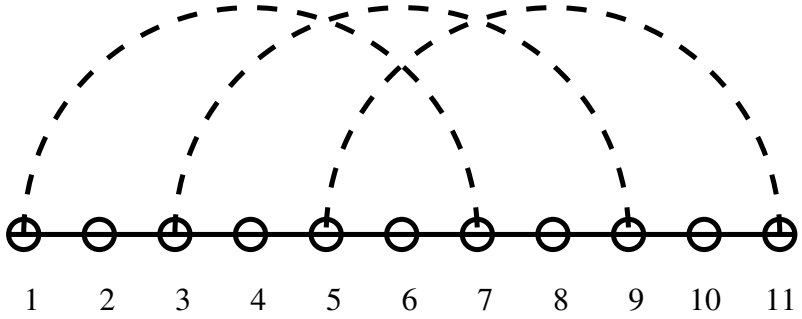


Abbildung 22: Contact-Map-Graph, der eine ‘queue’ ist, aber kein ‘2-stack’

Satz *Jeder ‘walk’ in der Ebene kann in zwei ‘stack’s und eine ‘queue’ zerlegt werden.*

Dieser Satz ist das “Dekompositionstheorem” aus Goldman [Gol00]. Dort ist auch der zugehörige Beweis zu finden.

5.6 Vereinigungsmenge von Queues

In der Abbildung 23 sehen wir zwei ‘queue’s, deren Vereinigung keine ‘queue’, sondern ein ‘stack’ ist.

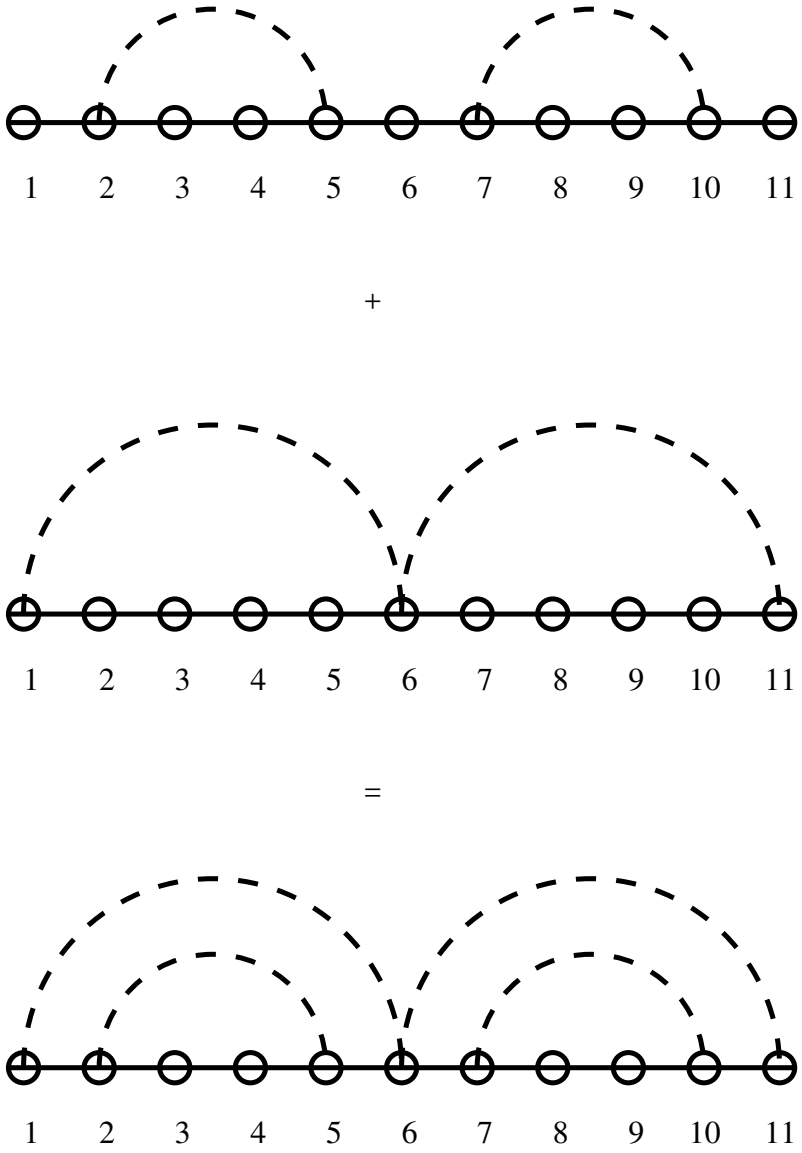


Abbildung 23: Contact-Map-Graphen ‘queue’, deren Vereinigung einen ‘stack’ ergibt

5.7 Resümee der eigenen Ergebnisse dieses Kapitels

In diesem Kapitel haben wir die folgenden eigenen Ergebnisse betrachtet.

1. Wir haben mit den Formeln 1 und 2 die Verbindung zwischen generierenden Sequenzformeln und Modellstrukturen geschaffen.

2. Ein Beweis der eindeutig optimalen Faltung in Abschnitt 5.2 wurde hiermit erstmals erbracht.

3. Weiterhin haben wir das Verhältnis der Oberfläche zur Fläche eines quadratischen hydrophoben Kerns betrachtet, wie es in der erwähnten Literatur steht.

4. In mehreren Beispielen haben wir Contact-Map-Graphen für H-P-Sequenzen und deren Modellstrukturen angegeben.

5. Insbesondere haben wir die Modellstrukturen ‘queue’ vs. ‘stack’ und ‘queue’ vs. ‘2-stack’ mit einander verglichen.

6 Alignment von RNA Strukturen

Die RNA (‘ribonucleic acid’ oder Ribonukleinsure) Struktur ist für viele Interaktionen mit verschiedenen Proteinen relevant. Vergleicht man zwei experimentell gewonnene oder berechnete RNA Sekundärstrukturen, so lässt sich bei starker Ähnlichkeit auf die Interaktion mit demselben Protein schließen. Die RNA-Sekundärstruktur ist der Anteil einer RNA Sequenz, der sich im gefalteten Zustand in einer zweidimensionalen Ebene darstellen lsst.

Ziel dieses Kapitels ist es, zwei komplexitätstheoretische Approximationsergebnisse aus der Literatur zu zitieren.

Es existiert ein Ergebnis für die Komplexität des CMOP für zwei beliebige RNA Sekundrstrukturen: In der unten angegebenen Literatur finden wir einen deterministischen, polynomiellen Approximationsalgorithmus für das CMOP für zwei RNA Sekundärstrukturen, deren Modellstruktur ein ‘2-stack’ ist.

Definition 16 *Eine* ‘staircase’ *ist eine ‘queue’, die Mengen von gegenseitig überlappenden Intervallen enthält. Zwei Intervalle derselben Menge haben eine nichttriviale, d. h. aus mehr als einem Knoten bestehende, Schnittmenge. Für zwei Intervalle in verschiedenen Mengen gilt:*

- *entweder sie treffen sich nicht*
- *oder sie haben genau einen gemeinsamen Knoten.*

Ein weiterer Spezialfall des CMOP für eine ‘queue’ und eine weitere ‘contact map’ kann approximiert werden, indem die ‘queue’ in zwei ‘staircase’s zerlegt wird (vgl. Goldman [Gol00]).

Für weitere Literatur zum Thema RNA-Alignment, s. Lenhof et al. [LRV98].

Beispiel 4 *In der Abbildung 24 sehen wir die graphische Veranschaulichung einer ‘staircase’.*

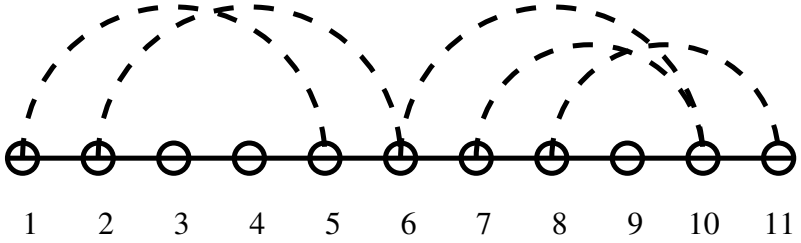


Abbildung 24: Contact-Map-Graph ‘staircase’

Zum Abschluss wiederholen wir das oben erwähnte Ergebnis zum CMOP von zwei RNA Sekundärstrukturen, wie wir es in der Literatur finden. Goldman [Gol00] benutzt für den 2-Approximationsalgorithmus den folgenden

Satz 3 *Es existiert ein $O(n^4)$ Algorithmus, um die maximale Überlappung von zwei Contact-Map-Graphen vom Grad zwei zu finden, von denen einer ein ‘stack’ ist.*

Den Beweis wollen wir hier nicht wiedergeben und verweisen auf Goldman [Gol00].

Alle bekannten RNA Sekundärstrukturen, bis auf eine, haben Contact-Map-Graphen, die aus einem ‘2-stack’ vom Grad eins bestehen. Daher gilt das folgende

Korollar 1 *Es existiert ein 2-Approximationsalgorithmus für die Berechnung der maximalen Überlappung von zwei beliebigen RNA Sekundärstrukturen.*

Beweis 1 *Ein c-Approximationsalgorithmus, für eine positive Konstante c, ist ein Polynomialzeitalgorithmus, der garantiert, dass das berechnete Ergebnis*

multipliziert mit dem Faktor c optimal ist. Daher benötigen wir für dieses Korollar einen Algorithmus, der garantiert, dass das berechnete Ergebnis ein Wert ist, der mindestens halb so groß wie die wahre maximale Überlappung ist.

Wir zerlegen einen der RNA Contact-Map-Graphen in zwei 'stack's. Einer dieser 'stack's enthält mindestens die Hälfte der Kanten, die vom optimalen Alignment erhalten bleiben. Wir optimieren die Überlappung von jedem 'stack' gegen den anderen Contact-Map-Graphen, wobei wir Satz 3 benutzen. Wenn wir den gößeren der beiden Werte für die Überlappung nehmen, dann haben wir garantiert einen Wert, der mindestens halb so groß wie das Optimum ist.

7 Ausblick

Die gezeigten Strukturen im H-P-Modell, die wir gefaltete und spiralförmige Struktur genannt haben, können in einer Computersimulation genauer untersucht werden. Hayes [Hay98a] benutzt für unseren so genannten 'walk' selber den Begriff 'self-avoiding walk'.

Offen ist auch, ob alle bekannten (nach Aichholzer et al. [ABD+03]) eindeutig optimal gefalteten Sequenzen untereinander verwandte Eigenschaften besitzen, wie die Erreichbarkeit mithilfe einer Distanzfunktion. Dazu müssen wir biologische Mutationen, Insertionen und Deletionen als mathematische Distanz modellieren.

Literatur

[ABD+03] O. Aichholzer, D. Bremner, E. D. Demaine, D. Meijer, V. Sacristán, and M. Soss. Long proteins with unique optimal foldings in the h-p model. *Computational Geometry: Theory and Applications*, 25:139–159, 2003. Brute force statement, that some H-P sequences of length 0<n<26 have a folding in the twodimensional plane, that possesses optimal points, and no other such folding exists.

[AM97] Tatsuya Akutsu and Satoru Miyano. On the approximation of protein threading. In *Computational Molecular Biology (RECOMB)*,

pages 3–8. ACM, 1997. PT approximation algorithm. PT is MAX SNP-hard.

[Bac76] Konrad Bachmann. *Biologie für Mediziner*. Springer, 1976.

[Bie04] Thorsten Biet. Anzahl verschiedener Aminosäuren in Proteinen. Private communication, January 2004.

[BL98] B. Berger and T. Leighton. Protein folding in the hydrophobic-hydrophilic (hp) model is NP-complete. *J. of Comp. Biology*, 5:27–40, 1998. NP-completeness of the three-dimensional H-P model.

[CGP+98] P. Crescenzi, D. Goldman, C. H. Papadimitriou, A. Piccolboni, and M. Yannakakis. On the complexity of protein folding. *J. of Comp. Biology*, 5(3):423–465, 1998. NP-completeness of the two-dimensional H-P model.

[Cho92] C. Chothia. One thousand families for the molecular biologist. *Nature*, 357:543–544, 1992. Proposes families of native states of proteins.

[Die96] Reinhard Diestel. *Graphentheorie*. Springer, 1996.

[GIP99] D. Goldman, S. Istrail, and C. H. Papadimitriou. Algorithmic aspects of protein structure similarity. In *Foundations of Computer Science*, pages 512–522. IEEE, 1999. Algorithms for matching and decomposing contact map graphs of proteins, a useful concept of similarity.

[Gol00] D. G. Goldman. *Algorithmic Aspects of Protein Folding and Protein Structure Similarity*. PhD thesis, University of California at Berkeley, 2000.

[Hay98a] Brian Hayes. How to avoid yourself. *American Scientist*, 86(4):314, July 1998. Sequences in the two-dimensional grid.

[Hay98b] Brian Hayes. Prototeins. *American Scientist*, 86(3):216, May 1998. Sequences in the two-dimensional H-P model with unique optimal folding.

[HI92] L. Heath and S. Istrail. The page number of genus g graphs is $o(g)$. *J. ACM*, 39(3):479–501, 1992. Definition of stack and 2-stack as book embeddings.

[HI94] W. E. Hart and S. Istrail. Fast protein folding in the hydrophobic-hydrophilic model within $3/8$ of optimal. In *Proceedings of the 34th Annual ACM Symposium on the Theory of Computing*, pages 157–168. ACM, 1994. PFP approximation algorithm.

[Lat94] Richard H. Lathrop. The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Engineering*, 7:1059–1068, 1994.

[Len96] Thomas Lengauer. *Molekulare Bioinformatik*, pages 83–111. I. Wegener, 1996. Overview molecular bioinformatics.

[LRV98] Hans-Peter Lenhof, Knut Reinert, and Martin Vingron. A polyhedral approach to RNA sequence structure alignment. *J. of Comp. Biology*, 5(3):517–530, 1998. Deterministic polynomial algorithm for RNA alignment.

[Lys04] Reidar P. Lystad. Structure and functions of proteins. http://home.online.no/~syverl/australia/protein.pdf, May 2004.

[Mül77] Hans F. Müller. *Das moderne Lexikon*. Bertelsmann, 1977.

[MW03] R. Merkl and S. Waack. *Bioinformatik Interaktiv*. Wiley-VCH, 2003.

[Pap94] Christos H. Papadimitriou. *Computational Complexity*. Addison Wesley, 1994.

[Rei99] K. R. Reischuk. *Komplexitätstheorie Band 1*. Teubner, 1999.

[VK97] M. Vendruscolo and E. Kussel. Recovery of protein structure from contact maps. *Folding & Design*, 2:295–306, 1997. Assign three-dimensional structure to a given contact map.

[Weg93] Ingo Wegener. *Theoretische Informatik: eine algorithmenorientierte Einführung*. Teubner, 1993.

[Weg96] Ingo Wegener. *Kompendium Theoretische Informatik*. Teubner, 1996.