Kapitel 2

Methoden

2.1 Energiefunktionen für die Proteinfaltung

Eine wichtige Voraussetzung für die Simulation der Proteinfaltung ist die Verwendung einer sinnvollen Energiefunktion, mit der es möglich ist, zu einer gegebenen Proteinkonformation eine potentielle Energie zu berechnen. Am genauesten lässt sich die Energie mit quantenmechanischen Methoden berechnen, welche die elektronische Struktur explizit berücksichtigen (Foresman & Frisch, 1996). Diese Methoden sind jedoch rechnerisch so aufwendig, dass Ihre Anwendung zur Zeit auf sehr kleine Systeme beschränkt ist. Wird das System mittels *Molecular Mechanics* (Burkert & Allinger, 1982) behandelt, so lassen sich auch größere Systeme, wie z.B. Protein/Lösungsmittel, Rezeptor/Ligand und ähnliche beschreiben. Hierbei wird die klassische Mechanik angewendet. Energieterme für Bindungslängen, Bindungswinkel sowie Torsionswinkel beschreiben deren Ablenkung vom Gleichgewichtszustand. Desweiteren werden die nichtbindenden Van der Waals und elektrostatischen Wechselwirkungen berücksichtigt.
Quantenmechanische Methoden lassen sich mit *Molecular Mechanics* kombinieren (QM/MM) (Warshel & Levitt, 1976). Hierbei werden besonders wichtige Bereiche, z.B. das reaktive Zentrum eines Enzyms, quantenmechanisch, das übrige System klassisch behandelt.
Mit Hilfe der dargestellten Methoden lassen sich schnelle Prozesse molekularer Systeme bereits mit einigem Erfolg realistisch simulieren (Vagedes *et al.*, 2000; de Groot & Grubmüller, 2001).
Bei Proteinen handelt es sich jedoch um relativ große molekulare Systeme deren Faltungsprozess sich im Millisekunden- bis Sekundenbereich abspielt. Trotz der schnellen Entwicklung der Rechnergeschwindigkeiten müssen daher für dieses Problem andere

Methoden angewendet werden.

Wissensbasierte Energiefunktionen und *Threading* Bedingt durch die hohe Komplexität des Proteinfaltungsproblems sind hochaufgelöste Modelle mit einer großen Zahl an Freiheitsgraden für entsprechende Simulationen des Faltungsprozesses nicht geeignet. Aus diesem Grund werden vereinfachte Modelle zur Darstellung der Proteinstrukturen verwendet.
Eine Möglichkeit die Komplexität der Darstellung zu verringern ist die Verwendung von Gittermodellen. Ein Protein kann auf einem Gitter z.B. als Kette von Monomeren dargestellt werden, wobei die Monomere von den C_α Atomen repräsentiert werden, welche auf Gitterpunkten liegen. Die Komplexität lässt sich hierbei über die Anzahl der möglichen Zustände pro Monomer bestimmen. Gitter mit sehr hoher Anzahl an möglichen Zuständen wurden untersucht (Ortiz *et al.*, 1998). Ist die Komplexität des Gitters nicht zu hoch, so kann der Grundzustand durch Ausprobieren aller möglichen Zustände aufgesucht werden. Die Gitterdarstellung lässt sich verbessern, indem z.B. die Proteinseitenkette durch ein *Pseudoatom* dargestellt wird.
In dieser Arbeit kommen *off-lattice* Methoden zur Anwendung, wobei die Proteinstrukturen teilweise in kartesischen Koordinaten, teilweise als „Kontaktmatrix“ (siehe 2.2.1) dargestellt werden.
Eine Möglichkeit an das Proteinfaltungsproblem heranzugehen ist die Verwendung von Datenbanken an Strukturen zur Generierung von Energiefunktionen (Levitt, 1976; Eisenberg & McLachlan, 1986; Sippl, 1990; Holm & Sander, 1992; Maiorov & Crippen, 1992a; Avbelj, 1992; Wallqvist & Ullner, 1994; Karlin *et al.*, 1994; Zhang & Eisenberg, 1994; Madej & Mossing, 1993).
In dieser Arbeit werden Proteinstrukturen aus der *Protein Daten Bank* (PDB) (Berman *et al.*, 2000, Web Seite: http://www.pdb.org/) sowie in Simulationen erzeugte Strukturen verwendet, um Energiefunktionen zu erzeugen, die in der Lage sind native Proteinstrukturen von nicht-nativen Proteinstrukturen zu unterscheiden. Ein Problem bei der Proteinstrukturvorhersage ist die Tatsache, dass es unter Umständen sehr schwierig sein kann, für eine Sequenz unter Verwendung einer gegebenen Energiefunktion die Struktur minimaler Energie aufzufinden. Auch hierbei können Datenbanken verwendet werden. Es werden einfach die verschiedenen bekannten Konformationen mit der gegebenen Sequenz zu einem Sequenz/Struktur Paar verknüpft und die Energie berechnet. Da die Anzahl der verwendeten Strukturen in der Regel sehr begrenzt ist, ist es meist kein Problem die Energien aller Strukturen zu berechnen und somit die energetisch niedrigste Konformation zu bestimmen. Das Verwenden von Bibliotheken an Faltungsmotiven zum Auffinden einer passenden Struktur zu einem Protein ist auch

als „*fold recognition*“ bekannt (Bowie *et al.*, 1991; Jones & Thornton, 1992; Fisher *et al.*, 1996). Auch das „inverse Faltungsproblem“, also das Aufsuchen einer passenden Sequenz zu einer gegebenen dreidimensionalen Proteinstruktur kann auf diese Weise angegangen werden: es werden Sequenzen aus einer Datenbank verwendet um Sequenz/Struktur Paare zu erzeugen und das energieärmste Paar herausgesucht. Das Verfahren, Strukturen und Sequenzen zu neuen Paaren zu verknüpfen, wird auch als „*Threading*“ bezeichnet.

Energiefunktionen die explizite Terme für Bindungswinkel, Bindungslängen und Torsionswinkel verwenden sind auf solche Struktur/Sequenz Paare schlecht anwendbar. In der Regel stimmen die Seitenketten der dreidimensionalen Struktur nicht mit der Sequenz überein. Somit lässt sich nur das Rückgrat für die Energieberechnung verwenden. Es müssen also entweder die Seitenketten nachträglich modelliert werden (z.B. mit dem Programm *Charmm* (Brooks *et al.*, 1983)) oder man verwendet eine entsprechende Energiefunktion die in der Lage ist, nur über das Rückgrat die Struktur sinnvoll zu bewerten. Empirische Energiefunktionen, die sich nicht aus *first principles* ableiten, sondern die die Informationen der Protein Daten Bank nutzen, sind hierfür geeignet. Es gibt hierbei hauptsächlich zwei Methoden: bei den *quasichemischen* Methoden werden die Häufigkeiten von Strukturmotiven in Proteinen genutzt um die Energieparameter abzuleiten (Miyazawa & Jernigan, 1985; Hendlich *et al.*, 1990; Skolnick *et al.*, 1997; Thomas & Dill, 1996). Bei den Optimierungsmethoden wird das Potential so konstruiert, dass die nativen Strukturen in einem gegebenen Ensemble von alternativen Strukturen möglichst stabil sind (V. Maiorov, 1992; V.N. Maiorov, 1994; M. Vendruscolo, 1998; Vendruscolo *et al.*, 2000; Goldstein *et al.*, 1992b, 1992a; Koretke *et al.*, 1998; Hao & Scheraga, 1996; Mirny & Shakhnovich, 1996; van Mourik *et al.*, 1999; Dima *et al.*, 2000). In dieser Arbeit kommen beide Methoden zur Anwendung.

Zum Überprüfen von Energiefunktionen für die Proteinfaltung wurden zahlreiche Tests der Proteinerkennung verwendet (Sippl & Weitckus, 1992; Maiorov & Crippen, 1992b; Bryant & Lawrence, 1993; Kocher *et al.*, 1994; Sippl, 1995; Hinds & Levitt, 1994; Covell & Jernigan, 1990; Jernigan & Bahar, 1996; Park & Levitt, 1996).

Wird eine Sequenz mit einer zu dieser Sequenz nicht-nativen Struktur verknüpft, so wird das resultierende Sequenz/Struktur Paar auch als *Decoy*, also als Köder bezeichnet. Ein Ziel ist es, eine Energiefunktion so zu konstruieren, dass sie in der Lage ist, eine native Struktur zwischen solchen Ködern zu erkennen. Eine Sequenz, deren native Struktur erkannt werden soll, wird im weiteren als „Zielsequenz“ bezeichnet. Je mehr Zielsequenzen und je mehr Strukturen zum Erzeugen der *Decoys* verwendet werden, desto schwieriger wird die Erkennung der einzelnen Zielsequenzen, da eine steigende Anzahl an Bedingungen gleichzeitig erfüllt werden muss.

2.2 Das Proteinmodell

Für Faltungssimulationen am Computer wird ein vereinfachtes Proteinmodell benötigt. Dieses Modell sollte so gewählt werden, dass die für eine gegebene Anwendung entscheidenden Eigenschaften von Proteinen erhalten bleiben. Gleichzeitig sollten möglichst nur Parameter, die hierbei von Bedeutung sind, berücksichtigt werden um den erforderlichen Aufwand so gering wie möglich zu halten.

2.2.1 Die Kontaktmatrix

Speziell für das Erzeugen von Proteinstrukturen mit *Threading* (siehe 2.6.1) bietet es sich an, nicht die kartesischen Koordinaten der Atome, sondern eine Kontaktmatrix zu verwenden (Lifson & Sander, 1979; Chan & Dill, 1990; Godzik *et al.*, 1993; Holm & Sander, 1993; Mirny & Domany, 1996). Für ein Protein der Länge N ergibt sich eine symmetrische $N \cdot N$ Matrix $\mathbf{C}$ für die gilt:

$$C_{ij} = \begin{cases} 1 & : \text{ wenn die Aminosäuren } i \text{ und } j \text{ in Kontakt sind} \\ 0 & : \text{ sonst} \end{cases} \tag{2.1}$$

Der „Kontakt“ von zwei Aminosäuren kann hierbei auf verschiedene Weise definiert werden. Im Falle, dass nur die C_α Atome benutzt werden („C_α“ Kriterium, (Vendruscolo *et al.*, 1997a)), bedeutet „Kontakt“, dass der Abstand zwischen den Atomen nicht größer ist als ein bestimmter Maximalabstand R_c. Für diesen Maximalabstand wird hier normalerweise ein Wert von 11Å verwendet.
Bei dem „*all atom*“ Kriterium (Hinds & Levitt, 1994; Mirny & Domany, 1996) bedeutet der Kontakt, dass ein beliebiges Paar von Schweratomen der beiden Aminosäuren einen Abstand aufweist, der kleiner als ein bestimmter Wert ist. Hier wird für dieses Kriterium ein Wert von 4.5Å verwendet. In beiden Fällen wird angenommen, dass ein Kontakt nur zwischen Paaren von Aminosäuren kl besteht, deren Sequenzabstand $\text{dis}_{\text{seq}} = l - k$ einen bestimmten Minimalwert nicht unterschreitet. Der Abstand der C_α Atome von in der Sequenz benachbarten Aminosäuren beträgt 3.8Å. Würde man diese Paare in die Kontakte mit einbeziehen, so würde in diesen Fällen immer Kontakt vorliegen. Das gleiche gilt für Aminosäurepaare, die durch nur ein Residuum voneinander getrennt sind. Diese Paare liegen dicht bei einander, weil sie Nachbarn in der Sequenz sind und nicht weil sie eine Präferenz für einander haben. Somit wäre es auch nicht sinnvoll, diese in das Training der Energiefunktion mit einzubeziehen. Ähnlich verhält es sich für das *all atom* Kriterium.
In dieser Arbeit kommt ein Sequenzmindestabstand dis_{seq} von mindestens drei Amino-

säuren zum Einsatz. Abb. 2.1 zeigt die Kristallstruktur einer Kette von Glutaredoxin sowie die Darstellung der Kontaktmatrix **C**. Das C_α Kriterium ist in der oberen Hälfte dargestellt, das *all atom* Kriterium in der unteren. Entsprechend der Strukturdarstellung sind alle interhelikalen Kontakte sowie Kontakte zwischen β Faltblättern farblich gekennzeichnet. α Helices erscheinen in der Kontaktmatrix als Bänder in der Hauptdiagonalen, β Faltblätter als Bänder parallel zur Hauptdiagonalen im Falle von parallelen Faltblättern und senkrecht zur Hauptdiagonalen im Falle von antiparallelen Faltblättern.

Wie man sofort erkennt, liegen beim C_α Kriterium weit mehr Kontakte vor als beim *all atom* Kriterium. So sind beim C_α Kriterium Kontakte zwischen β Faltblatt A und C als Band senkrecht zur Hauptdiagonalen zu erkennen. Hierbei handelt es sich jedoch nicht um ein antiparalleles Faltblatt, vielmehr bilden A und B ein paralleles sowie B und C ein antiparalleles Faltblatt. Die Kontakte zwischen Faltblatt A und C sind eine Folge des relativ großen Abstands von 11Å, der für das C_α Kontakt Kriterium verwendet wird. Ein solches Abstandskriterium erweist sich jedoch bei vielen Rechnungen als sinnvoll (siehe 3.3.2).

Die vielen geometrischen Bedingungen, die in solchen Kontaktmatrizen enthalten sind, lassen eine relativ genaue Rekonstruktion der Struktur zu, selbst wenn nur ein Teil der Kontaktmatrix verwendet wird. Für die Rekonstruktion von Strukturen wurden „Distance Geometry“ (Crippen & Havel, 1988), Molekular Dynamik (Brünger *et al.*, 1986) und Monte Carlo Verfahren (Vendruscolo *et al.*, 1997b) verwendet. Auch für Kontaktmatrizen für die keine Struktur existiert, weil die enthaltenen Bedingungen nicht gleichzeitig erfüllbar sind, lässt sich eine Struktur finden, deren Kontaktmatrix möglichst nahe an der Ursprungsmatrix liegt (Vendruscolo *et al.*, 1997b).

Eine Kontaktmatrix repräsentiert natürlich nicht genau eine Struktur, sondern ein Ensemble von Strukturen, die zu dieser Kontaktmatrix passen. Diese Strukturen weichen jedoch im allgemeinen nur geringfügig voneinander ab. Ein großer Vorteil der Kontaktmatrix gegenüber kartesischen Koordinaten ist die Unabhängigkeit vom Koordinatensystem. So können zwei Kontaktmatrizen schnell über die gemeinsame Zahl an Kontakten miteinander verglichen werden.

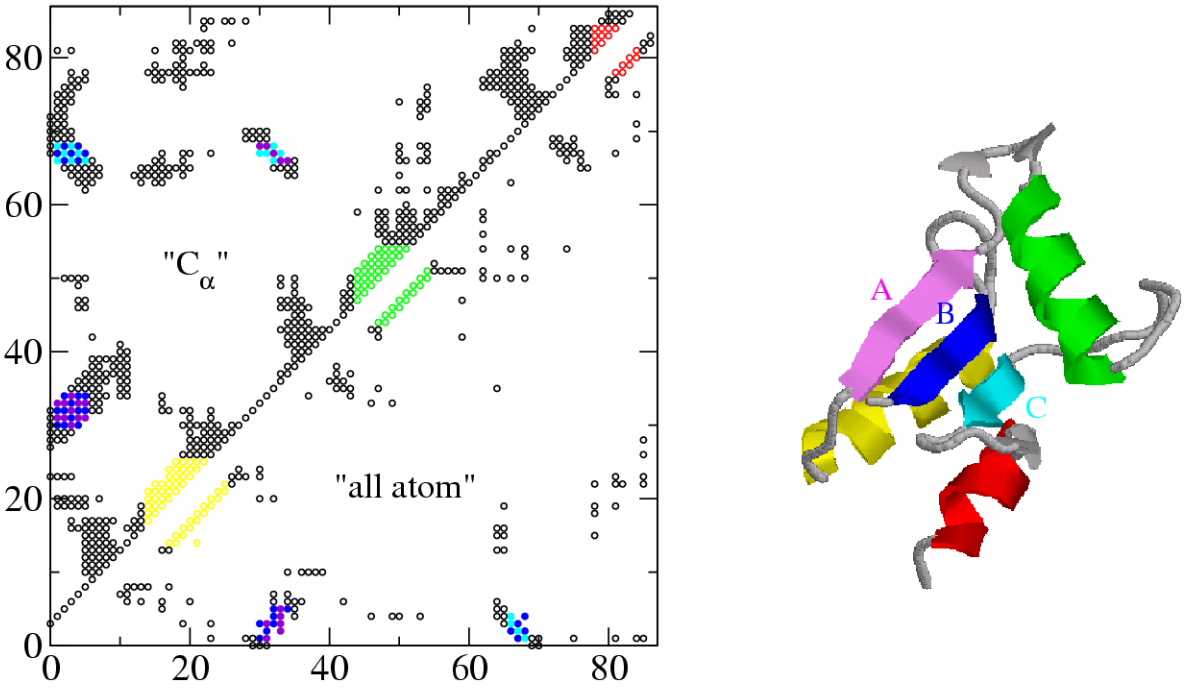

Abbildung 2.1: Strukturdarstellung und Kontaktmatrix von Glutaredoxin (A-Kette) nach dem C_α (Kontaktabstand r_c=11Å) und dem *all atom* Kriterium (Kontaktabstand r_c=4.5Å).

2.2.2 Die Energiefunktion

Wird die Kontaktmatrix **C** als Strukturmodell verwendet, so lässt sich beispielsweise folgende Energiefunktion verwenden:

$$E(\mathbf{S},\mathbf{C},\mathbf{U}) = \sum_{i<j}^{Ns} C_{ij} U(\alpha_i^s, \alpha_j^s) \tag{2.2}$$

Hierbei ist **U** die Matrix der Energieparameter, $\mathbf{S} = (\alpha_1, \alpha_2, ..., \alpha_{Ns})$ die Sequenz der Länge Ns, **C** die Kontaktmatrix und $U(a,b)$ der Energieparameter für das Aminosäurepaar a und b. Bei 20 Aminosäuren gibt es 210 mögliche Paare, also 210 Energieparameter. Dies ist die einfachste Form der in dieser Arbeit verwendeten Energiefunktionen. Die Energieparameter werden mit Hilfe eines Optimierungsverfahrens bzw. einer quasichemischen Methode bestimmt. Wichtigstes Ziel ist die Erkennung von nativen Strukturen. Gleichzeitig sollte die Energiefunktion korreliert sein. Strukturen, die der nativen Struktur ähnlich sind, sollten also eine relativ geringe Energie haben, unähnliche Strukturen eine hohe Energie (siehe 1).
Die Funktion lässt sich in vielerlei Hinsicht erweitern. Eine Möglichkeit ist die Er-

weiterung auf verschiedene Sequenzabstände dis_{seq}. So kann man Wechselwirkungen zwischen gleichen Aminosäurepaaren, die jedoch durch eine unterschiedliche Zahl an Residuen entlang der Peptidkette voneinander getrennt sind, unterscheiden. Auf diese Weise lassen sich verschiedene Sekundärstrukturmerkmale erfassen. In α Helices z.B. werden Wasserstoffbrücken zwischen dem Carbonylsauerstoff des Proteinrückgrates von Residuum i und dem Wasserstoff des Stickstoffs der Peptidbindung von Residuum $i+4$ ausgebildet. Desweiteren können bevorzugt die Seitenketten von Residuen in Position (i,i+3) und (i,i+4) wechselwirken. Es erscheint somit sinnvoll, eine gesonderte Betrachtung für Residuen in Position (i,i+3) bzw. (i,i+4) einzuführen. Für ein gegebenes Aminosäurepaar gäbe es dann z.B. einen Energieparameter für den Fall, dass der Abstand (i,i+3) oder (i,i+4) vorliegt und einen weiteren für größere Entfernungen entlang der Peptidkette. Auf diese Weise wird die Gesamtzahl an Kontaktenergieparametern von 210 auf 420 verdoppelt.

2.2.3 Disulfidbrücken

Disulfidbrücken stellen eine besonders starke Wechselwirkung zwischen Aminosäuren dar: zwei Cysteine bilden über ihre Schwefelatome eine kovalente Bindung. Disulfidbrücken unter Verwendung des C_α Modells explizit zu berücksichtigen, ist nicht ohne weiteres möglich. Sind zwei Cysteine in Kontakt, so ist nicht klar, ob eine solche Bindung vorliegt oder nicht. Über die Positionen der Schwefelatome lässt sich dies zwar leicht feststellen, jedoch sind diese im Modell nicht enthalten. Kennt man nur den C_α-C_α Abstand, so ist nicht klar, wie die Aminosäuren zueinander orientiert sind. Zudem kann ein gegebenes Cystein mit mehr als einem weiteren Cystein in Kontakt stehen. Disulfidbrücken werden also in dieser Arbeit nicht gesondert betrachtet. Bei einem Energieparameter für Cystein-Cystein Paare wird nicht unterschieden zwischen Paaren, die eine Disulfidbrücke ausbilden und Paaren die keine Disulfidbrücke ausbilden.

2.2.4 C_α oder *all atom* Kontaktkriterium?

Die beiden Kontaktkriterien C_α und *all atom* stellen zwei sehr unterschiedliche Möglichkeiten dar, einen Kontakt zwischen einem Aminosäurepaar zu definieren. Beim C_α Kriterium wird angenommen, jede Aminosäure besteht aus einer Kugel mit dem Radius $r = \frac{R_c}{2}$, wohingegen beim *all atom* Kriterium die spezifische geometrische Form einer Aminosäure berücksichtigt wird. Verwendet man für die Erstellung der Kontaktmatrizen das *all atom* Kriterium, so funktioniert die Erkennung von nativen Struktu-

ren wesentlich besser (siehe 3.3.1). Dies ist zu erwarten, da ein detaillierteres Modell verwendet wird. Ein wesentlicher Punkt muss jedoch berücksichtigt werden: bei der Erstellung der *all atom* Kontaktmatrizen werden, im Gegensatz zum C_α Kontaktkriterium, die Seitenketten der Aminosäuren berücksichtigt. Somit ist die zu einer Kontaktmatrix gehörende Struktur von der Sequenz abhängig. Unter Verwendung einer gegebenen Kontaktmatrix werden also für verschiedene Zielsequenzen verschiedene Strukturen erzeugt. Wird ein Decoy erzeugt, so werden die Sequenz des Zielproteins und die Kontaktmatrix eines Decoyproteins zu einem Sequenz/Struktur Paar vereinigt. Da die Kontaktmatrix jedoch unter Verwendung der Sequenz des Decoyproteins erstellt wurde, liegt mit großer Wahrscheinlichkeit nun nicht mehr die gleiche Struktur vor.
Natürlich kann es auch vorkommen, dass eine Kontaktmatrix zusammen mit einer bestimmten Sequenz überhaupt nicht realisiert werden kann. Es ist anzunehmen, dass die Strukturveränderungen, die beim Erzeugen der *Decoys* auftreten, mit einer Verringerung der Qualität der *Decoys* einhergehen. Somit wäre der Grund für die gute Erkennung bei Verwendung des *all atom* Kontaktkriteriums nicht ein besseres Funktionieren der Energiefunktion, sondern die Tatsache, dass die *Decoys*, bedingt durch ihre verzerrte Struktur, leichter zu erkennen sind. Es wird also nicht die richtige Struktur erkannt, sondern die richtige Sequenz.

2.2.5 Berechnen der Kontaktmatrizen

Aus den gegebenen Atomkoordinaten einer Struktur kann eine Kontaktmatrix erstellt werden. Handelt es sich um eine Struktur welche nur Standardaminosäuren enthält, so stellt dies kein weiteres Problem dar. Viele Strukturen enthalten jedoch modifizierte Aminosäuren (z.B. Selenomethionin oder S-Methyl-Cystein) oder Kofaktoren (z.B. Häm-Gruppen oder NAD). Liegt eine modifizierte Aminosäure vor, so werden für die Erstellung der Kontaktmatrix bei Verwendung des *all atom* Kontaktkriteriums die Koordinaten aller Schweratome der modifizierten Aminosäure verwendet. Beim C_α Kriterium werden ohnehin nur die Koordinaten des C_α Atoms verwendet, die Modifikation der Aminosäure muss also nicht berücksichtigt werden. Für die Wahl des Kontaktenergieparameters wird angenommen, es handelt sich um eine unmodifizierte Aminosäure. Wechselwirkungen mit modifizierten Aminosäuren werden sich von denen mit unmodifizierten unterscheiden. Um dies zu berücksichtigen, könnten neue Aminosäuretypen eingeführt werden. Es ist jedoch zu erwarten, dass aufgrund der geringen Anzahl modifizierter Aminosäuren keine aussagekräftige Statistik erhalten wird. Würde man solche Aminosäuren einfach bei der Erstellung der Kontaktmatrix ganz vernachlässigen,

so entstünde eine Lücke in der Struktur und die Kontinuität der Kette wäre unterbrochen. Somit erscheint die verwendete Vorgehensweise relativ sinnvoll.
Kofaktoren werden ganz vernachlässigt, da diese in der Regel nicht auf eine der zwanzig Aminosäuren zurückzuführen sind. Hierbei werden also wieder möglicherweise wichtige Wechselwirkungen vernachlässigt. Gerade bei größeren Kofaktoren (z.B. Häm), die zum Teil auch noch mehrfach in einem Protein vorliegen, können so relativ große Lücken in der Struktur entstehen. Beim Erzeugen von *Decoys* spielt es aber keine Rolle, ob Kofaktoren weggelassen werden. Es ist Aufgabe der Energiefunktion eine nicht-native Struktur als solche zu erkennen. Bei Zielsequenzen wird jedoch davon ausgegangen, dass eine native Struktur vorliegt. Da bei der Energieberechnung Kofaktoren unberücksichtigt bleiben, kann es sein, dass eine „falsche“ Struktur gelernt wird, da Kofaktoren häufig eine essentielle Rolle bei der Stabilisierung von Strukturen spielen. Wenn mit einer großen Menge an Zielsequenzen gelernt wird, kann es als Konsequenz passieren, dass ein Protein mit Kofaktoren nicht erkannt wird. Denkbar ist sogar, dass sich eine solche Zielsequenz so stark auf die Energiefunktion auswirkt, dass die Erkennung von „normalen“ Sequenzen negativ beeinflusst wird.
Ein weiteres Problem sind Lücken in PDB-Strukturen. Ist z.B. ein Teil einer Proteinstruktur sehr beweglich, so kann es sein, dass er in der Elektronendichtekarte nicht lokalisiert werden kann. In einem solchen Fall fehlen dann die Koordinaten der entsprechenden Aminosäuren in der PDB-Datei. Es liegen also Lücken in der Struktur vor, welche aus den oben genannten Gründen bei den Zielsequenzen zu Problemen führen können.
Die Menge an vollständig lückenlosen, unmodifizierten Proteinen ohne Kofaktoren in der Protein Daten Bank ist zu gering, als dass man sich bei der Zusammenstellung der Proteinsets nur auf diese Strukturen beschränken könnte. Daher lassen sich gerade bei den großen Sets die oben genannten Probleme nicht vollständig vermeiden.

2.3 Ähnlichkeits- und Distanzmaße

Eine wichtige Frage ist, wie strukturelle Unterschiede zwischen Proteinkonformationen unter Verwendung eines gegebenen Ähnlichkeits- bzw. Distanzmaßes wiedergegeben werden. Wünschenswert ist ein Maß, welches möglichst gut mit der Energie korreliert. Ein Problem hierbei ist sicherlich, dass strukturelle Schwankungen unterschiedlicher Bereiche einer Struktur sich unterschiedlich stark auf die Energie auswirken. So wird der Einfluss der Fluktuation eines *Loop* Bereiches geringer sein als der eines hydrophoben Kerns.

Wichtig ist auch, inwieweit ein Maß den Abstand zweier Atome berücksichtigt. Eine Abweichung bei kleinen Abständen kann einer sehr viel höheren Energieänderung entsprechen, als bei einem großen Abstand.

2.3.1 cRMSD

Zum Vergleich zweier Strukturen (1) und (2) mit N Atomen in kartesischen Koordinaten $\vec{r}_n^{(k)}$ mit $n = 1,2,...,N$ wird häufig die RMSD (Root-mean-square-deviation) der Koordinaten verwendet, welche dann als cRMSD bezeichnet wird:

$$\text{cRMSD} = \sqrt{\frac{1}{N} \cdot \sum_{n=1}^{N} \left(r_n^{(1)} - r_n^{(2)} \right)^2} \tag{2.3}$$

Hierfür müssen die Strukturen über Translation und Rotation so zur Deckung gebracht werden, dass der Wert der cRMSD minimal wird. Dieses Problem lässt sich analytisch lösen (Kabsch, 1976). Die cRMSD verwendet für alle Atome dasselbe Gewicht. Abweichungen von größeren Atomabständen werden genauso gewichtet wie die von kleinen Abständen.

2.3.2 dRMSD

Die dRMSD vergleicht alle $N_{\text{paar}} = \frac{N(N-1)}{2}$ interatomaren Abstände r_{ij} zweier Konformationen mit N Atomen. Sie berechnet sich nach:

$$\text{dRMSD} = \sqrt{\frac{1}{N_{\text{paar}}} \sum_{i<j} \left(r_{ij}^{(1)} - r_{ij}^{(2)} \right)^2} \tag{2.4}$$

Die dRMSD lässt sich leichter berechnen als die cRMSD, da sie unabhängig vom Koordinatensystem ist. Eine Minimierung, wie sie bei der cRMSD nötig ist, fällt hier weg. Ein Problem ist jedoch, dass Atompaare mit großen Abständen, also Paare die eher weniger zur Energie beitragen, einen eher größeren Beitrag zur dRMSD leisten.

2.3.3 *Power distance*

Die *power distance* vergleicht Atompaarabstände und verwendet für kleine Abstände ein größeres Gewicht:

$$D_{\text{pow}}^{(m)} = \sum_{i<j} \left| (r_{ij}^{(1)})^{-m} - (r_{ij}^{(2)})^{-m} \right| \tag{2.5}$$

Je größer der Parameter m gewählt wird, desto stärker werden kleine Abstände gewichtet. Ein zu kleiner Wert für m führt im allgemeinen zu einer schwachen Korrelation zwischen D_{pow} und der Energie. Wird hingegen ein zu großer Wert für m gewählt, so wird das Gewicht für kleine Abstände zu hoch und der Einfluss von lokalen Eigenschaften dominiert den Wert von D_{pow}. Dies kann dazu führen, dass bei kleinem D_{pow} die globale Faltung der verglichenen Strukturen grundlegend verschieden ist. Bei der Verwendung von Definition 2.5 ist D_{pow} stark von der Länge und Kompaktheit der Strukturen abhängig. D_{pow} lässt sich jedoch z.B wie folgt normieren:

$$\tilde{D}^{(m)}_{\text{pow}} = \frac{1}{N_{\text{paar}}} \sum_{i<j} \frac{\left| (r^{(1)}_{ij})^{-m} - (r^{(2)}_{ij})^{-m} \right|}{(r^{(1)}_{ij})^{-m} + (r^{(2)}_{ij})^{-m}} \tag{2.6}$$

$\tilde{D}^{(m)}_{\text{pow}}$ liegt immer zwischen 0 und 1, jedoch werden kleine Abstände nicht unbedingt höher gewichtet als große Abstände. Für ein bestimmtes Atompaar ij mit den Abständen $r^{(1)}_{ij}$ und $r^{(2)}_{ij}$, die mit dem gleichen Faktor λ reskaliert werden, bleibt $D^{(1)}_{\text{pow}}$ unverändert.
Folgende Normierung wird in dieser Arbeit verwendet:

$$\hat{D}^{(m)}_{\text{pow}} = \frac{\sum_{i<j} \left| (r^{(1)}_{ij})^{-m} - (r^{(2)}_{ij})^{-m} \right|}{\sum_{i<j} \left[(r^{(1)}_{ij})^{-m} + (r^{(2)}_{ij})^{-m} \right]} \tag{2.7}$$

2.3.4 *Overlap*

Der *Overlap* q vergleicht zwei Kontaktmatrizen:

$$q(\mathbf{C}, \mathbf{C}') = \frac{\sum_{ij} C_{ij} C'_{ij}}{\text{Max}(\sum_{ij} C_{ij}, \sum_{ij} C'_{ij})} \tag{2.8}$$

Mit dieser Definition liegt der *Overlap* zwischen 0 und 1 und ist genau dann eins, wenn die Matrizen identisch sind.
$D_{\text{cont}} = 1 - q$ bezeichnet die entsprechende Distanz zwischen zwei Kontaktmatrizen.

2.4 Test der Distanzmaße

Die verschiedenen Distanzmaße werden die Eigenschaften von Proteinstrukturen unterschiedlich wiedergeben. Zum Test dieser Distanzmaße wird hier ein diskretes Modell verwendet.

2.4.1 Ein diskretes Proteinmodell

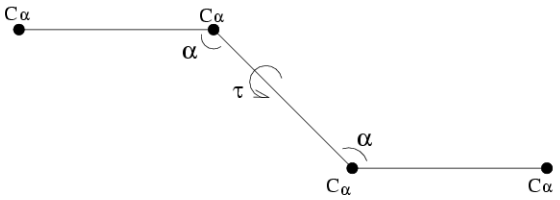

Abbildung 2.2: Das diskrete Proteinmodell verwendet nur die C_α Atome einer Proteinstruktur. Diese wird über die (α, τ) Winkel beschrieben.

Das verwendete Modell berücksichtigt nur die C_α Atome des Proteinrückgrates. Um eine gegebene Struktur zu beschreiben werden zwei Winkel eingeführt (siehe Abb. 2.2): Der Winkel α beschreibt Winkel zwischen drei aufeinanderfolgenden C_α Atomen. Da es sich hierbei nicht um einen „echten“ Bindungswinkel handelt, wird dieser Winkel als *Pseudo*-Bindungswinkel bezeichnet. Der Winkel τ zwischen vier aufeinanderfolgenden C_α Atomen wird entsprechend als *Pseudo*-Torsionswinkel bezeichnet. Die Zuordnung der α/τ Paare kann auf zweierlei Weise erfolgen: zu einem Winkel τ über die Atome $i, i+1, i+2, i+3$ kann der Winkel α über die Atome $i, i+1, i+2$ oder über die Atome $i+1, i+2, i+3$ definiert werden. In dieser Arbeit beginnt die Nummerierung am N-Terminus der Peptidkette und jeweils die Atome $i, i+1, i+2$ definieren einen Winkel α und die Atome $i, i+1, i+2, i+3$ den zugehörigen Winkel τ. Der letzte Winkel α in der Kette wird also nicht berücksichtigt.
Die Diskretisierung erfolgt, indem nur bestimmte (α, τ) Paare zugelassen werden. Park & Levitt (Park & Levitt, 1995) haben untersucht, wie gut sich dieses Modell mit realen Proteinstrukturen aus der PDB in Übereinstimmung bringen lässt. Als Distanzmaß wurde die cRMSD verwendet. Es wurde gezeigt, dass sich für ein Set von 149 Proteinen bei nur vier (α, τ) Paaren eine mittlere cRMSD von 2.2Å erreichen lässt. Zu einer gegebenen Sequenz die diskrete Modellstruktur zu finden, welche die geringste cRMSD zur realen Struktur aufweist, ist extrem aufwendig. So gibt es für Ketten mit einer Länge $N > 1$ bei k möglichen Zuständen je Residuum k^{N-2} mögliche Konformationen[1]. Minima lassen sich über übliche Minimierungsverfahren auffinden. Park und Levitt haben gezeigt, dass man über einen einfachen, deterministischen Algorithmus zu guten Minima gelangen kann (Park & Levitt, 1995).

[1] Sowohl für $N = 1$ als auch für $N = 2$ gibt es für die Darstellung einer Proteinstruktur in den Winkeln α, τ nur einen möglichen Zustand. Der Abstand zwischen den Residuen ist festgelegt und wird nicht variiert. Somit können die C_α Atome der ersten beiden Residuen im Abstand von r_c beliebig in den Raum gelegt werden.

2.4.2 *Build up* Algorithmus

Der *Build up* Algorithmus von Park und Levitt besteht in einem sukzessiven Aufbau der Peptidkette. Ziel ist es, zu einer Modellstruktur mit möglichst großer Ähnlichkeit zur Kristallstruktur zu gelangen. Hierbei wird unter Verwendung von k α/τ Winkelpaaren an eine wachsende Kette das nächste C_α Atom in allen k möglichen Positionen hinzugefügt und die erhaltenen Strukturen über das verwendete Distanzmaß bewertet. Zu jeder so erhaltenen Struktur wird wiederum das nächste C_α Atom in jeder möglichen Position hinzugefügt. Überschreitet die Anzahl der erzeugten Strukturen einen Grenzwert N_{keep} so wird nur noch mit den N_{keep} Strukturen größter Ähnlichkeit weiter verfahren. Diese Prozedur wird wiederholt bis das Kettenende erreicht ist. Von allen vollständigen Strukturen, die auf diese Weise erzeugt werden, dient jene mit größter Ähnlichkeit als Modellstruktur. Die Tatsache, dass nach jedem Schritt nicht nur die ähnlichste Struktur verwendet wird, ermöglicht das Erreichen einer guten globalen Struktur, auch wenn lokale Bereiche weniger gut mit der realen Struktur übereinstimmen. Als Distanzmaße werden hier die cRMSD, die Kontaktdistanz D_{cont} sowie die *power distance* D_{pow} verwendet (siehe 2.3).
Die erreichte Ähnlichkeit ist stark von den verwendeten Winkelpaaren abhängig. Zum Auffinden von möglichst geeigneten Winkelpaaren wird eine Monte Carlo (MC) Optimierung durchgeführt.

2.4.2.1 Optimierung der Winkelpaare

Abb. 2.3 zeigt die Verteilung der α/τ Winkel für native Proteine. Die Winkelpaare lassen sich nun z.B. aus dieser Verteilung entnehmen, indem man Werte aus Bereichen mit hoher Dichte auswählt. Es zeigt sich jedoch, dass sich hierbei ein Modell mit relativ geringer Qualität ergibt. Aus diesem Grund wird eine Optimierung der Winkelpaare mittels einer Monte Carlo Methode durchgeführt. Hierfür wird mit gegebenen Winkelpaaren begonnen und die erreichte Ähnlichkeit nach einem der vorgestellten Ähnlichkeitsmaße berechnet. Ein Winkel wird zufällig ausgewählt und eine Änderung aus einem gegebenen gleichverteilten Intervall vorgenommen. Nach dem Metropolis Kriterium (siehe Gleichung 2.11) wird entschieden, ob die Änderung angenommen wird oder nicht. Diese Prozedur wird fortgeführt bis eine hohe Ähnlichkeit erreicht ist.

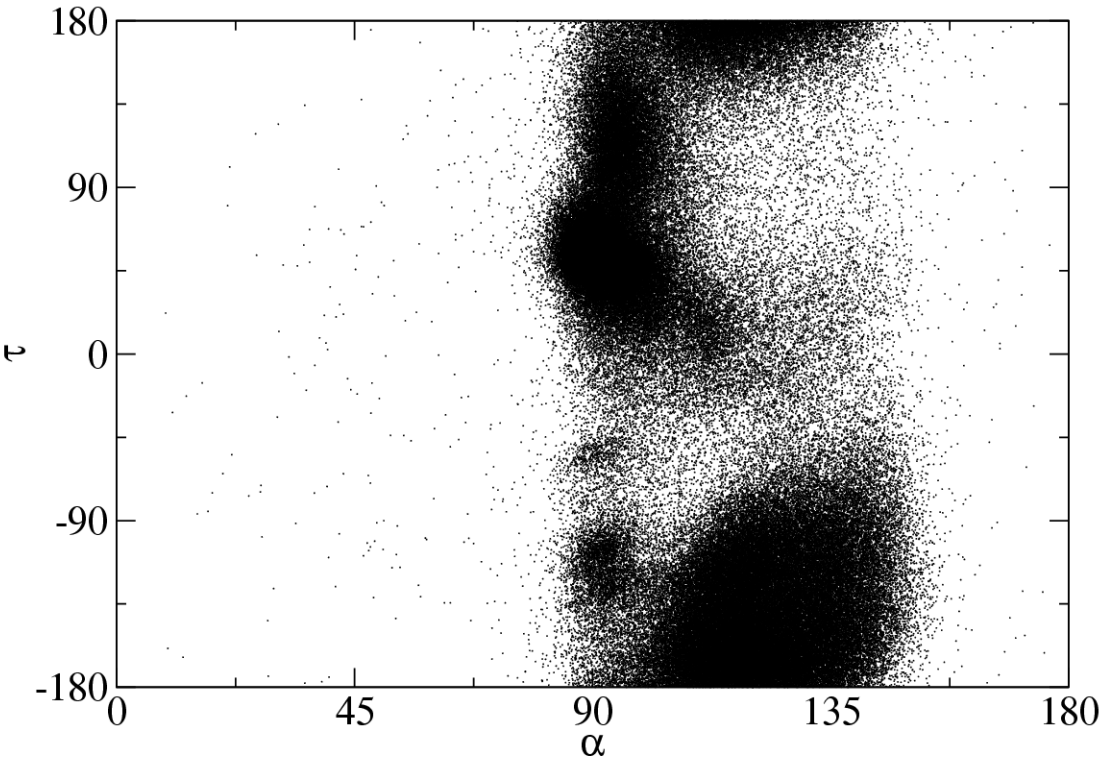

Abbildung 2.3: Verteilung der (α, τ) Winkel der 202 Zielproteine aus Set_{1014} (siehe 2.7). Der Winkel α beschreibt den Winkel zwischen drei aufeinanderfolgenden C_α Atomen, der Winkel τ den Winkel zwischen vier aufeinanderfolgenden C_α Atomen.

2.4.2.2 Verwendete Proteine

Für das diskrete Proteinmodell wird eine Teilmenge von Set_{1014} (siehe 2.7) mit 774 Proteinstrukturen verwendet. Diese Teilmenge enthält nur kontinuierliche Peptidketten. Proteine bei denen einzelne Residuen im Inneren der Peptidkette in der Kristallstruktur nicht sichtbar sind, sind also ausgenommen. Für die Optimierung der Winkelpaare wird jeweils eine Auswahl an wenigen Proteinen verwendet.

2.5 Simulation der Proteinfaltung mittels einer Monte Carlo Methode

Eine Energiefunktion, die in der Lage ist Proteinstrukturen sinnvoll zu bewerten, sollte sich auch für Monte Carlo Simulationen eignen. Sowohl für die Strukturvorhersage, als auch zum Erzeugen von *Decoys* zum Trainieren der Energiefunktion wird hier eine Monte Carlo Simulation im Raume der (ϕ, ψ)-Torsionswinkel verwendet. Eine reine Kontaktenergiefunktion ist hierfür nicht geeignet. Unter anderem muss darauf geachtet werden, dass sich einzelne Residuen nicht zu nahe kommen. Hierfür wird ein abstoßendes Potential E_{rep} implementiert:

$$E_{rep} = E_{rep0} \cdot \frac{r_0^{12}}{r^{12}} \tag{2.9}$$

r_0 kann hierbei in Abhängigkeit vom Aminosäurepaar gewählt werden.

Auch wenn mit dem abstoßenden Potential (2.9) verhindert wird, dass sich einzelne Residuen zu nahe kommen, kann es passieren, dass eine mit der MC Methode simulierte Proteinstruktur kompakter als die native Struktur wird. Um diesem Effekt entgegenzuwirken, wird eine große mittlere Anzahl an Kontakten mit einer positiven Energie E_c belegt:

$$E_c = \begin{cases} 0 & : \text{ wenn } 2N_c/N \leq C \\ ((2N_c/N) - C) \cdot E_{p0} & : \text{ wenn } 2N_c/N > C \end{cases} \tag{2.10}$$

Hierbei ist N_c die Gesamtzahl der Kontakte in einer Struktur und N ist die Sequenzlänge. Wenn die mittlere Kontaktzahl $2\frac{N_c}{N}$ den Wert C überschreitet wird eine positive Energie hinzugefügt. Die Höhe dieser Energie ist linear abhängig von $\frac{N_c}{N}$. Welcher Wert für C sinnvoll ist, hängt von der Sequenzlänge ab. Die mittlere Anzahl der Kontakte nimmt im allgemeinen mit zunehmender Länge zu, streut jedoch relativ stark (siehe Abb. 3.4).
Werden Kontaktenergieparameter nach der in Abschnitt 2.10 beschriebenen Methode erzeugt, so ist die Skalierung der Parameter von der Anzahl der verwendeten Strukturen abhängig. Um die Skalierung verschiedener Kontaktenergiefunktionen bezüglich der Energieparameter zu vereinfachen, wird die Summe der quadratischen Kontaktenergieparameter $\sum u_i^2$ (siehe Gleichung 2.19) für diese Energiefunktionen auf einen konstanten Wert, normalerweise 10, gesetzt.
In einer MC Simulation werden pro MC Schritt entweder ein Winkel oder zwei Winkel

am gleichen C_α Atom zufällig geändert. Ein Schritt wird mit folgender Wahrscheinlichkeit angenommen (Metropolis Kriterium):

$$W = \begin{cases} exp\left(-\frac{\Delta E}{T}\right) & : \text{ wenn } \Delta E > 0 \\ 1 & : \text{ sonst} \end{cases} \tag{2.11}$$

Die Monte Carlo Simulation kann sowohl für Proteinstrukturvorhersagen also auch zur Erzeugung von *Decoys* für das Training der Energiefunktion verwendet werden.

Simulated Annealing Die Temperatur T in Gleichung 2.11 ist in Verbindung mit den hier verwendeten Energiefunktionen keine „echte“ Temperatur. Will man eine Faltungssimulation durchführen, so ist die Wahl der richtigen Temperatur von großer Bedeutung. Da es schwierig ist eine allgemeingültig sinnvolle Temperatur für die Faltung zu ermitteln, bietet sich das Verfahren des *simulated annealing* an. Hierbei wird die Temperatur während der Simulation herunterskaliert. Bei der hier verwendeten Methode wird die Temperatur entweder nach einer bestimmten Zahl von erfolgreichen Monte Carlo Schritten oder nach Erreichen einer maximalen Anzahl an Schritten linear mit dem Faktor *scale* herunterskaliert:

$$T_{neu} = T_{alt} \cdot scale \tag{2.12}$$

Wird mit einer Temperatur, die oberhalb der Faltungstemperatur liegt, begonnen und dann weit genug heruntergekühlt, können relativ tiefe Energien erreicht werden.

2.6 Erzeugen von *Decoys*

Um eine Energiefunktion auf das Erkennen von nativen Proteinstrukturen zu trainieren, werden native und nicht-native Strukturen benötigt. Native Strukturen können z.B. der Protein Daten Bank (PDB) entnommen werden. Für das Erzeugen von nicht-nativen Strukturen gibt es eine Vielzahl von Möglichkeiten. Kontaktmatrizen zu diesem Zweck zufällig zu erstellen, ist keine sinnvolle Methode. Zu einer zufälligen Kontaktmatrix gibt es mit großer Wahrscheinlichkeit keine entsprechende Struktur, da die enthaltenen Bedingungen häufig nicht gleichzeitig erfüllt werden können. Selbst wenn eine Kontaktmatrix geometrisch realisierbar ist, heisst das nicht, dass sie auch physikalisch sinnvoll ist.

2.6.1 Erzeugen von *Decoys* mittels *Threading*

Eine einfache und schnelle Methode eine große Zahl an physikalisch sinnvollen *Decoys* zu erzeugen ist die *Threading* Methode. Dabei wird eine Zielsequenz der Länge N mit der Kontaktmatrix einer anderen Struktur der Länge $N' \geq N$ kombiniert.
Haben die beiden Proteine die gleicher Länge so werden sie einfach zu einem neuen Sequenz/Struktur Paar verknüpft. Ist das Protein zum Erzeugen der *Decoys* länger als die Zielsequenz, so werden Zeilen und Spalten der (N',N') Kontaktmatrix gestrichen, so dass Untermatrizen mit den Spalten und Zeilen $i+1, i+2, ..., i+N$ entstehen. Hierbei ist $i = 0, 1, ..., N'-N$. Die auf diese Weise erzeugten (N,N) Matrizen werden als Kontaktmatrix für die Zielsequenz verwendet. Zu einer (N',N') Kontaktmatrix gibt es $N'-N+1$ solcher Untermatrizen, die für die Erzeugung der *Decoys* verwendet werden können. Werden native Strukturen zum Erzeugen der *Decoys* verwendet, so erhält man physikalisch sinnvolle Strukturen, die typische Eigenschaften von Proteinen, wie z.B. die Sekundärstrukturmerkmale aufweisen.

2.6.2 Erzeugen von *Decoys* mittels einer Monte Carlo Methode

Zum Erzeugen von *Decoys* mit im Prinzip beliebiger Ähnlichkeit eignen sich Monte Carlo Methoden, die z.B. native Strukturen als Startpunkt verwenden. Hier wird eine Monte Carlo Methode im Raume der (ϕ, ψ)-Winkel verwendet. Die Strukturen der Trajektorie dienen als *Decoys*. Für die Monte Carlo Simulation werden natürlich initiale Kontaktenergieparameter benötigt. Hierfür können z.B. Parameter, die mit Hilfe von *Threading* erzeugt wurden, verwendet werden. Eine besonders genaue Energiefunktion ist hierbei nicht nötig. Verwendet man als Startkonformation eine native Struktur und erzeugt kurze Trajektorien, so lassen sich z.B. mit verschiedenen Temperaturen und verschiedenen *seeds* für die Erzeugung der Zufallszahlen, viele Strukturen mit hoher Ähnlichkeit zur nativen Konformation erzeugen.

2.7 Verwendete Proteine und Strukturen

Für das Training der Energiefunktionen wird für verschiedene Sequenzen die native Struktur zusammen mit einer großen Zahl nicht-nativer Strukturen betrachtet, wobei eine native Struktur, unter Verwendung einer gegebenen Energiefunktion, die niedrigste Energie aufweisen sollte. Die nativen Strukturen stammen aus der Protein Daten Bank.
Um eine gute Statistik zu erhalten ist es sinnvoll, eine größere Zahl an Zielsequenzen

zu betrachten und für diese eine große Anzahl an *Decoys* zu erzeugen. Es wird also eine große Zahl an nativen Proteinen benötigt. In dieser Arbeit werden verschiedene Sets an nativen Proteinen verwendet (siehe Anhang). Aus einem gegebenen Set werden alle Strukturen für die Erzeugung von *Decoys* benutzt. Als Zielsequenzen dienen nur jene Strukturen, die bestimmte Bedingungen erfüllen. In der Regel werden hierfür alle einzelkettigen Proteine mit einer Länge kleiner gleich 200 Aminosäuren verwendet. Da es nicht möglich ist, eine Kontaktmatrix für ein mehrkettiges Protein sinnvoll aufzustellen, kann man die einzelnen Ketten von mehrkettigen Proteinen nur separat anwenden. Um weitere *Decoys* zu erzeugen ist diese Vorgehensweise sinnvoll. Weist eine solche Struktur zusammen mit der beim *Threading* verwendeten Sequenz keine nativen Eigenschaften auf, so ist es Aufgabe der Energiefunktion dies zu erkennen. Wird eine solche Kette als Zielsequenz verwendet, so werden Wechselwirkungen zwischen den Ketten vernachlässigt. Somit wird unter Umständen eine falsche Struktur gelernt. Aus diesem Grund werden nur einzelkettige Proteine als Zielsequenz verwendet.
Je länger eine Sequenz ist, desto weniger *Decoys* lassen sich unter Verwendung eines gegebenen Sets an Proteinen erzeugen. Daher ist es sinnvoll nur Proteine mit begrenzter Länge als Zielsequenzen zu verwenden.
Die PDB enthält zur Zeit 30963 Strukturen[2]. Würde man alle oder einen willkürlichen Teil aller Proteinstrukturen aus diesem Set verwenden, so wäre die Gefahr einer ungleichen Verteilung sehr hoch. Werden zum Beispiel nur Strukturen als Zielsequenzen verwendet, die überwiegend helikal aufgebaut sind, so hätte die Funktion eine Tendenz zur Erkennung solcher helikalen Proteine. Sehr viele Proteine in der Protein Daten Bank sind homolog. Die Homologie kann so weit gehen, dass sich zwei Proteine in nur sehr wenigen Aminosäuren unterscheiden. Solche Proteine können in ihrer Struktur weitgehend identisch sein und bekommen ein zu großes Gewicht, wenn sie alle beim *Threading* berücksichtigt werden.
Kristallstrukturen und NMR Strukturen unterscheiden sich in ihren Eigenschaften und werden auch unterschiedlich gut von den Energiefunktionen erkannt. Godzik *et al.* haben festgestellt, dass sich Energiefunktionen, die durch statistische Analyse von Kristallstrukturen hergeleitet wurden, schlecht für die Erkennung von NMR-Strukturen verwenden lassen und umgekehrt (Godzik *et al.*, 1995). Aus diesem Grund werden Kristall- und NMR-Strukturen getrennt behandelt. Die verschiedenen verwendeten Proteinsets sind im Anhang aufgeführt.

[2]Stand 17.5.2005

Kristallstrukturen Hier werden hauptsächlich vier verschieden große Sets an Proteinen verwendet, wobei ein kleineres Set immer eine Teilmenge jedes der größeren Sets darstellt. Set_{1014} enthält 1014 Ketten, von 965 verschiedenen Proteinen. Das Set_{420} enthält nur einkettige Proteine ohne Häm-Gruppen. Set_{135} enthält nur einkettige Proteine ohne Kofaktoren. Das kleinste Proteinset Set_{45} enthält 45 relativ kurze Proteine mit einer maximalen Länge von 142 Aminosäuren. Unter Verwendung des C_α Kontaktkriteriums mit $r_c = 11$Å kommt jeder der 210 möglichen Kontakte mindestens einmal vor.

NMR Strukturen Das Set der NMR Strukturen enthält 156 Ketten von 156 verschiedenen Proteinen. 17 Ketten stammen aus mehrkettigen Proteinen. Das Set enthält 135 Zielsequenzen (einzelkettige Proteine mit einer Länge $N \leq 200$ Residuen). Ein Problem bei NMR Strukturen ist die Tatsache, dass in vielen Fällen mehrere Konformationen für eine Sequenz in einer PDB-Datei angegeben sind. Diese werden mit model 1, model 2 usw. bezeichnet. In einigen Fällen ist die Nummer des repräsentativen Modells angegeben, welches nicht unbedingt Nr. 1 sein muss. Diese Angabe erfolgt nur äusserst selten, so dass sie schlecht genutzt werden kann. In dieser Arbeit wird bei NMR Strukturen die erste in einer PDB Datei angegebene Konformation verwendet.
NMR Strukturen sind Konformationen in Lösung, was erwarten lässt, dass sich prinzipielle strukturelle Unterschiede zu Kristallstrukturen ergeben (siehe z.B. Abb. 3.4).
Der *Overlap* zwischen zwei Strukturen hängt in der Regel vom verwendeten Kontaktkriterium ab. Abb. 2.4 zeigt die Verteilungen der *Overlaps q* aller fünf Proteinsets für das C_α und das *all atom* Kontaktkriterium.

Proteine für Monte Carlo Faltungssimulationen Zum Testen verschiedener Energiefunktionen in Monte Carlo Faltungssimulationen werden drei verschiedene Proteine verwendet (siehe Abb. 2.5):

- Crambin (PDB Code: 1ejg)
  Ein häufig verwendetes Protein für das Testen von Kontaktenergiefunktionen ist das Pflanzensamen-Protein Crambin (Jelsch *et al.*, 2000). Es handelt sich um ein einkettiges Protein mit einer Länge von 46 Residuen. Diese Eigenschaften machen es gut handhabbar für Kontaktenergiefunktionen und Faltungssimulationen.

- *Mating Pheromone Er-1* (PDB Code: 2erl)
  Hierbei handelt es sich um ein trippelhelikales Protein aus 40 Aminosäuren.

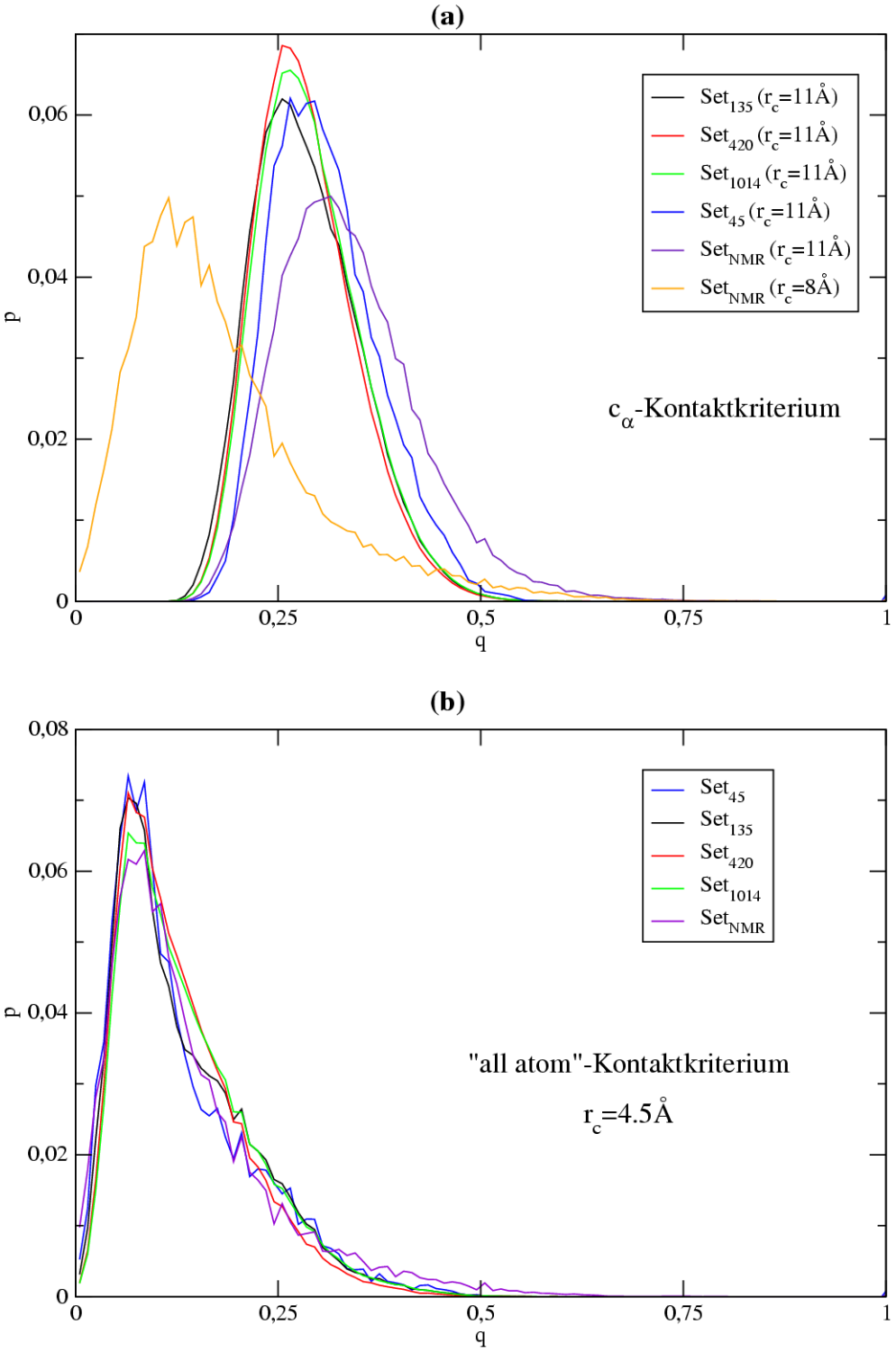

Abbildung 2.4: Verteilungen der *Overlaps* q für die fünf für *Threading* verwendeten Proteinsets unter Verwendung des C_α (a) sowie des *all atom* (b) Kontaktkriteriums. Für das C_α Kontaktkriterium werden für das $\mathrm{Set_{NMR}}$ die Kontaktabstände $r_c = 8$Å und $r_c = 11$Å verglichen.

- *Cro Repressor Insertion Mutant K56-[Dgevk]* (PDB Code: 1orc)
  Dieses Protein aus 64 Aminosäuren besteht aus mehreren α Helices und antiparallelen β Faltblättern.

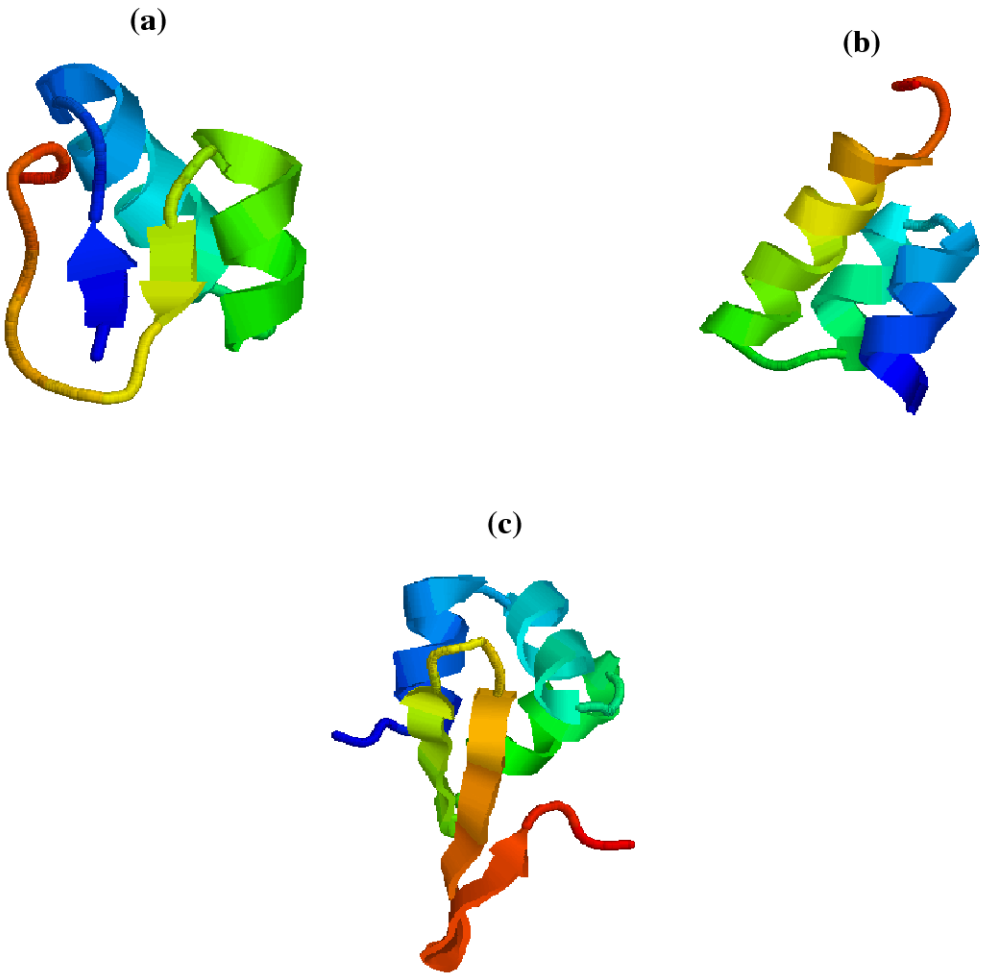

Abbildung 2.5: Zum Testen verschiedener Energiefunktionen in Monte Carlo Faltungssimulationen werden drei verschiedene Proteine verwendet.
(a): Crambin (1ejg), (b): *Mating Pheromone Er-1* (2erl), (c): *Cro Repressor Insertion Mutant K56-[Dgevk]* (1orc).

2.8 Beschreibung von Aminosäuren an der Proteinoberfläche

Berücksichtigt die Energiefunktion nur die Kontakte zwischen Aminosäuren, so wird die Wechselwirkung von Aminosäuren an der Proteinoberfläche mit Lösungsmittelmolekülen der Umgebung vernachlässigt. Es wird nur festgelegt welche Aminosäuren bevorzugt wechselwirken, eine Präferenz für das Auftreten einer Aminosäure an der Oberfläche lässt sich nicht einstellen. Nur wenn alle Wechselwirkungen einer Aminosäure abstoßend, also alle zwanzig Kontaktenergieparameter an denen die Aminosäure beteiligt ist positiv sind, ist eine Präferenz für die Proteinoberfläche gegeben. Da jede Aminosäure jedoch anziehende Wechselwirkungspartner benötigt, ist dies normalerweise nicht der Fall. Daher werden von der Energiefunktion kompakte Strukturen favorisiert, auch wenn eine Sequenz vorliegt, die eine relativ offene native Struktur besitzt. Es lässt sich jedoch relativ einfach ein zusätzlicher Oberflächenenergieparameter für jede Aminosäure einführen. Hierfür wird angenommen, dass eine Aminosäure mit Lösungsmittelmolekülen wechselwirkt, wenn sie weniger als eine vorgegebene Anzahl A an Nachbarn besitzt. Die zugehörige Energiefunktion lautet:

$$
\begin{aligned}
E(\mathbf{C},\mathbf{S},\mathbf{U}) &= E_c + \sum_{i=1}^{N} f(C_i) \\
f(C_i) &= \begin{cases} \left(A - \sum_{j=1}^{N} C_{ij}\right) \cdot U^{(o)}_{\alpha_i{}^s} & : \quad A > \sum_{j=1}^{N} C_{ij} \\ 0 & : \quad A \leq \sum_{j=1}^{N} C_{ij} \end{cases}
\end{aligned}
\tag{2.13}
$$

E_c: Kontaktenergie der Aminosäurepaare
N: Länge der Sequenz
$U^{(o)}_{\alpha_i{}^s}$: Oberflächenenergieparameter für Aminosäure $\alpha_i{}^s$
A: Oberflächenparameter: Anzahl der Nachbarresiduen einer Aminosäure im Proteininneren

Es wird also für Residuen, die weniger als A Nachbarn haben, die Anzahl an Kontakten mit dem Oberflächenenergieparameter $U^{(o)}_{\alpha_i{}^s}$ auf A aufgefüllt. Ein entscheidender Punkt ist hierbei die Wahl dieses Oberflächenparameters A. Abb. 2.6 zeigt die mittlere Anzahl der Kontakte für die verschiedenen Aminosäuren. Die Anzahl der Kontakte der individuellen Residuen variiert jedoch sehr stark. Abb. 2.7 zeigt die Häufigkeitsverteilungen für die Anzahl an Nachbarkontakten für Residuen vom Typ Glycin und Phenylalanin. In Übereinstimmung mit Abb. 2.6 liegt der Median für Phenylalanin bei höheren Werten als der für Glycin. Die Standardabweichungen sind mit 7.4 für Glycin

und 6.3 für Phenylalanin sehr hoch. Diese Tatsache macht es schwierig, einen sinnvollen Wert für A abzuschätzen. Im Ergebnissteil ist dargestellt, wie sich dieser Parameter auf die Erkennung nativer Strukturen auswirkt (siehe 3.3.4).

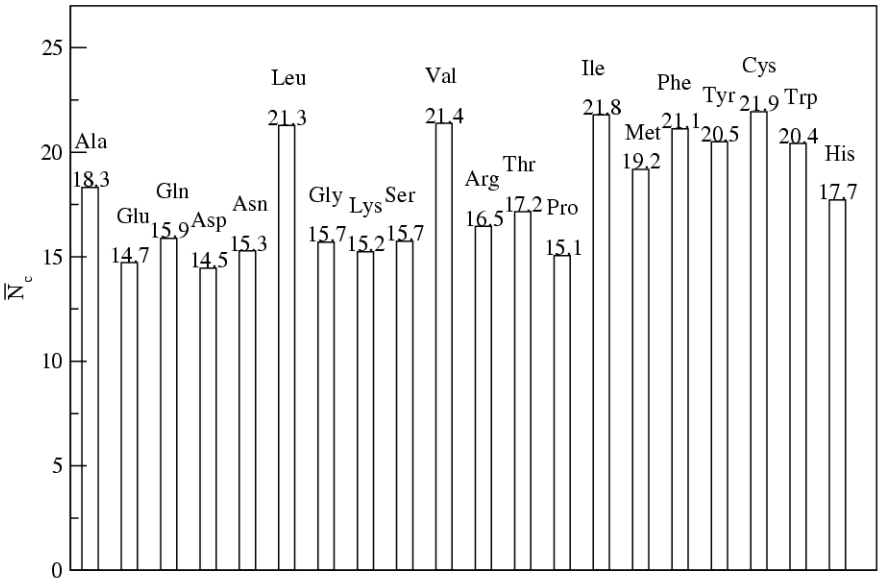

Abbildung 2.6: Mittlere Anzahl der C_α Kontakte pro Residuum der 20 verschiedenen Aminosäuren für die 202 Zielproteine aus Set_{1014}.

Von den unpolaren Aminosäuren haben Alanin, Cystein, Isoleucin, Leucin, Methionin, Phenylalanin, Tryptophan und Valin eine erwartungsgemäß hohe Anzahl an Kontakten. Diese Aminosäuren befinden sich vorzugsweise im Inneren der Proteine wo in jeder Richtung Kontakte vorliegen. Die verbleibende unpolare Aminosäure Prolin hat eine relativ geringe mittlere Anzahl an Nachbarn. Prolin findet sich häufig in *Loop* Regionen. Diese befinden sich vermehrt an der Proteinoberfläche, was ein Grund für das Auftreten von Prolin an der Oberfläche sein wird. Die Aminosäuren mit geladenen Seitenketten Glutaminsäure, Asparaginsäure, Lysin und Arginin befinden sich hauptsächlich an der Oberfläche, wo die Ladung durch polarisierte Wassermoleküle gut abgeschirmt (solvatisiert) wird. Wie zu erwarten, haben diese Aminosäuren im Mittel dann auch nur wenig Kontakte. Liegen sie im Inneren, so muss die Ladung über Salzbrücken ausgeglichen werden.

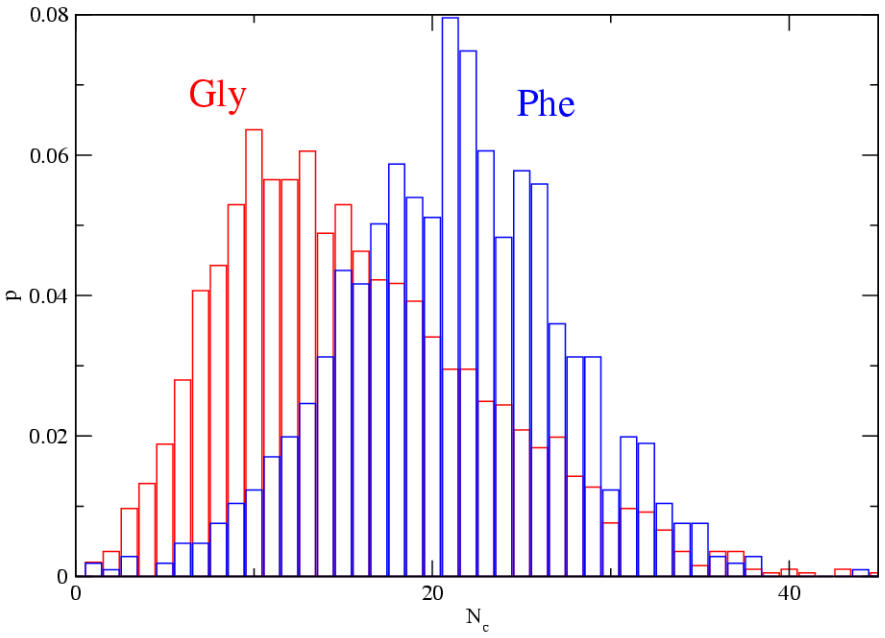

Abbildung 2.7: Histogramm der Anzahl der Nachbarn individueller Residuen. Es gilt das C_α Kontaktkriterium mit einem Abstandskriterium von $r_c = 11$Å. Als in Kontakt gelten nur Residuen mit einem Sequenzabstand von mindestens dis_{seq}=3.

2.9 Optimierung der Energieparameter durch Maximierung des Boltzmann-gewichteten *Overlaps*

Bastolla *et al.* verwenden für die Erzeugung der Energieparameter eine Methode, die darauf abzielt den Boltzmann-gewichteten *Overlap* Q zwischen *Decoy* Strukturen und experimentellen Strukturen zu maximieren (Bastolla *et al.*, 2001):

$$Q(\mathbf{S},\mathbf{U}) = \frac{\sum_\Gamma q(\mathbf{C}_\Gamma, \mathbf{C}_n(\mathbf{S})) e^{-E(\mathbf{S},\mathbf{C}_\Gamma,\mathbf{U})/k_bT}}{\sum_\Gamma e^{-E(\mathbf{S},\mathbf{C}_\Gamma,\mathbf{U})/k_bT}} \tag{2.14}$$

$q(\mathbf{C}_\Gamma, \mathbf{C}_n(\mathbf{S}))$ gibt hierbei den *Overlap* zwischen der nativen Kontaktmatrix $\mathbf{C}_n$ der Sequenz $\mathbf{S}$ und der Kontaktmatrix $\mathbf{C}_\Gamma$ der Konformation Γ an. Geht $Q(\mathbf{S},\mathbf{U})$ gegen eins, so ist die Kontaktmatrix der energieärmsten Struktur identisch oder sehr ähnlich zur Kontaktmatrix der nativen Struktur. Eine wichtige Rolle spielt die Temperatur T: ist

$Q(\mathbf{S},\mathbf{U})$ nahe eins und die Temperatur nicht zu hoch, weisen Strukturen die sehr unähnlich zur nativen Struktur sind hohe Energien auf. Wird die Optimierung also bei nicht zu hoher Temperatur durchgeführt, so ist zu erwarten, dass eine korrelierte Energiefunktion erzeugt wird.
Die Optimierung erfolgt mit einem Satz an *Zielsequenzen* und zugehörigen *Decoy* Strukturen. Optimiert wird hierbei die Summe der Boltzmann-gewichteten *Overlaps* der einzelnen Zielsequenzen:

$$\overline{Q} = \sum_{\mathbf{S}} Q(\mathbf{S},\mathbf{U}) \tag{2.15}$$

Als Optimierungsmethode dient hierbei eine Methode des steilsten Abstiegs (*steepest descent*). Ausgehend von einem zufällig erzeugten Set an Energieparametern oder einem Set aus einer vorangegangenen Optimierung werden $\overline{Q}$ sowie die Energie der nativen Struktur $E(\mathbf{C}_n,\mathbf{S},\mathbf{U})$ optimiert:

$$\tilde{\mathbf{U}}^{(t+1)} = \mathbf{U}^{(t)} + \delta\nabla_{\mathbf{U}} Q(\mathbf{S},\mathbf{U}) - \gamma\left(\frac{1-q_0}{q_0}\right)\nabla_{\mathbf{U}} E\left(\mathbf{C}_n,\mathbf{S},\mathbf{U}\right) \tag{2.16}$$

Die Parameter δ und γ geben hierbei die Schrittweiten der Optimierung an, q_0 ist der *Overlap* zwischen nativer Struktur und der Struktur niedrigster Energie.
Nach jedem Optimierungsschritt werden die Energieparameter reskaliert:

$$\mathbf{U}^{(t+1)} = \frac{1}{\tau}\tilde{\mathbf{U}}^{(t+1)} \tag{2.17}$$

Hierbei wird τ so gewählt, dass die Summe der quadratischen Energieparameter während der Optimierung konstant bleibt:

$$U^2 = \sum_{a,b} U^2(a,b) \tag{2.18}$$

Ohne diese Reskalierung würde U^2 während der Optimierung ansteigen, was den Grundzustand stabilisieren würde.

2.10 Optimierung der Energieparameter durch Minimieren der Abweichungen eines linearen Gleichungssystems

Hier wird eine neue Methode zur Erzeugung der Energieparameter vorgestellt: die Parameter ergeben sich als beste Lösung eines überbestimmten linearen Gleichungs-

systems. Hierfür wird die Darstellung der *Decoys* stark reduziert. Es wird gezählt, wie oft die verschiedenen Kontakte vorkommen und hieraus ein Vektor $\vec{a}$ gebildet. Die $i$te Komponente dieses Vektors entspricht hierbei der Anzahl der Kontakte vom Typ i. Die Energie einer Struktur ergibt sich aus dem Skalarprodukt von $\vec{a}$ mit dem Vektor $\vec{u}$ der Energieparameter:

$$E(\mathbf{C},\mathbf{S},\mathbf{U}) = \vec{a}^t \cdot \vec{u} \tag{2.19}$$

Für den Fall einer Kontaktenergiefunktion mit einem Energieparameter für jedes Aminosäurepaar, haben die Vektoren $\vec{a}$ und $\vec{u}$ je 210 Komponenten. Die Funktion lässt sich jedoch einfach erweitern (siehe z.B. 2.2.2). Die Energieparameter sollen nun so gewählt werden, dass alle Paare in einem Ensemble aus Struktur/Sequenz Paaren korrekt als nativ oder nicht-nativ erkannt werden. Die native Struktur einer Sequenz muss also eine niedrigere Energie aufweisen als alle zu der Sequenz gehörenden nicht-nativen Strukturen. Gleichzeitig ist es wünschenswert, dass Strukturen mit hoher Ähnlichkeit zur nativen Struktur eine niedrige Energie aufweisen, da eine korrelierte Energiefunktion eine wichtige Voraussetzung für einen erfolgreichen Einsatz in Monte Carlo Verfahren ist.
Zu diesem Zweck wird jeder Struktur eine Energie zugeordnet. Da die Energie vor der Optimierung natürlich nicht bekannt ist, wird zum Schätzen der Energiewerte eine Funktion der Ähnlichkeit verwendet. Verwendet man für die Ähnlichkeit den *Overlap* q, so ergibt sich:

$$\vec{a}^t(k) \cdot \vec{u} = f(q(k)) \qquad\qquad k = 1,2...K \tag{2.20}$$

für ein Ensemble aus K Strukturen. Es handelt sich hierbei um ein lineares Gleichungssystem von K Gleichungen mit den L=210 Unbekannten in $\vec{u}$. In Matrixform ergibt sich:

$$\mathbf{A}^t \cdot \vec{u} = \vec{f}(\vec{q}) \tag{2.21}$$

wobei

$$\mathbf{A} = (\vec{a}(1),\vec{a}(2),...,\vec{a}(k)) \tag{2.22}$$

eine rechteckige (L $\cdot$ K) dimensionale Matrix mit L=Anzahl der Parameter, K=Anzahl der Strukturen ist. K ist in der Regel sehr viel größer als L und kann im Bereich von mehreren Millionen liegen. Die verschiedenen Strukturen können beliebig gewichtet werden. Um eine Struktur mit einem bestimmten Gewicht zu versehen wird die entsprechende Gleichung mit dem Gewichtsfaktor multipliziert. Die Rechenzeit bleibt

hiervon praktisch unberührt.
Das vorliegende lineare Gleichungssystem besitzt normalerweise weit mehr Gleichungen als Unbekannte. Somit ist es überbestimmt und lässt sich im allgemeinen nicht exakt lösen. Man kann jedoch analytisch eine approximative Lösung bestimmen, für welche die euklidische Distanz $\left|\left|\mathbf{A}^t\vec{u}-\vec{f}(\vec{q})\right|\right|$ zwischen rechter und linker Seite minimal ist. Gesucht ist also:

$$\min_{\vec{u}}\left[\left(\vec{u}\cdot\mathbf{A}-\vec{f}^t(\vec{q})\right)\cdot\left(\mathbf{A}^t\cdot\vec{u}-\vec{f}(\vec{q})\right)\right] \tag{2.23}$$

Die Ableitung ergibt:

$$(\mathbf{A}\cdot\mathbf{A}^t)\cdot\vec{u}=\mathbf{A}\cdot\vec{f}(\vec{q}) \tag{2.24}$$

Die (L,L) dimensionale symmetrische Matrix $\mathbf{A}\cdot\mathbf{A}^t$ wird im folgenden mit $\mathbf{A}^2$ abgekürzt. Ein wichtiger Punkt ist nun, eine sinnvolle Funktion $f(q)$ zu bestimmen. Einfache Funktionen, die einer nativen Struktur ($q=1$) die niedrigste Energie zuweisen, sind z.B. $f(q)=1-q$ oder $f(q)=-q$. Sinnvoller sind jedoch Funktionen, die nichtlinear von q abhängen, also z.B. eine Exponentialfunktion $f(q)=-De^{(\beta\cdot q)}$, wobei die Parameter D und β frei gewählt werden können.
Als allgemeiner Ansatz für $f(q)$ eignet sich ein Polynom in $(1-q)$:

$$f(q)=d_0+\sum_{\delta=1}^{\Delta}d_\delta(1-q)^\delta \tag{2.25}$$

Die Energie der nativen Struktur ist durch den Koeffizienten d_0 bestimmt, da hier der Ausdruck $(1-q)$ verschwindet. d_0 lässt sich als Temperatur auffassen. Die Energieparameter sind proportional zu d_0. Für d_0 kann ein beliebiger Wert mit $d_0<0$ gewählt werden, für verschiedene d_0 ändert sich dann nur die Skalierung der Energieparameter. Die übrigen Koeffizienten lassen sich, wie die Energieparameter, ebenfalls optimieren. Hierfür wird das lineare Gleichungssystem einfach mit den variablen Koeffizienten erweitert:

$$\vec{u}_d^t=(u_1,u_2,...,u_L,d_1,d_2,...,d_\Delta) \tag{2.26}$$

und die erweiterte Matrix des Gleichungssystems wird zu:

$$\mathbf{A}_d=(\vec{a}_d(1),\vec{a}_d(2),...,\vec{a}_d(K)) \tag{2.27}$$

mit den $L+\Delta$ dimensionalen Vektoren:

$$\vec{a}_d^t(k) = \left(a_{d1}(k), a_{d2}(k), ..., a_{dl}(k), (1-q(k))^1, (1-q(k))^2, ..., (1-q(k))^\Delta\right) \quad (2.28)$$

Abb. 3.10 zeigt optimierte Polynome mit unterschiedlicher Anzahl an Koeffizienten. Je höher der Grad des Polynoms, desto geringer sollte theoretisch die Differenz $\left|\left|\mathbf{A}^t\vec{u} - \vec{f}(\vec{q})\right|\right|$ sein. Da jedoch nur mit endlicher Genauigkeit gearbeitet wird, ist es nicht sinnvoll, beliebig viele Koeffizienten zu verwenden. Bei einem Polynom von hohem Grad treten in der Matrix $\mathbf{A}$ sehr kleine Zahlen auf. Bei Verwendung von N Kontaktenergieparametern u_n, enthält Zeile $N+\delta$ den Wert $(1-q)^\delta$. Der Wert für den *Overlap* q kann zwischen 0 und 1 liegen. Wird für die Erzeugung der *Decoys Threading* und das C_α Kontaktkriterium verwendet, so liegt die große Mehrzahl der Werte q zwischen 0.2 und 0.5. Das Element a_{ij}^2 der Matrix $\mathbf{A}^2$ enthält die Summen $a_i \cdot a_j$ aller *Decoys*, wobei i und j über alle Parameter geht. Die Matrix $\mathbf{A}^2$ enthält also Werte, die sich um viele Größenordnungen voneinander unterscheiden können.
Um das Auftreten linearer Abhängigkeiten zwischen den Polynomen zu verhindern, werden *shifted Legendre* Polynome verwendet. Im Bereich [0,1] erfüllen diese Polynome zueinander die Orthogonalitätsrelation:

$$\int_0^1 f_n(x) f_m(x) dx = \delta_{nm} \quad (2.29)$$

lineare Abhängigkeiten dieser Polynome untereinander sind also ausgeschlossen.
In der Matrix $\mathbf{A}^2$ jedoch können weiterhin lineare Abhängigkeiten auftreten. Tritt ein gegebener Kontakt überhaupt nicht auf, so enthält Matrix $\mathbf{A}^2$ eine entsprechende Zeile, die nur Nullen enthält. Somit ist die Matrix singulär. Die Matrix kann aber auch „fast“ singulär sein, wenn z.B. eine Zeile nur sehr kleine Zahlen enthält. Eine solche Matrix kann zu sehr ungenauen, bis hin zu vollkommen falschen Ergebnissen bei der Invertierung führen. Um derartige Probleme zu vermeiden, wird für die Invertierung die Singulärwertzerlegung verwendet (siehe 2.10.2).

Die Matrix $\mathbf{A}$ erreicht schon bei mittelgroßen Sätzen an Strukturen Dimensionen, die nicht mehr problemlos gehandhabt werden können. Im Falle von 210 Energieparametern, müssen pro Struktur 210 Werte vom Typ „short Integer“ gespeichert werden. Wird für die Erzeugung von Strukturen das Set_{1014} verwendet, so liegen ca. $24.9 \cdot 10^6$ Strukturen vor, was heisst, dass rund 10GByte Speicher notwendig sind. Hat man soviel

Arbeitsspeicher nicht zur Verfügung, so müsste die Matrix auf Festplatte gespeichert und später wieder gelesen werden, was extrem zeitaufwendig wäre. Da das Element a_{ij}^2 der Matrix $\mathbf{A}^2$ jedoch nur die Summe aller $a_i \cdot a_j$ der einzelnen *Decoys* enthält, lässt sich $\mathbf{A}^2$ sukzessive aufbauen, indem in jedem Schritt die Produkte $a_i \cdot a_j$ eines *Decoys* aufaddiert werden. Hierfür muss immer nur ein Decoy im Speicher gehalten werden. Wird als Funktion des *Overlaps* das Polynom verwendet, so stehen die variablen q-abhängigen Koeffizienten in der Matrix $\mathbf{A}$. Der Vektor $\vec{f}(\vec{q})$ enthält nur noch den konstanten Koeffizienten d_0, alle Elemente sind also gleich. Somit enthält der Vektor:

$$\begin{gathered} \vec{g}(\vec{q}) = \mathbf{A} \cdot \vec{f}(\vec{q}) \\ \text{die Elemente} \\ g_i = \sum_{j=1}^{k} \mathbf{A}_{ji} \end{gathered} \tag{2.30}$$

Das Element g_i enthält also die Summe aller Kontakte vom Typ i. Wird ein Decoy erzeugt, müssen somit nur die verschiedenen Kontakttypen zum Vektor $\vec{g}$ hinzuaddiert werden. Die Matrix $\mathbf{A^2}$ und der Vektor $\vec{g}$ enthalten alle zur Berechnung der Energieparameter nötigen Informationen. Die vorgestellte Methode lässt sich problemlos auf verschiedene Varianten der Energiefunktion anwenden. Werden z.B. zwei unterschiedliche Sequenzabstandsbereiche verwendet (siehe 2.2.2), so verdoppelt sich die Anzahl der Parameter und das Gleichungssystem wird entsprechend erweitert.
Ein großer Vorteil der vorgestellten Optimierungsmethode liegt darin, dass bei Verwendung des Polynoms für die Funktion des *Overlaps* alle Informationen über einen gegebenen Satz an *Decoys*, die für die Berechnung von $\vec{u}$ benötigt werden, in der Matrix $\mathbf{A}^2$ und dem Vektor $\vec{g}$ gespeichert sind. Wie viele *Decoys* vorliegen ist hierbei unerheblich. Will man verschiedene Sätze an *Decoys* kombinieren, so werden die entsprechenden Matrizen $\mathbf{A}^2$ und Vektoren $\vec{g}$ einfach aufaddiert. Auf diese Weise kann man mit verschiedenen Methoden *Decoys* erzeugen und diese für die Berechnung der Energieparameter $\vec{u}$ verwenden, ohne dass große Datenmengen gespeichert werden müssen.

2.10.1 Wichtungsfaktoren

Um die Erkennung von nativen Strukturen weiter zu verbessern, können die Strukturen unterschiedlich gewichtet werden. So ist es z.B. möglich, die Sequenz/Struktur Paare zu verschiedenen Sequenzen mit unterschiedlichen Gewichten zu belegen. Wird eine native Struktur nicht erkannt, so können alle Strukturen zu der entsprechenden

Sequenz höher gewichtet werden, um die Erkennung eventuell noch zu erreichen. Es bietet sich an, z.B. mit einem iterativen Verfahren die Gewichte sukzessive zu verändern:

$$w^{(i+1)} = \begin{cases} w^{(i)} + \Delta w & : \quad \text{wenn} \quad E_{min} <= E_{nat} \\ w^{(i)} & : \quad \text{wenn} \quad E_{min} > E_{nat} \end{cases}$$

w: Gewicht aller Strukturen zu Sequenz w
Δw: verwendetes Inkrement für die Erhöhung der Gewichte
E_{min}: minimale Energie aller Decoys
E_{nat}: Energie der nativen Struktur

Für die Erzeugung der Energieparameter müssen die Teilmatrizen $\mathbf{A^2}$ sowie der Vektor $\vec{g}$ zu den einzelnen Sequenzen nur einmal am Anfang erzeugt werden. Danach werden die Matrizen und Vektoren zu den verschiedenen Sequenzen mit den entsprechenden Gewichten aufaddiert. Das Aufaddieren und darauffolgende Berechnen der Energieparameter $\vec{u}$ kostet bei moderater Dimension von $\vec{u}$ nur wenig Rechenzeit. Beträgt die Anzahl der Energieparameter $N = 210$, so spielt diese Rechenzeit praktisch keine Rolle. Für das Set_{135} werden auf einem 1.1GHz Athlon PC ca. 20s benötigt. Natürlich muss nachdem die Berechnung der Energieparameter $\vec{u}$ abgeschlossen ist, geprüft werden, wie gut diese bei der Erkennung der nativen Strukturen funktionieren. Dafür werden zu allen Sequenzen solange *Decoys* erzeugt und die Energien berechnet, bis entweder ein Decoy gefunden wird für den gilt $E \leq E_{nat}$, oder für alle *Decoys* gezeigt wurde, dass $E > E_{nat}$. Dies dauert für Set_{135} bei gleicher Gewichtung aller Strukturen (Erkennung 70%) ca. 3.5min. Sind zu den einzelnen Sequenzen die Matrix $\mathbf{A}^2$ und der Vektor $\vec{g}$ einmal erzeugt, was ca. 30min. dauert, so lassen sich die Iterationen zügig durchführen.

2.10.2 Singulärwertzerlegung

Die Singulärwertzerlegung erlaubt die Invertierung von schlecht konditionierten Matrizen. Die problematischen Bereiche werden hierbei erkannt und separiert. Für den Rest der Matrix lässt sich dann eine sinnvolle Invertierung durchführen. Der Singulärwertzerlegung liegt die Tatsache zugrunde, dass jede reell-symmetrisch, positiv semidefinite Matrix $\mathbf{A^2}$ sich darstellen lässt als:

$$\mathbf{A^2} = \mathbf{U} \cdot \mathbf{W} \cdot \mathbf{V^T} \tag{2.31}$$

Die Diagonalelemente der Matrix **W** sind entweder positiv oder Null (die *Singulärwerte*). Die Konditionszahl ist definiert als das Verhältnis von größtem zu kleinstem Element von **W**. Ist das Verhältnis unendlich groß, so ist die Matrix singulär, ist das Verhältnis größer als die reziproke Maschinengenauigkeit, so ist die Matrix schlecht konditioniert. Die entsprechenden Berechnungen werden in dieser Arbeit mit Variablen vom Typ „double“ durchgeführt, die reziproke Konditionszahl sollte also nicht kleiner als 10^{-12} sein. Ist ein $\frac{1}{w_j}$ größer als ein vorgegebenes Vielfaches von $\frac{1}{w_{max}}$, so wird $\frac{1}{w_j}$ gleich Null gesetzt. Es liegt also ein *Singulärwert* vor. Liegt die Singulärwertzerlegung einer Matrix **A** vor, so lässt sich die Berechnung der Inversen $\mathbf{A}^{-1}$ einfach durchführen. Da **U** und **V** orthogonal sind, gilt $\mathbf{U}^{-1} = \mathbf{U}^{\mathbf{T}}$ und $\mathbf{V}^{-1} = \mathbf{V}^{\mathbf{T}}$. Bei der Matrix **W** handelt es sich um eine Diagonalmatrix, die Inverse ist also ebenfalls eine Diagonalmatrix und enthält als Diagonalelemente die reziproken Elemente von **W**. Aus Gleichung 2.31 ergibt sich also:

$$\mathbf{A}^{-1} = \mathbf{V} \cdot \left[\mathrm{diag}\left(1/w_j\right)\right] \cdot \mathbf{U}^{\mathbf{T}} \tag{2.32}$$

2.11 Erzeugen der Energieparameter mittels einer quasichemischen Methode

Die in 2.10 vorgestellte Methode zielt darauf ab, ein lineares Gleichungssystem zu lösen. Energieparameter lassen sich auch über Betrachtungen der Häufigkeiten von Kontakten in nativen Proteinen (n) und nicht-nativen Strukturen (*Decoys*) (d) generieren, z.B. über die einfache Beziehung:

$$u_i = log_e \frac{N_d(i)}{N_n(i)} \tag{2.33}$$

hierbei sind $N_d(i)$ und $N_n(i)$ die relativen Häufigkeiten der verschiedenen Kontakte i in *Decoys* und nativen Strukturen:

$$N_d(i) = \frac{\sum_{k=1}^{K^{(d)}} n_i^{(d)}(k)}{\sum_{k=1}^{K^{(d)}} \sum_{l=1}^{L} n_l^{(d)}(k)} \qquad N_n(i) = \frac{\sum_{k=1}^{K^{(n)}} n_i^{(n)}(k)}{\sum_{k=1}^{K^{(n)}} \sum_{l=1}^{L} n_l^{(n)}(k)} \tag{2.34}$$

$K^{(d)}$: Anzahl der *Decoys*

$K^{(n)}$: Anzahl der nativen Strukturen

L: Anzahl der Kontaktenergieparameter

$n_i^{(d)}(k)$: Anzahl der Kontakte vom Typ i in Decoy k
$n_i^{(n)}(k)$: Anzahl der Kontakte vom Typ i in der nativen Struktur k

Anders als bei der linearen Optimierung hängt die Skalierung der Parameter hier nicht direkt von der Anzahl der Strukturen ab, da nur relative Häufigkeiten verwendet werden. Ein Nachteil ist, dass das Gewicht für verschiedene Sequenzen von der Anzahl der verwendeten *Decoys* abhängt. Je größer die Anzahl an *Decoys* zu einer gegebenen Sequenz ist, desto größer ist das Gewicht der Sequenz in Gleichung 2.33. Dies wird vor allem dann ein Problem sein, wenn die Anzahl an *Decoys* für die verschiedenen Sequenzen stark variiert. Dies ist am stärksten der Fall bei Set_{45}, wo für die längste Sequenz überhaupt kein Decoy und für die kürzeste Sequenz 1993 *Decoys* vorliegen. Eine Möglichkeit, alle Sequenzen gleich zu gewichten, ist $N_d(a_i)$ und $N_n(a_i)$ für die jeweiligen Sequenzen einzeln zu berechnen und dann die Summen zu bilden:

$$u_i = log_e \frac{\sum_{j=1}^{S} N_d^{s_j}(n_i)}{\sum_{j=1}^{S} N_n^{s_j}(n_i)} \tag{2.35}$$

s_j Sequenz j
S: Zahl der Sequenzen

Bei der hier aufgeführten Methode wird eine Struktur entweder als nativ oder als nicht-nativ behandelt, es gibt keinen Zwischenbereich. Eine Struktur, die nur geringfügig vom nativen Zustand abweicht wird genauso behandelt, wie eine vollkommen ungefaltete. Werden beim Lernen *Decoys* verwendet, die der nativen Struktur relativ ähnlich sind, so kann dies offensichtlich zu Problemen führen. So würde selbst ein Decoy mit einem *Overlap* von $q = 0.99$ als nicht-nativ gelernt werden. Aus diesem Grund erscheint es sinnvoll *Decoys* mit einem *Overlap* größer als ein bestimmter Grenzwert, beim Lernvorgang auszuschließen.
Im Set_{45} gibt es für das längste Protein keinen Decoy. Auch wenn für das Berechnen der Kontaktenergieparameter nur *Decoys* aus einem bestimmten Wertebereich für den *Overlap* q verwendet werden, kommt es vor, dass zu bestimmten Sequenzen kein *Decoy* verwendet wird. Somit ist für

$$N_d(i) = \frac{\sum_{k=1}^{K(d)} n_i^{(d)}(k)}{\sum_{k=1}^{K(d)} \sum_{l=1}^{L} n_l^{(d)}(k)} \tag{2.36}$$

$$\sum_{k=1}^{K^{(d)}} n_i^{(d)}(k) = 0 \quad \text{und} \quad \sum_{k=1}^{K^{(d)}} \sum_{l=1}^{L} n_l^{(d)}(k) = 0$$

In diesem Fall wird hier der Ausdruck $N_d(i)$ weggelassen. Es gilt also:

$$N_d(i) = 0$$

2.12 Strukturvorhersage

Um die Struktur einer gegebenen Sequenz vorherzusagen, benötigt man eine Methode, die sinnvolle Proteinstrukturen generiert sowie die Möglichkeit, diese Strukturen hinsichtlich ihrer Ähnlichkeit zur nativen Struktur zu bewerten. Nach Möglichkeit sollte die Bewertung nicht nur eine Rangfolge innerhalb der Menge der verschiedenen möglichen Strukturen ergeben, sondern auch eine Aussage hinsichtlich des zu erwartenden *Overlaps* q erlauben. Werden die Strukturen mittels der vorgestellten Kontaktenergiefunktion bewertet, so ist eine Abschätzung von q direkt über die Energie nicht ohne weiteres möglich. Man kann die Struktur mit der niedrigsten Energie bestimmen und diese sollte im Verhältnis zu den anderen Strukturen einen hohen Wert von q aufweisen. Ob dieser *Overlap* jedoch tatsächlich nahe $q = 1$ liegt, ist nicht ersichtlich. Eine Möglichkeit einen Hinweis auf den absoluten Wert von q zu erhalten ist der *Z-Score*. Dieser gibt die Stabilität einer Struktur relativ zum Mittelwert der Energien aller Strukturen $\bar{E}$ an:

$$Z = \frac{E - \bar{E}}{\sigma} \quad \text{wobei} \quad \sigma^2 = \left\langle (\bar{E} - E)^2 \right\rangle \tag{2.37}$$

Die native Konformation eines Proteins besitzt einen großen negativen *Z-Score* (Sippl, 1993a, 1993b). Weist eine Struktur unter Verwendung einer gegebenen

Energiefunktion einen negativen *Z-Score* mit hohem Betrag auf, so bedeutet dies eine hohe Stabilisierung gegenüber allen anderen Strukturen. Es liegt also ein Hinweis auf eine native oder zur nativen Struktur ähnliche Struktur vor.

Threading Eine sehr einfache Methode der Strukturvorhersage ist die Verwendung von *Threading* in Verbindung mit einer Kontaktenergiefunktion. Mittels *Threading* lässt sich eine große Zahl von Strukturen für eine gegebene Sequenz generieren, die dann über die Kontaktenergiefunktion bewertet werden. Die energieärmste Struktur dient als Vorhersage. Über den *Z-Score* kann die Qualität der Vorhersage eingeschätzt werden. Hierbei können natürlich nur Faltungsmotive vorhergesagt werden, die bereits bekannt sind. Wird versucht ein Protein mit einem neuartigen Faltungsmotiv vorherzusagen, so wäre es wünschenswert wenn z.B. über den *Z-Score* erkennbar ist, dass keine sinnvolle Vorhersage vorliegt. Eine Vorhersage muss nicht für eine vollständige Sequenz erfolgen. Es können auch Strukturen von Teilsequenzen vorhergesagt werden. Je länger eine Sequenz ist, desto geringer ist die Wahrscheinlichkeit, dass mittels *Threading* eine Struktur mit nativen Eigenschaften gefunden wird. Gibt es eine Möglichkeit, die Qualität einer Vorhersage einzuschätzen, so kann dies genutzt werden, um Fragmente mit sinnvoller Struktur zu bestimmen. Der Nachteil hierbei ist natürlich, dass Wechselwirkungen nur innerhalb des Fragmentes berücksichtigt werden.

Monte Carlo Simulation Eine weitere Methode für die Strukturvorhersage ist die Monte Carlo Simulation im Raume der (ϕ, ψ)-Torsionswinkel. Hierbei ist man im Prinzip nicht darauf angewiesen, dass eine untersuchte Sequenz ein bekanntes Faltungsmotiv besitzt. Da jedoch nicht wie beim *Threading* von vornherein zum großen Teil physikalisch sinnvolle Strukturen erzeugt werden, sind einige Ergänzungen der einfachsten Form der Kontaktenergiefunktion notwendig (siehe 2.5).

CASP **- Vergleich von verschiedenen Methoden zur Proteinstrukturvorhersage** 1994 fand erstmals das „*Critical Assessment of techniques for protein Structure Prediction*"(*CASP*) (http://predictioncenter.llnl.gov/) statt. Hier werden verschiedene Strukturvorhersagemethoden an „echten“ Vorhersagen getestet. „Echt“ heisst hierbei, dass zu den verschiedenen vorgegebenen Sequenzen noch keine Strukturen veröffentlicht sind. Verwendet werden Sequenzen, deren Strukturen aller Voraussicht nach in Kürze publiziert werden, so dass die Vorhersagen mit den experimentellen Strukturen verglichen werden können.
Ich habe mit Vorhersagen unter Verwendung einer Energiefunktion, die mit der

Boltzmann-gewichteten Optimierung (siehe 2.9) erzeugt wurde, an CASP4 teilgenommen. Sowohl für das Training der Energiefunktion als auch zum Erzeugen der Vorhersagen wird *Threading* verwendet. Für das Training wird das C_α Kontaktkriterium verwendet, für die Vorhersagen das *all atom* Kriterium. Die Vorhersage zu einer gegebenen Sequenz wird über den *Z-Score* beurteilt. Das gleiche geschieht für Fragmente der Sequenz. Als weiteres Mittel zum Abschätzen der Vorhersagen dient ein Vergleich mit Sekundärstrukturvorhersagen (Salamov & Solovyev, 1995). Die Ergebnisse finden sich unter http://predictioncenter.llnl.gov/casp4/Casp4.html (*Group number 492*).