Institut für Organische Chemie und Biochemie
der Technischen Universität München

Automatisierte Zuordnung von heteronuklearen Protein-NMR-Spektren

Jens Liermann

Vollständiger Abdruck der von der Fakultät für Chemie der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. St. Glaser
Prüfer der Dissertation:

1. Univ.-Prof. Dr. H. Kessler
2. Univ.-Prof. Dr. W. Hiller

Die Dissertation wurde am 18.12.2000 bei der Technischen Universität München eingereicht und durch die Fakultät für Chemie am 16.01.2001 angenommen.

Jabberwocky

'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy where the borogoves,
And the mome raths outgrabe.

Lewis Carrol, *Through the Looking Glass*

Folgenden Personen möchte ich im Rahmen dieser Arbeit besonders danken:

- Prof. Dr. H. Kessler für die Bereitstellung der hervorragenden Möglichkeiten, das Interesse an meiner Arbeit und den Freiraum bei der Gestaltung des Themas.

- PD Dr. G. Gemmecker (dem „Gustav“) für die vielen hilfreichen Diskussionen zum Thema (oder einfach auch nur so), sein ständig offenes Ohr für Probleme aller Art, das sorgfältige Korrekturlesen dieser Arbeit, ... und nicht zuletzt für meine Stelle.

- Dr. Murray Coles für die Einführung in die Praxis der Zuordnung von Protein-NMR-Spektren, sowie die zahlreichen Diskussionen, Hinweise und Tips, die bei der Entwicklung von PASTA Toolkit eingeflossen sind.

- Dr. Michael Leutner, der das ursprüngliche Konzept von PASTA entwickelte, auf dem diese Arbeit aufbaut.

- Dr. Ralph Peteranderl für die bereitwillige Beantwortung aller biochemischen und biologischen Fragen, die nachmittäglichen Eispausen und seine Hilfsbereitschaft.

- Dr. Rainer Haessner für Rat und Tat in allen Rechnerbelangen und den langen Geduldsfaden beim Entleihen von Japanischbüchern.

- Murray Coles, Tammo Diercks, Adriane Gröger, Martin Huenges, Michael John, Manfred Schwaiger und Vincent Truffault für die lockere Atmosphäre im „NMR-Zimmer“ und so manche Konzentrationspause. ;-)

- dem AK Kessler/AK Glaser für Spiel und Spaß. ;-)

- und besonders meinen Eltern, ohne die ich das ganze wahrscheinlich nie geschafft hätte.

Inhaltsverzeichnis

Abkürzungen

AAA	*ATPase associated with a variety of cellular activities*
ANSI	*american national standard institute*
BMRB	*biomagnetic research bank*
COSY	*correlation spectroscopy*
CSI	*chemical shift index*
CRINEPT	*cross-correlated cross-relaxation based polarization transfer*
DHFR	Dihydrofolatreduktase
DNS	Desoxyribonukleinsäure
DOSY	*diffusion ordered spectroscopy*
E.COSY	*exclusive correlation spectroscopy*
E. coli	Escherichia coli
Fast TA	*fast threshold accepting*
HNGAL	Humanes neutrophiles gelatinase-assoziiertes Lipocalin
HSQC	*heteronuclear single quantum coherence* (heteronukleare Einquantenkohärenz)
kDa	Kilo-Dalton (10^3 g/mol)
mM	10^{-3} mol/l (Konzentrationsangabe)
NADPH	Nicotinamid-Adenin-Dinucleotid-Phosphat
NMR	*nuclear magnetic resonance* (Kernresonanz)
NOE	*nuclear Overhauser-effect* (Kern-Overhauser-Effekt)
NOESY	*nuclear Overhauser spectroscopy*
PASTA	*protein assignment by threshold accepting*
PDB	*protein data bank*
ppm	*parts per million* (10^{-6})
RiSy	Riboflavinsynthase
RMSD	*root mean square deviation* (Wurzel des mittleren Fehlerquadrats)
SAR	*structure activity relationship* (Struktur-Wirkungs-Beziehung)
TOCSY	*total correlation spectroscopy*
TROSY	*transverse relaxation optimized spectroscopy*
[U-^{13}C-^{15}N]	vollständig ^{13}C-^{15}N-isotopenmarkiert
VAT	VCP-artige ATPase aus Thermoplasma

1 Einleitung und Aufgabenstellung

Eine Großzahl von Krankheiten wird durch Störungen des Proteinhaushalts verursacht [1-4], so dass Proteine ideale Angriffspunkte für die Suche nach neuen Medikamenten darstellen . Das Interesse an der Erforschung von Proteinen und deren biochemischen Funktionen hat daher im letzten Jahrzehnt rasant zugenommen. Modernste Techniken ermöglichten die Aufklärung des menschlichen Genoms, das als Grundlage für weitere Forschungen dient [5]: Mit Hilfe bioinformatischer Methoden [6, 7] lassen sich aus diesen Daten eine Vielzahl von interessanten Proteinen identifizieren, deren Funktionsweise und Struktur noch geklärt werden muss [8, 9].

Grundlage der modernen Wirkstoffentwicklung ist in vielen Fällen eine hochaufgelöste 3D-Struktur der entsprechenden Zielmoleküle. Beispielsweise werden die Wechselwirkungen potentieller Liganden und Rezeptoren beim strukturbasierten, rationalen Wirkstoffdesign anhand der 3D-Struktur erforscht, um so Leitstrukturen für neue Arzneistoffe zu entwickeln und zu optimieren.

Neben der Röntgenkristallographie hat sich hier insbesondere die magnetische Kernresonanz-(NMR)-Spektroskopie als ein wertvolles Hilfsmittel zur Charakterisierung und Strukturbestimmung von Molekülen etabliert. Im Vergleich zur Röntgenkristallographie erlaubt die NMR-Spektroskopie jedoch Proteinuntersuchungen in gelöstem Zustand. Zusätzlich können Flexibilität und Dynamik eines Moleküls gemessen werden, was insbesondere bei der Aufklärung der biochemischen Funktion von Proteinen wichtig ist. Neue experimentelle und technologische Fortschritte im Bereich der NMR-Spektroskopie wie SAR (*structure activity relationship*) by NMR [10], SHAPES-screening [11] und DOSY-(*diffusion-ordered*)-Spektroskopie [12] unterstreichen zusätzlich die Bedeutung von NMR in der Wirkstoffforschung.

Mit der Genomaufklärung hat sich auch das Bild der Strukturbestimmung von Proteinen gewandelt: *High-Throughput*-Methoden in der Kristallographie [13] und NMR-Spektroskopie [14, 15] ermöglichen die Akquisition sehr großer Datenmengen in kurzen Zeiträumen. Zur Unterstützung der Auswertung, Verwaltung und Klassifizierung dieser Daten während der Strukturaufklärung werden vielfältige Automatisierungsansätze [8, 16, 17] benötigt.

Zielsetzung dieser Arbeit war es, neue Automatisierungsmethoden und Zuordnungsstrategien für die Auswertung von heteronuklearen Protein-NMR-Spektren zu entwickeln und an verschiedenen Proteinsystemen zu testen.

Aufbauend auf den Arbeiten von Michael Leutner [18] sollte das Programm PASTA [19] unter besonderer Berücksichtigung der Erweiterbarkeit und Benutzerfreundlichkeit weitergeführt werden.

Aus den Konzepten von PASTA sollte schließlich ein vollkommen neues, streng modular aufgebautes Programmpaket, PASTA Toolkit, erstellt werden, das den Anwender interaktiv über den gesamten Prozess der Rückgratzuordnung begleitet.

Beide Programme sollten während der NMR-Zuordnung und Strukturaufklärung der N-terminalen Domäne des AAA-Proteins VAT eingesetzt werden.

Schließlich sollte ein neuartiges Konzept für die NMR-Zuordnung des NADPH-Komplexes von DHFR aus *E. coli* unter Verwendung der bereits bekannten Zuordnung des strukturhomologen Folat-Komplexes entwickelt werden.

2 Zuordnungsstrategien

2.1 Allgemeines

Die klassische Strategie zur Strukturaufklärung von Proteinen mit Kernresonanz-Spektroskopie wurde in den achtziger Jahren von Wüthrich entwickelt [20]. Nach dieser Methode werden zuerst die Protonenspinsysteme der einzelnen Aminosäuren mit COSY- und TOCSY-Experimenten charakterisiert. Anschließend erfolgt die sequentielle Verknüpfung über interresiduale Kontakte aus NOESY-Spektren. Dieser Ansatz ermöglicht die Aufklärung von Molekülen bis etwa 10 kDa unter ausschließlicher Verwendung 2-dimensionaler NMR-Spektren. Oberhalb dieser Grenze steigt die Zahl der Signalüberlagerungen stark an und macht damit eine Zuordnung unmöglich. Außerdem nimmt die Halbwertszeit der Kohärenzen mit steigender Molekülgröße stark ab. Da die Linienbreite indirekt proportional zur transversalen Relaxationszeit T_2 ist, wächst sie mit der Molekülgröße. Dieser Effekt verstärkt die Überlagerungsproblematik zusätzlich.

Die Einführung der Isotopenmarkierung von Proteinen, insbesondere ^{15}N- und ^{13}C-Markierung, ermöglicht eine Erweiterung der ursprünglichen Strategie. Neben günstigeren Relaxationseigenschaften (durchschnittliche Werte eines 25 kDa Proteins [21]: $T_2(^{15}N) = 60$ ms, $T_2(^{13}C) = 16$ ms), $T_2(^{1}H) = 12$ ms) sind die Heterokerne durch vergleichsweise große ^{1}J-Kopplungen (^{1}H-^{13}C = 120-160 Hz, ^{1}H-$^{15}N \approx 92$ Hz, ^{13}C-^{13}C = 30-55 Hz, ^{13}C-^{15}N = 9-15 Hz [22]) miteinander verknüpfbar. Somit lässt sich eine Reihe neuer, überwiegend dreidimensionaler Experimente einführen. Mit dieser Technik lassen sich Proteine bis ca. 25 kDa charakterisieren.

Jenseits dieser Größe können die Relaxationszeiten durch Probendeuterierung bzw. partielle Deuterierung [23] dramatisch verringert werden ($\gamma_D \sim 1/5{,}5\ \gamma_H$). Selektive Isotopenmarkierung des Proteinrückgrats [24] führt zu weniger Signalen und vermeidet damit Überlagerungen. TROSY-/CRINEPT-Techniken [25, 26] und die 4D-Spektroskopie [27] sowie gerätetechnische Verbesserungen (höhere Feldstärke, Kryoprobenköpfe, Gradienten, ...) erlauben die Untersuchung immer größerer Proteine. Ebenso gewinnen nicht NOE-basierte Strategien, wie die Untersuchung residualer dipolarer Kopplungen und kreuzkorrelierte Relaxationen, an

Bedeutung [28]. Die Grenze der momentan untersuchbaren Molekülgröße liegt damit etwa bei 65 kDa [26, 29, 30].

2.2 Zuordnung des Proteinrückgrats

Im Gegensatz zur protonenbasierten Strategie, bei der die Spinsysteme der Seitenketten eine entscheidende Rolle bei der sequentiellen Zuordnung spielen, benutzen moderne heterokern-basierte Techniken Kopplungen im Proteinrückgrat als wesentlichen Zuordnungsschritt. Üblicherweise werden eine Reihe von Tripelresonanzspektren verwendet. Unter der Voraussetzung einer $^{13}C/^{15}N$-isotopenmarkierten Proteinprobe lassen sich auf diese Weise alle Kerne des Protein-Rückgrats messen. Im folgenden sind die gebräuchlichsten Experimente aufgeführt [Abbildung 1]: HNCO [31], HN(CA)CO, HNCA [31], HN(CO)CA [32], HNCACB [21], CBCA(CO)NH [33], HNHA [34], HN(CA)HA [35, 36] und HNHB [37]. Der Name des Experiments gibt den Weg des Kohärenztransfers während des Experiments wieder. Alle Kerne, deren Verschiebung nicht als Dimension in das Spektrum eingehen, werden in Klammern angegeben.

Im ersten Schritt der Zuordnung des Proteinrückgrates werden die Verschiebungen in Spinsysteme gruppiert. Fast alle gebräuchlichen Spektren beinhalten die ^{15}N- und H^N-Verschiebung der Aminosäuren. Daher dient dieses Verschiebungspaar als Referenz bzw. Ankerwert für die Spinsysteme. Die weiteren Verschiebungen werden nach und nach basierend auf diesem Wertepaar ergänzt. Als Ausgangsspektrum dient meist das HNCO. Dieses Experiment ist äußerst empfindlich und eignet sich daher besonders gut zur Konstruktion der Spinsysteme.

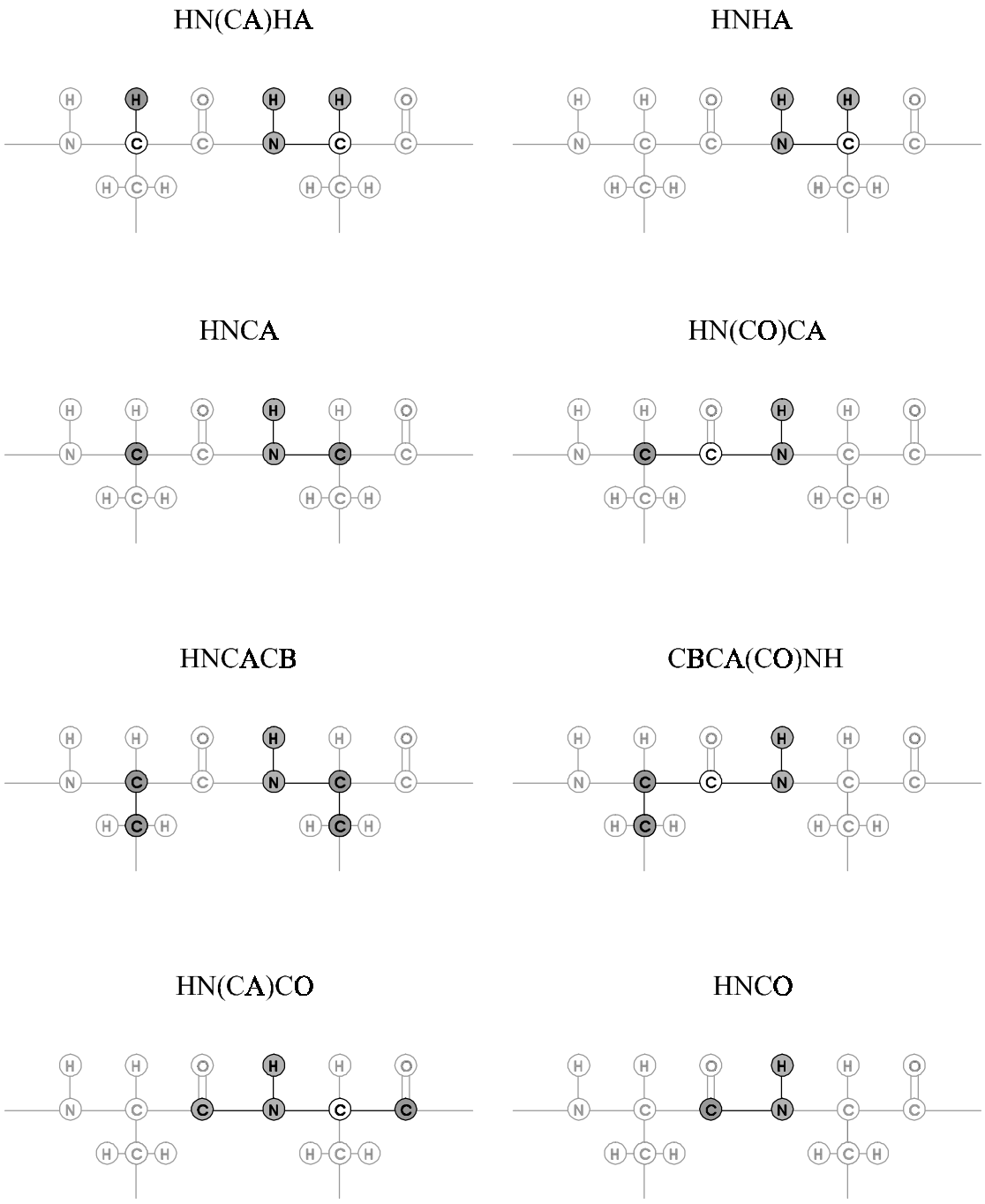

Abbildung 1: *Gebräuchliche Zuordnungsexperimente. Meist werden komplementäre Spektren verwendet, die sowohl Resonanzen der Aminosäure [i] als auch der Nachbaraminosäure [i-1] liefern und sich damit gegenseitig ergänzen.*

Die sequentielle Verknüpfung der Spinsysteme erfolgt im Allgemeinen über komplementäre Spektren, die selektiv die verschiedenen Kerne des Proteinrückgrates miteinander korrelieren. Dies sei am Beispiel des HNCA-Spektrums erklärt. Das HNCA-Spektrum enthält die Verschiebungen der Amidgruppe und des α-Kohlenstoffes der eigenen Aminosäure [i] und

die α-Kohlenstoff-Verschiebung der jeweilig benachbarten Aminosäure [i-1]: $H^N_{[i]}$, $N_{[i]}$, $C^\alpha_{[i]}$ und $C^\alpha_{[i-1]}$. Das komplementäre Spektrum zum HNCA ist das HN(CO)CA. Es enthält neben $H^N_{[i]}$ und $N_{[i]}$ selektiv nur $C^\alpha_{[i-1]}$. Auch ohne das komplementäre Spektrum lassen sich [i] und [i-1] meist unterscheiden, weil die [i]-Signale einer Aminosäure aufgrund ihrer größeren J-Kopplungskonstante eine höhere Intensität als deren [i-1]-Signale aufweisen. Eine zweifelsfreie Unterscheidung ist jedoch nur mithilfe des komplementären Spektrums möglich. Die sequentielle Anordnung der N/H^N-Paare erfolgt durch einen Abgleich der [i]- und [i-1]-Verschiebungen. Aufgrund der Überlagerung der C^α-Verschiebungen werden die Spinsysteme noch durch weitere Spektren-Kombinationen ergänzt [Abbildung 1]. Mit den obengenannten Experimenten stehen insgesamt folgende Verschiebungen zur Sequenzierung zur Verfügung: C^α, C^β, C', H^α und H^β. Davon besitzen die C^α- und C^β-Verschiebungen die größte Bedeutung für die Zuordnung, da sie über eine vergleichsweise große Dispersion verfügen ($C^\alpha \approx 25$ ppm, $C^\beta \approx 60$ ppm). Die C^α- und C^β-Verschiebungen lassen sich auch für die Aminosäureerkennung nutzen, da viele Aminosäuren charakteristische Kombinationen dieser beiden Werte besitzen.
Das wertvollste Experiment für die Zuordnung ist das HNCACB. Es enthält gleichzeitig $C^\alpha_{[i]}$, $C^\alpha_{[i-1]}$ und $C^\beta_{[i]}$, $C^\beta_{[i-1]}$. Die Signale von C^α und C^β lassen sich aufgrund unterschiedlicher Phasen einfach unterscheiden, [i] und [i-1] kann anhand der Intensität getrennt werden. Es ist damit theoretisch allein ausreichend für die Sequenzierung. In der Praxis sind jedoch die [i-1]-Signale häufig intensitätsschwach, so dass sich zumindest die Ergänzung mit einem CBCA(CO)NH-Spektrum empfiehlt.
Prolin ist mit den oben genannten N/H^N-basierten Experimenten aufgrund des fehlenden Amidprotons nicht zu erfassen. Die Prolinverschiebungen werden daher meistens indirekt über die [i-1] Verknüpfung (z.B. mit einem CBCA(CO)NH) ermittelt. An Sequenzstellen, an denen mehrere Prolinreste hintereinander folgen und ein solches Vorgehen nicht möglich ist, können die Verschiebungen auch mit prolinspezifischen Experimenten ermittelt werden [38, 39].

2.3 Seitenkettenzuordnung

Als klassische Experimente zur Seitenkettenzuordnung werden das HCCH-TOCSY [40], das C(CCO)NH-TOCSY [41] und das H(CCCO)NH-TOCSY [42] verwendet. Das HCCH-TOCSY liefert die Korrelationen sämtlicher Kohlenstoffe und Protonen der Seitenketten. Das

Experiment verfügt zwar im Allgemeinen über eine gute Signalintensität, ist aber wegen starker Signalüberlagerungen relativ schwierig auszuwerten. Die Verbindung der Seitenkettendaten zum Proteinrückgrat erfolgt entweder über die H^{α}/H^{β}- und C^{α}/C^{β}-Verschiebungen (diese sind normalerweise von der Rückgratzuordnung bekannt) oder über die N/H^{N}-Verschiebungen der C-terminal benachbarten Aminosäure. Dazu wird das HCCH-TOCSY häufig mit dem C(CCO)NH-TOCSY, das selektiv alle Kohlenstoffe der einzelnen Spinsysteme in Korrelation zu $N/H^{N}[i+1]$ zeigt, und dem H(CCCO)NH-TOCSY, das nur die Protonen der Spinsysteme in Korrelation zu $N/H^{N}[i+1]$ enthält, kombiniert. Da für diese beiden Experimente aber ein Magnetisierungstransfer über eine große Anzahl von Kernen notwendig ist, ist die Signalintensität bei Proteinen ab ca. 15 kDa aufgrund ungünstiger Relaxationseigenschaften oft sehr niedrig.

Alternativ bietet sich in diesem Fall das HCCH-COSY [43] zur Ergänzung des HCCH-TOCSY an. Dieses Spektrum zeigt im Gegensatz zum HCCH-TOCSY, in dem die Weitbereichskorrelationen des gesamten Spinsystems zu sehen sind, nur die Korrelationen zwischen benachbarten Seitenketteneinheiten. Die Signalintensität des HCCH-COSY ist mit der des HCCH-TOCSY vergleichbar. Die Kombination dieser beiden Spektren ist für die Zuordnung einer Großzahl der Seitenkettenresonanzen im Allgemeinen ausreichend.

Für die spätere Strukturrechnung ist zusätzlich eine stereospezifische Unterscheidung der diastereotopen Seitenkettenprotonen erforderlich. Für die H^{β}-Protonen können dazu die $^{3}J(H^{N},H^{\beta})$-Kopplungskonstanten in Verbindung mit intraresidualen NOE-Mustern der H^{α}- und H^{β}-Kerne verwendet werden. Die $^{3}J(H^{N},H^{\beta})$-Kopplungskonstanten lassen sich einfach aus den Signalintensitäten im HNHB-Spektrum abschätzen. Für jedes der drei möglichen Rotamere ergibt sich so eine spezifische Kombination, anhand deren es identifiziert werden kann.

Die diastereotope Zuordnung der Methylgruppen der Aminosäuren Valin und Leucin lässt sich einfach mit einer biosynthetisch fraktionell ^{13}C-markierten Proteinprobe (ca. 20% ^{13}C) durchführen. Die *pro R*- und *pro S*-Methylgruppen können schließlich in einem ^{1}H-entkoppelten ^{13}C-HSQC anhand der Signalform unterschieden werden (*pro R*: Dublett, *pro S*: Singulett) [44]. Alternativ kann die Stereo-Zuordnung ebenfalls wieder über charakteristische NOE-Kontakte erfolgen. Häufig ist dies erst während der Strukturrechnung der Fall.

Die Seitenketten-Amidprotonen der Aminosäuren Asparagin und Glutamin können schließlich mit Hilfe eines speziellen Experiments, dem H_2NCO-E.COSY [45], diastereotop unterschieden werden. Zur Zuordnung der diastereotopen Protonen an primären

Carboxamidgruppe werden die Kopplungskonstanten $^3J(H^{\delta},C^{\beta})$ für Asparagin bzw. $^3J(H^{\varepsilon},C^{\gamma})$ für Glutamin herangezogen.

2.4 Sekundärstrukturbestimmung

Nach der Zuordnung der chemischen Verschiebungen des Proteins kann bereits die Sekundärstruktur abgeleitet werden. Eine einfache Methode dazu beruht auf der Analyse der chemischen Verschiebungen. Die Rückgratresonanzen (C^{α}, C^{β}, C' und H^{α}) eines Proteins erhalten abhängig vom vorliegenden Sekundärstrukturelement charakteristische Hoch- bzw. Tieffeldverschiebungen, die sogenannten sekundären chemischen Verschiebungen. Man kann die sekundäre chemische Verschiebung berechnen, indem man den gemessen Verschiebungswert vom durchschnittlichen Verschiebungswert des entsprechenden Kerns, dem random coil-Wert, abzieht. Ein negativer Wert der sekundären chemischen Verschiebung bedeutet also eine Hochfeldverschiebung des Signals, ein positiver Wert eine Tieffeldverschiebung. Die einzelnen Kerne erfahren folgende Veränderungen bei den einzelnen Sekundärstrukturelementen:

	C^{α}	C^{β}	C'	H^{α}
α-Helix	> 0	< 0	> 0	< 0
β-Faltblatt	< 0	> 0	< 0	> 0

Tabelle 1: *Vorzeichen der sekundären chemischen Verschiebungen für C^{α}, C^{β}, C' und H^{α} bei unterschiedlicher Sekundärstruktur.*

Aus diesem Verhalten heraus wurde von Wishart und Sykes das CSI-Verfahren (chemical shift index) entwickelt [46]. Da die chemischen Verschiebungen nicht nur Abweichungen durch die Sekundärstruktur unterworfen sind, sondern auch stark durch lokale Bedingungen beeinflusst werden, werden die vier Parameter (C^{α}, C^{β}, C' und H^{α}) für eine Aussage miteinander kombiniert. Liegt über einen längeren Bereich, üblicherweise drei Aminosäuren, eine konstante Verschiebung des CSI vor, kann angenommen werden, dass das entsprechende

Sekundärstrukturelement an der untersuchten Stelle vorliegt. Aus diesem Grund ist es schwierig, mit der CSI-Methode die genaue Position der Sekundärstrukturelemente festzustellen.

Eine genauere Aussage lässt sich mit Hilfe der $^{3}J(H^{N},H^{\alpha})$-Kopplungskonstanten treffen. Nach der Karplus-Beziehung [47] lassen sich charakteristische Werte für α-Helices und β-Faltblätter angeben [48]:

- α-Helix ($\phi = -57°$) $^{3}J(H^{N},H^{\alpha}) = 3.9$ Hz
- antiparalleles β-Faltblatt ($\phi = -139°$) $^{3}J(H^{N},H^{\alpha}) = 8.9$ Hz
- paralleles β-Faltblatt ($\phi = -119°$) $^{3}J(H^{N},H^{\alpha}) = 9.7$ Hz

Für die genaue Bestimmung der Topologie ist es notwendig, die Verknüpfungen der einzelnen β-Stränge miteinander, bzw. die Art der Verknüpfung (paralleles oder antiparalleles Faltblatt) zu untersuchen. Die erfordert die Zuordnung von Kurzbereichs- bzw. Mittelbereichs-NOE-Kontakten. Für α-Helix, paralleles β-Faltblatt und antiparalleles β-Faltblatt existieren jeweils typische NOE-Muster [49]:

- α Helix:

$d_{NN}(i,i+1)$	stark
$d_{\alpha N}(i,i+1)$	schwach
$d_{\alpha N}(i,i+3)$	mittel
$d_{\beta N}(i,i+1)$	mittel

- β-Faltblatt:

$d_{NN}(i,i+1)$	schwach
$d_{\alpha N}(i,i+1)$	sehr stark
d_{NN}(cross strand)	schwach
$d_{\alpha N}$(cross strand)	sehr stark
$d_{\alpha\alpha}$(cross strand)	sehr stark in antiparallelen β-Faltblättern

2.5 Tertiärstrukturbestimmung

Die Bestimmung der Tertiärstruktur stützt sich für Proteine <30 kDa vorwiegend auf die Auswertung der Daten aus NOESY-Spektren. Der NOE-Effekt (Nuclear-Overhauser-Effekt) entspricht der Intensitätsänderung der Resonanz eines Kerns durch Anregung eines anderen Kerns. Der NOE ist ein reines Relaxationsphänomen und nicht von skalaren Kopplungen abhängig. Er eignet sich damit zur Abstandsermittlung zwischen zwei Kernen in der Proteinstruktur: Die Intensität der NOE-Signale bzw. die Kreuzrelaxationsrate ist proportional zum Faktor r^{-6} (r = Abstand zwischen zwei Kernen). In der Praxis müssen die Intensitäten der Signale anhand eines bekannten Abstandes (z.B. eine intraresiduale Entfernung) kalibriert werden. Mit dem NOE lassen sich Protonenabstände bis zu 5 Å messen.

Für Proteine >30 kDa macht die Spindiffusion die Analyse mittels NOE-Daten schwierig. Für solche Moleküle muss auf alternative Methoden wie beispielsweise die Messung von residualen dipolaren Kopplungen zurückgegriffen werden. Im Gegensatz zur NOE-Methode dürfen die Moleküle dazu jedoch nicht frei beweglich in Lösung gemessen werden. Eine Fixierung der Proteinmoleküle, beispielsweise durch eine Matrix von Bizellen [50] oder Phagen [51], ist erforderlich. Die Verfahren sind präparativ aufwendig, die erhaltenen Informationen sind jedoch oft genauer als die entsprechenden Daten aus dem NOE. Zusätzlich lassen sie auch eine räumliche Charakterisierung von Kernen mit Abständen >5 Å zu.

3 Automatisierung

3.1 Allgemeines

Eine Reihe von Verfahren zur Charakterisierung von Proteinen mit NMR-Spektroskopie sind bisher entwickelt worden [20, 52]. In der Praxis hat sich überwiegend die Strategie von Wüthrich et al. durchgesetzt (vgl. 2). Sie lässt sich in sieben Schritten zusammenfassen, deren Einteilung auch für Automatisierungsverfahren relevant ist[17, 53]:

- Gruppierung der Resonanzen in Spinsysteme und Zuordnung der Verschiebungen auf die einzelnen Kerne.
- Klassifizierung des Spinsystems in Hinsicht auf eine oder mehrere Aminosäuren.
- Suche nach sequentiellen Verbindungen zwischen den Spinsystemen.
- Individuelle Zuordnung der Spinsysteme/Spinsystemcluster auf Sequenzpositionen.
- (Stereospezifische) Zuordnung der Seitenketten.
- Generierung von Distanz-Kriterien aus NOE-Daten, skalaren und dipolaren Kopplungen, Wasserstoff/Deuterium-Austauschraten usw.
- Strukturrechnung unter Verwendung dieser Distanz-Kriterien.

Einzelne Programme übernehmen normalerweise nur Teilschritte des Charakterisierungsprozesses. Es ergibt sich eine Aufteilung in folgende Kategorien: Zuordnung der Rückgratresonanzen, Seitenkettenzuordnung, Sekundärstrukturbestimmung, NOE-Zuordnung und Strukturrechnung. Alle nachfolgenden Aussagen beziehen sich nur auf die Umsetzung von Zuordnungsstrategien in Hinblick auf Automatisierungstechniken und sind nicht unbedingt für eine allgemeine Betrachtung von Zuordnungsstrategien gültig (vgl. 2).

3.2 Rückgratzuordnung

Der Prozess der sequentiellen Zuordnung ist der erste Schritt zur Bestimmung der Proteinstruktur mit Hilfe der Kernresonanzspektroskopie. Das Verfahren ist äußerst zeitaufwendig, verfügt jedoch über eine Reihe repetitiver Teilschritte. Daher bietet es sich die Entwicklung von automatisierten Zuordnungsstrategien an. Folgende fünf Teilschritte lassen sich nach Moseley und Montelione formulieren [17]:

- *peak picking* bzw. *filtering and referencing* (Extrahieren von Peakdaten)
- *grouping* (Einsortieren der Daten in Spinsysteme)
- *typing* (Aminosäureklassifizierung)
- *linking* (Sequentielle Anordnung)
- *mapping* (Abbilden auf die Sequenz)

Für jeden dieser Einzelschritte ist eine Automatisierung möglich. Vielfältige informatische Methoden wie z.B. deterministische *best-first* Ansätze, vollständige Suche, *constraint satisfaction*, genetische Algorithmen, kombinatorische Minimierung und neuronale Netze werden zu diesem Zweck eingesetzt [17, 53]. Im Folgenden ist eine Auswahl an Lösungsansätzen für jeden der fünf Teilschritte beschrieben:

peak picking (*filtering and referencing*):
Der erste Schritt jedes Zuordnungsprozesses ist die Extraktion der Verschiebungsinformationen aus den experimentellen Daten (*peak picking*). Die Bedeutung des *peak picking* ist damit besonders groß, da die Güte der Datenlisten entscheiden für alle nachfolgenden Schritte ist.
Vor der eigentlichen Extraktion müssen die Datensätze auf einen gemeinsamen Standard referenziert werden, um ihre Vergleichbarkeit während des späteren Zuordnungsprozesses zu gewährleisten (*referencing*) [54]. Einige Programme berechnen dafür aus isolierten Signalen, die in allen Spektren enthalten sind, eine selbst-konsistente Referenzierung [55, 56].
Während des *peak picking* werden die einzelnen Signale in den Experimentdaten identifiziert und klassifiziert. Es wird zwischen „echten“ Signalen des untersuchten Moleküls und Artefakten unterschieden (*filtering*).

Das einfachste *peak picking* Verfahren ist die Suche nach lokalen Maxima bzw. Minima. Es ist jedoch äußerst anfällig gegenüber Artefakten wie z.B. t1-Rauschen. Mit diesem Verfahren allein ist es nicht möglich, überlagerte Positionen zu erkennen. Eine Linienformanalyse, z.B. mit dem Levenberg-Marquardt-Algorithmus [57], kann hier Abhilfe schaffen. In der Praxis kann diese jedoch nur für den eindimensionalen Fall effizient durchgeführt werden [58, 59]. Für zwei- oder höherdimensionale Datensätze bieten sich andere Methoden besser zur Klassifizierung der Signale an. Unter anderem werden hier neuronale Netze [60], statistische Methoden [61, 62] oder numerische Analyse [63] der Datenpunkte verwendet. Viele Verfahren versagen dennoch an Stellen mit starker Signalüberlappung oder Artefaktpositionen, da oft nur die unmittelbare Umgebung des lokalen Maximums bzw. Minimums untersucht wird.
Moderne Methoden [64, 65] verwenden im Allgemeinen verschiedene Kombinationen einer ganzen Reihe von Strategien zur Verfeinerung des *filtering* Prozesses:

- Bestimmung des Rauschlevels:
  Um auch intensitätsschwache Peaks identifizieren zu können, ist eine genaue Bestimmung des Rauschlevels unerlässlich. Effiziente Algorithmen führen anstatt einer globalen Analyse eine lokale Untersuchung in einzelnen Spektrenbereichen durch (z.B. jedem 1D-Ausschnitt). Die Klassifizierung der Peaks erfolgt über statistische Methoden [61, 62].
- Segmentierung:
  Nicht alle Bereiche eines Spektrums sind für die Untersuchung geeignet, da sie entweder keine relevanten Daten enthalten oder für eine Auswertung zu stark überlagert sind, wie z.B. der Diagonalbereich in NOESY- oder TOCSY-Spektren oder Wasserspuren. Diese Bereich können vor der eigentlichen Analyse des Spektrums ausgeschlossen werden.
- Identifikation einzelner Signalen und Auflösung von Überlappungen:
  Zur Unterscheidung der tatsächlichen Peaks vom Rauschen und Artefakten werden die einzelnen Signale an die theoretische Idealform, meist eine Gauss- oder Lorentzkurve, angepasst. Nicht überlappte Signale sollten ein symmetrisches Profil aufweisen. Die Symmetrieabweichungen dürfen daher für ein isoliertes Signal nicht über dem Rauschlevel liegen. Zur endgültige Klassifizierung bieten sich wiederum statistische Bayes-Methoden an [62].

Überlagerungen lassen sich nur schwer auflösen. Neuronale Netze erzielen hier sehr gute Ergebnisse, müssen aber für jeden Spektrentyp einzeln trainiert werden [60]. Ein anderer Ansatz untersucht die Krümmung des Datenraumes um jeden Punkt eines Signals [66]. Dieses Verfahren ist in der Lage Signale aufzulösen, bei denen sich die Oberflächenkrümmung innerhalb von minimal drei Datenpunkten ändert.

- Symmetrisierung:
  In NMR-Spektren, die voll symmetrisch zur Diagonale sind, kann über einen Signalabgleich die Auswahl der Peaks überprüft werden [67]. Diese Technik wird in der Praxis aufgrund starker Intensitätsunterschiede oder Symmetrieverschiebungen der korrespondierenden Kreuzsignale jedoch eher selten benutzt.

Andere Ansätze, wie das Programm von Croft et al. [68] oder v. Geerestein-Ujah et al. [69], verwenden zusätzlich *pattern recognition* Algorithmen oder graphentheoretische Methoden um Signalpositionen aufzulösen bzw. zu identifizieren. Es handelt sich hierbei um eine Erweiterung des *peak picking* Konzepts. *Peak picking* und *grouping* (s.u.) werden zu einem Schritt zusammengefasst. Bei solchen Ansätzen wird ein definierter Satz von Experimenten parallel analysiert und deren komplementäre Eigenschaften genutzt, um Überlagerungen oder fehlende Signale auszugleichen. Die Zusammenfassung der Daten (*grouping*) erfolgt anhand vordefinierter Muster. Diese Muster müssen spezifisch für jede Spektrenkombination erstellt werden. Das Verfahren ist anderen Referenzmethoden zwar überlegen, erfordert jedoch eine umfangreiche Parametrisierung vor jeder Rechnung. Durch die aufwendige Definition der Vergleichsmuster eignen sich solche Methoden nur für standardisierte Verfahren, bei denen stets die gleiche Spektrenkombination zur Verfügung steht. Die Integration neuer Experimente ist schwierig. Die Anwendbarkeit ist damit auf spezialisierte Bereiche der Protein-NMR-Spektroskopie begrenzt.

grouping:

Beim *grouping* werden die extrahierten Daten zu Spinsystemen geordnet. Diese Aufgabe kann entweder parallel zum *peak picking* verlaufen [68, 69] (s.o.) oder in einem separaten Schritt erledigt werden.

Meist werden die Signale, die zu einem Spinsystem gehören, jedoch über ein Paar von Referenzverschiebungen, die in fast allen gebräuchlichen NMR-Spektren enthalten sind

(z.B. $^{15}N/H^N$), identifiziert [55, 56, 70-72]. Zu Beginn des *grouping* wird ein Startspektrum benötigt, anhand dessen, stellvertretend für jede Aminosäure des Proteins, ein Pseudorest konstruiert wird, der die Referenzverschiebungen enthält. Die Pseudoreste werden anschließend mit weiteren Experimentdaten unter Berücksichtigung der beiden Referenzverschiebungen schrittweise vervollständigt.

typing:

Die klassische Methode zur Aminosäuretypisierung ist die Klassifizierung der Spinsysteme über Seitenkettendaten aus HCCH-COSY und HCCH-TOCSY Spektren [68, 70, 72]. Diese Art der Auswertung ist jedoch im Allgemeinen aufgrund fehlender Signale, geringer Intensitäten und Überlagerungen schwierig (s. Seitenkettenzuordnung). Es empfiehlt sich daher normalerweise die Zuordnung der Seitenkettensignale erst nach der Typisierung durchzuführen, wenn schon zielgerichtet nach den einzelnen Signalen der Spinsysteme gesucht werden kann.

Aminosäureselektive Experimente bzw. die Isotopenmarkierung ausgewählter Aminosäuren lassen zwar eine genaue Charakterisierung der Aminosäuren ohne Zuordnung der Seitenkettensignale zu, erfordern jedoch einen zusätzlichen experimentellen Aufwand. [39, 73-75]. Daher werden sie höchstens ergänzend zu anderen Methoden verwendet und eignen sich im Moment noch nicht für automatisierte Standardprozeduren.

In der Praxis wird die Aminosäureerkennung meist über chemische Verschiebungen und nicht über Spinsystemanalyse durchgeführt. Hierfür eignen sich aufgrund ihrer vergleichsweise großen Dispersion insbesondere die C^α– und C^β–Verschiebungen [56, 76-79], ergänzend kann noch die Stickstoff- bzw. Protoneninformation herangezogen werden [55].

Die Kombination dieser beiden Kohlenstoffverschiebungen lässt für die meisten Aminosäuren keine genaue Bestimmung, sondern nur eine Einschränkung der möglichen Zuordnungen auf circa fünf Kandidaten zu (s. 4.2.5). Die genaue Typisierung erfolgt erst nach dem *linking* (s.u.) durch den Sequenzvergleich von Clustern benachbarter Aminosäurereste (vgl. *mapping*). Direkt können mit Hilfe der C^α– und C^β–Verschiebungen die Aminosäuren A, G, S, T und V bestimmt werden (s. 4.2.5, Abbildung 23). Sie spielen im Sequenzvergleich daher eine zentrale Rolle. Die Typisierung von Einzelresten kann mit Bayes-statistischen Ansätze und der Verwendung von Zuordnungsdaten aus der BioMagRes-Datenbank (www.bmrb.wisc.edu) noch verbessert werden. Für jeden Aminosäuretypus wird mit Hilfe der

Datenbankzuordnungen eine Wahrscheinlichkeitsverteilung in Abhängigkeit der chemischen Verschiebungen angelegt [56], die zusätzlich durch Sekundärstrukturfaktoren modifiziert werden kann [80]. Auch neuronale Netze werden zur Typisierung eingesetzt [81, 82].

linking:

Unter *linking* versteht man die sequentielle Anordnung der Pseudoreste. Für das *linking* werden hauptsächlich zwei verschiedenen Methoden benutzt: deterministische *best first* Ansätze [55, 56, 72, 77, 79, 83-85] und kombinatorische Minimierungsverfahren [70, 86].

Bei deterministischen Ansätzen wird zu Beginn der Rechnung für jedes mögliche Aminosäurepaar der Zuordnungsliste eine Bewertung ermittelt. Anschließend wird jeweils das Aminosäurepaar mit der höchsten Bewertung aus der Menge der nicht zugeordneten Reste herausgesucht, bis alle Reste sequentiell miteinander verbunden sind.

Das Verfahren beruht auf der ständigen Verkleinerung der Lösungsmenge, indem sichere Lösungen zuerst gebildet werden. Für mehrdeutige Paare soll sich das Problem während des Verfahrens soweit vereinfachen, dass schließlich nur noch eine Lösung übrigbleibt. Solche Algorithmen sind im Allgemeinen eher fehleranfällig, da falsche Zuordnungen auf alle nachfolgenden Rechenschritte vererbt werden und daher weitere Fehler nach sich ziehen können. Fehlerhafte Pseudoreste können die Lösung so nicht nur lokal beeinflussen, sondern sie völlig unbrauchbar machen. Da als Zuordnungskriterium nicht die Qualität der Gesamtlösung herangezogen wird sondern nur lokale Bedingungen untersucht werden, ist die Auflösung von Überlagerungen bzw. die Zuordnung unvollständiger Reste nur bedingt möglich.

Kombinatorische Minimierungsverfahren versuchen dagegen das Minimum einer sogenannten Pseudoenergiefunktion zu ermitteln, die die Qualität der gesamten Zuordnung mittels empirisch bestimmter Kriterien beschreibt. Durch zufällige Modifikationen der Lösung wird versucht, sich dem Optimum anzunähern. Zu den kombinatorischen Minimierungsverfahren zählen beispielsweise Methoden wie das *simulated annealing* oder Monte-Carlo-Verfahren (s. 3.7). Sie zeichnen sich durch eine große Fehlertoleranz aus. Bei fehlenden oder falschen Pseudoresten wird die Gesamtlösung im Gegensatz zu den *best first* Verfahren nur lokal beeinflusst, d.h. falsche Entscheidungen haben während der Optimierung keinen Einfluss auf nachfolgende Schritte, da alle Modifikationen während des Rechenverlaufs reversibel bleiben.

mapping:

Der abschließende Schritt in der Sequenzzuordnung ist das Abbilden oder *mapping* der Pseudoreste auf die Sequenz. Ähnlich wie beim *linking* werden im großen und ganzen zwei Methoden verwendet: deterministische best-first Methoden [56, 72, 79, 85-87] und kombinatorische Minimierungsalgorithmen [55, 70, 71].

Häufig werden bei den deterministischen Methoden sogenannte *constraint propagation* Algorithmen, wie z.B. *backtracking* oder *forward checking*, zur Verbesserung ihrer Zuverlässigkeit und Effizienz benutzt. Das bedeutet, die Konsistenz der Daten wird während jeden Rechenschrittes durch Vergleich mit alten Lösungen bzw. Vorausberechnen möglicher Lösungen überprüft.

Die Anwendung von kombinatorischen Minimierungsmethoden für die Abbildung auf die Aminosäuresequenz ist schwierig, da zur Definition der Pseudoenergiefunktion zu wenig eindeutige Kriterien zur Verfügung stehen. Es ist in den meisten Fällen nicht möglich, aus den Verschiebungsdaten eine Zuordnung eindeutig als falsch zu qualifizieren. Es können lediglich Zuordnungswahrscheinlichkeiten angegeben werden.

Die Effizienz des *mapping* Schritts bestimmt sich hauptsächlich aus der Qualität der *typing* bzw. *linking* Routinen.

3.3 Seitenkettenzuordnung

Zur automatisierten Zuordnung der Seitenkettenresonanzen werden bisher grundsätzlich die Daten von HCCH-TOCSY- oder HCCH-COSY-Spektren analysiert. Dies geschieht im Allgemeinen über folgende drei Strategien:

- Gruppierung von Daten aus Verschiebungslisten in Spinsysteme anhand von Referenzverschiebungen (z.B. H^N/N) [70, 72]
- direkte Identifikation von Spinsystemen in den Spektren über Mustervergleich mit vordefinierten Templaten für jede Aminosäure [68, 72, 88]
- Datenanalyse mit Expertensystemen [81]

Die automatische Seitenkettenzuordnung wird durch Probleme wie hohe Signaldichte oder unvollständigen Magnetisierungstransfer im Spinsystem erschwert. Werden anstatt der unmittelbaren Spektrendaten Verschiebungslisten benutzt, müssen diese manuell erzeugt und sorgfältig korrigiert werden. Automatisch generierte Peaklisten besitzen eine zu geringe Genauigkeit für diese Aufgabe und bedingen eine große Zahl an Folgefehlern. Aufgrund dieser Problematik kommt der vollautomatischen Seitenkettenzuordnung momentan noch eine relativ geringe Bedeutung zu, lediglich der Ansatz von Croft et al. [68] und Huang et al. [81] stehen hier zur Verfügung. Einige ältere Programme beschränken sich stattdessen auf eine Reihe interaktiver Hilfsskripten [77, 78].

3.4 Sekundärstrukturbestimmung

Eine grobe Abschätzung der Sekundärstruktur lässt sich am einfachsten mittels der chemischen Verschiebung ermitteln. Beim CSI-Verfahren (***c**hemical **s**hift **i**ndex*) werden die sekundären chemischen Verschiebungen der Reste untersucht, d.h. die Differenz zwischen den tatsächlich gemessenen Werten und den *random coil*-Daten der einzelnen Aminosäuren. Befinden sich mindestens drei Reste mit den selben Verschiebungstendenzen hintereinander, kann auf die Anwesenheit des entsprechenden Sekundärstrukturelements geschlossen werden [46, 89, 90]. Das CSI-Verfahren eignet sich sehr gut für die Umsetzung am Rechner, da es als Datengrundlage lediglich die Zuordnung der Rückgratresonanzen des Proteins und eine Liste mit den durchschnittlichen chemischen Verschiebungen der Aminosäuren (*random coil*-Daten) benötigt.

Neben dem methodischen Verfahren des CSI werden auch heuristische Ansätze wie neuronale Netze [91] oder Wahrscheinlichkeitsanalysen bzw. Datenbankvergleiche zur Sekundärstrukturbestimmung verwendet [92-94]. Diese Ansätze sind im Allgemeinen jedoch weniger verlässlich als das CSI-Verfahren.

Die Analyse der $^{3}J(H^{N},H^{\alpha})$-Kopplungskonstanten, die bei einer manuellen Auswertung normalerweise durchgeführt wird, eignet sich nur bedingt zur Umsetzung in ein Programm, da

die automatische Extraktion der Daten aus den Spektren äußerst schwierig ist. Daher wird diese Methode in den momentan existierenden Automatisierungsansätzen nicht verwendet.

3.5 NOE-Zuordnung und Strukturrechnung

Für die Strukturrechnung werden überwiegend NOE-Kontakte und skalare Kopplungen eingesetzt. Diese lassen sich in Abstandsparameter bzw. Torsionswinkelangaben übersetzen, die anschließend während der Strukturrechnung verwendet werden können.

Der erfolgsreichste Ansatz für die automatisierte NOE-Zuordnung ist das Programm ARIA von Nilges et al. [95]. ARIA verwendet ein iteratives Protokoll, das stark an die Strukturrechnung gekoppelt ist. Nach Vorgabe einer Startstruktur, die beispielsweise auf Basis der bis dahin bekannten Topologiedaten des Proteins erstellt werden kann, wird diese im Verlauf der Rechnung immer weiter verfeinert. Dabei wechseln sich Schritte zur NOE-Suche und zur Strukturrechnung jeweils ab. Voraussetzung für eine Bestimmung der NOE-Signale ist eine möglichst vollständige Zuordnung der Verschiebungen aller Spinsysteme des Proteins. Das Konzept ist sehr flexibel und lässt sich auf die meisten Datensätze anwenden. In der Praxis empfiehlt sich jedoch ein manuelles Überprüfen der Zuordnungen zwischen den einzelnen Iterationsschritten, da Fehlzuordnungen zu einer großen Zahl an Folgefehlern und damit zu einer Nicht-Konvergenz der Struktur führen können.

Eine zweite, ebenfalls sehr erfolgreiche Strategie wird von dem Programm NOAH [96] verfolgt: Mit Hilfe der sogenannten „selbst korrigierenden Distanzgeometrie“ werden für jede Zuordnung die Parameter-Verletzungen in einem Strukturensemble berechnet. Das Programm NOAH ist daher ebenfalls über ein iteratives Verfahren eng mit der Strukturrechnung verbunden und wurde mit den Programmen DYANA [97] und DIAMOD [98] kombiniert.

Alternative Ansätze versuchen über das Abbilden von simulierten Erwartungswerten auf die experimentellen Peaklisten [71, 99, 100] oder die Analyse von Peakmustern mit graphentheoretischen Ansätzen [69] zum Ziel zu gelangen. Solche Ansätze sind jedoch weniger allgemein anwendbar als das ARIA-Konzept, so dass die Qualität des Ergebnisses von Datensatz zu Datensatz stark variieren kann.

Neben der automatisierten NOE-Zuordnung wird die Generierung von Parametern zur Strukturrechnung noch durch eine Vielzahl weiterer Programme unterstützt. So existieren beispielsweise Ansätze zur Abschätzung der Torsionswinkel aus der chemischen Verschiebung [101] oder Stereounterscheidung von H^{β}-Protonen [102]. Das Programm von Perlman et al. unterstützt den Prozess der Strukturrechnung durch die automatische Suche nach problematischen Parametern [103].

Die Strukturberechnung selbst erfolgt im großen und ganzen über zwei Methoden: distanzgeometrische und kraftfeldbasierte Rechenverfahren.

Die Distanzgeometrie ist ein mathematisches Verfahren zur Ermittlung von strukturellen Informationen aus Atomabständen [104-106]. Die vorgegebenen Abstandsparameter werden in einem definierten, erlaubten Bereich variiert und somit eine optimale Konformation für das gesamte Molekül ermittelt.

Kraftfeldrechnungen verwenden zur Beschreibung der Struktur bzw. Dynamik eines Moleküls ein empirisch bestimmtes Kraftfeld, das sämtliche für die Rechnung relevanten interatomaren Wechselwirkungen in mathematischer Form enthält. Zur Strukturbestimmung wird die so definierte Potentialhyperfläche nach ihrem globalen Energieminimum abgesucht [107]. Dies geschieht im Allgemeinen mit kombinatorischen Minimierungsprotokollen wie z.B. *simulated annealing* (vgl. Kombinatorische Minimierung).

Das Programm XPLOR von Brünger et al. [108] ist das in der Praxis am weitesten verbreitete Programm zur Strukturrechnung. Es verwendet sowohl Distanzgeometrie- als auch Kraftfeldrechnungen. Andere bekannte Programme sind DIANA von Güntert et al. [109], das mit distanzgeometrischen Methoden arbeitet, oder GROMOS96 von van Gunsteren et al. [110] für Kraftfeldrechnungen.

3.6 Kombinatorische Minimierung

Die sequentielle Anordnung der Spinsysteme eines Proteins ist ein sogenanntes np-vollständiges Problem (*nondeterministic polynomial*). Das bedeutet, dass keine einfache Lösung für das Problem gefunden werden kann und daher ein gerichteter Ansatz nicht

möglich ist. Zu dieser Klasse von Problemen gehört auch das bekannte *travelling salesman* Problem (TSP), bei dem die kürzeste Reiseroute für eine Rundreise über eine gegebene Anzahl von Orten bestimmt werden soll. Am Beispiel des TSP lässt sich die Schwierigkeit der Lösung von np-vollständigen Problemen anschaulich demonstrieren: Die Anzahl der möglichen Reiserouten verhält sich zur Anzahl n der Orte wie n!. Zur Berechnung der kürzesten Reiseroute durch nur 25 Städte, müssen so beispielsweise mehr als 10^{25} Lösungen berücksichtigt werden.

In der Praxis werden np-vollständige Probleme über sogenannte Heuristiken gelöst. Heuristiken sind Ansätze, mit deren Hilfe sich eine Lösung für ein Problem finden lässt, ohne dieses exakt beschreiben zu können. Die gefundene Lösung ist nicht unbedingt die tatsächliche (d.h. bestmögliche) Lösung des Problems, aber zumindest eine Lösung, die dem Optimum sehr nahe kommt.

Eine Klasse von Heuristiken sind die kombinatorischen Minimierungsverfahren. Diese Verfahren versuchen nicht, einem festgelegtem Schema zu folgen, sondern über zufällige Modifikationen irgendwann eine akzeptable Lösung zu erreichen. Dabei nutzen sie die Tatsache, dass sich die Qualität der momentanen Lösung im Allgemeinen einfach bestimmen lässt, ohne die exakte Formulierung des zugrundeliegenden Problems zu kennen. Für das Beispiel des TSP muss dazu nur die Gesamtlänge der momentanen Reiseroute aufaddiert werden. Im Fall der sequentiellen Zuordnung ist die Bewertung etwas schwieriger, da mehrere Kriterien die Qualität einer Zuordnung beeinflussen. Hier wird eine spezielle Pseudoenergie-Funktion zur Bewertung definiert, die das Zusammenpassen der Pseudo-Reste beschreibt (s. 4.2.3). Bekannte Beispiele für kombinatorische Minimierungsverfahren sind das *simulated annealing* und das *Monte-Carlo*-Verfahren.

Das Programm PASTA verwendet zur Lösung des Zuordnungsproblems den *threshold-accepting* Algorithmus (s. 3.7). *Threshold accepting* zählt ebenfalls zu den kombinatorischen Minimierungsmethoden und ist nah mit dem oben genannten *simulated annealing* verwandt. Ähnlich dem *simulated annealing* wird während der Rechnung ein Schwellenwert verwendet, der über die Akzeptanz einer Lösung entscheidet. Dieser Schwellenwert wird im Verlauf der Rechnung immer weiter abgesenkt, so dass immer weniger neue Lösungen gefunden werden und die Qualität der neuen Lösungen stetig ansteigt. Die Toleranzschwelle verhindert die ziellose Wanderung des Algorithmus und treibt ihn schließlich zu einem, möglicherweise lokalen, Maximum. *Threshold Accepting* unterscheidet sich von einfachem *simulated annealing* hauptsächlich durch das Verfahren zur Absenkung der Toleranzschwelle. Während

beim *simulated annealing* die Toleranzschwelle jeweils zu festen Zeitpunkten herabgesetzt wird, orientiert sich *threshold accepting* aktiv an der Qualität der Lösung. Der Algorithmus wird so schneller und effizienter, da selbständig erkannt wird, ob eine weitere Lösungssuche zu den momentanen Suchkriterien sinnvoll ist und gegebenenfalls sofort darauf reagiert werden kann. Das Ändern von Bewertungskriterien spielt auch bei einem anderen bekannten Verfahren, dem sogenannten *tabu search* [111], eine zentrale Rolle. Beim *tabu search* wird angenommen, dass nicht alle Bewertungskriterien zu jedem Zeitpunkt der Optimierung gleich wichtig sind bzw. manche Kriterien zum falschen Zeitpunkt auch hinderlich sein können. Die Bewertungsfunktion wird also ständig dem Problem angepasst. In PASTA wurde eine ähnliche Technik zur Optimierung unter Berücksichtigung der Aminosäuresequenz angewendet. Der Aminosäureabgleich wird erst zugeschaltet, nachdem die Pseudoenergie einen Wert < 0 angenommen hat, da zuvor davon ausgegangen wird, dass noch nicht genügend Fragmente mit einer für den Abgleich ausreichenden Länge gebildet wurden (s. 4.1.1).

3.7 Der *threshold accepting*-Algorithmus

Der Algorithmus wurde 1990 von Dueck und Scheuer erstmals veröffentlicht [112] und besteht, auf PASTA bezogen, vereinfacht aus folgenden Schritten:

1. Wähle eine zufällige Lösung x aus dem Lösungsraum.
2. Erzeuge eine neue Lösung y durch Austausch einzelner Reste oder ganzer Fragmente der Pseudo-Rest-Liste.
3. Vergleiche $E_{pseudo}(x)$ und $E_{pseudo}(y)$. Falls $E_{pseudo}(y) \leq E_{pseudo}(x) + T$ (Toleranzschwelle), verwende $E_{pseudo}(y)$ anstatt von $E_{pseudo}(x)$ als neue Lösung.
4. Falls nach einer bestimmten Anzahl von Durchläufen immer noch keine bessere Lösung gefunden worden ist, wird die Toleranzschwelle T um 1 erniedrigt.
5. Die Optimierung ist beendet, wenn $T < 0$ erreicht ist.

In Abbildung 2 ist das Flussdiagramms des *threshold accepting*-Algorithmus dargestellt. Für das Zuordnungsproblem wird bei Schritt zwei, dem Erzeugen einer neuen Lösung, zwischen zwei Möglichkeiten unterschieden: Die neue Lösung entsteht entweder durch Austausch nur eines Restes oder durch Vertauschen eines ganzen Fragments aus mehreren Resten. Der Austausch ganzer Fragmente ermöglicht es, innerhalb eines Zyklus des Algorithmus an eine von der derzeitigen Lösung weit entfernte Position des Lösungsraumes zu gelangen. Damit ist ein schnelles Abtasten des Lösungsraumes gewährleistet und die Gefahr, sich in lokalen Minima zu verfangen, eingeschränkt. Um die Konvergenzfähigkeit des Algorithmus zu bewahren, wird nur bei etwa jedem fünften Schritt ein Fragment anstatt eines einzelnen Restes ausgetauscht. Die Auswahl erfolgt zufällig.

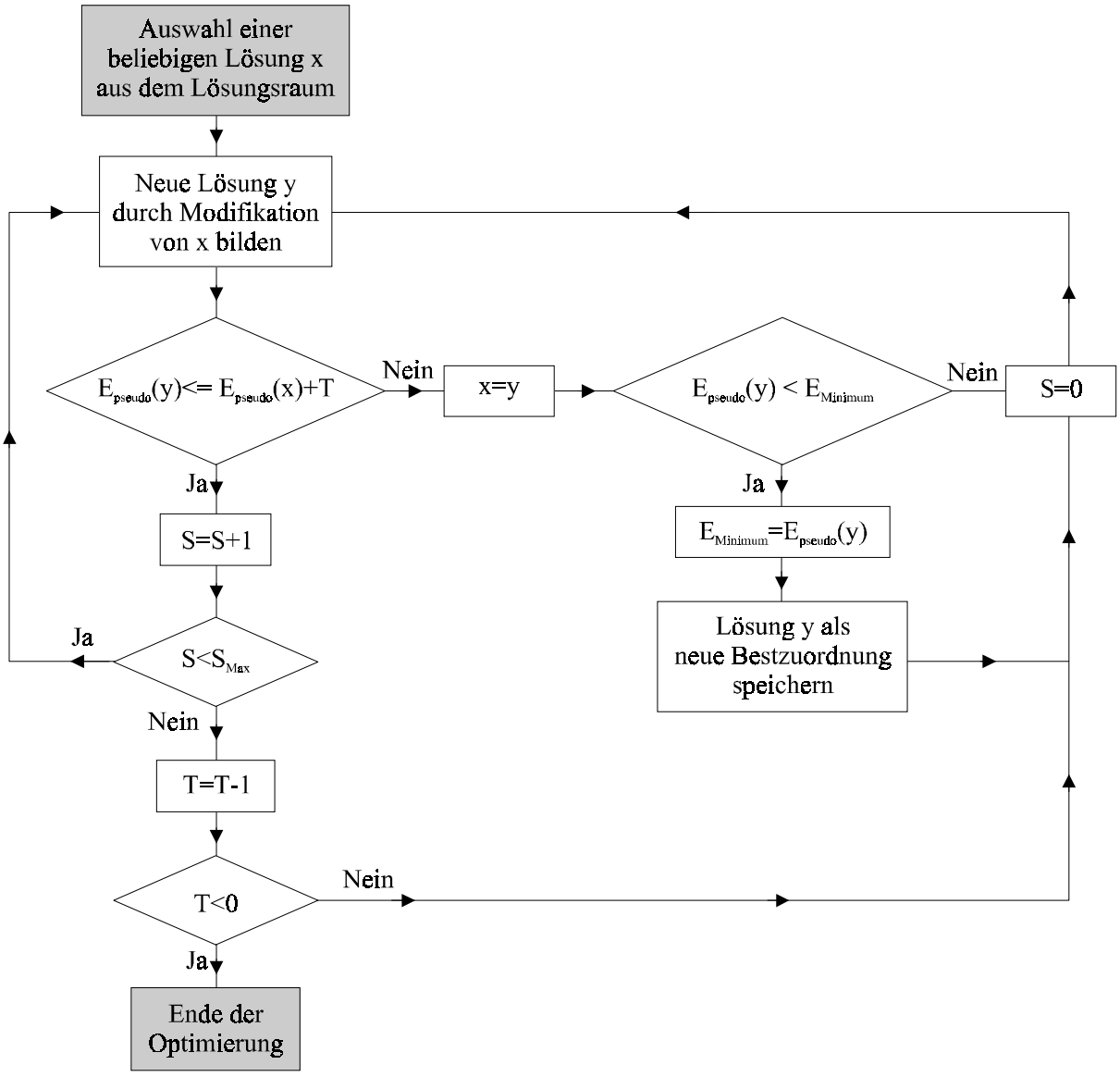

Abbildung 2: *Flussdiagramm des threshold accepting-Algorithmus (T = Toleranzschwelle; S = Schrittanzahl).*

4 PASTA (*P*rotein *A*ssignment by *T*hreshold *A*ccepting)

4.1 PASTA V3.0

4.1.1 Vorarbeiten

Das Programm PASTA (**P**rotein **As**signment by **T**hreshold **A**ccepting) wurde entwickelt, um den zeitaufwendigen Schritt der Proteinrückgratzuordnung zu automatisieren. Die Aufgaben des Programms liegen dabei sowohl in der Zuordnung der Daten als auch in deren Verwaltung. Der ursprüngliche Ansatz geht auf die Arbeit von Michael Leutner zurück [19] [18].

Als Datengrundlage für die Sequenzierung wurde eine beliebige Kombination der chemischen Verschiebungen von C^{α}, C^{β}, C' und H^{α} gewählt. Die Daten werden in sogenannten Pseudo-Rest-Listen gespeichert. Ein Pseudo-Rest enthält alle bekannten Daten eines Aminosäure-Spinsystems. Die Pseudo-Rest-Liste liegt in Form einer ASCII-Text-Datei vor [Abbildung 3].

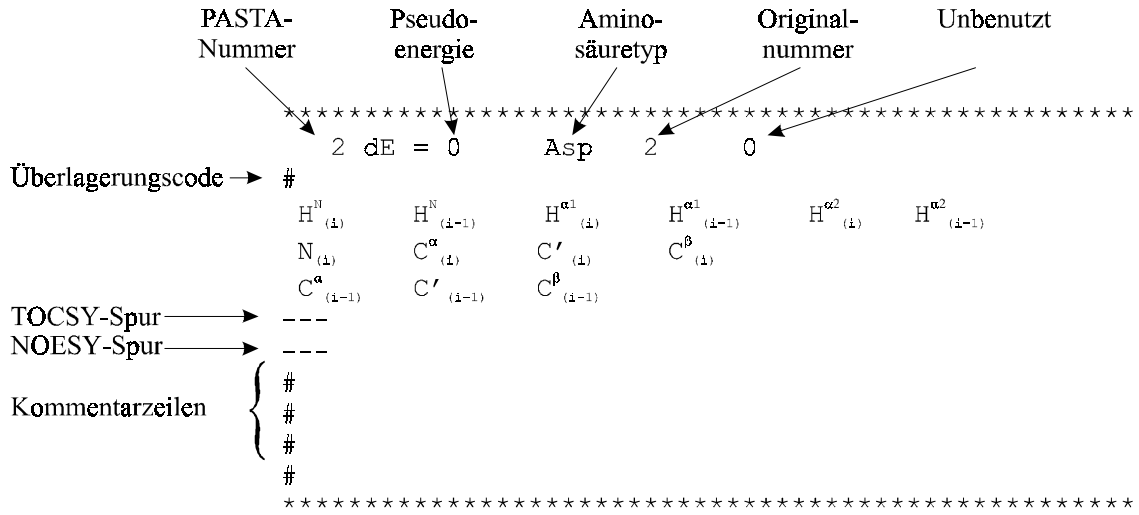

Abbildung 3: *Format der Pseudo-Restliste in PASTA V3.0.*

Das Programm nutzt die Daten verschiedener gängiger Experimente [Abbildung 4]. Für jedes dieser Experimente verfügt es über einen speziellen Einlesefilter. Die Filtermodule lesen die Daten in Form von ASCII-Peaklisten ein, wie sie mit vielen Standard-Software-Paketen erzeugt werden können (z.B. AURELIA (BRUKER)[113], TRIAD (TRIPOS). Zu Beginn der Zuordnung muss ein sogenanntes Startspektrum eingelesen werden, anhand dessen Daten eine Pseudo-Restliste konstruiert wird. Als Startspektren können entweder das ^{15}N-HSQC oder das HNCO verwendet werden. Nach dem Erzeugen der Pseudo-Restliste können weitere Experimente für die Zuordnung eingelesen werden. Die Einlesefilter versuchen dabei die Verschiebungsdaten automatisch zu klassifizieren und in die entsprechenden Pseudo-Reste der Zuordnungsliste einzuordnen. Das Ergebnis wird in einer Reportdatei gespeichert und kann schließlich manuell korrigiert und vervollständigt werden.

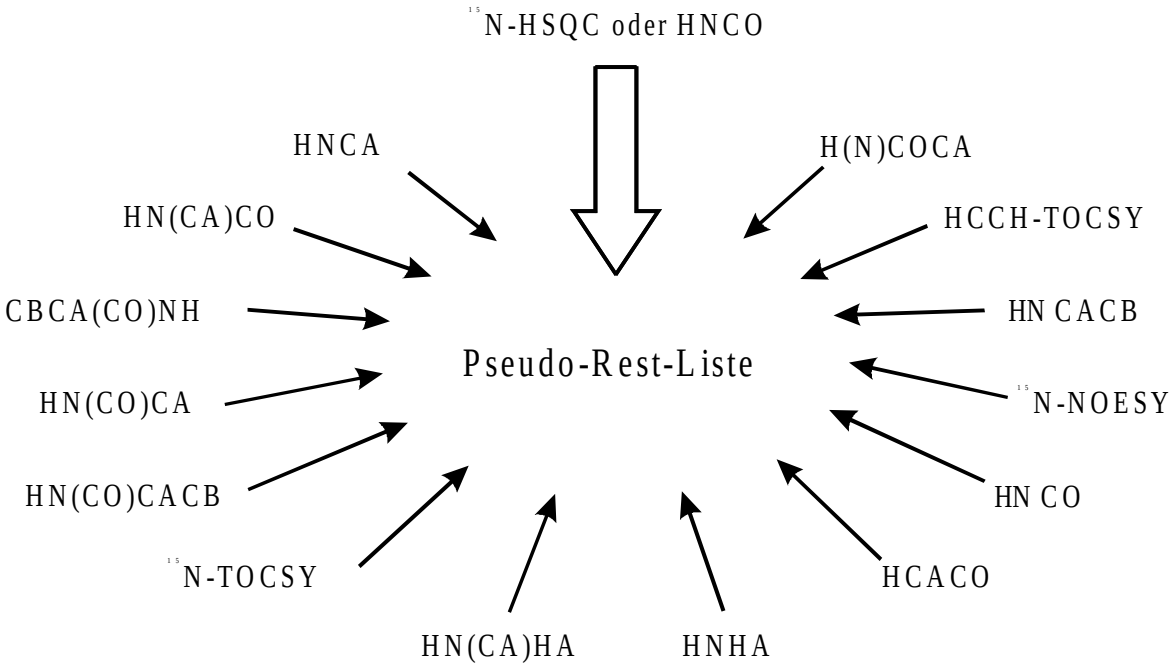

Abbildung 4: *Experimente, die bei PASTA V3.0 in die Pseudo-Restliste eingelesen werden können. ^{15}N-HSQC und HNCO können dabei die Rolle sogenannter Startspektren übernehmen.*

Kernstück des Programms ist eine Optimierungsroutine, mit deren Hilfe die Pseudo-Reste der Zuordnungsliste in die richtige sequentielle Anordnung gebracht werden. Dies geschieht mittels *threshold accepting* (s. 3.7), einer kombinatorischen Minimierungsmethode ähnlich

dem *simulated annealing*. Die Pseudoenergiefunktion für die Minimierung definiert sich wie folgt:

$$E_{GES} = \sum E_{RES} = \sum (E_{MATCH} \{+E_{SEQ}\})$$

$$E_{MATCH} = E_{C^\alpha} + E_{C^\beta} + E_{C'} + E_{H^\alpha}$$

E_{GES} ist die Energie der gesamten Pseudo-Restliste. Sie ist gleich der Summe über alle E_{RES} , d.h. der Energien aller Einzelreste. E_{RES} definiert sich wiederum aus E_{MATCH}, einem Maß für die Übereinstimmung der [i]-Verschiebungswerte mit den [i-1]-Werten des sequentiellen Nachbarn der Zuordnungsliste, und ggf. E_{SEQ}, das aus dem Vergleich der Zuordnungsliste mit der Proteinsequenz bestimmt wird. E_{SEQ} ist optional und wird nicht unbedingt zur Optimierung benötigt. Für die Berechnung von E_{SEQ} wird ein Fenster von vier aufeinanderfolgenden Aminosäuren mit der Aminosäuresequenz des Proteins verglichen. Dieses Verfahren macht nur Sinn, wenn die Zuordnungsliste bereits längere, zusammenhängende Fragmente enthält. Die Optimierung schaltet E_{SEQ} daher erst bei einer Pseudoenergie $E_{GES} < 0$ zu. Die Verwendung des E_{SEQ}-Terms setzt eine Aminosäureerkennung (s.u.) voraus.

Die Energiewerte $E_{C\alpha}$, $E_{C\beta}$, $E_{C'}$ und $E_{H\alpha}$ wurden empirisch bestimmt [Tabelle 2]. Sie richten sich nach der Überlagerungstendenz und dem Informationsgehalt der Kerne.

$E_{C\alpha}$	$E_{C\beta}$	$E_{C'}$	$E_{H\alpha}$	E_{Fehler}
-12	-12	-15	-10	+130

Tabelle 2: *Definition der Energiewerte für die Pseudoenergiefunktion E_{MATCH}.*

Die Werte für E_{SEQ} sind wesentlich geringer angesetzt, da dieser Energieterm während der Optimierung nur zur Stabilisierung bereits bestehender, korrekter Fragmente dient. Für die Übereinstimmung des Aminosäuretyps eines Restes mit der entsprechenden Sequenzposition ist ein Energiewert von –5 definiert, bei Auftreten eines Fehlers beträgt die Energie +15.

Im Rahmen der Optimierung bietet das Programm die Option eine einfache Aminosäureerkennung durchzuführen. Diese basiert auf den chemischen Verschiebungen der C^α- und C^β-Kerne eines Aminosäurerestes. Ausgehend von den *random coil*-Verschiebungen lassen sich für die Aminosäuren A, G, S, T und V Verschiebungsmasken definieren, die eine

eindeutige Identifikation ermöglichen. Außerdem können noch die Aminosäuren I, F, Y als Gruppe erkannt werden.
Die Manipulation der Daten erfolgt über eine Konsolenschnittstelle. Der Zugriff auf die einzelnen Programmfunktionen ist über die Eingabe von Buchstabenkürzeln möglich.
Das Programm ist in der Programmiersprache C für das Betriebssystem IRIX (SGI) implementiert. Der durchschnittliche Rechenzeitbedarf eines Optimierungslaufes für ein Protein mit 150 Aminosäuren beträgt ca. 3 h auf einer SGI Indy R4600/133 MHz Workstation.

4.1.2 Eigene Arbeiten

4.1.2.1 Entwicklung von *Fast TA*

Die Optimierung einer Pseudo-Restliste stellt ein hochspezialisiertes Optimierungsproblem dar. Deswegen können zur Beschleunigung des Optimierungsalgorithmus einige Modifikationen eingeführt werden, die bei allgemeineren Aufgabenstellungen zu einem Verlust von potentiellen Lösungen führen können. Im Rahmen dieser Arbeit wurden aufbauend auf *threshold accepting* (s. 3.7) zwei Verfahren entwickelt, mit denen sich die Rechengeschwindigkeit im Vergleich zum Originalalgorithmus um etwa den Faktor 10 steigern lässt.
Das effektivste Mittel zur Beschleunigung ist eine Einschränkung des zu untersuchenden Lösungsraumes, da der Bedarf an Rechenzeit exponentiell mit der Anzahl der Elemente im Lösungsraum ansteigt.
Dazu wird vor dem Beginn der eigentlichen Optimierung für jeden Aminosäurerest eine Liste mit allen erlaubten Nachfolgern erzeugt (Liste 1), d.h., für jede mögliche Kombination zweier Aminosäurereste wird vorab die Pseudoenergie berechnet. Entspricht der Energiewert einem Zuordnungsfehler, wird diese Kombination der beiden Aminosäurereste für die spätere Rechnung ignoriert und nicht aktiv in die Optimierung mit einbezogen.
Eine solche Vereinfachung ist zulässig, da in der Pseudoenergiefunktion nicht zwischen verschiedenen Arten von Fehlern bei der Bewertung eines Aminosäurepaares unterschieden

wird. Es spielt keine Rolle, wie viele Verschiebungswerte nicht zueinander passen, da im Überlagerungsfall schon eine einzelne Abweichung für einen Zuordnungsfehler ausreicht. Fehlerhaften Reste wird deswegen grundsätzlich ein Energiewert von +130 zugeordnet.

Daraus folgt, dass die Bewertungen der sequentiellen Verknüpfung zweier Reste rein lokal ist und auch in jeder Teilmenge der ursprünglichen Startmenge eine eindeutige Aussage über das Vorliegen eines Zuordnungsfehlers gemacht werden kann. Das bedeutet, jedes einzelne Restepaar, das einen Zuordnungsfehler liefert, führt auch in der Gesamtzuordnung zu einem Fehler und muss daher im Optimierungsprozess nicht mehr berücksichtigt werden.

Auf diese Weise lässt sich die Zahl der zu untersuchenden Gesamtlösungen üblicherweise um mehrere Zehnerpotenzen verringern.

Ein zweites Mittel zur Beschleunigung des Algorithmus besteht in einer zielgerichteten Optimierung an Problemstellen. Das bedeutet, jeder Aminosäurerest wird proportional zur Anzahl seiner potentiellen Nachfolger bei der Erzeugung neuer Lösungen berücksichtigt. Dazu wird eine weitere Liste (Liste 2) angelegt, die für jeden Aminosäurerest so viele Einträge enthält, wie er potentielle Nachfolger (Liste 1, s.o.) besitzt.

Während des Optimierungslaufes werden neue Lösungen auf Basis dieser beiden Listen erzeugt [Abbildung 5]. Die Erzeugung einer neuen Lösung verläuft in drei Stufen. Zuerst wird ein zufälliges Element A aus Liste 2 gewählt. Dieses Element entspricht der Stelle der Pseudo-Restliste, an der ein neuer Nachfolger B bestimmt werden soll. B wird wiederum zufällig unter den potentiellen Nachfolgern von A, die in Liste 1 gespeichert sind, gewählt. Um auch weitreichendere Veränderungen während eines einzelnen Modifikationsschrittes zu ermöglichen, wird in der dritten Stufe festgelegt, ob nur ein Aminosäurerest oder ein ganzes Fragment aus mehreren Resten ausgetauscht werden soll (s. Abbildung 5).

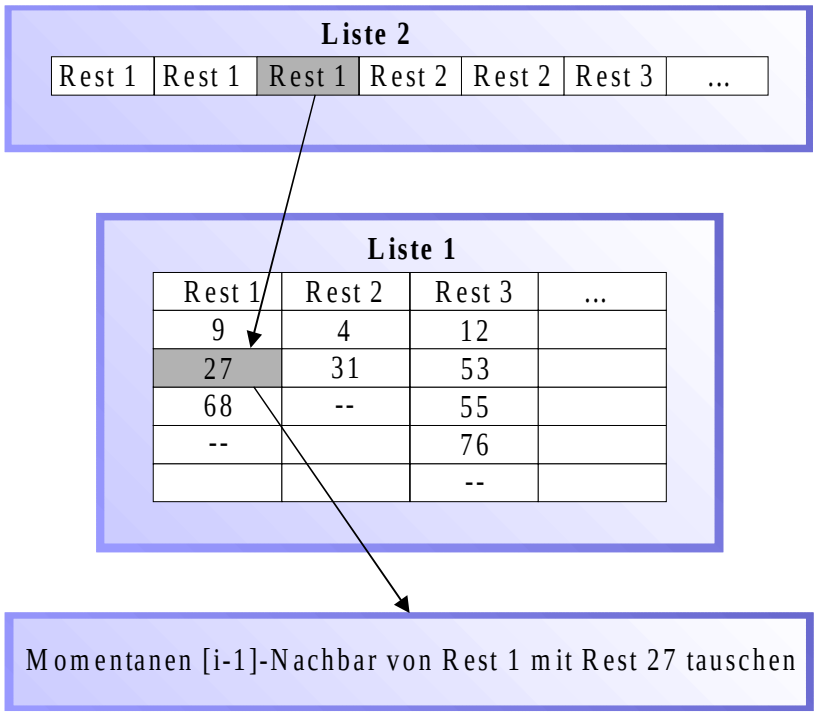

Abbildung 5: Fast TA *verwendet zwei Hilfslisten zur Erzeugung einer neuen Lösung. Liste 1 enthält für jeden Aminosäurerest alle zulässigen Nachfolger. In Liste 2 wird jeder Rest so oft eingetragen, wie er potentielle Nachfolger in Liste 1 besitzt.*

Im Vergleich zum normalen *threshold accepting* wird die neue Lösung also nicht mehr rein zufällig, sondern „gerichtet“ zufällig erzeugt. Dies führt zu einer wesentlich schnelleren Konvergenz des Energiewertes [Abbildung 6], was ebenfalls eine höhere Zuverlässigkeit und Qualität der Lösungen bedingt. Die Robustheit des Verfahrens gegenüber fehlerhaften Resten bzw. fehlenden Verschiebungsinformation bleibt unverändert.

Alle Rechnungen der folgenden Beispiele wurde an einem Datensatz des Proteins NusB durchgeführt. Der Datensatz enthält 129 Pseudo-Reste. Optimiert wurde über die Parameter C^{α}, C^{β}, C’ und H^{α}. Der Energieterm E_{SEQ} wurde nicht verwendet. Abzüglich der Werte für fehlende Verschiebungen und Prolinpositionen entspricht das globale Minimum der Energiefunktion für diesen Datensatz einem Wert von –2239.

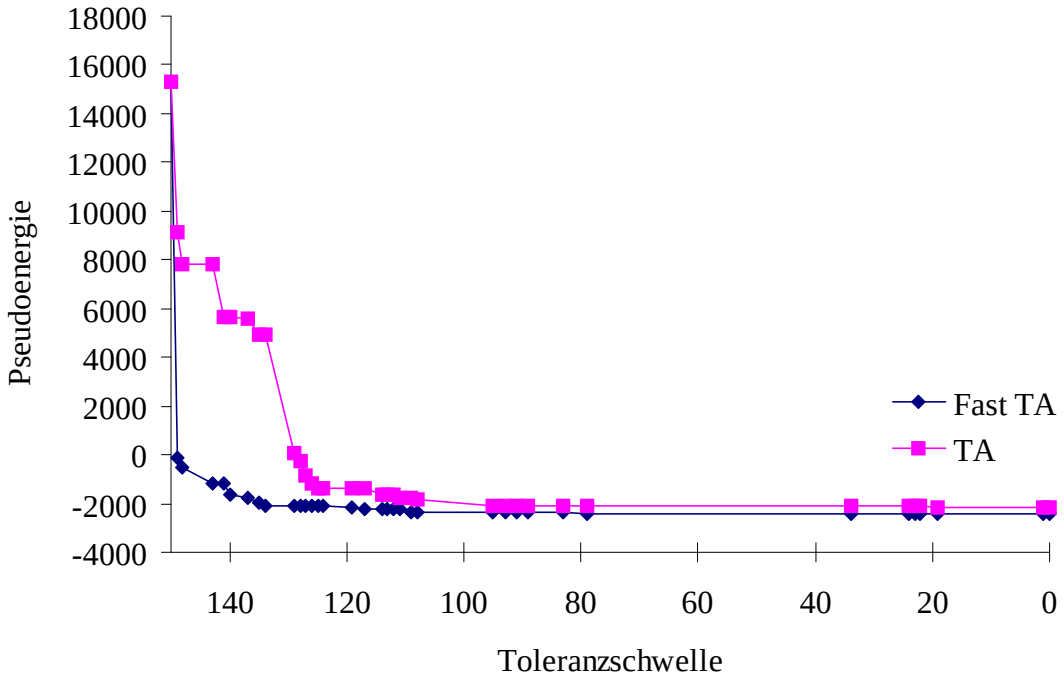

Abbildung 6: *Vergleich der Konvergenz der Pseudoenergie unter Verwendung von* fast TA *und* threshold accepting *während des Rechenverlaufs (Datensatz: NusB; 129 Reste; T=150; S=2000).*

Abbildung 6 zeigt das Konvergenzverhalten der beiden Algorithmen. Die Pseudoenergie erreicht bei *fast TA* schon vor dem ersten Absenken der Toleranzschwelle (T = 150) einen negativen Energiewert. Bei der Rechnung mit *threshold accepting* beginnt die Pseudoenergie erst ab $T < 130$ unter 0 zu sinken. Der Wert 130 entspricht der Bewertung eines Zuordnungsfehlers in der Pseudoenergiefunktion (s. 4.1.1). $T < 130$ entspricht damit dem Zeitpunkt der Rechnung, ab dem bei der Generierung neuer Lösungen keine Zuordnungsfehler mehr eingeführt werden dürfen.

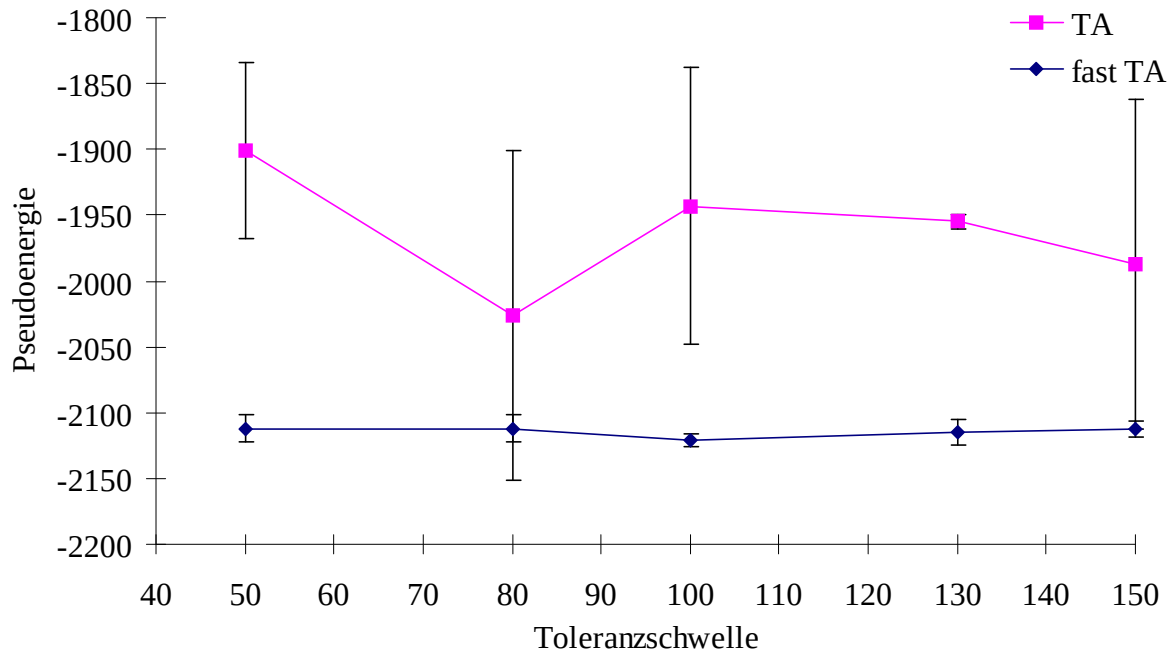

Abbildung 7: *Pseudoenergie der Gesamtlösung in Abhängigkeit der Toleranzschwelle für* fast TA *und* threshold accepting *(Datensatz: NusB; 129 Reste; T=50, 80, 100, 130, 150; S=4000; 5 Rechnungen pro Datenpunkt).*

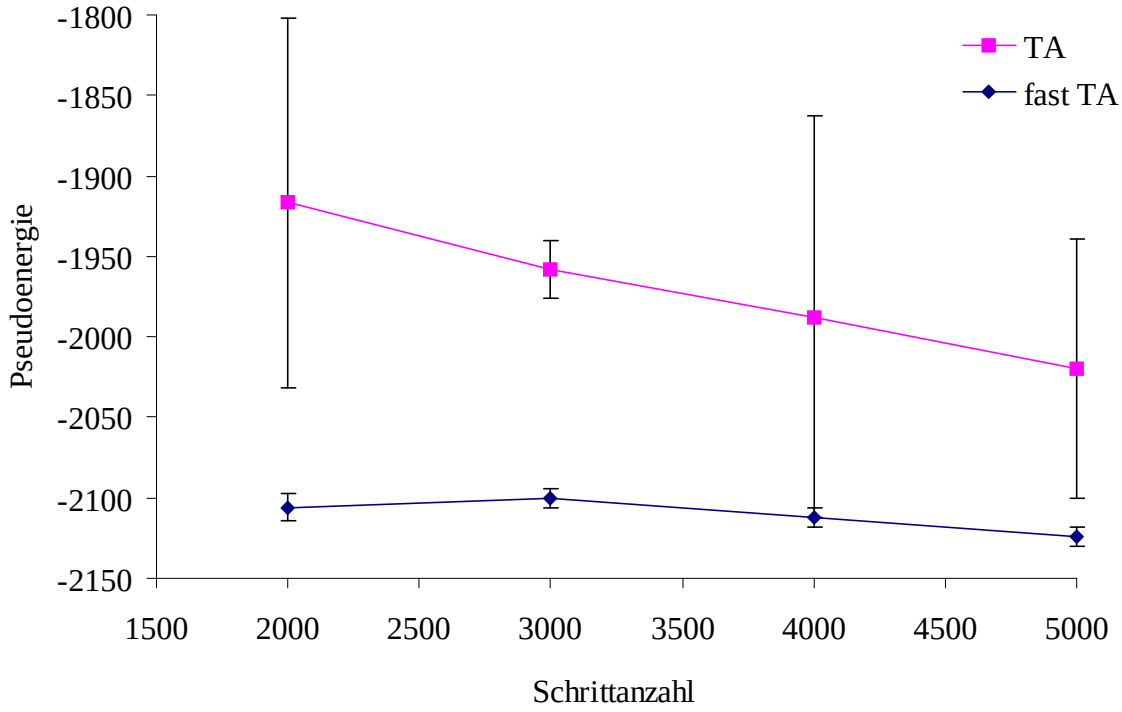

Abbildung 8: *Pseudoenergie der Gesamtlösung in Abhängigkeit der Schrittanzahl für* fast TA *und* threshold accepting *(Datensatz: NusB; 129 Reste; T=150; S=2000, 3000, 4000, 5000; 5 Rechnungen pro Datenpunkt).*

In Abbildung 7 und Abbildung 8 sind beide Algorithmen in Hinblick auf ihre Zuverlässigkeit gegenübergestellt. *Fast TA* erreichte in 42 von 45 Rechnungen eine niedrigere Pseudoenergie als *threshold accepting*. Für *fast TA* ist im getesteten Bereich kein deutlicher Einfluss der Parameter T und S auf die Pseudoenergie festzustellen. Im Gegensatz dazu steigt die Energie für *threshold accepting* bei Erhöhung der T und S Werte noch deutlich an. Die Standardabweichung für fünf Rechnungen mit denselben Parametern liegt für *threshold accepting* ebenfalls wesentlich höher: Für *fast TA* beträgt sie 8.7, für *threshold accepting* liegt sie bei 98.3. Daraus lässt sich schlussfolgern, dass *threshold accepting* zum Erzielen der gleichen Lösungsqualität eine wesentlich höhere Anzahl an Rechenschritten als *fast TA* benötigt. Für *fast TA* genügen im Testbeispiel die Werte T = 100 und S = 2000 zum Erreichen des Optimalwertes, während *threshold accepting* auch bei Werten von T = 150 bzw. S = 5000 noch deutlich unter dem Bereich des Maximums liegt.

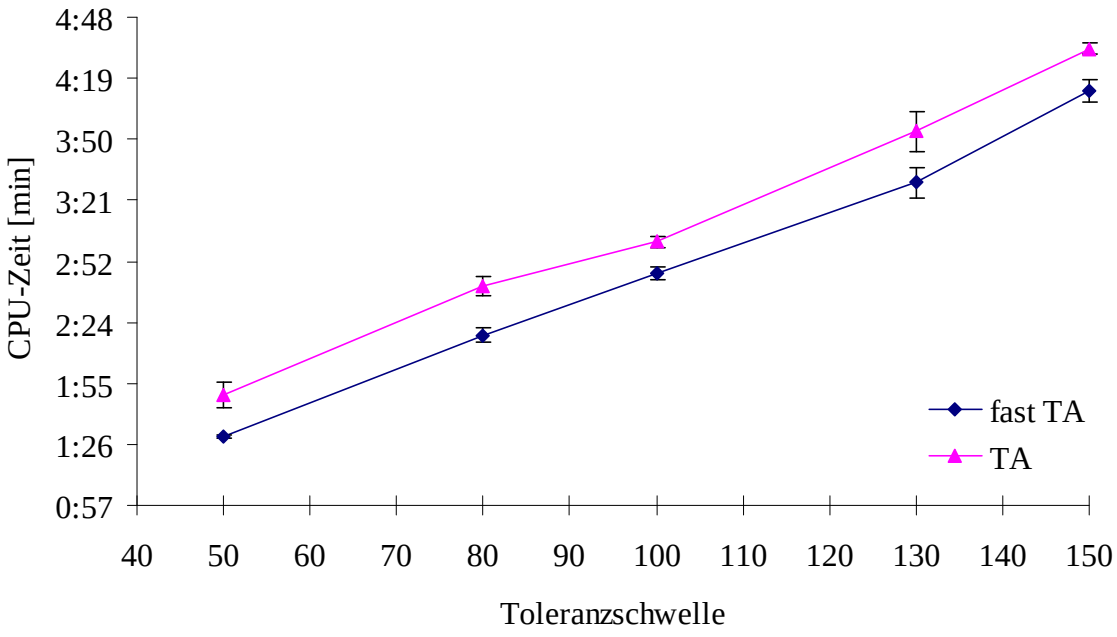

Abbildung 9: *Rechenzeit in Abhängigkeit der Toleranzschwelle für* fast TA *und* threshold accepting *(Datensatz: NusB; 129 Reste; T=50, 80, 100, 130, 150; S=2000; 5 Rechnungen pro Datenpunkt; Alle Rechnungen wurden auf einer SGI Indy R4600/133MHz durchgeführt).*

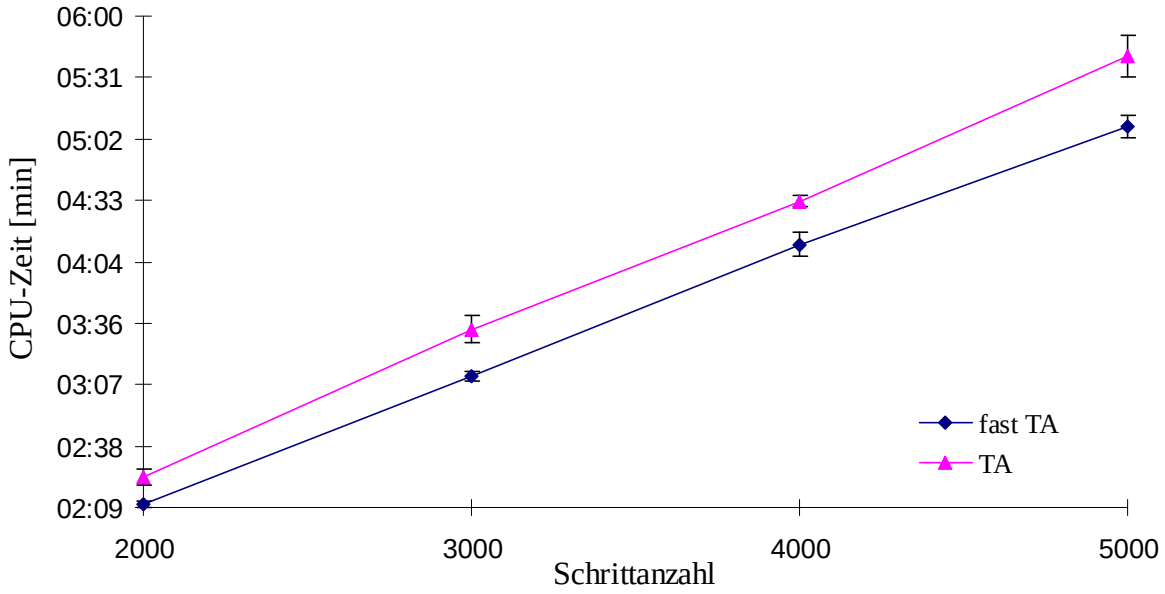

Abbildung 10: *Rechenzeit in Abhängigkeit der Schrittanzahl für fast TA und threshold accepting (Datensatz: NusB; 129 Reste; T=150; S=2000, 3000, 4000, 5000; 5 Rechnungen pro Datenpunkt; Alle Rechnungen wurden auf einer SGI Indy R4600/133MHz durchgeführt).*

In Abbildung 9 und Abbildung 10 ist der Rechenzeitbedarf der beiden Algorithmen in Abhängigkeit der Parameter T und S dargestellt. Für beide Algorithmen besitzen die resultierenden Kurven bei Variation von T etwa eine Steigung von $1.1*10^{-3}$. In Abhängigkeit von S ist die Steigung der Kurve für *threshold accepting* ($5*10^{-5}$) geringfügig höher als für die Kurve von *fast TA* ($4*10^{-5}$). Die Zeitwerte liegen für *fast TA* grundsätzlich etwas niedriger. Die Unterschiede erklären sich aus der schnelleren Konvergenz von *fast TA*. Mit steigender Schrittanzahl S werden die Auswirkungen der Konvergenz langsam sichtbar: Während der Rechenzeitunterschied für 2000 Schritte nur 13 Sekunden beträgt, unterscheidet sich die Rechendauer bei 5000 Schritten schon um 33 Sekunden. Eine Variation von T zeigt dagegen keine unterschiedlichen Auswirkungen im Vergleich der beiden Algorithmen. Der konstante Zeitunterschied zwischen *fast TA* und *threshold accepting* erklärt sich durch die unterschiedlichen Methoden zur Variation der Lösung. *Fast TA* benötigt aufgrund der problemorientierten Modifikation weniger Modifikationsschritte als *threshold accepting*. Dieser Unterschied ist nicht von den Startwerten für S und T abhängig.

Die rechenzeitbestimmenden Kriterien des Algorithmus sind die Werte für die Toleranzschwelle T und die Schrittanzahl S. Der Rechenzeitbedarf steigt für beide Parameter linear im Verhältnis 1:1 an (Abbildung 9 und Abbildung 10). Aufgrund der schnelleren

Konvergenz von *fast TA* ist es möglich, wesentlich niedrigere Startwerte für T und S als bei *threshold accepting* zu wählen ohne dabei an Lösungsqualität zu verlieren (Abbildung 7 und Abbildung 8). Im Allgemeinen werden bei einer Halbierung von T und S mit fast TA immer noch energetisch günstigere Lösungen erzielt. In der Praxis sind damit Zeiteinsparungen von mehr als 75% möglich.

4.1.2.2 Abbilden der Zuordnungsdaten auf die Aminosäuresequenz (*mapping*)

Der letzte Schritt zur automatischen Rückgratzuordnung von Proteinen ist das Abbilden der Pseudo-Reste auf die Aminosäuresequenz. Das aus den Vorarbeiten übernommene Konzept von PASTA (Kapitel 4.1.1) wurde um eine entsprechende Routine ergänzt. Das erstellte Programm beruht auf der Annahme, dass innerhalb größerer Zuordnungsfragmente (≥ 5 Reste) zumeist auch mehrere Ankeraminosäuren (A, G, S, T, V) enthalten sind, die von der Aminosäureerkennung identifiziert werden können. Aufgrund der charakteristischen Muster der Ankeraminosäuren innerhalb der Zuordnungsfragmente ist für viele größere Fragmente das Abbilden auf die Sequenz möglich. In Abbildung 11 ist der Prozess schematisch dargestellt.

Die Routine verfügt über zwei Eingabeparameter. Der erste Parameter legt die Mindestanzahl von Ankeraminosäuren pro Fragment fest, die für eine Zuordnung erforderlich sind. Besitzt ein Fragment weniger als die geforderte Mindestanzahl von Ankeraminosäuren, wird es während der Rechnung nicht berücksichtigt. Mit dem zweiten Eingabeparameter lässt sich die Anzahl der zulässigen Vergleichsfehler pro Fragment festlegen. So können fehlerhafte oder unvollständige Ergebnisse der Aminosäureerkennung bei Fragmenten mit ansonsten ausreichender Anzahl von Ankerpunkten ausgeglichen werden.

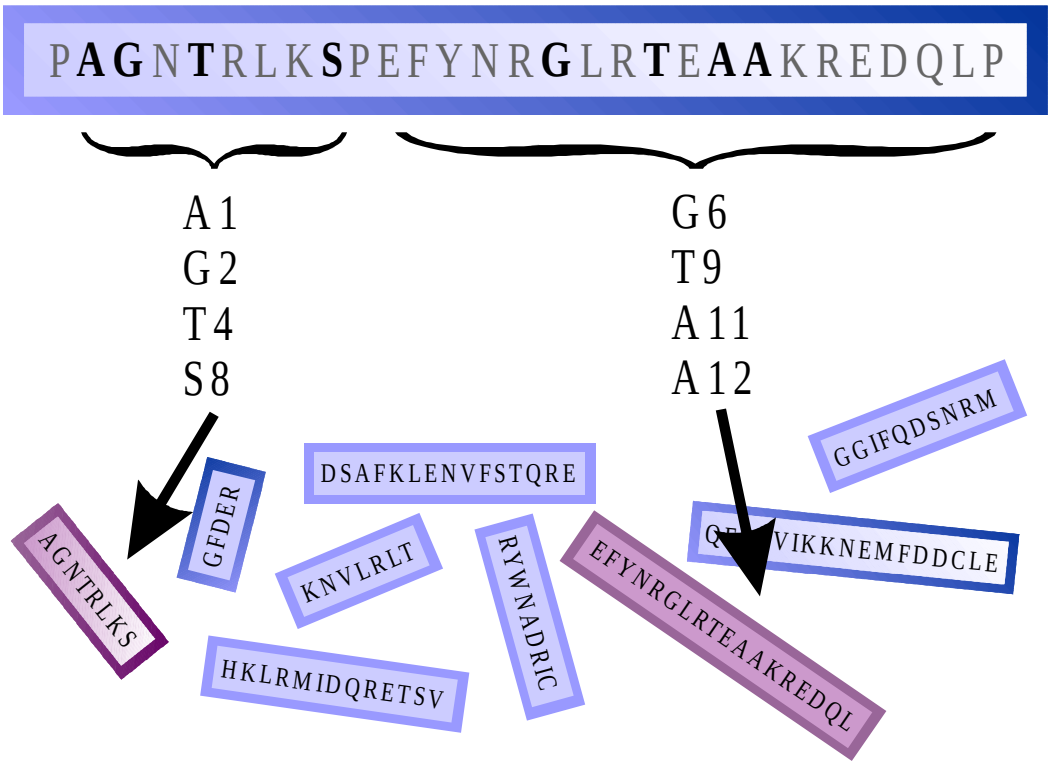

Abbildung 11: *Schematische Darstellung des* mapping-*Konzepts. Zur Abbildung der einzelnen Zuordnungsfragmente werden charakteristische Muster von Ankeraminosäuren genutzt.*

Abbildung 12 zeigt den entwickelten Algorithmus. Die Routine arbeitet die Fragmente der Zuordnungsliste sequentiell ab. Als Fragmente werden dabei alle zusammenhängenden Stücke der Pseudo-Restliste betrachtet, in denen alle enthaltenen Reste durch mindestens zwei sequentielle Informationen miteinander verknüpft sind. Für jedes Fragment wird zuerst die Sequenzposition herausgesucht, die auf das Muster der Ankeraminosäuren passt. Existieren mehrere Lösungen, so kann das Fragment nicht zugeordnet werden. Wurde eine passende Stelle gefunden, wird getestet, ob für das Fragment an dieser Position genügend freie, d.h. bisher nicht zugeordnete, Plätze in der Sequenz vorhanden sind. Falls ausreichend viele freie Positionen zur Verfügung stehen, werden diese als besetzt markiert und die Zuordnung in die Pseudo-Restliste eingetragen. Auf diese Weise werden auch fehlerhafte Reste innerhalb der Fragmente auf die Sequenz übertragen. Durch die fälschliche Besetzung von Sequenzpositionen kann die Zuordnung nachfolgender Fragmente gestört werden.

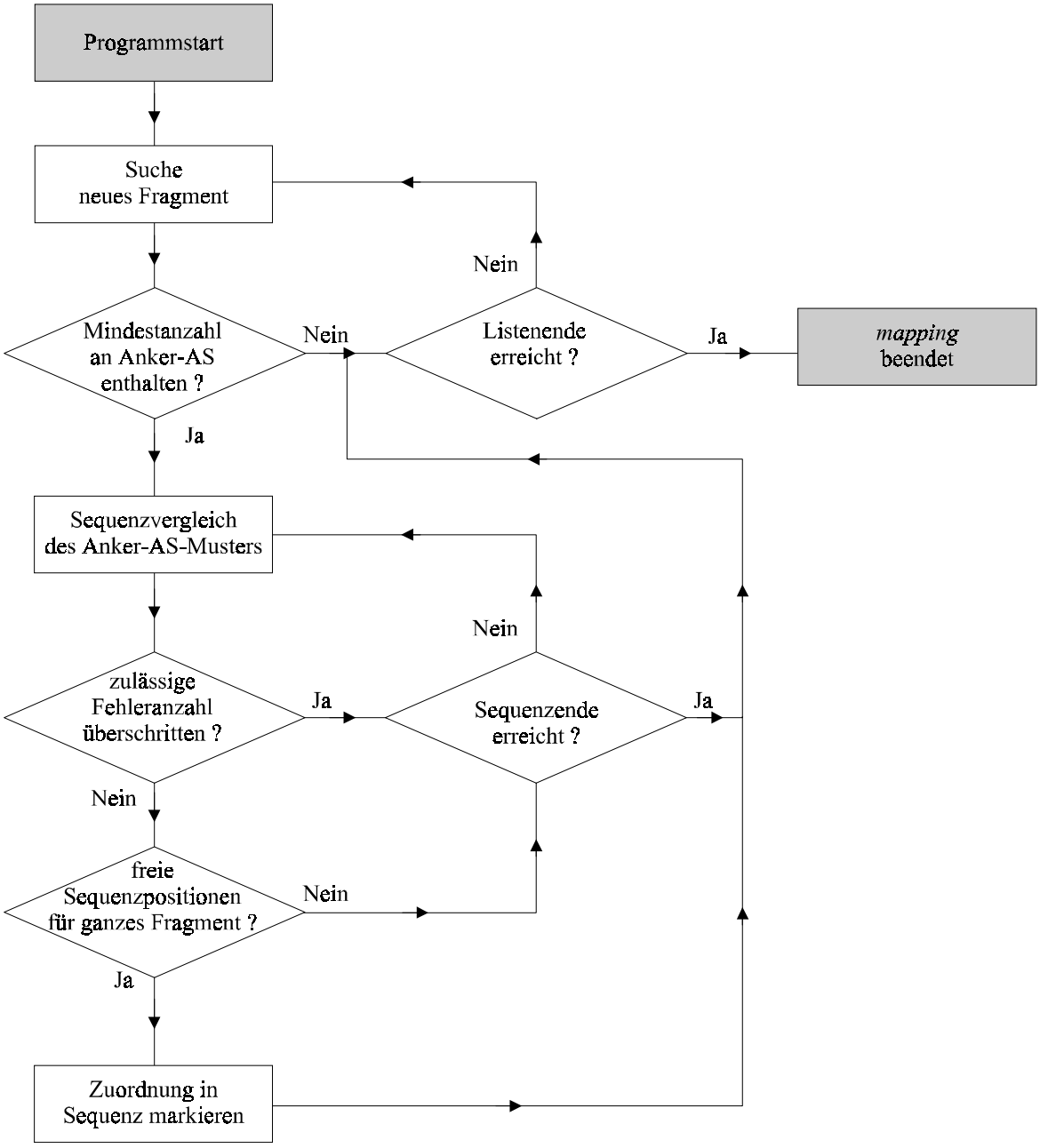

Abbildung 12: *Flussdiagramm des* mapping-*Algorithmus.*

Der Algorithmus wurde anhand eines Datensatzes des Proteins NusB getestet. Der Datensatz enthält 129 Pseudo-Reste, 9 davon besitzen eine unvollständige C^{α}- und C^{β}-Zuordnung. Die Aminosäureerkennung klassifiziert 37 Pseudo-Reste als Ankeraminosäuren A, G, S, T oder V. Die Optimierung der Pseudo-Restliste liefert 9 Bruchstücke mit einer Mindestlänge von 2 Pseudo-Resten. In Tabelle 3 ist das Ergebnis der Sequenzabbildung zusammengefasst. Wählt man für die Mindestanzahl von Ankeraminosäuren pro Fragment 2 und erlaubt 0 fehlerhafte Reste während des Vergleichs, lässt sich nur das Bruchstück zwischen H120 und K129 auf die

Sequenz abbilden. Wird die Zahl der zulässigen Fehler auf 1 erhöht, steigt die Erfolgsquote auf fünf zugeordnete Fragmente. Das Protein NusB verfügt über eine überwiegend α-helikale Struktur. Dies kann für manche Aminosäuren (z.B. K) zu außergewöhnlichen Verschiebungen führen, die mit den von der Aminosäureerkennung verwendeten Verschiebungsgrenzen für V überlappen. Dieses Problem tritt in allen vier neu hinzugekommenen Bruchstücken auf. Alle Bruchstücke verfügen über mindestens sechs Ankeraminosäuren und sind damit trotz des erlaubten Fehlers ausreichen definiert. Eine weitere Erhöhung der Fehlerzahl ist nicht sinnvoll, da dann keine eindeutige Sequenzposition mehr für alle Fragmente gefunden werden kann. So fehlt bei 2 zulässigen Fehlern des Bruchstück H120 – K129, da nur noch eine der drei Ankeraminosäuren für die Sequenzzuordnung übereinstimmen muss. Damit kann H120 – K129 auf fast jede beliebige Sequenzposition zugeordnet werden. Falsche Sequenzzuordnungen treten im Allgemeinen nur bei sehr schlechten Datensätzen auf, d.h. wenn in einem einzelnen Fragment eine große Anzahl von Fehlern enthalten ist. Da durch die Erhöhung der zulässigen Fehlerzahl keine Zuordnungen ungültig gemacht werden, sondern lediglich neue Positionsmöglichkeiten hinzukommen, können korrekte Fragmente nicht auf eine falsche Sequenzposition zugeordnet werden. Mehrdeutige Fragmente werden vom Algorithmus nicht zugeordnet (s.o.).

Mindestanzahl: 2 Erlaubte Fehler: 0	Mindestanzahl: 2 Erlaubte Fehler: 1	Mindestanzahl: 2 Erlaubte Fehler: 2
H120 – K129 (3 Anker-AS)	H120 - K129 (3 Anker-AS)	
	Y100 - G115 (7 Anker-AS) K112 als V klassifiziert	Y100 - G115 (7 Anker-AS) K112 als V klassifiziert
	K82 - V98 (7 Anker-AS) K82 als V klassifiziert	K82 - V98 (7 Anker-AS) K82 als V klassifiziert
	E50 - K67 (9 Anker-AS) K67 als V klassifiziert	E50 - K67 (9 Anker-AS) K67 als V klassifiziert
	E11 - F34 (7 Anker-AS) F34 als V klassifiziert	E11 - F34 (7 Anker-AS) F34 als V klassifiziert

Tabelle 3: *Ergebnis der Sequenzabbildung anhand eines Datensatzes des Proteins NusB (139 AS, 129 Reste im Datensatz). In Klammern hinter den Bruchstücken ist jeweils die Anzahl der Ankeraminosäuren im entsprechenden Fragment angegeben. Die zweite Zeile des Eintrags enthält bei Fragmenten mit fehlerhafter Aminosäureerkennung eine Beschreibung des aufgetretenen Fehlers.*

4.1.2.3 Grafische Benutzeroberfläche

Um das Programm einem größeren Anwenderkreis zugänglich zu machen und die Datenverwaltung komfortabler zu gestalten, wurde das Programm mit einer grafischen Benutzeroberfläche ausgestattet. Die Benutzeroberfläche wurde mit Hilfe des Tcl/Tk-Pakets [114] implementiert.

Das Programm wurde dazu von einem sequentiellem Ablauf auf eine *event* basierte Verwaltung umstrukturiert. Die Kommunikation zwischen Programm und grafischer Benutzeroberfläche erfolgt mittels einer Reihe von speziell entwickelten Schnittstellenroutinen.

Im Rahmen der grafischen Benutzeroberfläche wurde die Datenverwaltung um verschiedene Such- und Sortierfunktionen erweitert. Abbildung 13 zeigt das Hauptmenü von PASTA V3.0. Es enthält Informationen über den momentan geöffneten Datensatz. In Abbildung 14 ist die Maske zur Bearbeitung von Pseudo-Resten gezeigt. Mittels dieser Maske ist eine einfache Manipulation aller in der Pseudo-Restliste enthaltenen Daten möglich

Abbildung 13: *Hauptmenü von PASTA V3.0.*

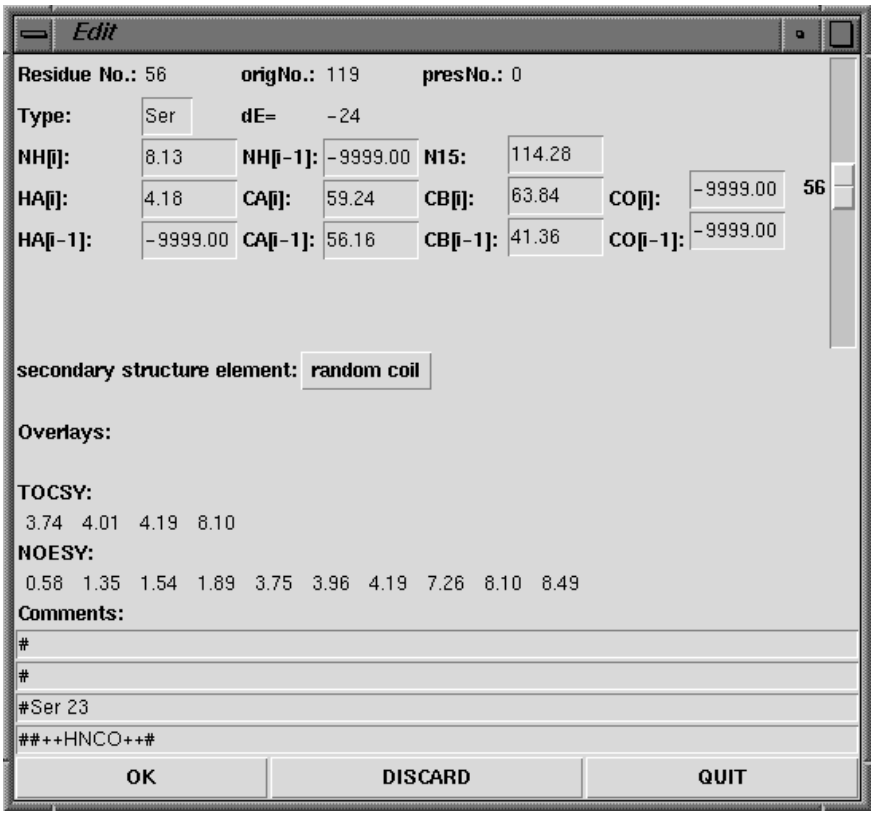

Abbildung 14: *Maske zur Bearbeitung von Pseudo-Restdaten.*

4.2 PASTA Toolkit

4.2.1 Grundkonzept

Das PASTA Toolkit stellt ein Paket von Routinen zur automatisierten Rückgratzuordnung von Proteinen zur Verfügung. Es baut lose auf dem Konzept des Zuordnungsprogramms PASTA [18, 19] auf. Im Hinblick auf die ständig wachsende Anzahl von Programmen [17] [53] gewinnen jedoch Anforderungen wie die freie Erweiterbarkeit, die Kommunikation mit anderen Softwarepaketen und die transparente Datenverwaltung immer mehr an Bedeutung [115] [116]. Während der Entwicklung von PASTA Toolkit standen diese Aspekte stets im Vordergrund der Konzeptionierung [Abbildung 15].

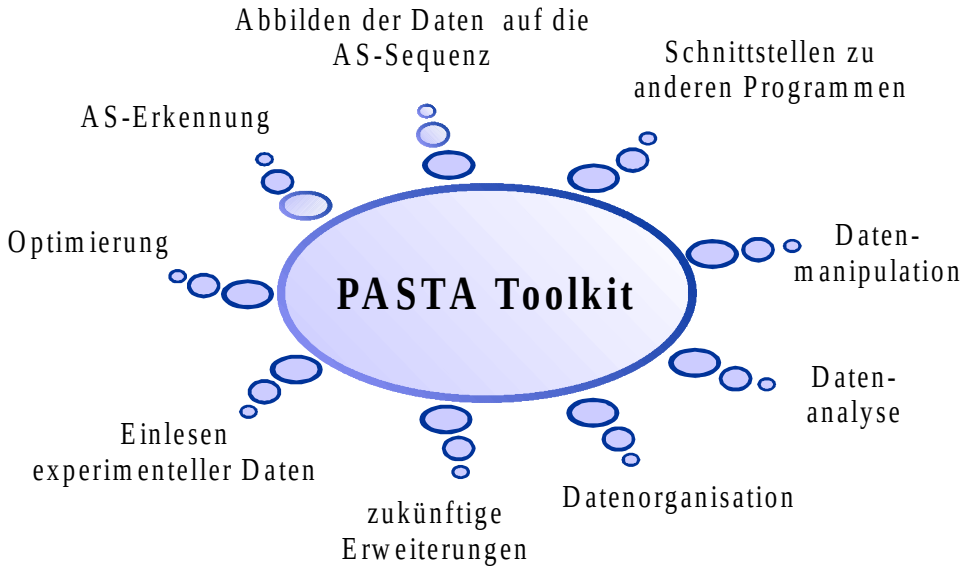

Abbildung 15: *Konzept von PASTA Toolkit.*

Das Paket enthält eine Reihe von Werkzeugen, die den Anwender über den gesamten Prozess der Rückgratzuordnung begleiten. Da die einzelnen Module des PASTA Toolkit voneinander völlig unabhängig sind, ist zu jedem Zeitpunkt der Zuordnung ein Datenaustausch mit anderen Programmen gewährleistet.

Eine vollständige Automatisierung der Zuordnung ist mit den momentan vorliegenden Methoden im Allgemeinen nicht möglich. Daher wird besondere Aufmerksamkeit auf die Datenmanipulation über die grafische Benutzeroberfläche gelegt. Die Daten werden über ein

frei konfigurierbares *spreadsheet* angezeigt und können mit einer Vielzahl von Such- und Sortierfunktionen überprüft werden.

Der Programmcode für die grafische Benutzeroberfläche ist weitgehend vom Code der einzelnen Module getrennt. Dies ermöglicht eine übersichtliche Strukturierung und damit einfachere Wartung des Programmes. Auch im Hinblick auf eine Veröffentlichung des Programms unter *open source*-Richtlinien ist der Programmcode auf diese Weise einfacher für Dritte zu verstehen.

Die Programmimplementierung erfolgte in ANSI-C und Tcl/Tk [114]. Das Programm lässt sich damit problemlos unter den meisten Betriebssystemtypen installieren. Zum Abgabezeitpunkt dieser Arbeit ist das Programm im Internet unter http://www.org.chemie.tu-muenchen.de/people/jl erhältlich.

4.2.2 Der Multieingabefilter

Der erste Schritt einer Zuordnung ist das Einlesen experimenteller Daten und die Konstruktion von Spinsystemen. Dieser Schritt muss besonders sorgfältig erfolgen, da er als einzige Schnittstelle zum Experiment entscheidend für das Ergebnis aller weiteren automatisierten Funktionen ist.

Im PASTA Toolkit geschieht das Einlesen der Daten mit dem Multieingabefilter. Die Experimentdaten müssen dazu als ASCII-Peaklisten vorliegen, wie sie aus gängigen Standard-NMR-Software-Paketen exportiert werden können. Im Moment werden nur die Formate von AURELIA (Bruker), TRIAD (Tripos) und XEASY [65] direkt unterstützt. Eine Konvertierung von ASCII-Listen anderer Formate in ein für PASTA Toolkit lesbares Format ist jedoch problemlos möglich.

Mit dem Multieingabefilter können alle im Moment gebräuchlichen 3D-Experimente zur Zuordnung des Proteinrückgrats verarbeitet werden. Darüber hinaus bietet der Filter die Möglichkeit, eigene 3D-Experimente zu definieren und in einer Vorlagen-Datenbank abzuspeichern. 2D-Experimente werden, mit Ausnahme des ^{15}N-HSQC, momentan nicht unterstützt. Das Einlesen von 4D-Experimenten wird unterstützt, konnte jedoch aufgrund mangelnder Testdatensätze nicht ausreichend geprüft werden.

Die eingelesenen Daten werden in eine sogenannte Pseudo-Rest-Liste einsortiert. Jeder Pseudo-Rest repräsentiert alle bekannten Daten einer Aminosäure (s. 4.2.7). Besteht noch keine Pseudo-Rest-Liste, wird das eingelesene Experiment als *Startspektrum* verwendet. Das bedeutet, zu jedem Signal der Peakliste wird ein Pseudo-Rest erzeugt. Werden die Signale in eine bereits bestehende Pseudo-Rest-Liste eingelesen, versucht der Eingabefilter die Signal-Daten automatisch den richtigen Pseudo-Resten zuzuordnen. Dazu werden bei einem 3D-Experiment zwei Dimensionen als Referenz-Dimensionen definiert. Existiert zu den Referenzwerten ein Signal in der Liste, wird dieser an passender Stelle in die Pseudo-Rest-Liste eingetragen. Findet sich mehr als ein Signal zu den Referenzverschiebungen, stehen drei Vergleichsmethoden zur Verfügung, um trotzdem die richtige Einordnung zu erreichen:

- Unterscheidung der Signalphase
- Intensitätsvergleich
- Vergleich mit schon vorhandenen Verschiebungsdaten

Üblicherweise dient das Verschiebungspaar $^{15}N/H^N$ als Referenz. Diese Verschiebungen finden sich in den meisten Standardexperimenten wieder. Als *Startspektren* empfehlen sich somit das ^{15}N-HSQC oder bei größeren Molekülen das HNCO, da dieses über eine ausgezeichnete Signalintensität und Dispersion verfügt.

Die Parametereingabe erfolgt über eine ASCII-Datei, die wahlweise manuell oder über eine grafische Benutzeroberfläche konfiguriert werden kann. Die Benutzeroberfläche stellt eine Bibliothek von vordefinierten Experimenten (*presets*) zur Verfügung. Diese Bibliothek kann vom Anwender mit neuen Einträgen vervollständigt werden. Abbildung 16 zeigt die grafische Benutzeroberfläche des Multieingabefilters.

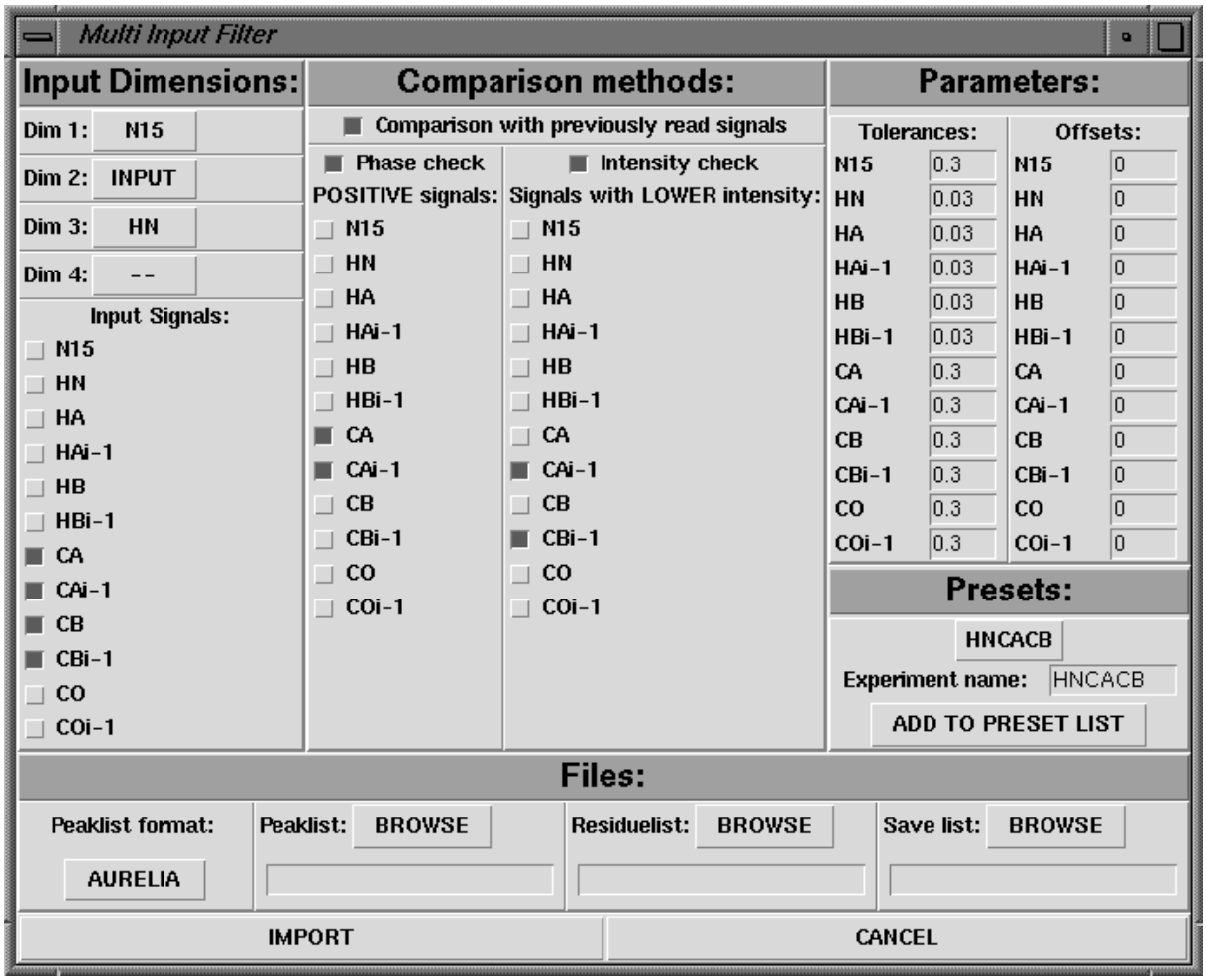

Abbildung 16: *Grafische Benutzeroberfläche des Multieingabefilters, konfiguriert für ein HNCACB-Spektrum.*

Die Funktion des Eingabefilters sei beispielhaft anhand eines HNCACB-Spektrums erklärt. Das HNCACB enthält folgende Daten: ^{15}N, H^N, $C^{\alpha}[i]$, $C^{\alpha}[i-1]$, $C^{\beta}[i]$, $C^{\beta}[i-1]$. ^{15}N und H^N werden als Referenz definiert. Pro Pseudo-Rest, d.h. für jedes Paar von Referenzverschiebungen, werden folglich 4 Signale in der Kohlenstoff-Dimension erwartet. C^{α}- und C^{β}-Verschiebungen können im HNCACB über unterschiedliche Phasen voneinander getrennt werden. Die Unterscheidung zwischen [i]- bzw. [i-1]-Verschiebungen erfolgt anschließend über die Signalintensität. Da die Intensität jedoch kein eindeutiges Kriterium darstellt und in manchen Fällen zu einer falschen Identifizierung des Signals führen kann, wird zusätzlich noch ein Abgleich mit zuvor eingelesenen Verschiebungen, z.B. einem HNCA, durchgeführt. Auf diese Weise werden die gesammelten Daten aus der Kombination

mehrerer Spektren mit komplementärem Informationsgehalt, wie z.B. HNCA, HNCACB und CBCACONH, zur Einordnung der Signale in Pseudo-Reste genutzt.

Das Ergebnis der Einleseprozedur wird in einem Protokoll-File festgehalten (s. Anhang). Das Protokoll zeigt für alle Pseudo-Reste an, welche Signale aufgrund der Referenzverschiebungen gefunden wurden und welche Fehler bei der Zuordnung der Signale auftraten.

Die Qualität des Ergebnisses ist stark von der Qualität der automatisch generierten Peaklisten abhängig. Es wurde jedoch bewusst darauf verzichtet, eine eigene Routine zum Erzeugen einer Peakliste zur Verfügung zu stellen, da die obengenannte Software hierfür schon eine Reihe leistungsfähiger Routinen zur Verfügung stellt. Allerdings sind dem Standardansatz des sogenannten *peak pickings* Grenzen gesetzt, da die *peak picking*-Module normalerweise keine Information über den Inhalt der untersuchten Spektren verwenden. Dies führt zu fehlenden Signale bzw. Artefakten. Der Eingabefilter ist gegenüber diesen Problemen relativ robust. Für eine weitere Automatisierung könnten die Konstruktion der Spinsysteme mit automatisch generierten Peaklisten im Multieingabefilter durch leistungsfähigere Methoden ersetzt werden.

Ein Beispiel für eine solche Methode ist der Ansatz von Croft et al. [68], bei dem nicht einzelne Signale aus den Spektren herausgesucht werden, sondern über eine Kombination mehrerer Experimente ganze Spinsysteme mit Verfahren aus der Mustererkennung analysiert werden (s. Automatisierung).

Der modulare Aufbau des PASTA Toolkit erlaubt solche Modifikationen in der Zukunft, da jedes einzelne Modul unabhängig von den restlichen Programmteilen arbeitet und somit zusätzliche Verfeinerungen in den Zuordnungsprozess mit einbezogen werden können.

4.2.3 Das Optimierungsmodul

Das Optimierungsmodul ist das Herzstück des Programmpakets. Es hat die Aufgabe, die Pseudo-Reste so zu sortieren, dass ihre Reihenfolge der realen Anordnung der Aminosäuren in der Proteinsequenz entspricht. Im PASTA Toolkit wird für diese Aufgabe der *fast TA*

Algorithmus eingesetzt. Der Algorithmus wurde im Rahmen dieser Arbeit auf Basis von *threshold accepting* [112] entwickelt und auf das Zuordnungsproblem angepasst (s. 4.1.2.1).
Die Pseudoenergiefunktion wurde weitgehend von PASTA V3.0 übernommen (s. 4.1). Der optionale Energiewert zum Sequenzvergleich E_{SEQ} wird jedoch nicht mehr verwendet. Ein Vergleich der Rechenergebnisse von PASTA-V3.0-Programmläufen mit und ohne Berechnung von E_{SEQ}, aber ansonsten identischer Parameterwahl zeigt keine signifikanten Unterschiede in der Lösungsqualität (Tabelle 4). Die Anzahl der Zuordnungsfehler bleibt unverändert. Alle fehlerhaft zugeordneten Reste sind mit anderen Positionen überlagert bzw. verfügen nicht über den vollständigen Signalsatz von C^{α}, C^{β}, C' und H^{α}. Sie können daher auch bei Verwendung von E_{SEQ} nicht weiter voneinander unterschieden werden.

Schrittanzahl	1000	2000	3000	4000	5000
Fragmentabbrüche (E = 130) **mit** E_{SEQ}	12	11	8	9	13
Fragmentabbrüche (E = 130) **ohne** E_{SEQ}	11	12	13	9	12
Zuordnungsfehler trotz (E < 0) **mit** E_{SEQ}	6	7	7	7	7
Zuordnungsfehler trotz (E < 0) **ohne** E_{SEQ}	6	7	7	7	6

Tabelle 4: *Ergebnisse der Rechnungen mit und ohne* E_{SEQ} *(Datensatz: NusB, 129 AS, T = 150, S = 200). Jeder Wert entspricht dem Durchschnitt von 5 Rechnungen.*

Für einen erfolgreichen Einsatz von E_{SEQ} wird eine möglichst große Anzahl typisierter Aminosäuren zum Vergleich mit der Sequenz benötigt. Da mit der verwendeten Aminosäureerkennung nur fünf der zwanzig Aminosäuren sicher erkannt werden können, ist der Einfluss der Sequenzvergleichsenergie auf die Rechnung gering. Zusätzlich werden für die Typisierung der Aminosäure die komplette C^{α}- und C^{β}-Zuordnung eines Restes benötigt. Fehlt eine der beiden Verschiebungen, kann von der Aminosäureerkennungsroutine keine Aussage gemacht werden. Insgesamt stehen damit in einem durchschnittlichen Datensatz weniger als 25% aller Aminosäurereste für die Sequenzbewertung zur Verfügung. Im

untersuchten Beispieldatensatz (NusB, 129 Reste) ist mit 56 von 129 Resten ein überdurchschnittlich hoher Anteil der Zuordnungsliste von der Aminosäureerkennung klassifiziert worden. 8 dieser 56 Reste sind jedoch fehlerhaft erkannt worden und stören bei der Energiebewertung mit E_{SEQ}. Trotz des großen Anteils an identifizierten Resten im Datensatz bringt der Einsatz von E_{SEQ} keine Verbesserung der Lösungsqualität. Die Ergebnisse in Tabelle 4 sind für beide Rechenmethoden fast identisch. Die Anzahl der fehlerhaften Pseudo-Reste verändert sich nicht.

Im Gegensatz dazu nimmt jedoch die Rechenzeit bei Benutzung von E_{SEQ} in etwa um einen Faktor 5 zu (Abbildung 17). Ein weiterer Einsatz von E_{SEQ} ist daher nicht sinnvoll.

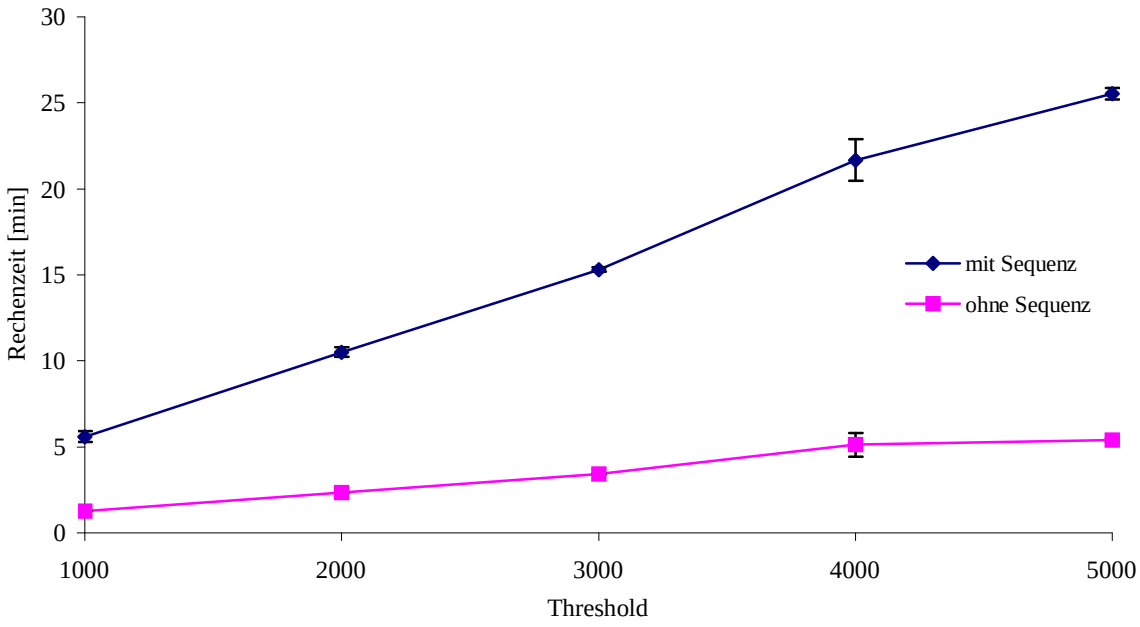

Abbildung 17: *Vergleich des Zeitbedarfs von PASTA-Rechnungen mit und ohne* E_{SEQ} *(Datensatz: NusB, 129 AS, S = 200, 5 Rechnungen pro Datenpunkt; Alle Rechnungen wurden auf einer SGI Octane R10000/180MHz durchgeführt).*

Um ein breiteres Spektrum von Informationen während der Optimierung nutzen zu können, wurde die Eingabe der Optimierungsparameter flexibel gestaltet. Dies ermöglicht eine sinnvolle Auswahl der in der Pseudo-Restliste gespeicherten Informationen. Da die Pseudo-Restliste als Grundlage der einzelnen PASTA-Toolkit-Module gleichzeitig noch Datenbankfunktionen übernimmt, ist es nicht immer erwünscht, alle vorhandenen Daten in die Optimierung mit einzubeziehen. Insbesondere wenn Daten aus Experimenten

unterschiedlicher Qualität verwendet werden, kann das Rechenergebnis empfindlich gestört werden, wenn grundsätzlich die Information aller eingelesenen Kerne verwendet werden muss. In Verbindung mit dem Modul zum Matrixvergleich (s. 4.2.4) lassen sich durch mehrfaches Rechnen mit unterschiedlichen Parametersätzen Artefakte aus dem Datensatz herausmitteln.

Zusätzlich wurde neben der Optimierung über [i-1]-Verknüpfungen auch die Möglichkeit der Berücksichtigung von [i+1]-Informationen hinzugefügt. Im Augenblick hat die Verwendung von [i+1]-Verschiebungen allerdings nur für die ^{15}N-Kerne des Proteinrückgrates praktische Relevanz [117].

Ebenso wurde eine Möglichkeit zur Änderung der Energiewertdefinitionen mittels einer einfachen Parameterübergabe implementiert. Somit können jederzeit neue, bisher noch nicht zur Optimierung vorgesehene Kerne aufgenommen werden, ohne den Quellcode des Programms zu verändern.

Das Optimierungsmodul verfügt damit über folgende Eingabeparameter:

- Startwert für die Toleranzschwelle T
- Schrittanzahl S
- Dateiname Eingabedatei/Ausgabedatei
- Optimierungsparameter für jeden zu berücksichtigenden Kern:
    - Kernname
    - Zu optimierende sequentielle Information, d.h. [i-1] und/oder [i+1]
    - Pseudoenergiebewertung

Alle Parameter werden mittels eines ASCII-Files übergeben (s. Anhang). Die Ausgabe des Rechenergebnisses erfolgt in Form der optimierten Pseudo-Restliste. In Abbildung 18 ist die grafische Benutzeroberfläche des Optimierungsmoduls gezeigt. Im Moment steht über die grafische Benutzeroberfläche nur eine eingeschränkte Auswahl häufig verwendeter Parameter zur Verfügung, um die Benutzung möglichst einfach zu gestalten. Damit alle zur Verfügung stehenden Parameter genutzt werden können, muss eine manuelle Konfiguration der ASCII-Eingabedatei vorgenommen werden.

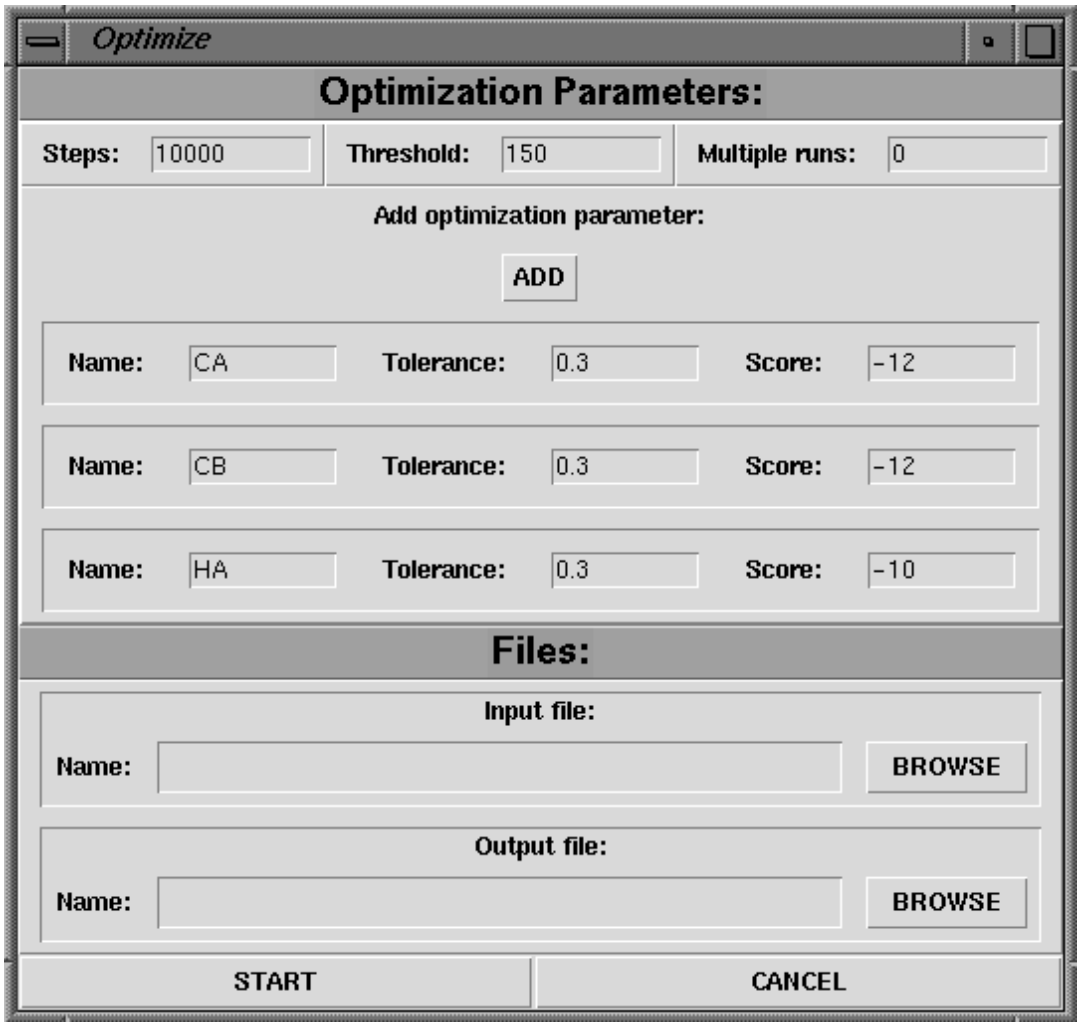

Abbildung 18: *Grafische Benutzeroberfläche zur Parametereingabe für das Optimierungsmodul.*

4.2.4 Die Vergleichsmatrix

Das Matrixmodul schließt sich an die Optimierung an. Es dient zum Vergleich der Ergebnisse mehrerer Einzelrechnungen. Da die Optimierung mittels kombinatorischer Minimierung erfolgt, können die Lösungen verschiedener Rechnungen durchaus unterschiedlich ausfallen. Solche Unterschiede treten im Allgemeinen an Stellen der Pseudo-Restliste auf, an denen keine eindeutige Zuordnung möglich ist, d.h. an denen also starke Überlagerungen oder falsche Verschiebungswerte vorliegen. Anhand eines einzelnen Optimierungslaufes kann nur

schwer abgeschätzt werden, welche Auswirkungen Überlagerungen auf die Gesamtlösung zeigen und welche Zuordnungspositionen als gesichert betrachtet werden können.

Das Matrixmodul ermöglicht durch den Vergleich mehrerer Optimierungsläufe ein Abschätzen der Lösungsqualität bzw. der generellen Lösbarkeit des Problems. Eine solche Abschätzung ist insbesondere im Anfangsstadium des Zuordnungsprozesses wertvoll, wenn nur wenige Verschiebungsdaten pro Rest vorliegen. Über den Matrixvergleich kann entschieden werden, welche Stellen der Zuordnungsliste schon ausreichend definiert sind und wo zu einer Verbesserung des Ergebnisses noch weitere Daten eingelesen werden müssen. Abbildung 19 zeigt eine schematische Darstellung des Matrixvergleichs.

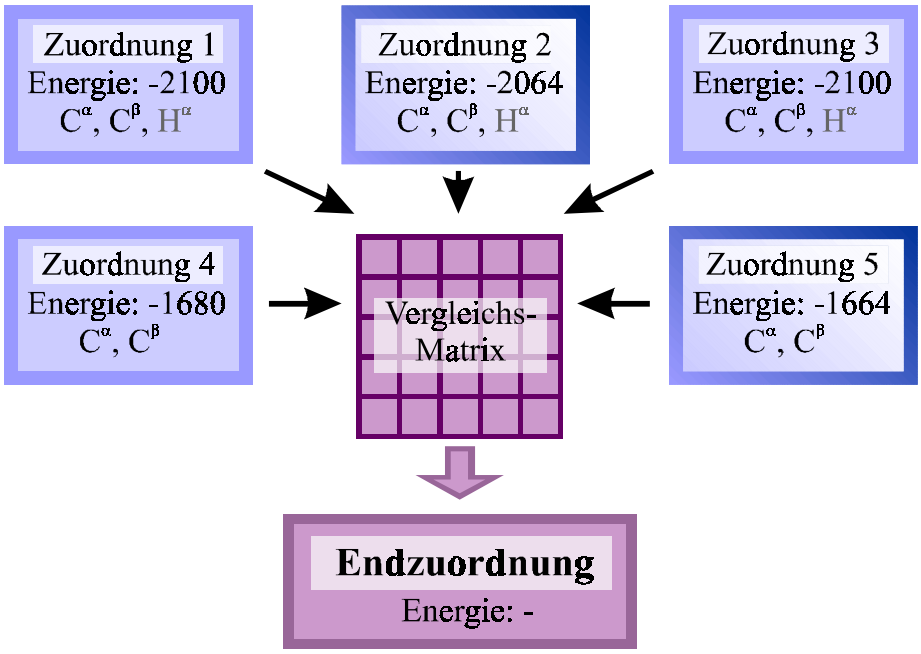

Abbildung 19: *Schematische Darstellung des Matrixvergleichs im PASTA Toolkit.*

Die eigentlich Aufgabe des Matrixvergleichs besteht jedoch in der Bestätigung und Extraktion gesicherter Bruchstücke der Zuordnungsliste. Insbesondere an den Enden eines Zuordnungsfragments treten häufig Fehler auf, die auf den ersten Blick nur schwierig zu identifizieren sind. Dies ist der Fall, wenn der eigentliche Rest entweder fehlt, unzureichende Daten oder falsche Verschiebungsinformationen enthält. Unter Verwendung des Matrixvergleichs werden solche Fehlzuordnungen herausgemittelt, da die betroffenen Reste meist statistisch über mehrere Zuordnungspositionen verteilt werden. Während des

Matrixvergleichs werden nur die Verknüpfungen in die endgültige Zuordnung übernommen, die signifikant häufiger auftreten als andere Verknüpfungsmöglichkeiten.

Auch zur Nutzung von Daten schlechter Qualität zur Optimierungsrechnung kann der Matrixvergleich verwendet werden. Verfügt ein Datensatz beispielsweise über verlässliche C^{α}- und C^{β}-Informationen aber sehr schlechte H^{α}-Werte [Abbildung 19], können mehrere Optimierungsläufe mit unterschiedlichen Parametersätzen miteinander verknüpft werden. In diesem Fall könnte z.B. die eine Hälfte der Optimierungsläufen unter Berücksichtigung von C^{α}, C^{β} und H^{α} durchgeführt werden, während die andere Hälfte nur mit C^{α} und C^{β} unter Vernachlässigung von H^{α} gerechnet werden wird (vgl. 5). Reste mit korrekten, nicht überlagerten Kohlenstoffverschiebungen, aber falschem H^{α}-Wert weisen in der Vergleichsmatrix immer noch in mindestens 50% der Zuordnungen eine feste Verknüpfung auf und lassen sich daher als korrekte Zuordnung erkennen. Für Reste, deren Kohlenstoffverschiebungen überlagert sind und die ohne zusätzliche Informationen nicht zugeordnet werden können, ist es aber trotzdem möglich, den H^{α}-Wert zu nutzen. Alle Reste mit einer korrekten H^{α}-Verschiebung, für die sich über die Protonenverschiebung eine eindeutige Zuordnung ergibt, zeigen ebenfalls in mindestens 50% der Optimierungsläufe eine einheitliche Verknüpfung. Reste mit falschen H^{α}-Werten stören die Zuordnung nicht, da nur die Hälfte aller Rechnungen mit H^{α} durchgeführt wird.

Der Matrixvergleich steht in enger Verbindung mit dem Optimierungs- und dem *mapping*-Modul. Im Optimierungsmodul ist eine Option zur Mehrfach-Optimierung mit gleichen Parametersätzen vorgesehen (s. 4.2.3), die als Grundlage für den Matrixvergleich dienen kann. Das *mapping*-Modul verwendet die Matrixvergleichstechnik zur Extraktion der gesicherten Fragmente aus der Zuordnungsliste vor dem eigentlichen Abbildungsschritt auf die Proteinsequenz (s. 4.2.6). Da die Ergebnisse der Vergleichsmatrix nicht innerhalb der Pseudo-Restliste gespeichert werden können, verfügt das Matrixmodul über eine spezielle Ausgabedatei. Diese ASCII-Datei enthält für jede der möglichen Restekombinationen die Anzahl, wie oft dieses Restepaar bei den bisher eingelesenen Zuordnungslisten aufgetreten ist (s. 4.2.7.1). Ein Restepaar liegt vor, wenn es in der Zuordnungsliste durch eine negative Pseudoenergie verbunden ist. Die Matrixdatei kann zur Speicherung von Zwischenergebnissen dienen und somit als Datengrundlage des *mapping*-Moduls eingelesen werden.

Nachfolgend ist die Funktionsweise des Matrixvergleichs beschrieben. Zum Matrixvergleich wird eine $N \times (N-1)$-Matrix (N ist die Anzahl der Pseudo-Reste) aufgestellt, die alle möglichen Pseudo-Restkombinationen der Zuordnungsliste enthält. Wird eine neue (bereits optimierte) Pseudo-Restliste eingelesen, werden diejenigen Positionen der Matrix um 1 erhöht, für die eine Verknüpfung, also ein Verbindung durch einen negativen Energiewert, gefunden wird. Die Koordinaten für ein Restepaar in der Matrix orientieren sich dabei an der Originalnummer der beiden Pseudo-Reste. Der Algorithmus zur Extraktion der Fragmente funktioniert dann wie folgt:

1. Suche ein Restepaar (x,y) als Startpunkt, das mindestens einmal häufiger als alle anderen Zuordnungsmöglichkeiten dieser beiden Reste vorkommt. Streiche x aus der Zuordnungsliste. Setze x gleich y.
2. Suche wiederum ein Paar (x,y), dass mindestens einmal häufiger als alle anderen Zuordnungsmöglichkeiten vorkommt.

   Existiert ein solches Paar, streiche x aus der Zuordnungsliste, setze x gleich y und wiederhole Schritt 2.

   Existiert kein solches Paar, streiche x aus der Zuordnungsliste und markiere es als nicht zugeordnet. Beginne anschließend wieder bei Schritt 1.
3. Der Algorithmus ist beendet, wenn in der Zuordnungsliste nur noch ein Rest enthalten ist.

Der Algorithmus sucht zuerst ein gesichertes Zuordnungspaar heraus und verwendet dieses als Saatpunkt für die weitere Konstruktion eines Fragments. Die Fragmentbildung bricht ab, wenn an einer Stelle keine eindeutige Zuordnung mehr gefunden werden kann. In diesem Fall wird nach dem Anfang eines neuen Fragments gesucht. Kann kein Rest mehr zugeordnet werden, ist der Algorithmus beendet. Die Ausgabe des Programms gibt für jeden Rest die getroffene Zuordnung bzw. die Anzahl der möglichen Zuordnungen an. Abbildung 20 zeigt ein Flussdiagramm dieses Algorithmus.

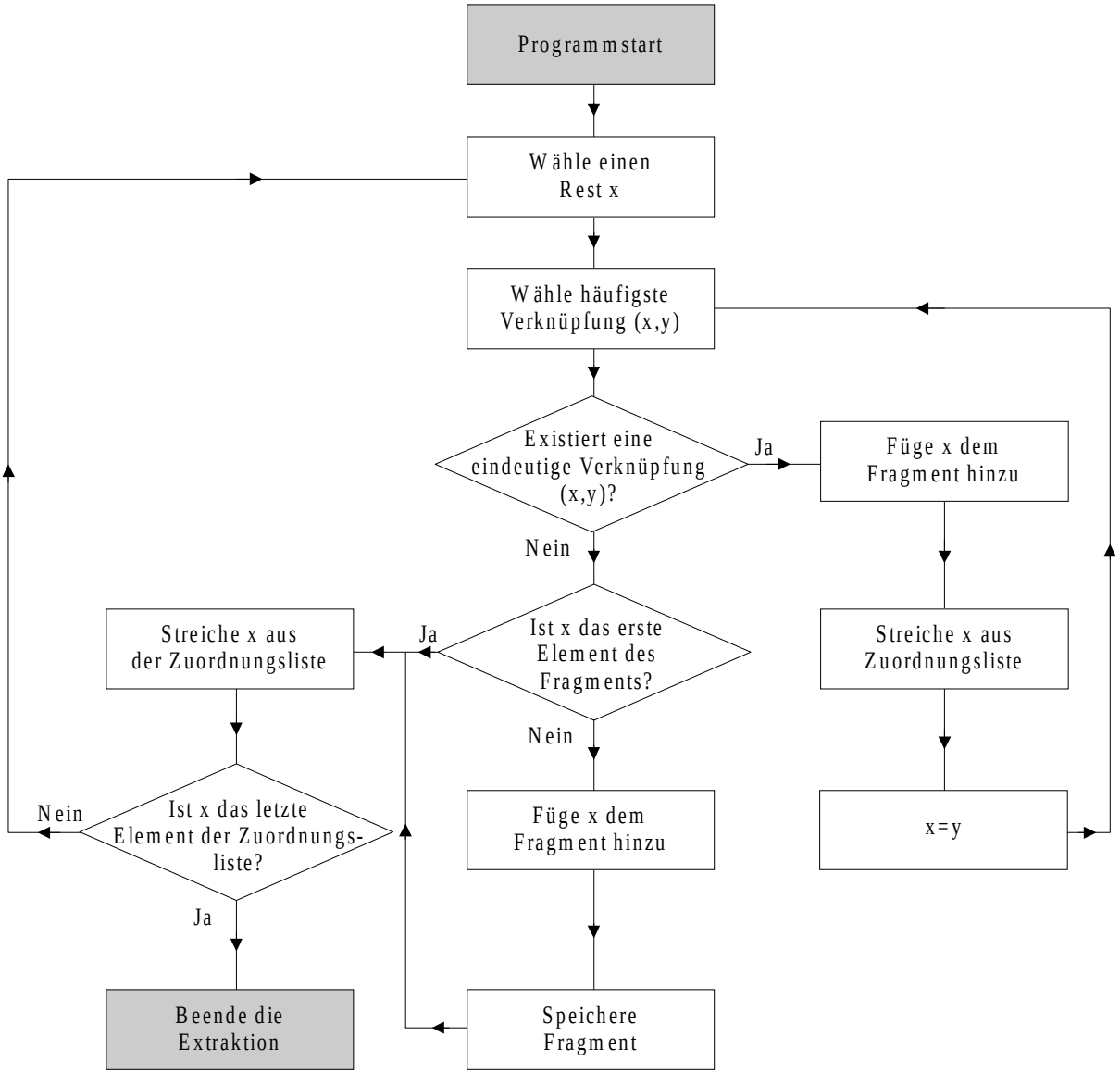

Abbildung 20: *Flussdiagramm des Algorithmus zur Extraktion der Zuordnungsfragmente aus der Pseudo-Restliste.*

Die Effizienz des Matrixvergleichs ist nachfolgend an zwei Beispielen gezeigt.

Für den Datensatz von NusB (139 Aminosäuren, 129 Reste im Datensatz) zeigen sich nur geringe Unterschiede zwischen Einzelrechnung und Matrixvergleich (Tabelle 5). Die Anzahl der erhaltenen Bruchstücke steigt von 12 auf 15 an. Dies ist hauptsächlich auf die abnehmende Zahl der Zuordnungsfehler zurückzuführen: Treten Zuordnungsfehler nicht am Rand, sondern innerhalb von Bruchstücken auf, werden diese Bruchstücke beim Matrixvergleich in mehrere Teile gespalten. Auf diese Weise erhöht sich die Gesamtanzahl der Fragmente durch die Spaltung fehlerhafte Bruchstücke. Ansonsten ändert sich das

Ergebnis nur unwesentlich. Alle Zuordnungsfehler werden durch Überlagerungen der chemischen Verschiebung verursacht, da der Datensatz ausschließlich verifizierte Reste, d.h. keine Artefakte oder falsche Verschiebungswerte, enthält. Eine Erhöhung der Anzahl der in den Matrixvergleich einbezogenen Rechnungen könnte die Fehleranzahl noch weiter absenken, ist aber angesichts der geringen Fehleranzahl nicht sinnvoll.

Für VAT-N zeigen sich die Auswirkungen des Matrixvergleichs wesentlich deutlicher (Tabelle 6). Der Datensatz enthält 40 Reste mehr als für die Zuordnung erforderlich sind (216 Reste befinden sich im Datensatz; VAT-N besitzt 185 Aminosäuren, 9 davon sind P). Ein Aussortieren der überflüssigen Reste aufgrund einfacher Kriterien ist jedoch nicht möglich. Mithilfe des Matrixvergleichs lässt sich dennoch ein Großteil der korrekten Restepaare extrahieren. Wie schon beim Datensatz von NusB steigt die Gesamtzahl der Bruchstücke nach dem Matrixvergleich an. Auch hier befinden sich unsichere Positionen offensichtlich nicht nur am Rand, sondern auch innerhalb der Fragmente. Die Zahl der fehlerhaften Bruchstücke sinkt von 10 auf 4 ab. Die Anzahl der insgesamt zugeordneten Reste ist bei der Einzelrechnung deutlich höher als nach dem Matrixvergleich. Sie liegt mit 194 Resten sogar höher als die theoretisch mögliche Anzahl von 176 Aminosäuren. Das korrigierte Ergebnis nach Abzug von falsch zugeordneten oder mehrdeutigen Resten zeigt für die Einzelrechnung mit 91,5% Zuordnung (161 Reste) eine etwas höhere Erfolgsquote als für den Matrixvergleich (87,5%, 154 Reste). Die Fehlerrate beträgt jedoch mit 22,2% mehr als das doppelte der Fehlerrate des Matrixvergleichs (9,4%). Da das Ergebnis als Grundlage für das *mapping*-Modul dienen soll, ist ein möglichst verlässliches Ergebnis erforderlich, um Folgefehler zu vermeiden.

Aus den Rechenergebnissen für die beiden Datensätze kann gefolgert werden, dass der Matrixvergleich insbesondere dann effektiv ist, wenn die Zuordnungsliste, wie bei VAT-N, eine größere Anzahl an Artefakten enthält. In diesem Fall können die gesicherten Bruchstücke der Zuordnung einfach identifiziert und für weitere Schritte verwendet werden. Der Matrixvergleich eignet sich auch, um überlagerte Positionen aufzuzeigen, an denen weitere Informationen für eine eindeutige Zuordnung benötigt werden (s. NusB-Datensatz). Ebenso kann die generelle Lösbarkeit eines Zuordnungsproblems mit den vorliegenden Daten beurteilt werden.

NusB (129 Reste)	1 Rechnung	Matrixvergleich (5 Rechnungen)
Bruchstücke	12	15
Bruchstücke ≤ 3 Reste	5	7
Fehlerhafte Bruchstücke	8	2
Zugeordnete Reste	126	123
Fehler	8	3
Mehrdeutigkeiten	0	0
Korrekt zugeordnete Reste	118	120

Tabelle 5: *Ergebnisse des Matrixvergleichs anhand einer Zuordnungsliste des Proteins NusB (139 AS, 129 Reste im Datensatz).*

VAT-N (216 Reste)	Einzelrechnung	Matrixvergleich (10 Rechnungen)
Bruchstücke	24	29
Bruchstücke ≤ 3 Reste	12	12
Fehlerhafte Bruchstücke	10	4
Zugeordnete Reste	194	170
Fehler	16	5
Mehrdeutigkeiten	27	11
Korrekt zugeordnete Reste	161	154

Tabelle 6: *Ergebnisse des Matrixvergleichs anhand einer Zuordnungsliste des Proteins VAT-N (185 AS, 216 Reste im Datensatz).*

Abbildung 21 zeigt die grafische Benutzeroberfläche zur Konfiguration des Matrixvergleichs. Die Ergebnisse können in einer ASCII-Protokolldatei abgespeichert werden. Diese Protokolldatei zeigt für jeden Rest den häufigsten Nachfolger und die Anzahl der Rechnungen, in denen diese Verknüpfung auftritt. Ebenso sind die extrahierten Fragmente angegeben. Das Dateiformat ist im Anhang beschrieben (s. Anhang).

Abbildung 21: *Grafische Benutzeroberfläche des Matrixvergleichs.*

4.2.5 Aminosäureerkennung

Die Aminosäureerkennung (*typing*) ist ein notwendiger Schritt für das Abbilden der Zuordnungsdaten auf die Sequenz. Es existieren zwei unterschiedliche Ansätze zur Aminosäureerkennung: Die Analyse von chemischen Verschiebungen und die Typisierung von Spinsystemen. Im PASTA Toolkit erfolgt die Aminosäureerkennung mittels chemischer Verschiebungen, da für die Untersuchung der Spinsysteme eine vollständige Auswertung der Seitenkettendaten notwendig ist. Diese ist im Allgemeinen aufgrund starker Überlagerungen und fehlender Signale nicht mittels automatisch generierter Peaklisten möglich (s. 3).

Die Aminosäureerkennung im PASTA Toolkit verwendet die C^{α}- und C^{β}-Verschiebungen als Datengrundlage. Beide Verschiebungen sind aufgrund ihrer vergleichsweise großen

Dispersion auch für fast alle anderen Zuordnungsschritte relevant. Daher kann für die meisten Datensätze angenommen werden, dass sie eine ausreichende Zahl von C^{α}- und C^{β}-Verschiebungen für die Aminosäureerkennung enthalten.

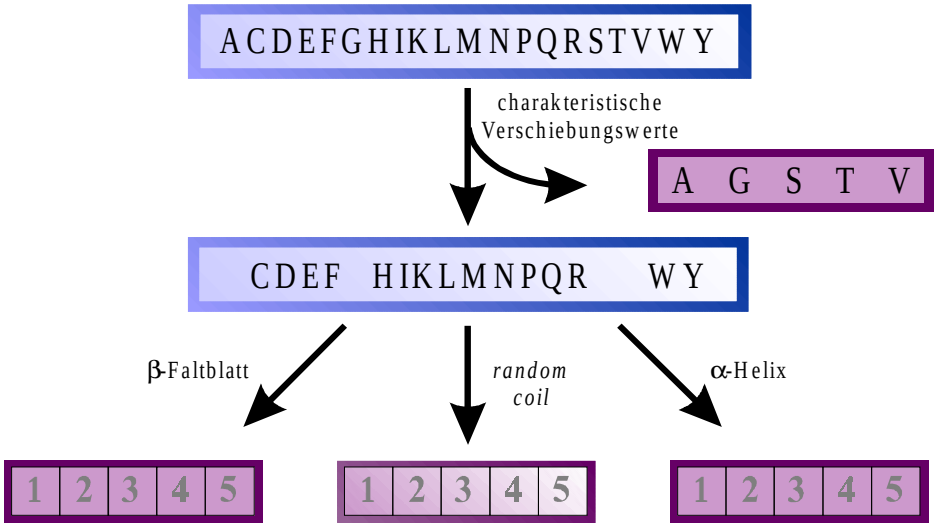

Abbildung 22: *Schematische Darstellung der Aminosäureerkennung in PASTA Toolkit. Die Aminosäureerkennung besteht aus zwei Schritten: 1. Erkennung der Ankeraminosäuren A, G, S, T und V. 2. Bestimmung der fünf besten Zuordnungsmöglichkeiten (unter Berücksichtigung der Sekundärstruktur).*

Die Aminosäureerkennung teilt sich in zwei Schritte (Abbildung 22). Zuerst wird eine Erkennung der sogenannten Ankeraminosäuren durchgeführt. Dabei handelt es sich um die Aminosäuren A, G, S, T und V, die allein aufgrund ihrer chemischen Verschiebungen eindeutig von den restlichen Aminosäuren unterschieden werden können. Abbildung 23 verdeutlicht dies anhand eines C^{α},C^{β}-Verschiebungsdiagrammes. Die im Programm verwendeten Verschiebungsgrenzen sind in Tabelle 7 angegeben. S und T nehmen eine Sonderstellung ein, da für diese beiden Aminosäuren der C^{β}-Wert weiter tieffeldverschoben liegt als der C^{α}-Wert. Die Typisierung von S und T ist daher besonders einfach. Die Erkennung von V ist dagegen etwas fehleranfälliger als die der anderen Ankeraminosäuren, da die Sekundärstruktur einen äußerst großen Einfluss auf die C^{β}-Verschiebung ausübt. Dies macht sich insbesondere im β-Faltblatt-Bereich bemerkbar, wo die chemische Verschiebung für C^{β} sogar Werte größer 36 ppm annehmen kann. Da die Verschiebungsgrenzen für C^{β} jedoch aufgrund von Überlappungen mit I nicht weiter als 35 ppm ausgedehnt werden können,

wird V nicht immer korrekt identifiziert. P stört die V-Erkennung trotz ähnlicher C^{α}- und C^{β}-Verschiebungen nicht, da die Aminosäureerkennung unter der Annahme von N/H^{N}-basierten Datensätzen stattfindet, in denen Prolin aufgrund des fehlenden Amidprotons nicht direkt beobachtet werden kann.

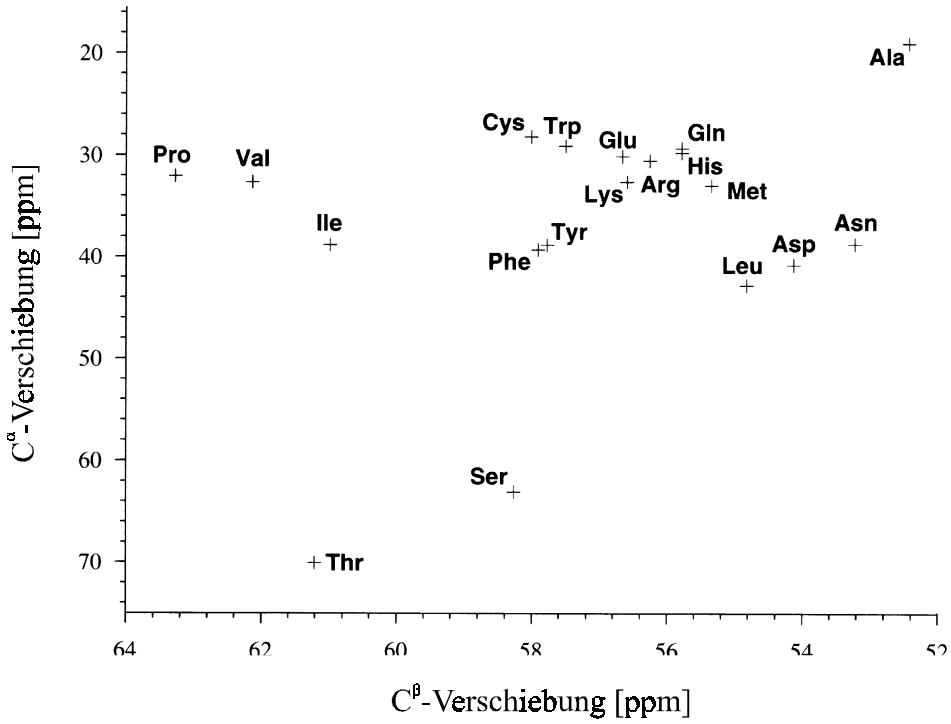

Abbildung 23: *Diagramm der durchschnittlichen C^{α}- und C^{β}-Verschiebungen der zwanzig natürlichen Aminosäuren. Die Aminosäuren A, G, S, T und V können aufgrund ihrer isolierten Lage mit Hilfe der C^{α}- und C^{β}-Verschiebung direkt identifiziert werden [55].*

	A	G	S	T	V
C^{α}	> 49 ppm	< 49 ppm	≥ 53 ppm	≥ 61 ppm	≥ 61 ppm
C^{β}	≤ 22.5 ppm	-	> 61.5 ppm	> 64 ppm	≥ 30 ppm, ≤ 35ppm
C^{α} - C^{β}	-	-	< 0	< 0	

Tabelle 7: *Von PASTA Toolkit verwendete Verschiebungsgrenzen für die Erkennung der Ankeraminosäuren A, G, S, T und V.*

Nach der Erkennung der Ankeraminosäuren erfolgt die Bestimmung der restlichen Aminosäuren. Eine genaue Bestimmung einzelner Aminosäuren ist beim zweiten Schritt der Aminosäureerkennung nicht möglich. Vielmehr kann die Anzahl der Zuordnungen auf ein überschaubares Maß eingeschränkt werden. Die Auswahl der Zuordnungen erfolgt durch den Vergleich mit literaturbekannten Verschiebungslisten [55] (s. Anhang). Für jede der neunzehn möglichen Aminosäuren (P wird nicht berücksichtigt) wird $|\Delta C^{\alpha}|+|\Delta C^{\beta}|$, die Abweichung zwischen theoretischer und experimenteller Verschiebung, berechnet. Die fünf Kandidaten mit der geringsten Abweichungen werden als potentielle Zuordnungen registriert und anschließend sortiert in die Pseudo-Restliste eingetragen (*Top5*).

Da alle chemischen Verschiebungen sekundärstrukturabhängig sind, können drei verschiedene Listen für den Vergleich eingesetzt werden: *random coil*, α-Helix und β-Faltblatt. Üblicherweise wird die Sekundärstruktur erst nach der Rückgratzuordnung mit Hilfe des CSI (*chemical shift index*) [46, 90] bestimmt. Es lassen sich jedoch auch vor der Rückgratzuordnung schon Angaben über die Sekundärstruktur eines Restes machen: beispielsweise mittels Kopplungskonstanten ($^3J(H^N,H^{\alpha})$), chemischer Verschiebung (H^{α}) oder Sekundärstrukturvorhersage aus der Aminosäuresequenz [91, 93, 94].

Ein Einsatz sekundärstrukturtypischer Verschiebungslisten lohnt sich jedoch nur, falls die genaue Verteilung der Sekundärstrukturelemente bekannt ist. Kann die Topologie nur grob als überwiegend α-helikal oder β-Faltblatt abgeschätzt werden, z.B. über ^{13}C-HSQC-Daten, zeigte sich in zwei von drei Testfällen eine Verschlechterung des Gesamtergebnisses bei Verwendung der entsprechenden Verschiebungsliste (s.u.). Allgemein empfiehlt sich daher meistens die Verwendung von *random coil*-Daten.

Das Verhalten des zweiten Schritts der Aminosäureerkennung wurde an fünf Testdatensätzen überprüft. Als Testdatensätze dienten die Zuordnungslisten der Proteine NusB (129 AS) [118], Parvulin (96 AS) [119], RiSy (92 AS) [120], HNGAL (179 AS) [121] und VAT-N (183 AS) [122]. Abbildung 24 zeigt die Ergebnisse dieser Untersuchungen.

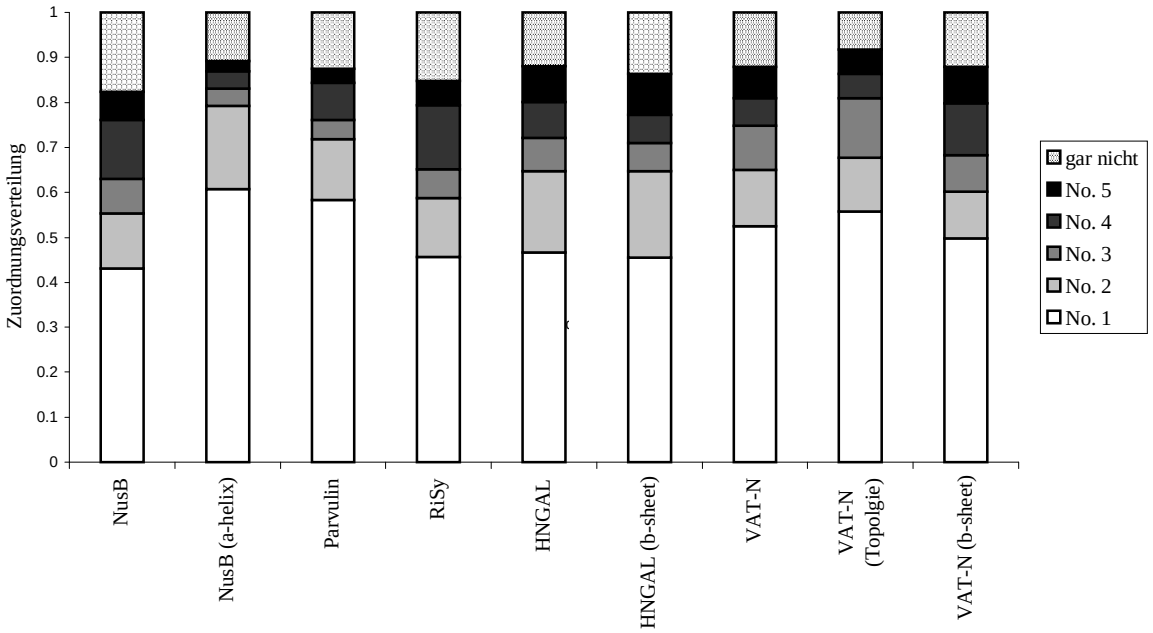

Abbildung 24: *Verteilung der korrekt erkannten Aminosäuren auf die* Top5 *der Erkennungsroutine. Die Rechnungen wurden anhand der Zuordnungslisten von fünf Proteinen (Nusb, 129 AS; Parvulin, 96 AS; RiSy, 92 AS; HNGAL, 179 AS; VAT-N, 183 AS) durchgeführt.*

Für jeden der fünf Datensätze wurde eine Rechnung zum Vergleich mit der *random coil* Liste durchgeführt. Für NusB, HNGAL und VAT-N wurden aufgrund ihrer Strukturbesonderheiten zusätzlich sekundärstrukturspezifische Rechnungen durchgeführt.

Bei Verwendung der *random coil* Liste ergibt sich für die verschiedenen Testrechnungen eine Fehlerquote zwischen 8% und 18%. Als Fehler werden alle Reste gewertet, für die sich die richtige Zuordnung nicht unter den *Top5* der Aminosäureerkennung befindet. Diese Fehlerquoten wurden einschließlich der Reste mit unvollständiger C^{α}- und C^{β}-Zuordnung erstellt, da bei einer manuellen Bewertung normalerweise auch unvollständige Reste berücksichtigt werden. Die korrigierten Fehlerquoten, unter ausschließlicher Berücksichtigung von Resten mit vollem Verschiebungssatz, liegen zwischen 3% und 14%. Dabei weisen nur die beiden Datensätze von HNGAL und VAT-N Werte größer 10% auf. Beide verfügen über Topologien mit hohem β-Faltblatt-Anteil. Setzt man für HNGAL und VAT-N anstatt der *random coil* Liste die β-Faltblatt-Liste zum Verschiebungsvergleich ein, verbessert sich das Ergebnis nur für die Reste, die tatsächlich in einer β-Faltblattstruktur angeordnet sind. Die Gesamtqualität des Ergebnisses nimmt im Vergleich jedoch sogar ab. Erwartungsgemäß treten

die meisten Fehler in den helikalen Bereichen der beiden Proteine auf, da α-Helix und β-Faltblatt gegenläufige Sekundärstruktureinflüsse auf die C^{α}- und C^{β}-Verschiebungen zeigen. Anders verhält sich das Ergebnis, wenn genaue Angaben über die Sekundärstruktur gemacht werden. Gibt man für das Protein VAT-N selektiv das Sekundärstrukturelement zu jedem Rest an („VAT-N Topologie“ in Abbildung 24), verbessert sich das Ergebnis etwas. Die Anzahl der fehlerhaft erkannten Reste sinkt von 12% auf 8%.

Für das Protein NusB zeigt sich ein anderes Verhalten. Durch Einsatz der α-Helix-Vergleichsliste lässt sich eine deutliche Verbesserung des Gesamtergebnisses erzielen: Die Anzahl der nichterkannten Reste sinkt um 7%, die Anzahl der Reste, für die die richtige Zuordnung an erster Stelle steht, steigt um 17%. NusB enthält keinerlei β-Faltblattstrukturen, so dass sich im Gegensatz zu HNGAL und VAT-N für keinen Bereich der Zuordnungsliste größere Abweichungen von den Verschiebungslisten ergeben. Vergleicht man die Verschiebungen der einzelnen Reste von NusB mit den *random coil* Werte für die entsprechenden Aminosäuren, zeigt sich sogar für mehr als 70% der Reste eine Tendenz zu den α-helikalen Werten.

Im Durchschnitt befindet sich die richtige Aminosäure in ca. 50% aller Fälle an erster Stelle in den *Top5*. Betrachtet man die erste und zweite Position der *Top5* gemeinsam, treten starke Schwankungen in der Lösungsqualität auf. Die richtige Lösung wird hier in 60% - 80% der Fälle erreicht. Um eine garantierte Erkennungsrate von ca. 90% zu erreichen, ist mindestens die Auswahl der fünf nächsten Aminosäuren notwendig. Damit lässt sich die Auswahl der möglichen Aminosäuren zumindest auf ein Viertel der ursprünglichen Zahl einschränken. Dieser Wert ist ausreichend für eine sinnvolle Ergänzung des Ergebnisses aus der Erkennung der Ankeraminosäuren während des *mapping*-Prozesses.

4.2.6 Abbilden der Daten auf die Aminosäuresequenz

Der letzte Schritt der Rückgratzuordnung des Proteins ist das *mapping*, d.h. das Abbilden der Pseudo-Reste auf die Sequenz. Beim *mapping* werden alle Informationen aus den vorhergehenden Zuordnungsschritten genutzt, um möglichst jedem Pseudo-Rest eine Aminosäure der Sequenz eindeutig zuweisen zu können. Da jedoch nicht gewährleistet ist,

dass zu allen Sequenzpositionen überhaupt ein passender Pseudo-Rest in der Zuordnungsliste existiert bzw. auch falsche Pseudo-Reste in der Zuordnungsliste enthalten sein können, ist eine eindeutige Lösung nicht immer möglich.

Aus diesem Grund wird die Funktion *f(x,y)* zur Bewertung der Reste eingeführt. Die Variable x entspricht hierbei der Sequenzposition, die Variable y dem betrachteten Pseudo-Rest. *f(x,y)* gibt ein Wahrscheinlichkeitsmaß für die Zuordnung von Rest y auf Sequenzposition x an.

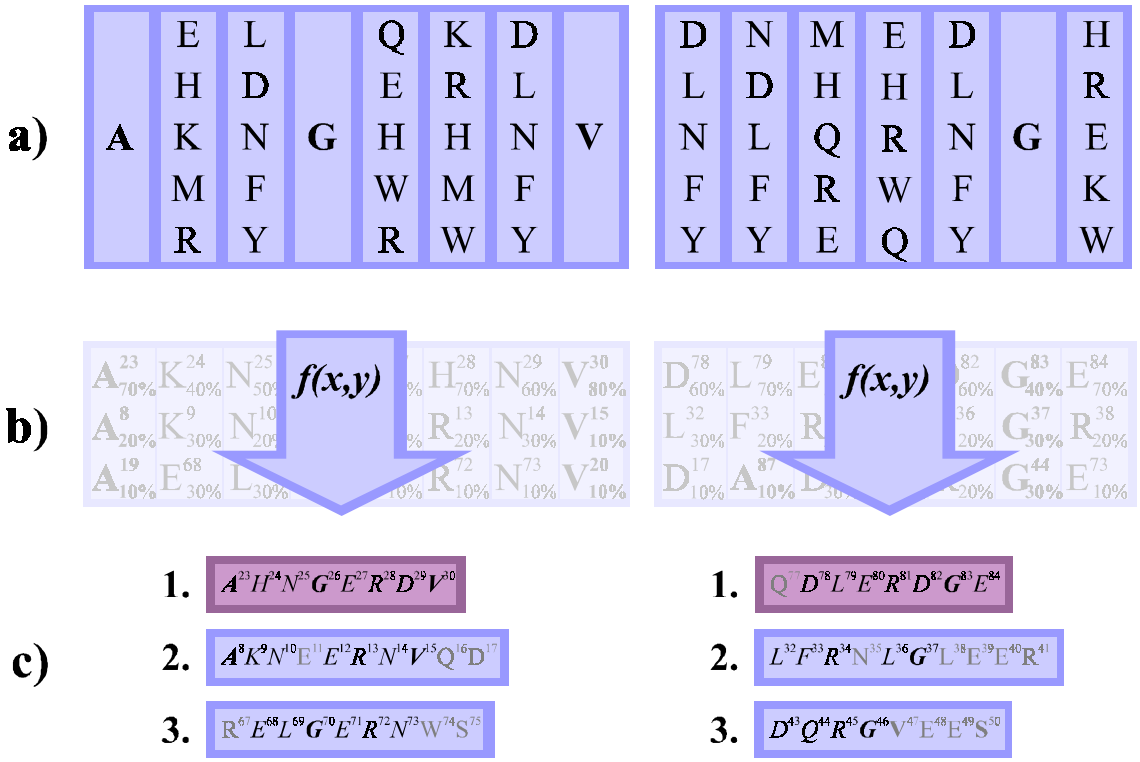

Abbildung 25: *Schematische Darstellung des Sequenz-Mappings im PASTA Toolkit: a) Fragmente aus vorhergehenden Zuordnungsschritten; b) Bewertung der Fragmente für alle Sequenzpositionen; c) Extraktion der wahrscheinlichsten Zuordnungen.*

Abbildung 25 zeigt eine schematische Darstellung des *mapping* im PASTA Toolkit. Die einzelnen Teilschritte werden im Folgenden erklärt.

Zu Beginn des *mapping* werden die Funktionswerte aller möglichen Wertepaare (x,y) berechnet (s.u.). Anschließend wird für jedes aus dem Matrixvergleich erhaltene Fragment (s. 4.2.4) nach einem *best first* Ansatz die optimale Sequenzzuordnung bestimmt. Das bedeutet, für jede Sequenzposition wird die Summe der einzelnen *f(x,y)* des Fragments berechnet:

$$\sum_{x=x_{Anfang},y=y_{Anfang}}^{x+(l-1),y+(l-1)} f(x,y)$$

l = Fragmentlänge

Als optimale Zuordnung wird die Sequenzposition angenommen, für die diese Summe maximal ist. Die Fragmente werden in der Reihenfolge ihrer Größe abgearbeitet. Für die kürzeren Bruchstücke stehen damit weniger Zuordnungspositionen als zu Beginn der Rechnung zur Verfügung, so dass diese trotz weniger Anhaltspunkte an die richtige Stelle der Sequenz abgebildet werden können.
Mit dieser Methode können auch unterbestimmte Reste, die bei einer Einzelbetrachtung nicht zuzuordnen wären, auf die Sequenz abgebildet werden, falls sie zu einem zusammenhängenden Fragment der Pseudo-Restliste gehören.
Die Funktion *f(x,y)* definiert sich wie folgt:

$$f(x,y) = f_{Top5}(x,y) + f_{Anker}(x,y) + f_{Nachbar}(x,y) + f_{Fragment}(x,y)$$

x = Position in der Aminosäuresequenz

y = Pseudo-Rest

Handelt es sich bei der Aminosäure an Sequenzposition x um Prolin, wird *f(x,y)* grundsätzlich der maximale Negativwert –9999 zugeordnet, da Prolin aufgrund des fehlenden Amidprotons in $^{15}N/H^N$-basierten Experimenten nicht direkt zu beobachten ist. Dies bedeutet, dass keine Prolin-Pseudo-Reste in der Zuordnungsliste enthalten sein sollten. Die Zuordnung eines Pseudo-Restes auf eine Prolin-Position muss also zwangsläufig fehlerhaft sein.
Die Einzelfunktionen von *f(x,y)* haben folgende Bedeutungen:

- $f_{Top5}(x,y)$:
  Diese Funktion bewertet die Übereinstimmung des Aminosäuretyps der Sequenzposition mit den *Top5* Aminosäuren aus der Aminosäurebestimmung (s. 4.2.5). Befindet sich die korrekte Aminosäure unter den *Top5*, wird eine positive Bewertung vergeben, andernfalls eine negative Bewertung. Da die Bestimmung der sogenannten Ankeraminosäuren A,G,S,T und V wesentlich zuverlässiger verläuft als bei den restlichen 15 Aminosäuren, wird bei einer negativen Beurteilung zwischen

zwei Fällen unterschieden. Die einzelnen Bewertungen sind in Tabelle 8 angegeben. Die Position der richtigen Aminosäure innerhalb der *Top5* wird nicht bewertet, da diese nach den Ergebnissen von Testläufen der Aminosäureerkennung (s. 4.2.5, Abbildung 2) nur eine begrenzte Aussagekraft besitzt.

	Übereinstimmung	Fehler
A,G,S,T,V	+10	-50
C,D,E,F,H,I,K,L,M,N,R,S,T,V,W,Y	+10	-10

Tabelle 8: *Pseudoenergiewerte für* $f_{Top5}(x,y)$.

- $f_{Anker}(x,y)$:

Aufgrund der besonderen Bedeutung der Ankeraminosäuren werden sie ein zweites Mal gesondert bei der Bewertung berücksichtigt. Dies liefert folgende Werte für $f_{Anker}(x,y)$:

Übereinstimmung: +50
Fehler: -50
Ausnahmen:
S und T vertauscht: +10
V und I vertauscht: -10

Bei der Bewertung werden zwei aminosäurespezifische Spezialfälle unterschieden:

- Die Ähnlichkeit der C^{α} bzw. C^{β} chemischen Verschiebungen von Serin und Threonin können leicht zur Verwechslung der beiden führen. Da diese beiden Aminosäuren jedoch ansonsten eindeutig von den restlichen unterscheidbar sind, wird bei einer Vertauschung von S und T trotzdem ein positiver Wert für $f_{Anker}(x,y)$ zurückgegeben. Dieser Wert ist allerdings geringer als bei einer tatsächlichen Übereinstimmung von S bzw. T und entspricht dem positiven Wert von $f_{Top5}(x,y)$.

- Auch Valin kann aufgrund seiner C^{α} und C^{β} chemischen Verschiebungen verwechselt werden. Ist Valin mit der Aminosäure Isoleucin vertauscht, erhält $f_{Anker}(x,y)$ einen kleineren Strafwert als bei einer Verwechslung mit anderen Aminosäuren. Der Wert entspricht dem Strafwert für Nicht-Ankeraminosäuren aus $f_{Top5}(x,y)$.

- $f_{Nachbar}(x,y)$:

  Zur Berechnung dieser Energie werden zu jeder Position noch die vorhergehende und die nachfolgende Position einbezogen. Das bedeutet, Tripel von aufeinanderfolgenden Pseudo-Resten werden mit Aminosäuretripeln der Sequenz verglichen.

$$f_{Nachbar}(x,y) = \frac{\sum_{x-1,y-1}^{x+1,y+1} f_{Top5}(x,y) + f_{Anker}(x,y)}{3}$$

  Liegt das Reste-Tripel am Rande eines Fragments erhält es nur dann eine positive Bewertung, wenn das Aminosäuretripel der Sequenz von einem Prolin abgeschlossen wird. Dies geschieht unter der Annahme, dass der Optimierungsalgorithmus grundsätzlich alle theoretisch möglichen Fragmente findet und diese jeweils die optimale Länge besitzen.

- $f_{Fragment}(x,y)$:

  Jeder Rest erhält einen Gewichtungsterm entsprechend des Fragments, dem er angehört. Der Wert ergibt sich durch den Vergleich des Musters an Ankerpunkten des Fragments mit der Sequenz. Dieser Term entspricht damit in etwa dem *mapping*-Konzept von PASTAV3.0 (s. 4.1.2.2).

$$f_{Fragment}(x,y) = \frac{\sum_{x=x_{Anfang},y=y_{Anfang}}^{x+(l-1),y+(l-1)} f_{Anker}(x,y)}{l}$$

  l = Fragmentlänge

Zwei Spezialfälle werden zusätzlich berücksichtigt:

- Befindet sich ein Prolin an einer Position innerhalb des Fragments, wird für den entsprechenden Rest der fünffache Wert für $f_{Anker}(x,y)$ verwendet, (also $5 \cdot (-50)$). Aufgrund der ähnlich Verschiebungen von V und P ist eine Überlagerung eines V- und P-Restes theoretisch denkbar. Eine Zuordnung eines Fragments auf ein Sequenzstück, das Prolin beinhaltet, ist jedoch immer noch äußerst unwahrscheinlich.
- Ist ein Fragment länger als die in der Sequenz freien, zusammenhängenden Zuordnungspositionen, kann das gesamte Fragment nicht beurteilt werden. Der Rückgabewert für $f_{Fragment}(x,y)$ wird gleich 0 gesetzt.

Die Routine von PASTA Toolkit benötigt im Gegensatz zu dem für PASTAV3.0 entwickelten Algorithmus (s. 4.1.2.2) keine feste Mindestanzahl an sicher zugeordneten Ankerpunkten, um die korrekte Sequenzposition für ein Fragment zu finden. Auch für Fragmente, in denen keine Ankeraminosäure vorhanden sind, kann eine Aussage über die Zuordnung getroffen werden. Es werden lediglich ausreichend viele Reste mit einer vollständige Zuordnung der C^{α}- und C^{β}-Verschiebung für die Beurteilung eines Fragmentes benötigt.
Die beiden Parameter von PASTAV3.0, die Mindestanzahl von Ankerpunkten pro Fragment und die Anzahl erlaubter Fehler, sind nicht mehr erforderlich, da das *mapping* nun über Zuordnungswahrscheinlichkeiten bestimmt wird. Sind einzelne Reste eines Fragments verkehrt, üben diese nur einen geringen Einfluss auf die Zuordnungswahrscheinlichkeit des ganzen Fragments aus. Die gesamte Zuordnung bleibt in den meisten Fällen trotzdem unverändert.
Die Routine gibt optional für jeden Rest die Einzelbewertungen der drei wahrscheinlichsten Zuordnungen aus. Damit können zweifelhafte Fragmentzuordnungen leicht überprüft und verbessert werden. Ebenso ist es möglich, Reste ohne Fragmentzugehörigkeit zu beurteilen.
Nachfolgend sind die Ergebnisse des Matrixvergleichs für die zwei Testdatensätze NusB (139 Aminosäuren, 129 Pseudo-Reste) und VAT-N (185 Aminosäuren, 216 Pseudo-Reste) angegeben. Insbesondere wird der Unterschied des für PASTA V3.0 entwickelten *mapping*-Konzepts mit dem erweiterten Algorithmus von PASTA Toolkit verglichen.

NusB (139 AS)	PASTA Toolkit	PASTA V3.0
Gesamtanzahl der Reste	129	129
Abgebildete Reste	100	86
Korrekt abgebildete Reste	92	86
Falsche Reste	8	0
Abgebildete Bruchstücke	12	8
Korrekt abgebildete Bruchstücke	9	8
Falsche Bruchstücke	3	0

Tabelle 9: *Ergebnisse der Sequenzabbildung von PASTA Toolkit im Vergleich zu PASTA V3.0. Datensatz: NusB, 139 AS, 129 Reste im Datensatz.*

VAT-N (185 AS)	PASTA Toolkit	PASTA V3.0
Gesamtanzahl der Reste	216	216
Abgebildete Reste	130	64
Korrekt abgebildete Reste	97	60
Falsche Reste	33	4
Abgebildete Bruchstücke	24 (13 davon mit $l \leq 4$)	8
Korrekt abgebildete Bruchstücke	16 (13 davon mit $l \leq 4$)	7
Falsche Bruchstücke	8 (6 davon mit $l \leq 4$)	1

Tabelle 10: *Ergebnisse der Sequenzabbildung von PASTA Toolkit im Vergleich zu PASTA V3.0. Datensatz: VAT-N, 185 AS, 216 Reste im Datensatz.*

Beide Datensätze unterscheiden sich in der Qualität der Daten (s. 4.2.4). Während der Datensatz für NusB nur verifizierte Reste enthält, die tatsächlich für die Zuordnung relevant sind, befinden sich im Datensatz für VAT-N noch 40 überschüssige Reste, die nicht ohne Weiteres von den tatsächlichen Resten des Proteins unterschieden werden können. Zudem stehen bei NusB durchschnittlich 5 oder mehr Ankerpunkte pro Zuordnungsfragment zur Verfügung, für VAT-N dagegen nur 3 Ankerpunkte. Dementsprechend sind die Ergebnisse für die beiden Datensätze sehr unterschiedlich.

Ein Vergleich der Ergebnisse der beiden Routinen von PASTA Toolkit und PASTA V3.0 für NusB liefert nahezu identische Resultate. Die Anzahl der zugeordneten Reste ist bei PASTA

Toolkit etwas höher (100 Reste gegenüber 86 Resten), die Zahl der korrekt abgebildeten Reste verändert sich aber kaum. Die größere Fehlerzahl der PASTA Toolkit Routine ist auf die höhere Anzahl abgebildeter Fragmente zurückzuführen. Die Fehler treten an Stellen auf, an denen sich Reste mit außergewöhnlichen oder fehlenden chemischen Verschiebungen befinden, die schon von der Aminosäureerkennung falsch bewertet wurden. PASTA V3.0 versucht nicht diese Fragmente zuzuordnen, weil sie keine Ankeraminosäuren enthalten. Da der NusB-Datensatz in den meisten Fragmenten jedoch überdurchschnittlich viele Ankeraminosäuren (≥ 5) aufweist, wirkt sich die Unabhängigkeit von Ankeraminosäuren bei PASTA Toolkit nur gering aus. Mit PASTA V3.0 lassen sich in diesem Fall in etwa die gleichen Ergebnisse erzielen.

Für den Datensatz von VAT-N zeigen sich deutliche Unterschiede zwischen den beiden Routinen. Die Zahl der zugeordneten Reste steigt von 64 auf 130, die Zahl der korrekt abgebildeten Reste von 60 auf 97. Die hohe Fehlerquote ergibt sich aus der Zuordnung vieler kleiner Fragmente (≤ 4 Reste): 13 der 24 zugeordneten Fragmente fallen in diese Kategorie. PASTA V3.0 ist nicht in der Lage, diese Fragmente abzubilden, da keines mehr als einen Ankerpunkt enthält. PASTA Toolkit dagegen kann 7 dieser 13 Fragmente korrekt auf die Sequenz abzubilden. Betrachtet man nicht das Abbilden ganzer Fragmente, sondern die Einzelbewertungen der Sequenzposition für jeden Rest, dann ist die Erfolgsquote von PASTA Toolkit noch wesentlich höher: Für 83% der Reste wird die korrekte Sequenzposition als wahrscheinlichste Zuordnung angegeben. Für 12% befindet sich die korrekte Lösung unter den ersten drei Möglichkeiten (diese können vom Programm ausgegeben werden, s.o.). Lediglich für 5% der Reste liegt die Bewertung der korrekten Zuordnung nicht unter den drei vorgeschlagenen Positionen.

Die *mapping*-Routine von PASTA Toolkit eignet sich damit insbesondere für Datensätze, die eine Reihe von unsicheren Positionen enthalten. Die vorhergehenden Zuordnungsschritte liefern in diesem Fall eine Reihe kleiner Fragmente, die sich nicht allein über Ankerpunkte zuordnen lassen. Auch für Reste, die keinem Fragment angehören, kann eine Bewertung erstellt werden.

Abbildung 26 zeigt die grafische Benutzeroberfläche des *mapping*-Moduls. Die Ergebnisse des Matrixvergleichs können in die Sequenzabbildung mit einbezogen werden (s. 4.2.4). In diesem Fall werden zum endgültigen Abbilden der Reste auf die Sequenz nur die gesicherten Fragmente aus dem Matrixvergleich verwendet. Für alle anderen Reste wird lediglich die Einzelbewertung durchgeführt.

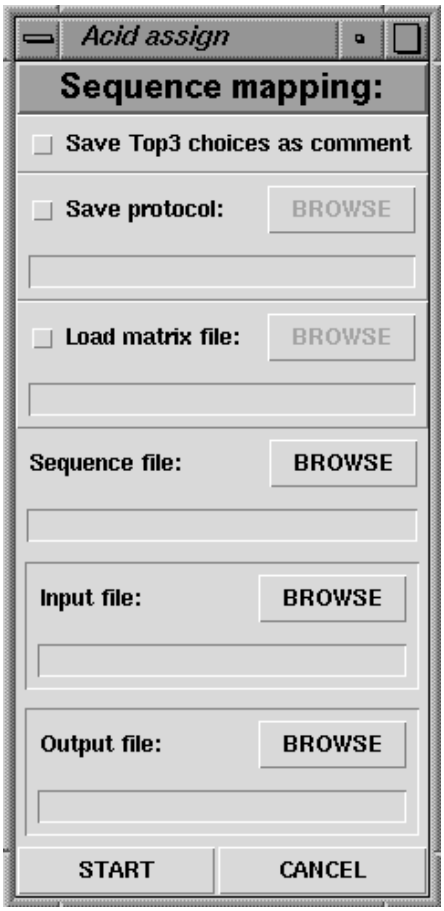

Abbildung 26: *Grafische Benutzeroberfläche des* mapping-*Moduls.*

4.2.7 Datenverwaltung

4.2.7.1 Dateiformate

Ein äußerst wichtiges Element eines Programmpakets ist die Datenverwaltung. Bei der Entwicklung eines Datenformates müssen zuerst die Anforderungen aller vorgesehenen Einsatzbereiche definiert werden.

Da es sich bei PASTA Toolkit um ein Paket unabhängiger Einzelmodule handelt, soll der Datenaustausch zwischen den Modulen über möglichst wenige Dateien erfolgen. Das heißt, es muss ein zentrales Format entwickelt werden, das alle Informationen in leicht abrufbarer Form enthält und beliebig erweiterbar für die Entwicklung neuer Datentypen ist, ohne ältere Programmfunktionen zu beeinflussen. Darüber hinaus soll das Format für den Benutzer auch ohne den Einsatz von PASTA Toolkit lesbar und verständlich sein. Dieses Datenformat muss gegebenenfalls auch als Schnittstelle zu anderen Programmen dienen und sollte daher mit einfachen Mitteln in deren Eingabeformat umzuformatieren sein.

Für solche Aufgaben bietet sich das Konzept der relationalen Datenbanken an [123]. Datenbanken dienen zur Speicherung und Verwaltung von Daten, die persistent sind, d.h. die über die Laufzeit eines Programms hinaus aufbewahrt werden. Die Daten werden so aufbereitet, dass jedes Element über einen sogenannten Schlüssel eindeutig identifizierbar ist. Soll später eine bestimmte Information wieder aus der Datenbank abgerufen werden, können mit diesem Schlüssel einfache Anfragen an die Datenbank gestellt werden, die das passende Element zurückgeben.

Als zentrale Datei wurde die Pseudo-Rest-Liste entwickelt, die zur Verwaltung der gesamten Daten eines Zuordnungsprojektes dient. Ein Pseudo-Rest enthält alle bekannten Daten einer einzelnen Aminosäure. Der Datenaustausch zwischen den verschiedenen Modulen des PASTA Toolkit erfolgt mit Hilfe der Pseudo-Restliste.

Für die Entwicklung des Formates der Pseudo-Rest-Liste wurde das Datenbankkonzept entsprechend der oben genannten Anforderungen modifiziert und angepasst. Dabei wurden weitgehend die Kriterien für eine Datenbank in 3. Normalform eingehalten (s.u.). Um die direkte Lesbarkeit der Dateien ohne Verwendung von PASTA Toolkit zu verbessern, enthält das Format jedoch verschiedene redundante Informationen. So werden Aminosäuretypus/Restname und Originalnummer eines Restes mehrmals gespeichert. Die

dadurch entstehende Änderungsanomalie muss bei der internen Datenverwaltung berücksichtigt werden. Unter Änderungsanomalie versteht man allgemein das Problem der Datenaktualisierung von redundanten Daten: Wird die gleiche Information mehrmals gespeichert, können nach einer Änderung der Daten mehrere, inkonsistente Versionen der gleichen Information im Datensatz enthalten sein. Im Fall des PASTA Toolkit bedeutet dies, dass bei einer Änderungen des Restnamens nicht nur eine Position in der Datenbank aktualisiert werden muss, sondern jedes einzelne SHIFT-Element (s. Anhang Datentypen) des entsprechenden Restes. Ansonsten sind jedoch alle Kriterien der 3. Normalform für eine relationale Datenbank erfüllt:

- Jedes Attribut ist elementar, d.h. es besitzt keine Mengen oder Reihen als Werte.
- Jedes Nichtschlüsselattribut ist von jedem Schlüssel voll funktional abhängig.
- Jedes Nichtschlüsselattribut ist außer von sich selbst nur von jedem Schlüssel voll funktional abhängig.

Neben der Pseudo-Rest-Liste kann zum Datenaustausch zwischen dem Matrixvergleichs-Modul (s. 4.2.4) und dem Mapping-Modul (s. 4.2.6) noch eine weitere Datei verwendet werden. Diese Datei wird jedoch lediglich optional zur Speicherung von Zwischenergebnissen benutzt. Sie dient nicht der Datenverwaltung.
Alle PASTA-Toolkit-Dateien werden im ASCII-Format gespeichert. Dieses ist mit jedem herkömmlichen Texteditor lesbar und kann mit einer Reihe von Skriptsprachen bearbeitet werden.

4.2.7.2 Interne Verwaltung

Für die interne Verwaltung werden die Daten in verketteten Listen gespeichert. Man unterscheidet zwischen einfach und doppelt verketteten Listen. In einer einfach verketteten Liste enthält jedes Element nur einen Zeiger auf seinen Nachfolger (Abbildung 27). So können die Einzelelementen zu einer Kette beliebiger Länge verknüpft werden. Um auf ein bestimmtes Element der Kette zuzugreifen, muss die Kette vom Beginn Element für Element

durchwandert werden, bis der gewünschte Punkt erreicht ist. In einer doppelt verketteten Liste existiert zu jedem Kettenglied außerdem noch ein Zeiger auf seinen Vorgänger (Abbildung 28). Dies bietet den Vorteil, dass die Liste in beiden Richtungen durchwandert werden kann. Dadurch kann die Suche nach einem bestimmten Element der Liste wesentlich beschleunigt werden.

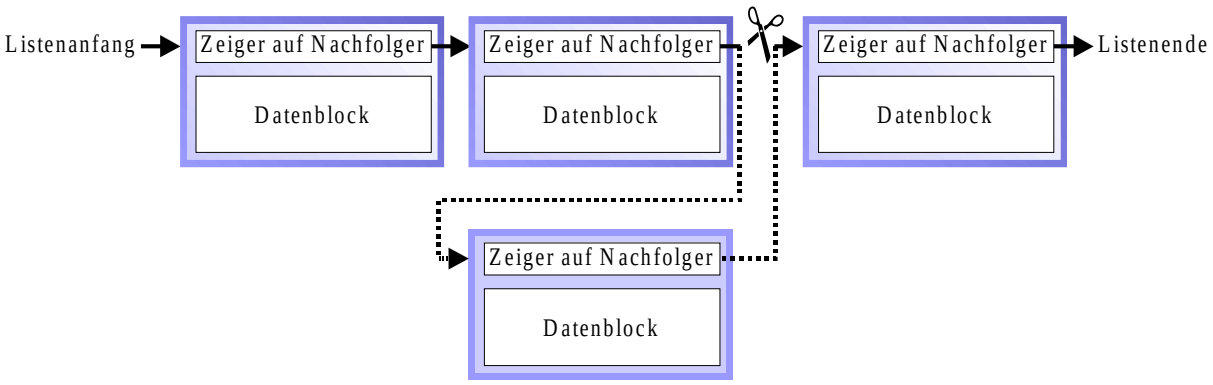

Abbildung 27: *Schematische Darstellung einer einfach verketteten Liste. Jedes Element enthält einen Zeiger auf seinen Nachfolger. Soll ein neues Element in die Liste eingefügt werden, müssen nur die Zeiger entsprechend aktualisiert werden.*

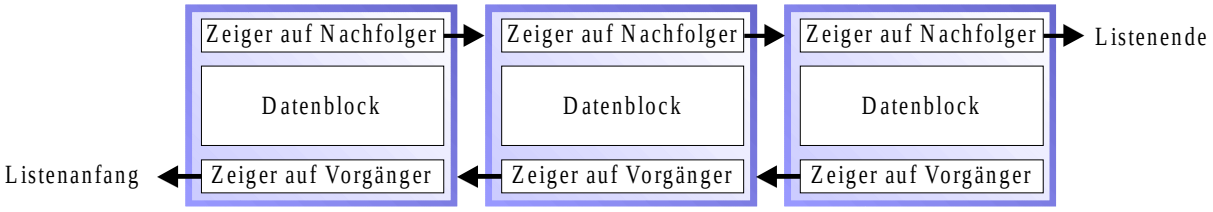

Abbildung 28: *Schematische Darstellung einer doppelt verketteten Liste. Jedes Element kann über seine beide Nachbarn erreicht werden.*

Im Vergleich zu herkömmlichen Feldern bieten verkettete Listen die Möglichkeit einer dynamischen Speicherverwaltung. Das heißt, ihre Größe kann während des Programmablaufs modifiziert werden. Soll ein neues Element in die Liste eingefügt werden, müssen lediglich die Zeiger aller betroffenen Elemente aktualisiert werden (Abbildung 27). Daher eignen sich verkettete Listen grundsätzlich zur Verwaltung der Zuordnungsdaten nach Datenbankprinzipien. Auch eine einfache Erweiterung für zukünftige Neuerungen im Listenformat ist problemlos möglich.

Für das Optimierungsmodul (s. 4.2.3) wird die verkettete Liste, die die Pseudo-Restdaten enthält, zu einem Ring geschlossen. Diese Ringanordnung vereinfacht den Modifikationsschritt des *fast TA*-Algorithmus, da beim Austausch der Pseudo-Reste keine Randpositionen berücksichtigt werden müssen. Der verringerte Verwaltungsaufwand führt zu einer Steigerung der Rechengeschwindigkeit.
Die Implementierung des Programms erfolgte in der Programmiersprache C. Dazu wurden verschiedene Datentypen entwickelt. Eine Beschreibung der einzelnen Datentypen findet sich im Anhang unter 9.3 Datentypen.

4.2.8 Grafische Benutzeroberfläche

Die grafische Benutzeroberfläche von PASTA Toolkit wurde mit der Skriptsprache Tcl/Tk [114] entwickelt. Diese stellt ein leistungsfähiges Set von Befehlen zur Verfügung, die auf einfache Weise das Erstellen grafischer Objekte für eine Benutzeroberfläche ermöglichen. Da es sich bei Tcl/Tk um eine Skriptsprache handelt, ist der erstellte Code plattformunabhängig. Ein entsprechender Interpreter existiert für alle gängigen Unix-Dialekte, Linux und Windows-NT.
Die grafische Benutzeroberfläche von PASTA Toolkit erfüllt drei Aufgaben:

- Ansicht und Manipulation der Zuordnungsdaten
- Erstellung der ASCII-Eingabedateien für die einzelnen Toolkit-Module
- Integration der einzelnen Module zu einem Gesamtpaket

Alle Bereiche werden über das Hauptmenü der Oberfläche (Abbildung 29) miteinander verbunden.

Abbildung 29: *Hauptmenü der grafischen Benutzeroberfläche*

Zur Ansicht und Manipulation der Daten wurde eine Schnittstelle zum C-Code des Programmpakets geschaffen. Dazu wurden in C eine Reihe spezieller Befehle zur Kommunikation mit der Benutzeroberfläche definiert. Für die Darstellung der Pseudo-Restliste wurde ein *spreadsheet*-Ansatz gewählt (Abbildung 30), der dem Anwender eine selektive Konfiguration der dargestellten Informationen ermöglicht. Alle implementierten Sortier- und Suchfunktionen verwenden das *spreadsheet* als Basis.

Die Parameter der Oberfläche können über ASCII-Parameter-Dateien eingestellt werden (s. Anhang).

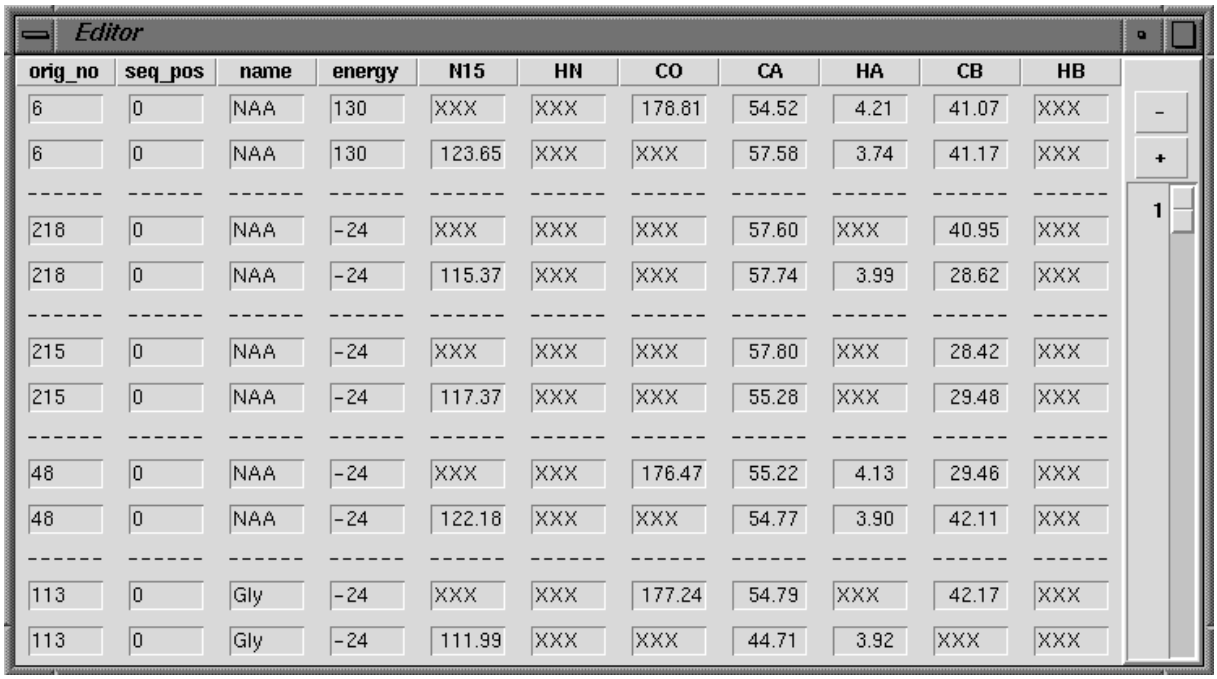

Abbildung 30: *Datenspreadsheet der grafischen Benutzeroberfläche*

Die einzelnen Zuordnungsmodule sind unabhängig von der Benutzeroberfläche gestaltet. Sie können in der Form einfacher Shell-Befehle aufgerufen werden und benötigen als Eingabe ausschließlich ein Parameter-File im ASCII-Format (s. Anhang). Auf diese Weise lässt sich der Satz an durch die Benutzeroberfläche verwalteten Modulen jederzeit erweitern, ohne dabei auf das C-Programm zurückgreifen zu müssen. Zur Integration eines neuen Moduls ist es lediglich notwendig, alle neuen Dateien im Hauptteil des Oberflächenskripts zu registrieren. Die Aufnahme neuer Funktionen in das Programmpaket ist damit auch ohne genaue Kenntnisse des Quellcode möglich.

5 Zuordnung und Strukturbestimmung der N-terminalen Domäne von VAT

5.1 Biochemischer Hintergrund

VAT (***V****CP ähnliche* ***A****TPase aus* ***T****hermoplasma*) ist ein Protein der AAA-Familie (***A****TPase* ***a****ssociated with a variety of cellular* ***a****ctivities*). Die AAA-Proteine zeichnen sich durch eine 220-250 Aminosäuren lange konservierte Domäne, die AAA-Domäne, aus [124]. Die AAA-Domäne enthält zwei typische Strukturmotive der A/GTPase-Superfamilie, das *Walker-box-A*- und das *Walker-box-B*-Motiv. Die AAA-Domäne kann in AAA-Proteinen ein- oder zweimal enthalten sein. Man unterscheidet die AAA-Proteine demnach in Typ-I- und Typ-II-Proteine [125].

Allen AAA-Proteinen ist die Mg^{2+}-abhängige ATPase-Funktion gemeinsam, die von der AAA-Domäne ausgeht. ATPasen wurden zuerst in Bakterien beobachtet, sie spielen aber auch in eukariotischen Zellen eine zentrale Rolle: ATPasen setzen die Energie aus der ATP-Hydrolyse für vielfältige biologische Funktionen um, wie beispielsweise den Membrantransfer der Na^{+}-K^{+}-Pumpe oder andere Transportreaktionen [2].

Neben der ATPase-Wirkung besitzen die AAA-Proteine noch eine große Anzahl weiterer, bisher nicht vollständig aufgeklärter Funktionen. So reicht die Palette der bekannten Funktionen vom Spindelaufbau während der Meiose-Phase der Zellteilung über intrazellularen Vesikeltransport bis zum Proteinabbau als Metalloprotease. Für einige AAA-Proteine wird darüber hinaus eine *chaperone*-Wirkung während der Protein-Faltung und -Entfaltung vermutet [125]. In der Zelle befinden sich die AAA-Proteine erwartungsgemäß in den unterschiedlichsten Bereichen. Viele AAA-Proteine zeigen eine hohe Tendenz zur Oligomerisierung und liegen in dimeren, trimeren oder hexameren Verbänden vor, so z.B. auch NSF, Cdc48 und VAT.

Tabelle 11 gibt einen Überblick über einige Vertreter der AAA-Familie und deren biologische Funktion.

Proteinname	Typ	Funktion
VAT	II	Entfaltung/Abbau von Proteinkomplexen (noch nicht genau geklärt)
VCP	II	Golgi-Membranfusion
NSF	II	Membranfusion für den intrazellularen Vesikeltransport
CDC48	II	ER-Membranfusion
PEX1	II	Peroxisomale Biogenese
SUG2	I	Proteasomale Untereinheit
VPS14	I	Vakuolen/Endosom-Transport und Adressierung
YTA10	I	Metalloprotease der inneren Membran von Mitochondrien
Meil	I	Spindelaufbau
FtsH	I	Bakterielle Metalloprotease zum Abbau der Transkriptionfaktoren σ^{32} und λ IIC

Tabelle 11: *Einige AAA-Proteine und ihre biologische Funktion [125].*

Das Typ-II-AAA-Protein VAT (745 Aminosäuren) aus *Thermoplasma acidophilum* ist das archaebakterielle Homologe der Cdc48/p97-Proteine [126] [127]. Es bildet wie diese eine hexamere Ringstruktur aus, die entfernt an das bakterielle *chaperone* GroEL erinnert [128]. Die Funktion von VAT ist noch nicht genau geklärt. Es wird jedoch vermutet, dass VAT ebenfalls *chaperone*-Funktionen während der Entfaltung bzw. des Abbaus von Proteinkomplexen übernimmt. Wahrscheinlich ist die in dieser Arbeit untersuchte N-terminale Domäne VAT-N (183 Aminosäuren) hauptsächlich für die *chaperone*-Funktion des Proteins verantwortlich. In biochemischen Untersuchungen wurde nachgewiesen, dass VAT-N Polypeptide binden kann, die Proteinaggregation verhindern und die Rückfaltung geeigneter Substrate (wie z.B. Cyclophilin oder β-Lactamase) katalysieren [129]

5.2 Experimentelles

Die Durchführung des Projektes erfolgte in Zusammenarbeit mit M. Coles (Auswertung, Strukturrechnung), T. Diercks (NMR-Messungen) und A. Gröger (Auswertung). Die Protein-Proben wurde von J. Peters (MPI für Biochemie Martinsried) zur Verfügung gestellt und wie in der Literatur beschrieben aus *Thermoplasma acidophilum* exprimiert und gereinigt [130]. Zur nativen Proteinsequenz von 183 Aminosäuren kamen aus Expressions- und Reinigungsgründen am C-Terminus noch zwei G als *spacer* und sechs H als sogenannter *his-tag* hinzu. Damit verfügte die verwendete Probe über 191 Aminosäuren.
Für die NMR-Messungen wurden zwei Proben verwendet:

- 1,4 mM [U-^{15}N]
  in H_2O (10%D_2O), pH 5,9 – pH 6,0; 40 mM Phosphatpuffer, 80 mM NaCl
- 1,2 mM [U-^{13}C-^{15}N]
  in H_2O (10% D_2O); pH 5,9 – pH 6,0; 80 mM Phosphatpuffer, 200 mM NaCl

Beide Proben enthielten zur Probenkonservierung zusätzlich noch geringe Mengen an NaN_3.
Die Aufnahme der NMR-Spektren erfolgte an einem DMX600- und einem DMX750-Gerät der Firma Bruker. Die Messtemperatur wurde über eine Reihe von ^{15}N-HSQC-Spektren auf einen optimalen Wert von 321 K eingestellt. Genauere Angaben zur Aufnahme der NMR-Spektren sind der Dissertation von T. Diercks zu entnehmen [131].
Die Strukturrechnungen wurden mit dem Programm XPLOR [108] durchgeführt und mit dem Programm PROCHECK [132, 133](www.biochem.ucl.ac.uk/~roman/procheck/procheck.html) überprüft.
Die Ergebnisse wurden veröffentlicht [122], die Zuordnungsdaten sind in der BMRB-Datenbank (www.bmrb.wisc.edu) unter der Nummer 4376 abgelegt. Die Strukturdaten wurden in der PDB-Datenbank (www.rcsb.org/pdb/index.html) unter dem Code 1CZ4 archiviert.

5.3 Zuordnung

5.3.1 Rückgrat

Die Rückgratzuordnung erfolgte mit dem Programm PASTA V3.0 (s. 4.1). Die Daten der Rückgratzuordnung dienten gleichzeitig zur Entwicklung einiger Module von PASTA Toolkit (s. 4.2.3, 4.2.4), die für eine erfolgreiche Bearbeitung des Projektes benötigt wurden.

Die Zuordnung wurde unter Verwendung der chemischen Verschiebungen von C^{α}, C^{β}, H^{α} und H^{β} durchgeführt, wie in den Kapiteln 2 und 3 erläutert. Folgende Experimente wurden ausgewertet: HNCA, HNCACB, CBCA(CO)NH, HN(CA)HA, HN(CACB)HAHB, HBHA(CBCA)CONH und HCACO. Zunächst wurde versucht, ausschließlich mit C^{α}, C^{β} und H^{α} zu arbeiten. Aufgrund der Überlagerungsproblematik (s.u.) konnten auf diese Weise jedoch nur 156 Pseudoreste identifiziert werden. Für die übrigen Reste wurden daher die H^{β}-Verschiebungen bzw. selektive NOE-Kontakte, die sich während der Topologiebestimmung aus den Sekundärstrukturelementen ergaben, zur Auswertung hinzugenommen. Die endgültige Zuordnung umfasst 181 der 185 Aminosäuren (es fehlen M1, E2, S16 und T17).

VAT-N besitzt eine interne Sequenzhomologie von 38% zwischen den beiden Abschnitten G6-R48 sowie K49-T92 (vgl. 5.4, Abbildung 33). Zusammen mit den durch die Molekülgröße bedingten Überlagerungen ergibt sich so eine außergewöhnlich hohe Zahl an mehrdeutigen Signalen. Schon geringe Signaldifferenzen zwischen den einzelnen Spektren führten daher beim Einordnen der Signale in die passenden Pseudoreste bzw. der Identifikation von Artefakten zu erheblichen Problemen. Alle Pseudoreste im Datensatz sowie die automatisch generierten Peaklisten mussten aus diesem Grund manuell überprüft und korrigiert werden. Ebenfalls bedingt durch die große Zahl an überlagerten Verschiebungen ergaben sich während der Optimierungsläufe von PASTA viele nicht-konvergente, d.h. nicht eindeutige Bereiche in den Zuordnungslisten. Diese Bereiche lassen sich manuell nur mit großem Aufwand in den Zuordnungslisten erkennen, da sie unterbestimmte, vollständig überlagerte Pseudoreste enthalten, die während der Zuordnung dennoch mit einer negativen Pseudoenergie bewertet werden (vgl. 4.2.3). Aus diesem Grund wurde während des Projektes der Matrixvergleich (s. 4.2.4) entwickelt. Mit Hilfe des Matrixvergleichs lassen sich alle Bereiche einer Zuordnungsliste, die über eindeutige Verknüpfungen miteinander verbunden sind, ohne

großen Aufwand extrahieren. Die übrigen Pseudoreste der Zuordnungsliste können anschließend selektiv überprüft bzw. mit zusätzlichen Informationen ergänzt werden.

Der ursprüngliche Datensatz zur Rückgratzuordnung von VAT-N enthielt 216 Pseudoreste, bestehend aus C^{α}-, C^{β}- und H^{α}-Verschiebungen. Ca. 80% der Pseudoreste verfügten über den vollständigen Signalsatz, ein Aussortieren der 40 überschüssigen Pseudoreste (für eine volle Zuordnung werden 176 Reste benötigt: 185 Aminosäuren – 9 P) war zu diesem Zeitpunkt der Zuordnung aufgrund unzureichender Informationen nicht möglich. Ein Matrixvergleich der Ergebnisse aus 5 unterschiedlichen Optimierungsläufen ergab für 120 der 216 Pseudoreste eine feste Zuordnung in 12 Fragmenten mit einer Mindestlänge von 4 Resten. Alle kleineren Fragmente wurden ignoriert, da sie im allgemeinen zu wenige Informationen für ein Abbilden auf die Sequenz enthalten.

Aufgrund geringfügiger Signalverschiebungen im HN(CA)HA (bis ca. 0,1 ppm für H^{N} und ca. 0,4 ppm für N) im Vergleich zu HNCA, HNCACB und CBCA(CO)NH war das Einordnen der H^{α}-Verschiebungen in die korrekten Pseudoreste schwierig. Daher wurde ein zweiter Matrixvergleich durchgeführt, bei dem die H^{α}-Verschiebung geringer gewichtet wurde als die Kohlenstoffinformation (vgl. 4.2.4). Der Vergleich wurde mit 10 Optimierungsläufe durchgeführt: 5 dieser Rechnungen verfügten über den vollen Parametersatz C^{α}, C^{β} und H^{α}, die restlichen 5 Rechnungen lediglich über C^{α} und C^{β}. Auf diese Weise ergaben sich 16 weitere Zuordnungen, d.h. 136 fest verknüpfte Reste in 14 Fragmenten (Länge ≥ 4).

Für die fehlenden Pseudoreste wurde der Verschiebungssatz mit den H^{β}-Werten aus dem HN(CACB)HAHB bzw. dem HBHA(CBCA)CONH ergänzt.

Die Prolinreste wurden über die [i-1]-Verschiebungen der sequentiellen Nachfolger bzw. ein HCACO-Spektrum bestimmt. Die endgültige Zuordnung umfasst 181 von 185 Resten.

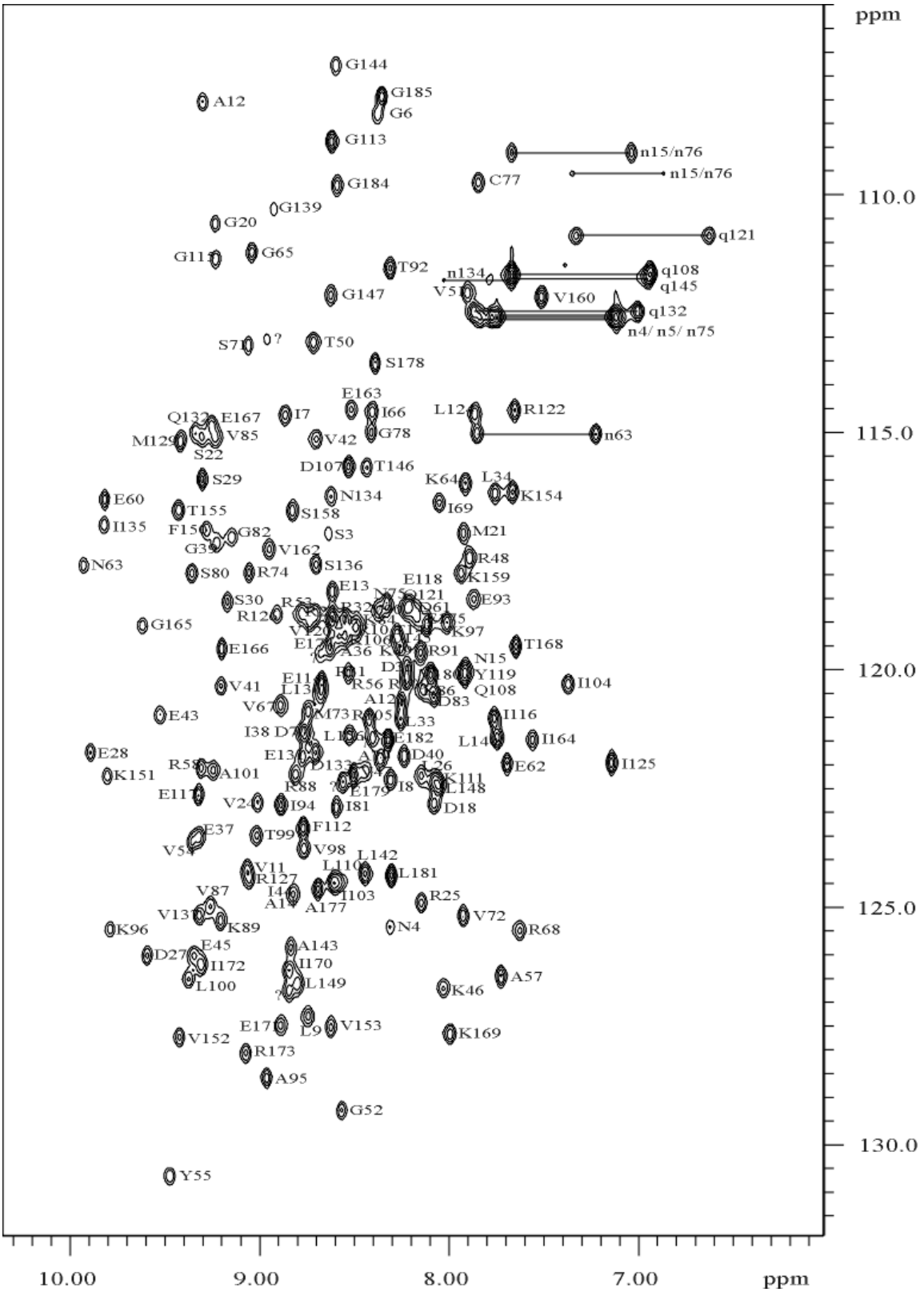

Abbildung 31: *Zugeordnetes ^{15}N-HSQC-Spektrum von VAT-N.*

5.3.2 Seitenketten

Die Zuordnung der Seitenkettensignale erfolgte mittels einer Kombination aus 3D-(H)CCH-TOCSY und 3D-H(C)CH-COSY (vgl. 2.3). Diese beiden Spektren ergänzen sich aufgrund der komplementären ^{13}C- bzw. ^{1}H-Korrelationen zu einem Pseudo-4D-Spektrum. Die größere Dispersion der ^{13}C-Verschiebungen zusammen mit der niedrigeren Signaldichte ermöglichen im (H)CCH-COSY die Identifikation der meisten überlagerten Positionen aus dem H(C)CH-TOCSY. Für längere Seitenketten, wie K oder R, eignet sich der Ansatz zumindest zur eindeutigen Unterscheidung der Kohlenstoffresonanzen, da sich die Verknüpfung der einzelnen Kerne direkt aus den COSY-Korrelationen ergibt. Die Zuordnung der Seitenkettenprotonen für K und R bleibt wegen der unzureichenden HCCH-TOCSY-Auflösung aber dennoch schwierig.

Die Seitenketten-Amidprotonen von Asparagin und Glutamin wurden über ein CBCA(CO)NH_2-Experiment zugeordnet. Für K und R war keine Zuordnung der Seitenketten-Amidprotonen möglich. Insgesamt wurden 95% der Seitenkettenverschiebungen zugeordnet.

Der Bezug der Seitenkettenresonanzen zum Proteinrückgrat wurde über die aus der sequentiellen Zuordnung bekannten $C^{\alpha/\beta}$- und $H^{\alpha/\beta}$-Verschiebungen hergestellt.

Die diastereotope Unterscheidung der Signale erfolgte wie in Kapitel 2.3 beschrieben über charakteristische NOE-Muster. Insgesamt konnten auf die Weise die H^{β}-Resonanzen von 50 Resten und die Methylprotonen von 18 der 20 Valin zugeordnet werden.

5.3.3 Sekundärstruktur und Topologie

Die Sekundärstruktur von VAT-N wurde über den *chemical shift index* [46], charakteristische NOE-Muster und die $^{3}J(H^{N},H^{\alpha})$-Kopplungskonstanten aus dem HNHA-Spektrum ermittelt (vgl. Zuordnungsstrategie Sekundärstr.). Die Verknüpfung der Faltblätter wurde über NOE-Kontakte festgelegt. Auf diese Weise ergeben sich 12 β-Stränge und 3 α-Helices. Die β-Stränge sind zu 7 antiparallelen und 4 parallelen Faltblättern verbunden. β7 und β10 beinhalten jeweils einen sogenannten *β-bulge*.

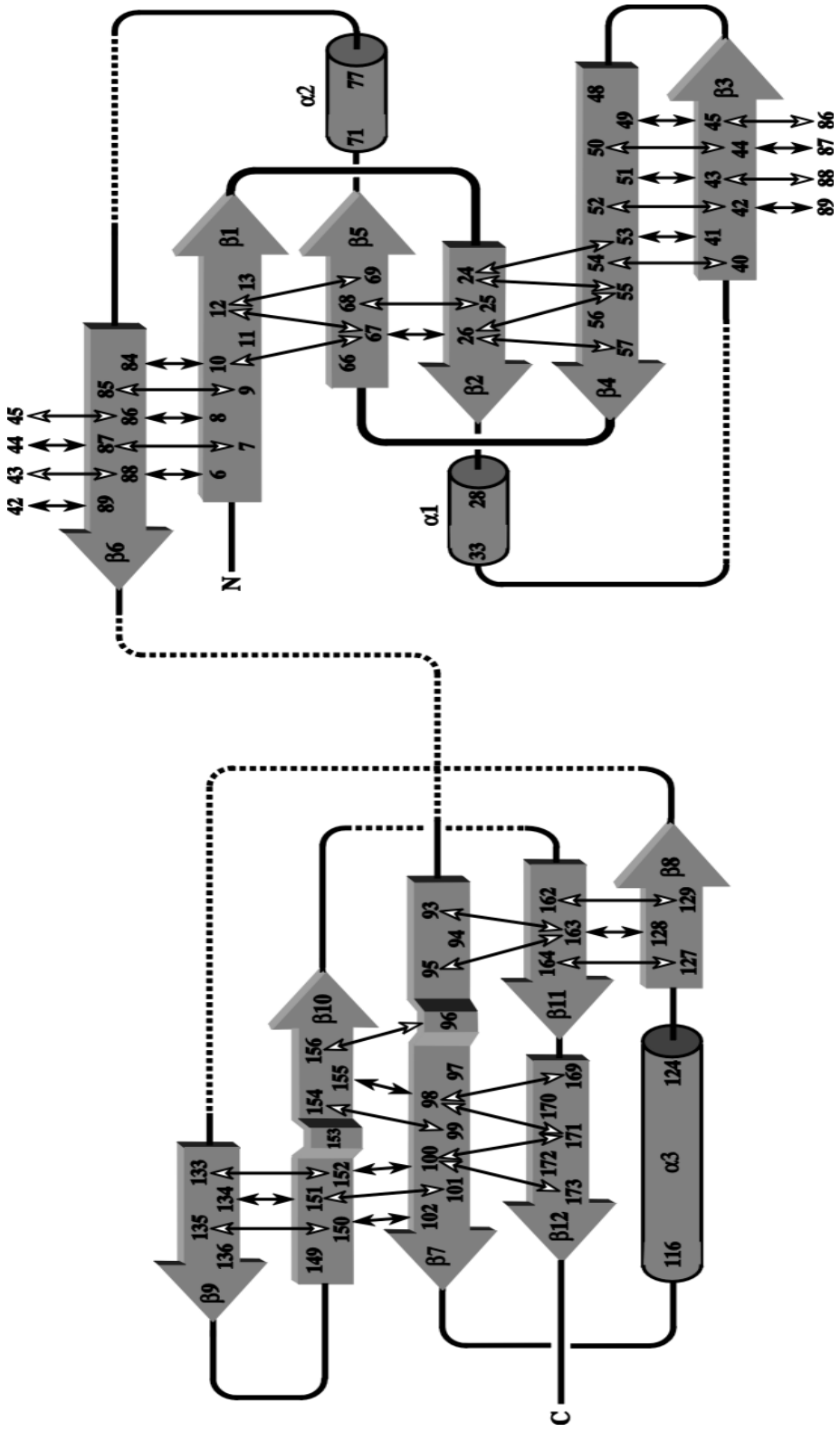

Abbildung 32: *Topologie von VAT-N. Die NOE-Kontakte zwischen den einzelnen Strängen der β-Faltblätter sind mit Pfeilen markiert (weiße Pfeile: H^N-H^N-Kontakte; schwarze Pfeile: H^α-H^α-Kontakte). Die beiden für die N-terminale Unterdomäne charakteristischen Ψ-Schleifen sind etwas dicker eingezeichnet.*

VAT-N lässt sich formal in zwei Unterdomänen, M1-T92 und E93-E183 teilen. Die N-terminale Unterdomäne VAT-Nn verfügt über eine bemerkenswerte Symmetrie, was durch den Befund der internen Sequenzhomologie unterstützt wird. Die Faltblätter in VAT-Nn schließen sich zu einem *β-barrel*. Deutlich sind zwei sogenannte Ψ-Schleifen-Motive zwischen β1-β2 und β4-β5 zu erkennen (vgl. 5.4).
Die resultierende Topologie von VAT-N ist in Abbildung 32 gezeigt.

5.3.4 NOE-Kontakte und Strukturrechnung

Zur Zuordnung der NOE-Daten wurde eine Kombination aus vier komplementären, doppelt heteronuklear editierten 3D-NOESY-Spektren (CCH-, NCH-, NNH- und CNH-NOESY) eingesetzt. Mit Hilfe dieser Technik lässt sich die NOE-Information in Pseudo-4D-Auflösung erhalten [131, 134]. Zur Ergänzung wurden zusätzlich das HNH- und das HCH-NOESY verwendet. Die Signalintensitäten wurden für die Distanzbestimmung in vier Entfernungsklassen eingeteilt: 2,8 Å; 3,3 Å; 4,2 Å und 5,0 Å. Die Kalibration der Signalintensitäten erfolgte mit Hilfe von Referenz-HSQC-Spektren.
Zur Bestimmung der Torsionswinkelangaben wurden ein HNHA- bzw. HNHB-Spektrum benutzt. Die Wasserstoffbrücken wurden über charakteristische H^N-H_2O-Austauschraten aus einem New-MEXICO-Experiment [135] bestimmt.
Die Strukturrechnungen wurden mit dem Programm XPLOR durchgeführt [108]. Die Ausgangsstruktur für die Rechnungen wurde mit 151 ϕ-Winkel-Angaben aus der Topologie-Bestimmung (vgl. 5.3.3) und 562 eindeutigen NOE-Kontakten erzeugt. Die weitere Zuordnung der NOE-Daten erfolgte über einen iterativen Prozess von Strukturrechnung und Spektrenauswertung. Zur Filterung und Abstandsklassifizierung der spektralen Daten wurde eine Reihe von halbautomatischen Hilfsskripten eingesetzt. Die relative Orientierung der beiden Subdomänen wurde über 31 Interdomänen-Kontakten festgelegt. Die Orientierung stützte sich dabei zunächst auf den Kontakt des in der Cdc48-Proteinfamilie hochkonservierten N76-Restes [129] zu N134 (s. 5.4). Dieser konnte über den iterativen Zuordnungsprozess durch weitere 30 Kontakte bestätigt und ergänzt werden.

In Tabelle 12 sind alle für die Strukturrechnung verwendeten Parameter aufgeführt.

Parameter	Anzahl
NOE-Kontakte	
Insgesamt	1923
Intraresidual	460
sequentiell ($\|\Delta i\| = 1$)	643
mittlerer Bereich ($1 < \|\Delta i\| \leq 4$)	224
Weitbereich ($\|\Delta i\| > 4$)	459
Interdomänen	31
Diëder Winkel (ϕ und χ_1)	196
$^3J(H^N,H^\alpha)$	56
Wasserstoffbrücken	51

Tabelle 12: *Auflistung der für die Strukturrechnung verwendeten Parameter. Δi entspricht dem Abstand zweier Reste in der Sequenz.*

Das endgültige Strukturensemble besteht aus 26 Einzelstrukturen und wurde mit den in Tabelle 12 angegeben Parametern bestimmt. In Tabelle 14 sind die RMSD-Werte des Ensembles angegeben. Es wird jeweils zwischen dem Wert für das Proteinrückgrat und dem Wert für alle Schweratome des Proteins unterschieden. Neben den RMSD-Werten des gesamten Proteins wurden noch die RMSD-Werte unter ausschließlicher Berücksichtigung aller sekundärstruktur-fixierten Bereiche bestimmt.

Proteinbereich	Proteinrückgrat	alle Schweratome
gesamtes Protein		
Gesamt	1,48 ± 0,26 Å	1,88 ± 0,26 Å
6-14, 23-57, 66-92, 93-104, 114-136, 149-176*	0,50 ± 0,08 Å	1,03 ± 0,07 Å
N-terminale Subdomäne		
Gesamt	1,58 ± 0,35 Å	1,58 ± 0,35 Å
6-14, 23-57, 66-92*	0,36 ± 0,06 Å	0,36 ± 0,06 Å
C-terminale Subdomäne		
Gesamt	1,48 ± 0,26 Å	1,48 ± 0,26 Å
93-104, 114-136, 149-176*	0,41 ± 0,07 Å	0,41 ± 0,07 Å

Tabelle 13: *RMSD-Werte des endgültigen Strukturensembles (26 Strukturen). Die mit * gekennzeichneten Zeilen enthalten nur Bereiche mit definierter Struktur.*

Für die Überprüfung der Strukturgüte wurde das Programm PROCHECK [132, 133] eingesetzt, mit dessen Hilfe die ϕ,ψ-Winkelverteilung im Ramachandran-Diagramm dargestellt werden kann (Tabelle 14). Es ergeben sich keinerlei Winkel-Verletzungen für das endgültige Strukturensemble, 74,5 % aller Reste befinden sich im Idealbereich des Diagramms. (Die Aminosäuren G, P und der N-Terminus werden bei der Rechnung nicht berücksichtigt).

Bereich im Ramachandran-Diagramm	Prozentsatz der darin enthaltenen Reste
bevorzugte Bereiche	74,5 %
erlaubte Bereiche	22,4 %
Erweitert erlaubte Bereiche	3,1 %
verbotene Bereiche	0 %

Tabelle 14: *Überprüfung der Struktur mit dem Programm PROCHECK [132, 133] ϕ,ψ-Winkelverteilung im Ramachandran-Diagramm. (Die Aminosäuren G, P und der N-Terminus werden nicht berücksichtigt).*

5.4 Struktur

Die Struktur von VAT-N teilt sich in zwei Domänen identischer Länge. Die N-terminale Subdomäne VAT-Nn (M1 bis T92) besteht überwiegend aus zwei Untereinheiten, G6 bis R48 und K49 bis T92. Diese beiden Untereinheiten verfügen über eine Sequenzhomologie von 38% (Abbildung 33).

```
VAT-Nn
     1-5  MESNN
     6-48 GIILRVAEANSTDPGMSRVRLDESSRELLDAEIGDVVEIEKVR
     49-92     KTVGRVYRAROEDENKGIVRIDSVMRNNCGASIGDKVKVRKVRT
     VAT-Nc
     93-138      EIAKKVTLAPIIRKDQRLKFGEGIEEYVQRALIRRPMLEQDNISVP
     139-183   GLTLAGQTGLLFKVVKTLPSKVPVEIGEETKIEIREEPASEVLEE
     (gghhhhhh)
```

Abbildung 33: *Aminosäuresequenz von VAT-N. VAT-N ist in zwei Subdomänen aufgeteilt. Die N-terminale Subdomäne VAT-Nn enthält wiederum zwei Untereinheiten, die über eine Sequenzhomologie von 38% verfügen. Der in Klammern angegebene Bereich ist ein expressionsbedingtes Konstrukt und gehört nicht zur eigentlichen Proteinsequenz.*

Sowohl die Sekundär- als auch die Tertiärstruktur dieser beiden Untereinheiten sind fast identisch. Sie weisen jeweils ein βαββ-Motiv auf. Die beiden strukturhomologen Hälften ergänzen sich zu einem *β-barrel*, bestehend aus vier antiparallelen und zwei parallelen β-Faltblättern. Die β-Stränge der beiden Einheiten lagern sich dabei so aneinander, dass jeder β-Strang ausschließlich Kontakte zu β-Strängen der anderen Einheit bildet. Die beiden Stränge β1 und β4 sind stark gekrümmt und ermöglichen dadurch den Zusammenschluss zum *β-barrel*. Die Schleifen zwischen den Faltblättern β2-β3 und β5-β6 werden aufgrund ihrer charakteristischen Topologie (Abbildung 34) als *Ψ-loops* bezeichnet. Daraus ergibt sich die Bezeichnung „doppeltes *Ψ-barrel*“ für das Strukturmotiv von VAT-Nn (vgl. 5.3.3). Dieses Motiv ist literaturbekannt und wird, Homologievorhersagen zufolge, auch für die N-terminalen Domänen anderer Proteine der Cdc48/p97-Familie erwartet [129, 136].

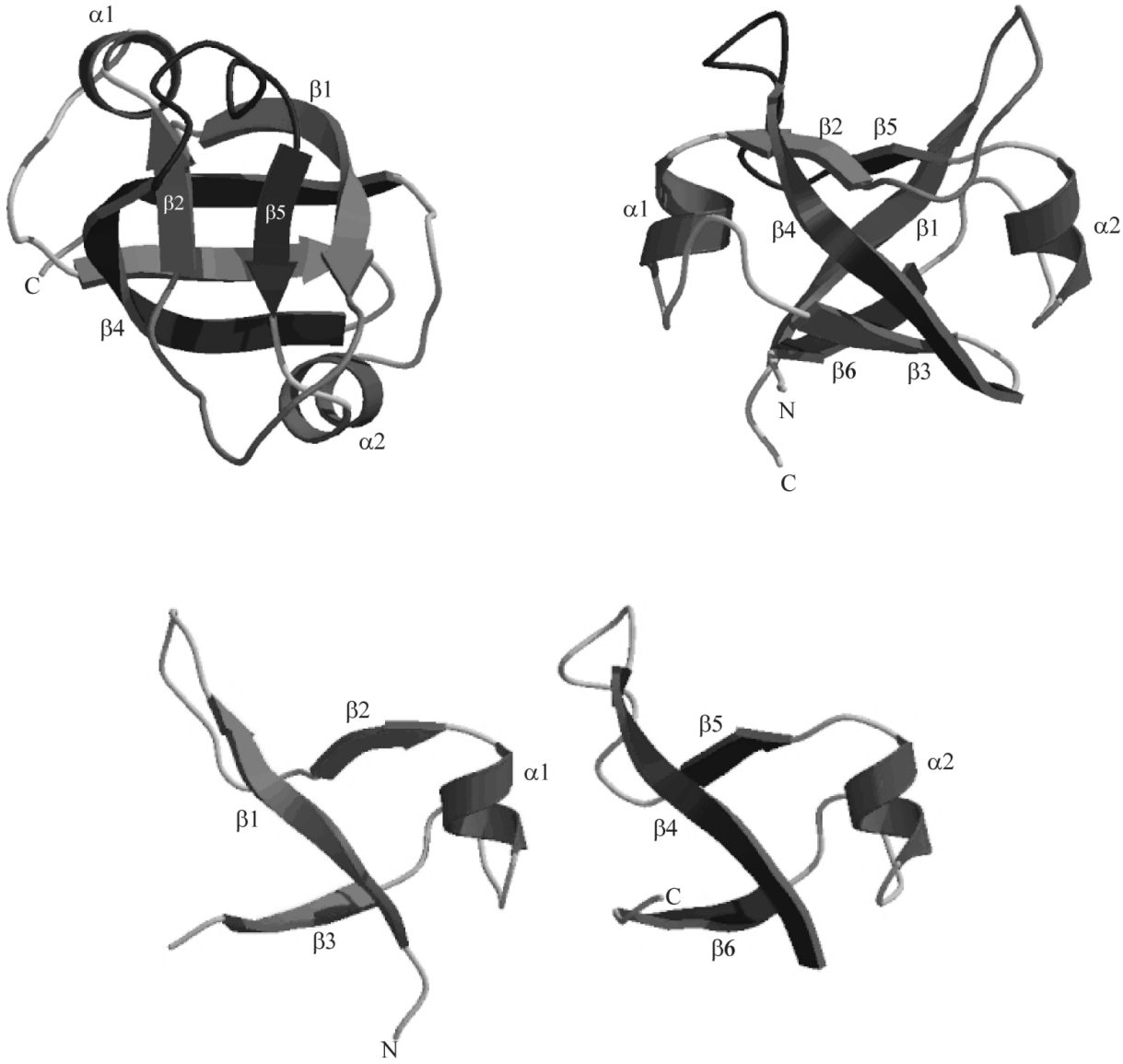

Abbildung 34: *Struktur von VAT-Nn. VAT-Nn (Zeile 1) besteht aus zwei strukturhomologen Untereinheiten mit einem βαββ-Motiv (Zeile 2), die sich zu einem sogenannten „doppelten* Ψ-barrel*“ ergänzen.*

Die C-terminale Unterdomäne VAT-Nc (E93 bis E183) besteht wie VAT-Nn ebenfalls aus sechs β-Strängen. Diese formen jedoch anstatt eines *β-barrels* eine neuartige halboffene *β-clam*-Struktur. Über der durch die Muschel-Konformation gebildeten Mulde liegt quer die Helix α3 und schließt damit einen hydrophoben Bereich von VAT-Nc ab. Besonders auffällig ist der starke Knick *(β-kink*) in β-Strang β7, der nahezu rechtwinklig verläuft. Er bildet das zentrale Element für die *β-clam*-Struktur. Zusammen mit den Strängen β11 und β12 formt er

jeweils ein paralleles β-Faltblatt, mit β10 ein antiparalleles β-Faltblatt. Abbildung 35 zeigt die Struktur von VAT-Nc.

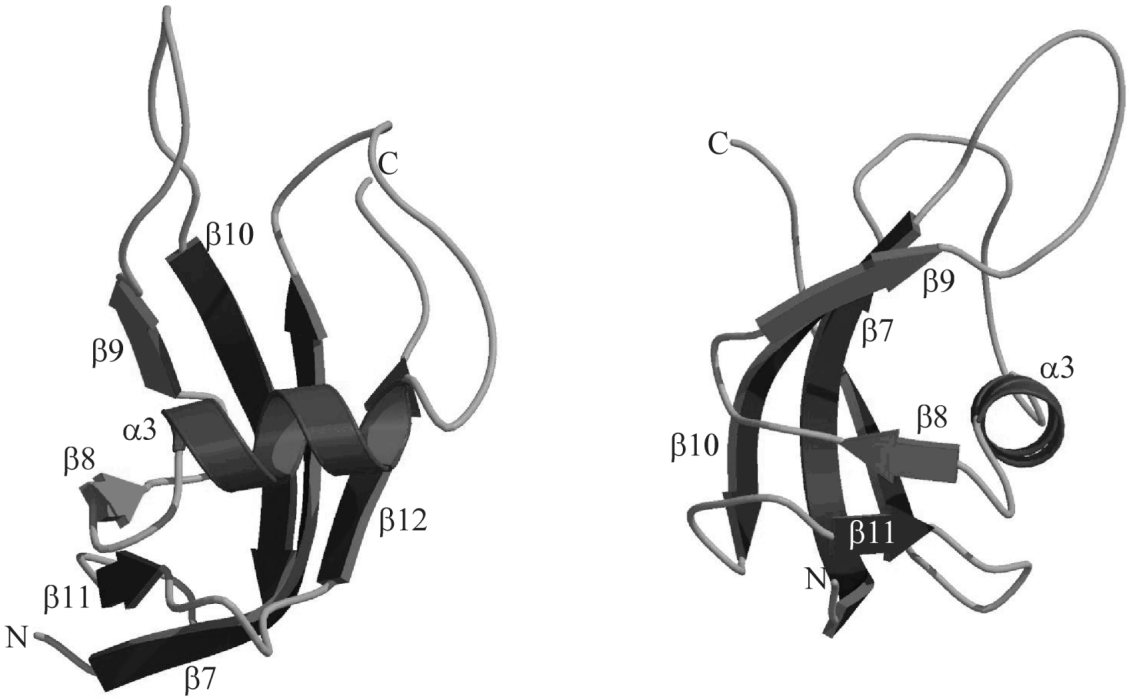

Abbildung 35: *Struktur von VAT-Nc. Die C-terminale Unterdomäne von VAT-N bildet eine halboffene* β-clam-*Struktur.*

Beide Unterdomänen sind über eine kurze Schleife miteinander verbunden (V90 bis T92). Die relative Orientierung der beiden Unterdomänen zueinander wird durch 31 NOE-Kontakte festgelegt (Abbildung 36, Abbildung 37). Das Protein erhält damit eine nierenförmige Gesamtstruktur. Der bedeutendste Kontakt für die Fixierung der Form ist dabei N76/N134. Betrachtet man die Berührungsfläche zwischen den beiden Unterdomänen in der Gesamtstruktur, befindet sich der Kontakt N76/N134 genau an der entgegengesetzten Seite wie die Schleife V90-V92. Er ist damit allein ausreichend, um die Orientierung der beiden Domänen zueinander festzulegen. N76 ist zudem in den Proteinen der Cdc48-Familie hochkonserviert [129], was seine Bedeutung für die Struktur unterstreicht. Vergleicht man eine Superposition von 26 Strukturen des endgültigen Strukturensembles zeigen sich lediglich in den Schleifenbereichen β2-β3, β5-β6, β7-α3 und β9-β10 größere Flexibilitäten. Für die restlichen Bereiche berechnet sich der RMSD-Wert zu 0,50Å für das Proteinrückgrat bzw.

1,03Å, wenn auch die Schweratome der Seitenketten einbezogen werden (vgl. 5.3.4, Tabelle 13). Die Orientierung der beiden Unterdomänen zueinander bleibt fest.

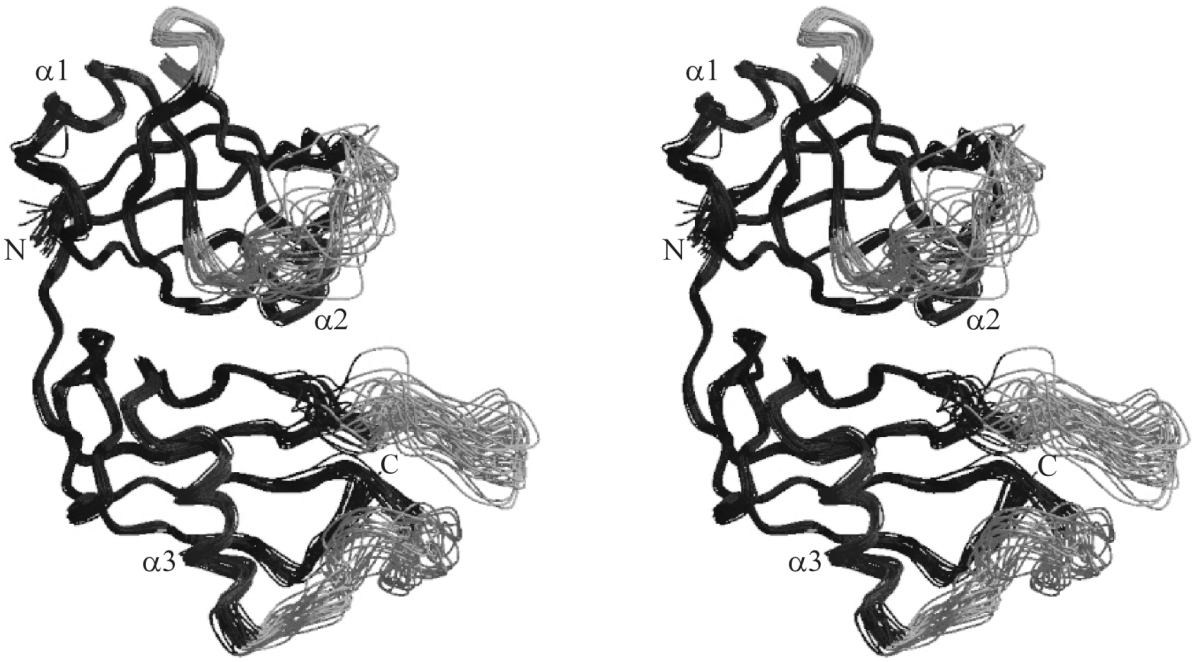

Abbildung 36: *Gesamtstruktur von VAT-N. Eine Superposition von 26 Strukturen des endgültigen Strukturensembles zeigt eine sehr gute Definition sowohl der beiden Subdomänen als auch deren relative Orientierung zueinander.*

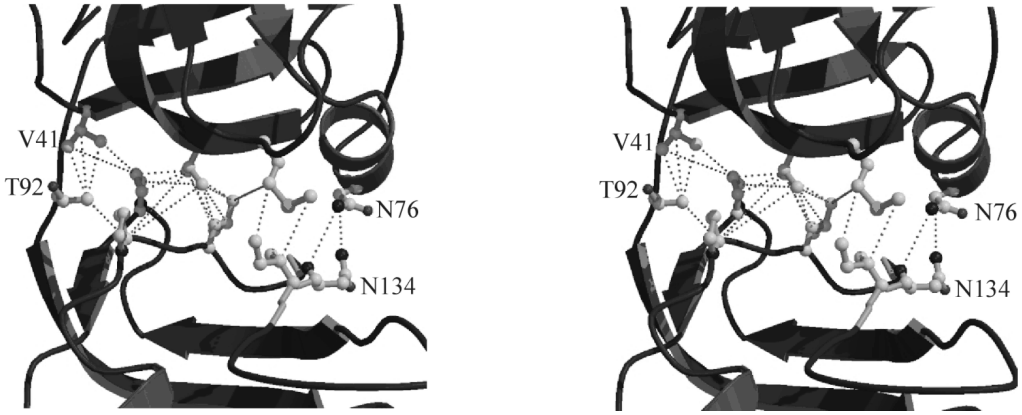

Abbildung 37: *Die gegenseitige Ausrichtung der beiden Unterdomänen von VAT-N wird über 31 NOE-Kontakte fixiert. Der Kontakt N76-N134 spielt dabei eine wichtige Rolle.*

Die Ergebnisse der Strukturrechnung stimmen sehr gut mit elektronenmikroskopischen Aufnahmen von VAT überein. Diese Aufnahmen zeigen, dass sechs VAT-Einheiten sich zu einem Ring mit ca. 15,5 nm Durchmesser zusammenschließen [130]. Vergleicht man die elektronenmikroskopischen Aufnahmen mit denen der Deletionsmutante VAT(ΔN), kann ein Differenzbild erzeugt werden. Die Lage und Struktur von VAT-N kann auf diese Weise gut überprüft werden [137]. Abbildung 38 zeigt in a) die Aufnahme von VAT und in b) die Aufnahme von VAT(ΔN). Das Differenzbild zwischen a) und b) ist in c) dargestellt. Im Differenzbild ist die nierenförmige N-terminale Domäne von VAT an der Außenseite des hexameren Ringes deutlich zu erkennen. Alle elektronenmikroskopischen Bilder verfügen über eine Auflösung von 20 nm. Die Bilder e) und f) zeigen jeweils die ermittelte Struktur von VAT-N, einmal als *ribbon*-Modells und einmal als Kalotten-Modell. In g) ist eine Simulation der Oberfläche der Struktur dargestellt, wie sie mit der elektronenmikroskopischen Auflösung von 20 nm erscheinen würde. Bild d) zeigt schließlich eine Kombination der Oberfläche der ermittelten NMR-Struktur von VAT-N (g) mit dem Konturmodell des elektronenmikroskopischen Differenzbildes (c). Die ermittelte NMR-Struktur stimmt sehr gut mit den elektronenmikroskopischen Ergebnissen überein. Die relative Orientierung der beiden Subdomänen von VAT-N wird damit weiter bestätigt.

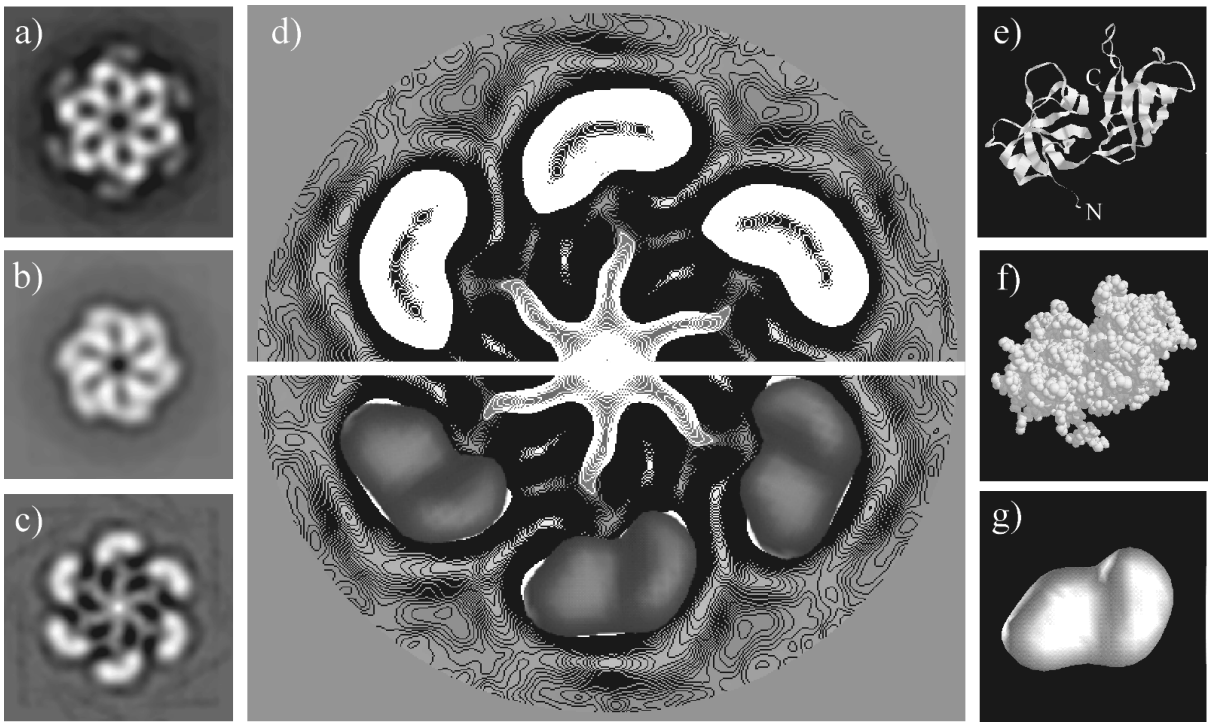

Abbildung 38: *Elektronenmikroskopische Aufnahme von VAT und VAT(ΔN) und deren Übereinstimmung mit der ermittelten Struktur für VAT-N. Siehe Text für eine genauere Erklärung der einzelnen Bereiche der Abbildung.*

6 Zuordnung des NADPH-Komplexes von Dihydrofolat-reduktase aus *Escherichia coli*

6.1 Biochemischer Hintergrund

Dihydrofolatreduktase (DHFR) ist ein essentielles Enzym, dass in der Zelle für die Reduktion von 7,8-Dihydrofolat zu 5,6,7,8-Tetrahydrofolat mit NADPH als Coenzym benötigt wird. Tetrahydrofolat dient in vielen Reaktionen, wie z.B. der Thymidylsäuresynthese für den DNS-Aufbau, als C1-Gruppen-Überträger und Reduktionsmittel.

Da die Kohlenstoffübertragung nur von Tetrahydrofolat nicht jedoch von Dihydrofolat erfolgen kann, muss Tetrahydrofolat nach der Reaktion wieder regeneriert werden. Diese Regeneration wird durch das Enzym Dihydrofolatreduktase katalysiert. Als Reduktionsmittel dient dabei das Coenzym NADPH. Während der Reaktion sind NADPH und Folat an das aktive Zentrum der DHFR gebunden. Beide Substrate sind eng aneinander gebunden. Ihr Abstand ist geringer als der van-der-Waals-Abstand und begünstigt damit die Bildung eines Übergangszustandes für den Elektronentransfer. Im Gegensatz zu vielen Mikroorganismen und Pflanzen kann der Mensch Tetrahydrofolat nicht *de novo* synthetisieren, sondern ist auf die Aufnahme von Folsäure mit der Nahrung angewiesen. Für die Umsetzung von Folsäure zur Dihydrofolsäure wird im menschlichen Organismus ebenfalls das Enzym DHFR benötigt.

Aufgrund der zentralen Bedeutung von Tetrahydrofolat in vielen Zellreaktionen, stellt die Tetrahydrofolatsynthese einen Angriffspunkt für zahlreiche Chemotherapeutika dar. So inhibieren beispielsweise die beiden Folatanaloga Trimethoprim und Methotrexat die bakterielle bzw. die menschliche DHFR, während Sulfonamide direkt in die Biosynthese von Tetrahydrofolat eingreifen.

Methotrexat wird insbesondere als Zytostatikum bei der Krebstherapie eingesetzt. In Zellen mit einer sehr hohen Teilungsrate, wie z.B. Tumorzellen, werden große Mengen an Nucleotidbausteinen zum DNS-Aufbau benötigt. Für die Krebstherapie wird die Hemmung von DHFR ausgenutzt, um störend in die Synthese des Nucleotids Thymin einzugreifen und damit die Replikation von Tumorzellen zu hemmen. Methotrexat zeigt jedoch viele toxische Nebenwirkungen, da es unselektiv auf alle Zellen mit hoher Teilungsrate wirkt. Außerdem

tritt während der Behandlung eine Resistenz der Zellen gegen die hemmende Wirkung von Methotrexat ein, so dass eine hohe Dosierung notwendig ist.

Trimethoprim wird zur Behandlung von Infektionen verwendet. Aufgrund geringer Unterschiede am aktiven Zentrum bindet es um Faktor 10^5 stärker an die DHFR anfälliger Mikroorganismen als an die DHFR von Säugern.

Die DHFR aus *Escherichia coli* ist neben dem Enzym aus *Lactobacillus casei* die sowohl strukturell als auch reaktionskinetisch am genauesten untersuchte DHFR. Zu verschiedenen Ligandkomplexen liegen Röntgenstrukturen vor [138]. Aufgrund der strukturellen und kinetischen Untersuchungen kann auf ein ausgeprägtes dynamisches Verhalten des Moleküls geschlossen werden, das entscheidend an den katalytischen Funktionen von DHFR beteiligt ist [139]. Es konnten zwei unterschiedliche Konformationen des Moleküls nachgewiesen werden, von denen nur eine in der Lage ist, NADPH zu binden [140]. Die strukturellen Unterschiede zwischen den beiden Konformationen beschränken sich hauptsächlich auf die Ausrichtung der drei Schleifen 14-24, 64-71 und 116-125. Mit Hilfe von NMR-Untersuchungen konnten diese Ergebnisse bestätigt und auf andere Komplexe übertragen werden. Zusammenfassend konnten folgende Aussagen getroffen werden: Die Komplexe DHFR-NADPH und DHFR-Folat liegen unikonformationell vor, für die beiden Komplexe DHFR-Methotrexat und DHFR-$NADP^+$-Trimethoprim existiert wie beim unligandierten Protein ein Gemisch aus zwei Konformeren [141, 142].

6.2 Konzept

Die Dihydrofolatreduktase weist in unligandierter Form mehrere Konformationen auf [140]. Die resultierenden NMR-Spektren können aus diesem Grund nur sehr schwer ausgewertet werden und sind für SAR-*screening*-Verfahren [143] ungeeignet. Da die DHFR über zwei Bindungsstellen verfügt, kann der *screening*-Vorgang jedoch gesondert für jede der beiden Bindungstaschen durchgeführt werden. Die zweite Bindungsstelle kann jeweils mit einem Liganden versehen werden, der die DHFR in einer Konformation fixiert. Für dieses Verfahren bieten sich die Liganden Folat und NADPH an, da die DHFR mit beiden einen

unikonformeren Komplex bildet und die Liganden unterschiedliche Bindungstaschen belegen. Die NMR-Zuordnung für den Folat-Komplex war bereits zu Beginn des Projekts literaturbekannt [144]. Für den NADPH-Komplex musste eine neue Zuordnung erstellt werden.

Neben der NMR-Zuordnung des Folat-Komplexes sind sowohl die Röntgenstruktur des Folat- als auch die des NADPH-Komplexes von DHFR aus *Escherichia coli* bekannt [138]. In beiden Komplexen weist die DHFR die gleiche Struktur (RMSD-Wert bezogen auf das Proteinrückgrat: 1,6Å) auf. Ein Vergleich zweier HSQC-Spektren des Folat- bzw. des NADPH-Komplexes zeigt jedoch deutliche Unterschiede. Eine direkte Übernahme der NMR-Zuordnung von einem Komplex zum anderen ist somit trotz identischer Struktur von DHFR nicht möglich.

Um dennoch nutzbringend auf die Literaturdaten zurückgreifen zu können wurde eine neue Strategie zur Zuordnung von Proteinkomplexen entwickelt. Das Zuordnungskonzept setzt voraus, dass bereits die NMR-Zuordnung eines anderen Komplexes desselben Proteins vorliegt und das Protein sich in beiden Komplexen nur unwesentlich in der Struktur unterscheidet. Dies lässt die Annahme zu, dass sich die chemischen Verschiebungen eines Großteils der Signale trotz starker optischer Unterschiede der HSQC-Spektren nur geringfügig verändern. Im HSQC-Spektrum ist eine eindeutige Identifikation der verschobenen Peaks aufgrund der geringen Signaldispersion schwierig. Überlagerungen, insbesondere im Verschiebungsbereich 117 ppm – 123 ppm (^{15}N) und 7,5 ppm – 9 ppm (^{N}H), machen eine Zuordnung der Signale anhand eines 2D-Spektrums unmöglich. Daher wurde als Datengrundlage für die Signalzuordnung ein dreidimensionales HNHA-Spektrum gewählt. Die Aufnahme eines solchen Spektrums erfordert lediglich eine ^{15}N-Isotopenmarkierung der Proteinprobe. Das HNHA eignet sich damit sehr gut als Basis für die Entwicklung einer standardisierten Zuordnungsprozedur.

Zunächst wird eine Initialzuordnung durch Abbilden des NADPH-HNHA-Spektrums auf das bereits zugeordnete Folat-HNHA erzeugt. Die Initialzuordnung wird anschließend mit Hilfe von NOESY-Daten verifiziert und korrigiert (Abbildung 39). Die beiden angegebenen NOESY-Spektren (HNH-NOESY und NNH-NOESY) benötigen ebenfalls nur eine ^{15}N-isotopenmarkierte Probe, so dass das gesamte Verfahren im Idealfall mit dieser Probe durchgeführt werden kann.

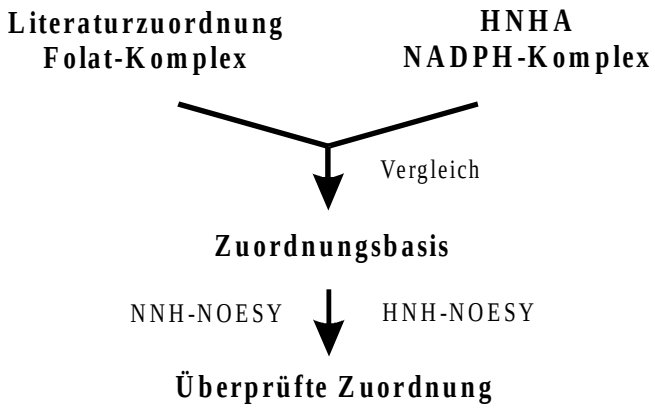

Abbildung 39: *Konzept für die NMR-Zuordnung des NADPH-Komplexes von DHFR unter Verwendung der Literaturzuordnung des Folat-Komplexes.*

Das Konzept ist in der Lage, alle geringfügig verschobenen Signale zu identifizieren. Diese können wiederum als Grundlage für die weitere Zuordnung von stark verschobenen Resten dienen, wie sie z.B. in der Nähe der Bindungsstelle zu erwarten sind.

6.3 Experimentelles

Die Durchführung des Projektes erfolgte in Zusammenarbeit mit K. Abelmann (NMR-Messungen). Die Protein-Proben wurde von H. Lüttgen (TUM, Institut f. Org. Chemie u. Biochemie) aus *E. coli* exprimiert. Nähere Angaben über die Probenexpression finden sich in [145].

Für die NMR-Messungen wurden drei Proben verwendet:

- 1,0 mM [U-^{15}N], 1,3 mM Folat:
  wässrige Lösung (DMSO < 5%); pH 6,9 – pH 7,0; 20 mM Phosphatpuffer, 100 mM KCl
- 1,2 mM [U-^{15}N], 1,5 mM NADPH:
  wässrige Lösung (DMSO < 5%); pH 6,9 – pH 7,0; 20 mM Phosphatpuffer, 100 mM KCl

- 2,5 mM [U-^{13}C-^{15}N], 3,0 mM NADPH:
  wässrige Lösung (DMSO < 5%); pH 6,9 – pH 7,0; 100 mM Phosphatpuffer, 200 mM KCl

Alle Messungen wurden bei einer Temperatur von 300 K auf einem DMX600-Spektrometer der Firma Bruker durchgeführt.
Für die Zuordnung wurden die folgenden Experimente aufgenommen:

- Folat-Komplex:
  [U-^{15}N] Probe: ^{15}N-HSQC, HNHA, HNH-NOESY, NNH-NOESY
- NADPH-Komplex:
  [U-^{15}N] Probe: ^{15}N-HSQC, HNHA, HNH-NOESY, NNH-NOESY
  [U-^{13}C,^{15}N] Probe: HNCACB, HNCAHA, HNCA

6.4 Abbildung der Folat-Daten

Vor der Analyse der NADPH-Spektren muss zunächst die Literaturzuordnung des Folat-Komplexes auf dessen experimentelle Daten abgebildet werden. Eine direkte Übernahme auf Basis der HSQC-Daten ist nicht möglich, da die meisten Signale geringfügige Verschiebungsunterschiede gegenüber den Literaturwerten aufweisen. Aufgrund der oben geschilderten Überlagerungsproblematik (s. 6.2) wurde die Literaturzuordnung des Folatkomplexes ebenfalls anhand eines HNHA-Spektrums nachvollzogen (s. 6.5). Abbildung 40 zeigt die erhaltene Zuordnung.

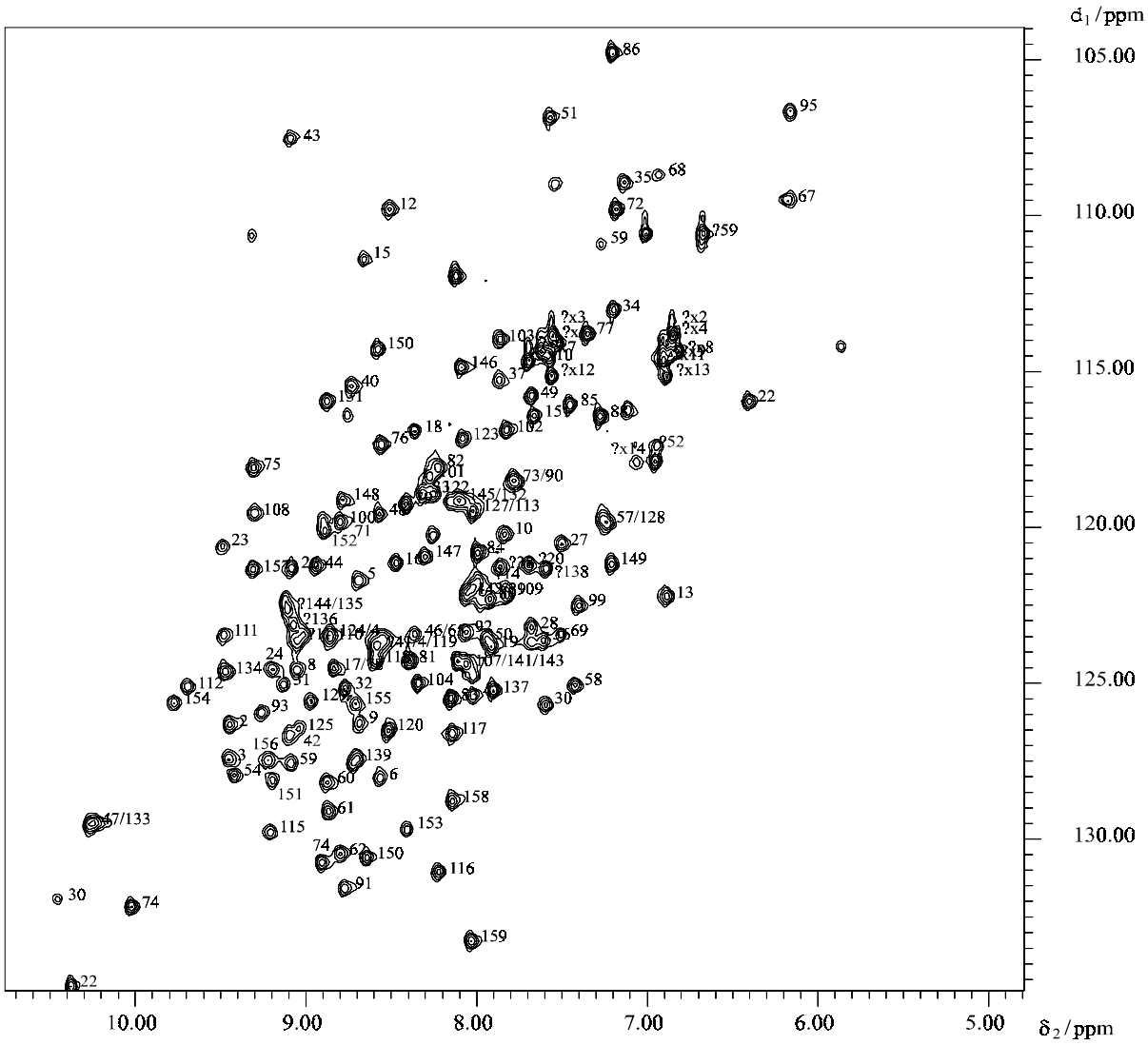

Abbildung 40: *^{15}N-HSQC-Zuordnung des Folat-Komplexes von DHFR aus* E. coli.

6.5 Zuordnungsbasis

Die Zuordnung der Signale wurde zunächst über eine Rückrechnung der chemischen Verschiebung aus der Röntgenstruktur versucht. Dazu wurde das Programm XPLOR [108] verwendet: XPLOR bietet ein Modul zum Abgleich der Dihedralwinkel einer Struktur mit einer Liste von experimentellen chemischen Verschiebungen an. Es berechnet aus der Struktur die Abweichungen zwischen den vorgegebenen experimentellen Verschiebungen und den

nach den Dihedralwinkeln zu erwartenden Werten. Gibt man eine Liste mit lauter Null-Werten für die chemische Verschiebung vor, kann mit XPLOR ein Vorschlag für die chemische Verschiebung aus der Struktur errechnet werden.

Als Testdatensatz wurde die bekannte Zuordnung des Folat-Komplexes eingesetzt. Die Abweichungen zwischen berechneten und gemessenen Verschiebungen sind jedoch deutlich zu hoch, um die berechneten Verschiebungen für eine Zuordnung sinnvoll nutzen zu können (Abweichung für H^{α} bis zu 50% von den tatsächlichen Werten). Die großen Unterschiede sind zum einen auf die Ungenauigkeit der Methode [108], zum anderen auf die fehlende Möglichkeit zur Berücksichtigung der Ligandeneinflüsse während der Rechnung zurückzuführen.

Als alternative Zuordnungsmethode wurde die Nachbarschaftssuche gewählt. Als Zuordnungsbasis wurde wiederum das HNHA-Spektrum verwendet, da hier drei verschiedene Verschiebungsdimensionen für eine Nachbarsuche zur Verfügung stehen.

Für die Nachbarsuche wird die Summe über die Differenzen der chemischen Verschiebungen zweier Peaks in allen drei Dimensionen des HNHA gebildet. Jede Dimension kann zusätzlich noch mit einem Wichtungsfaktor versehen werden, um die unterschiedliche Breite des Verschiebungsbereiches für die verschiedenen Kerne auszugleichen. Für die Rechnungen in dieser Arbeit wurden die Wichtungsfaktoren wie folgt gesetzt: $^{15}N = 1$, $H^{N} = 3$, $H^{\alpha} = 4$. Dies entspricht in etwa dem Verhältnis der Verschiebungsintervalle für die einzelnen Kerne. Jedem Peak wird schließlich das Signal des korrespondierenden Spektrums zugeordnet, das die geringste Verschiebungsdifferenz aufweist. Mehrfachzuordnungen eines Peaks sind auf diese Weise möglich. Ein Ausschluss von Mehrfachzuordnungen mittels *best first* oder Minimierungs-Methoden ist nicht sinnvoll, da für das Rechenverfahren aus Vereinfachungsgründen Peaklisten anstatt von unmittelbaren Spektrendaten verwendet werden. Eine genauere Charakterisierung der gewanderten Peaks, wie sie z.B. durch eine Linienformanalyse erreicht werden kann [57], ist mit Peaklisten nicht möglich. Aus diesem Grund wird zunächst der jeweils nächste Nachbar eines Peaks als wahrscheinlichste Lösung angenommen, bis im weiteren Verlauf der Zuordnung eine genauere Untersuchung möglich ist.

Diese Methode wurde ebenfalls zur Abbildung der experimentellen Daten des Folat-Komplexes auf die Literaturzuordnung verwendet (s. 6.4). Sie liefert einen vollständigen Übertrag der Literaturdaten und bestätigt damit die Methode.

Für die NADPH-Daten kann die Zuordnung jedoch lediglich als Basis für nachfolgende Schritte gesehen werden, da die Nachbarschaftssuche nur unter der Annahme geringer Verschiebungsunterschiede erfolgreich ist. Die Nachbarschaftssuche liefert damit eine Vorauswahl der durch den Liganden unbeeinflussten Signale. Diese können anschließend einfach durch weitere Informationen bestätigt werden (s. 6.6, 6.8). In der Nähe der Bindungsstelle greift das Konzept wegen der zu erwartenden Verschiebungsänderungen nicht. Lediglich die Anzahl der möglichen Zuordnungen wird durch die Nachbarschaftssuche auf ein übersichtliches Maß eingeschränkt.

6.6 Vergleich von NOESY-Spuren

Eine Möglichkeit zur weiteren Klassifizierung der Peaks ist der Vergleich von NOESY-Spuren. NOESY-Spektren liefern Abstandsinformationen zwischen einzelnen Atompaaren eines Moleküls. Für jede Aminosäure eines Proteins ergibt sich damit ein charakteristisches -Signalmuster in den NOESY-Spektren. Diese Signalmuster können zur Zuordnung des NADPH-Komplexes eingesetzt werden. Da Folat- und NADPH-Komplex über die gleiche Struktur verfügen, kann angenommen werden, dass die korrespondierenden NOESY-Spuren für beide Komplexe ähnlich sind. Es wird wie bei der Nachbarschaftssuche vorausgesetzt, dass ein Großteil der Signale nur geringe Änderungen erfährt und das charakteristische Signalmuster erhalten bleibt. Diese Annahme wird zusätzlich durch die Tatsache gestützt, dass mehr als die Hälfte der beobachteten Signale Kontakte zu Seitenkettenprotonen darstellen. Deren chemische Verschiebungen sind im Vergleich zu den Rückgratprotonen nur über einen geringen Bereich variabel. Die Einflüsse von Verschiebungsänderungen auf das Signalmuster sind folglich gering.

Zudem finden sich in den NOESY-Spuren im Allgemeinen die chemischen Verschiebungen von sequentiell benachbarten Aminosäuren wieder (z.B. $H^{\alpha}[i-1]$, $H^{N}[i+1]$), die ebenfalls als Bewertungskriterium für die Zuordnung verwendet werden können. Für den NOESY-Vergleich wurden ein HNH-NOESY und ein NNH-NOESY verwendet.

Mit Hilfe dieser beiden Spektren konnte die Zuordnung des NADPH-Komplexes aus der Nachbarschaftssuche verfeinert und Mehrfachzuordnungen teilweise eliminiert werden.

Zur Beurteilung der Zuordnung der einzelnen Aminosäuren wurde die empirische Bewertungsfunktion *f(x)* (x = Sequenzposition) aufgestellt. Positive Funktionswerte kennzeichnen eine hohe Wahrscheinlichkeit, die korrekte Zuordnung getroffen zu haben. Die Funktion *f(x)* berücksichtigt für jeden Rest komplett alle vorliegenden Daten und gewichtet sie entsprechend ihrer Aussagekraft.

Bewertungsfunktion *f(x)*:

$$f(x) = 5a + 5b + 10c - 2d \cdot e + 5f$$

$$a = \begin{cases} 1 = \textit{Übereinstimmung der HNH} - \textit{NOESY} - \textit{Spuren von NADPH und Folat} \\ 0 = \textit{Abweichung der HNH} - \textit{NOESY} - \textit{Spuren von NADPH und Folat} \end{cases}$$

$$b = \begin{cases} 1 = \textit{Signale zum sequentiellen Nachbarn im HNH} - \textit{NOESY} \\ 0 = \textit{keine Signale zum sequentiellen Nachbarn im HNH} - \textit{NOESY} \end{cases}$$

$$c = \begin{cases} 1 = \textit{Signale zum sequentiellen Nachbarn im NNH} - \textit{NOESY} \\ 0 = \textit{keine Signale zum sequentiellen Nachbarn im NNH} - \textit{NOESY} \end{cases}$$

$$d = \frac{\Delta H^{N} + 3\Delta N + 4\Delta H^{\alpha}}{ppm}$$

$$e = \textit{Anzahl der umliegenden Peaks in } 0{,}7\ \textit{ppm Radius}$$

$$f = \begin{cases} 1 = \textit{keine weiteren Peaks in } 0{,}7\ \textit{ppm Radius} \\ 0 = \textit{noch andere Peaks in } 0{,}7\ \textit{ppm Radius} \end{cases}$$

Sequentielle Kontakte des NNH-NOESY (c) werden doppelt so stark bewertet wie die des HNH-NOESY (b). Die Information des NNH-NOESY ist aufgrund der geringeren Signaldichte des Spektrums und der größeren Verschiebungsdispersion für ^{15}N wesentlich zuverlässiger als die des HNH-NOESY. Für den Mustervergleich der NOESY-Spuren (a) wird dagegen nur das HNH-NOESY verwendet, da die Spuren des NNH-NOESY mit typischerweise nur zwei bis drei Signale über zu wenige charakteristische Details verfügen.
Die Abweichung der chemischen Verschiebungen zweier Signale (d) wird durch die Anzahl der umliegenden Peaks in einem bestimmten Radius modifiziert (e). Je höher die Anzahl der benachbarten Peaks ist, desto höher ist die Wahrscheinlichkeit, dass die richtige Zuordnung auch auf einen anderen Peak der Umgebung zutreffen kann. Die Wahrscheinlichkeit für eine korrekte Zuordnung ist in Gebieten mit einer hohen Peakdichte daher niedriger als für isolierte Signale. Der Suchradius orientiert sich dabei an den Auflösungstoleranzen der verwendeten

Spektren (^{15}N: 0,3 ppm, H^N: 0,01 ppm, H^α: 0,01 ppm). Über Testrechnungen wurde der effektivste Wert für den Radius zu 0,7 ppm ermittelt. Die entspricht in etwa dem Doppelten der Summe der Auflösungstoleranzen. Befindet sich kein weiterer Wert im angegebenen Testradius, wird ein zusätzlicher Bonus auf die Zuordnungswahrscheinlichkeit vergeben. Der Wert bekräftigt Zuordnungen, die sich in Gebieten mit geringer Signaldichte befinden.

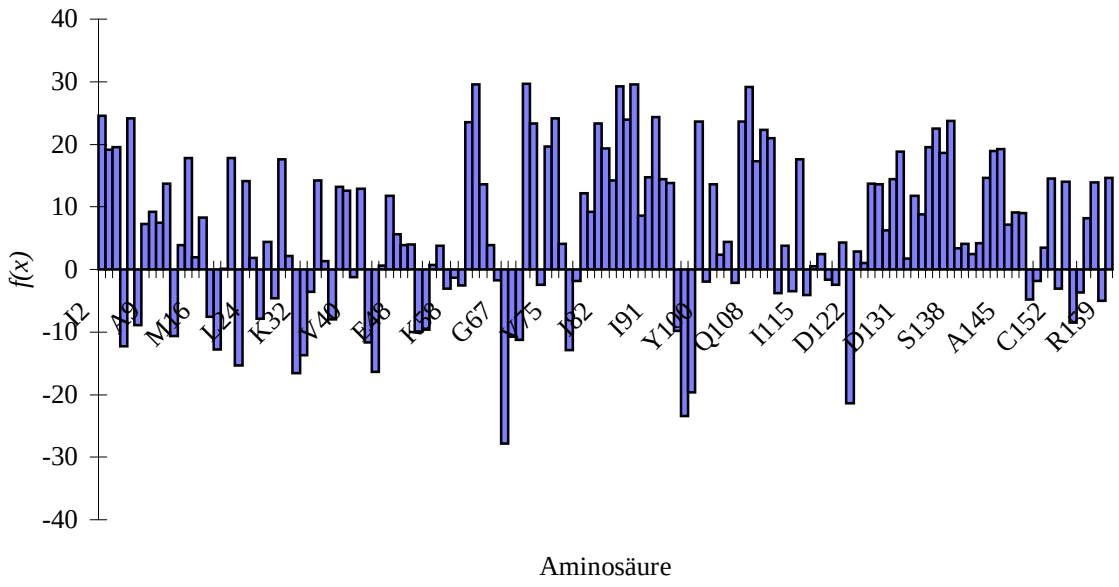

Abbildung 41: *Bewertung der Zuordnung des NADPH-Komplexes mittels einer empirischen Bewertungsfunktion.*

In Abbildung 41 ist das Ergebnis der Bewertung dargestellt. Positive Werte zeigen eine gesicherte Zuordnung, negative Werte weisen auf unsichere Stellen hin. Auffällig ist insbesondere der Bereich zwischen den Aminosäuren 14 und 70. Dieser Bereich wird hauptsächlich durch die beiden Bindungsstellen von NADPH und Folat geprägt. In diesem Bereich befinden sich erwartungsgemäß viele negative Werte, da in der Nähe der Bindungsregionen die stärksten Änderungen der chemischen Verschiebung zu erwarten sind.

6.7 Probeninstabilität der NADPH-Proben

Die Zuordung des NADPH-Komplexes wurde durch Probeninstabilitäten während der Messung wesentlich erschwert. Ein optischer Vergleich zwischen dem HNH-NOESY des NADPH- und des Folat-Komplexes zeigt, dass im NADPH-Spektrum deutlich mehr Signale enthalten sind (Abbildung 42). Dies könnte auf eine Zersetzung des Komplexes während der Messung und daraus folgend ein Gleichgewicht mehrerer Konformationen hindeuten. Für manche Strips ist deutlich ein zweites, verschobenes Abbild der selben Spur im NOESY-Spektrum zu erkennen. Auch ein Vergleich von HSQC-Spektren, die vor bzw. nach jeder einzelnen Messung als Referenz aufgenommen wurden, zeigt deutliche Veränderungen der Probe. Ein ähnliches Verhalten wird in der Literatur [146] auch für den Methotrexat-Komplex von DHFR aus *E. coli* berichtet. Die Autoren beschreiben zwei unterschiedliche Sets von Signalen im Verhältnis 2:1. Bei Mischzeiten größer 700 ms konnten Austauschsignale zwischen den beiden Konformeren beobachtet werden und damit eine langsame Umlagerung der beiden Komplexe jenseits der üblichen NMR-Zeitskala nachgewiesen werden. Das Konformerengleichgewicht von DHFR ist gut untersucht und wird auch in anderen Arbeiten beschrieben [140]. Nur eine der beiden Konformationen ist jedoch in der Lage NADPH zu binden.

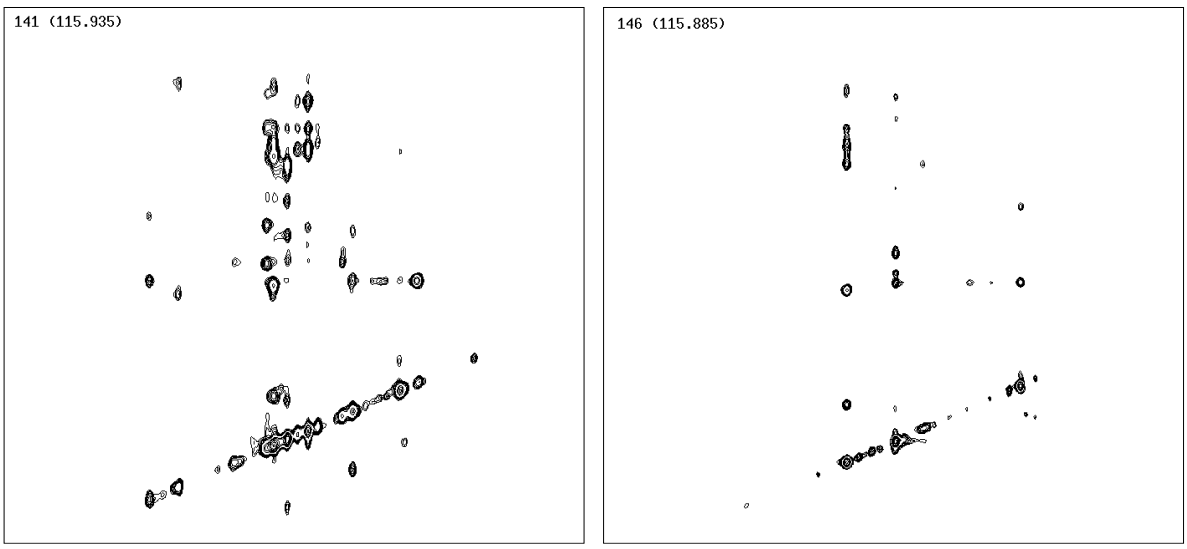

Abbildung 42: *Vergleich zwischen zwei Ebenen mit gleicher ^{15}N-Verschiebung eines HNH-NOESY-Spektrums des NADPH-Komplexes (links) und des Folat-Komplexes (rechts).*

Hsu et al. [147] beobachteten, dass DHFR die Oxidation von NADPH zu $NADP^+$ auch in Abwesenheit eines reduzierbaren Substrats katalysieren kann. Dies legt nahe, dass die Veränderungen der Probe hauptsächlich auf die Entstehung von $NADP^+$ zurückzuführen sind.
Über eine Messreihe von mehreren HSQC-Spektren nach verschiedenen Zeitabschnitten wurde die durchschnittliche Haltbarkeit der Probe auf ca. zwei bis drei Tage bestimmt.

6.8 Verwendung einer [U-^{13}C,^{15}N]-isotopenmarkierten Probe

Aufgrund der genannten Probeninstabilität ist eine zweifelsfreie Zuordnung des NADPH-Komplexes mit den bisher verwendeten Spektren nicht möglich. Um dennoch eine eindeutige Aussage über die Konnektivität bzw. die Zuordnung treffen zu können, sind Spektren notwendig, die definierte Korrelationen zwischen den Kernen zweier benachbarter Aminosäuren zeigen. Dazu wird eine [U-^{13}C,^{15}N]-markierte Probe benötigt, um den Magnetisierungstransfer über Kohlenstoff hinweg zu ermöglichen. Um das Problem der Probeninstabilität zu umgehen, wurde beschlossen eine besonders stark konzentrierte Proteinprobe zu erwenden (2,5 mM). Diese ließ für jedes der aufgenommenen Spektren eine Verkürzung der Messzeit auf ca. ein Drittel des ursprünglichen Zeitbedarfs zu.
Es wurde ein minimaler Spektrensatz bestehend aus einem HNCACB, HNCA und HNCAHA aufgenommen. Die Aufnahme weiterer Spektren war aufgrund der eingetretenen Probenveränderungen nicht möglich. Bereits nach Aufnahme des ersten Spektrums (HNCACB) traten deutliche Unterschiede in den Referenz-HSQC-Spektren auf. Für das HNCA und HNCAHA wurde daher das HNCACB als Vorlage benutzt, um die Signale des NADPH-Komplexes gegenüber neuentstandenen Signalen zu identifizieren.
Die ^{13}C-Verschiebungen wurden außerdem zur Bestimmung des Aminosäuretyps durch Vergleich mit *random coil*-Tabellen verwendet werden. Der Aminosäuretyp dient als weiteres Kriterium zur Zuordnungsbewertung.
Mittels dieser Spektren und den Informationen aus den vorhergehenden Schritten erfolgte die endgültige Zuordnung. In Abbildung 43 ist das zugeordnete ^{15}N-HSQC-Spektrum des

NADPH-Komplexes von DHFR gezeigt. Bei den unbeschrifteten Signalen handelt es sich um Seitenkettenresonanzen bzw. Signale, die nicht eindeutig identifiziert werden konnten. Insgesamt wurden 145 der 159 Aminosäuren zugeordnet.

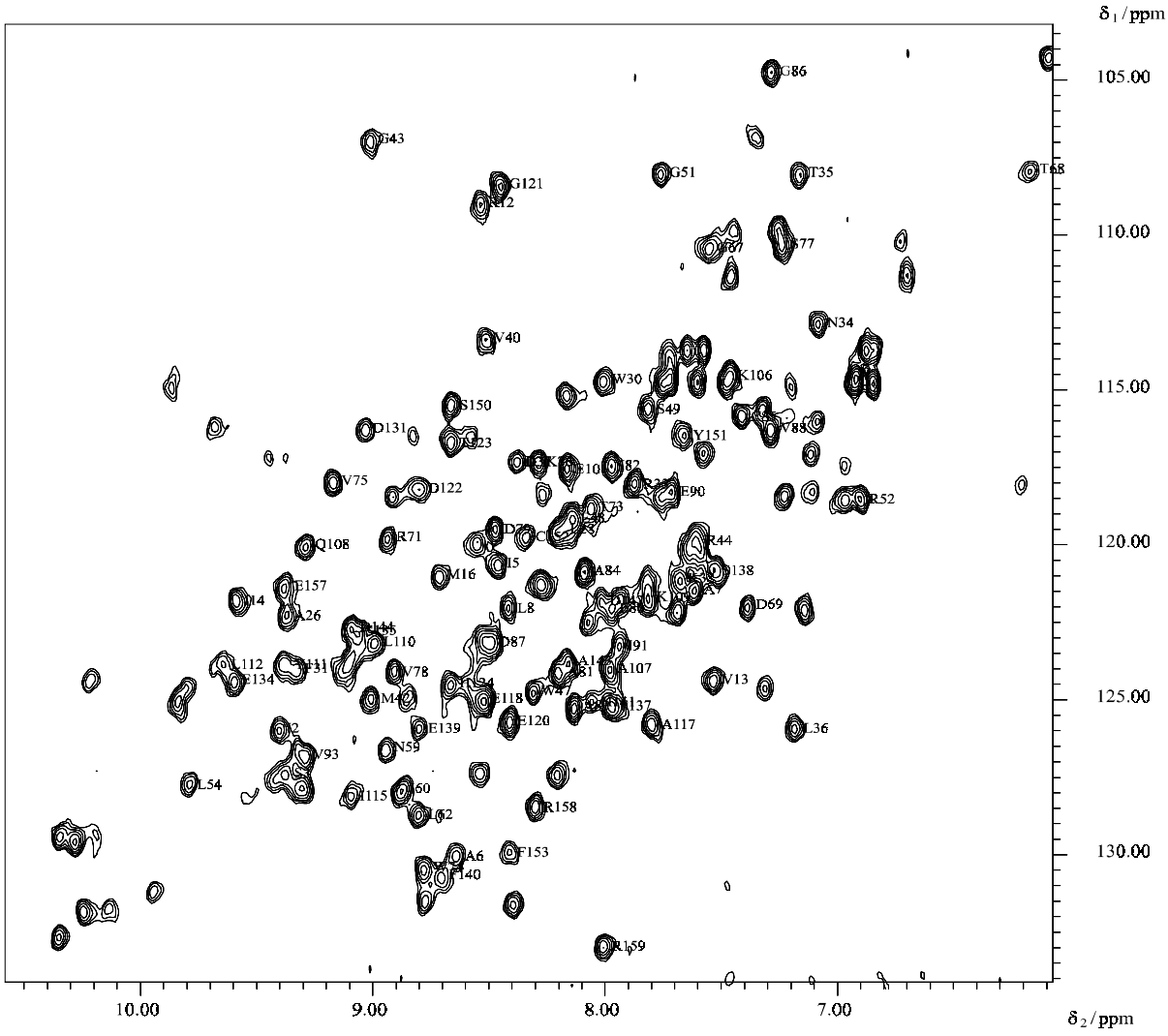

Abbildung 43: *^{15}N-HSQC-Zuordnung des NADPH-Komplexes von DHFR aus* E. coli.

6.9 Vergleich der Zuordnungen von NADPH- und Folat-Komplex

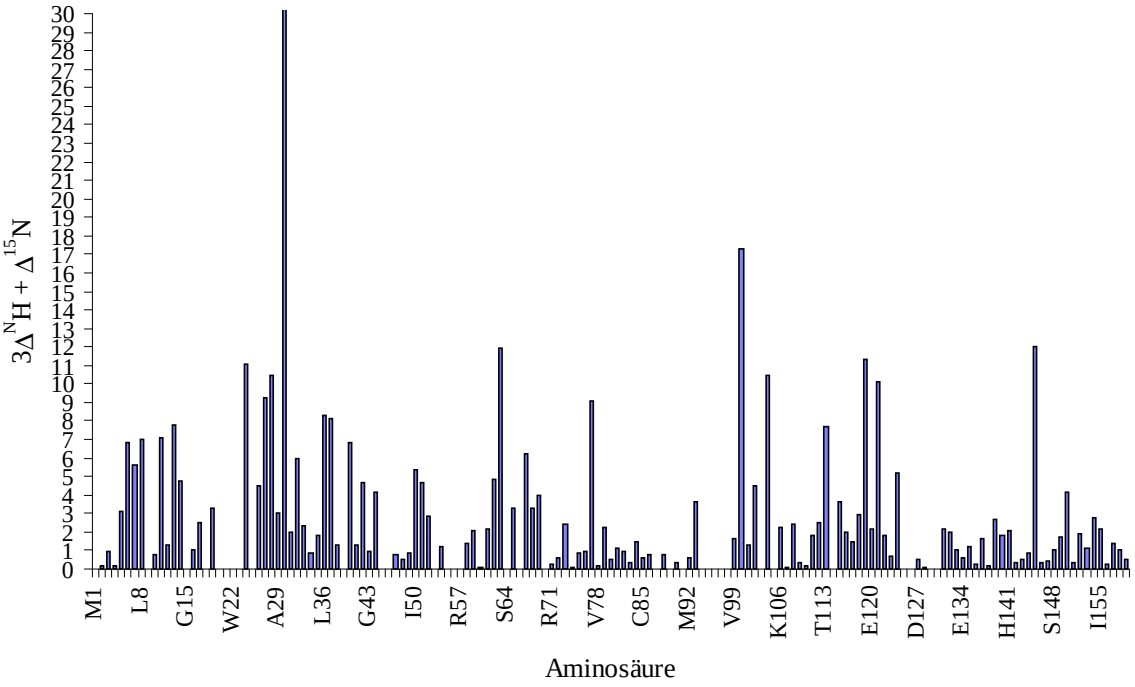

Abbildung 44: *$3\Delta H^N + \Delta^{15}N$ aus dem Vergleich der korrespondierenden Daten des NADPH- und des Folat-Komplexes.*

Abbildung 44 zeigt die Unterschiede der chemischen H^N- und ^{15}N-Verschiebungen aufgetragen nach Aminosäuren im Vergleich des NADPH- und des Folat-Komplexes von DHFR nach der endgültigen Zuordnung. Die Werte für die H^N-Verschiebungen werden mit Wichtungsfaktor 3 (in Relation zu ^{15}N) korrigiert, um einen ausgeglichenen Einfluss der beiden Verschiebungstypen zu gewährleisten.

Die stärksten Verschiebungsunterschiede sind wiederum im Bindungsstellenbereich zwischen M16 und K32 zu verzeichnen. Da relativ große Teile des Moleküls durch die Ligandbindung beeinflusst werden, sind jedoch auch für die restlichen Bereiche des Proteins starke Verschiebungsänderungen zu beobachten. Veränderte Dynamikeigenschaften nach der Ligandbindung können ebenfalls für eine Änderung der chemischen Verschiebung außerhalb der Bindungsstelle verantwortlich sein [148]. In Simulationen wurde festgestellt, dass während der Katalyse vom Liganden abhängige konformationelle Änderungen in den Loopbereichen

14-24, 64-71 und 116-125 auftreten [139]. Der Verschiebungsvergleich bestätigt dieses Verhalten. In allen drei Schleifenbereichen zeigen sich erwartungsgemäß große Abweichungen. Ein Vergleich der chemischen Verschiebungen der beiden Komplexe anhand der Struktur ist in Abbildung 45 dargestellt. Blau markiert Bereiche mit geringen Verschiebungsänderungen, Purpur zeigt Bereiche mit starken Verschiebungsunterschieden. Die stärksten Änderungen sind zwar im Bereich der Bindungsstellen zu erkennen (in der Abbildung die Furche zwischen den beiden Helices in der oberen Hälfte des Moleküls), es sind jedoch auch Teile des Moleküls, die weiter entfernt von der Bindungsstelle liegen, von starken Verschiebungsänderungen betroffen. Eine eindeutige Korrelation zwischen Verschiebungsänderungen und Bindungsstelle [149], lässt sich bei DHFR nicht feststellen. Aufgrund der betroffenen Schleifenbereiche sind jedoch auch weitreichende Änderungen im Dynamikverhalten des gesamten Moleküls zu erwarten.

Der Befund bestätigt auch die deutlichen Änderungen, die beim Vergleich der HSQC-Spektren beobachtet werden, da etwa die Hälfte der Signale eine Gesamtabweichung von mehr als 1 ppm in der H^N- und ^{15}N-Dimension aufweist.

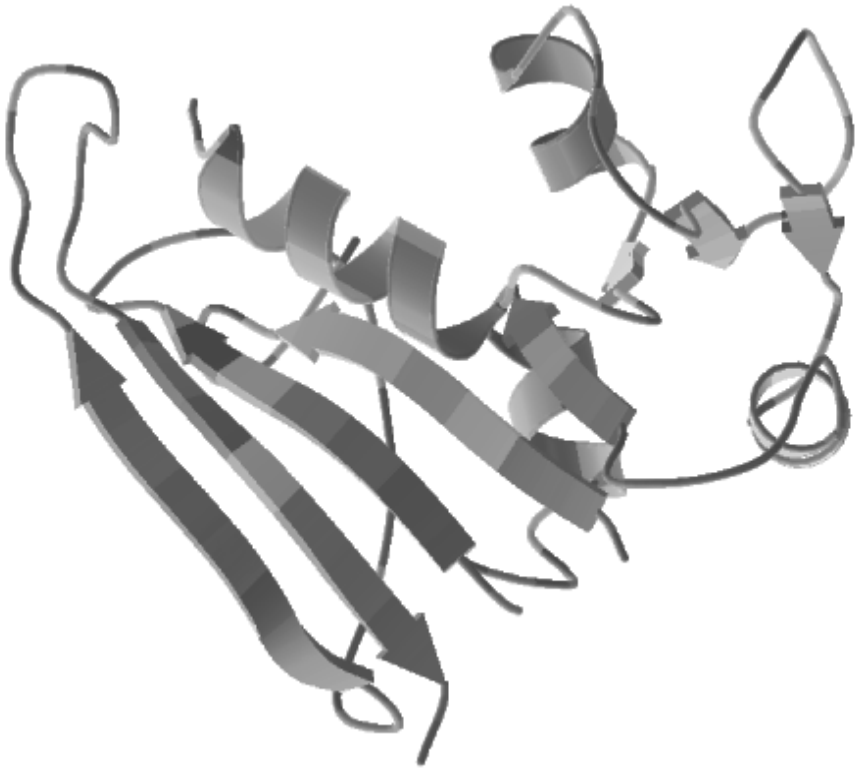

Abbildung 45: *Vergleich der chemischen Verschiebungen des Folat- bzw. des NADPH-Komplexes von DHFR anhand der Struktur. Blau markiert Bereiche mit geringen Verschiebungsänderungen, Purpur zeigt Bereiche mit starken Verschiebungsunterschieden.*

6.10 Vergleich mit der Literaturzuordnung des NADPH-Komplexes

Nach Abschluss der Arbeiten erschien eine Publikation zur Zuordnung des NADPH-Komplexes von DHFR aus *E. coli* [150]. Die Daten sind in der BMRB-Datenbank (www.bmrb.wisc.edu) unter BMRB-4554 abgelegt. Die Autoren beschreiben ebenfalls Probleme mit Probeninstabilitäten. Als Messstrategie führen die Autoren Spektrenaufnahmen bei 9°C und pH 7,6 in 70 mM Phosphatpuffer an. Aufgrund von starken Linienverbreiterungen verursacht durch die niedrigen Temperaturen wurde eine tripelmarkierte ^{13}C,^{15}N,^{2}H-Proteinprobe verwendet. Zur Zuordnung wurden folgende Spektren verwendet: HNCA, HNCACB, CBCA(CO)NH, HNCO, HN(CA)CO und HNH-NOESY. Die Zuordnung umfasst C^{α}, C^{β}, C', N und H^{N}. Eine Zuordnung der Protonen erfolgte aufgrund der benötigten Deuterierung nicht.

Für einen Vergleich der beiden Zuordnungen sind in Abbildung 46 und Abbildung 47 die Unterschiede der ^{15}N- und H^{N}-Verschiebungen aufgeführt. Im Diagramm der ^{15}N-Verschiebungsdifferenz (Abbildung 46) ist zusätzlich eine Analyse der abweichenden Zuordnungen gezeigt. Blaue Quadrate zeigen eine Übereinstimmung in allen zugeordneten Dimensionen für die entsprechende Aminosäure an. Große Differenzen bei blau markierten Aminosäuren weisen auf das Fehlen des ^{15}N-Verschiebungswerte in einer der beiden Zuordnungen hin. Rote Quadrate markieren abweichende Zuordnungen eines Signals: Das Signal aus den Literaturdaten ist zwar deutlich in den aufgenommenen Spektren zu erkennen, wurde jedoch nicht zugeordnet. Das stattdessen zugeordnete Signal fehlt in den Literaturdaten. Dies legt nahe, dass an den rot markierten Positionen möglicherweise Signale zugeordnet wurden, die auf die Probenveränderungen zurückzuführen sind. Gelbe Quadrate weisen auf Literaturzuordnungen hin, die nicht anhand der experimentellen Ergebnisse nachvollzogen werden können. An diesen Positionen ist kein Signal in den entsprechenden Spektren vorhanden. Grüne Quadrate zeigen Positionen an, die in den Spektren stark überlagert sind und für deren Auflösung weitere Informationen benötigt werden. Diese stehen jedoch aufgrund der genannten Probeninstabilitäten nicht zur Verfügung. Eine genauere Beurteilung ist für diese Positionen daher nicht möglich.

Der Vergleich zeigt eine weitgehende Übereinstimmung der beiden Zuordnungen. Vier Abweichungen sind eindeutig zu identifizieren. Eine (D27) befindet sich davon im

Hauptbereich der Bindungsstelle für NADPH. Die anderen Abweichungen sind ohne offensichtlichen strukturellen Zusammenhang über das Molekül verteilt.

Vergleichswerte mit mehr als 15 ppm für $\Delta^{15}N$ oder mehr als 5 ppm für ΔH^{N} zeigen Reste an, die in einer der beiden Zuordnungen fehlen und daher nicht verglichen werden können. Insgesamt konnten 15% der Reste nicht verglichen werden. Für 6% der Reste (alle farbig markierten Reste) wurde eine andere Zuordnung als in der Literatur angegeben gefunden. Für 79% kann eine Übereinstimmung festgestellt werden.

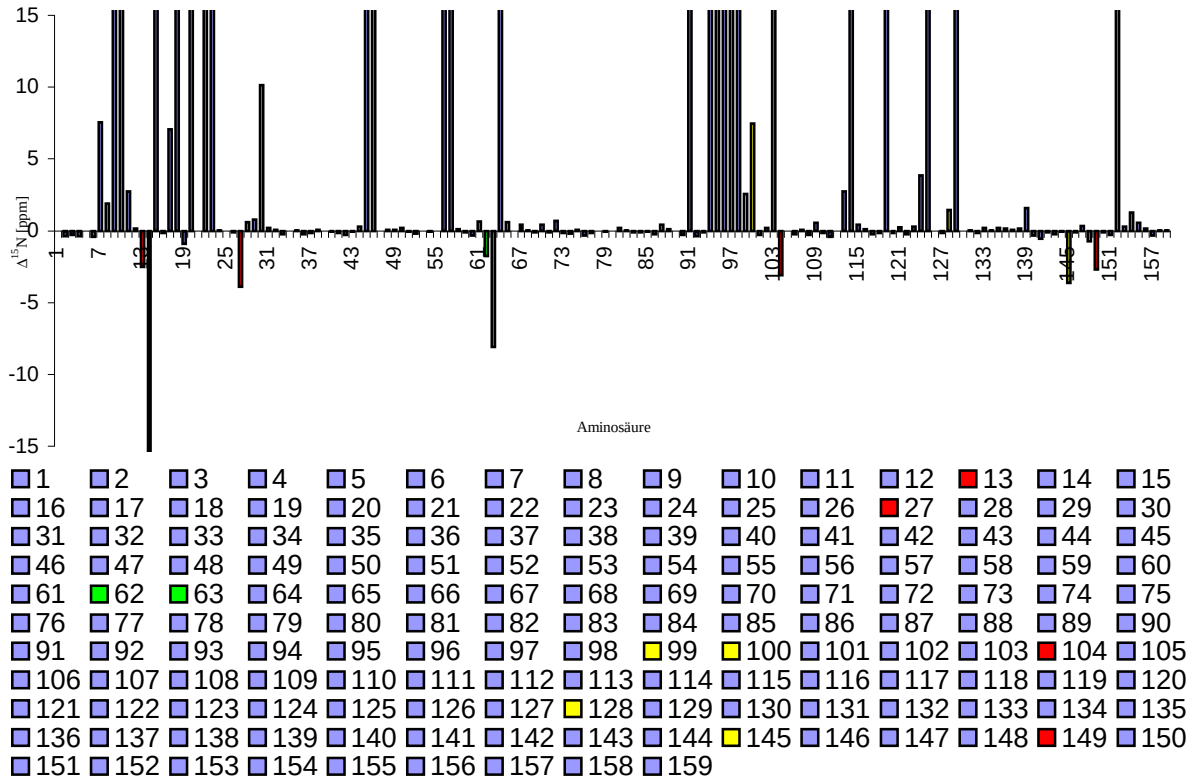

Abbildung 46: *Diagramm der ^{15}N-Verschiebungsdifferenz der experimentellen bzw. der Literaturzuordnung des NADPH-Komplexes von DHFR. Siehe Text zur Beschreibung des Farbcodes.*

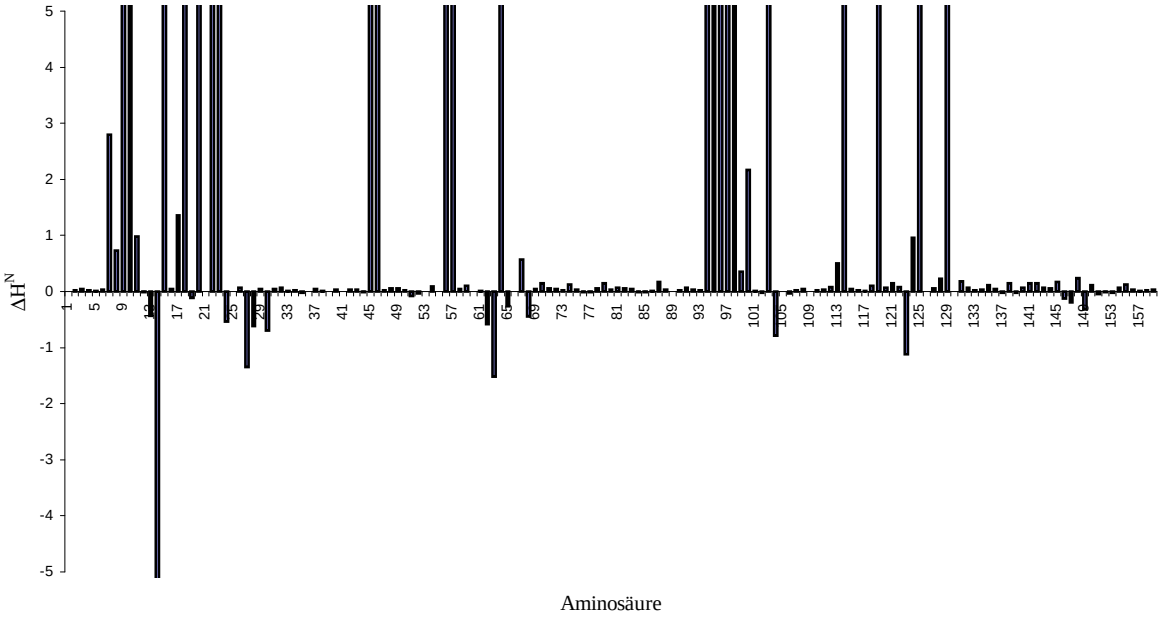

Abbildung 47: *Diagramm der H^N-Verschiebungsdifferenz zwischen der experimentellen bzw. der Literaturzuordnung des NADPH-Komplexes von DHFR.*

7 Zusammenfassung

In der vorliegenden Arbeit wurden Methoden zur automatisierten Zuordnung von heteronuklearen Protein-NMR-Spektren entwickelt und an zwei Proteinen eingesetzt und getestet.

Ausgehend von den Vorarbeiten von Michael Leutner [18, 19] wurde das Programm PASTA V3.0 erstellt. Zunächst wurde ein neuer Optimierungsalgorithmus (*Fast TA*) für die sequentielle Anordnung der Pseudoreste in der Zuordnungsliste geschrieben. Es handelt sich dabei um eine Weiterentwicklung des *threshold accepting*-Algorithmus, einer Methode zur kombinatorischen Minimierung. Der neue Algorithmus verfügt sowohl über eine höhere Rechengeschwindigkeit als auch eine bessere Konvergenz und Zuverlässigkeit. In der Praxis lässt sich damit in PASTA V3.0, im Vergleich zu vorherigen Versionen des Programms, eine durchschnittliche Steigerung der Rechengeschwindigkeit um ca. einen Faktor 50 erreichen. Die Rechenzeit für einen 180 Pseudoreste großen Datensatz beträgt auf einer SGI Octane R10000/180MHz ca. 2 min. Ebenso wurde für PASTA V3.0 eine *mapping*-Routine erstellt, mit deren Hilfe die Daten der optimierten Zuordnungslisten auf die Aminosäuresequenz des Proteins abgebildet werden können. Die Routine nutzt die Aminosäuren A, G, S, T und V, die mittels einfacher Kriterien aus den C^{α}- und C^{β}-Verschiebungen eines Pseudorestes bestimmt werden können, als Ankerpunkte, um die korrekte Sequenzposition zu ermitteln. Des weiteren wurde PASTA V3.0 mit einer grafischen Benutzeroberfläche ergänzt. Dazu wurde der bestehende Ansatz von einem sequentiellen Programmablauf auf eine *event*-basierte Verwaltung umstrukturiert.

In Anlehnung an die Konzepte von PASTA V3.0 wurde ein vollständig neues Programmpaket, PASTA Toolkit, erstellt. Dabei standen sowohl die Benutzerfreundlichkeit als auch die Strukturierung und Erweiterbarkeit für zukünftige Neuerungen im Vordergrund. Gemäß aktueller Entwicklungsstandards stellt PASTA Toolkit fünf eigenständige Module zur Rückgratzuordnung von Proteinen zur Verfügung:

- Multieingabefilter:

  Der Multieingabefilter ermöglicht das Einlesen von Tripelresonanz-NMR-Spektren und die automatische Gruppierung der Verschiebungswerte in Pseudoreste. Der Einlesefilter kann über die grafische Benutzeroberfläche für jedes Experiment frei konfiguriert werden.

- Optimierung:

  Das Optimierungsmodul ist für die richtige sequentielle Anordnung der Pseudoreste verantwortlich. Die Optimierung erfolgt mit dem in dieser Arbeit entwickelten *Fast TA*-Algorithmus unter Ausnutzung der sequentiellen Verschiebungsinformationen (i+1; i; i-1).

- Vergleichsmatrix:

  Die Vergleichsmatrix dient zur Extraktion der zuverlässigen Zuordnungen bzw. der Überprüfung der generellen Lösbarkeit eines Zuordnungsproblems aus einer Kombination der Ergebnisse von mehreren Optimierungsläufen eines Datensatzes.

- Aminosäureerkennung:

  Die Aminosäureerkennung liefert in einem zweistufigen Prozess über die Analyse der C^{α}- und C^{β}-Verschiebungen eines Pseudorestes Auskünfte über dessen Aminosäuretypus:

  - Identifikation der Aminosäuren A, G, S, T und V mit Verschiebungsmasken.
  - Bestimmung der fünf wahrscheinlichsten Aminosäuretypen durch Vergleich mit Verschiebungslisten (Sekundärstruktur-Berücksichtigung möglich).

- Sequenz-*mapping*:

  Dieses Modul bildet die Pseudoreste mit Hilfe einer Bewertungsfunktion *f(x,y)* auf die Aminosäuresequenz ab. *f(x,y)* entspricht dabei einem Wahrscheinlichkeitsmaß für die Zuordnung von Rest x auf Sequenzposition y. An der korrekten Sequenzposition ist die Summe über alle *f(x,y)* eines Zuordnungsfragments maximal.

Die einzelnen Module von PASTA Toolkit kommunizieren über eine zentrale ASCII-Datei, die sowohl zur Speicherung der Zuordnungsdaten als auch als Schnittstelle zu anderen Programmen dient. Die Datenverwaltung orientiert sich dabei am Konzept der relationalen Datenbanken. Auf diese Weise ist eine einfache Erweiterung und Anpassung des Programmpakets für die Zukunft gewährleistet. Alle Einzelmodule sind über eine grafische Benutzeroberfläche miteinander verknüpft.

PASTA V3.0 und PASTA Toolkit wurden in ANSI-C implementiert. Für die Programmierung der grafischen Benutzeroberflächen wurde das Tcl/Tk-Paket verwendet.

PASTA V3.0 und PASTA Toolkit wurden sorgfältig an realen Probedatensätzen getestet.

Beide Programme kamen auch bei der Zuordnung der N-terminalen Domäne des AAA-Proteins VAT zum Einsatz. VAT-N (183 Aminosäuren) verfügt zwischen G6-R48 und K49-T92 über eine interne Sequenzhomologie von 38%. Die Zuordnung der Rückgratresonanzen des Proteins wurde aus diesem Grund durch eine ungewöhnlich große Zahl von Signalüberlagerungen erschwert. Insbesondere mit Hilfe des Matrixmoduls von PASTA

Toolkit konnte schließlich die Rückgratzuordnung von 179 der 183 Aminosäuren des Proteins erreicht werden.

Die Struktur von VAT-N teilt sich in zwei Unterbereiche: M1-T92 und E93-E183. Die N-terminale Subdomäne VAT-Nn zeigt das Strukturmotiv eines *double-Ψ-barrels*. Dieses setzt sich aus den beiden sequenzhomologen Einheiten G6-R48 und K49-T92 zusammen, die über eine identische Struktur mit einem βαββ-Motiv verfügen. Die C-terminale Subdomäne zeigt eine neuartige *β-clam* Struktur bestehend aus 6 β-Strängen und einer α-Helix. Die relative Orientierung der beiden Domänen ist durch 31 spezifische Kontakte fixiert und wird durch elektronenmikroskopische Experimente zusätzlich bestätigt.

Für den NADPH-Komplex von DHFR aus *E. coli* wurde ein neuartiges Konzept zur vereinfachten Zuordnung von Protein-Komplexen entwickelt. Das Konzept setzt voraus, dass bereits die NMR-Zuordnung eines strukturhomologen Komplexes des Proteins bekannt ist.

Zuerst muss eine Zuordnungsbasis erzeugt werden. Dazu wurde ein HNHA-Spektrums der NADPH-Probe auf die Literaturzuordnung des Folat-Komplexes von DHFR mit einer Nachbarschaftssuche abgebildet. Anschließend erfolgte eine Verifikation der Zuordnungsbasis durch den Vergleich charakteristischer Signalspuren im HNH- und NNH-NOESY beider Komplexe. Alle Spektren benötigen lediglich eine [U-^{15}N]-markierte Probe.

Aufgrund von Instabilitäten der NADPH-Probe (Probenhaltbarkeit ca. 2 Tage) konnte jedoch für viele Signale keine eindeutige Zuordnung ermittelt werden. Daher wurde zusätzlich eine hochkonzentrierte (2,5mM) [U-^{13}C,^{15}N]-markierte Probe eingesetzt, die eine Reduktion der Messzeit für jedes Einzelexperiment auf ein Drittel des normalen Zeitbedarfs ermöglichte.

Damit konnte ein zersetzungsproduktfreies HNCACB Spektrum als Referenz aufgenommen werden. Mit Hilfe dieses Referenz-HNCACB, einem HNCA und einem HNCAHA konnte schließlich durch Überprüfen und Korrigieren der bisherigen Ergebnisse die endgültige Zuordnung erreicht werden.

Ein Vergleich mit den nach Abschluss des Projektes veröffentlichten Zuordnungsdaten [150] zeigt geringe Unterschiede der Ergebnisse: lediglich 6% der Reste weisen in den beiden Datensätzen eine abweichende Zuordnung auf.

8 Literatur

[1] F. J. Stevens, P. R. Pokkuluri, M. Schiffer, *Biochemistry 39* (2000) 15291-15296.

[2] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, J. D. Watson, *The Cell*, Garland Publishing, New York 1994.

[3] G. Taubes, *Science 271* (1996) 1493-1495.

[4] E. Mutschler, *Arzneimittelwirkungen*, WVG mbH, Stuttgart 1991.

[5] P. S. Lee, K. H. Lee, *Curr. Opin. Biotechnol. 11* (2000) 171-175.

[6] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, D. J. Lipman, *Nucl. Acid. Res. 25* (1997) 3389-3402.

[7] A. Müller, R. M. MacCallum, M. J. E. Sternberg, *J. Mol. Biol. 293* (1999) 1257-1271.

[8] M. Gerstein, R. Jansen, *Curr. Opin. Struct. Biol. 10* (2000) 574-584.

[9] M. J. Dutt, K. H. Lee, *Curr. Opin. Biotechnol. 11* (2000) 176-179.

[10] P. J. Hajduk, R. P. Meadows, S. W. Fesik, *Science 278* (1997) 497-499.

[11] J. Fejzo, C. A. Lepre, J. W. Peng, G. W. Bemis, M. A. Ajay Murcko, J. M. Moore, *Chem. Biol. 6* (1999) 755-769.

[12] C. S. Johnson Jr, *Progr. Nuclear Magn. Reson. Spectr. 34* (1999) 203.

[13] R. C. Stevens, *Curr. Op. Struct. Biol. 10* (2000) 558-563.

[14] P. J. Hajduk, T. Gerfin, J.-M. Boehlen, M. Häberli, D. Marek, S. W. Fesik, *J. Med. Chem. 42* (1999) 2315-2317.

[15] A. Ross, G. Schlotterbeck, W. Klaus, H. Senn, *J. Biomol. NMR 16* (2000) 139-146.

[16] P. D. Adams, R. W. Grosse-Kunstleve, *Curr. Opin. Struct. Biol. 10* (2000) 564-568.

[17] H. N. B. Moseley, G. T. Montelione, *Curr. Op. Struct. Biol. 9* (1999) 635-642.

[18] M. Leutner, *Inst. f. Org. Chemie u. Biochemie*, Technische Universität München, München 1997.

[19] M. Leutner, R. M. Gschwind, J. Liermann, C. Schwarz, G. Gemmecker, H. Kessler, *J. Biomol. NMR 11* (1998) 31-43.

[20] K. Wüthrich, *NMR of Proteins and Nucleic Acids*, John Wiley and Sons, New York 1986.

[21] M. Wittekind, L. Mueller, *J. Magn. Reson. Ser. B 101* (1993) 201-205.

[22] V. I. Bystrov, *Prog. Nucl. Magn. Reson. Spectrosc. 10* (1976) 41-81.

[23] D. M. LeMaster, *Annu. Rev. Biophys. Chem. 19* (1990) 243-266.

[24] P. E. Coughlin, F. E. Anderson, E. J. Oliver, J. M. Brown, S. W. Homans, S. Pollak, J. W. Lustbader, *J. Am. Chem. Soc. 121* (1999) 11871-11874.

[25] M. Salzmann, G. Wider, K. Pervushin, H. Senn, K. Wüthrich, *J. Am. Chem. Soc. 121* (1999) 844-848.

[26] G. Wider, K. Wüthrich, *Curr. Op. Struct. Biol. 9* (1999) 594-601.

[27] L. E. Kay, M. Ikura, G. Zhu, A. Bax, *J. Magn. Reson. 91* (1991) 422-428.

[28] V. Dötsch, G. Wagner, *Curr. Op. Struct. Biol. 8* (1998) 619-623.

[29] L. E. Kay, K. H. Gardner, *Curr. Op. Struct. Biol. 7* (1997) 722-731.

[30] G. M. Clore, A. M. Gronenborn, *Curr. Op. Chem. Biol. 2* (1998) 564-570.

[31] L. E. Kay, M. Ikura, R. Tschudin, A. Bax, *J. Magn. Reson. 89* (1990) 496-514.

[32] A. Bax, M. Ikura, *J. Biomol. NMR 1* (1991) 99-104.

[33] S. Grzesiek, A. Bax, *J. Am. Chem. Soc. 114* (1992) 6291-6293.

[34] G. W. Vuister, A. Bax, *J. Am. Chem. Soc. 115* (1993) 7772-7777.

[35] S. Seip, J. Balbach, H. Kessler, *J. Magn. Reson. 100* (1992) 406-410.

[36] R. T. Clubb, V. Thanabal, G. Wagner, *J. Biomol. NMR 2* (1992) 203-210.

[37] S. J. Archer, M. Ikura, D. A. Torchia, A. Bax, *J. Magn. Reson. 95* (1991) 636-641.

[38] M. J. Bottomley, M. J. Macias, Z. Liu, S. M., *J. Biomol. NMR 13* (1999) 381-385.

[39] M. Schubert, L. J. Ball, H. Oschkinat, P. Schmieder, *J. Biomol. NMR 17* (2000) 331-335.

[40] L. E. Kay, M. Ikura, A. Bax, *J. Am. Chem. Soc. 112* (1990) 888-889.

[41] G. T. Montelione, B. A. Lyons, S. D. Emerson, M. Tashiro, *J. Am. Chem. Soc. 114* (1992) 10974-10975.

[42] T. M. Logan, E. T. Olejniczak, R. X. Xu, S. W. Fesik, *FEBS Lett. 314* (1992) 413-418.

[43] M. Ikura, L. E. Kay, A. Bax, *J. Biomol. NMR 1* (1991) 299-304.

[44] D. Neri, T. Szyperski, G. Otting, H. Senn, K. Wüthrich, *Biochemistry 28* (1989) 7510-7516.

[45] F. Löhr, H. Rüterjans, *J. Magn. Reson. 9* (1997) 255-258.

[46] D. S. Wishart, B. D. Sykes, F. M. Richards, *Biochemistry 31* (1992) 1647-1651.

[47] M. Karplus, *J. Chem. Phys. 30* (1959) 11.

[48] J. N. S. Evans, *Biomolecular NMR Spectroscopy*, Oxford University Press, New York, USA 1995.

[49] B. Whitehead, C. J. Craven, J. P. Waltho, in R. D. G. (Ed.): *Protein NMR Techniques, Vol. 60*, Humana Press, Totowa, New Jersey 1997, p. 29-52.

[50] N. Tjandra, A. Bax, *Science 278* (1997) 1111-1114.

[51] M. R. Hansen, P. Hanson, A. Pardi, *Methods Enzymol. 317* (2000) 220-240.

[52] A. J. Wand, S. W. Englander, *Biochemistry 25* (1986) 1100-1106.

[53] D. E. Zimmerman, G. T. Montelione, *Curr. Opin. Struct. Biol. 5* (1995) 664-673.

[54] D. S. Wishart, C. G. Bigam, J. Yao, F. Abildgaard, H. J. Dyson, E. Oldfield, J. L. Markley, B. D. Sykes, *J. Biomol. NMR 6* (1995) 135-140.

[55] J. A. Lukin, A. P. Gove, S. N. Talukdar, C. Ho, *J. Biomol. NMR 9* (1997) 151-166.

[56] D. E. Zimmerman, C. A. Kulikowski, Y. Huang, W. Feng, M. Tashiro, S. Shimotakahara, C.-Y. Chien, R. Powers, G. T. Montelione, *J. Mol. Biol. 269* (1997) 592-610.

[57] D. W. Marquardt, *J. Soc. Ind. App. Mat. 11* (1963) 431-441.

[58] D. J. Gerth, T. Howell, S. I. Shupack, *Computers Chem. 16* (1992) 35.

[59] R. A. Caruna, R. B. Searle, S. I. Shupack, *Anal. Chem. 58* (1986) 1162.

[60] G. J. Kleywegt, R. M. Lamerichs, R. Boelens, R. Kaptein, *J. Magn. Reson. 85* (1989) 186-197.

[61] C. Antz, K.-P. Neidig, H. R. Kalbitzer, *J. Biomol. NMR 5* (1995) 287-296.

[62] A.-C. Schulte, A. Görler, C. Antz, K.-P. Neidig, H. R. Kalbitzer, *J. Magn. Reson. 129* (1997) 165-172.

[63] H. Gesmar, N. P. Faester, J. J. Led, *J. Magn. Reson. B 103* (1994) 10-18.

[64] R. Koradi, M. Billeter, M. Engeli, P. Güntert, K. Wüthrich, *J. Magn. Reson. 135* (1998) 288-297.

[65] C. Bartels, T.-h. Xia, M. Billeter, P. Güntert, K. Wüthrich, *J. Biomol. NMR 6* (1995) 1-10.

[66] C. Cieslar, M. Clore, A. Gronenborn, *J. Magn. Reson. 80* (1988) 119-127.

[67] S. Glaser, H. R. Kalbitzer, *J. Magn. Reson. 74* (1987) 450-463.

[68] D. Croft, J. Kemmink, K. P. Neidig, H. Oschkinat, *J. Biomol. NMR 10* (1997) 207-219.

[69] E. C. van Geerestein-Ujah, M. Mariani, H. Vis, R. Boelens, R. Kaptein, *Biopolymers 39* (1996) 387-405.

[70] N. E. Buchler, E. R. P. Zuiderweg, H. Wang, R. A. Goldstein, *J. Magn. Reson. 125* (1997) 34-42.

[71] C. Bartels, P. Güntert, M. Billeter, K. Wüthrich, *J. Comput. Chem. 18* (1997) 139-149.

[72] K.-B. Li, B. C. Sanctuary, *J. Chem. Inf. Comput. Sci. 37* (1997) 467-477.

[73] V. Dötsch, G. Wagner, *J. Magn. Reson. B 111* (1996) 310-313.

[74] V. Dötsch, K. Matsuo, G. Wagner, *J. Magn. Reson. B 112* (1996) 95-100.

[75] M. Schubert, M. Smalla, P. Schmieder, H. Oschkinat, *J. Magn. Reson. 141* (1999) 34-43.

[76] S. Grzesiek, A. Bax, *J. Biomol. NMR 3* (1993) 185-204.

[77] M. S. Friedrichs, L. Mueller, M. Wittekind, *J. Biomol. NMR 4* (1994) 703-726.

[78] R. P. Meadows, E. T. Olejniczak, S. W. Fesik, *J. Biomol. NMR 4* (1994) 79-96.

[79] H. S. Atreya, S. C. Sahu, K. V. Chary, G. Govil, *J. Biomol. NMR 17* (2000) 125-136.

[80] M. Iwadate, T. Asakura, M. P. Williamson, *J. Biomol. NMR 13* (1999) 199-211.

[81] K. Huang, M. Andrec, S. Heald, P. Blake, J. H. Prestegard, *J. Biomol. NMR 10* (1997) 45-52.

[82] J. L. Pons, M. A. Delsuc, *J. Biomol. NMR 15* (1999) 15-26.

[83] J. B. Olson Jr, J. L. Markley, *J. Biomol. NMR 4* (1994) 385-410.

[84] T. Szyperski, B. Baneck, D. Braun, R. W. Glaser, *J. Biomol. NMR 11* (1998) 387-405.

[85] W. Gronwald, L. Willard, T. Jellard, R. F. Boyko, K. Rajarathnam, D. S. Wishart, F. D. Sönnichsen, B. D. Sykes, *J. Biomol. NMR 12* (1998) 395-405.

[86] N. Morelle, B. Brutscher, J. P. Simorre, D. Marion, *J. Biomol. NMR 5* (1995) 154-160.

[87] P. Güntert, M. Salzmann, D. Braun, K. Wüthrich, *J. Biomol. NMR 18* (2000) 129-137.

[88] J. Xu, S. K. Straus, B. C. Sanctuary, L. Trimble, *J. Magn. Reson. 103* (1994) 53-58.

[89] L. Willard, T. Jellard, , PENCE, University of Alberta, Canada, Alberta, Canada 1994.

[90] D. S. Wishart, B. D. Sykes, *J. Biomol. NMR 4* (1994) 171-180.

[91] D. T. Jones, *J. Mol. Biol. 292* (1999) 195-202.

[92] W. Y. Choy, B. C. Sanctuary, *J. Chem. Inf. Comput. Sci. 37* (1997) 1086-1094.

[93] J. A. Cuff, G. J. Barton, *Proteins 40* (2000) 502-511.

[94] B. Rost, *Meth. in Enzym. 266* (1996) 525-539.

[95] M. Nilges, M. J. Macias, S. O'Donoghue, H. Oschkinat, *J. Mol. Biol. 269* (1997) 408-422.

[96] C. Mumenthaler, P. Güntert, W. Braun, K. Wüthrich, *J. Biomol. NMR 10* (1997) 351-362.

[97] P. Güntert, C. Mumenthaler, K. Wüthrich, *J. Mol. Biol. 273* (1997) 283-298.

[98] Y. Xu, J. Wu, D. Gorenstein, W. Braun, *J. Magn. Reson. 136* (1999) 76-85.

[99] C. Bartels, M. Billeter, P. Güntert, K. Wüthrich, *J. Biomol. NMR 7* (1996) 207-213.

[100] A. Görler, W. Gronwald, K.-P. Neidig, H. R. Kalbitzer, *J. Magn. Reson. 137* (1999) 39-45.

[101] G. Cornilescu, F. Delaglio, A. Bax, *J. Biomol. NMR 13* (1999) 289-302.

[102] R. Tejero, D. Monleon, B. Celda, R. Powers, G. T. Montelione, *J. Biomol. NMR 15* (1999) 251-264.

[103] D. A. Perlman, *J. Biomol. NMR 13* (1999) 325-335.

[104] G. M. Crippen, *J. Comp. Physiol. 24* (1977) 96-107.

[105] W. Braun, C. Bösch, L. R. Brown, N. Go, K. Wüthrich, *Biochim. Biophys. Acta 667* (1981) 377-396.

[106] T. F. Havel, K. Wüthrich, *J. Mol. Biol. 182* (1985) 281-294.

[107] M. Nilges, G. M. Clore, A. M. Gronenborn, *FEBS Lett. 239* (1988) 129-136.

[108] A. T. Brünger, *X-PLOR Version 3.1*, Yale University Press, New Haven 1992.

[109] P. Güntert, W. Braun, K. Wüthrich, *J. Mol. Biol. 217* (1991) 517-530.

[110] W. van Gunsteren, H. J. C. Berendsen, *Angew. Chem. 102* (1990) 1020-1055.

[111] F. Glover, *Modern heuristic techniques for combinatorial problems.*, Blackwell Scientific Publications, Oxford, UK 1993.

[112] G. Dueck, T. Scheuer, *J. Comp. Physics 90* (1990) 161-175.

[113] K. P. Neidig, M. Geyer, A. Görler, C. Antz, R. Saffrich, W. Beneicke, H. R. Kalbitzer, *J. Biomol. NMR 6* (1995) 255-270.

[114] J. K. Ousterhout, *Tcl and the TK Toolkit.*, Addison-Wesley, Reading, USA 1994.

[115] P. Domaille, *XIX ICMRBS* (Florenz, Italien), CCPN - A Collaborative Computing Project for NMR Spectroscopy (2000).

[116] J. P. Linge, M. Nilges, L. Ehrlich, *J. Biomol. NMR 15* (1999) 169-172.

[117] S. Grzesiek, J. Anglister, H. Ren, A. Bax, *J. Am. Chem. Soc. 115* (1993) 4369-4370.

[118] M. Huenges, C. Rölz, R. Gschwind, R. Peteranderl, F. Berglechner, G. Richter, A. Bacher, H. Kessler, G. Gemmecker, *EMBO J. 17* (1998) 4092-4100.

[119] A. Kühlewein, G. Voll, B. Schellbert, H. Kessler, G. Fischer, J. Rahfeld, G. Gemmecker, *Manuskript in Vorbereitung* .

[120] V. Truffault, M. Coles, T. Diercks, K. Abelmann, S. Eberhardt, H. Lüttgen, A. Bacher, H. Kessler, *Manuskript in Vorbereitung* .

[121] M. Coles, T. Diercks, B. Muehlenweg, S. Bartsch, V. Zölzer, H. Tschesche, H. Kessler, *J. Mol. Biol. 289* (1999) 139-157.

[122] M. Coles, T. Diercks, J. Liermann, A. Gröger, B. Rockel, W. Baumeister, K. K. Koretke, A. Lupas, J. Peters, H. Kessler, *Curr. Biol. 9* (1999) 1158-1168.

[123] G. Matthiessen, M. Unterstein, *Relationale Datenbanken und SQL*, Addison-Wesley, Bonn 1997.

[124] A. Beyer, *Prot. Sci. 6* (1997) 2043-2058.

[125] S. Patel, M. Latterich, *Trends Cell Biol. 8* (1998) 65-71.

[126] J. M. Peters, J. R. Harris, A. Lustig, S. Müller, A. Engel, S. Volker, W. W. Franke, *J. Mol. Biol. 223* (1992) 557-571.

[127] K. U. Fröhlich, H. W. Fries, J. M. Peters, D. Mecke, *Biochim. Biophys. Acta 1253* (1995) .

[128] F. Weber, F. Keppel, C. Georgopoulos, M. K. Hayer-Hartl, F. U. Hartl, *Nat. Struct. Biol. 5* (1998) 977-985.

[129] R. Golbik, A. N. Lupas, K. K. Koretke, W. Baumeister, J. Peters, *Biol. Chem. 380* (1999) 1049-1062.

[130] V. Pamnani, T. Tamura, A. Lupas, J. Peters, Z. Cejka, W. Ashraf , W. Baumeister, *FEBS Lett. 404* (1997) 263-268.

[131] T. Diercks, *Institut f. Org. Chem. u. Biochem.*, Technische Universität München, München 1999.

[132] A. L. Morris, M. W. MacArthur, E. G. Hutchinson, J. M. Thornton, *Proteins 12* (1992) 345-364.

[133] R. A. Laskowski, M. W. MacArthur, D. S. Moss, J. M. Thornton, *J. Appl. Cryst. 26* (1993) 283-291.

[134] T. Diercks, M. Coles, H. Kessler, *J. Biomol. NMR 15* (1999) 177-180.

[135] S. Koide, W. Jahnke, P. E. Wright, *J. Biomol. NMR 6* (1995) 306-312.

[136] R. M. Castillo, K. Mizuguchi, V. Dhanaraj, A. Albert, T. L. Blundell, A. G. Murzin, *Structure Fold Des. 7* (1999) 227-236.

[137] B. Rockel, J. Walz, R. Hegerl, J. Peters, D. Typke, W. Baumeister, *FEBS Lett. 451* (1999) 27-32.

[138] C. Bystroff, S. J. Oatley, J. Kraut, *Biochemistry 29* (1990) .

[139] J. L. Radkiewicz, C. L. Brooks, *J. Am. Chem. Soc. 122* (2000) 225-231.

[140] C. A. Fierke, K. A. Johnson, S. J. Benkovic, *Biochemistry 26* (1987) 4085-4092.

[141] C. J. Falzone, S. J. Benkovic , P. E. Wright, *Biochemistry 29* (1990) 9667-9677.

[142] F.-Y. Huang, Q.-X. Yang, T.-H. Huang, L. Gelbaum, L. F. Kuyper, *FEBS Lett. 283* (1991) 44-46.

[143] S. B. Shuker, P. J. Hajduk, R. P. Meadows, S. W. Fesik, *Science 274* (1996) 1531-1534.

[144] C. J. Falzone, J. Cavanagh, M. Cowart, A. G. Palmer, C. R. Matthews, S. J. Benkovic, P. E. Wright, *J. Biomol. NMR 4* (1994) 349-366.

[145] H. Lüttgen, *Institut f. Org. Chemie u. Biochemie*, Technische Universität München, München 1999.

[146] C. J. Falzone, P. E. Wright, S. J. Benkovic, *Biochemistry 30* (1991) 2184-2191.

[147] M.-C. Hsu, Y. Ho, F.-Y. Huang, *J. Chinese Chem. Soc. 45* (1998) 115-121.

[148] M. P. Foster, D. S. Wuttke, K. R. Clemens, W. Jahnke, I. Radhakrishnan, L. Tennant, M. Reymond, J. Chung, P. E. Wright, *J. Biomol. NMR 12* (1998) 51-71.

[149] B. T. Farmer, K. L. Constantine, V. Goldfarb, M. S. Friedrichs, M. Wittekind, J. Yanchunas, J. G. Robertson, L. Mueller, *Nature Struct. Biol. 3* (1996) 995-997.

[150] E. Zaborowski, J. Chung, G. Kroon, H. J. Dyson, P. E. Wright, *J. Biomol. NMR 16* (2000) 349-350.

9 Anhang

9.1 Fileformate

9.1.1 Optimierung

```
#PASTA OPTIMIZATION INPUT FILE

#threshold = 150
#steps = 10000
#input_file = /test_data/nusb.list
#output_file = /test_data/nusb_opt.list
#multiple = 3

#name1 = HA
#tol1 = 0.03
#score1 = -10
#layer1 = i-1

#name2 = CA
#tol2 = 0.3
#score2 = -12
#layer2 = i-1

#name3 = CB
#tol3 = 0.3
#score3 = -12
#layer3 = i-1
```

Fileidentifikation:

#PASTA OPTIMIZATION INPUT FILE

Erklärung der einzelnen Parameter:

Threshold: Toleranzschwelle für den *Threshold Accepting* Algorithmus

Steps: Anzahl der Schritte bis zur Erniedrigung der Toleranzschwelle für den *Threshold Accepting* Algorithmus

Input_file: Filename der zu optimierenden Pseudorest-Liste

Output_file: Filename unter dem die optimierte Pseudorest-Liste abgespeichert wird

Multiple: Anzahl der Optimierungsdurchläufe für eine mehrfache Optimierung der gleichen Liste. Der Outputfilename erhält automatische eine Zahl entsprechend des Optimierungslaufes als Anhängsel.

Die folgenden Parameter müssen für jeden Verschiebungstyp der Optimierung einzeln angegeben werden. An jeden Parameternamen muss eine Nummer zur Identifikation aller Daten eines Verschiebungstyps angefügt werden. Die Nummern werden von 1 aufsteigend vergeben.

Name: Name des Verschiebungstyps

Tol: Toleranz für den Vergleich der sequentiellen Verschiebungswertes

Score: Bewertung für die Pseudoenergiefunktion

Layer: Art der sequentiellen Information über die optimiert werden soll.

Zulässige Werte: i-1

i+1

i-1, i+1

9.1.2 Multi-Eingabefilter

Eingabefile

```
#EXPERIMENT FILTER INPUT FILE

#signal_name = CA CAi-1 CB CBi-1
#signal_tol = 0.3
#signal_dim = 2
#ref_dim1 = 1
#ref_name1 = N15
#ref_tol1 = 0.3
#ref_dim2 = 3
#ref_name2 = HN
#ref_tol2 = 0.03
#low_intensity = CAi-1 CBi-1
#pos_phase = CB CBi-1
#data_format = AURELIA
#experiment_name = HNCACB
#peak_file = /test_data/hncacb.pks
#list_file = /test_data/nusb.list
#save_file = /test_data/nusb1.list
```

Fileidentifikation:

#EXPERIMENT FILTER INPUT FILE

Erklärung der einzelnen Parameter:

Signal_name: Bezeichnung der einzulesenden Kerne in der Signaldimension

Signal_tol: Toleranz für das Einlesen in der Signaldimension

Signal_dim: Nummer der Signaldimension im Experiment

Ref_dim1: Dimensionsnummer der 1. Referenzverschiebung im Experiment

Ref_name1: Bezeichnung der Kerne in der Referenzdimension 1

Ref_tol: Toleranz beim Vergleich der Referenzdaten mit bestehenden Daten

Low_intensity: Kerne mit niedriger Verschiebung (Intensitätsvergleich)

Pos_phase: Kerne mit positiven Signalen (Phasenvergleich)

Data_format: Datenformat der Peakliste

Experiment_name: Experimentname

Peak_file: Name des Peaklistenfiles

List_file: Name der Pseudorest-Liste, zu der die Experimentdaten hinzugefügt werden sollen

Save_file: Name, unter dem die neue Pseudorest-Liste gespeichert werden soll

Reportfile

Beim Einlesen eines Experimentes wird ein Reportfile erzeugt.

Die Einträge sind entsprechend der Pseudoreste der verwendeten Zuordnungsliste geordnet.

Kopfzeile:

#PASTA PEAK REPORT FILE: Experimentname

Format für einen Pseudorest:

Kopzeile:

Residue no.: Aktuelle PASTA-Nummer der Pseudorests

Orig.no.: Originalnummer der Pseudorests

Peakzeile (Einzelelemente durch tab getrennt):

Peaknummer: Nummer des Peaks aus der Peakliste

Chemische Verschiebung: je nach Anzahl der Dimensionen im Experiment

Intensität: Peakintensität aus der Peakliste

Kommentar:

--XX--: Information XX zu diesem Rest kann nicht in der Peakliste gefunden werden.

++XX++: Mehr als ein Signal der Peakliste kann auf diese Position zugeordnet werden. Keine automatische Entscheidung möglich.

9.1.3 Pseudorest-Liste

```
#PASTA residue list

#PASTA no.: 1     Pseudo energy: 130      Sequence: 0 Secondary: RC
NAA   N15   123.650002  (6)
NAA   NH    8.790000    (6)
NAA   HA    3.740000    (6)
NAA   HAi-1 4.210000    (6)
NAA   CA    57.580002   (6)
NAA   CAi-1 54.520000   (6)
NAA   CB    41.169998   (6)
NAA   CBi-1 41.070000   (6)
NAA   COi-1 178.809998  (6)
//#ueberlagert mit NAA 7
//#Leu 74
//#
//#
//#ACIDS: Asp Leu Asn Phe Tyr
//#ACIDSi-1: Asp Leu Asn Phe Tyr
```

Fileidentifikation:

#PASTA residue list

Zeilenformat (einzelne Elemente durch *tab* oder *space* getrennt):

Kopfzeile:

#:	Kennzeichnet Kopfzeile
PASTA No.:	momentane Position in der Pseudorestliste
Pseudo energy:	Pseudoenergie des Reste nach der letzten Optimierung
Sequence:	zugeordnete Sequenzposition
Secondary:	Sekundästrukturelement: RC (random coil), AH (α-Helix), BS (β-Faltblatt)

Verschiebungszeile:

Aminosäurebezeichnung (max. 5 Zeichen)

Kernbezeichnung (max. 5 Zeichen)

Verschiebungswert

Originalnummer (in runden Klammern)

Kommentarzeile:

// :	Beginn einer Kommentarzeile
#ACIDS:	Eintrag der automatischen Aminosäureerkennung
#ACIDSi-1:	Eintrag der automatischen Aminosäureerkennung

9.1.4 Aminosäuresequenz

```
#PASTA SEQUENCE FILE
#lettercode = 1

#MKPAARRRARECAVQALYSWQLSQNDIADVEYQFLAEQDV
#KDVDVLYFRELLAGVATNTAYLDGLMKPYLSRLLEELGQV
#ELAVLRIALYELSLRSDVPYKVAINEAIELAKSFGAEDSH
#KFVNGVLDKAAPVIRPNKK
```

Fileidentifiaktion:

#PASTA SAEQUENCE FILE

Parameter:

#lettercode: 1 für Einbuchstabencode, 3 für Dreibuchstabencode

Jede weitere Zeile der Sequenz muss mit # beginnen. Die Reihenfolge der Aminosäuren muss vom N-Terminus aufsteigend sein. Zahlen, Leer- und Tabulatorzeichen werden ignoriert. D.h. auch folgende Formate sind zulässig:

```
#1 Ala
#2 Gly
#3 Asn
```

9.1.5 Vergleichsmatrix

```
#PASTA Comparison Matrix
#Size: 7

*************************
#301 301 0
#301 89 1
#301 162 0
#301 53 0
#301 76 0
#301 83 0
#301 136 0
```

Fileidentifikation:

#PASTA Comparison Matrix

Parameter:

#Size: Gibt die Größe der Matrix an. Die Matrix ist grundsätzlich quadratisch.

Zeilenformat (einzelne Elemente durch *tab* oder *space* getrennt):

#:	kennzeichnet Datenzeile
Zeilennummer:	Originalnummer des entsprechenden Restes
Spaltennummer:	Originalnummer des entsprechenden Restes
Bewertung:	Anzahl der Fälle in denen diese beiden Reste benachbart sind.

9.1.6 Peaklisten

Folgende Peaklistenformate werden unterstüzt:

AURELIA:	(normales Fileformat, kein Userpeaklist-Format
TRIAD:	nur als ASCII-Format
XEASY:	testweise Unterstützung

9.1.7 Grafische Benutzeroberfläche

p_opt.def

```
N15 0.3 -10
HN 0.03 -20
HA 0.03 -10
HB 0.03 -10
CA 0.3 -12
CB 0.3 -12
CO 0.3 –15
```

Beschreibung: Parameter des Optimierungsmodul

Zeilenformat (einzelne Elemente durch *tab* oder *space* getrennt):

Kernbezeichnung (ohne i-1, i+1)

Toleranz

Bewertung in der Pseudoenergiedefinition

f_signal.def

```
N15 0.30
HN 0.030
HA 0.030
HAi-1 0.030
HB 0.030
HBi-1 0.030
CA 0.30
CAi-1 0.30
CB 0.30
CBi-1 0.30
CO 0.30
COi-1 0.30
```

Beschreibung: Parameter des Multieingabefilters

Zeilenformat (einzelne Elemente durch *tab* oder *space* getrennt):

Kernbezeichnung (i-1, i+1 müssen gesondert als eigene Zeile eingetragen werden)

Toleranz

F_preset.def

```
HNCACB
N15
INPUT
HN
--
CA
CAi-1
CB
CBi-1
prev_cmp
phase_cmp
CA
CAi-1
--
--
intens_cmp
CAi-1
CBi-1
--
--
```

Beschreibung: Experiment-Datenbank des Multieingabefilters, wird von der grafischen Benutzeroberfläche automatisch generiert.

Jeder Eintrag wird in eine gesonderte Zeile geschrieben:

Experimentname

Bezeichnung Dimension (maximal 4 Dimensionen)

Erwartete Signale der Inputdimension (maximal 4 Signale)

Prev_cmp (Abgleich mit vorhandenen Daten der Pseudorestliste)

Phase_cmp(Phasenverschiedene Signale)

Signale mit positiver Phase (nur bei phase_cmp relevant, maximal 4)

Intens_cmp (Intensitätsvergleich der Signale)

Signale mit niedriger Intensität (nur bei intens_cmp relevant, maximal 4)

~/.pasta

Beschreibung: Konfigurationsfile, wird automatisch im Heimatverzeichnis angelegt.

Parameter:

#WORDKING_DIR:	Arbeitsverzeichnis des Benutzers
#TEMPLATES:	Definition der Spreadsheetparameter

9.2 Befehlsbeschreibungen PASTA Toolkit

- **acid_assign.exe [-Optionen] Pseudorestliste**

  Optionen:

  -m: Zuordnung der Aminosäuren für die [i-1]-Verschiebungen.

  -o: Alte Zuordnungen in der Pseudorestliste nicht überschreiben.

  -s: Speichern des Ergebnisses in einer Pseudorestliste. Erfordert als Parameter die Angabe des Dateinamens.

  -t: Textausgabe in das Konsolenfenster.

- **list_cmp.exe [-Optionen] Pseudorestliste [weitere Pseudorestlisten]**

  Optionen:

  -o filename: Altes Matrixfile laden

  -s filename: Matrixfile abspeichern

  -t: Textausgabe in das Konsolenfenster

- **multi_input.exe [Eingabefile]** (Standardeingabefile: ../input_files/multi.input)

- **optimize.exe [-Optionen] [Eingabefile]** (Standardeingabefile: optimize.inp)

  Optionen:

  -t: Textausgabe im Konsolenfenster

- **sequence_map.exe [-Optionen] Sequenzfile Pseudorestliste**

  Optionen:

  -c: Top3-Sequenzpositionen in Kommentarzeile speichern

  -m filename: Matrix-File laden

  -s filename: Abgebildete Pseudorestliste speichern

  -t: Textausgabe in das Konsolenfenster

9.3 Datentypen

```
/*
##############################################
*/

typedef struct PEAK {

      struct PEAK *prev;      /* Zeiger auf vorhergendes Listenelement */
      struct PEAK *next;      /* Zeiger auf naechstes Listenelement */
      int peak_no;            /* Peak-Nummer*/
      int intensity;          /* Intensitaet*/
      float shift[4];         /* maximal 4 Shiftwerte*/
      char name[50];          /* Peak-Bezeichnung */

} PEAK;

/*
##############################################
*/

typedef struct RESIDUE {

      struct RESIDUE *prev;   /* Zeiger auf vorhergendes Listenelement */
      struct RESIDUE *next;   /* Zeiger auf naechstes Listenelement */
      char name[6];           /* Bezeichnung fuer die Aminosaeure */
      int pasta_no;           /* Nummer der momentanen PASTA-Position */
      int orig_no;            /* Nummer, die das Residue beim erzeugen der
                              Startliste erhaelt */
      int seq_pos;            /* Sequenzposition des Residues */
      int energy;             /* Pseudoenergie aus der Minimierung */
      int misc;               /* Integerwert, der fuer verschiedene
                              Anwendungen zur Verfuegung steht */
      int secondary_struct;   /* Sekundaerstrukturelement: 0 random coil, 1
                              a-helix, 2 b-sheet*/
      int im1_no;             /* Anzahl der moeglichen i-1 Nachfolger*/
      int ip1_no;             /* Anzahl der moeglichen i+1 Nachfolger*/
      float *opt_shifts[3];   /* Zeiger auf die Verschiebungen, die fuer die
                              Optimierung ausgewaehlt wurden, 0:i, 1:i-1,
                              2:i+1*/
      struct RESIDUE **im1_list;    /* Zeiger auf die Liste der moeglichen
                                    i-1 Nachfolger*/
      struct RESIDUE **ip1_list;    /* Zeiger auf die Liste der moeglichen
                                    i+1 Nachfolger*/
      struct COMMENT *comments;     /* Zeiger auf Kommentarzeilen */
      struct ATOM *atoms;           /* Zeiger auf die zugehoerigen Atom
                                    einer bestehenden Struktur */
      struct SHIFT *shifts;         /* Zeiger auf den ersten Shiftwert */

} RESIDUE;

/*
##############################################
*/

typedef struct SHIFT {

      struct SHIFT *next;     /* Zeiger auf naechstes Listenelement */
      float value;            /* Shiftwert */
      char type[10];          /* Identifikationsname */

} SHIFT;
```

```
/*
##############################################
*/

typedef struct COMMENT {

      struct COMMENT *next;   /* Zeiger auf naechstes Listenelement */
      char string[256];       /* Textzeile fuer Kommentar */

} COMMENT;

/*
##############################################
*/

typedef struct ACID {

      struct ACID *prev;      /* Zeiger auf vorhergehendes Listenelement */
      struct ACID *next;      /* Zeiger auf naechstes Listenelement */
      int sequence_no;        /* Position in der Sequenz */
      int misc;               /* Integerwert fuer verschiedene Anwendungen
*/
      char name[6];           /* Name der Aminosaeure */
int secondary_struct;         /* Sekundaerstrukturelement: 0 random coil, 1
                              a-helix, 2 b-sheet*/

} ACID;

/*
##############################################
*/

typedef struct OPT_PARAM {

      int score;        /* Wert fuer die Pseudoenergie, bei 0
Standardwert*/
      int layer;        /* Zu optimierende Dimensionen: 1 (i-1), 2 (i+1), 3
                        (beide)*/
      float tolerance;  /* Toleranz fuer die Optimierung */
      char type[10];          /* Identifikationsname */
} OPT_PARAM;

/*
##############################################
*/
```

9.4 Zuordnungsliste von VAT-N

Rest	H^N ^{15}N	H^α C^α	H^β C^β	H^γ C^γ	H^δ C^δ	H^ε C^ε	C’
M1	- -	-/- -	-/- -	-/- -	-/- -	-/- -	-
E2	- -	4.64/- 56.87	2.37/2.37 30.68	-/- -	-/- -	-/- -	176.09
S3	8.56 116.97	4.68/- 58.38	4.08/4.08 64.53	-/- -	-/- -	-/- -	173.63
N4	8.29 125.91	4.74/- 55.17	3.00/3.00 41.3	-/- -	7.77/7.12 112.49	-/- -	175.74
N5	8.51 123.47	4.94/- 53.55	3.05/3.05 39.37	-/- -	7.75/7.75 112.47	-/- -	175.53
G6	8.47 108.03	4.47/4.18 45.15	-/- -	-/- -	-/- -	-/- -	173.37
I7	8.87 114.53	5.05/- 59.8	2.16/- 41.41	1.50/1.50 25.5	1.11/- 14.41	-/- -	173.59
I8	8.3 122.25	5.41/- 59.45	1.98/- 38.46	1.70/1.70 27.33	1.00/- 11.82	-/- -	176.65
L9	8.73 127.13	4.88/- 52.96	1.34/1.30 48.47	1.54/- 26.62	0.93/0.93 27.39	-/- -	174.75
R10	8.48 119	5.27/- 55.11	1.86/1.86 32.47	1.82/1.82 27.93	3.44/3.44 43.56	-/- -	176.49
V11	9.05 124.28	4.36/- 64.15	2.35/- 32.27	1.10/- 24.89	-/- -	-/- -	175.96
A12	9.3 108.02	4.93/- 50.49	1.55/- 23.52	-/- -	-/- -	-/- -	174.27
E13	8.61 118.29	4.46/- 56.55	2.15/2.15 30.68	2.50/2.50 36.16	-/- -	-/- -	176.13
A14	8.8 124.89	4.21/- 54.01	1.47/- 20.16	-/- -	-/- -	-/- -	176.65
N15	7.86 119.82	4.64/- 54.93	2.96/2.96 42	-/- -	-/- -	-/- -	175.82
S16	- -	-/- -	-/- -	-/- -	-/- -	-/- -	-
T17	- -	4.62/- 62.07	4.58/- 69.36	1.45/- 21.84	-/- -	-/- -	175.1
D18	8.07 122.66	4.92/- 55.15	3.10/2.87 41.77	-/- -	-/- -	-/- -	176.79
P19	- -	4.73/- 66.06	2.53/2.53 31.54	2.34/2.34 27.94	4.24/4.24 50.71	-/- -	177.6
G20	9.23 110.46	4.38/4.00 46.13	-/- -	-/- -	-/- -	-/- -	174.02
M21	7.93 116.97	4.87/- 54.91	2.42/2.42 33.27	-/- -	-/- -	2.26/- 17.67	175.79
S22	9.29 114.94	4.60/- 59.6	3.77?/3.77? 61.37	-/- -	-/- -	-/- -	173.84
R23	8.69 118.65	4.83/- 56.25	1.93/1.93 32.06	-/- -	-/- -	-/- -	175.83

Rest	H^N ^{15}N	H^α C^α	H^β C^β	H^γ C^γ	H^δ C^δ	H^ε C^ε	C’
V24	9.01 122.74	4.55/- 61.14	1.98/- 32.29	0.85/- 20.69	-/- -	-/- -	175.87
R25	8.14 124.69	5.12/- 56.18	1.99/1.60 30.98	1.84/1.84 27.47	-/- -	-/- -	175.65
L26	8.14 122.05	5.24/- 52.86	1.72/1.72 48.03	1.52/- 25.96	1.04/1.04 24.2	-/- -	176.43
D27	9.59 125.91	4.84/- 53.06	3.96/3.96 40.84	-/- -	-/- -	-/- -	175.35
E28	9.89 121.64	3.94/- 61.26	2.44/2.44 29.75	2.84/2.84 36.47	-/- -	-/- -	179.67
S29	9.29 115.95	4.33/- 62.1	4.20/4.20 62.79	-/- -	-/- -	-/- -	177.55
S30	9.16 118.49	4.33/- 63.12	3.75/3.75 63.34	-/- -	-/- -	-/- -	175.96
R31	8.52 120.02	3.96/- 62.04	2.38/2.38 29.06	2.40?/2.40? 29.16?	3.12/3.12 43.47	-/- -	179.19
R32	8.55 118.8	4.44/- 59.6	2.26/2.26 29.75	2.05/2.05 27.33	3.48/3.48 43.47	-/- -	180.66
L33	8.25 120.77	4.32/- 58.14	2.23/1.70 42.69	2.15/- 26.97	1.18/1.18 25.27	-/- -	179.32
L34	7.75 116.16	4.45/- 54.82	1.96/1.96 44.54	2.01/- 26.41	1.22/1.22 23.98	-/- -	176.58
D35	8.22 119.82	4.46/- 55.1	3.35/3.35 40.29	-/- -	-/- -	-/- -	174.79
A36	8.58 119.21	5.09/- 50.42	1.31/- 21.44	-/- -	-/- -	-/- -	176.17
E37	9.31 123.47	4.59/- 53.94	2.21/2.21 31.6	2.64/2.64 36.86	-/- -	-/- -	177.08
I38	8.76 121.23	3.59/- 63.31	2.03/- 36.22	1.72/1.72 27.94	0.97/- 11.8	-/- -	177.94
G39	9.22 117.17	4.70/4.70 45.25	-/- -	-/- -	-/- -	-/- -	174.92
D40	8.23 121.64	5.03/- 55.18	3.09/3.09 41.3	-/- -	-/- -	-/- -	174.53
V41	9.19 120.22	4.92/- 62.53	2.22/- 32.29	1.17/- 22.22	-/- -	-/- -	175.79
V42	8.69 115.14	5.47/- 57.65	2.21/- 34.84	0.80/- 18.24	-/- -	-/- -	174.15
E43	9.52 120.83	5.57/- 53.71	2.13/2.13 34.6	2.34/2.34 37.24	-/- -	-/- -	175.14
I44	8.82 124.59	5.19/- 59.88	1.68/- 40.38	1.62/1.62 27.94	0.95/- 15.45	-/- -	175.18
E45	9.34 125.91	5.46/- 54.82	2.20/1.95 35.07	2.26/2.26 36.05	-/- -	-/- -	176.48
K46	8.02 126.52	4.73/- 59.02	2.02?/2.02? 33.22	1.55/1.55 26.1	1.88/1.88 29.24	3.20/3.20 42.15	177.34
V47	8.44 122.05	4.29/- 64.91	2.28/- 33.08	1.40/- 21.93	-/- -	-/- -	176.09
R48	7.89 117.58	4.90/- 54.23	2.22/2.22 33.68	1.85/1.85 27.68	3.64/3.64 43.17	-/- -	174.75
K49	8.25 119.41	5.49/- 55.7	1.74/1.74 35.84	1.37/1.37 25.26	1.79/1.79 29.51	3.12/3.12 42.16	174.88

Rest	H^N ^{15}N	H^α C^α	H^β C^β	H^γ C^γ	H^δ C^δ	H^ε C^ε	C’
T50	8.7 113.01	4.98/- 61.26	4.10/- 69.49	1.33/- 18.35	-/- -	-/- -	172.81
V51	7.9 111.98	6.13/- 58.33	2.09/- 36.92	1.02/- 22.02	-/- -	-/- -	174.75
G52	8.56 129.16	4.49/3.90 45.93	-/- -	-/- -	-/- -	-/- -	171.08
R53	8.77 118.6	5.54/- 55.73	2.09/2.09 32.81	1.91/1.91 29.16	3.61/3.61 42.86	-/- -	176.04
V54	9.35 123.46	4.42/- 64.54	2.32/- 32.58	1.13/- 24.74	-/- -	-/- -	175.53
Y55	9.47 130.59	4.81/- 57.06	3.17/2.71 42.69	-/- -	-/- -	-/- -	173.97
R56	8.4 121.24	4.39/- 57.44	2.00/2.00 31.37	-/- -	3.44?/3.44? -	-/- -	176.65
A57	7.72 126.32	4.66/- 51.89	1.52/- 18.71	-/- -	-/- -	-/- -	177.85
R58	9.3 121.85	4.66/- 55.4	2.34/1.89 28.9	-/- -	-/- -	-/- -	176.24
P59	- -	4.65/- 65.99	2.65/2.18 31.89	2.46/2.46 27.72	4.11/4.11 50.1	-/- -	179.93
E60	9.8 116.36	4.40/- 58.82	2.25/2.25 28.37	2.50/2.50 36.1	-/- -	-/- -	176.91
D61	8.16 118.8	4.87/- 54.56	3.04/3.04 42	-/- -	-/- -	-/- -	176.99
E62	7.69 121.85	4.39/- 58.78	2.31/2.31 29.98	2.70/2.70 35.77	-/- -	-/- -	177.73
N63	9.92 117.78	4.77/- 55.7	3.35/3.35 37.15	-/- -	7.85/7.85 114.94	-/- -	176.04
K64	7.91 115.96	4.66/- 56.48	2.17/2.17 33.68	1.57/1.57 24.44	1.94/1.94 28.73	3.22/3.22 42.14	177.29
G65	9.03 111.07	4.15/4.15 47.41	-/- -	-/- -	-/- -	-/- -	175.53
I66	8.4 114.53	5.76/- 59.11	1.91/- 42.69	1.62/1.62 25.81	0.89/- 13.63	-/- -	174.88
V67	8.89 120.63	4.95/- 58.55	1.98/- 33.91	0.96/- 23.17	-/- -	-/- -	173.5
R68	7.62 125.3	5.25/- 55.89	1.73/1.73 31.37	-/- -	2.92/2.92 43.16	-/- -	176.52
I69	8.04 116.36	5.14/- 59.6	2.25/- 42.92	1.77/1.77 24.34	0.81/- 13.02	-/- -	174.49
D70	8.77 121.24	4.81/- 53.84	3.62/3.62 42.23	-/- -	-/- -	-/- -	176.48
S71	9.05 113.11	4.07/- 62.14	4.23/4.23 63.04	-/- -	-/- -	-/- -	176.82
V72	7.92 125.09	3.95/- 66.44	2.52/- 31.83	1.23/- 23.4	-/- -	-/- -	178.85
M73	8.74 120.83	4.41/- 59.8	2.48?/2.48? 33.45	-/- -	-/- -	2.09/- 17.58	179.93
R74	9.05 117.78	3.99/- 61.55	2.16?/2.16? 33.89	2.43?/2.43? -	-/- -	-/- -	178.93
N75	8.32 118.49	4.80/- 56.49	3.21/3.21 38.47	-/- -	7.83/7.83 112.5	-/- -	178.24

Rest	H^N ^{15}N	H^α C^α	H^β C^β	H^γ C^γ	H^δ C^δ	H^ε C^ε	C’
N76	8.61 118.8	4.98/- 55.56	3.24/3.24 38.3	-/- -	7.32/7.32 109.45	-/- -	176.73
C77	7.84 109.66	5.04/- 57.45	3.40/3.40 29.29	-/- -	-/- -	-/- -	175.1
G78	8.41 114.94	4.22/4.22 47.23	-/- -	-/- -	-/- -	-/- -	173.28
A79	8.36 121.84	5.00/- 51.01	1.43/- 23.75	-/- -	-/- -	-/- -	176.65
S80	9.35 117.78	4.86/- 57.16	4.06/3.80 64.87	-/- -	-/- -	-/- -	174.79
I81	8.59 122.86	3.51/- 64.39	1.99/- 36.92	1.80/1.80 28.55	1.06/- 12.61	-/- -	177.51
G82	9.14 117.07	4.74/4.74 45.25	-/- -	-/- -	-/- -	-/- -	174.4
D83	8.08 120.43	4.86/- 54.91	3.15/3.15 42.23	-/- -	-/- -	-/- -	176.35
K84	8.61 118.8	5.08/- 56.38	1.91/1.76 33.22	1.46/1.46 24.33	1.85/1.85 28.74	3.20/3.20 42.09	176.99
V85	9.22 114.94	4.97/- 58.87	2.19/- 35.5	0.91/- 23	-/- -	-/- -	173.71
K86	8.13 120.22	5.25/- 55.4	1.96/1.96 34.6	1.58/1.58 24.89	1.86/1.86 28.89	3.10/3.10 42.02	176.3
V87	9.26 124.89	5.49/- 60.73	2.02/- 34.5	1.03/1.03 21.78	-/- -	-/- -	174.71
R88	8.8 122.05	5.08/- 54.82	1.97/1.97 34.37	1.84/1.84 27.03	3.44/3.44 43.47	7.39/- 111.22	174.66
K89	9.19 125.09	4.91/- 58.14	2.14/2.14 33.91	1.55/1.55 24.79	1.88/1.88 29.98	3.20/3.20 42.03	176.22
V90	8.36 118.6	4.96/- 59.62	2.35/- 35.85	1.03/- 22.37	-/- -	-/- -	173.46
R91	8.15 119.61	4.90/- 55	2.05/2.05 32.06	1.87/1.87 27.04	3.45/3.45 43.17	-/- -	176.73
T92	8.3 111.48	4.62/- 59.89	4.19/- 70.41	1.16/- 24.57	-/- -	-/- -	175.18
E93	7.87 118.39	4.85/- 54.43	2.20/2.20 32.53	2.49/2.49 35.58	-/- -	-/- -	176.3
I94	8.89 122.66	4.33/- 61.07	2.03/- 36.45	1.74/1.74 27.63	0.98/- 11.19	-/- -	177.16
A95	8.96 128.55	4.56/- 52.67	1.35/- 18.43	-/- -	-/- -	-/- -	178.46
K96	9.79 125.3	4.63/- 58.04	2.12/2.12 32.76	1.74/1.74 25.31	2.15/2.15 28.68	3.23/3.23 42.05	178.67
K97	8 118.9	5.67/- 55.7	1.94/1.94 36.45	1.47/1.47 24.97	1.98/1.98 28.68	2.97/2.97 42.05	174.75
V98	8.76 123.67	4.68/- 62.04	1.95/- 36.45	1.05/- 21.71	-/- -	-/- -	173.46
T99	9.01 123.27	5.49/- 62.06	4.05/- 69.72	1.33/- 21.84	-/- -	-/- -	173.76
L100	9.37 126.32	5.74/- 52.86	1.80/1.46 46.16	1.65/- 27.02	0.83/0.83 24.66	-/- -	175.74
A101	9.24 122.05	5.66/- 49.25	1.63/- 22.13	-/- -	-/- -	-/- -	174.58

Rest	H^N ^{15}N	H^α C^α	H^β C^β	H^γ C^γ	H^δ C^δ	H^ε C^ε	C’
P102	- -	4.35/- 62.87	2.91/2.91 32.34	2.23/2.23 27.1	4.29/4.29 51.4	-/- -	175.78
I103	8.56 124.29	4.45/- 61.44	1.93/- 35.83	1.65/1.65 27.33	0.96/- 12.92	-/- -	174.84
I104	7.36 120.22	4.77/- 59.4	2.20/- 41.77	1.62/1.62 24.73	0.94/- 13.32	-/- -	175.14
R105	8.41 120.94	4.50/- 56.7	2.18/1.98 32.06	1.90/1.90 27.63	3.48/3.48 43.47	-/- -	177.47
K106	8.54 119	4.20/- 58.86	2.05/2.05 32.54	1.72/1.72 24.69	2.09/2.09 28.67	3.25/3.25 42.04	176.65
D107	8.53 115.55	4.73/- 54.23	3.10/3.10 40.15	-/- -	-/- -	-/- -	175.91
Q108	7.89 120.02	4.66/- 55.57	2.28/2.28 31.14	2.50/2.50 34.5	-/- -	7.67/7.67 111.55	174.88
R109	8.23 120.02	4.70/- 55.3	1.92/1.92 32.29	1.98?/1.98? 27.33	3.45/3.45 43.47	-/- -	175.61
L110	8.6 124.29	4.65/- 54.49	1.55/1.42 43.94	1.64/- 26.72	1.02/1.02 25.37	-/- -	175.31
K111	8.07 122.25	4.57/- 55.17	1.91/1.91 33.57	1.55/1.55 24.26	1.91/1.91 28.66	3.20/3.20 42.1	175.83
F112	8.76 123.26	4.94/- 57.55	3.35/3.19 40.61	-/- -	-/- -	-/- -	176.39
G113	8.62 108.84	4.43/4.43 44.66	-/- -	-/- -	-/- -	-/- -	174.1
E114	8.66 120.22	4.49/- 58.22	2.27/2.27 30	2.61/2.61 36.16	-/- -	-/- -	178.33
G115	9.23 111.28	4.44/4.44 46.23	-/- -	-/- -	-/- -	-/- -	176.82
I116	7.76 120.93	4.37/- 62.14	2.26/- 37.61	1.48/1.48 28.47	0.99/- 13.01	-/- -	176.48
E117	9.32 122.46	3.79/- 61.67	2.32/2.32 28.38	2.69/2.69 37.38	-/- -	-/- -	178.5
E118	8.21 118.59	4.30/- 59.74	2.32/2.32 29.6	2.58/2.58 36.81	-/- -	-/- -	178.67
Y119	7.91 119.92	4.23/- 61.94	3.37/3.37 38.3	-/- -	-/- -	-/- -	178.16
V120	8.73 118.8	3.40/- 67.19	2.12/- 31.14	1.12/- 23.82	-/- -	-/- -	176.6
Q121	8.22 118.59	3.63/- 60.92	2.43/2.43 28.05	2.58/2.58 33.11	-/- -	7.33/7.33 110.78	177.34
R122	7.65 114.43	4.08/- 58.83	2.05/2.05 29.98	2.03/2.03 27.33	3.47/3.47 43.17	-/- -	179.45
A123	8.25 120.63	4.24/- 54.13	1.38/- 18.9	-/- -	-/- -	-/- -	179.45
L124	7.86 114.53	4.63/- 53.84	1.88/1.88 43.85	2.03/- 25.81	1.04/1.04 22.82	-/- -	176.04
I125	7.14 121.84	3.73/- 63.31	2.13/- 37.61	1.72/1.72 28.85	0.99/- 13.01	-/- -	176.39
R126	8.91 118.6	4.05/- 59.04	2.26/2.26 28.77	2.02/2.02 27.33	3.49/3.49 43.17	-/- -	175.92
R127	9.06 124.08	4.85/- 54.24	2.09/2.09 30.68	1.88/1.88 27.03	3.46/3.46 43.47	-/- -	176.33

Rest	H^N ^{15}N	H^α C^α	H^β C^β	H^γ C^γ	H^δ C^δ	H^ε C^ε	C’
P128	- -	6.05/- 60.65	2.06/2.06 32.71	2.28/2.28 26.87	4.45/4.45 50	-/- -	175.7
M129	9.41 115.14	4.74/- 56.18	2.56/2.56 34.6	2.21?/2.21? 29.6	-/- -	1.98/- 17.37	172.64
L130	8.67 120.43	4.93/- 52.78	1.76/1.76 46.62	1.77/- 27.33	0.95/- 24.61	-/- -	176.6
E131	8.77 121.84	3.86/- 59.8	2.36/2.18 30.68	2.57/2.57 38.47	-/- -	-/- -	176.26
Q132	9.33 114.94	3.81/- 59.69	2.57/2.57 26.29	2.97/2.97 34.47	-/- -	7.87/7.87 112.35	176.48
D133	8.7 121.64	4.51/- 56.77	3.17/2.66 42.46	-/- -	-/- -	-/- -	176.04
N134	8.61 116.16	6.16/- 52.08	3.24/2.52 39.23	-/- -	8.02/7.77 111.68	-/- -	176.17
I135	9.81 116.77	4.91/- 59.8	1.99/- 43.38	1.67/1.67 26.18	0.93/- 14.84	-/- -	173.76
S136	8.7 117.58	5.28/- 56.67	4.05/4.05 65.56	-/- -	-/- -	-/- -	174.02
V137	9.31 125.09	4.71/- 59.46	2.44/- 33.13	1.13/- 21.71	-/- -	-/- -	174.24
P138	- -	4.63/- 63.84	2.51/2.51 32.25	2.21/2.21 27.33	3.92/3.92 50.53	-/- -	178.24
G139	8.93 110.26	4.36/4.00 46.16	-/- -	-/- -	-/- -	-/- -	175.44
L140	7.77 121.24	4.75/- 54.91	2.00/2.00 42.92	1.77/- 26.92	1.03/1.03 24.82	-/- -	176.13
T141	8.49 119	4.87/- 61.16	4.31/- 71.41	1.32/- 21.11	-/- -	-/- -	173.76
L142	8.44 124.08	4.74/- 54.62	1.79/1.79 44.08	1.80/- 27.03	1.11/1.11 25.2	-/- -	176.39
A143	8.83 125.71	4.35/- 53.26	1.61/- 18.2	-/- -	-/- -	-/- -	177.98
G144	8.6 107.22	4.33/4.33 45.93	-/- -	-/- -	-/- -	-/- -	174.49
Q145	8.27 119.2	4.80/- 55.4	2.41/2.41 30.22	2.56/2.56 33.67	-/- -	7.67/7.67 111.64	175.83
T146	8.42 115.55	4.69/- 62.21	4.42/- 69.95	1.46/- 21.35	-/- -	-/- -	175.05
G147	8.62 111.89	4.18/4.14 46.03	-/- -	-/- -	-/- -	-/- -	173.59
L148	8.05 122.35	4.53/- 55.34	1.72/1.72 43.89	1.43/- 26.72	0.90/0.90 23.98	-/- -	175.44
L149	8.8 126.52	5.22/- 53.55	1.85/1.51 45.92	1.87/- 27.03	1.10/1.10 25.69	-/- -	175.35
F150	9.28 116.97	5.23/- 56.38	2.88/2.88 43.38	-/- -	-/- -	-/- -	174.4
K151	9.8 122.05	5.33/- 54.52	1.88/1.88 34.37	1.57/1.57 24.79	1.87/1.87 28.77	3.09/3.09 42.06	176.65
V152	9.42 127.54	4.37/- 62.24	2.62/- 29.98	1.11/- 21.34	-/- -	-/- -	176
V153	8.62 127.33	4.13/- 64.48	2.16/- 32.53	1.17/- 21.87	-/- -	-/- -	176

Rest	H^N ^{15}N	H^α C^α	H^β C^β	H^γ C^γ	H^δ C^δ	H^ε C^ε	C’
K154	7.66 116.16	4.92/- 55.85	1.95/1.95 36.45	1.60/1.60 24.88	1.98/1.98 28.77	2.97/2.97 42.06	175.1
T155	9.42 116.56	5.13/- 60.87	4.39/- 72.03	1.24/- 21.53	-/- -	-/- -	172.16
L156	8.52 121.24	4.79/- 52.46	1.90/1.71 45.46	1.73/- 27.33	1.18/1.18 24.56	-/- -	-
P157	- -	4.93/- 63.38	2.67/2.19 34.6	2.12/2.12 24.78	3.82/3.82 50.71	-/- -	175.61
S158	8.82 116.57	4.75/- 58.14	4.33/4.06 65.1	-/- -	-/- -	-/- -	175.65
K159	7.93 117.78	3.93/- 59.99	2.27/2.27 30.45	1.63/1.63 25.49	1.93/1.93 29.17	3.28/3.28 42.13	174.96
V160	7.51 112.1	4.94/- 58.23	2.45/- 33.22	1.16/- 21.54	-/- -	-/- -	176.25
P161	- -	4.67/- 64	2.54/2.54 31.8	2.27/2.27 27.03	4.11/4.11 50.65	-/- -	176.73
V162	8.95 117.37	6.06/- 58.14	2.30/- 36.68	1.17/- 21.7	-/- -	-/- -	174.92
E163	8.51 114.33	5.69/- 53.65	1.79/1.79 35.07	2.04/2.04 36.3	-/- -	-/- -	175.4
I164	7.55 121.44	4.25/- 60.58	2.63/- 35.11	1.98/1.98 27.94	0.96/- 11.38	-/- -	177.16
G165	9.61 119	4.92/4.92 43.78	-/- -	-/- -	-/- -	-/- -	174.32
E166	9.19 119.41	4.08/- 60.09	2.23/2.23 29.75	2.59/2.59 36.77	-/- -	-/- -	177.98
E167	9.24 114.74	4.63/- 56.08	2.38/2.38 29.95	2.42/2.42 36.47	-/- -	-/- -	176.91
T168	7.64 119.41	4.01/- 64.59	4.20/- 69.49	1.23/- 23.93	-/- -	-/- -	173.8
K169	7.99 127.54	4.68/- 55.4	2.11/2.11 32.99	1.68/1.68 24.75	1.95/1.95 28.73	3.26/3.26 42.16	174.84
I170	8.83 126.22	4.98/- 58.72	2.04/- 38.3	1.44/1.44 27.03	0.82/- 11.49	-/- -	174.84
E171	8.89 127.33	4.93/- 54.23	2.23/2.05 32.99	2.47/2.47 36.16	-/- -	-/- -	174.79
I172	9.3 126.15	4.94/- 59.56	1.89/- 38.53	1.51/1.51 27.63	0.64/- 13.22	-/- -	176.04
R173	9.06 127.94	4.72/- 55.11	2.36/2.36 31.6	1.99/1.99 26.72	3.38/3.38 43.17	-/- -	175.14
E174	8.62 119.31	4.50/- 57.6	2.39/2.39 30.86	2.55/2.55 36.19	-/- -	-/- -	176.91
E175	8.11 118.8	4.77/- 54.47	2.20/2.20 29.75	2.64/2.64 36.05	-/- -	-/- -	-
P176	- -	4.73/- 62.71	2.54/2.14 32.99	2.31/2.31 27.94	4.24/4.24 50.78	-/- -	177.6
A177	8.69 124.49	4.42/- 53.06	1.49/- 19.36	-/- -	-/- -	-/- -	177.81
S178	8.39 113.31	4.52/- 59.31	4.12/4.12 63.71	-/- -	-/- -	-/- -	175.22
E179	8.49 122.25	4.48/- 57.32	2.33/2.33 30.53	2.53/2.53 36.2	-/- -	-/- -	177.04

Rest	H^N ^{15}N	H^α C^α	H^β C^β	H^γ C^γ	H^δ C^δ	H^ε C^ε	C’
V180	8.09 120.02	4.20/- 63.12	2.26/- 32.53	1.15/1.15 21.24	-/- -	-/- -	176.56
L181	8.3 124.08	4.51/- 55.6	1.83/1.77 42.46	1.84/- 27.03	1.11/1.11 24.89	-/- -	177.21
E182	8.31 121.44	4.50/- 56.83	2.28/2.28 30.45	2.51/2.51 36.2	-/- -	-/- -	175.61
E183	8.06 125.91	4.48/- 57.24	2.32/2.32 30.45	2.52/2.52 36.2	-/- -	-/- -	177.34
G184	8.58 109.66	4.21/4.21 45.83	-/- -	-/- -	-/- -	-/- -	175.01
G185	8.38 108.23	4.16/4.16 45.48	-/- -	-/- -	-/- -	-/- -	174.3

9.5 Zuordnungsliste des NADPH-Komplexes von DHFR

Rest	H^N	^{15}N	H^α
M1	-	-	4.24
I2	9.41	124.54	4.23
S3	9.37	126.00	5.98
L4	8.48	121.87	-
I5	8.46	119.12	5.99
A6	8.64	128.66	4.94
A7	7.61	120.06	4.49
L8	8.41	120.66	4.30
A9	8.72	124.60	4.98
V10	8.55	118.50	-
D11	8.27	119.69	-
R12	8.54	107.57	3.77
V13	7.53	122.96	3.74
I14	9.59	120.42	4.34
G15	8.64	108.30	3.86
M16	8.72	119.57	4.66
E17	8.85	123.33	-
N18	8.31	114.80	4.47
A19	7.69	120.66	4.16
M20	7.66	119.30	4.34
P21	-	-	4.40
W22	6.36	114.30	4.95
N23	9.47	118.90	4.84
L24	9.93	126.24	5.69
P25	-	-	4.29
A26	9.37	120.90	4.14
D27	9.09	121.15	4.66
L28	8.19	118.00	4.58
A29	7.69	119.60	4.16
W30	8.01	113.27	4.41
F31	9.33	122.59	3.66
K32	8.49	121.49	3.44
R33	7.87	116.66	3.82
N34	7.08	111.45	4.16
T35	7.16	106.59	3.91
L36	7.18	124.54	3.59
D37	8.38	115.93	3.56
K38	7.68	119.69	4.73
P39	-	-	5.07
V40	8.49	111.69	5.62
I41	8.48	121.39	4.83
M42	9.00	123.57	5.67
G43	9.01	105.51	5.64
R44	7.65	118.46	4.05
H45	-	-	4.47
T46	8.32	121.50	4.03
W47	8.31	123.33	4.69

E48	8.14	117.75	3.50
S49	7.82	114.24	4.21
I50	7.93	119.91	-
G51	7.76	106.59	3.67
R52	6.90	117.15	4.53
P53	-	-	4.27
L54	9.79	126.36	4.62
P55	-	-	4.45
G56	-	-	3.75
R57	7.23	118.20	4.42
K58	8.02	123.44	4.33
N59	8.95	125.15	4.86
I60	8.88	126.48	4.83
I61	8.86	126.48	4.44
L62	8.80	127.20	4.51
S63	10.21	122.84	-
S64	-	-	4.40
Q65	8.92	122.4	-
P66	-	-	4.01
G67	7.56	109.03	3.99
T68	6.20	106.17	4.24
D69	7.39	120.42	-
D70	-	-	-
R71	8.93	118.36	4.23
V72	7.24	107.82	-
T73	8.06	117.39	4.37
W74	8.78	129.02	5.05
V75	9.17	116.54	4.87
K76	8.29	115.93	5.04
S77	7.24	108.91	4.89
V78	8.91	122.72	3.39
D79	8.47	118.12	4.34
E80	7.97	120.66	4.02
A81	8.20	122.72	3.92
I82	7.97	116.05	3.60
A83	8.13	123.80	4.11
A84	8.09	119.44	4.11
C85	7.41	114.48	3.96
G86	7.28	103.33	3.77
D87	-	-	-
V88	7.28	114.84	4.76
P89	-	-	4.36
E90	7.71	116.900002	4.92
I91	8.78	129.70	3.88
M92	7.94	121.87	5.18
V93	9.30	125.38	4.64
I94	8.90	118.1	5.61
G95	6.16	105.00	2.41
G96	8.08	108.90	-
G97	-	-	-
R98	-	-	4.21
V99	7.14	120.42	3.88
Y100	7.45	112.70	4.38

E101	8.17	116.18	3.77
Q102	7.32	113.88	4.03
F103	7.86	112.20	4.46
L104	9.33	126.44	4.33
P105	-	-	4.58
K106	7.47	113.15	4.42
A107	7.98	122.59	4.58
Q108	9.28	118.59	4.62
K109	7.82	120.30	5.75
L110	8.99	121.75	5.09
Y111	9.37	122.48	5.25
L112	9.65	122.48	5.53
T113	7.58	115.01	5.06
H114	9.17	126.40	4.87
I115	9.09	126.72	4.00
D116	8.39	129.80	4.65
A117	7.79	124.41	4.37
E118	8.53	123.57	4.62
V119	8.55	118.54	4.43
E120	8.41	124.30	4.38
G121	8.45	106.97	4.37
D122	8.8	116.77	4.88
T123	8.66	115.33	5.29
H124	8.66	123.08	4.69
F125	9.07	124.80	4.43
P126	-	-	4.27
D127	8.17	117.82	4.53
Y128	7.33	118.30	-
E129	8.92	123.70	4.61
P130	-	-	4.43
D131	9.04	114.84	4.58
D132	8.19	118.12	4.58
W133	7.82	119.69	4.81
E134	9.60	122.95	4.69
S135	9.05	121.38	4.87
V136	9.04	121.87	4.62
F137	7.97	123.80	4.76
S138	7.53	119.44	5.39
E139	8.80	124.54	4.23
F140	8.71	129.27	4.16
H141	8.05	123.69	4.12
D142	8.02	120.41	4.48
A143	8.15	122.36	4.06
D144	9.09	121.26	4.65
A145	8.11	121.60	-
Q146	8.10	113.20	4.40
N147	8.45	119.30	5.60
S148	8.78	117.75	4.15
H149	7.24	120.06	4.82
S150	8.61	113.95	4.31
Y151	7.69	115.00	5.32
C152	8.33	118.00	4.92
F153	8.41	128.18	5.08

K154	9.80	123.00	-
I155	8.58	123.20	5.27
L156	9.29	125.75	5.04
E157	9.37	119.93	5.39
R158	8.30	127.08	3.38
R159	8.01	131.57	3.99