Anwendung von künstlichen neuronalen Netzen in der Analyse zweidimensionaler NMR-Spektren

Dissertation

zur Erlangung des Doktorgrades
des Fachbereichs Chemie
der Universität Hamburg

von
Christian Seeberger
aus Hamburg

April 2002

Universität Hamburg

Die vorliegende Arbeit wurde vom November 1998 bis zum Januar 2002 am Institut für organische Chemie der Universität Hamburg durchgeführt.

Herrn Prof. Dr. B. Meyer danke ich für die interessante Themenstellung, sowie für viele fruchtbare und anregende Diskussionen, die zum Erfolg dieser Arbeit geführt haben.

1 Einleitung

1.1 NMR-Spektroskopie zur Strukturaufklärung von Peptiden

Die Kenntnis über die Struktur von Proteinen ist eine wichtige Voraussetzung, um deren Funktion zu verstehen. Die klassische Methode zur Strukturaufklärung ist die Röntgenkristallographie[1,2]. Sie liefert Daten über die Struktur von Proteinen im kristallinen Zustand. Eine Aussage über das Verhalten in Lösung kann nicht getroffen werden.

Mit der Hochfeld-NMR-Spektroskopie und der Entwicklung von Techniken, um ^{15}N- und ^{13}C-markierte Proteine darzustellen[3,4], wurde es möglich, dreidimensionale Strukturen von Proteinen in Lösung zu bestimmen[5,6,7]. Dazu sind verschiedene, größtenteils mehrdimensionale NMR-Experimente[8] notwendig, deren Auswertung bedingt durch die große Anzahl an überlappenden Signalen sehr schwierig werden kann.

An erster Stelle einer NMR-basierten Strukturaufklärung steht stets die möglichst lückenlose Zuordnung aller Resonanzen zu den Atomen der Aminosäuren im Protein. Das bereits erwähnte Problem der Signalüberlappung kann durch die Aufnahme von zwei- oder auch dreidimensionalen Spektren reduziert werden. Bei mehrdimensionalen Spektren[9,10,11] kann eine Korrelation zwischen verschiedenen Atomen innerhalb des untersuchten Moleküls erhalten werden. Diese Korrelation kann sowohl homonuklear als auch heteronuklear erfolgen, so daß sich die im Protonenspektrum auftretenden Überlagerungen, z.B. durch eine C-H-Korrelation, auflösen lassen. Dadurch lassen sich unter anderem Informationen über die Konnektivitäten in diesem Molekül gewinnen.

Auf Proteine angewandt kann man durch Aufnahme eines sogenannten TOCSY-Spektrums[12] die chemischen Verschiebungen der Protonen der Seitenketten einzelner Aminosäuren bestimmen. Dabei erfolgt ein Magnetisierungstransfer innerhalb eines Spinsystems, welcher nur über skalar gekoppelte Protonen verläuft. Die wichtigste Region in solchen Spektren ist der Bereich zwischen 7 und 8.5 ppm, die sogenannte

Fingerprint-Region. Hier liegen die chemischen Verschiebungen der amidischen Protonen, die üblicherweise gut dispergiert sind. Von den amidischen Protonen erfolgt der Magnetisierungstransfer innerhalb der Seitenketten, jedoch nicht über die Peptidbindung. Jedes im TOCSY-Spektrum in dieser Region beobachtete Spinsystem entspricht bei Proteinen also einer Aminosäure. Nur Prolin ist hier nicht beobachtbar, da diese Aminosäure im Peptidrückgrat über kein NH-Proton verfügt. Die einzelnen Aminosäuren geben je nach der Art des Spinsystems ihrer Seitenkette charakteristische Spuren, wobei nahezu identische Aminosäuren (z.B. Glutamin und Glutamat) entsprechend auch nahezu identische Signalmuster erzeugen. Die NH-Region eines solchen TOCSY-Spektrums von einem Peptid aus 30 Aminosäuren ist in Abbildung 1 dargestellt.

Die chemischen Verschiebungen der amidischen und der α-Protonen sind besonders sensitiv hinsichtlich struktureller Einflüsse im Protein[13,14]. So können die NH-Hα-Kreuzsignale eines Aminosäuretyps, der in einer Sequenz mehrmals auftritt, je nach Sekundärstruktur im entsprechenden Proteinabschnitt deutlich unterschiedliche Positionen einnehmen. Umgekehrt können aus den Werten der NH- und Hα-Verschiebungen Rückschlüsse auf die Sekundärstruktur des Proteins gezogen werden, ein Verfahren, das als *Chemical Shift Index* bekannt ist[15,16]. Allerdings haben auch andere Faktoren, wie z.B. pH-Wert oder Temperatur Einfluß auf die Position der Spuren im Spektrum.

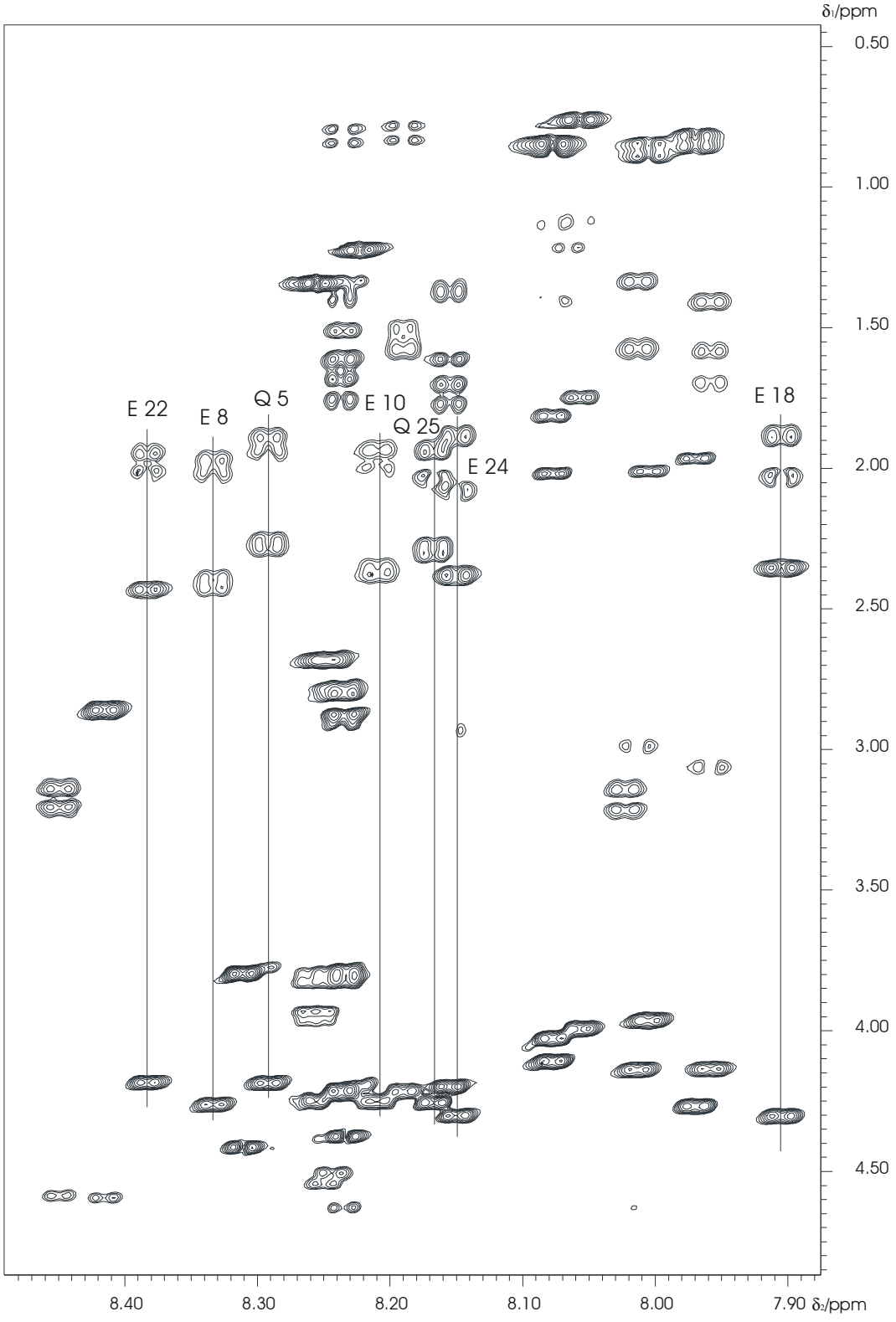


Abbildung 1: **NH-Bereich des TOCSY-Spektrums eines aus 30 Aminosäuren bestehenden Peptids aus dem Motorprotein Kinesin (H_2O/D_2O 9:1, 300 K, pH=3.5). Die Sequenz des Peptids lautet KSVIQHLEVELNRWRNGEAVPEDEQISAKD. Die ähnlichen Spinmuster der Aminosäuren E und Q sind durch Linien gekennzeichnet.**

Zur Bestimmung von strukturrelevanten Parametern wie Bindungs- und Torsionswinkeln sowie Atomabständen steht eine Anzahl verschiedener spektroskopischer Techniken zur Verfügung. An erster Stelle ist hier die NOE-Spektroskopie[17] zu nennen, mit deren Hilfe sich Informationen über Atomabstände gewinnen lassen. Die hierfür ausschlaggebende Wechselwirkung ist der *Nuclear Overhauser Effect* (NOE)[18]. Beim NOE ändert sich die Intensität des Signals eines Kerns, wenn sich die Population der Spinzustände eines benachbarten Kerns ändert. Die Kreuzrelaxationsrate σ und damit die Intensität eines Kreuzsignals in einem NOESY-Spektrum ist abhängig vom Abstand der beiden Kerne (r) zueinander. Außerdem hängt sie von der Spektrometerfrequenz (ω), dem gyromagnetischen Verhältnis der beteiligten Kerne γ, und von der Beweglichkeit des Moleküls, ausgedrückt durch die Korrelationszeit (τ_c), ab. Sichtbare NOEs treten auf, wenn der Abstand zwischen den Protonen nicht größer als ca. fünf Å ist.

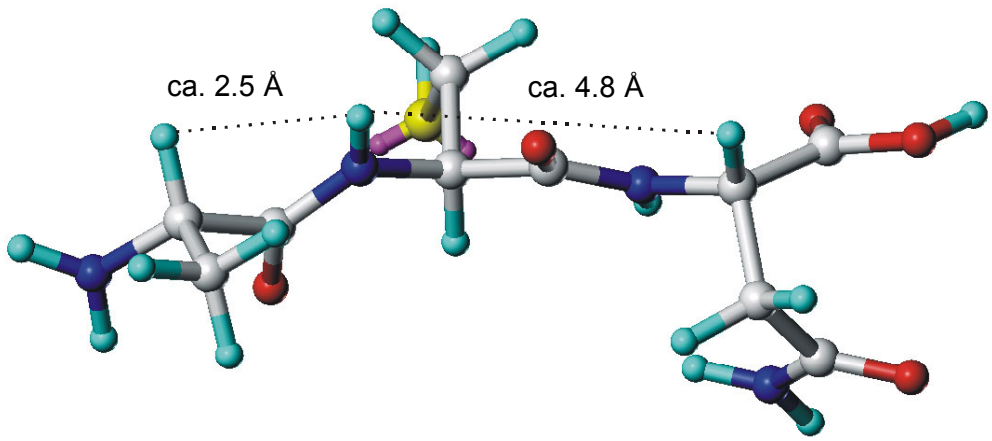


Abbildung 2: **Darstellung des Tripeptids Ala-Cys-Asn. Zwischen dem Amidproton des Cysteins und dem α-Proton des Alanins ist ein NOE sichtbar. Das α-Proton des Asparagins ist zu weit entfernt, der auftretende NOE ist nur schwach oder gar nicht beobachtbar.**

Um die im TOCSY-Spektrum auftretenden Spuren nun einzelnen Aminosäuren in der Proteinsequenz zuzuordnen, kann der sequentielle NOE zwischen dem amidischen Proton der Aminosäure an Position *n* und dem α-Proton der in der Sequenz vorhergehenden Aminosäure (Position *n-1*) herangezogen werden (Abbildung 2). In der entsprechenden NH-Spur der Aminosäure *n* ist dann im NOESY-Spektrum ein weiteres Kreuzsignal sichtbar, dessen chemische Verschiebung in der *f1*-Domäne der Verschiebung des α-Protons der Aminosäure *n-1* entspricht.

Somit können die vorher nur einem Aminosäuretyp zugewiesenen Spinsysteme genau einer Aminosäure in der Proteinsequenz zugeordnet werden[19,20]. Es ist nun auch eine Unterscheidung zwischen im TOCSY-Spektrum identisch erscheinenden Aminosäuren möglich, da die zusätzliche Information über die sequentielle Position verfügbar ist.

1.2 Neuronale Netze und ihre Anwendungen in der Chemie

Die NMR-Spektroskopie ist nur ein Beispiel unter vielen Bereichen der Chemie, in denen große Datenmengen bewältigt werden müssen. Tatsächlich ist es heute weniger das Problem, Daten zu messen oder zu erfassen, als vielmehr die große verfügbare Informationsmenge sinnvoll und effizient nach der gewünschten Auskunft zu durchsuchen. Das menschliche Gehirn ist in dieser Hinsicht jedem heute erhältlichem Computer überlegen. Während herkömmliche Computer zur Informationsverarbeitung sequentiell einen Satz von Befehlen abarbeiten, operiert das Gehirn in höchstem Maße parallel. Die Information wird gleichzeitig von vielen miteinander verbundenen Einheiten, den Neuronen, verarbeitet. Dies verkürzt die benötigte Zeit um auf einen Reiz zu reagieren um ein vielfaches. Die genaue Funktionsweise des Gehirns ist aber bei weitem nicht aufgeklärt. Künstlichen neuronalen Netzen[21,22,23] liegt somit ein stark vereinfachtes mathematisches Modell zu Grunde, mit dem die Arbeitsweise von Neuronen simuliert werden soll. Dieses kann bei bestimmten Problemstellungen zu befriedigenden Lösungen führen. Die Aufgabenstellungen für neuronale Netze umfassen[24]:

- Klassifikation: die Zuordnung eines Objektes zu einer bestimmten Kategorie anhand bestimmter Eigenschaften.
- Modellierung: die Vorhersage von Eigenschaften eines Objektes aus anderen, bekannten Eigenschaften heraus. Diese Vorhersage kann auch dann getroffen werden, wenn der explizite mathematische Zusammenhang zwischen den beiden Eigenschaften unbekannt ist.
- Abbildung: die Überführung komplexer Sachverhalte in eine einfachere Darstellung, wie z.B. die Projektion in eine Ebene.

Die Bereiche, in denen neuronale Netze für die angesprochenen Problemfelder eingesetzt werden, sind äußerst vielfältig. Als Beispiele seien hier nur Schrift- und Spracherkennung[25-28] oder die Vorhersage von Tendenzen am Aktienmarkt[29,30] genannt.

In der Chemie können neuronale Netze vielfältig eingesetzt werden[24,31,32]. In den Bereich der Klassifikation (ein bestimmtes Merkmal ist vorhanden oder nicht) gehört die Analyse von IR-Spektren, bei der die Zuordnung bestimmter Banden zu charakteristischen funktionellen Gruppen erfolgt[33-37]. Auch in der NMR-Spektroskopie wurden neuronale Netze mit Erfolg eingesetzt. So konnten ^{1}H-NMR-Spektren von komplexen Oligosacchariden identifiziert werden[38] oder auch stark verrauschte Spektren noch zugeordnet werden[39]. Weite Anwendung hat auch die Vorhersage oder Simulation von ^{13}C-NMR-Spektren mittels neuronaler Netze[40,41,42]. Die Anwendung neuronaler Netze zur Analyse zweidimensionaler Spektren hingegen gestaltet sich aufgrund der großen Datenmengen schwieriger[43,44,45]. Schließlich ist auch die Sekundärstrukturvorhersage von Proteinen ein Problem in dem neuronale Netze mit steigendem Erfolg angewendet werden[46-50].

Die Modellierung bestimmter Eigenschaften von Molekülen, also nicht nur eine einfache ja/nein-Aussage, sondern eine quantitative Angabe des gesuchten Parameters kann ebenfalls durch neuronale Netze durchgeführt werden[51,52]. Beispielsweise wurde der Selektivitätsfaktor bei HPLC-Analysen von Weinen in Abhängigkeit von Alkoholgehalt und pH-Wert der

mobilen Phase abgeleitet[53]. Ein breites Anwendungsfeld bietet die quantitative Analyse von Struktur-Wirkungs-Beziehungen (QSAR)[54,55]. Hier sollen aus der Molekülstruktur bestimmte Eigenschaften wie z.B. Toxizität oder pharmakologisches Potential abgeleitet werden. Neuronale Netze wurden hier unter anderem zur Vorhersage von anti-carcinogener[56,57], antihypertensiver[58] oder hypotensiver Wirkung[59] sowie Carcinogenität[60] eingesetzt. Auch physikalische Parameter wie Löslichkeit[61], Verteilungskoeffizienten (logP)[62,63] oder Siedepunkte[64,65] können von neuronalen Netzen berechnet werden. Ein weiteres Beispiel sind Übergangstemperaturen von Flüssigkristallen, die von neuronalen Netzen aus der Struktur der Verbindungen berechnet wurden[66,67]. Die Vorhersage der Isomerenverteilung bei der elektrophilen aromatischen Substitution an monosubstituierten Benzolen ist ein weiteres Beispiel, in dem die Ergebnisse der neuronalen Netze sogar besser waren als die auf Erfahrung basierenden Aussagen von Chemikern[68,69].

Chemische Eigenschaften werden oftmals von vielen verschiedenen Faktoren beeinflußt. Es ist meistens nicht möglich, den Einfluß dieser Faktoren schnell zu erfassen, da sie einen mehrdimensionalen Raum aufspannen, das menschliche Gehirn jedoch mit mehr als drei Dimensionen überfordert ist. Daher ist es nötig, diesen mehr-dimensionalen Raum auf eine Ebene abzubilden ohne die Bedeutung der einzelnen Parameter abzuschwächen. Eine von Kohonen eingeführte Art von neuronalen Netzen[70,71] kann dies bewerkstelligen. Damit lassen sich z.B. die elektrostatischen Potentiale auf Molekülоberflächen als eine Fläche darstellen[72]. Die so erhaltenen *Feature Maps* einzelner Moleküle lassen sich einfach miteinander vergleichen und können Aufschluß über die Ähnlichkeit und damit auch über ähnliche Eigenschaften geben[73].

1.3 Funktionsweise neuronaler Netze

Ein Grundmerkmal von neuronalen Netzen ist die Fähigkeit, selbständig anhand von Beispielen zu lernen und die so gewonnenen Zusammenhänge auf bisher unbekannte Fälle anzuwenden. Dazu wird

dem Netz ein Satz an Mustern über eine Anzahl von Zyklen immer wieder präsentiert. Beinhalten die Muster auch die zu den Eingabedaten gehörigen korrekten Ausgabedaten, so spricht man von einem überwachten Training oder auch von *supervised learning*. Sind die Ausgabewerte nicht explizit vorgegeben, so erfolgt das Training nicht überwacht (*unsupervised learning*). Die Anwendungsbereiche für diese beiden Klassen von neuronalen Netzen sind unterschiedlich. Die bereits erwähnten Kohonen-Netzwerke sind ein Beispiel für durch *unsupervised learning* trainierte Netze.

Bei der Methode des *unsupervised learning* lernt das neuronale Netz Zusammenhänge und Ähnlichkeiten der präsentierten Datensätze selbst zu finden. Am Ende steht also die Aussage, daß bestimmte Muster einander ähnlich sind, andere hingegen nicht. Wodurch diese Ähnlichkeit hervorgerufen wird kann jedoch oft nicht genau quantifiziert werden.

Beim *supervised learning* wird das Netz darauf optimiert, zu den angebotenen Eingabedaten die bekannten Ausgabedaten zu berechnen. Das Training wird beendet, sobald die Muster des Trainingsdatensatzes mit einer als ausreichend eingestuften Genauigkeit erkannt werden. Nun können dem Netz bisher unbekannte Daten präsentiert werden und die dazu gehörigen Ausgabedaten vorhergesagt werden.

Ein klassischer Lernalgorithmus für mittels *supervised learning* trainierte Netze ist die *Feed-forward Back-propagation* Methode[21,74,75]. Im ersten Schritt (*Feed-forward*) wird das Muster dem Netz präsentiert, und der Fehler der daraus resultierenden Ausgabewerte wird ermittelt. Dieser Fehler wird nun quasi rückwärts (*Back-propagation*) durch das neuronale Netz geleitet und dazu benutzt das Netz zu optimieren, so das der Fehler im nächsten Schritt kleiner wird. Die genauere mathematische Beschreibung dieses Verfahrens soll im folgendem gegeben werden.

1.3.1 Aufbau neuronaler Netze

Neuronale Netze stellen ein stark vereinfachtes Modell des menschlichen Gehirns dar. Sie bestehen aus einzelnen Neuronen (Abbildung 3), die in Schichten angeordnet sind. Die Neuronen

verschiedener Schichten sind über variable Gewichte miteinander verbunden, zwischen den Neuronen einer Schicht besteht jedoch keine Verbindung. Diese Gewichte sind das wichtigste Element eines künstlichen neuronalen Netzes. Durch die Änderung der Gewichte während des Trainings "lernt" das neuronale Netz die Zusammenhänge zwischen Ein- und Ausgabedaten.

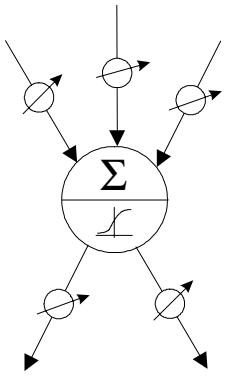


Abbildung 3: **Aufbau eines Neurons. Dargestellt sind die variablen Gewichte, die das Neuron mit den benachbarten Schichten verbinden.**

Die gewichtete Summe der Eingabewerte wird durch eine Transferfunktion auf das Intervall von 0 bis 1 abgebildet und dann als Ausgabe weitergeleitet. Jedes Neuron erhält von den mit ihm verbundenen Neuronen einen Input und gibt diesen nach Anwendung der Transferfunktion weiter an die Neuronen der nächsten Schicht.

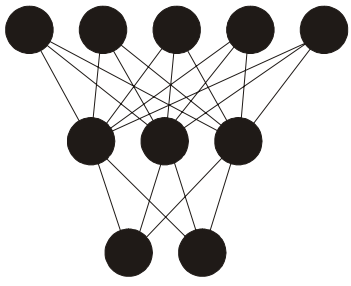


Abbildung 4: **Schematische Darstellung eines Neuronalen Netzes mit fünf Eingabe-, drei versteckten und zwei Ausgabeneuronen. Innerhalb einer Schicht sind die Neuronen nicht miteinander verbunden.**

Die meisten neuronalen Netze bestehen aus einer Eingabe-, einer versteckten und einer Ausgabeschicht. Da innerhalb der Eingabeschicht keinerlei Berechnungen erfolgen, wird diese oft nicht zur Anzahl der aktiven Schichten gezählt. Das in Abbildung 4 dargestellte Netzwerk besteht somit aus zwei prozessierenden Schichten.

1.3.2 Der Feed-forward Back-propagation Algorithmus

Das Training nach dem *Back-propagation*-Algorithmus läuft vereinfacht in den beiden folgenden Schritten ab:

- Berechnung der Ausgabe und der Fehler für ein Muster des Trainingsdatensatzes.
- Anpassung der Gewichte und Schwellwerte, um die aufgetretenen Fehler zu minimieren.

Diese beiden Schritte werden für jedes Muster im Trainingsdatensatz durchgeführt. Wurden alle Muster präsentiert, so ist ein Trainingszyklus beendet. Die Trainingszyklen werden solange wiederholt, bis ein bestimmtes Abbruchkriterium erfüllt ist. Dieses Kriterium kann z.B. der RMSD-Wert (*root mean square deviation*) sein, der die Abweichung der vorgegebenen Werte zu den berechneten Ausgaben widerspiegelt. Sobald er klein genug ist, kann das Training abgebrochen werden.

Der Eingabewert, den ein Neuron bekommt, ist die Summe aller Ausgabewerte der Neuronen in der vorherliegenden Schicht, multipliziert mit der jeweiligen Stärke des die Neuronen verbindenden Gewichts (Gleichung 1). Weiterhin wird jedem Neuron ein eigener Schwellwert t_j zugeordnet. Dieser Schwellwert hat einen Einfluß auf die Stärke des von dem Neuron weitergegebenen Signals. Für die Eingabeschicht ist diese Berechnung aus naheliegenden Gründen nicht notwendig. Für ein neuronales Netz mit N_e Eingabeneuronen, N_v versteckten Neuronen und N_a Ausgabeneuronen werden die Eingabewerte der versteckten Neuronen nach Gleichung 1 ermittelt.

$$S_j = s_j + \sum_{i=1}^{N_e} a_i w_{ij} \qquad j = 1,\ldots, N_v$$

Gleichung 1: Berechnung des Eingabewerts S_j eines Neurons *j*.
a_i: Ausgabewert des Neurons *i*.
w_{ij}: Gewicht welches die Neuronen *i* und *j* miteinander verbindet.
s_j: Schwellwert des Neurons *j*.

Die so gebildete gewichtete Summe kann beliebige Werte annehmen und wird nun mit der Transferfunktion f(x) auf das Intervall von 0 bis 1 abgebildet. Die ersten neuronalen Netze nutzten hierfür eine einfache Vergleichsfunktion[76]: Ist der Ausgabewert größer als ein bestimmter Schwellenwert, so wird er auf 1 gesetzt, ansonsten auf 0. Es konnte jedoch gezeigt werden, daß die Verwendung dieser Funktion dazu führt, daß nichtlineare Zusammenhänge vom neuronalen Netz nicht erkannt werden[77]. Um nichtlineare Zusammenhänge erfassen zu können und um möglichst hohe Flexibilität hinsichtlich verschiedener Trainingssituationen zu gewährleisten, werden in den meisten Fällen Transferfunktionen mit einem sigmoidem Verlauf eingesetzt. Eine solche Funktion ist in Abbildung 5 dargestellt.

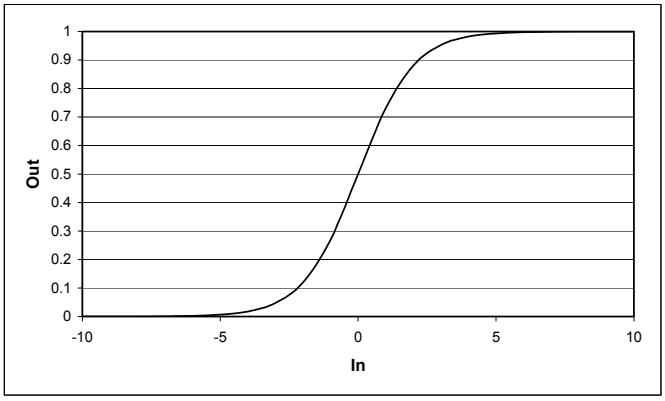


Abbildung 5: Verlauf der sigmoiden Transferfunktion $f(x) = 1/(1+e^{-x})$. Für kleine bzw. große x-Werte nähert sich die Funktion asymptotisch den Werten 0 bzw. 1 an.

Der bereits angesprochene Schwellwert eines Neurons verschiebt die gezeigte Kurve nach links oder rechts. Der Ausgabewert eines Neurons ist somit definiert als:

$$Out_j = f(S_j)$$

Gleichung 2: Ermittlung des Ausgabewertes Out_j des Neurons j.

Die so ermittelten Ausgabewerte bilden die Eingabewerte für die anschließende Neuronenschicht, für die die Berechnung auf die gleiche Art durchgeführt wird. Die Ausgabe des neuronalen Netzes setzt sich aus den Werten der Neuronen in der Ausgabeschicht zusammen. Die Ausgabe eines Musters wird bestimmt und mit dem bekanntem Zielwert T_j verglichen.

$$\varepsilon_j = T_j - Out_j \quad j = 1,...,N_a$$

Gleichung 3: Ermittlung des Fehler ε_j eines Ausgabeneurons.
T_j: bekannter Zielwert.
Out_j: berechneter Ausgabewert.

Der so ermittelte Fehler ε_j bildet nun die Grundlage für den zweiten Schritt, die Korrektur der Gewichte.

Die Korrektur eines Gewichtes zu einem Ausgabeneuron ist proportional zum aufgetretenen Fehler und zum Eingabewert, der diesen Fehler verursacht hat. Durch Verwendung einer Proportionalitätsfunktion anstatt einer Konstanten kann gewährleistet werden, daß die Korrektur stärker ist, wenn der Ausgabewert um den Wert 0.5 liegt. Somit wird erreicht, daß Gewichte die zu Neuronen führen deren Ausgabewerte schon nahe an 0 oder 1 liegen weniger stark korrigiert werden. Eine entsprechende Funktion ist in Gleichung 4 definiert.

$$g(x) = x \cdot (1 - x)$$

Gleichung 4: Proportionalitätsfunktion zur Angleichung der Gewichte.

Für den Korrekturwert δ_j für ein Ausgabeneuron ergibt sich somit:

$$\delta_j = g(Out_j) \cdot \varepsilon_j \text{ ; mit } g(Out) = Out \cdot (1 - Out)$$

$$\delta_j = Out_j \cdot (1 - Out_j) \cdot \varepsilon_j \qquad j = 1,.., N_a$$

Gleichung 5: Berechnung des Korrekturwerts δ_j eines Ausgabeneurons.
Out_j: berechneter Ausgabewert.
ε_j: Fehler des Ausgabewerts.

Die Änderung der Gewichte, die von den versteckten zu den Ausgabeneuronen führen, werden nun berechnet, wobei der Wert der Korrektur Δw_{ij} von folgenden Parametern abhängig ist:

- dem für das Ausgabeneuron j ermitteltem Korrekturwert δ_j.
- dem Wert Out_i des versteckten Neurons i.
- der Lernrate l.
- der Korrektur des entsprechenden Gewichtes im vorherigen Schritt $\Delta w_{ij}(alt)$.
- dem Momentum m.

Es ergibt sich also:

$$\Delta w_{ij} = m \cdot \Delta w_{ij}(alt) + l \cdot \delta_j \cdot Out_i \qquad i = 1,.., N_v \quad j = 1,.., N_a$$

Gleichung 6: Berechnung der Änderung des Gewichtes w_{ij}, welches ein Neuron i in der verdeckten mit einem Neuron j in der Ausgabeschicht verbindet.

Das Momentum m ist ein Parameter, der es ermöglicht, die Korrektur im vorherigen Schritt - also die Suchrichtung - einzubringen. Er kann Werte zwischen 0 und 1 annehmen und auch während des Trainings variiert werden. Durch Verwendung des Momentums ist es während des Trainings möglich, aus lokalen Minima zu entkommen. Die Lernrate l gibt

an, wie stark der Fehler des aktuellen Schrittes gewichtet werden soll, sie kann beliebige positive Werte annehmen. Ist sie zu groß gewählt, so erreicht das Netz möglicherweise nie ein globales Minimum. Ist sie hingegen zu klein gewählt, so werden die Trainingszeiten deutlich verlängert. Da die Lernrate l und der Wert des Neurons Out_j in Gleichung 6 positiv sind, wird das Vorzeichen der Änderung für den aktuellen Schritt nur von δ_j bestimmt. Ist der ausgegebene Wert größer als der Zielwert, so hat ε_j und damit auch δ_j ein negatives Vorzeichen, der Wert des entsprechenden Gewichtes wird also verkleinert.

Die Korrekturen für die Schwellwerte der Ausgabeneuronen werden nach einer ähnlichen Gleichung ermittelt, hier hat der Wert der versteckten Neuronen keinerlei Einfluss.

$$\Delta s_j = m \cdot \Delta s_j(alt) + l \cdot \delta_j \quad j = 1,..,N_a$$

Gleichung 7: Ermittlung der Korrektur der Schwellwerte in der Ausgabeschicht.

Sobald die Korrekturwerte für sämtliche Gewichte zwischen versteckter und Ausgabeschicht sowie die Schwellwerte bestimmt sind, wird die Berechnung für die nächste Schicht durchgeführt. Da es für die versteckten Neuronen keinen Zielwert gibt, gestaltet sich die Ermittlung des Fehlers etwas komplizierter. Die grundlegende Annahme hierbei ist, das sich der Fehler eines versteckten Neurons additiv aus den Fehlern aller mit ihm verbundenen Ausgabeneuronen zusammensetzt. Die Fehler der Ausgabeneuronen werden also rückwärts durch das Netz propagiert, um die Gewichte zu den versteckten Neuronen zu korrigieren. Für jedes versteckte Neuron kann ein Fehler nach Gleichung 8 angegeben werden.

$$\varepsilon_i = \sum_{j=1}^{N_a} w_{ij} \cdot \delta_j \qquad i = 1,..,N_v$$

Gleichung 8: Ermittlung des Fehlers eines Neurons *i* in der versteckten Schicht.
w_{ij}: Gewicht, das Neuron *i* in der versteckten Schicht mit Neuron *j* in der Ausgabeschicht verbindet.
δ_j: Korrekturwert für Neuron j.

Aus diesem Fehler kann mit einer Gleichung 5 entsprechenden Gleichung ein Korrekturwert errechnet werden (Gleichung 9).

$$\delta_i = g(Out_i) \cdot \varepsilon_i \text{; mit } g(Out) = Out \cdot (1 - Out)$$

$$\delta_i = Out_i \cdot (1 - Out_i) \cdot \varepsilon_i \qquad i = 1,..,N_v$$

Gleichung 9: Ermittlung des Korrekturwertes für ein Neuron *i* in der versteckten Schicht.

Die Änderung der Gewichte zwischen Eingabe- und verdeckter Schicht wird nun aus dem erhaltenen Korrekturwert errechnet. Die in Gleichung 10 auftauchenden Ausgabewerte der Eingabeschicht Out_h sind nichts anderes als die Eingaben in das neuronale Netz.

$$\Delta w_{hi} = m \cdot \Delta w_{hi}(alt) + l \cdot \delta_i \cdot Out_h \qquad h = 1,..,N_i \quad i = 1,..,N_v$$

Gleichung 10: Berechnung der Änderung des Gewichtes w_{hi}, welches ein Neuron *h* in der Eingabeschicht mit einem Neuron *i* in der verdeckten Schicht verbindet.

Entsprechend kann nun die Änderung der Schwellwerte für die verdeckten Neuronen bestimmt werden.

$$\Delta s_i = m \cdot \Delta s_i(alt) + l \cdot \delta_i \quad i = 1,..,N_v$$

Gleichung 11: Ermittlung der Korrektur der Schwellwerte in der verdeckten Schicht.

Nachdem nun sämtliche Korrekturwerte Δw_{hi}, Δw_{ij}, Δs_i und Δs_j bekannt sind, können alle Gewichte und Schwellwerte aktualisiert werden.

$$w_{hi} = w_{hi}(alt) + \Delta w_{hi} \quad h = 1,..,N_e \quad i = 1,..,N_v$$

$$w_{ij} = w_{ij}(alt) + \Delta w_{ij} \quad i = 1,..,N_v \quad j = 1,..,N_a$$

$$s_i = s_i(alt) + \Delta s_i \quad i = 1,..,N_v$$

$$s_j = s_j(alt) + \Delta s_j \quad j = 1,..,N_a$$

Mit der Änderung der Gewichte ist ein Trainingsschritt beendet. Es wird nun das nächste Muster präsentiert, wobei jetzt mit den korrigierten Gewichten gerechnet wird. Sind alle Muster einmal durch das neuronale Netz geschickt worden ist ein Trainingszyklus beendet. Im nächsten Zyklus werden nun erneut alle Muster präsentiert, wobei die Reihenfolge verändert wird.

1.3.3 Wichtige Parameter für neuronale Netze

Vor dem ersten Trainingszyklus muß ein neuronales Netz initialisiert werden, d.h. die Gewichte und Schwellwerte müssen mit Anfangswerten belegt werden. Dabei werden die Werte meist zufällig gewählt und liegen für eine Schicht im Intervall (−1/n, ..., 1/n), wobei n die Anzahl der Gewichte in der Schicht angibt.

Die Anzahl der Neuronen in Eingabe- und Ausgabeschicht wird durch die Problemstellung festgelegt. Die ideale Anzahl der Neuronen in der versteckten Schicht und damit die Anzahl der Verbindungen im Netz läßt sich oftmals nur durch Versuche herausfinden. Sind zu wenige versteckte Neuronen vorhanden, so können nicht alle für die Korrelation zwischen Ein- und Ausgabe notwendigen Parameter bestimmt werden. Zu viele versteckte Neuronen hingegen können zu übertrainierten Netzen führen: die Muster des Trainingssatzes werden hier exakt erkannt, das Netz hat jedoch nur eingeschränkte Fähigkeiten, für unbekannte Testdaten korrekte Ergebnisse auszugeben. Es ist also nur auf die Trainingsdaten optimiert und kann die darin enthaltenen Informationen nicht verallgemeinern.

Ein sehr wichtiger Parameter ist die Lernrate *l*. Sie bestimmt wesentlich die Geschwindigkeit mit der ein neuronales Netz lernt. Eine große Lernrate resultiert in entsprechend großen Gewichtsänderungen pro Trainingsschritt. Die Gefahr dabei in einem lokales Minimum auf der Fehleroberfläche zu geraten ist gering, allerdings konvergieren solche Netze sehr langsam oder gar nicht. Ist die Lernrate zu klein werden deutlich mehr Trainingszyklen benötigt, da die Änderungen an den Gewichten ebenso kleiner sind. In der Praxis hat es sich bewährt mit relativ hohen Lernraten zu beginnen und diese im Verlauf des Trainings kontinuierlich zu verringern.

Das Momentum *m* bestimmt wie weit die Suchrichtung im vorherigen Schritt für den aktuellen Schritt berücksichtigt wird. Somit können Oszillationen der Korrekturwerte gedämpft werden. Dabei ist wichtig, daß das Momentum nicht größer als 1 sein darf, da sonst im Verlauf des Trainings der Einfluß der vorherigen Suchschritte immer mehr anwächst. Das Momentum darf auch nicht isoliert von der Lernrate betrachtet werden, da beide Faktoren Einfluß auf die Änderung der Gewichte haben. Ist die das Momentum deutlich kleiner als die Lernrate, so hat die vorherige Suchrichtung kaum Einfluß auf die Korrektur des Gewichtes.

2 Problemstellung

Ziel dieser Arbeit ist es, Methoden zu finden um die Auswertung von Peptid-NMR-Spektren zu vereinfachen. Dabei soll versucht werden, bereits aus einfach zu erhaltenden TOCSY-Spektren ein Maximum an Informationen zu gewinnen. Vor allem die sequentielle Zuordnung der einzelnen Spuren ohne die Verwendung von NOESY-Spektren würde eine erhebliche Erleichterung darstellen. Falls trotzdem NOESY-Spektren verwendet werden müssen, so soll der Aufwand diese auszuwerten und mit den Daten aus den TOCSY-Spektren abzugleichen minimiert werden. Aber auch die automatisierte Erkennung der einzelnen Spuren verringert den nötigen Arbeitsaufwand beträchtlich. Es soll geprüft werden, inwiefern neuronale Netze für Zuordnung oder Vorhersage einzelner Spuren bzw. Kreuzsignale benutzt werden können.

Diese Aufgabe lässt sich in folgende Teilprobleme untergliedern:

1. Es muß ein Verfahren gefunden werden, die Eingabedaten möglichst aussagekräftig für die Eingabeschicht eines neuronalen Netzes zu kodieren. Dabei muß für die sequentielle Zuordnung die Aminosäuresequenz und für die Spurzuordnung die Lage der NMR-Signale dargestellt werden.
2. Auch die Ausgabewerte müssen in eine für das neuronale Netz verständliche Form überführt werden. Hier sind ebenfalls zwei verschiedene Parameter zu kodieren: die chemische Verschiebung von amidischen und α-Protonen einerseits, der Aminosäuretyp andererseits.
3. Die Ergebnisse der neuronalen Netze müssen evaluiert werden und sinnvoll miteinander verknüpft werden. Eine komplette Zuordnung eines TOCSY-Spektrums besteht in der Bestimmung des Aminosäuretyps einer Spur und in der Zuordnung dieser Spur zu genau einer Aminosäure in der Sequenz. Hierfür muß ein geeigneter Algorithmus entwickelt werden, der die Ausgaben der neuronalen Netze sinnvoll miteinander in Beziehung setzt.

Im folgenden Abschnitt sollen die verfolgten Strategien zur Lösung dieser Teilprobleme erörtert werden. Die eigentlichen Ergebnisse finden sich im Anschluß daran.

3 Methoden

3.1 Vorbereitung des zur Verfügung stehenden Datenmaterials

Um neuronale Netze erfolgreich trainieren zu können, muß eine ausreichend große Datenmenge verfügbar sein. Die Daten sollten einen statistisch relevanten Teil der für die jeweilige Problemstellung möglichen Fälle abdecken.

Für NMR-spektroskopische Daten von Proteinen und Peptiden steht die BioMagResBank[78,79] (BMRB) zur Verfügung. Diese enthält zur Zeit (Stand November 2001) Daten von über 1700 Proteinen und Peptiden, worin unter anderem über 330000 chemische Verschiebungen von Protonen enthalten sind. Die Datensätze können als Textdateien abgerufen werden, in denen die chemischen Verschiebungen und Zuordnungen als Listen aufgeführt sind. Aus diesen Dateien lassen sich die benötigten Informationen, z.B. mit der Programmiersprache PERL, bequem aufbereiten und in Muster für neuronale Netze überführen.

Die in dieser Arbeit verwendeten Daten wurden im März 1999 von der BMRB abgerufen, wobei zunächst alle verfügbaren Datensätze geladen wurden. Es handelte sich dabei um 1357 Datensätze. Die weiteren Auswahlkriterien und Kodierungsmethoden für die Daten werden im folgenden beschrieben.

3.2 Bestimmung des Aminosäurerestes

Die Zuordnung von Signalen zu Aminosäuretypen im TOCSY-Spektrum erfolgt durch Analyse der einzelnen Spuren. Um auftretende Signale auf die Eingabeschicht eines neuronalen Netzes zu kodieren wurde der Bereich von 0.0 ppm bis 6.5 ppm in 0.01 ppm große Intervalle aufgeteilt und einzelnen Eingabeneuronen zugeordnet. Je nach Lage der Signale konnten nun die entsprechenden Eingabeneuronen mit Werten belegt

werden. Die Ausgabeschicht bestand aus 20 Neuronen, von denen jedes einer bestimmten Aminosäure entsprach (Tabelle 1).

AS	*A*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*K*	*L*	*M*	*N*	*P*	*Q*	*R*	*S*	*T*	*V*	*W*	*Y*
Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Tabelle 1: Kodierung der Aminosäuren auf der Ausgabeschicht.

Da einige Aminosäuren im TOCSY-Spektrum nahezu identische Spinmuster aufweisen, wurden diese für die Auswertung zu Gruppen zusammengefaßt (Tabelle 2). Dabei galt eine Spur bereits als richtig klassifiziert, wenn für sie ein Ausgabeneuron der entsprechenden Gruppe aktiviert wurde. Die richtigen Ausgabeneuronen für zu Glutaminsäure gehörigen Mustern sind also '4' und '14'.

Aminosäuren	*Ausgabeneuronen*	*Gruppenbezeichnung*
Glu, Gln	4, 14	e
Asp, Asn	3, 12	d
Ile, Leu	8, 10	i
Lys, Arg	9, 15	k
Cys, His, Trp, Phe, Tyr	2, 5, 7, 19, 20	c

Tabelle 2: Gruppierung von Aminosäuren, die ähnliche TOCSY-Spuren verursachen. Die Bestimmung wurde als korrekt angesehen, wenn eines der Ausgabeneuronen der entsprechenden Gruppe aktiviert wurde. Die in dieser Tabelle nicht aufgeführten sieben Aminosäuren bilden jeweils eine eigene Gruppe, so daß insgesamt zwölf Klassen entstehen.

Die zusammengefaßten Aminosäuren wurden für manche Zwecke in der Sequenz mit kleinen Buchstaben bezeichnet, die alle in der Gruppe enthaltenen Aminosäuren beschreiben. Aminosäuren, die zu keiner der benannten Gruppen gehören, wurden weiterhin mit Großbuchstaben bezeichnet. Aus einer Sequenz 'EDFGQTRVVN' würde somit die korrespondierende Sequenz 'edcGeTkVVd'.

3.2.1 Statistische Mustererzeugung

Die Muster für die Zuordnung des Aminosäuretyps zu einer Spur wurden zunächst nach einer statistischen Methode erstellt. Dazu wurden von der BMRB erstellte Tabellen verwendet, in denen für jedes in Aminosäuren vorkommende Proton Werte für die mittlere chemische Verschiebung und die entsprechende Standardabweichung aufgeführt sind. Diese statistischen Daten wurden aus ca. 170 000 Werten der Datenbank errechnet und sind in Tabelle 3 dargestellt. Dabei wurden nur Protonen berücksichtigt, die in der NH-Spur ein Signal aufweisen. Der Vorteil dieser Methode ist, daß beliebig viele theoretische Muster pro Aminosäure erzeugt werden können, die innerhalb der statistischen Verteilung liegen. Man ist also nicht auf die limitierte Anzahl aus der Datenbank beschränkt.

Aus diesen Daten kann mit Hilfe der standardisierten Normalverteilung (Gleichung 12) die Wahrscheinlichkeit berechnet werden, mit der eine gegebene chemische Verschiebung innerhalb eines bestimmten Intervalls liegt.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Gleichung 12: Dichtefunktion der Normalverteilung. Die standardisierte Normalverteilung hat einen Mittelwert von $\mu = 0$ und eine Standardabweichung $\sigma = 1$.

Die Wahrscheinlichkeit P(x_1), mit der ein gegebener Wert x_1 in dieser Verteilung auftritt, kann mit Hilfe der integrierten Dichtefunktion angegeben werden (Gleichung 13).

Dabei ist zu beachten, daß die Wahrscheinlichkeit für einen diskreten Wert nicht berechnet werden kann, sondern nur die Wahrscheinlichkeit das ein Wert innerhalb eines bestimmten Bereichs liegt. Die Größe des Bereichs kann durch Änderung von n angepaßt werden.

AS	Proton	μ	σ	AS	Proton	μ	σ	AS	Proton	μ	σ
A	Hα	4.26	0.43								
	Hβ	1.38	0.28								
C	Hα	4.72	0.58	F	Hα	4.63	0.57	H	Hα	4.62	0.52
	Hβ2	2.94	0.43		Hβ2	2.96	0.35		Hβ2	3.1	0.49
	Hβ3	3.01	0.43		Hβ3	3.01	0.34		Hβ3	3.13	0.51
				W	Hα	4.72	0.58	Y	Hα	4.65	0.54
					Hβ2	3.19	0.35		Hβ2	2.89	0.35
					Hβ3	3.23	0.33		Hβ3	2.96	0.33
D	Hα	4.62	0.31	N	Hα	4.69	0.39				
	Hβ2	2.74	0.30		Hβ2	2.77	0.35				
	Hβ3	2.75	0.28		Hβ3	2.81	0.33				
E	Hα	4.26	0.42	Q	Hα	4.28	0.44				
	Hβ2	2.04	0.22		Hβ2	2.03	0.26				
	Hβ3	2.05	0.21		Hβ3	2.06	0.27				
	Hγ2	2.32	0.22		Hγ2	2.31	0.3				
	Hγ3	2.33	0.21		Hγ3	2.32	0.27				
I	Hα	4.2	0.56	L	Hα	4.31	0.45				
	Hβ	1.79	0.34		Hβ2	1.63	0.32				
	Hγ12	1.25	0.40		Hβ3	1.61	0.34				
	Hγ13	1.27	0.42		Hγ	1.53	0.31				
	Hγ2	0.79	0.32		Hδ1	0.76	0.26				
	Hδ1	0.7	0.33		Hδ2	0.77	0.28				
K	Hα	4.26	0.42	R	Hα	4.28	0.43				
	Hβ2	1.77	0.26		Hβ2	1.76	0.30				
	Hβ3	1.79	0.26		Hβ3	1.79	0.26				
	Hγ2	1.36	0.27		Hγ2	1.58	0.25				
	Hγ3	1.37	0.28		Hγ3	1.58	0.24				
	Hδ2	1.61	0.24		Hδ2	3.13	0.20				
	Hδ3	1.61	0.24		Hδ3	3.14	0.19				
	Hε2	2.92	0.2								
	Hε3	2.92	0.19								
G	Hα2	3.9	0.43								
	Hα3	3.98	0.39								
M	Hα	4.41	0.42								
	Hβ2	2.01	0.4								
	Hβ3	2.01	0.41								
	Hγ2	2.47	0.34								
	Hγ3	2.46	0.31								
P	Hα	4.41	0.36								
	Hβ2	2.01	0.4								
	Hβ3	2.09	0.42								
	Hγ2	1.91	0.39								
	Hγ3	1.94	0.37								
	Hδ2	3.61	0.38								
	Hδ3	3.65	0.42								
S	Hα	4.51	0.43								
	Hβ2	3.84	0.31								
	Hβ3	3.85	0.32								
T	Hα	4.49	0.51								
	Hβ	4.18	0.38								
	Hγ2	1.16	0.30								
V	Hα	4.15	0.56								
	Hβ	1.97	0.34								
	Hγ1	0.82	0.32								
	Hγ2	0.83	0.34								

Tabelle 3: Mittelwerte μ und Standardabweichungen σ für chemische Verschiebungen von Seitenkettenprotonen der Standardaminosäuren.

$$P(x_1) = \int_{-\infty}^{x_1+n} f(x)dx - \int_{-\infty}^{x_1-n} f(x)dx$$

Gleichung 13: Wahrscheinlichkeit für das Auftreten des Wertes x_1, der im Intervall $[x_1\text{-}n; x_1+n]$ liegt, in einer standardisierten Normalverteilung.

Um nun Wahrscheinlichkeiten für nicht standardisierte Normalverteilungen ($\sigma \neq 1$, $\mu \neq 0$) zu erhalten, müssen die entsprechenden Werte, hier also die chemischen Verschiebungen, zunächst auf das Normalkollektiv (*Z-Score*) standardisiert werden. Dies kann durch Gleichung 14 erreicht werden.

$$Z = \frac{X - \mu}{\sigma}$$

Gleichung 14: Standardisierung von Werten für die chemische Verschiebung.

X: chemische Verschiebung

μ: Mittelwert aller Werte

σ: Standardabweichung aller Werte

Zur Berechnung der Wahrscheinlichkeiten muß zunächst Gleichung 12 integriert werden. Da eine analytische Lösung dieses Integrals nicht möglich ist, wurden die Werte dafür im Intervall [-3;3] mit einer Schrittweite von 0.01 berechnet. Bei einer Normalverteilung liegen über 99 % aller Fälle in diesem Intervall. Der durch die Intervallgrenzen verursachte Fehler ist somit vernachlässigbar. In Abbildung 6 ist der Verlauf der standardisierten Normalverteilung dargestellt.

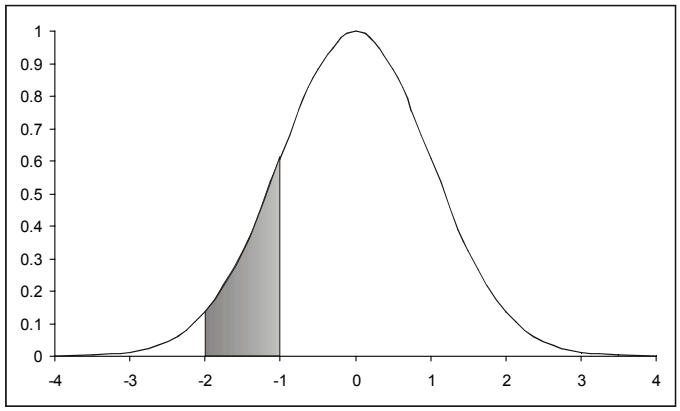


Abbildung 6: **Verlauf der Standardnormalverteilung im Intervall von -4 bis 4. Die hervorgehobene Fläche stellt die Wahrscheinlichkeit dar, mit der ein Wert zwischen -2 und -1 auftritt.**

Die eigentliche Berechnung der Wahrscheinlichkeit soll an einem Beispiel verdeutlicht werden, bei dem ein Signal für ein α-Proton eines Alaninrestes kodiert wird.

1. Eine Zufallszahl x im Bereich $(\mu-3\sigma) < x < (\mu+3\sigma)$ wird erzeugt. Für α-Protonen von Alanin sind die entsprechenden Werte $\mu = 4.26$ und $\sigma = 0.43$. Ein möglicher Wert ist also 4.14 ppm.
2. Die zufällig ausgewählte chemische Verschiebung wird mit Gleichung 14 normiert. Für eine Verschiebung von 4.14 ppm ergibt sich somit ein normierter Wert von -0.28.
3. Die Schrittweite des Integrationsintervalls von 0.01 wird auf den normierten Wert addiert bzw. von dem Wert subtrahiert. Man erhält die Grenzen für die Wahrscheinlichkeitsberechnung. Für das Beispiel bedeutet das, daß nun die Wahrscheinlichkeit bestimmt werden kann, mit der ein normierter Wert zwischen -0.27 und -0.29 auftritt.
4. Die Werte der entsprechenden Integrale werden bestimmt und nach Gleichung 13 voneinander subtrahiert. Dies gibt die gesuchte Wahrscheinlichkeit an, in diesem Fall 0.77 %.

5. Eine neue Zufallszahl *y* zwischen 0 und 100 wird erzeugt. Liegt ihr Wert unter der ermittelten Wahrscheinlichkeit, so wird die in Schritt 1 ermittelte chemische Verschiebung akzeptiert.

Dieses Verfahren kann für alle in Tabelle 3 aufgeführten Protonen angewandt werden um beliebig viele virtuelle Spuren für alle Aminosäuren zu erzeugen. Die Verteilung der chemischen Verschiebungen in diesem Ensemble entspricht dann der statistischen Verteilung.

Aus diesen generierten Signalen konnten nun Muster für das neuronale Netz erzeugt werden. Dazu wurden die entsprechenden Neuronen in diesem Ansatz auf den Wert '1' gesetzt, alle anderen Neuronen auf den Wert '0'. Eine Alaninspur mit den chemischen Verschiebungen 4.15 ppm und 1.22 ppm würde also Neuron 416 und Neuron 123 auf der Eingabeschicht auf 1 setzen, da das erste Eingabeneuron den Wert 0.00 ppm repräsentiert.

Diese Variante berücksichtigt allerdings den Zusammenhang zwischen einzelnen Protonen innerhalb einer Seitenkette nicht. Es ist zum Beispiel zu erwarten, daß Faktoren die zu einem vergleichsweise hohen Wert für die Verschiebung des α-Protons führen, auch Einfluß auf die chemische Verschiebung der übrigen Protonen in der Seitenkette haben. Diese Abhängigkeit wird durch ein statistisches Verfahren nicht abgebildet.

3.2.2 Muster aus realen Meßwerten

In einem weiteren Versuch wurden die Muster aus den Datensätzen der BMRB direkt erzeugt. Auch hier wurden nur die Protonen der einzelnen Aminosäuren kodiert, die in der NH-Spur Signale liefern. Weiterhin mußte für jede Aminosäure, die als Muster abgebildet werden sollte, eine minimale Anzahl an Signalen im Datensatz vorhanden sein (Tabelle 4).

minimale Signalanzahl	*Aminosäure(n)*
1	G
2	A C D F H N S W Y
3	E M Q T V
4	I L P R
5	K

Tabelle 4: Minimale Anzahl an gemessenen Signalen für die Eingabekodierung. Nur Datensätze die mindestens diese Anzahl an Signalen enthielten, wurden in Muster umgewandelt.

Die chemische Verschiebung wird auch durch die Aufnahmebedingungen der Spektren, wie z.B. Temperatur und Lösungsmittel, beeinflußt. Um das neuronale Netz hinsichtlich dieser Faktoren robuster zu gestalten, wurden die chemischen Verschiebungen aus der Datenbank mit einer gewissen Unschärfe versehen. Hierzu wurden zwei Verfahren angewendet.

3.2.2.1 Breite Kodierung

Im ersten wurde das Neuron, das der gemessenen chemischen Verschiebung entsprach, auf den Wert '1' gesetzt. Dann wurden die links und rechts benachbarten Neuronen mit linear abfallenden Werten belegt, so daß die Signale virtuell breiter wurden. Dabei konnte die Anzahl der benachbarten aktivierten Neuronen, also die Linienbreite, variiert werden. Die Werte V_p der benachbarten Neuronen waren abhängig von der Anzahl der ebenfalls angeregten Neuronen b und wurden nach Gleichung 15 berechnet.

$$V_p = 1 - \left(p \cdot \frac{1}{b+1} \right)$$

Gleichung 15: Berechnung der Eingabewerte für eine unscharfe Kodierung.

b: Breite der Kodierung; außer dem zentralen Neuron werden noch b vorhergehende und folgende Neuronen angeregt, insgesamt also 2b + 1 Neuronen.

p: Position des anzuregenden Neurons, bezogen auf das zentrale Neuron. p = 1 repräsentiert die direkten Nachbarn in beide Richtungen, p = 2 die übernächsten Neuronen usw.

Ein Signal bei 4.15 ppm wurde nach diesem Verfahren also wie in Abbildung 7 gezeigt auf die Eingabeschicht abgebildet.

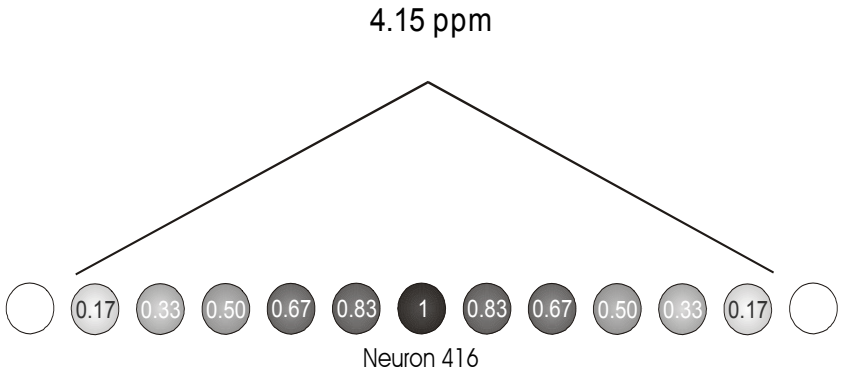


Abbildung 7: **Graphische Darstellung einer unscharfen Eingabekodierung mit einer Breite von fünf Neuronen.**

Bei dieser Eingabekodierung mußte noch berücksichtigt werden, daß Signale diastereotoper Protonen oft sehr nahe beieinander liegen. Durch die virtuelle Linienverbreiterung kommt es zur Überlagerung dieser Signale. Dieses Problem wurde gelöst, indem die Werte der überlappenden Neuronen aufsummiert wurden. Der maximale Wert, der bei diesen neuronalen Netzen auf der Eingabeschicht kodiert werden konnte, war '1'. Falls die resultierende Summe größer war, wurde sie entsprechend auf '1' gesetzt. Der Grund hierfür ist, daß keinerlei Daten über die Intensitäten der Signale vorhanden waren. Es war nur bekannt, ob das Signal auftritt oder nicht. Es sollte also kein NMR-Signal im eigentlichem Sinne abgebildet werden, sondern die Wahrscheinlichkeit, das ein Signal an der jeweiligen Stelle überhaupt beobachtbar ist. Die Kodierung von überlagerten Signalen ist in Abbildung 8 dargestellt.

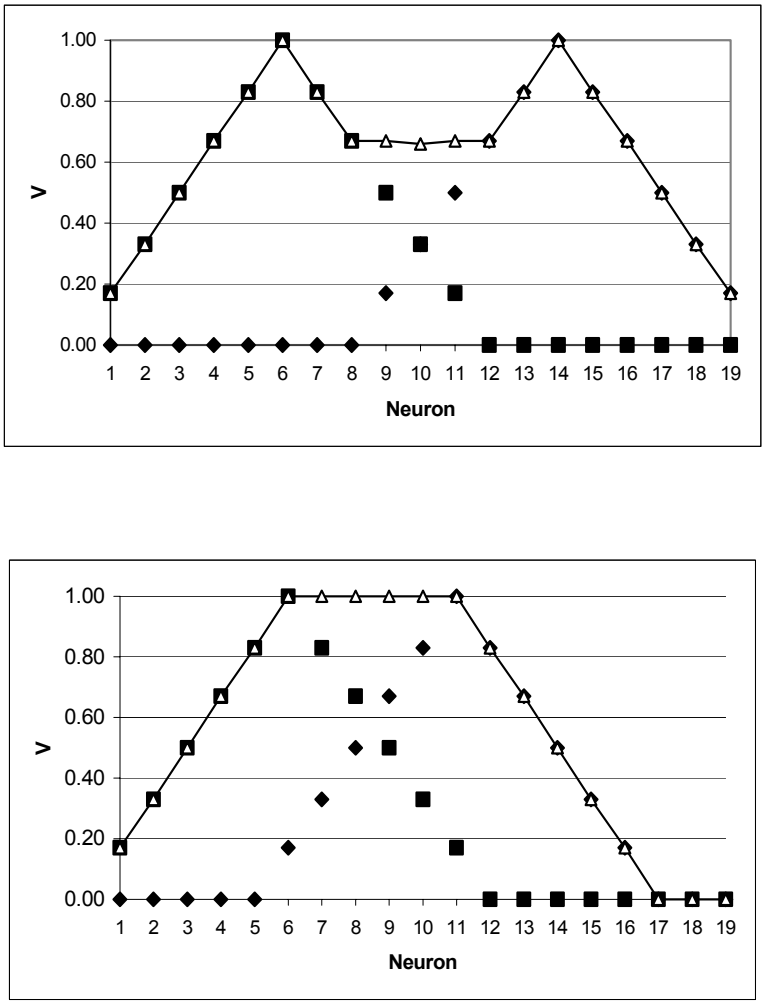


Abbildung 8: **Eingabekodierung von schwach (oben) bzw. stark (unten) überlagerten Signalen bei einer Breite von fünf Neuronen. Quadrate und Rauten stellen die einzelnen Signale dar, leere Dreiecke das resultierende Eingabesignal. Der maximale Wert von '1' für einzelne Neuronen wird nicht überschritten.**

3.2.2.2 Generierende Kodierung

Das zweite Verfahren erzeugte aus den gemessenen Werten innerhalb festgelegter Grenzen mehrere Muster, indem die chemischen Verschiebungen aus der Datenbank zufällig variiert wurden. Dabei konnten die Anzahl der erzeugten Muster und die maximale Änderung der Verschiebung angepaßt werden. Auch diese Methode soll am Beispiel einer Alaninspur mit den chemischen Verschiebungen 4.15 und 1.22 ppm verdeutlicht werden.

1. Aus der Spur wurde zunächst ein Muster erzeugt, bei dem die Neuronen 123 und 416 den Wert '1' bekamen.
2. Es wurde festgelegt, daß die chemischen Verschiebungen maximal um ± 0.05 ppm verändert werden dürfen. Diese Änderung entspricht der Linienbreite in der unscharfen Kodierung.
3. Aus der eigentlichen Spur wurden nun 4 weitere Spuren erzeugt, bei denen jede chemische Verschiebung auf einen zufälligen Wert aus dem Intervall $x \pm 0.05$ ppm gesetzt wurde (x entspricht dem gemessenen Wert aus der Datenbank).

Bei beiden Verfahren blieb der grundsätzliche Zusammenhang zwischen den chemischen Verschiebungen der einzelnen Protonen erhalten. Zusätzlich wurde die Möglichkeit, daß ein Signal je nach Aufnahmebedingungen leicht verschoben auftreten kann, in der Kodierung abgebildet.

Die drei erläuterten Kodierungsmöglichkeiten sollen im weiteren als statistische Kodierung (*SK*), breite Kodierung (*BK*) und generierende Kodierung (*GK*) bezeichnet werden. In allen drei Methoden gibt es verschiedene Parameter, deren Einfluß im Ergebnisteil diskutiert wird.

3.3 Vorhersage der Position des NH/Hα–Kreuzsignals

Da in längeren Peptiden einzelne Aminosäuren mehrmals auftreten, müssen in einem weiteren Schritt die zu Aminosäuretypen zugeordneten Spuren einzelnen Aminosäuren in der Sequenz zugeordnet werden. Die chemische Verschiebung der Hα- und NH-Protonen, und damit die Lage der einzelnen Spuren, wird von der chemischen Umgebung der jeweiligen Aminosäure und von Sekundärstrukturmotiven beeinflußt. Als Eingabe in ein neuronales Netz kann also die Peptidsequenz dienen, als Ausgabe werden dann die gesuchten chemischen Verschiebungen berechnet.

Die Sequenz des Proteins kann mit einem Sequenzfenster definierter Breite ausgelesen werden und dem neuronalen Netz präsentiert werden. Um mögliche lokale Sekundärstrukturmotive zu erfassen, sollte dieses

Fenster nicht zu schmal sein. In dieser Arbeit wurde meistens ein neun Aminosäurereste breiter Sequenzabschnitt verwendet. Die chemischen Verschiebungen der zentralen Aminosäure in diesem Fenster sollten vom neuronalen Netz vorhergesagt werden. Dann wurde das Fenster eine Position weiter geschoben und die nächste Aminosäure betrachtet, bis die komplette Sequenz untersucht war (Abbildung 9). Um die terminalen Aminosäuren ebenfalls auslesen zu können, wurden am Anfang und am Ende der Sequenz Platzhalter angefügt, die als 'O' dargestellt wurden.

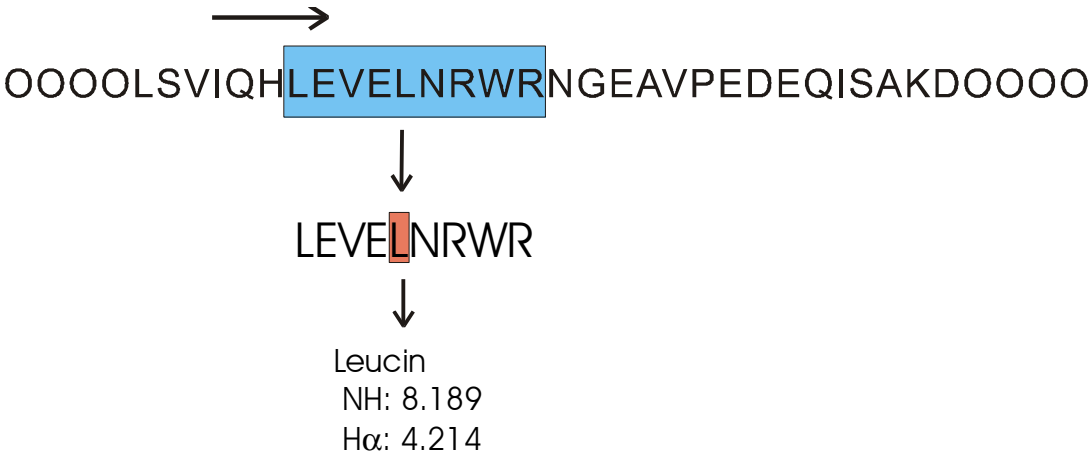


Abbildung 9: **Auslesen einer Proteinsequenz mit einem neun Aminosäuren breitem Sequenzfenster. Die chemischen Verschiebungen der zentralen Aminosäure sollen bestimmt werden. An beiden Termini der Sequenz wurden jeweils vier Platzhalter (O) angefügt.**

Auch für die Darstellung einer Peptidsequenz auf der Eingabeschicht eines neuronalen Netzes wurden verschiedene Methoden erarbeitet.

3.3.1 Standardkodierung für Aminosäuren

Die einfachste Variante eine Sequenz abzubilden entspricht der Ausgabekodierung für die neuronalen Netze zur Spurerkennung. Jeder Position im Sequenzfenster werden 21 Neuronen zugeordnet: 20 Neuronen für die verschiedenen Aminosäuren und ein weiteres für den möglicherweise an dieser Position auftretenden Platzhalter. Die Eingabeschicht für ein neun Reste breites Leseraster besteht somit aus

189 Neuronen. Die Zuordnung der einzelnen Neuronen entspricht der in Tabelle 1, wobei ein zusätzliches Neuron an Position 21 für den Platzhalter 'O' eingeführt wurde. Je nach der an der jeweiligen Position auftretenden Aminosäure wird nun eines dieser 21 Neuronen auf '1' gesetzt, die anderen erhalten den Wert '0'.

Bei dieser Kodierung sind nur neun der 189 vorhandenen Neuronen pro Muster aktiviert, die Informationsdichte auf der Eingabeschicht ist also sehr gering. Außerdem sind keinerlei Informationen über die Struktur einzelner Aminosäurereste enthalten. Um diese Probleme zu lösen, wurden noch weitere Kodierungsvarianten verwendet.

3.3.2 Kompakte Kodierung für Aminosäuren

Um die chemische Struktur der Seitenketten besser abzubilden wurde zunächst betrachtet, welche funktionellen Gruppen in den einzelnen Aminosäuren an welcher Position innerhalb der Kette auftreten. Zwischen den verschiedenen aromatischen Gruppen von Phenylalanin, Histidin, Tyrosin und Tryptophan wurde dabei nicht weiter differenziert. Auch die Guanidinofunktion von Arginin wurde zur Vereinfachung als eine einzige funktionelle Gruppe betrachtet. Diese Auflistung ist in Tabelle 5 dargestellt.

Wie aus der Tabelle ersichtlich, haben die 20 Standardaminosäuren maximal sechs Positionen in der Seitenkette, die mit funktionellen Gruppen belegt sein können. In manchen Fällen (Valin, Leucin, Isoleucin, Threonin) sind Positionen doppelt besetzt.

Aminosäure	α	β	γ	δ	ε	ϕ
aliphatische Seitenketten						
Gly	CH_2					
Ala	CH	CH_3				
Val	CH	CH	2 x CH_3			
Leu	CH	CH_2	CH	2 x CH_3		
Ile	CH	CH	CH_2, CH_3	CH_3		
Pro	CH	CH_2	CH_2	CH_2		
aromatische Seitenketten						
Phe	CH	CH_2	Aromat			
Tyr	CH	CH_2	Aromat			
Trp	CH	CH_2	Aromat			
polare, ungeladene Seitenketten						
Ser	CH	CH_2	OH			
Thr	CH	CH	CH_3, OH			
Cys	CH	CH_2	SH			
Met	CH	CH_2	CH_2	S	CH_3	
Asn	CH	CH_2	CO	NH_2		
Gln	CH	CH_2	CH_2	CO	NH_2	
negativ geladene Seitenketten						
Asp	CH	CH_2	CO	OH		
Glu	CH	CH_2	CH_2	CO	OH	
positiv geladeneSeitenketten						
Lys	CH	CH_2	CH_2	CH_2	CH_2	NH_2
Arg	CH	CH_2	CH_2	CH_2	Guanidino	
His	CH	CH_2	Aromat			

Tabelle 5: Funktionelle Gruppen in den Seitenketten von Aminosäuren.

Eine Aminosäure kann also durch sechs Neuronen in ihrer Struktur dargestellt werden, die Zuordnung von Werten zu einer bestimmten

funktionellen Gruppe ist dabei willkürlich. Da man bei einer Sequenzlänge von neun Aminosäuren nun mit 54 Eingabeneuronen auskommt, ist diese Form der Kodierung deutlich kompakter. Zusätzlich sind nun weitaus mehr Eingabeneuronen mit Werten belegt. Die Zuordnung einzelner funktioneller Gruppen zu Zahlenwerten ist in Tabelle 6 dargestellt.

Gruppe	*CH*	*CH_2*	*CH_3*	*Aromat*	*Guanidino*	*S*	*SH*	*OH*	*CO*	*NH_2*
Wert	1	2	3	4	5	6	7	8	9	10

Tabelle 6: Zuordnung funktioneller Gruppen zu Eingabewerten.

Für die vier Aminosäuren, die eine doppelt belegte Position besitzen, muß festgelegt werden, wie diese Sonderfälle zu behandeln sind. Bei Valin und Leucin sind die jeweils terminalen Positionen der Seitenketten mit je zwei Methylgruppen belegt. Hier wurde nur eine dieser Methylgruppen kodiert. Bei Isoleucin wurde die Methylengruppe an der γ-Position kodiert, da angenommen wurde das der Einfluß von Methyl- und Methylengruppen auf die chemische Verschiebung von benachbarten Protonen nahezu gleich ist. Im Falle des Threonins wurde die Hydroxylgruppe in der γ-Position kodiert, da diese deutlich größeren Einfluß auf die gesuchten chemischen Verschiebungen hat. Abbildung 10 stellt diese Kodierungsform an zwei Beispielen dar.

Die bei dieser Kodierung auftretenden Eingabewerte sind, im Gegensatz zu allen anderen bisher angesprochenen Methoden, ganze Zahlen, die auch größer als '1' werden können. Eine Hydroxylgruppe (Wert '8') wird in diesem Verfahren vier mal höher gewichtet als eine Methylengruppe (Wert '2'). Eine Kodierung, die die Effekte der funktionellen Gruppen auf die chemische Verschiebung besser beschreibt, konnte aufgrund fehlender Daten diesbezüglich nicht erarbeitet werden. Da der genaue Einfluß dieser Faktoren schwer abzuschätzen war, wurde ein weiteres Kodierungs-schema entwickelt. Dieses beinhaltete sowohl Strukturinformationen als auch digitale Eingabewerte.

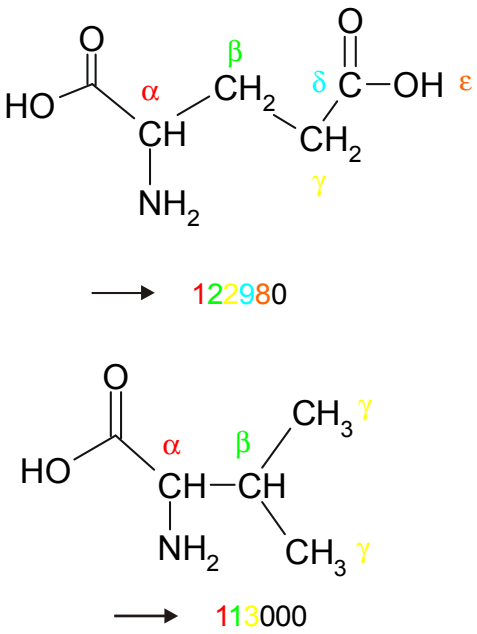


Abbildung 10: **Kompakte Kodierung für Glutaminsäure und Valin. Nicht belegte Positionen in der Seitenkette werden durch den Wert '0' dargestellt.**

3.3.3 Bitstring Kodierung für Aminosäuren

Auch diese Kodierungsmethode betrachtet die Positionen in der Seitenkette von Aminosäuren. Allerdings wird hier nicht für jede der sechs potentiellen Positionen nur ein Neuron verwendet, vielmehr richtet sich die Anzahl der Neuronen nach der Anzahl der möglichen Gruppen an dieser Position. Für die α-Position existieren beispielsweise nur zwei mögliche Belegungen: eine Methylengruppe bei Glycin oder eine Methingruppe bei allen anderen Aminosäuren. Die α-Position kann also durch zwei Neuronen dargestellt werden, von denen eines beim Auftreten einer Methingruppe aktiviert wird, das andere hingegen beim Auftreten einer Methylengruppe. Für jede Position gibt es also, je nach funktioneller Gruppe, eine charakteristische Folge von Bits. Eine komplette Aminosäure wird durch Zusammenfügen der entsprechenden Zeichenfolgen repräsentiert (Tabelle 7).

Ein Vorteil dieser Methode ist es, daß Positionen die mit zwei verschiedenen Gruppen belegt sind (γ-Position von Isoleucin und Threonin), korrekt wiedergegeben werden können.

	Position	α	β	γ	δ	ε	ϕ
Gruppe							
CH		10	100	1000000	000000	00000	0
CH_2		01	010	0100000	100000	10000	0
CH_3		00	001	0010000	010000	01000	0
CO		00	000	0001000	001000	00000	0
Aromat		00	000	0000001	000000	00000	0
OH		00	000	0000010	000100	00100	0
S		00	000	0000000	000010	00000	0
SH		00	000	0000100	000000	00000	0
Guanidino		00	000	0000000	000000	00010	0
NH_2		00	000	0000000	000001	00001	1

Tabelle 7: Kodierungstabelle für die Bitstring Kodierung. Je nach funktioneller Gruppe an den einzelnen Positionen werden die entsprechenden Strings aneinander gefügt.

In diesen Fällen werden einfach beide entsprechenden Bits auf '1' gesetzt. Die γ-Position von Threonin kann somit durch den String '0010010' dargestellt werden. Diese Kodierung lässt nur Werte von '0' oder '1' für die Neuronen zu. Bei zwei gleichen Gruppen an einer Position (terminale Methylgruppen in Valin und Leucin) wird also nur das entsprechende Bit aktiviert, da eine korrekte Abbildung erfordern würde, die entsprechende Position mit dem Wert '2' zu belegen (Abbildung 11).

Bei dieser Art der Kodierung geben strukturell ähnliche Aminosäuren auch sehr ähnliche Eingabemuster. So unterscheiden sich die Muster für Glutaminsäure und das entsprechende Säureamid Glutamin nur in der fünften Stelle, die die ε-Position kodiert.

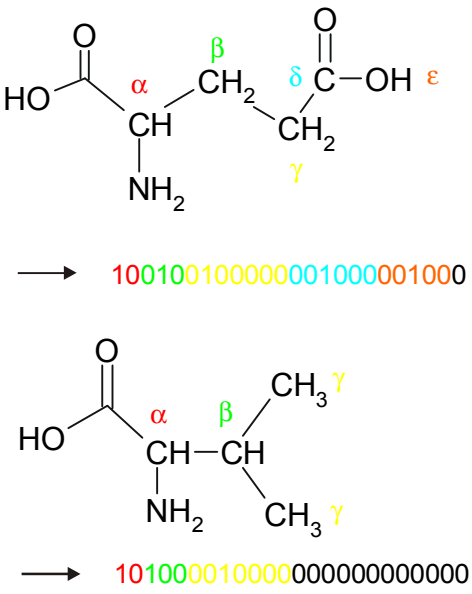


Abbildung 11: **Bitstring Kodierung für Glutaminsäure und Valin. Nicht belegte Positionen werden durch die entsprechende Anzahl von Nullen kodiert.**

3.4 Ausgabekodierung

Die Ausgabeneuronen der verwendeten neuronalen Netze können aufgrund der verwendeten Transferfunktion zwischen versteckter und Ausgabeschicht nur Werte zwischen '0' und '1' annehmen. Die genauen Werte '0' und '1' werden allerdings nie erreicht. Deswegen wurde als minimale Ausgabe der Wert '0.05', als maximale Ausgabe der Wert '0.95' definiert. Mit einer einfachen linearen Funktion können chemische Verschiebungen auf dieses Intervall abgebildet werden. Dazu müssen zunächst das Minimum und das Maximum der zu kodierenden Werte festgelegt werden. In den verwendeten Daten wurden als Minimum (sh_{min}) 0.00 ppm und als Maximum (sh_{max}) 12.22 ppm gefunden. Mit Gleichung 16 kann nun jede chemische Verschiebung zwischen sh_{min} und sh_{max} in Werte zwischen '0.05' und '0.95' umgerechnet werden.

$$x = \left(\frac{sh - sh_{\min}}{sh_{\max} - sh_{\min}} \right) \cdot 0.9 + 0.05$$

Gleichung 16: Umrechnung von chemischen Verschiebungen *sh* in Werte für Ausgabeneuronen *x*.
***sh*$_{min}$: minimale auftretende Verschiebung**
***sh*$_{max}$: maximal auftretende Verschiebung**

Für die verwendeten Werte ergibt sich mit dieser Gleichung die in Abbildung 12 dargestellte Gerade.

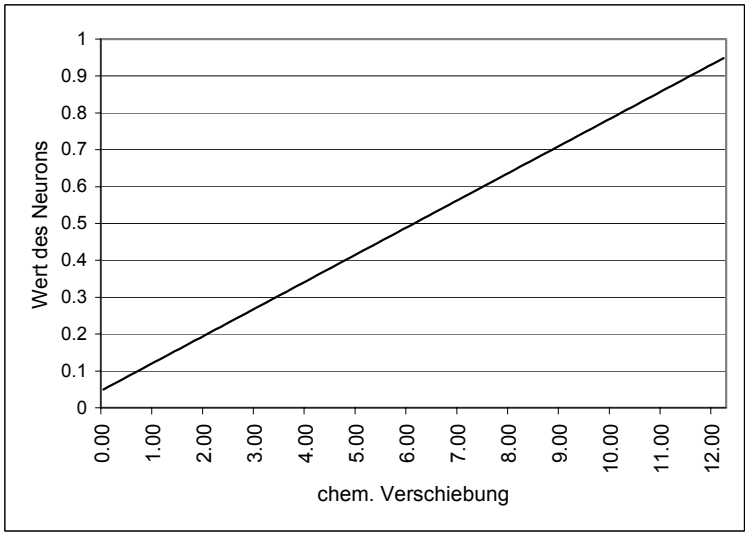


Abbildung 12: Abbildung von chemischen Verschiebungen zwischen 0.00 ppm und 12.22 ppm auf das Intervall 0.05 - 0.95 .

Prinzipiell bestehen für die Ausgabeschicht dieser Netze zwei Möglichkeiten. Entweder trainiert man für jede gesuchte Verschiebung ein eigenes Netz mit nur einem Ausgabeneuron, oder die Ausgabeschicht der neuronalen Netze besteht aus zwei Neuronen. Das erste kodiert dann die chemische Verschiebung des amidischen Protons, das zweite die Verschiebung des Hα-Protons. Bei neuronalen Netzen mit nur einem Ausgabeneuron können die Parameter sh_{min} und sh_{max} dann der gesuchten chemischen Verschiebung entsprechend angepaßt werden. So wurden für Netze, welche nur für α-Protonen verwendet werden sollten, sh_{min} auf 3 ppm und sh_{max} auf 5 ppm gesetzt. Die entsprechenden Werte für Netze,

die chemische Verschiebungen von amidischen Protonen berechnen sollten, wurden auf 7 ppm bzw. 9 ppm gesetzt.

3.5 Inkrementsystem

Zusätzlich zu den bisher beschriebenen neuronalen Netzen wurde ein Inkrementsystem zur Berechnung der chemischen Verschiebungen entwickelt. Auch hier sollten die Werte für die zentrale Aminosäure innerhalb eines Fensters berechnet werden, dessen Breite variabel gestaltet werden kann. Die Daten für das Inkrementsystem wurden aus den gleichen Datensätzen erstellt, mit denen auch die neuronalen Netze trainiert wurden.

Zuerst wurden die Sequenzen aller Datensätze mit einem festgelegtem Sequenzfenster ausgelesen. In diesem Beispiel wird von einer Breite von neun Resten ausgegangen. Diese neun Aminosäuren langen Fragmente wurden nun danach sortiert, welche Aminosäure an der zentralen Position auftrat. Aus diesen sortierten Fragmenten wurden nun für jede der 20 Aminosäuren Mittelwerte für die chemischen Verschiebungen der NH- und Hα-Protonen berechnet. Diese sollen im weiteren als $V\text{-}MW_{AS}$ bezeichnet werden, wobei V für die entsprechende chemische Verschiebung (NH oder Hα) und AS für den Typ der Aminosäure steht. Ausgehend von diesen 40 Mittelwerten wurde nun analysiert, wie sich diese in Abhängigkeit der benachbarten Aminosäuren ändern. Dabei wurden die NH- und Hα-Verschiebungen jeder Aminosäure für sich betrachtet, insgesamt also 40 verschiedene Systeme entwickelt.

Da die zentrale Aminosäure im Fenster festgelegt war, waren nur noch acht Positionen innerhalb der Sequenzabschnitte variabel. Für jedes Sequenzfragment wurde nun untersucht, welche Aminosäuren sich an welcher der variablen Positionen befanden und wie die chemischen Verschiebungen der zentralen Aminosäure waren. Diese Verschiebungen wurden in Listen gespeichert, welche über drei Indices referenziert wurden. Der erste Index gibt an, für welche variable Position die entsprechende Liste gilt. Der zweite Index definiert, welche Aminosäure

sich an dieser Stelle befindet. Der dritte Index gibt dann die Position innerhalb der Liste an. So wird durch das Element *NH-Verschiebungen$_Q$[-4][K][1]* der erste Eintrag der Liste referenziert, in der die Werte für die chemische Verschiebung von amidischen Protonen aller Glutaminreste stehen, bei denen vier Positionen vorher in der Sequenz ein Lysin auftaucht (vgl. Abbildung 13). Bei acht variablen Positionen, die mit jeweils 20 Aminosäuren belegt sein können, ergeben sich bei dieser Analyse pro Aminosäure und chemischer Verschiebung also 160 Listen.

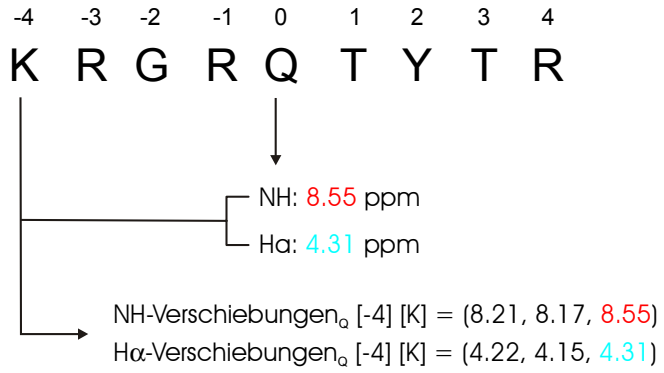


Abbildung 13: **Datenstruktur zur Speicherung der chemischen Verschiebungen. Für jede zentrale Aminosäure (hier Glutamin) wurden Listen erstellt, in denen die Werte über drei Indices abgelegt wurden. Der erste Index gibt an, welche variable Position in der Kette besetzt ist. Der zweite Index gibt die Aminosäure an, die sich an dieser Position befindet. Der dritte Index bestimmt die Position innerhalb der Liste. Markiert sind also die Einträge NH$_Q$[-4][K][3] und Hα$_Q$[-4][K][3].**

Aus den chemischen Verschiebungen in diesen Listen wurden nun wieder die jeweiligen Mittelwerte berechnet. Somit existierten nun für jede Aminosäure und Verschiebung ein genereller Mittelwert *V-MW$_{AS}$* und 80 weitere Mittelwerte *V-MW$_{AS}$[POS][PAS]*. Die Indices *POS* (Position) und *PAS* (potentielle Aminosäure) geben wie bei den Listen an, an welcher Position welche Aminosäure sitzt. Der Wert *NH-MW$_T$[-4][K]* ist also die mittlere chemische Verschiebung, die amidische Protonen in Threoninresten haben, wenn vier Positionen vorher in der Sequenz ein Lysinrest steht.

Durch Subtraktion des generellen Mittelwerts von diesen speziellen Mittelwerten ließ sich für jeden der betrachteten Fälle ein Inkrement berechnen. Für das obige Beispiel ist dies in Gleichung 17 verdeutlicht.

$$NHINC_T[-4][K] = NHMW_T[-4][K] - NHMW_T$$

Gleichung 17: Berechnung der Inkremente. Gezeigt ist die Ermittlung des Inkrements für amidische Protonen von Threoninresten, denen vier Positionen vorher in der Sequenz ein Lysinrest vorangeht.

Diese Inkremente können in Tabellen gespeichert und graphisch dargestellt werden (Abbildung 14).

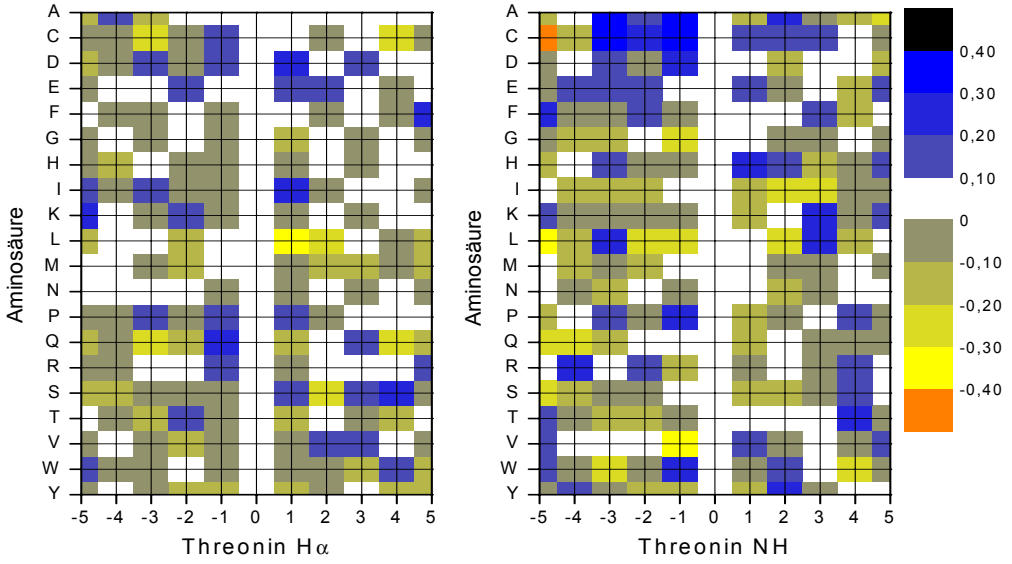


Abbildung 14: **Graphische Darstellung der Inkrementmatrizen für Threonin. Zur Vereinfachung sind Durchschnittswerte für die Inkremente dargestellt, die eigentlichen Inkrementtabellen enthalten genaue Werte. Folgt einem Threoninrest in der Sequenz beispielsweise ein Valin, so ist die chemische Verschiebung des Hα-Protons durchschnittlich um 0.0 bis 0.10 ppm verringert, während sich die chemische Verschiebung des amidischen Protons um 0.10 bis 0.20 ppm erhöht.**

Auf die gleiche Art können auch längere Sequenzfenster behandelt werden, und somit Inkrementsysteme entwickelt werden, die noch weiter

in der Sequenz entfernte Seitenketten berücksichtigen. Je größer die Fenster allerdings werden, desto weniger Werte stehen für die Aminosäuren an den Enden des Fensters zur Verfügung. Die aus diesen Werten ermittelten Inkremente werden dadurch immer unzuverlässiger.

Um nun die chemischen Verschiebungen in einer Sequenz vorherzusagen, muß diese ebenfalls mit einem Sequenzfenster in entsprechende Fragmente aufgeteilt werden. Das Problem der terminalen Reste wird, wie bei den neuronalen Netzen auch, durch Anfügen von Platzhaltern an beiden Enden der Sequenz gelöst. Diesen Platzhaltern wird kein Einfluß auf die chemische Verschiebung zugeordnet, die entsprechenden Inkremente betragen also '0'. Für jedes Fragment wird nun die zentrale Aminosäure ermittelt und der generelle Mittelwert als Startwert verwendet. Aus den entsprechenden Tabellen können nun die zugehörigen Inkremente ausgelesen und zu dem Startwert addiert werden. Bei einer Sequenzbreite von neun Aminosäuren errechnen sich die gesuchten chemischen Verschiebungen also aus den entsprechenden Startwerten und acht Inkrementen. Für ein Fragment *GHTRTHFDD* würde sich die chemische Verschiebung des zentralen Threonins also folgendermaßen berechnen lassen:

$$\delta_{NH} = NHMW_T + NHINC_T[-4][G] + NHINC_T[-3][H] + NHINC_T[-2][T] + NHINC_T[-1][R] + NHINC_T[1][H] + NHINC_T[2][F] + NHINC_T[3][D] + NHINC_T[4][D]$$

Bei dieser Methode werden die einzelnen Einflüsse isoliert voneinander betrachtet. Es wird immer nur eine variable Position berücksichtigt, möglicherweise auftretende Korrelationen werden nicht einbezogen. Es wäre beispielsweise durchaus denkbar, das eine aromatische Aminosäure direkt gefolgt von noch einer aromatischen Seitenkette einen komplett anderen Einfluß ausübt, als eine aromatische Aminosäure gefolgt von einer Aminosäure mit polarer Seitenkette.

3.6 Suchalgorithmus

Mit den beschriebenen neuronalen Netzen und dem Inkrementsystem kann die chemische Verschiebung einzelner Aminosäuren in Abhängigkeit von der Sequenz berechnet werden. Da die Genauigkeit dieser Vorhersagen für eine akkurate Spektrensimulation nicht ausreicht, ist es sinnvoller die Vorhersagen als Hilfsmittel für die Zuordnung bereits gemessener Spektren heranzuziehen. Die berechnete Lage des NH/Hα-Kreuzsignals dient dabei als Startpunkt, von dem aus das nächstliegende, tatsächlich auftretende Kreuzsignal gesucht wird. Gehört dieses zu einer Aminosäurespur, die dem gesuchtem Typ entspricht, so kann angenommen werden, daß diese Spur zu der berechneten Aminosäure gehört. In Abbildung 15 ist dieses Verfahren dargestellt. Hier wurde ein Valin aus einer Sequenz mit insgesamt drei Valinresten zugeordnet.

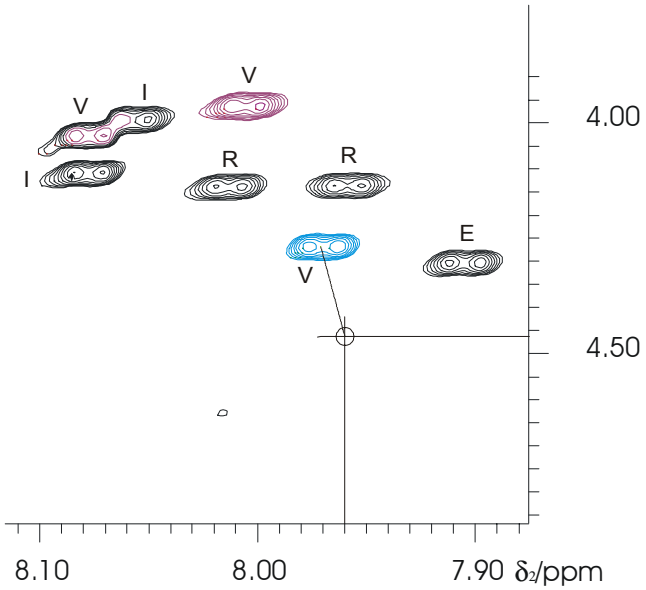


Abbildung 15: **Zuordnung einzelner Kreuzsignale mit Hilfe der berechneten Werte. Die Vorhersage für einen der drei vorhandenen Valinreste ist 7.96/4.43 ppm. Die diesen Werten nächstgelegene Valinspur (blau) wird der entsprechenden Aminosäure zugeordnet.**

Ein großes Problem hierbei ist die Beurteilung der Ergebnisse, die durch die unterschiedlichen Methoden erhalten werden. Da diese mehr

oder weniger stark voneinander abweichen, wurde nach einem Verfahren gesucht, die verschiedenen Werte miteinander in Einklang zu bringen.

Um die Qualität der von den neuronalen Netzen berechneten Verschiebungen besser einschätzen zu können, wurden die Trainingsdaten für die jeweiligen Netze zufällig in vier gleich große Datensätze aufgeteilt. Mit diesen Daten wurden vier verschiedene Netze bei ansonsten konstanten Parametern (Anzahl der Neuronen in den Schichten, Lernrate, Anzahl der Trainingszyklen usw.) trainiert.

Für die beiden relevanten Verschiebungen (NH, Hα) einer Aminosäure wurden nun mit diesen vier Netzen vier verschiedene Vorhersagen berechnet. Für die vier Werte pro Verschiebung wurde die Standardabweichung bestimmt. Aus diesen beiden Standard-abweichungen σ_{NH} und $\sigma_{H\alpha}$ wurde eine gemittelte Standardabweichung σ' berechnet. Wenn diese gemittelte Standardabweichung kleiner war als ein festgelegtes Limit L_σ, so wurden diese Vorhersagen akzeptiert. Aus den vier Werten wurde dann der Mittelwert (μ_{NH} und $\mu_{H\alpha}$) berechnet und als Vorhersage für die Zuordnung verwendet (Abbildung 16).

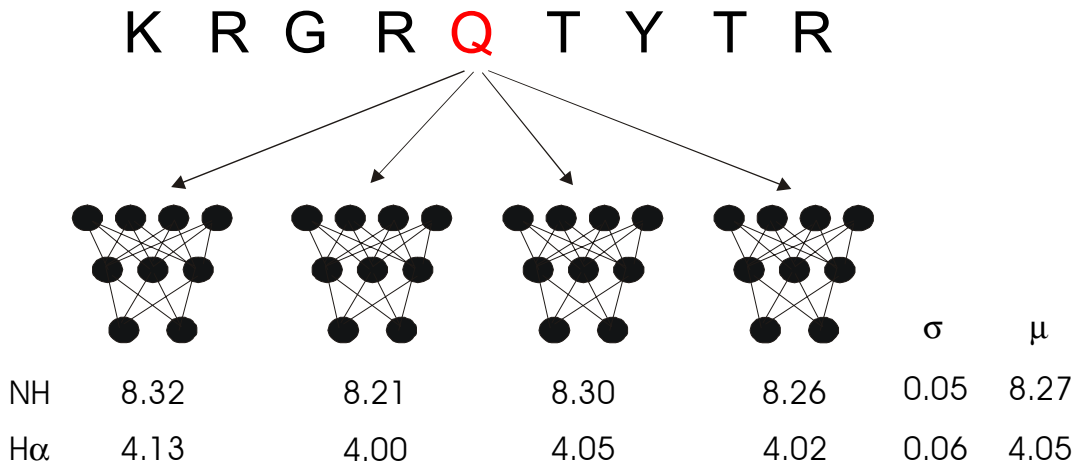


Abbildung 16: **Berechnung der chemischen Verschiebung mit vier neuronalen Netzen. Wenn die gemittelte Standardabweichung σ' über NH und Hα (hier: σ' = 0.055 ppm) kleiner als ein festgelegter Grenzwert ist, so werden die Durchschnittswerte μ_{NH} und $\mu_{H\alpha}$ als Vorhersage akzeptiert.**

Dieses Vorgehen filtert also stark voneinander abweichende Berechnungen heraus. Sequenzen, die zu solchen stark divergierenden Werten führen, scheinen von den neuronalen Netzen nur schwer klassifizierbar zu sein.

Durch schrittweises Erhöhen des Limits für σ' lassen sich so auch die einzelnen Vorhersagen nach Ihrer Qualität sortieren. Die Zuordnung der Spuren kann somit in mehreren Zyklen durchgeführt werden. Dabei werden im ersten Zyklus nur Vorhersagen benutzt, bei denen σ' unter einem minimalem Akzeptanzlimit L_σ liegt. In dieser Arbeit wurde ein Minimalwert für σ' von 0.1 ppm verwendet. Diese Vorhersagen dienen als Startpunkte im Spektrum, von denen aus das nächstliegende Kreuzsignal gesucht wird, das zu dem entsprechendem Aminosäuretyp paßt. Hier wurde die Äquivalenz einiger Aminosäuretypen, die in Tabelle 2 dargestellt ist, berücksichtigt. Wenn also ein Glutaminrest gesucht wurde, wurden sowohl Glutaminspuren als auch Glutaminsäurespuren als korrekte Zuordnungen akzeptiert. Sobald das passende Kreuzsignal gefunden wurde, wurde der gesuchten Aminosäure in der Sequenz diese Spur eindeutig zugeordnet. Für die nächsten Durchläufe stehen dann sowohl diese Spur als auch die dazugehörige Position in der Sequenz nicht mehr zur Verfügung. Damit wird gewährleistet, daß einerseits jede Spur nur einmal zugeordnet wird und andererseits für jede Aminosäure in der Sequenz auch nur eine Spur gefunden wird. Im nächsten Durchlauf wird L_σ um 0.1 ppm erhöht und erneut nur die Vorhersagen in Betracht gezogen, deren σ' unter diesem Wert liegt. Die Suche innerhalb des Spektrums wird dann wieder durchgeführt. Es werden so lange Suchzyklen durchlaufen, bis L_σ einen vorher festgelegten Maximalwert von 2 ppm erreicht hat (Abbildung 17). Die dann noch nicht gefundenen Spuren können nur durch genauere manuelle Analyse des Spektrums zugeordnet werden.

Für die Suche im Spektrum kann festgelegt werden, ob die gefundenen Spuren beliebig weit vom Startpunkt liegen können. Es wird also so lange gesucht, bis ein der Aminosäure entsprechendes Kreuzsignal gefunden

wird. Oder es wird eine maximale Distanz D_{max} vorgegeben, innerhalb der gesucht werden darf. Bildlich lässt sich dieses Verfahren mit einem runden Fenster veranschaulichen. Der Mittelpunkt dieses Fensters liegt auf der Vorhersage, und nur passende Kreuzsignale innerhalb des Fensters werden als korrekte Treffer gewertet. Durch Variation der unteren und oberen Grenze für L_σ und des Wertes für D_{max} können die Ergebnisse dieser Zuordnung beeinflußt werden.

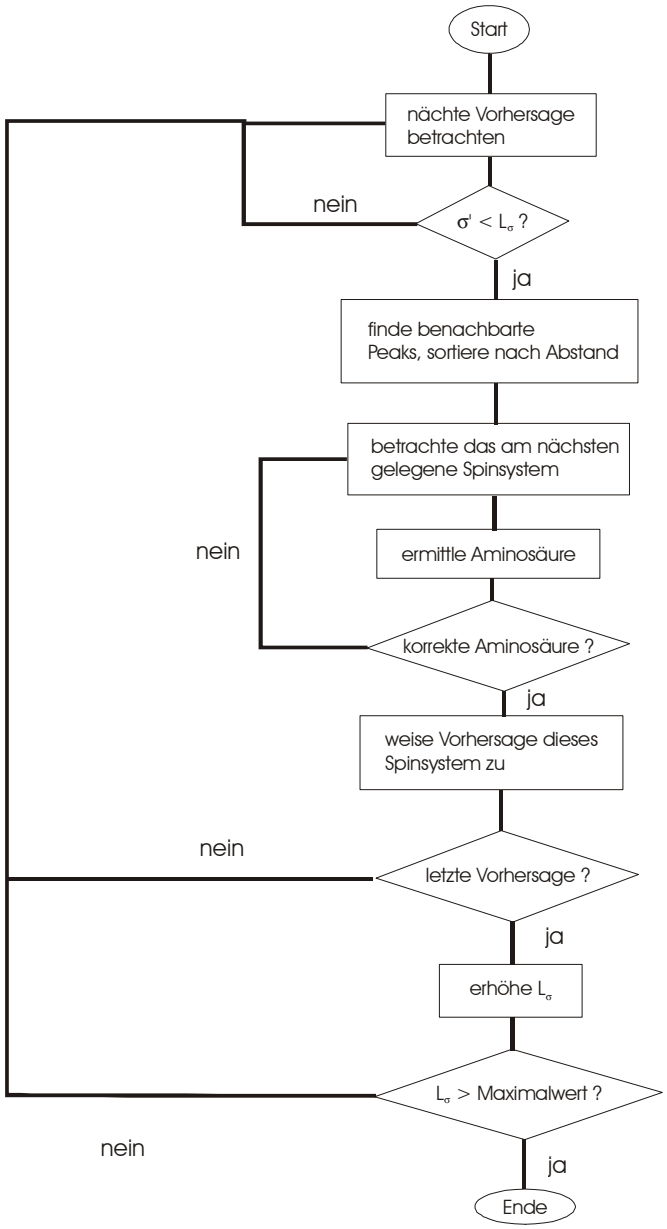


Abbildung 17: **Algorithmus für die Zuordnung der Spuren zu einzelnen Aminosäuren durch mehrere neuronale Netze.**

Jede auf diese Weise getroffene Zuordnung verfügt über zwei charakteristische Werte. Der erste ist σ', die mittlere Standardabweichung der zu Grunde liegenden Vorhersagen. Der zweite ist die Entfernung *D*, gemessen in ppm, die angibt wie weit neben dem gefundenen Kreuzsignal die Vorhersage lag. Da mit dem geschilderten Algorithmus sehr viele Kreuzsignale zugeordnet werden, wurde angenommen, daß man über diese beiden Werte die richtigen von den falschen Zuordnungen unterscheiden kann.

3.7 NOE-Validierung

Da die Unterscheidung zwischen richtigen und falschen Zuordnungen sich als äußerst schwierig erwies, wurden die Vorhersagen in einem nächsten Schritt mit NOE-Daten verifiziert. Dabei wurde besonderes Augenmerk darauf gelegt, daß Vorbereitung und Auswertung der entsprechenden NOESY-Spektren möglichst einfach gehalten wurden.

So wurde nur der NH/Hα-Bereich der zugehörigen NOESY-Spektren betrachtet. Weiterhin wurde angenommen, daß nur sequentielle NOE-Kontakte auftreten. Die einzelnen Signale wurden mittels eines in der Software *AURELIA* implementierten, automatisierten *Peak-Pickings* in diesem Bereich gesucht. Da dieses Verfahren vor allem bei stark überlagerten Signalen nicht mehr zuverlässig arbeitet, wurden noch einige Signale manuell ausgewählt. Dabei wurde darauf geachtet, daß nur subjektiv eindeutig erkennbare Signale verwendet wurden. Die komplette Aufbereitung der NOESY-Daten dauerte im Regelfall nicht länger als 15 Minuten.

Die so ermittelten NOE-Signale wurden in Form von Listen zur weiteren Verwendung gespeichert. In diesen Listen waren die einzelnen Signale durch ihre Verschiebungen in den beiden spektralen Dimensionen charakterisiert.

Da nun deutlich mehr Informationen über die Konnektivitäten innerhalb der Peptidkette zur Verfügung standen, wurden Methoden entwickelt, diese in die sequentielle Zuordnung einfließen zu lassen. Dabei

wurden bei einem Verfahren die Vorhersagen der neuronalen Netze in den Vordergrund gestellt. Bei einem anderen wurden die NOE-Kontakte als Basis für die Zuordnung gewählt.

Die erste Methode, im folgenden als *NOEV-1* bezeichnet, geht von den zugeordneten Spuren aus. Für die im Spektrum auftretenden Spuren gibt es nach der in Abschnitt 3.6 beschriebenen Analyse in den meisten Fällen eine Zuordnung zu einer diskreten Aminosäure innerhalb der Sequenz. Diese kann in der Form S_x-$P_{y(AS)}$ notiert werden, wobei S_x die Nummer der Spur angibt und $P_{y(AS)}$ die Position in der Sequenz. Der Ausdruck S_{11}-$P_{13(ARG)}$ bedeutet somit, daß Spur 11 einem Arginin an Position 13 in der Sequenz zugeordnet wurde. Jede der getroffenen Zuordnungen wurde nun überprüft. Zuerst wurde die chemische Verschiebung des α-Protons der Spur gesucht. Dann wurden aus der Liste der NOE-Signale alle Signale herausgesucht, deren Lage in der F_1-Domäne um maximal 0.01 ppm von dieser Verschiebung abwich. Zu den gefundenen Kontakten wurden die zugehörigen Verschiebungen in der F_2-Domäne ermittelt. Somit stand nun eine Liste potentieller Kontakte zu der in der Sequenz folgenden Aminosäure zur Verfügung (Abbildung 18).

Für das oben genannte Beispiel S_{11}-$P_{13(ARG)}$ hieße das, daß einer der gefundenen Kontakte dem NH/Hα-Kreuzsignal der Aminosäure an Position 14 entsprechen sollte, wenn die Zuordnung korrekt ist. Im nächsten Schritt wurde für jedes gefundene NOESY-Signal überprüft, auf welche Spuren im TOCSY-Spektrum es deutet. Falls zu diesen Spuren ebenfalls Zuordnungen existierten, wurden diese nun betrachtet. Im Idealfall wäre nun genau eine der über den NOE verbundenen Spuren der gesuchten sequentiellen Position zugeordnet. Die Hypothese S_{11}-P_{13} wäre also zum Beispiel über einen sequentiellen NOE mit der Hypothese S_9-$P_{14(ILE)}$ verbunden. In diesem Fall können beide Hypothesen als wahr angesehen werden.

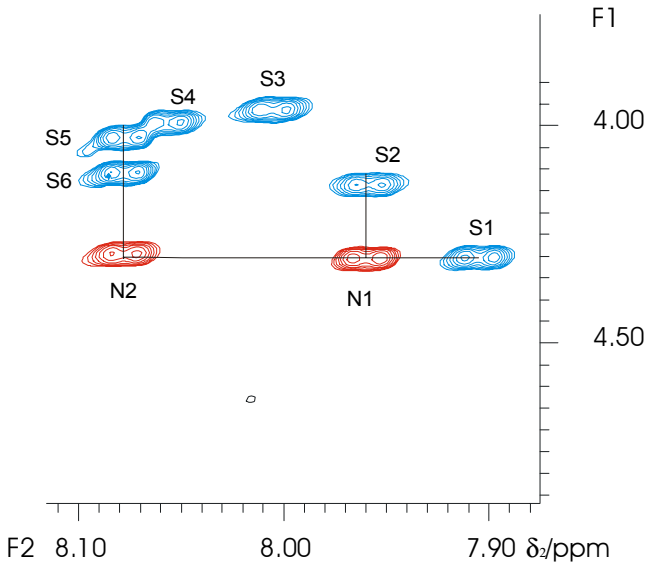


Abbildung 18: **Bestimmung zusammengehöriger Spuren mit NOE-Kontakten. Blau sind TOCSY-Signale gekennzeichnet, rot NOESY-Signale. In diesem Beispiel ist die Spur S1 über den NOE N1 mit Spur S2 verbunden. Weiterhin existiert über den NOE N2 eine Verbindung zu den Spuren S5 und S6. Daraus resultieren die Hypothesen S1-S2, S1-S5 und S1-S6 für die sequentielle Zuordnung der einzelnen Spuren. In einem realen Spektrum treten meist deutlich mehr NOE-Kontakte auf.**

War dieses nicht gegeben, so wurde überprüft, ob zumindest der Aminosäuretyp der verbundenen Spur korrekt ist. Dabei wurde die in Tabelle 2 dargestellte Gruppierung berücksichtigt. Wenn der Typ der Aminosäure mit dem der Aminosäure an der gesuchten Position übereinstimmte, so wurde die zu dieser Spur gehörige Hypothese korrigiert. Auch dieser Fall soll an einem Beispiel verdeutlicht werden. Angenommen wird eine Sequenz, die an Position 13 ein Arginin und an Position 14 ein Isoleucin beinhaltet. Die Numerierung der Spuren erfolgt willkürlich. Zu der Spur S_{11}-$P_{13(ARG)}$ werden NOE-Kontakte zu den zwei Spuren S_8 und S_{14} gefunden. Keiner dieser Spuren ist allerdings die Position 14 in der Sequenz zugeordnet. An dieser Position befindet sich ein Isoleucinrest. Die Spur S_8 ist einem Argininrest zugeordnet, fällt also aus. Die Spur S_{14} hingegen ist einem Leucinrest zugeordnet worden, der in der Sequenz an Position 16 steht. Isoleucin und Leucin sind für die Ermittlung des Aminosäuretyps zu einer Gruppe zusammengefaßt (vgl. Tabelle 2). Da das Auftreten des NOEs gegen diese Zuordnung spricht,

wird die entsprechende Hypothese korrigiert. Aus der vorherigen Zuordnung S_{14}-$P_{16(LEU)}$ wird also S_{14}-$P_{14(ILE)}$.

Wenn nicht einmal der Aminosäuretyp übereinstimmte, so wurde die betrachtete Hypothese als falsch eingestuft. Nachdem zu allen Spuren auf diese Art NOE-Kontakte gesucht worden waren, existierte ein Satz an - teilweise korrigierten – Hypothesen. Aus diesem Satz mußten nun noch mehrdeutige Zuordnungen aussortiert werden. Da dies auch nach der Validierung mit der zweiten Methode erfolgt, soll zunächst dieses, als *NOEV-2* bezeichnete Verfahren, beschrieben werden.

Bei dieser Methode wurden zunächst zu allen Spuren im Spektrum sequentielle NOEs gesucht. Dabei wurde wie im Verfahren *NOEV-1* vorgegangen. Die gefundenen Konnektivitäten können in der Form S_x-S_y beschrieben werden, d. h. Spur *x* ist über einen NOE mit Spur *y* verbunden. Im nächsten Schritt wurden die Zuordnungen dieser Spurpaare zu Aminosäuretypen überprüft, wobei auch hier die in Tabelle 2 beschriebene Vereinfachung angewandt wurde. Aus den 20 Aminosäuren werden somit nur noch zwölf Aminosäureklassen. Dies führte zu Hypothesen, die Spurpaare mit Aminosäurepaaren korrelieren. Die Hypothesen können als S_x-$S_y(R_1$-$R_2)$ notiert werden, wobei R_1 und R_2 entweder Aminosäuren im Einbuchstabencode sind oder die eingeführten Aminosäureklassen. Das Beispiel S_{11}-$S_{14}(R$-$i)$ bedeutet also, das Spur 11 und Spur 14 über einen NOE verbunden sind. Außerdem ist Spur 14 ein Isoleucin- oder Leucinrest, Spur 11 ein Argininrest zugeordnet.

Nun wurde geprüft, ob die gefundenen Aminosäurepaare überhaupt in der Sequenz vorhanden waren. Dazu wurde die Sequenz in die in Abschnitt 3.2 angesprochene vereinfachte Notation übersetzt. Da z.B. das Paar 'ei' vier verschiedene tatsächliche Aminosäurepaare (EI, QI, EL, QL) darstellt, war es durchaus möglich, daß einige Hypothesen mehrmals in der Sequenz gefunden wurden. Jede Hypothese war nun zusätzlich mit genau definierten Positionen in der Sequenz versehen, was durch die Beschreibung S_x-$S_y(P_{a(AS)}$-$P_{b(AS)})$ wiedergegeben wird. Die Indices *a(AS)* und *b(AS)* geben nun genau eine Aminosäure in der Sequenz wieder. Aus

S_{11}-S_{14}*(R-i)* könnte somit, bei entsprechender Sequenz, S_{11}-S_{14}($P_{13(R)}$-$P_{14(I)}$) werden.

Die nach diesem Test noch verbleibenden Hypothesen wurden nun mit der Zuordnung der neuronalen Netze verglichen. Alle dazu benötigten Informationen waren in den Hypothesen enthalten. Wenn im obigen Beispiel Spur 11 dem Arginin 13 und Spur 14 dem Isoleucin 14 zugeordnet war, so wurden diese beiden Vorhersagen als richtig angesehen. Wenn nur eine der beiden Vorhersagen mit den Informationen aus der NOE-Analyse übereinstimmte, so wurde die entsprechend andere korrigiert. Nur wenn keine der beiden Spuren Übereinstimmung zeigte, wurden die dazu gehörigen sequentiellen Zuordnungen als falsch gewertet.

In beiden Verfahren mußten am Ende die mehrdeutigen Zuordnungen aussortiert werden. Da immer nur isolierte Spurpaare berücksichtigt wurden, und die Zuordnungen möglicherweise noch geändert wurden, konnte es nun wieder vorkommen, daß eine Spur mehreren verschiedenen Positionen zugeordnet war. Auch der andere Fall, daß für eine Position mehrere Spuren in Frage kamen, war wieder möglich. Diese Fälle wurden am Ende des Validierungsschrittes gelöscht, so daß nur noch eindeutige Zuordnungen übrig waren. Weiterhin war es möglich, die Zuordnungen vor der Validierung mit den Zuordnungen nach der Validierung zu vergleichen. Dies lieferte ein Maß dafür, wie weit die zusätzlichen Informationen über NOE-Kontakte Einfluß auf die Ergebnisse der Vorhersagen hatten. Je weniger Zuordnungen durch die Validierung korrigiert wurden, desto besser waren die Vorhersagen der neuronalen Netze. Die NOE-Analyse diente in diesem Fall also nur als Filter, um die durch den Zuordnungsalgorithmus auftretenden Fehler zu eliminieren.

Die Unterschiede der beiden Verfahren *NOEV-1* und *NOEV-2* liegen in der unterschiedlichen Gewichtung der Vorhersagen. Im ersten Algorithmus wird die Zuordnung einer Spur a priori als wahr angenommen. Nur wenn keine über einen NOE verknüpfte, passende Spur gefunden wird, wird die Zuordnung verworfen. Im zweiten Verfahren werden zunächst, basierend auf den NOE-Daten, Fragmente konstruiert,

die mit der bekannten Sequenz in Übereinstimmung gebracht werden. Auch hier muß nur eine Zuordnung der Spuren übereinstimmen um beide Hypothesen zu validieren. Die Annahme, die der Methode *NOEV-1* zu Grunde liegt lautet:

"Wenn diese Vorhersage stimmt, dann muß ein entsprechender NOE vorhanden sein."

Für die Variante *NOEV-2* hingegen gilt der Grundsatz:

"Zwischen diesen beiden Spuren existiert ein NOE, die Vorhersagen sollten also die entsprechenden Werte haben."

Das Problem, richtige von falschen Zuordnungen zu unterscheiden, wurde somit quasi von zwei verschiedenen Seiten angegangen.

Kombiniert man die verschiedenen neuronalen Netze mit den beiden Methoden zur NOE-Analyse, so erhält man unter Umständen verschiedene Zuordnungen für die einzelnen Spuren. Verwendet man beispielsweise alle drei Kodierungsvarianten so ergeben sich pro Spur sechs Resultate, die auch voneinander abweichen können. Hier kann nun erneut ein Filtermechanismus eingebaut werden. Die Zuordnungen werden nur akzeptiert, wenn von den sechs Ergebnissen mindestens vier übereinstimmen. Auch hier kann das Limit variiert werden. Eine strengere Auswahl wäre es, wenn alle Zuordnungen identisch sein müssten.

4 Ergebnisse und Diskussion

Die bisher beschriebenen Methoden wurden mit verschiedenen Datensätzen getestet. Diese Datensätze stammten entweder aus der BMRB-Datenbank oder waren im Arbeitskreis gemessene Spektren von Peptiden und Glycopeptiden. Das Kriterium für die Auswahl der Testdaten aus der BMRB-Datenbank war in erster Linie die Länge der Sequenz. Es wurde darauf geachtet, daß die Datensätze den Bereich von ca. 20 – 120 Aminosäuren abdecken.

Die Qualität der Ergebnisse wurde für jeden Teilschritt anhand verschiedener Kriterien bewertet. Diese sind in den jeweiligen Abschnitten genauer erläutert.

4.1 Bestimmung des Aminosäuretyps

Benutzt man neuronale Netze, um den Aminosäuretyp aus dem Signalmuster zu ermitteln, so bekommt man eine eindeutige Aussage darüber, um welche Aminosäure es sich handelt. Im Idealfall ist nur das Neuron der Ausgabeschicht aktiviert, das die gesuchte Aminosäure repräsentiert. Alle anderen Ausgabeneuronen sind nicht aktiv. Aber auch in weniger eindeutigen Fällen, wenn beispielsweise Signale fehlen oder überlagert sind, kann eine Aussage getroffen werden. Dann repräsentiert das Neuron, das den höchsten Wert liefert, die gesuchte Aminosäure. In seltenen Fällen kann es vorkommen, das sich die Werte mehrerer Ausgabeneuronen kaum unterscheiden. In solchen Fällen kann also keine eindeutige Antwort des neuronalen Netzes ermittelt werden.

Um die eingesetzten neuronalen Netze zu testen wurden, die in Tabelle 8 aufgelisteten Datensätze aus der BMRB-Datenbank benutzt.

Datensatz	*Protein*	*Länge der Sequenz*
bmr1700.str	Sarafotoxin S6b	21
bmr1728.str	Endothelin	21
bmr3449.str	Parathyroidhormon	34
bmr162.str	Transforming growth factor	50
bmr1495.str	Soybean proteinase inhibitor	71
bmr2065.str	Phospholipidtransferprotein	90
bmr1766.str	Neocarzinostatin	113

Tabelle 8: Die im Testdatensatz enthaltenen Einträge der BMRB-Datenbank.

Die Auswertung der Testdaten erweist sich für diese Problemstellung als einfacher Vergleich des ermittelten Aminosäuretyps mit dem bekannten Ergebnis. Stimmen diese beiden überein, so war das Netz erfolgreich. Anderenfalls liegt ein Fehler vor. Die Leistung der neuronalen Netze kann also als prozentualer Anteil an richtig zugeordneten Mustern vom gesamten Testdatensatz gemessen werden.

4.1.1 Ergebnisse der statistischen Kodierung

Für die statistische Kodierung wurden nach dem in Abschnitt 3.2.1 beschriebenen Verfahren Muster für das Training der neuronalen Netze erzeugt. Zunächst wurden drei Trainingssätze erzeugt, in denen für jede der 20 Aminosäuren 100, 200 und 400 Muster enthalten waren. Die Gesamtzahl der Muster betrug somit 2000, 4000 und 8000. Diese Datensätze wurden für das Training von drei verschiedenen neuronalen Netzen herangezogen, im folgenden *AS-s100, AS-s200* und *AS-s400* bezeichnet. Die Architektur dieser Netze war identisch. Sie bestanden aus 650 Eingabeneuronen, welche den Bereich von 0.00 bis 6.49 ppm abbildeten. Die Anzahl der versteckten Neuronen war 50, die Anzahl von Ausgabeneuronen wurde gemäß Tabelle 1 auf 20 gesetzt. Die anfängliche Lernrate betrug 2.0. Während des Trainings wurde diese alle 500 Zyklen um 0.02 verringert. Als Abbruchkriterium für das Training diente der RMS-Wert. Wenn dieser 0.001 unterschritt wurde das Training beendet,

spätestens aber nach 50000 Zyklen. Für das Netz *AS-s100* wurde der RMS nach 16000 Zyklen klein genug um das Training zu beenden. Die Netze *AS-s200* und *AS-s400* hingegen wurden über 50000 Zyklen trainiert und erreichten finale RMS-Wert von 0.004 bzw. 0.008. Nach Abschluß des Trainings wurden die Netze mit den Testdaten geprüft.

Die Muster im Testdatensatz wurden aus den in Tabelle 8 aufgezählten Einträgen der BMRB erzeugt. Für jede Aminosäure in diesen Datensätzen wurden die chemischen Verschiebungen zwischen 0.00 und 6.49 ppm ermittelt und in Muster umgewandelt. Für jeden der sieben Datensätze wurde eine eigene Musterdatei erstellt. Diese wurde den fertig trainierten Netzen präsentiert und die Ausgabe zu jedem Muster mit der erwarteten, korrekten Ausgabe verglichen.

4.1.1.1 Einfache Auswertung

In Tabelle 9 sind die Ergebnisse der drei Netze dargestellt. Angegeben ist der prozentuale Anteil an korrekt zugeordneten Aminosäureresten pro getestetem Datensatz.

Netz	*bmr1700 (20 AS)*	*bmr1728 (21 AS)*	*bmr3449 (33 AS)*	*bmr162 (48 AS)*	*bmr1495 (63 AS)*	*bmr2065 (84 AS)*	*bmr1766 (108 AS)*
AS-s100	20	14	21	37	21	39	38
AS-s200	25	24	36	35	37	32	56
AS-s400	20	29	27	35	29	38	56

Tabelle 9: Ergebnisse der drei Netze *AS-s100, AS-s200* und *AS-s400*. Angegeben ist die prozentuale Erfolgsrate mit der die Aminosäuren korrekt klassifiziert wurden.

Hierbei wurde eine Aminosäure als korrekt erkannt gewertet, wenn das entsprechende Ausgabeneuron den höchsten Wert aller 20 Ausgabeneuronen hatte. Es zeigt sich, daß mit 200 Mustern pro Aminosäure bei dieser Kodierung insgesamt die besten Ergebnisse erzielt werden, wobei die Gesamtleistung allerdings noch sehr unbefriedigend ist.

4.1.1.2 Gruppierte Auswertung

Da, wie bereits erwähnt, einige Aminosäuren ein sehr ähnliches Signalmuster aufweisen, wurden diese für eine weitere Auswertung zu Gruppen zusammengefaßt (vgl. Tabelle 2). Eine Aminosäure wurde nun schon als richtig erkannt eingeordnet, wenn nur eines der zugehörigen Ausgabeneuronen den höchsten Wert aufwies. Wie in Tabelle 10 ersichtlich, steigt die Leistungsfähigkeit dadurch deutlich. Auch hier ist die prozentuale Erkennungsrate angegeben. Diese liegt allerdings immer noch nicht auf dem erhofften Niveau von 80 - 90 %. Das Netz *AS-s200* zeigt wieder die beste Leistung.

Netz	*bmr1700 (20 AS)*	*bmr1728 (21 AS)*	*bmr3449 (33 AS)*	*bmr162 (48 AS)*	*bmr1495 (63 AS)*	*bmr2065 (84 AS)*	*bmr1766 (108 AS)*
AS-s100	50	33	42	65	46	51	47
AS-s200	65	48	64	65	62	51	67
AS-s400	50	43	42	60	54	57	66

Tabelle 10: Ergebnisse der drei Netze *AS-s100* , *AS-s200* und *AS-s400*. Aminosäuren, die ein ähnliches Signalmuster aufweisen, wurden für die Auswertung zu Gruppen zusammengefaßt.

4.1.1.3 Gestaffelte Netze

Die Gruppierung ähnlicher Aminosäuren lässt sich auch durch eine abgeänderte Netzarchitektur verwirklichen. Statt 20 Ausgabeneuronen werden nun nur noch zwölf Neuronen verwendet. Von diesen zwölf Neuronen geben sieben die Aminosäuren wieder, die in keiner Gruppe enthalten sind. Die anderen fünf kodieren die fünf gebildeten Gruppen. Je nachdem, welches der zwölf Neuronen den maximalen Wert erhält, gehört das Eingabemuster also entweder zu einer der Gruppen oder zu einer einzelnen Aminosäure. Für jede Gruppe kann nun noch ein weiteres Netz trainiert werden, daß darauf spezialisiert wird, die einzelnen Typen innerhalb einer Gruppe zu differenzieren. Das zweite Netz benötigt dabei nur noch zwei Ausgabeneuronen, da in den meisten Gruppen nur zwei

Aminosäuren enthalten sind. Eine Ausnahme stellt die Gruppe *c* dar, die Cystein und die aromatischen Aminosäuren beinhaltet. Da die aromatischen Aminosäuren in TOCSY-Spuren jedoch absolut nicht unterscheidbar sind, können diese zu einer virtuellen Aminosäure zusammengefaßt werden. Das zweite Netz liefert also nur die Aussage, ob es sich um ein Cystein oder eine der vier aromatischen Aminosäure handelt.

Auch für diese Netze wurden die Muster im ersten Ansatz statistisch erzeugt. Für das erste Netz mit zwölf Ausgabeneuronen, im folgenden als *AS-s12out* bezeichnet, wurden 200 Muster pro Aminosäure berechnet. Die sonstige Architektur und das Trainingsprotokoll stimmten mit dem Netz *AS-s200* überein. Nach 17000 Zyklen wurde das Abbruchkriterium erfüllt und das Training beendet. Die Auswertung erfolgte wie bei den Netzen *AS-s100*, *AS-s200* und *AS-s400* nach Datensätzen getrennt und ist in Tabelle 11 dargestellt.

Netz	*bmr1700 (20 AS)*	*bmr1728 (21 AS)*	*bmr3449 (33 AS)*	*bmr162 (48 AS)*	*bmr1495 (63 AS)*	*bmr2065 (84 AS)*	*bmr1766 (108 AS)*
AS-s12out	62	52	41	60	51	58	73

Tabelle 11: Ergebnisse des Netzes *AS-s12out*. Die Gruppierung ähnlicher Aminosäuren erfolgte hier direkt auf der Ausgabeschicht. Dabei hatten beispielsweise die beiden Aminosäuren Leucin und Isoleucin ein gemeinsames Ausgabeneuron.

Für die spezialisierten Netze wurden pro Aminosäure 1200 Muster erzeugt. Die versteckte Schicht bestand bei jedem der fünf verschiedenen Netze aus 20 Neuronen, die Ausgabeschicht aus zwei Neuronen. Je nach der zu differenzierenden Gruppe werden diese Netze als *AS-sCAr*, *AS-sEQ*, *AS-sDN*, *AS-sIL* oder *AS-sKR* bezeichnet. Für die Auswertung dieser Netze wurde für die sieben Testdatensätze ermittelt, wie oft jede der in den Gruppen enthaltenen Aminosäuren insgesamt auftrat. Ausgehend von dieser Summe wurde geprüft, wie oft die entsprechende Aminosäure korrekt zugeordnet wurde und eine prozentuale Erkennungsrate berechnet (Tabelle 12). Auffällig ist hier die Erkennungsrate für

Argininreste. Auf den ersten Blick erscheinen die 100 % beeindruckend. Dabei ist jedoch zu beachten, daß zwar jedes Arginin richtig erkannt wurde, der Umkehrschluß allerdings nicht gilt. Nicht alles, was als Arginin klassifiziert wurde ist tatsächlich die entsprechende Aminosäure. In elf Fällen, die das neuronale Netz Arginin zuordnete, lag tatsächlich ein Lysinrest vor.

Auch dieser Ansatz erscheint nicht sehr erfolgversprechend, da schon die erste Einordnung mit dem Netz *AS-s12out* tendenziell schlechter verläuft als bei dem Netz *AS-s200* mit gruppierter Auswertung. Ausgehend von dieser unzuverlässigen Klassifikation muß nun in vielen Fällen ein zweites neuronales Netz herangezogen werden, dessen Ergebnisse ebenfalls unbefriedigend sind.

Aminosäure	*Summe*	*Richtig*	*Erkennungsrate*
Cystein	40	12	30
Aromatisch	41	31	76
Asparagin	19	5	26
Aspartat	32	21	66
Glutamin	16	3	19
Glutamat	13	8	62
Leucin	25	16	64
Isoleucin	13	5	38
Lysin	15	4	27
Arginin	14	14	100

Tabelle 12: Ergebnisse der Netze *AS-sCAr*, *AS-sEQ*, *AS-sDN*, *AS-sIL* und *AS-sKR*. In den Testdatensätzen kommt Cystein z.B. 40 mal vor. In nur elf dieser 40 Fälle konnte das Netz *AS-sCAr* das Cystein von den aromatischen Aminosäuren unterscheiden.

Zusammenfassend lässt sich sagen, daß die Methode der statistischen Mustererzeugung zu keinen sinnvollen Ergebnissen führt. Die bereits angesprochenen Nachteile (Verlust der Information über Einflüsse

innerhalb der Seitenkette) können durch die hohe Anzahl an erzeugbaren Mustern nicht kompensiert werden.

4.1.2 Ergebnisse der breiten Kodierungen

Da mit rein statistisch erzeugten Mustern keine weiter verwertbaren neuronalen Netze trainiert werden konnten, wurden nun aus den in der BMRB-Datenbank enthaltenen Spektren Muster generiert. Nach dem Aussortieren der für Testzwecke ausgewählten Spektren (vgl. Tabelle 8) blieben 1357 Datensätze übrig. In Abschnitt 3.2.2 sind die Kriterien und Methoden der Mustererzeugung bereits beschrieben worden.

Für die breite Kodierung wurden verschiedene Netze trainiert, welche sich zunächst in der virtuellen Linienbreite unterschieden. Diese wurde in drei verschiedenen Versuchen auf fünf, zehn und 20 Neuronen gesetzt. Mit der Ausnahme von Tryptophan wurden für alle Aminosäuren jeweils 400 Muster aus den Spektren kodiert. Tryptophan war in den 1357 benutzen Datensätzen nur 390 mal enthalten. Auf der Eingabeschicht wurde der Bereich von 0.00 bis 6.49 ppm mit einer Auflösung von 0.01 ppm kodiert, dazu waren 650 Eingabeneuronen nötig. Die versteckte Schicht bestand aus 100 Neuronen, die Ausgabeschicht repräsentierte mit 20 Neuronen die verschiedenen Aminosäuren. Die Netze sollen als *AS-b5*, *AS-b10* und *AS-b20* bezeichnet werden. Das Training lief mit kontinuierlich abnehmender Lernrate über maximal 50000 Epochen (vgl. Abschnitt 4.1.1).

4.1.2.1 Einfache und gruppierte Auswertung

Auch die Ergebnisse dieser Netze wurden mit und ohne die Gruppierung ähnlicher Aminosäuren betrachtet.

In Tabelle 13 sind die entsprechenden Erkennungsraten dargestellt. Es zeigt sich eine deutliche Verbesserung im Vergleich zur statistischen Kodierung. Vor allem bei einer gruppierten Auswertung sind die Erkennungsraten im Mittel um ca. 25 % besser. In Einzelfällen zeigt sich eine Verbesserung um fast 50 %.

Weiterhin zeigte sich, das eine Kodierung, die zehn benachbarte Neuronen mit aktiviert, die besten Ergebnisse liefert. Die durchschnittliche Erkennungsrate für dieses Netz liegt bei 82 %. Diese Genauigkeit kann bei bekannter Sequenz bereits ausreichen, um die Spuren größtenteils automatisch zu klassifizieren.

Netz	*bmr1700 (20 AS)*	*bmr1728 (21 AS)*	*bmr3449 (33 AS)*	*bmr162 (48 AS)*	*bmr1495 (63 AS)*	*bmr2065 (84 AS)*	*bmr1766 (108 AS)*
Ungruppiert							
AS-b5	45	48	61	52	41	52	49
AS-b10	50	57	70	63	46	54	49
AS-b20	45	52	58	48	52	51	52
Gruppiert							
AS-b5	75	81	91	71	78	71	63
AS-b10	85	90	88	88	79	79	65
AS-b20	75	81	88	77	84	67	67

Tabelle 13: Ergebnisse der breiten Eingabekodierung. Beide Auswertemethoden sind hier zusammengefasst. Angegeben ist die prozentuale Erkennungsrate.

4.1.2.2 Gestaffelte Netze

Auch für diese Muster wurde der Ansatz der gestaffelten Netze überprüft. Es wurde zunächst also ein Netz mit zwölf Ausgabeneuronen trainiert, das die Aminosäuren in Klassen einteilen sollte. Im Anschluß daran sollten die gruppierten Aminosäuren durch spezialisierte Netze mit jeweils zwei Ausgabeneuronen genau klassifiziert werden. Für diese Netze wurde eine virtuelle Linienbreite von fünf Neuronen festgelegt. Die Anzahl der versteckten Neuronen wurde auf 50 gesetzt. Die Anzahl der Muster pro Aminosäure betrug 400 bzw. 390 für Tryptophan. Das Trainingsprotokoll wurde nicht verändert. Das erste Netz wird als *AS-b12out* bezeichnet, die Netze mit zwei Ausgabeneuronen entsprechend *AS-bCAr*, *AS-bDN*, *AS-bEQ*, *AS-bIL* und *AS-bKR*. Die Resultate des Netzes *AS-b12out* sind in Tabelle 14 zusammengefasst.

Auch hier ist, abgesehen von dem Datensatz *bmr1766*, eine deutliche Verbesserung im Vergleich zur statistischen Kodierung sichtbar. Im Vergleich zur gruppierten Auswertung des Netzes *AS-5b* lässt sich kein eindeutiger Trend feststellen. Manche Datensätze werden deutlich besser klassifiziert, andere hingegen schlechter.

Netz	*bmr1700 (20 AS)*	*bmr1728 (21 AS)*	*bmr3449 (33 AS)*	*bmr162 (48 AS)*	*bmr1495 (63 AS)*	*bmr2065 (84 AS)*	*bmr1766 (108 AS)*
AS-b12out	70	90	76	81	76	71	64

Tabelle 14: Ergebnisse des auf Erkennung von Aminosäuregruppen trainierten Netzes *AS-b12out*. Die Linienbreite bei der Kodierung betrug fünf Neuronen.

Die Zuordnung der zu Gruppen zusammengefaßten Aminosäuren ist im allgemeinen mit der breiten Kodierung ebenfalls besser durchführbar. Vor allem die Erkennungsraten für Isoleucin, Arginin und aromatische Aminosäuren zeigen eine deutliche Verbesserung (Tabelle 15).

Aminosäure	*Summe*	*Richtig*	*Erkennungsrate*
Cystein	40	16	40
Aromatisch	41	37	90
Asparagin	19	7	37
Aspartat	32	17	53
Glutamin	16	5	31
Glutamat	13	9	69
Leucin	25	17	68
Isoleucin	13	12	92
Lysin	15	13	87
Arginin	14	10	71

Tabelle 15: Ergebnisse der Netze *AS-bCAr*, *AS-bDN*, *AS-bEQ*, *AS-bIL* und *AS-bKR*. Es sind signifikante Verbesserungen zur statistischen Kodierung sichtbar.

Die mit dieser Eingabekodierung trainierten Netze sind in der Lage, die Klasse der gesuchten Aminosäure mit einer Verlässlichkeit von 70 – 90 % vorherzusagen. Da bei der Spektrenanalyse zusätzlich die Sequenz des Peptids zur Verfügung steht, können offensichtlich falsche Zuordnungen leicht erkannt und entfernt werden. Somit ist es möglich, jeder Spur zumindest eine der zwölf Aminosäureklassen vergleichsweise schnell zuzuordnen.

4.1.3 Ergebnisse der generierenden Kodierung

Als dritte Variante wurde eine Kodierung getestet, die ausgehend von den gemessenen Verschiebungen eines Spektrums neue Muster erzeugt. Dazu werden sämtliche Signale einer Spur um einen zufälligen Betrag in eine zufällige Richtung verschoben. Die maximal zulässige Verschiebung und die Anzahl der zusätzlich erzeugten Muster pro Spur konnten dabei variiert werden. Jedes Signal wurde jedoch wie in der statistischen Kodierung nur durch ein Neuron dargestellt.

In diesen Versuchen wurden aus jedem realen Muster durch Verschiebung drei weitere Muster erzeugt. Pro Aminosäure wurden 400 Muster zum Training verwendet, von denen somit jeweils 100 auf tatsächlichen Meßwerten beruhten. Es wurden drei verschiedene Netze trainiert, bei denen die maximal zulässige Signalverschiebung ± 0.05, ± 0.10 und ± 0.20 ppm betrug. Die Architektur und das Trainingsprotokoll entsprachen den Parametern der breit kodierten Netze (650 Eingabeneuronen, 100 versteckte Neuronen, 20 Ausgabeneuronen, variable Lernrate über 50000 Zyklen). Die Netze werden als *AS-g5*, *AS-g10* und *AS-g20* bezeichnet.

4.1.3.1 Einfache und gruppierte Auswertung

Die Netze wurden mit den Testdatensätzen geprüft und die Erkennungsraten ohne und mit Gruppierung ähnlicher Aminosäuren ermittelt. Diese sind in Tabelle 16 aufgezeigt.

Da diese Erkennungsraten unter denen der Netze mit breiter Eingabekodierung liegen, wurde auf eine weitere Ausarbeitung dieser

Methode verzichtet. Der Ansatz der zwei hintereinander gestaffelten Netze wurde für diese Kodierung nicht mehr untersucht.

Netz	*bmr1700 (20 AS)*	*bmr1728 (21 AS)*	*bmr3449 (33 AS)*	*bmr162 (48 AS)*	*Bmr1495 (63 AS)*	*bmr2065 (84 AS)*	*bmr1766 (108 AS)*
Ungruppiert							
AS-g5	15	33	55	48	38	48	44
AS-g10	25	48	36	38	43	40	42
AS-g20	35	29	52	42	30	40	48
Gruppiert							
AS-g5	60	62	76	67	68	69	54
AS-g10	65	76	55	73	71	61	53
AS-g20	55	52	61	63	60	51	61

Tabelle 16: Resultate der generierenden Kodierung. Gruppierte und ungruppierte Auswertung sind zusammengefasst.

4.1.4 Zusammenfassung der Ergebnisse zur Aminosäurebestimmung

Die Leistung der neuronalen Netze zur Bestimmung des Aminosäuretyps ist stark von der Art der Mustererzeugung abhängig. Eine nur auf statistischen Daten beruhende Methode kann die realen Verhältnisse offensichtlich nicht gut abbilden. Eine Erklärung hierfür ist, daß Faktoren, die auf die ganze Seitenkette einer Aminosäure wirken, in die so erhaltenen Muster nicht eingehen.

Dieses Problem tritt auch in abgeschwächter Form bei der generierenden Kodierung auf. Das ursprüngliche Muster spiegelt die realen Verhältnisse wieder. Die zufällige Variation dieser Signale innerhalb definierter Grenzen kann nun ebenfalls dazu führen, das der im Muster enthaltene Zusammenhang zwischen den chemischen Verschiebungen einzelner Seitenkettenprotonen abgeschwächt oder aufgehoben wird. Dies zeigt sich auch daran, daß mit Erhöhung der tolerierten Variation das Netz tendenziell schlechtere Erkennungsraten liefert. So zeigt das Netz AS-

g5 im Durchschnitt eine Erkennungsrate von 65 %. Das Netz AS-g20, das mit einer Variation von ± 0.2 ppm trainiert wurde, weist nur noch eine durchschnittliche Erkennungsrate von 57 % auf.

Die besten Ergebnisse liefern Netze, bei denen die Signale mit einer linear abfallenden Intensität über mehrere benachbarte Neuronen kodiert sind. Diese Darstellung sollte nicht die tatsächliche Signalform oder Intensitäten wiedergeben. Vielmehr sollte sie die Wahrscheinlichkeit wiederspiegeln, mit der Signale abhängig von den Aufnahmebedingungen leicht verschoben auftreten können. Ein Vorteil dieser Methode ist, daß mehr Neuronen auf der Eingabeschicht aktiviert werden. Somit stehen mehr Informationen zur Verfügung. Muster, die auf sehr ähnlichen Spektren beruhen, werden damit auch für das neuronale Netz ähnlicher kodiert. Um die Ähnlichkeit zweier Muster zu beurteilen, kann der Tanimotokoeffizient[80,81] herangezogen werden. Streng genommen wird dieser Wert verwendet, um Muster, bei denen die Neuronen nur Werte von ’0’ oder ’1’ annehmen, zu vergleichen. Der Tanimotokoeffizient ist nach Gleichung 18 definiert. Für identische Muster nimmt er einen Maximalwert von ’1’ an.

$$TC_{ij} = N_{ij} / (N_i + N_j - N_{ij})$$

Gleichung 18: **Definition des Tanimotokoeffizienten zum Vergleich von zwei Mustern *i* und *j*. Zwei genau gleiche Muster liefern einen Koeffizienten von ’1’.**
***N*$_{ij}$: Anzahl an Neuronen, die in beiden Mustern gesetzt sind.**
***N*$_i$, *N*$_j$: Anzahl an Neuronen, die in Muster *i* bzw. *j* gesetzt sind.**

Um diesen Koeffizienten für die breite Kodierung zu berechnen, wird vereinfachend angenommen, daß sämtliche Neuronen, die mit einem Wert belegt sind, als gesetzt gelten. Dieser Wert muß nun nicht notwendigerweise '1' sein. Um den Einfluss der breiten Kodierung auf die Ähnlichkeit der Muster darzustellen, soll der Tanimotokoeffizient hier für einen exemplarischen Fall berechnet werden. Für dieses Beispiel sollen zwei Muster miteinander verglichen werden, bei denen sich ein Signal nur um 0.01 ppm unterscheidet. Weiterhin soll angenommen werden, das

beide Muster *i* und *j* den gleichen Aminosäuretyp als Ausgabe haben. In Tabelle 17 sind die entsprechenden Muster dargestellt.

Muster	*breite Kodierung*													
I	0	0	0.17	0.33	0.5	0.67	0.83	1	0.83	0.67	0.5	0.33	0.17	0
J	0	0.17	0.33	0.5	0.67	0.83	1	0.83	0.67	0.5	0.33	0.17	0	0
Muster	*normale Kodierung*													
I	0	0	0	0	0	0	0	1	0	0	0	0	0	0
J	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Tabelle 17: Darstellung zweier Signale, die sich um 0.01 ppm unterscheiden, in zwei Kodierungsvarianten. Grau unterlegt sind die Neuronen, die in beiden Mustern aktiv sind (N_{ij} in Gleichung 18). Der Tanimotokoeffizient für die breite Kodierung beträgt '0.83', der für die normale Kodierung '0'.

Für die breite Kodierung ergibt sich unter Berücksichtigung der getroffenen Vereinfachung ein Tanimotokoeffizient von '0.83'. Für eine Kodierung, die nur ein Neuron pro Signal verwendet, ist der Tanimotokoeffizient hingegen '0'. Zwei Muster, die die gleiche Aminosäure darstellen sollen, und sich nur minimal unterscheiden, werden bei der breiten Kodierung also als sehr ähnlich erkannt. Bei der einfachen Kodierung hingegen wird überhaupt keine Ähnlichkeit festgestellt. Der Vorteil der breiten Kodierung ist bei zehn benachbarten angeregten Neuronen am größten. Bei 20 Neuronen überwiegen die Überlappungseffekte zwischen den Signalen, so daß auch Muster die verschiedene Aminosäuren kodieren, einander zu ähnlich werden.

4.2 Sequentielle Zuordnung

Die neuronalen Netze, die für die genaue sequentielle Zuordnung der einzelnen Spuren eingesetzt werden sollten, geben als Ausgabewerte die chemischen Verschiebungen der amidischen und Hα-Protonen von Aminosäuren. Es wird also keine einfache ja/nein-Antwort wie für die Spurzuordnung gegeben, sondern ein quantitativer Wert. Die Qualität

dieses vorhergesagten Wertes kann durch Vergleich mit bekannten Zielwerten und Ermittlung des auftretenden Fehlers beurteilt werden. Durch eine statistische Analyse der Fehler für den Testdatensatz lassen sich verschiedene Parameter berechnen. So kann ein Maß für die Verläßlichkeit dieser neuronalen Netze gefunden werden. Mit diesem können die Vorhersagen für neue, unbekannte Fälle eingeordnet werden.

Auch für die Auswertung der Netze für die sequentielle Zuordnung wurde ein Testdatensatz erstellt. Hier waren Sequenzlänge und Sekundärstrukturmotive die Auswahlkriterien. Der Datensatz deckt Sequenzlängen von 18 bis 162 Aminosäuren ab. Die darin enthaltenen Proteine und Peptide enthalten α-helicale Bereiche, β-Faltblattstrukturen und ungeordnete *random coil* Abschnitte. Die Daten über die Sekundärstruktur wurden aus der *Brookhaven Protein Data Base* ermittelt. In Tabelle 18 sind die verwendeten Datensätze und die darin vorherrschenden Strukturmotive aufgeführt.

Um zu verhindern, daß in Trainings- und Testdatensatz nahezu identische Muster stehen, wurde ermittelt ob zu den in Tabelle 18 aufgeführten Proteinen noch weitere Einträge in der Datenbank vorhanden waren. Diese wurden gegebenenfalls aussortiert und bei der Erzeugung der Trainingsdaten nicht verwendet.

Es wurden verschiedene Netze trainiert, die sich in der Art der Eingabekodierung und in der Anzahl der Ausgabeneuronen unterschieden. Ein weiterer Ansatz war, für jede der 20 natürlichen Aminosäuren ein eigenes, spezialisiertes Netz zu trainieren. Im Gegensatz zu diesen Netzen standen solche, die die chemischen Verschiebungen jeder Aminosäure vorhersagen sollten. Zwei neun Reste lange Sequenzfragmente mit verschiedenen Aminosäuren in der zentralen Position würden im ersten Fall von zwei verschiedenen Netzen berechnet werden. Im zweiten Fall käme für jedes Fragment das gleiche Netz zum Einsatz.

Protein	*BMRB Nr.*	*PDB ID*	*Länge*	*Struktur*
Nucleocapsidprotein[82]	1656	1ncp	18	R
Endothelin[83]	1728	1edn	21	R
Bovine seminal fluid protein[84]	1474	1pdc	45	R
Echistatin[85]	2061	2ech	49	R
Phospholipidtransferprotein[86]	2065	1lpt	90	R / B
Neocarzinostatin[87]	1766	1neo	113	R / B
Sarafotoxin S6b[88]	1700	1srb	21	A
Enhancerbinding Protein[89]	1336	4znt	30	A
Parathyroidhormon[90]	1666	1hph	34	A
Cytochrom b562[91]	1672	1apc	106	A
Parvalbumin[92]	144	2pas	109	A
Transforming growth factor[93]	162	4tgf	50	B
Complementfactor H[94]	1479	1hcc	59	B
Soybean proteinase inhibitor[95]	1495	2bbi	71	B
Glucopermease[96]	1663	1gpr	162	B
Male associated protein[97]	1500	5znt	30	A / B
Bull seminal inhibitor[98]	146	1bus	57	A / B
Phosphocarrierprotein[99]	2060	1hdn	85	A / B
Flavodoxin[100]	1580	1rcf	169	A / B

Tabelle 18: **Zusammensetzung des Testdatensatzes für die sequentielle Zuordnung. Angegeben sind die Länge der Sequenz, die Bezeichnung der Proteine aus den *BMRB*-Datensätzen, die Nummer des zugehörigen Datensatzes, die vorherrschenden Strukturmotive und die Bezeichnungen der zugehörigen Einträge in der *PDB*-Datenbank. Bei den Strukturmotiven steht A für α-Helices, B für β-Faltblatt und R für ungeordnete *random coil* Strukturen.**

Der Verlauf des Trainings und die Leistungsfähigkeit des neuronalen Netzes können aufgrund der anfallenden Datenmengen am besten graphisch dargestellt werden. Dazu muß zunächst eine möglichst aussagekräftige Kenngröße gefunden werden. Für die folgende Auswertung wurde die Standardabweichung der auftretenden Fehler

gewählt. Die Standardabweichung ist ein Maß dafür, wie weit die Verteilung einer bestimmten Größe um den Mittelwert streut. Um sie zu bestimmen, werden für jedes Muster im Trainings- und Testsatz die Fehler berechnet (Gleichung 3). Diese Fehler werden durch Umkehrung der in Abschnitt 3.4 dargestellten Transformation zunächst in ppm umgerechnet. Aus der Verteilung dieser Fehler lässt sich die Standardabweichung nach Gleichung 19 ermitteln.

$$\sigma = \sqrt{\frac{n\sum \varepsilon^2 - \left(\sum \varepsilon\right)^2}{n^2}}$$

Gleichung 19: Berechnung der Standardabweichung σ.

***n*: Anzahl der Muster**

***ε*: Fehler in ppm**

Eine solche Verteilung ist exemplarisch in Abbildung 19 gezeigt. Die dazu gehörige Normalverteilung wurde nach Gleichung 12 berechnet.

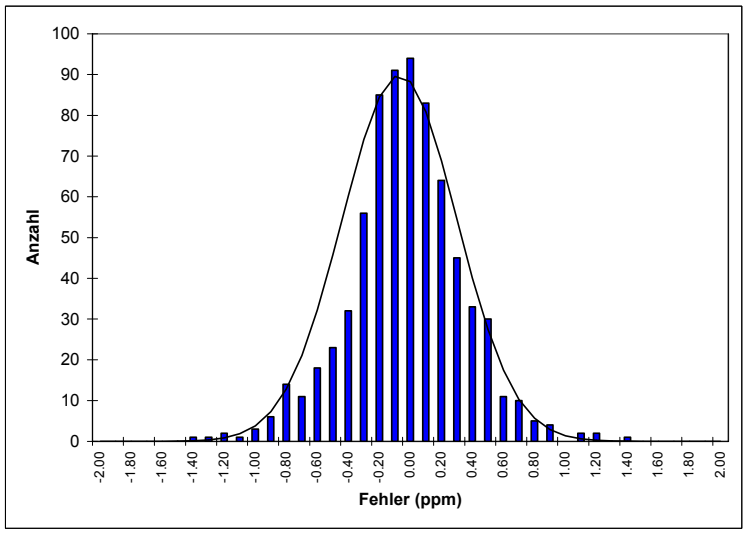


Abbildung 19: **Beispiel für eine Fehlerverteilung. Die blauen Balken sind die tatsächlich auftretenden Fehler. Die schwarze Linie stellt die Normalverteilung dar, die sich mit dem dazugehörigem Mittelwert $\mu = -0.07$ ppm und der Standardabweichung $\sigma = 0.37\ ppm$ ergibt.**

Da hier die Fehlerverteilung analysiert wird, ist ein möglichst kleiner Wert für die Standardabweichung angestrebt. Je geringer diese ist, desto kleiner ist auch der zu erwartende Fehler der Vorhersagen.

Die Ausgabewerte und Fehler für die Trainingsmuster eines neuronalen Netzes können während des Trainings gespeichert werden. Dabei kann diese Standardabweichung ermittelt werden. Die Auftragung dieser Werte gegen die Anzahl an Trainingszyklen liefert ein Bild über Entwicklung des neuronalen Netzes während des Lernvorgangs. Mit den Testmustern kann entsprechend verfahren werden, so daß ein Eindruck über die Leistungsfähigkeit des Netzes während des Trainings gewonnen werden kann. Für die folgenden Analysen wurde diese Art der Darstellung gewählt.

4.2.1 Eingabekodierung

Zunächst sollte geprüft werden, welche der oben erläuterten Kodierungsvarianten die besten Ergebnisse liefert. Zu diesem Zweck wurden für jede der drei Methoden neuronale Netze konzipiert und mit identischen Datensätzen trainiert. Für jede Kodierung wurden zwei Netze trainiert, die jeweils über ein Ausgabeneuron verfügten. Ein Netz diente dann zur Berechnung der α-Protonen, das andere zur Berechnung der amidischen Protonen.

Die Muster für den Trainingsdatensatz wurden aus 373 Datensätzen der *BMRB*-Datenbank generiert. Die Datensätze wurden nach folgender Methode ausgewählt: die für den Testdatensatz vorgesehenen Einträge wurden, wie bereits beschrieben, entfernt. Für die verbleibenden Einträge wurde ebenfalls geprüft, ob ein Protein in mehreren Datensätzen beschrieben wird. In diesem Fall wurde nur der Datensatz, der die meisten chemischen Verschiebungen enthielt, verwendet.

Da getrennte Netze für die beiden vorherzusagenden chemischen Verschiebungen benutzt werden sollten, wurden die unteren und oberen Limits für die lineare Abbildung der ppm-Skala auf die Ausgabeschicht (vgl. Abschnitt 3.4) entsprechend angepaßt. Für Netze zur Berechnung

von Amidprotonen wurde sh_{min} auf 7.00 ppm, sh_{max} auf 9.00 ppm gesetzt. Entsprechend wurden für die Hα-Protonen Werte von 3.00 ppm bzw. 5.00 ppm gewählt.

Unter diesen Einschränkungen konnten für die Trainingssätze 11284 Muster generiert werden. Diese Muster wurden für jede der drei Kodierungsmethoden erzeugt. Die Anzahl an Eingabeneuronen, die zur Kodierung einer Aminosäure nötig sind, ist bei den drei Varianten unterschiedlich. Aus diesem Grund wurde die Anzahl an versteckten Neuronen für die entsprechenden Netze so gewählt, daß das Verhältnis N_E/N_V annähernd gleich blieb. Die entsprechenden Parameter sind in Tabelle 19 aufgelistet. Die Netze werden im folgenden als *SEQ-st*, *SEQ-comp* und *SEQ-bit* bezeichnet.

Für das Training der Netze wurde eine innerhalb von 50000 Zyklen von 2.0 auf 0.01 abfallende Lernrate verwendet. Als Abbruchkriterium diente auch hier der RMS-Wert. Dieser wurde auf 0.0025 gesetzt. Während des Trainings konnte dieser sehr niedrige Wert allerdings von keinem der in diesem Abschnitt beschriebenen Netze erreicht werden.

Netz	*Neuronen pro Aminosäure*	*Neuronen für 9 Aminosäuren (N_E)*	*versteckte Neuronen (N_V)*	N_E / N_V
SEQ-st	21	189	50	3.78
SEQ-comp	6	54	15	3.6
SEQ-bit	24	216	60	3.6

Tabelle 19: Architektur der neuronalen Netze zur sequentiellen Zuordnung. Angegeben ist die Anzahl an Neuronen in Eingabe- und versteckter Schicht. Die Ausgabeschicht bestand aus einem einzelnen Neuron. In der letzten Spalte ist das Verhältnis von Eingabeneuronen zu versteckten Neuronen aufgezeigt. Die Zahl der versteckten Neuronen wurde so gewählt, daß dieses Verhältnis für die drei Netze möglichst gleich war.

Für die Netze, die die Standard- und Bitstringkodierung verwendeten, wurde die *invx*-Funktion (Gleichung 20) als Transferfunktion zwischen den Schichten benutzt. Für die Neuronen wurden die Werte ´0.05´ und

´0.95´ als Minimum bzw. Maximum definiert. Alle Werte unter bzw. über diesen Grenzen wurden als ´0´ oder ´1´ interpretiert. Bei Betrachtung des Verlaufs der *invx*-Funktion fällt auf, daß diese Limits sehr bald erreicht werden. Da bei der kompakten Kodierung die Eingabeneuronen Werte von ´1´ bis ´10´ annehmen können, würden bei Verwendung dieser Funktion viele Muster für das Netz zunächst sehr ähnlich aussehen. Auch wenn die Änderung der Schwellwerte die Transferfunktion entlang der Ordinate verschiebt, liegen sehr viele Neuronen während des Trainings über lange Zeit auf hohen Werten. Die nötige Korrektur der Schwellwerte und Gewichte, um diesem Effekt entgegenzuwirken, benötigt sehr viele Trainingszyklen. Dieser Effekt wurde in ersten Versuchen auch beobachtet: die neuronalen Netze mit kompakter Kodierung und *invx*-Funktion lernten sehr langsam. Nur eine deutliche Erhöhung der Zyklenzahl - und damit der benötigten Rechenzeit - hätte bei Einsatz dieser Funktion voraussichtlich bessere Resultate bewirkt. Die Verwendung der deutlich flacher verlaufenden *sqrlog*-Funktion (Gleichung 21) brachte hier, zumindest für die Trainingsdaten, eine erkennbare Verbesserung.

$$f(x) = \begin{cases} \dfrac{1}{2(1-x)} & x < 0 \\ 1 - \dfrac{1}{2(1+x)} & x \geq 0 \end{cases}$$

Gleichung 20: Definition der *invx*-Funktion.

$$f(x) = \begin{cases} \dfrac{\ln(2-x)}{2\sqrt{\ln(2)^2 - x}} & x < 0 \\ 1 - \dfrac{\ln(2+x)}{2\sqrt{\ln(2)^2 + x}} & x \geq 0 \end{cases}$$

Gleichung 21: Definition der *sqrlog*-Funktion

Der Verlauf der beiden Funktionen im Intervall $-10 \leq x \leq 10$ ist in Abbildung 20 dargestellt. Selbst bei hohen Eingabewerten ($x >> 10$) ist die sqrlog-Funktion noch nicht bei den angesprochenen Limits angelangt.

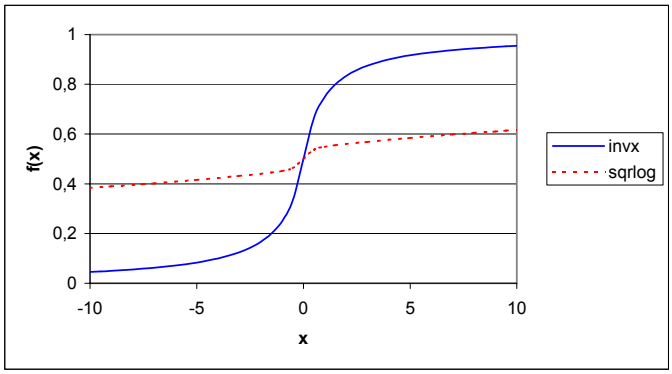


Abbildung 20: Verlauf der beiden Transferfunktionen *invx* und *sqrlog*.

Insgesamt wurden mit diesen Parametern und Mustern sechs Netze trainiert, für jede der drei Kodierungen und der zwei vorherzusagenden Verschiebungen eines. Nach jeweils 500 Trainingszyklen wurden sowohl Trainings- als auch Testdaten einmal berechnet und die Standardabweichung σ der aufgetretenen Fehler in ppm ermittelt. Abbildung 21 zeigt eine graphische Darstellung von σ für beide Datensätze während des Trainings.

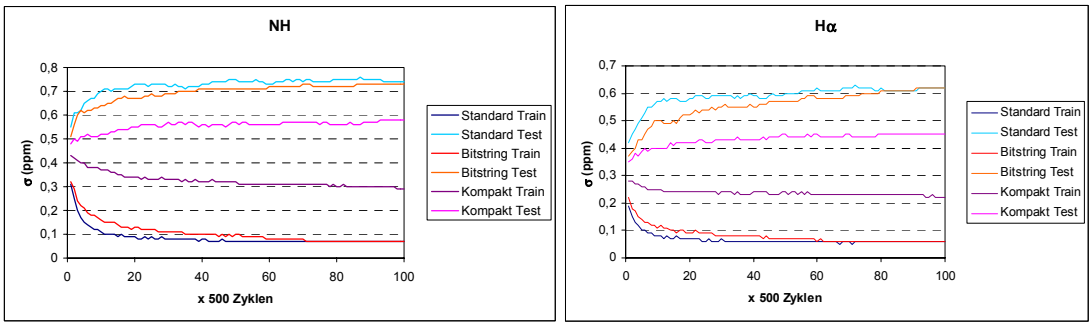


Abbildung 21: Verlauf von σ für Trainings- und Testdaten während 50000 Trainingszyklen.

Es zeigte sich, daß die Trainingsdaten während des Lernprozesses erwartungsgemäß immer besser erkannt wurden. Im gleichen Maß fiel die

Leistung für die Testdaten jedoch ab (obere Kurven in Abbildung 21). Ebenfalls erkennbar ist, daß die kompakte Kodierung zu Netzen führt, die sehr langsam und deutlich schlechter lernen als Netze mit den anderen beiden Varianten. Zwischen der Standard- und der Bitstringkodierung hingegen lassen sich am Ende des Trainings keine Unterschiede feststellen.

Es ist erkennbar, daß Hα-Protonen besser vorhergesagt werden können als amidische Protonen. Die zugehörigen σ-Werte liegen um ca. 0.1 ppm unter den Werten für NH-Protonen. Die chemischen Verschiebungen von Amidprotonen sind gegenüber Änderungen im pH-Wert und der Sekundärstruktur deutlich empfindlicher. Diese Effekte führen dazu, daß die NH-Verschiebungen ein und desselben Peptids schon bei winzigen Änderungen des pH-Wertes starken Schwankungen unterliegen. Diese Varianz kann von einem neuronalen Netz nicht erkannt werden.

Die erreichten σ-Werte für die Testdaten sind vor dem Hintergrund zu bewerten, daß der Bereich, in dem die gesuchten Signale liegen, in beiden Domänen nur 0.5 bis 0.8 ppm breit ist. Um akkurate Voraussagen in diesem Fenster treffen zu können, müßten Werte um 0.1 ppm erreicht werden.

4.2.2 Einfluß der versteckten Neuronen

Die bisher betrachteten Netze zeigten ein Verhalten, daß als *Overfitting* interpretiert werden kann: die Trainingsdaten werden sehr gut erkannt, die Testdaten hingegen nur unzureichend. Diese Tendenz nimmt im Verlauf der Trainingsphase sogar noch zu. Das Netz ist also zu sehr auf die Trainingsdaten spezialisiert, aus unbekannten Daten kann es nicht ausreichend extrapolieren. Ein Grund für dieses Phänomen kann die Anzahl der versteckten Neuronen sein. Ist diese zu hoch, so werden die Trainingsdaten nur „auswendig“ gelernt, jedoch keine Gesetzmäßigkeiten in den Mustern erkannt.

Um zu prüfen, ob tatsächlich die Anzahl der versteckten Neuronen hierfür verantwortlich war, wurde eine entsprechende Testreihe berechnet. Dazu wurden die Muster in der Bitstringkodierung verwendet und Netze mit 60, 30, 15 und fünf versteckten Neuronen trainiert. Die Netze hatten jeweils ein

Ausgabeneuron, auf das die chemischen Verschiebungen abgebildet wurden. Als Grenzen wurden erneut 3.00 ppm und 5.00 ppm für Hα-Protonen bzw. 7.00 ppm und 9.00 ppm für amidische Protonen festgelegt. Die weiteren Trainingsparameter (Anzahl der Zyklen, abfallende Lernrate, Transferfunktion) wurden nicht verändert.

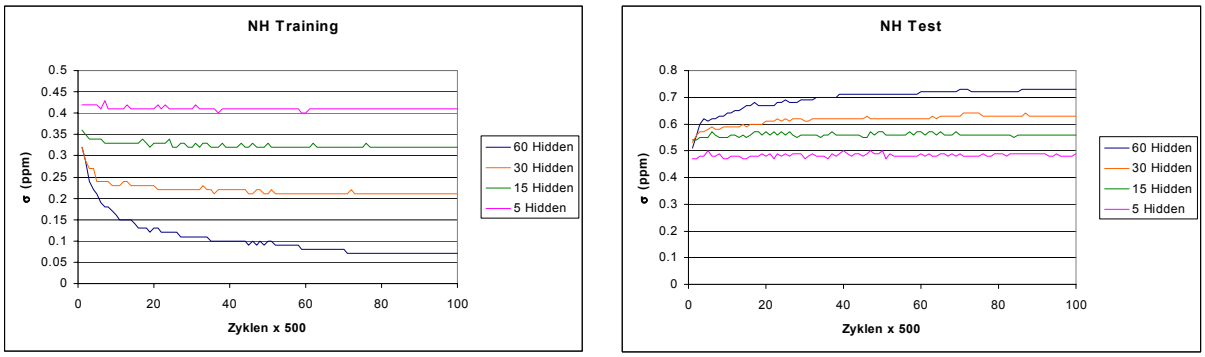


Abbildung 22: **Vergleich von σ für neuronale Netze zur Berechnung der chemischen Verschiebung von amidischen Protonen. Die Netze wurden mit der Bitstringkodierung trainiert.**

In Abbildung 22 ist der Einfluß der versteckten Neuronen für die Vorhersage der amidischen Protonen aufgezeigt. Ab 15 versteckten Neuronen zeigt sich, daß das Netz nicht mehr lernt. Mit 30 versteckten Neuronen werden einerseits die Trainingsdaten schlechter erkannt, andererseits ist die Leistung in Bezug auf die Testmuster um ca. 0.1 ppm besser als bei dem Netz mit 60 versteckten Neuronen. Weiterhin bleibt σ für die Testdaten durch das ganze Training hindurch konstant.

Die gleichen Effekte sind für Hα-Protonen in Abbildung 23 aufgezeigt. Auch hier führen 30 versteckte Neuronen zu einem Kompromiß zwischen guter Trainingsleistung und verbesserter Erkennung der Testdaten. Die erreichten Werte für σ weisen hier sogar eine Verbesserung um ca. 0.15 ppm, bezogen auf das Netz mit 60 versteckten Neuronen, auf.

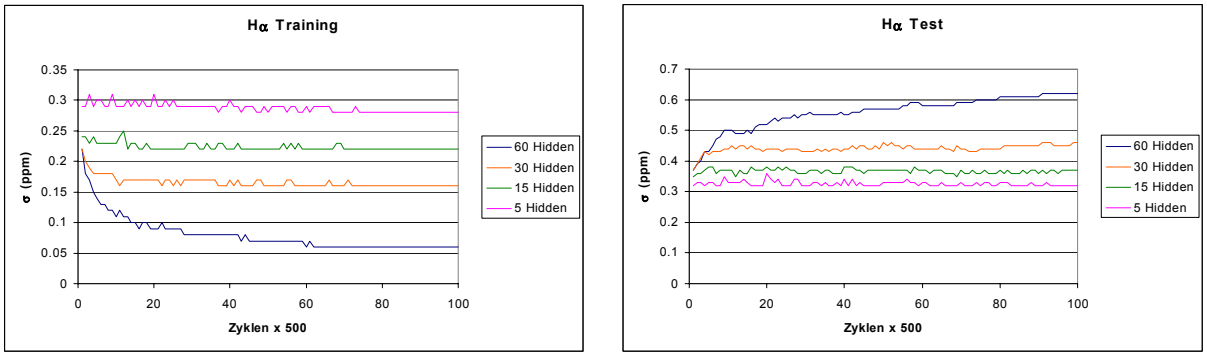


Abbildung 23: **Vergleich von σ für neuronale Netze zur Berechnung der chemischen Verschiebung von Hα Protonen. Die Netze wurden mit der Bitstringkodierung trainiert.**

Bei 216 Eingabeneuronen und 30 versteckten Neuronen ergibt sich ein Verhältnis von N_E/N_V von 7.2. Für Netze mit der Standardkodierung beträgt N_E 189. Um annähernd den gleichen Quotienten zu erhalten, wurden diese Netze mit 25 versteckten Neuronen erneut trainiert. Die übrigen Parameter wurden ebenfalls nicht verändert. Die Verringerung von N_V bringt auch bei der Standardkodierung eine verbesserte Berechnung der Testdaten. Gleichzeitig ist eine Verschlechterung der Trainingsergebnisse zu erkennen (vgl. Abbildung 24 und Abbildung 25).

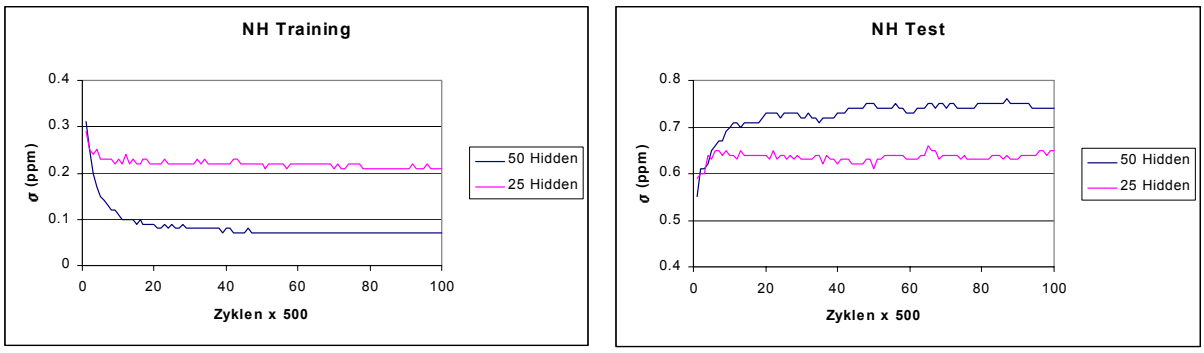


Abbildung 24: **Vergleich von σ für neuronale Netze zur Berechnung der chemischen Verschiebung von amidischen Protonen. Die Netze wurden mit der Standardkodierung trainiert.**

Die Standardabweichungen verbessern sich auch in dieser Kodierung um ca. 0.1 ppm für amidische Protonen bzw. um ca. 0.15 ppm für Hα Protonen.

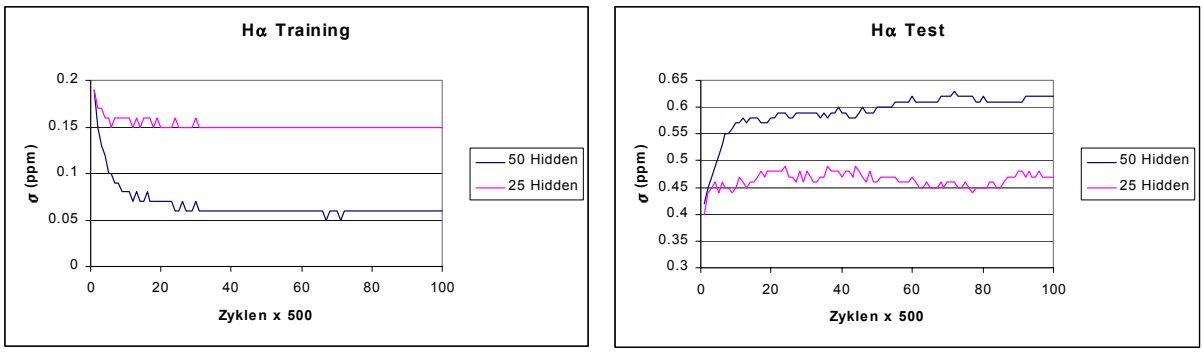


Abbildung 25: **Vergleich von σ für neuronale Netze zur Berechnung der chemischen Verschiebung von Hα Protonen. Die Netze wurden mit der Standardkodierung trainiert.**

Eine weitere Verringerung von N_V wurde nicht durchgeführt, da erwartet wurde, daß auch diese Netze nicht mehr lernen würden.

4.2.3 Netze für einzelne Aminosäuren

In einem weiteren Ansatz wurde geprüft, ob eine Spezialisierung der Netze die Ergebnisse verbessert. Dazu wurde für jede der 20 natürlichen Aminosäuren ein eigenes Netz trainiert. Je nach gesuchter Aminosäure im Testsatz wurde dann das entsprechende Netz verwendet.

Zur Erstellung der Trainingsmuster wurden dieselben Datensätze herangezogen, die auch für die bisher besprochenen neuronalen Netze verwendet wurden. Die Muster wurden für alle drei Kodierungsvarianten erzeugt und dann in 20 Sätze für die jeweiligen Aminosäuren aufgeteilt. In Tabelle 21 ist aufgeführt, wie viele Muster pro Aminosäure zur Verfügung standen. Die Anzahl schwankt von 67 Mustern für Tryptophan bis zu 816 Mustern für Leucin (vgl. Tabelle 21).

Auch diese Netze wurden mit nur einem Ausgabeneuron ausgestattet, es wurden also pro Aminosäure zwei Netze benötigt. Das erste war für die

Bestimmung der Amidprotonen vorgesehen, das zweite für Hα-Protonen. Somit wurden für jede Kodierung insgesamt 60 neuronale Netze trainiert. Für die lineare Abbildung der chemischen Verschiebungen wurden sh_{min} und sh_{max} 3.0 ppm und 5.0 ppm für Hα, bzw. 5.0 ppm und 7.0 ppm für Amidprotonen gesetzt. Die Netze mit kompakter Kodierung wurden mit der *sqrlog*-Funktion trainiert, die beiden anderen Kodierungen mit der *invx*-Funktion. Die Anzahl der Neuronen in Eingabe- und versteckter Schicht ist in Tabelle 20 aufgeführt. Hier wurden die Ergebnisse aus den Untersuchungen zum Einfluß von N_V berücksichtigt und eine geringere Anzahl an versteckten Neuronen gewählt. Die Netze sind als *SEQ-st-sp*, *SEQ-comp-sp* und *SEQ-bit-sp* bezeichnet. Der Zusatz „-sp“ steht hier für „spezialisiertes Netz“.

Netz	*Eingabeneuronen (N_E)*	*Versteckte Neuronen (N_V)*
SEQ-st-sp	189	25
SEQ-comp-sp	54	10
SEQ-bit-sp	216	30

Tabelle 20: Architektur der auf einzelne Aminosäuren spezialisierten Netze.

Das Training wurde mit der bereits beschriebenen abfallenden Lernrate über maximal 50000 Zyklen durchgeführt. Manche Netze erreichten den als Abbruchkriterium definierten RMS-Wert von 0.0025 vor Ablauf der 50000 Epochen. Denkbar wäre, daß dies für Netze für die nur sehr wenige oder aber sehr viele Trainingsmuster zur Verfügung haben, der Fall ist. Bei wenigen Mustern wäre wieder ein "Auswendiglernen" zu beobachten. Bei sehr vielen Mustern hingegen würden die Netze die Gesetzmäßigkeiten korrekt erkennen und entsprechend optimiert werden. Allerdings läßt sich dieser Zusammenhang nicht feststellen. Die Aminosäuren, bei denen das Abbruchkriterium in mindestens einer Kodierung und für mindestens eine chemische Verschiebung erreicht wurde, sind in Tabelle 21 grau hervorgehoben. Man erkennt, daß auch bei Seitenketten für die

durchschnittlich viele Muster erzeugt werden konnten, das Training vorzeitig beendet wurde.

W	*H*	*C*	*M*	*Y*	*F*	*P*	*N*	*Q*	*I*	*T*	*V*	*D*	*S*	*G*	*R*	*E*	*K*	*A*	*L*
62	148	152	197	199	228	282	308	349	384	397	409	420	436	499	539	587	647	722	816

Tabelle 21: Anzahl der Muster, die für die einzelnen Aminosäuren zur Verfügung standen. Wurde für mindestens ein Netz (unabhängig von Kodierung und chemischer Verschiebung) einer Aminosäure das Training wegen Erreichen des Abbruchkriteriums vorzeitig beendet, so ist die entsprechende Spalte grau hervorgehoben.

Um die Ergebnisse dieser insgesamt knapp 120 Netze noch anschaulich darstellen zu können, wurden die σ-Werte nur noch am Ende des Lernvorgangs sowohl für Trainings- als auch für Testdaten ermittelt. Man erhält somit eine Momentaufnahme der Leistungsfähigkeit der neuronalen Netze. Diese Werte können nun für die einzelnen Kodierungen und chemischen Verschiebungen aufgezeichnet werden. In Abbildung 26 sind die Resultate als Balkendiagramm aufgetragen.

Für die Standardkodierung können diese Ergebnisse mit den Kurven für 25 versteckte Neuronen in Abbildung 24 und Abbildung 25 verglichen werden. Dabei fällt auf, daß die Werte für das Training für beide Verschiebungen nach der Verteilung auf 20 verschiedene Netze deutlich besser sind. Bei den nicht spezialisierten Netzen erreicht σ am Ende des Trainings Werte von ca. 0.2 ppm (NH) und ca. 0.15 ppm (Hα). Für die 20 einzelnen Netze hingegen liegen diese Werte zwischen 0.04 ppm und 0.11 ppm für Amidprotonen bzw. zwischen 0.03 ppm und 0.1 ppm für Hα-Protonen. Bei den Testdaten ist das Ergebnis nicht so eindeutig. Für amidische Protonen ist meistens eine kleine Verbesserung gegenüber einem einzelnen Netz (σ ≈ 0.65 ppm) sichtbar. Besonders bei den Aminosäuren *M*, *Q* und *W* wird ein deutlich geringerer Wert von ca. 0.40 ppm erreicht. Auch im Fall der Hα-Protonen sind die Werte meist nur wenig besser als für das einzelne Netz (σ ≈ 0.46 ppm). Besonders

hervorzuheben sind die Aminosäuren *M* und *P*. Hier liegt σ zwischen 0.23 ppm und 0.16 ppm, was eine erhebliche Verbesserung darstellt.

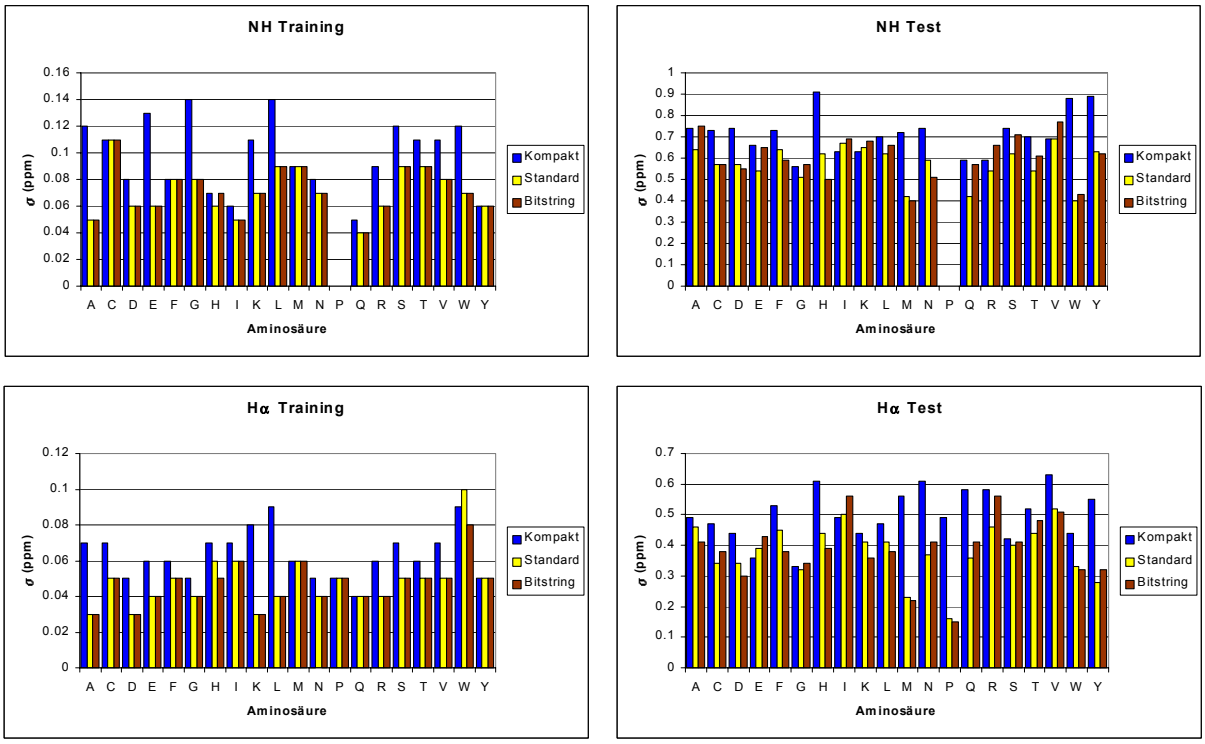


Abbildung 26: **Standardabweichungen der Fehler für neuronale Netze, die chemische Verschiebungen von Amidprotonen bzw. Hα-Protonen berechnen sollen. Die Netze waren auf jeweils einen Aminosäuretyp spezialisiert. Da Prolin kein Amidproton besitzt, ist die entsprechende Spalte leer.**

Die Netze, bei denen die Bitstringkodierung verwendet wurde, entsprechen in der Architektur dem Netz mit 30 versteckten Neuronen in Abbildung 22 und Abbildung 23. Bei den Trainingsdaten liegt die Verbesserung in der gleichen Größenordnung wie auch für die Standardkodierung. Auch für die Testdaten zeichnet sich ein ähnlicher Trend ab. Im Fall der Amidprotonen liegt σ meist um den Wert des einzelnen Netzes ($\sigma \approx 0.61$ ppm) mit minimalen Verbesserungen. Die deutlichsten Steigerungen stellen sich für die Aminosäuren *M* und *W* ein. Die Netze für Hα-Protonen zeigen das gleiche Verhalten: viele Aminosäuren erreichen Werte, die mit dem des einzelnen Netzes ($\sigma \approx 0.45$ ppm) vergleichbar sind. Die deutlichsten Verbesserungen werden bei

den Aminosäuren *D*, *M* und *P* erzielt. Die Werte für σ sind hier um bis zu 0.3 ppm geringer.

Bei der kompakten Kodierung zeigt sich im Training eine deutliche Verbesserung. Das einzelne Netz erreicht σ-Werte von 0.30 ppm (NH) und ca. 0.20 ppm (vgl. Abbildung 21). Dem gegenüber stehen Werte von 0.05 ppm bis 0.14 ppm bei den Amidprotonen und von 0.04 ppm bis 0.09 ppm bei den Hα-Protonen, die nach Aufteilung auf 20 Netze erreicht werden. Allerdings werden die Testdaten tendenziell weniger gut erkannt. Für das einzelne Netz sind die entsprechenden Standardabweichungen 0.60 ppm (NH) und 0.45 ppm (Hα). In 15 von 20 Fällen liegt σ für NH-Protonen teilweise deutlich über dem Wert von 0.60 ppm; bei den Hα-Protonen ist für 14 Aminosäuren ein schlechterer Wert als 0.45 ppm zu erkennen. Für die 20 separaten Netze wurden weniger versteckte Neuronen verwendet als für das einzelne Netz. Die im Training deutlich besseren Ergebnisse sprechen jedoch gegen einen starken Einfluß der geänderten Netzarchitektur.

Im allgemeinen bewirkt die Aufteilung auf einzelne Netze für die 20 Aminosäuren somit lediglich eine Optimierung der Trainingsergebnisse, die Testdaten werden nur minimal besser zugeordnet. Nach wie vor ist die Genauigkeit damit nicht hoch genug um für unbekannte Sequenzen akkurate Vorhersagen zur Lage der Spuren erstellen zu können.

4.2.4 Verteilung auf vier verschiedene Netze

Bei den bisher beschriebenen Versuchen wurden die Trainingsdaten den jeweiligen Netzen als kompletter Satz präsentiert. Das in Abschnitt 3.6 beschriebene Verfahren verschiedene Vorhersagen zu berechnen und daraus einen Mittelwert zu bilden, wurde mit diesen Netzen nicht durchgeführt.

Um diese Methode zu testen, wurde ein weiterer Satz neuronaler Netze trainiert, die als *SEQ-st-sp4*, *SEQ-comp-sp4* und *SEQ-bit-sp4* bezeichnet werden sollen. Diese Netze hatten, im Gegensatz zu den bisher besprochenen, zwei Ausgabeneuronen. Das erste diente zur Darstellung

der NH-Verschiebung, das zweite zur Ermittlung der Hα-Verschiebung. Da hier ein größerer Bereich der chemischen Verschiebung abgebildet werden mußte, wurden sh_{min} und sh_{max} für die Abbildung der chemischen Verschiebung entsprechend auf 0.00 ppm bzw. 12.22 ppm gesetzt. Für alle drei Kodierungen wurden nach Aminosäuren sortierte Trainingsdaten erstellt. Hierfür wurden nun, mit Ausnahme der für die Testdaten vorgesehenen Datensätze, alle verfügbaren Einträge in der *BMRB*-Datenbank herangezogen. Es wurde also nicht mehr berücksichtigt, ob für ein Protein möglicherweise mehrere Einträge vorhanden waren. Insgesamt lagen den Trainingsdaten damit 1357 *BMRB*-Einträge zu Grunde. Die Anzahl an erzeugten Mustern reicht von 217 für Tryptophan bis zu 1678 für Lysin (vgl. Tabelle 22).

W	*H*	*C*	*M*	*Y*	*F*	*P*	*N*	*Q*	*I*	*T*	*V*	*D*	*S*	*G*	*R*	*E*	*K*	*A*	*L*
217	378	603	377	655	702	719	912	777	961	1157	1236	1180	1131	1515	957	1413	1574	1500	1678

Tabelle 22: Anzahl der Muster, die zum Training der verteilten Netze pro Aminosäure zur Verfügung standen. Jeder dieser Mustersätze wurde zufällig in vier gleich große Teile aufgeteilt. Diese wurden zum Training von vier Netzen pro Aminosäure verwendet.

Die 20 Datensätze pro Kodierung wurden nun noch weiter aufgeteilt. Die Muster in einem Satz wurden gleichmäßig auf vier Sätze verteilt, mit denen dann vier Netze trainiert werden konnten. Insgesamt wurden für diesen Ansatz also 240 Netze trainiert: je vier Netze für jede der 20 Aminosäuren in drei Kodierungen. Die Architektur der verschiedenen Netze ist in Tabelle 23 dargestellt. Jedes der vier Netze pro Aminosäure wurde mit der der Kodierung entsprechenden Anzahl an Neuronen trainiert.

Auch diese Netze wurden über 50000 Zyklen mit einer abfallenden Lernrate trainiert, wobei ebenfalls in mehreren Fällen das Training vorzeitig durch Erreichen des Abbruchkriteriums beendet wurde.

Kodierung	*Eingabeneuronen*	*versteckte Neuronen*	*Ausgabeneuronen*
Standard	189	50	2
Kompakt	54	20	2
Bitstring	216	40	2

Tabelle 23: Architektur der neuronalen Netze, bei denen die Trainingsdaten für jede Aminosäure in vier Sätze aufgeteilt wurden.

Bedingt durch die große Menge anfallender Daten wurden die Resultate an dieser Stelle nicht systematisch ausgewertet. Statt dessen wurden die trainierten Netze in weitergehende Zuordnungsmethoden implementiert und ihre Leistungsfähigkeit erst dann beurteilt. Die Ergebnisse für diese Netze werden deshalb in einem späteren Abschnitt näher beschrieben.

Zusätzlich wurde ein weiterer Satz neuronaler Netze, bei denen die Aufteilung für einzelne Aminosäuren nicht erfolgte, trainiert. Diese Netze werden im weiteren als *SEQ-st4*, *SEQ-comp4* und *SEQ-bit4* bezeichnet. Pro Kodierung wurden somit vier Netze, welche die chemischen Verschiebungen jeder Aminosäure berechnen sollten, erstellt. Die Anzahl an Neuronen in den einzelnen Schichten und sonstigen Trainingsparameter stimmten mit denen der spezialisierten Netze überein (Tabelle 23). Die Leistung der fertig trainierten Netze wurden ebenfalls erst nach weiterführenden Tests beurteilt.

4.2.5 Inkrementsystem

Die für die neuronalen Netze verwendeten Testdaten wurden auch für die Bewertung des Inkrementsystems herangezogen. Als Qualitätskriterium diente ebenfalls die Standardabweichung der Fehler σ. Nach dem beschriebenen Verfahren wurden zunächst Inkrementtabellen erstellt, die ein bis zu 21 Reste breites Sequenzfenster ermöglichten. Es wurde also der Einfluß von Seitenketten bis zu zehn Positionen vor bzw. nach der zu bestimmenden Aminosäure berücksichtigt. Von diesen maximal 20 Inkrementen kann nun eine beliebige Anzahl benutzt werden, um eine chemische Verschiebung zu bestimmen.

Um die chemischen Verschiebungen einer Aminosäure zu berechnen, müssen zu den gefundenen Mittelwerten dieser Verschiebungen nun die korrekten Inkremente addiert werden (vgl. Abschnitt 3.5). Zunächst wurde geprüft, welchen Einfluß die Anzahl der verwendeten Inkremente, die Breite des Lesefensters also, hat. Dazu wurden sämtliche Aminosäuren im Testsatz unter Berücksichtigung von jeweils 1 - 10 Inkrementen in C- und N-terminaler Richtung berechnet. Die Fenster in denen die benachbarten Aminosäuren berücksichtigt wurden, waren also 3 - 21 Aminosäuren breit. In Abbildung 27 sind verschiedene Sequenzfenster exemplarisch aufgezeichnet.

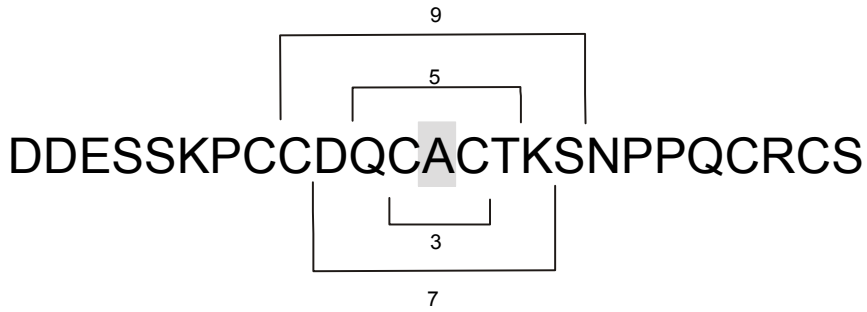


Abbildung 27: **Verschiedene Sequenzfenster für die Berechnung mit dem Inkrementsystem. Bei einem drei Reste breitem Fenster werden nur die Inkremente für unmittelbar benachbarte Aminosäuren benutzt. Mit zunehmender Breite fließen weitere Inkremente in die Berechnung ein.**

Für die erhaltenen Werte wurde die Standardabweichung der Fehler ermittelt und gegen die verwendete Inkrementanzahl aufgetragen. Für den Vergleich mit den neuronalen Netzen ist die Spalte mit der Sequenzlänge ´9´ in Abbildung 28 am wichtigsten, da auch die Netze mit dieser Sequenzlänge arbeiten.

Bei dieser Sequenzlänge hat σ für NH-Protonen einen Wert von 0.53 ppm. Für Hα-Protonen wird ein Wert von 0.37 ppm erreicht. Diese Werte sind um ca. 0.1 ppm besser als die Ergebnisse der neuronalen Netze, die sämtliche Aminosäuren berechnen. Ebenfalls ersichtlich ist, daß die Ergebnisse mit steigender Sequenzlänge langsam schlechter werden.

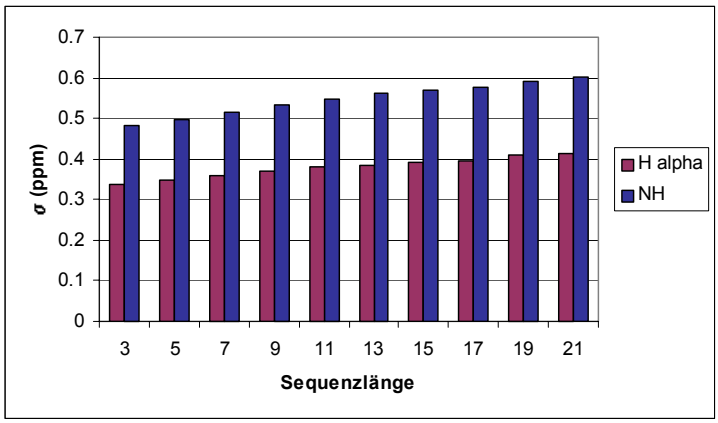


Abbildung 28: **Ergebnisse für das Inkrementsystem in Abhängigkeit von der Breite des Lesefensters.**

Die Testdaten wurden für eine weitere Analyse in 20 einzelne Datensätze aufgeteilt. Jeder dieser Datensätze hatte eine einzige Aminosäure in der mittleren Position eines neun Reste langen Abschnitts. Somit konnte geprüft werden, ob einige Aminosäuren besser berechnet werden können als andere. Dieser Ansatz ist mit dem der spezialisierten neuronalen Netze vergleichbar. Die Ergebnisse sind in Abbildung 29 dargestellt. Zur besseren Übersicht wurden die Werte für die kompakt kodierten Netze weggelassen, da diese Netze im allgemeinen die schlechtesten Ergebnisse lieferten.

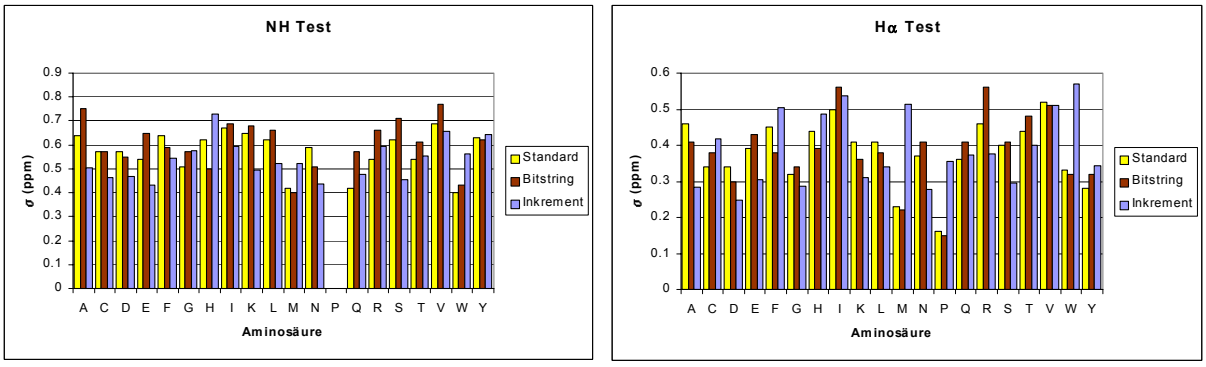


Abbildung 29: **Vergleich von neuronalen Netzen und dem Inkrementsystem für einzelne Aminosäuren.**

Es zeigt sich, daß das Inkrementsystem in den meisten Fällen bessere Ergebnisse liefert als mindestens eines der neuronalen Netze. Besonders für amidische Protonen ist diese Tendenz deutlich erkennbar. Bei den Hα-Protonen zeigen sich größere Diskrepanzen vor allem für Methionin, Tryptophan und Prolin. Hier ist das Inkrementsystem deutlich schlechter als die neuronalen Netze. Für die übrigen Aminosäuren überwiegt eine geringfügige Verbesserung der Vorhersagen.

4.3 Spurzuordnung

Alle bisher beschriebenen Netze für die sequentielle Zuordnung wurden im folgenden herangezogen, um Methoden zur Auswertung von TOCSY-Spektren zu testen. Dieser Weg wurde gewählt, da die Genauigkeit aller Netze nicht hoch genug ist, um die Position einzelner Spuren korrekt vorherzusagen. Allerdings geben die Vorhersagen einen ersten Anhaltspunkt, wo im Spektrum die gesuchte Spur zu finden sein sollte. Ausgehend von diesem Wert kann nun nach der nächsten tatsächlich auftretenden Spur gesucht werden. Diese Zuordnung kann dann noch durch NOE-Daten verifiziert werden.

Für diese Untersuchungen wurde ein neuer Testdatensatz erstellt. Dieser bestand teilweise aus Spektren aus der *BMRB*-Datenbank. Da diese keinerlei Informationen über NOE-Daten enthält, wurde die entsprechende Originalliteratur herangezogen, um diese Daten zu erhalten. Weiterhin wurden Daten aus einigen zusätzlichen Veröffentlichungen verwendet. Spektren von einigen im Arbeitskreis synthetisierten Peptiden wurden ebenfalls in den Testdatensatz mit aufgenommen. Tabelle 24 listet die benutzten Daten auf.

Die Spuren und ihre Zuordnungen waren für die aufgeführten Beispiele bekannt. Es sollte nun geprüft werden, ob die entwickelten Methoden zu den gleichen Ergebnissen führten.

Name	*Sequenzlänge*	*Quelle*
1656	18	BMRB[82]
ns4bsubst	19	sonstige Literatur[101]
ns5ainhib	19	sonstige Literatur[101]
Rhod	25	sonstige Literatur[102]
j-v3-9	29	synthetisiertes Peptid[103]
j-v3-10	29	synthetisiertes Peptid[103]
Kin	30	synthetisiertes Peptid[104]
1336	30	BMRB[89]
1500	30	BMRB[97]
1666	34	BMRB[90]
Ata	38	sonstige Literatur[105]
2061	49	BMRB[85]
1479	59	BMRB[94]

Tabelle 24: Zusammensetzung des Testdatensatzes für die Analyse der Zuordnungsmethoden. In den Fällen, in denen der Name nur aus einer Zahl besteht, gibt diese die Nummer des Eintrags in der *BMRB*-Datenbank an.

Dabei wurde davon ausgegangen, daß die Zuordnung der Spuren zu einem Aminosäuretyp bereits erfolgt ist. Allerdings wurden die mehrdeutigen Spuren, die bei der Analyse durch neuronale Netze in Gruppen zusammengefaßt wurden, entsprechend dieser Vereinfachungen behandelt. Trat in einem Spektrum z.B. eine Glutaminsäurespur auf, so wurde diese im Folgenden als eine Spur angesehen, die entweder Glutamin oder Glutaminsäure sein konnte. Es standen nun also Listen mit NH/Hα-Kreuzsignalen, denen eine der in Abschnitt 3.2 beschriebenen Aminosäuregruppen zugeordnet war, als Ausgangsdaten zur Verfügung. Zusätzlich war die Sequenz des untersuchten Peptids bekannt.

Für die eigentliche Zuordnung wurden zunächst die folgenden Schritte durchlaufen:

1. Aus der Sequenz wurden entsprechende Muster für die verschiedenen neuronalen Netze erzeugt.
2. Für jede Aminosäure in der Sequenz wurden die relevanten chemischen Verschiebungen mit verschiedenen neuronalen Netzen und dem Inkrementsystem berechnet.
3. Für jede getroffene Vorhersage wurde das nächstliegende Kreuzsignal im Spektrum gesucht, das der gesuchten Aminosäureklasse entsprach. Das gefundene Kreuzsignal wurde einer diskreten Aminosäure in der Sequenz zugeordnet.
4. Die Zuordnungen, die aus verschiedenen Netzen stammten, wurden miteinander verglichen. Wenn ein vorher definiertes Maß an Übereinstimmung festgestellt wurde, so wurde diese Zuordnung endgültig akzeptiert.

Die in Schritt 2 verwendeten Netze können beliebig ausgewählt werden. Auch der Zuordnungsalgorithmus, der vier verschiedene Netze benutzt, kann an dieser Stelle verwendet werden. In diesem Fall wird wie in Abbildung 17 (Seite 49) dargestellt verfahren, um den Kreuzsignalen bestimmte Seitenketten zuzuordnen.

Man erhält so für jedes benutzte Netz einen Satz Zuordnungen. Benutzt man beispielsweise drei neuronale Netze und das Inkrementsystem, so werden jedem Kreuzsignal vier Seitenketten zugeordnet. Diese Zuordnungen können nun miteinander verglichen werden, um voneinander abweichende Ergebnisse auszuschließen. Beispielsweise kann festgelegt werden, das nur Ergebnisse, bei denen mindestens 60 % der Zuordnungen übereinstimmen, akzeptiert werden (vgl. Tabelle 25). Dieses Limit kann variabel gestaltet werden, so daß evtl. auch alle vier Zuordnungen identisch sein müssen.

Aminosäure	*Kompakt*	*Bitstring*	*Standard*	*Inkrement*	*Zuordnung*	*Sollwert*
K1	12	6	6	12	-	
G2	8	5	8	0	-	0
C3	4	1	1	2	-	1
W4	1	4	4	4	**4**	2
K5	3	3	3	3	**3**	3
C6	2	2	2	14	**2**	4
G7	5	8	5	8	-	5
K8					-	6
E9	7	7	7	10	**7**	7
G10	0	0	0	5	**0**	8
H11	9	9	9	9	**9**	9
Q12	10	10	10	7	**10**	10
M13	11	11	11	11	**11**	11
K14	6	12	12	6	-	12
D15	13	13	13	13	**13**	13
C16	14	14	14	1	**14**	14
T17	15	15	15	15	**15**	15
E18	16	16	16	16	**16**	16

Tabelle 25: **Automatische Zuordnung der Kreuzsignale des Datensatzes bmr1656 mit neuronalen Netzen, die auf einzelne Aminosäuren spezialisiert sind. Jede Nummer steht für ein Kreuzsignal, die korrekten Sollwerte sind in der entsprechenden Spalte angegeben. Die Numerierung erfolgte automatisch bei der Eingabe der Signale und ist willkürlich. Da die NH-Verschiebung des N-terminalen Lysins nicht bekannt war, existiert für das entsprechende Signal kein Sollwert. Wenn mindestens 3 der 4 Zuordnungen übereinstimmten, so wurde die Zuordnung akzeptiert. Grau unterlegt sind Zuordnungen, die nicht korrekt sind. Blau unterlegte Zeilen geben korrekte Zuordnungen wieder.**

In Tabelle 25 fällt auf, das für Lysin 8 keine einzige Zuordnung gemacht wurde. Der Grund hierfür ist in der Anzahl der auftretenden Signale zu sehen. Da der N-Terminus bei diesem Peptid als freie Aminofunktion

vorlag, ist das entsprechende Kreuzsignal bedingt durch Austausch mit dem Lösungsmittel nicht sichtbar. Somit fehlt ein Lysinsignal. Die übrigen drei liegen offensichtlich näher an den Vorhersagen und werden zuerst gefunden.

Da die korrekten Zuordnungen bekannt sind, kann ermittelt werden, wie viele getroffene Zuordnungen richtig sind. Daran kann die Qualität der vorgestellten Methoden gemessen werden. So konnten im obigen Beispiel zwölf von 17 Signalen überhaupt zugeordnet werden. Von diesen Zuordnungen stellten sich neun als richtig heraus. Bezogen auf die Anzahl der Signale entspricht das einer Erkennungsrate von 53 %. Bezieht man sich auf die Anzahl der überhaupt zugeordneten Signale, so ergibt sich eine relative Erkennungsrate von 75 %. Die relative Erkennungsrate ist somit ein Maß für die Zuverlässigkeit der getroffenen Zuordnungen. Im Zweifelsfall ist sie höher einzustufen als die absolute Erkennungsrate, da sich bei der manuellen Interpretation von Spektren schon wenige, sichere Anhaltspunkte als sehr hilfreich erweisen. Beide Erkennungsraten sollen im folgenden als Bewertungskriterium dienen.

4.3.1 Vergleich der Netzarchitekturen für spezialisierte Netze

An dieser Stelle sollen zunächst die in Abschnitt 4.2.4 (Seite 83) besprochenen Netze weiter untersucht werden. Mit diesen jeweils vier Netzen pro Kodierung und Aminosäure wurden die erwähnten Schritte durchlaufen und die Erkennungsraten bestimmt. Außerdem wurde das Inkrementsystem in allen Untersuchungen als weitere Methode benutzt.

Bei der Verwendung von vier Netzen wird nicht mehr garantiert jedem Kreuzsignal eine Aminosäure zugeordnet. Der Grund hierfür sind Vorhersagen, bei denen die vier Netze zu sehr voneinander abweichen. Falls die Standardabweichung der vier berechneten Werte zu hoch (größer als 1 ppm) ist, wird diese Vorhersage nicht verwendet. Da dieser Fall bei den verschiedenen Kodierungsmöglichkeiten auftritt, ist es möglich, daß für ein und die selbe Aminosäure ein Satz Netze eine Zuordnung findet, der andere jedoch nicht. Für den abschließenden Vergleich muß nun zusätzlich eingeführt werden, wie viele Zuordnungen für eine Aminosäure

überhaupt mindestens getroffen wurden. Von diesen muß dann weiterhin ein bestimmter Prozentsatz übereinstimmen.

Aminosäure	*Standard 4*	*Bitstring 4*	*Kompakt 4*	*Inkrement*	*Zuordnung*	*Sollwert*
K1	3		3	12	-	
G2	8			0	-	0
C3	9		1	2	-	1
W4	1	2	2	4	-	2
K5	6	3		3	-	3
C6	4	4	4	14	**4**	4
G7		5		8	-	5
K8		6	6		-	6
E9	16	10	7	10	-	7
G10	5	8	5	5	**5**	8
H11	14	9	14	9	-	9
Q12	10	7	10	7	-	10
M13	11	11	11	11	**11**	11
K14	12	12	12	6	**12**	12
D15	13	13	13	13	**13**	13
C16	2	1	9	1	-	14
T17	15	15	15	15	**15**	15
E18	7	16	16	16	**16**	16

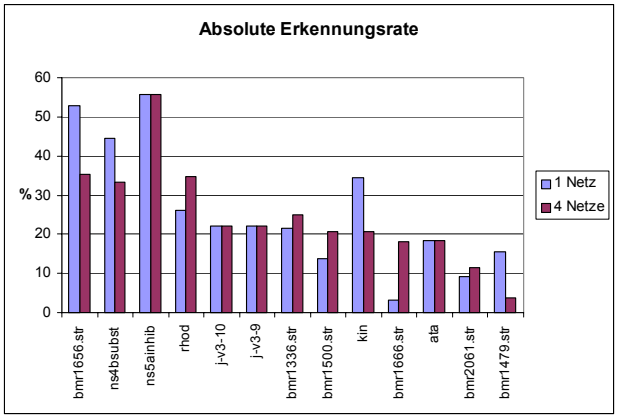
Tabelle 26: **Automatische Zuordnung der Kreuzsignale des Datensatzes bmr1656 mit neuronalen Netzen. Jede Nummer steht für ein Kreuzsignal, die korrekten Sollwerte sind in der entsprechenden Spalte angegeben. Die Numerierung erfolgte automatisch bei der Eingabe der Signale und ist willkürlich. Da die NH-Verschiebung des N-terminalen Lysins nicht bekannt war, existiert für das entsprechende Signal kein Sollwert. Es wurden Sätze von jeweils vier neuronalen Netzen pro Kodierung und Aminosäure sowie das Inkrementsystem benutzt. Wenn mindestens drei Methoden eine Aminosäure zuordnen konnten und mindestens 60% dieser Zuordnungen übereinstimmten, wurde das Ergebnis akzeptiert. Falsche Zuordnungen sind grau unterlegt. Blau unterlegte Zeilen geben korrekte Zuordnungen wieder.**

Für das Beispiel in Tabelle 26 wurde festgelegt, daß mindestens drei Methoden die Aminosäure zuordnen konnten und mindestens zwei der drei Zuordnungen identisch sein mußten. Außerdem müssen die Zuordnungen nach diesem Vergleich eindeutig sein. Es darf also nicht für ein Kreuzsignal mehrere Möglichkeiten geben. Im dargestellten Beispiel gäbe es für die Aminosäuren K1 und K5 eine Zuordnung, bei der zwei von drei Ergebnissen übereinstimmen (66 %). Allerdings würde in jedem der beiden Fälle Kreuzsignal Nr. 3 der entsprechenden Aminosäure zugeordnet. Da dies nicht zulässig ist, werden die Zuordnungen zurückgewiesen.

Mit diesen Parametern ergeben sich Erkennungsraten von ca. 35 % (absolut) und ca. 86 % (relativ). Unter den gleichen Bedingungen wurden nun sämtliche Beispiele im Testdatensatz berechnet und die entsprechenden Erkennungsraten ermittelt.

Die gleichen Testdaten wurden unter Verwendung anderer neuronaler Netze untersucht. Dabei kamen die in Abschnitt 4.2.3 (Seite 79) besprochenen Netze mit einem Ausgabeneuron zum Einsatz. Diese Netze sind ebenfalls auf einzelne Aminosäuren spezialisiert, verwenden aber nur einen einzigen Vorhersagewert zur Zuordnung. Der Unterschied liegt in den drei beschriebenen Kodierungsvarianten. Zusätzlich wurde das Inkrementsystem benutzt, so daß für jede Aminosäure maximal vier Zuordnungen berechnet werden konnten. Die Zuordnungen wurden akzeptiert, wenn mindestens 60 % der Vorhersagen übereinstimmten. Die Ergebnisse für diese beiden Testreihen sind in Abbildung 30 dargestellt.

Ersichtlich ist hier, daß die absolute Erkennungsrate nicht eindeutig von der Anzahl der benutzten Netze abhängt. Für einige Datensätze werden mit vier Netzen bessere Ergebnisse erzielt, für andere hingegen mit nur einem Netz. Außerdem wird deutlich, daß mit steigender Sequenzlänge immer weniger Spuren überhaupt gefunden werden. Die Qualität der Zuordnungen, ausgedrückt durch die relative Erkennungsrate, ist hingegen bei Einsatz von vier Netzen meist besser. Der zusätzliche "Filtereffekt", der bei vier Netzen schwer einzuordnende Aminosäuren abfängt, erhöht somit die Verläßlichkeit der Zuordnungen.



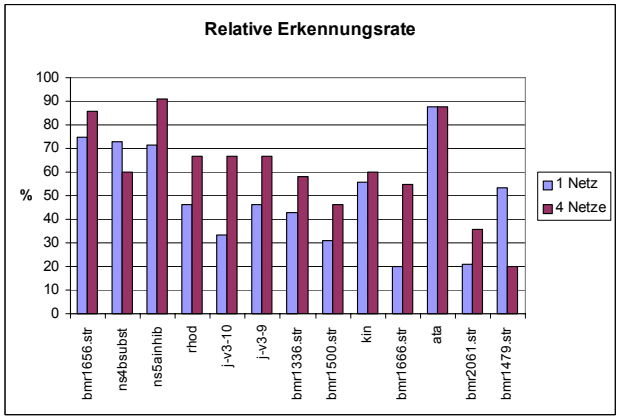


Abbildung 30: **Erkennungsraten bei der Zuordnung der Testdaten. Die Sequenzlänge der Peptide steigt nach rechts an. Das Zuordnungsverfahren wurde mit verschiedenen, auf einzelne Aminosäuren spezialisierten neuronalen Netzen durchgeführt. Violette Balken geben die Ergebnisse bei Einbeziehung von vier Netzen pro Aminosäure wieder. Blaue Balken stellen die Resultate für ein Netz pro Aminosäure dar. Die absolute Erkennungsrate gibt wieder, wie viele Signale zugeordnet wurden. Die relative Erkennungsrate gibt an, wie viele der getroffenen Zuordnungen korrekt sind.**

4.3.2 Vergleich der unspezialisierten Netze

Eine entsprechende Analyse wurde für Netze durchgeführt, die nicht auf einzelne Aminosäuren spezialisiert waren. Hier wurde ebenfalls untersucht, wie weit die Einführung von vier Vorhersagen pro Aminosäure die Ergebnisse beeinflußt. Für die Zuordnung wurden einerseits die am Ende von Abschnitt 4.2.4 (Seite 83) erwähnten Netze benutzt. Die dazu korrespondierenden Netze, die nicht mit vier Vorhersagen arbeiten, sind in Abschnitt 4.2.1 (Seite 72) beschrieben. Das Verfahren an sich blieb unverändert: von wenigstens drei Zuordnungen mußten mindestens 60 % zu identischen Ergebnissen führen. Die Resultate sind in Abbildung 31 dargestellt.

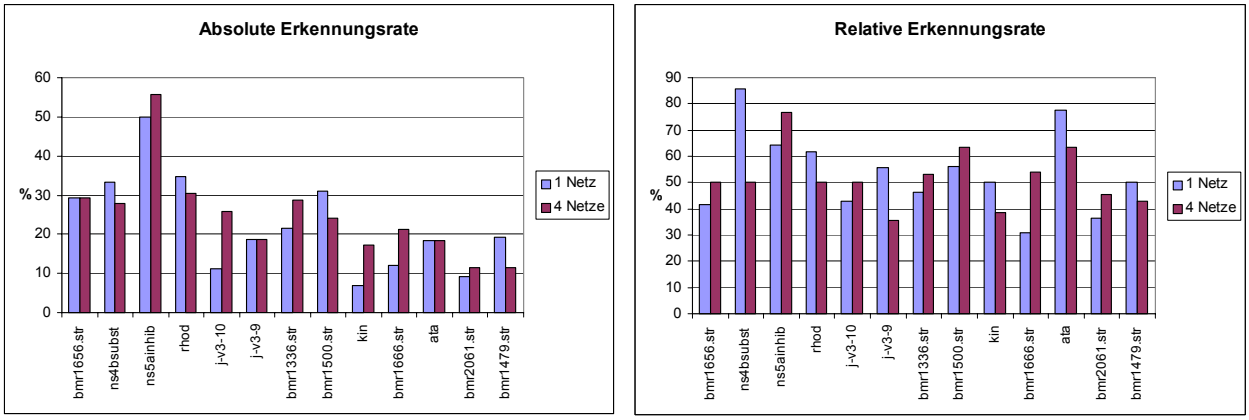


Abbildung 31: **Erkennungsraten bei der Zuordnung der Testdaten. Das Zuordnungsverfahren wurde mit neuronalen Netzen durchgeführt, die auf Erkennung sämtlicher Aminosäurearten trainiert waren. Violette Balken geben die Ergebnisse bei Einbeziehung von vier Netzen pro Aminosäure wieder. Blaue Balken stellen die Resultate für ein Netz pro Aminosäure dar. Die absolute Erkennungsrate gibt wieder, wie viele Signale zugeordnet wurden. Die relative Erkennungsrate gibt an, wie viele der getroffenen Zuordnungen korrekt sind.**

Hier läßt sich für keine Erkennungsrate ein eindeutiger Trend feststellen. Auch die Verschlechterung der Ergebnisse bei steigender Sequenzlänge ist nicht erkennbar. Für die allgemein trainierten Netze zeigt die Verwendung mehrerer Vorhersagen also keinen Effekt. Dies könnte in der zufälligen Auswahl der Trainingsdaten begründet sein. Falls in einem der vier Datensätze bestimmte Aminosäuren häufiger enthalten sind als andere, so sind für diese wenig repräsentierten Aminosäuren schlechtere Vorhersagen zu erwarten. Dies führt zu hohen Standardabweichungen der Ergebnisse. Diese Standardabweichung bildet das Kriterium zur Unterscheidung guter und schlechter Vorhersagen, welche somit nicht mehr effizient möglich ist.

4.3.3 Zusammenfassung der Ergebnisse zur Spurzuordnung

Da die Erkennungsraten für die einzelnen Testdatensätze zum Teil erheblich voneinander abweichen, wurden diese zur besseren Vergleichbarkeit der einzelnen Methoden gemittelt. Die durchschnittlichen Erkennungsraten der in diesem Abschnitt besprochenen Netze sind in Abbildung 32 aufgezeigt.

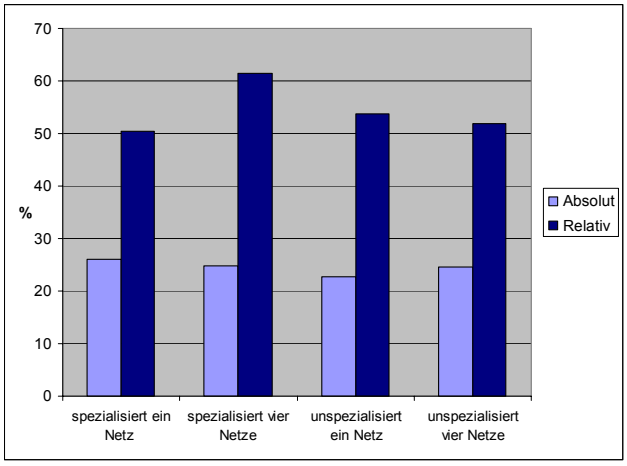


Abbildung 32: **Durchschnittliche Erkennungsraten (absolut und relativ) für die Spurzuordnung durch neuronale Netze. Die Netze unterscheiden sich in der Spezialisierung auf einzelne Aminosäuren. Außerdem wurden für die Zuordnungen entweder vier verschiedene Netze oder nur ein Netz benutzt.**

Hierbei wird deutlich, das die absolute Erkennungsrate für alle Varianten um den Wert von 25 % liegt. Jede Methode kann also ca. ein Viertel der NH/Hα-Kreuzsignale in einem Spektrum zuordnen. Die von den einzelnen neuronalen Netzen gefundenen Signalsets können dabei durchaus unterschiedlich sein.

Ein prinzipielles Problem für diese Netze ist die schlechte Abdeckung der möglichen Muster durch die Trainingsdaten. Bei einer Sequenzlänge von neun Aminosäuren können theoretisch 20^9 verschiedene Muster auftreten. Bei auf einzelne Aminosäuren spezialisierten Netzen, bei denen die zentrale Aminosäure konstant ist, bestehen noch 20^8 Varianten. Diese enorme Anzahl wird von den zur Verfügung stehenden Daten nur minimal erfaßt. Somit ist es für die neuronalen Netze kaum möglich, aus diesem stark reduziertem Datensatz allgemeine Gesetzmäßigkeiten zu ermitteln. Vielmehr neigen die Netze, wie bereits in Abschnitt 4.2.2 angesprochen, zu dem als *Overfitting* bekanntem Problem. Eine Verringerung der Sequenzlänge führt zwar zu einer besseren Repräsentation der möglichen Muster, verringert aber gleichzeitig den Informationsgehalt eines Musters, so daß eine Verschlechterung der Ergebnisse zu erwarten ist.

Entsprechende Versuche mit fünf oder drei Resten langen Sequenzfenstern (hier nicht weiter aufgezeigt) bestätigten dies.

Für die relative Erkennungsrate zeigt sich, daß die Verwendung von vier auf einzelne Aminosäuren spezialisierten Netzen die besten Ergebnisse liefert. Hier ist die relative Erkennungsrate um ca. 10 % besser als bei den anderen drei Varianten. Die Verteilung auf vier Netze alleine hat keinen deutlichen Effekt, was durch die nahezu gleichen Ergebnisse der unspezialisierten Netze (die beiden rechten Spalten in Abbildung 32) erkennbar ist. Aber auch die Spezialisierung der Netze hat für sich genommen keinen deutlichen Effekt: die erste und dritte Spalte in Abbildung 32 sind ebenfalls nahezu identisch. Erst die Kombination dieser beiden Methoden führt zu der angesprochenen, relevanten Verbesserung.

4.4 Validierung der Zuordnungen durch NOESY-Signale

Die bisher vorgestellten Methoden beruhen auf zum Teil mehrfachem Vergleich der Vorhersagen aus verschiedenen Netzen und dem Inkrementsystem. Das Hauptproblem an dieser Stelle ist es, die korrekten Zuordnungen von den falschen zu trennen. In einem konkreten Anwendungsfall hätte man nun eine Anzahl von Zuordnungen, von denen man als Grundlage für die weitere Auswertung eines Spektrums ausgehen können sollte. Die bisher erreichten relativen Erkennungsraten lassen diese Vorgehensweise allerdings nicht zu, da sie nach wie vor zu niedrig sind.

Die Zuordnung eines Kreuzsignals erfolgt erst nach einer Suche im Spektrum. Ausgehend von der Vorhersage muß also ein bestimmter "Weg" zurückgelegt werden bis ein passendes Signal gefunden wird. Die naheliegende Vermutung, daß korrekte Zuordnungen in den Fällen auftreten, in denen dieser Weg am kürzesten ist, konnte allerdings nicht bestätigt werden. Bei genauerer Betrachtung ließ sich kein Zusammenhang zwischen Richtigkeit der Zuordnung und Abstand der

Vorhersage zum zugehörigen Kreuzsignal finden. Dieses Kriterium kam für die Beurteilung der Zuordnungen somit nicht in Frage.

Auch nach längerer Analyse war es nicht möglich, diese Entscheidung nur aufgrund der vorhandenen Daten zu treffen. Von der ursprünglichen Absicht, eine sequentielle Zuordnung nur aufgrund der TOCSY-Spuren treffen zu können, mußte somit abgerückt werden. Zur Verifizierung der Vorhersagen wurden in den folgenden Schritten weitere Signale aus NOESY-Spektren herangezogen. Um den entstehenden zusätzlichen Aufwand möglichst gering zu halten, wurden nur NOE-Signale im NH/Hα-Bereich des Spektrums betrachtet. Zusätzlich wurden die hier auftretenden Signale nur als sequentielle NOEs interpretiert. Möglicherweise auftretende Signale zu in der Sequenz weiter entfernt liegenden Aminosäuren sind nicht gesondert berücksichtigt worden. Die bereits beschriebenen Algorithmen zur NOE-Analyse *NOEV-1* und *NOEV-2* (vgl. Abschnitt 3.7, Seite 50) wurden in die Zuordnung implementiert. Das so modifizierte Verfahren stellt sich nun folgendermaßen dar:

1. Für alle Aminosäuren der Sequenz wurden chemische Verschiebungen mit verschiedenen neuronalen Netzen und dem Inkrementsystem berechnet. Für jede Berechnungsmethode erhält man einen Satz Zuordnungen.
2. Jeder dieser Sätze wurde nun mit beiden NOE-Algorithmen geprüft. Man erhält nun aus jedem Satz zwei neue Zuordnungslisten.
3. Aus diesen Zuordnungen werden nun ebenfalls die akzeptiert, die ein vorher bestimmtes Maß an Übereinstimmung zeigen.

Bei Einsatz von drei neuronalen Netzen und dem Inkrementsystem erhält man somit bis zu acht Zuordnungen pro Aminosäure. Es wurde festgelegt, daß mindestens drei Zuordnungen vorhanden sein mußten. Weiterhin mußten mindestens zwei der drei getroffenen Zuordnungen übereinstimmen. In Tabelle 27 ist die Zuordnung eines einzelnen Datensatzes mit diesem Verfahren dargestellt. Benutzt wurden die drei neuronalen Netze in verschiedenen Kodierungsvarianten und das Inkrementsystem.

Amino-säure	*Komp. NOEV1*	*Komp. NOEV2*	*Bits. NOEV1*	*Bits. NOEV2*	*Stand. NOEV1*	*Stand. NOEV2*	*Inkr. NOEV1*	*Inkr. NOEV2*	*Zuord-nung*	*Soll-wert*
K1										
G2				8			0	0	**0**	0
C3	1	1			1	1	1	1	**1**	1
W4	2	2		2	2	2		2	**2**	2
K5					3					3
C6										4
G7										5
K8										6
E9										7
G10				0						8
H11				1						9
Q12										10
M13	11	11	11	11	11	11	11	11	**11**	11
K14	12	12	12	12	12	12	12	12	**12**	12
D15		13	13	13		13		13	**13**	13
C16	14	14		14		14		14	**14**	14
T17	15	15	15	15	15	15	15	15	**15**	15
E18	16	16	16	16	16	16	16	16	**16**	16

Tabelle 27: **Zuordnung des Datensatzes bmr1656.str mit drei verschiedenen neuronalen Netzen und dem Inkrementsystem. Jede Nummer steht für ein Kreuzsignal, die korrekten Sollwerte sind in der entsprechenden Spalte angegeben. Die Numerierung erfolgte automatisch bei der Eingabe der Signale und ist willkürlich. Da die NH-Verschiebung des N-terminalen Lysins nicht bekannt war, existiert für das entsprechende Signal kein Sollwert. Die Zuordnungen wurden mit zwei verschiedenen NOE-Analysealgorithmen verifiziert. Wenn mindestens drei Methoden eine Aminosäure zuordnen konnten und mindestens 60% dieser Zuordnungen übereinstimmten, wurde das Ergebnis akzeptiert. Sämtliche akzeptierten Zuordnungen sind korrekt.**

Für dieses Beispiel ergibt sich eine absolute Erkennungsrate von ca. 53 %. Die relative Erkennungsrate beträgt durch die zusätzlichen

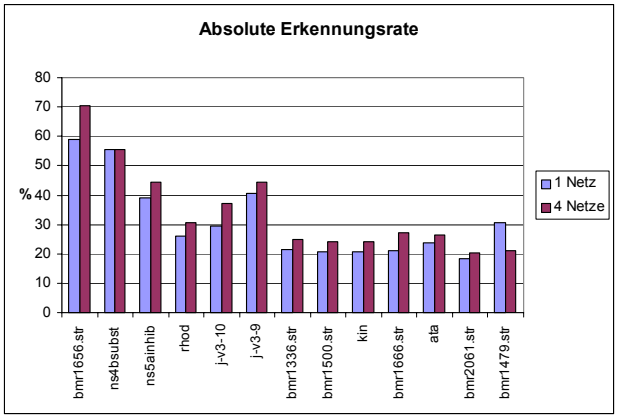
Informationen aus NOE-Daten 100 %. Im Vergleich mit den Ergebnissen aus Tabelle 25 (Seite 91) fällt auf, daß sämtliche falschen Zuordnungen eliminiert oder korrigiert werden konnten. Zudem wurden genauso viele Kreuzsignale überhaupt zugeordnet. Diese Werte stellen somit den gewünschten Startpunkt einer weiterführenden Spektrenauswertung dar.

Für die NOE-validierte Zuordnung wurden im Anschluß die gleichen Tests durchgeführt; es wurden also spezialisierte und unspezialisierte Netze verwendet. Weiterhin wurde der Effekt von vier Netzen gegenüber einem Netz pro Aminosäure untersucht.

4.4.1 Spezialisierte Netze mit NOE-Validierung

Die chemischen Verschiebungen der Testdatensätze wurden mit auf die Berechnung einzelner Aminosäuren spezialisierten neuronalen Netzen vorhergesagt. Dabei wurden in einem Versuch einzelne Netze benutzt; in einem anderen wurden jeweils vier Netze pro Aminosäure verwendet (vgl. Abschnitt 3.6, Seite 46). Alle Vorhersagen wurden dann in zwei Varianten mit NOE-Daten abgeglichen. Schließlich wurden die Zuordnungen miteinander verglichen und bei einer Übereinstimmung von mindestens 60 % akzeptiert, wobei mindestens drei Zuordnungen überhaupt vorhanden sein mußten. Absolute und relative Erkennungsraten für die einzelnen Datensätze sind in Abbildung 33 aufgezeigt.

Es zeigt sich ein ähnliches Bild wie bei den unvalidierten Zuordnungen. Die absolute Erkennungsrate nimmt mit steigender Sequenzlänge ab, die relative Erkennungsrate ist bei Einsatz von vier Netzen pro Aminosäure meist besser. So wird der Maximalwert von 100 % hier in sechs Fällen erreicht, bei einem verwendeten Netz nur in drei Fällen. Vor allem bei den relativen Erkennungsraten tritt nach der Einbeziehung von NOE-Daten eine deutliche Verbesserung auf.



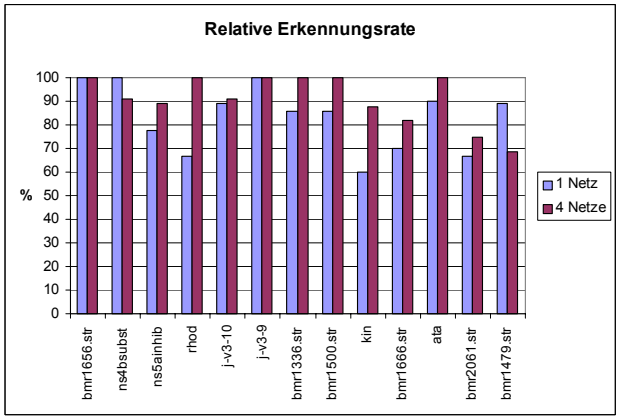
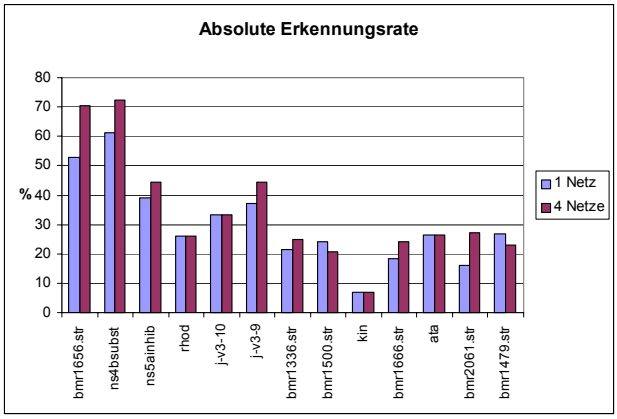


Abbildung 33: **Ergebnisse für die NOE-validierte Zuordnung mit spezialisierten Netzen. Verglichen sind Zuordnungen, die pro Aminosäure vier Vorhersagen benutzen mit Zuordnungen, die auf nur einer Vorhersage pro Aminosäure beruhen. Die absolute Erkennungsrate gibt wieder, wie viele Signale zugeordnet wurden. Die relative Erkennungsrate gibt an, wie viele der getroffenen Zuordnungen korrekt sind.**

4.4.2 Unspezialisierte Netze mit NOE-Validierung

Die entsprechenden Tests wurden ebenfalls mit den korrespondierenden neuronalen Netzen durchgeführt, die alle Aminosäuren berechnen konnten. Die sonstigen Parameter, wie zum Beispiel die benötigte Mindestanzahl der Zuordnungen, wurden unverändert übernommen (Abbildung 34).

Die relativen Erkennungsraten werden durch Einbeziehung von NOE-Daten ebenfalls deutlich besser, ohne allerdings die Qualität der spezialisierten Netze zu erreichen. Für die absolute Erkennungsrate lässt sich auch hier keine eindeutige Aussage bezüglich der Sequenzlänge treffen. Erkennbar ist allerdings, daß mit vier Netzen im allgemeinen bessere Ergebnisse erzielt werden.



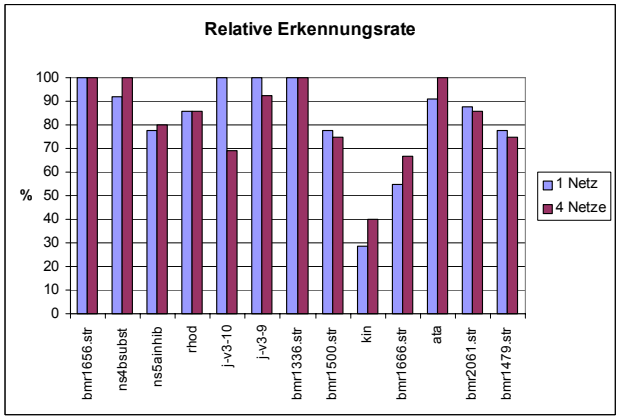


Abbildung 34: **Ergebnisse der NOE-validierten Zuordnung mit neuronalen Netzen, die nicht auf einzelne Aminosäuren spezialisiert waren. Verglichen sind Zuordnungen, die pro Aminosäure vier Vorhersagen benutzen mit Zuordnungen, die auf nur einer Vorhersage pro Aminosäure beruhen. Die absolute Erkennungsrate gibt wieder, wie viele Signale zugeordnet wurden. Die relative Erkennungsrate gibt an, wie viele der getroffenen Zuordnungen korrekt sind.**

4.4.3 Zusammenfassung der Ergebnisse zur NOE-Validierung

Auch hier wurden nun zur besseren Abschätzung die über alle Testdaten erreichten Erkennungsraten gemittelt. Eine graphische Darstellung dieser Mittelwerte findet sich in Abbildung 35.

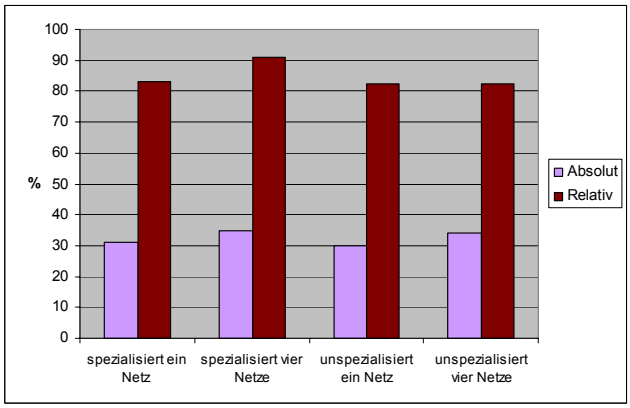


Abbildung 35: **Gemittelte absolute und relative Erkennungsraten der neuronalen Netze bei Einbeziehung der NOE-Daten. Die eingesetzten neuronalen Netze sind dieselben wie in Abschnitt 4.3.3, lediglich die Validierung der Ergebnisse durch weitere Kreuzsignale aus NOESY-Spektren wurde zusätzlich durchgeführt.**

Im Vergleich zu den Ergebnissen ohne NOESY-Daten ist die absolute Erkennungsrate bei allen Netzen um etwa fünf Prozent besser, wobei auch hier keine Methode ein deutliches Maximum aufweist. Bei den relativen Erkennungsraten zeigt sich ebenfalls ein ähnliches Bild: die Ergebnisse sind für vier spezialisierte Netze am besten und erreichen Werte von ca. 90 %. Die zusätzlichen Daten bringen somit eine deutliche Verbesserung, es werden mehr Signale mit einer deutlich höheren Verläßlichkeit zugeordnet. Dies wirft die Frage auf, wie weit eine Zuordnung nur durch die NOE-Informationen möglich ist. Versuche mit komplett zufällig getroffenen Zuordnungen, welche dann mit NOE-Daten verifiziert wurden (nicht gesondert dargestellt), zeigten allerdings, das so nur relative Erkennungsraten von ca. 65 % erreicht wurden. Der Einfluß der neuronalen Netze ist also deutlich erkennbar.

4.5 Überblick

In den letzten Abschnitten wurden mehrere Varianten dargestellt, einzelne Aminosäuren in einer bekannten Sequenz den entsprechenden Kreuzsignalen im NH/Hα-Bereich von TOCSY-Spektren zuzuordnen. Dabei stellte sich heraus, daß eine Methode, die für ein Beispiel die besten Ergebnisse liefert, nicht notwendigerweise für einen anderen Datensatz ebenfalls die optimale Lösung darstellt. Um aus allen geprüften Ansätzen nun den besten auswählen zu können, wurden die durchschnittlichen Erkennungsraten ermittelt. Dazu wurden für jeden Ansatz die absoluten und relativen Erkennungsraten aufsummiert und durch die Anzahl der Muster im Testdatensatz dividiert. So lässt sich für künftige Anwendungen der ideale Algorithmus finden.

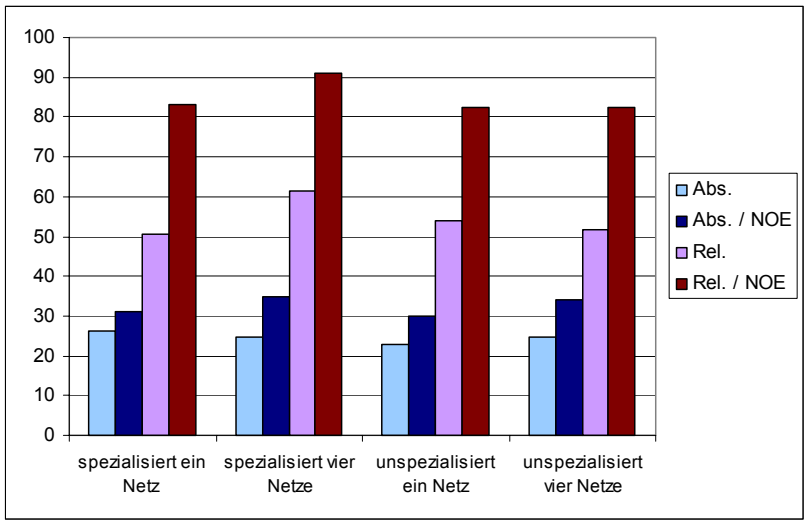


Abbildung 36: **Überblick über die Leistungsfähigkeit der verwendeten Verfahren. Dargestellt sind die durchschnittlichen absoluten und relativen Erkennungsraten, die mit verschiedenen neuronalen Netzen erzielt werden. Die Einbeziehung von NOE-Daten bringt eine deutliche Verbesserung.**

In Abbildung 36 sind die diskutierten Verfahren dargestellt. Prinzipiell unterscheiden sich diese in drei Punkten:

- es können auf einzelne Aminosäuren spezialisierte Netze oder Netze zur Berechnung aller 20 Aminosäuren verwendet werden.
- die Berechnung der chemischen Verschiebungen erfolgt durch ein einziges Netz oder durch Mittelwertbildung der Vorhersagen von vier verschiedenen Netzen.
- zur Verifizierung der Zuordnungen werden Daten aus NOESY-Spektren herangezogen oder nicht.

Wie bereits erwähnt, ist die relative Erkennungsrate das wichtigere Kriterium, da sie Aussagen über die Verläßlichkeit der getroffenen Zuordnungen zulässt. Vor diesem Hintergrund stellt sich die Methode vier spezialisierte Netze zu verwenden als die beste dar. Sie liefert sowohl ohne als auch mit NOE-Daten die höchste relative Erkennungsrate von 61 % bzw. 91 %. Zur besseren Übersicht wird diese Methode nun noch einmal detailliert beschrieben.

Voraussetzung ist eine vorhergehende Klassifizierung der Spuren zu den zwölf eingeführten Aminosäureklassen (vgl. Tabelle 2). Dies ist mit den zu diesem Zweck trainierten neuronalen Netzen und anschließendem Vergleich mit den in der Sequenz enthaltenen Aminosäuren durchführbar. Findet das neuronale Netz beispielsweise für eine Sequenz, in der Glycin und Serin jeweils zwei mal auftreten, drei Glycinreste und ein Serin, so liegt die Vermutung nah, daß eine der drei hypothetischen Glycinspuren tatsächlich ein Serin ist. Durch genauere Analyse der entsprechenden Spuren lassen sich solche Fehler manuell korrigieren. Nach diesem Schritt kann eine Liste mit NH/Hα-Kreuzsignalen, denen eine Aminosäureklasse zugeordnet ist, erstellt werden.

Die Sequenz wird nun mit einem neun Reste breiten Fenster ausgelesen. Die Aminosäure, deren NH- und Hα-Verschiebungen vorhergesagt werden soll, befindet sich in der zentralen Position dieses Fensters. Dabei werden die Muster für jede Aminosäure erzeugt. Je nach Aminosäure werden die zugehörigen vier neuronalen Netze verwendet, um die chemischen Verschiebungen zu berechnen. Weiterhin werden Mittelwerte und Standardabweichungen für diese vier Vorhersagen bestimmt. Zunächst werden die Vorhersagen betrachtet, bei denen die Standardabweichung unter einem vorgegebenem Limit liegt. Die Mittelwerte dieser Vorhersagen bilden den Startpunkt für die Suche im Spektrum nach passenden Signalen. Hier werden auch schon Signale, die in der gleichen Aminosäureklasse enthalten sind, als korrekte Treffer akzeptiert. Soll beispielsweise eine Leucinspur vorhergesagt werden, so gelten Leucin- oder Isoleucinsignale als korrekte Treffer. Nach einem Durchlauf wird das Limit für die zulässige Standardabweichung erhöht und für die nächsten Vorhersagen werden korrelierende Signale gesucht. Ein Flußdiagramm für diesen Algorithmus ist in Abbildung 17 (Seite 49) gezeigt. Nach dieser Suche liegen zu den Signalen Zuordnungen vor.

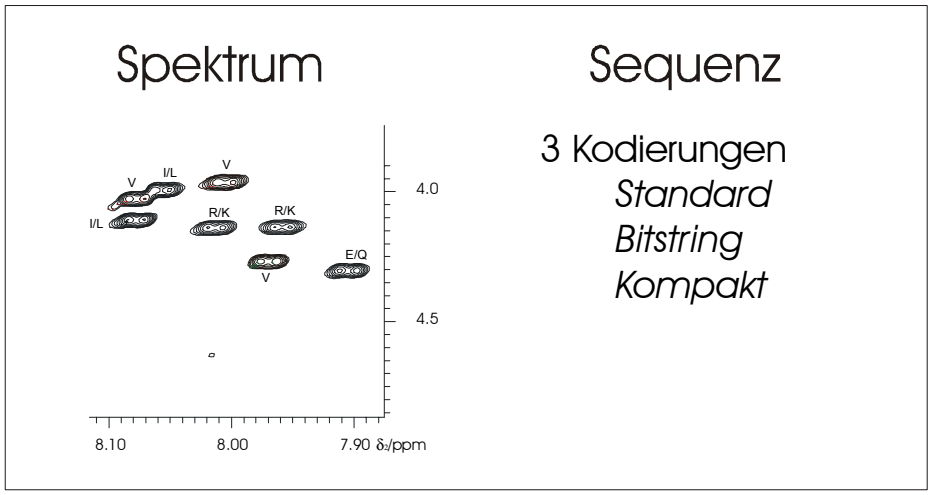
Die Zuordnungen werden im nächsten Schritt mit NOE-Daten validiert. Hierfür stehen zwei Methoden zur Verfügung, die in Abschnitt 3.7 näher beschrieben sind. Die beschriebenen Schritte werden für alle drei Kodierungsvarianten durchgeführt. Zusätzlich werden sämtliche

chemischen Verschiebungen einmal von dem Inkrementsystem berechnet und entsprechende Signale im Spektrum gesucht. Die Gesamtzahl an möglichen Zuordnungen pro Signal beträgt somit acht (drei Kodierungen und eine Zuordnung aus dem Inkrementsystem mit jeweils zwei NOE-Validierungen).

Liegen nun für ein Signal eine gewisse Mindestanzahl an Zuordnungen vor und stimmen diese zu einem bestimmten Prozentsatz überein, so wird die Zuordnung akzeptiert. Alle anderen Zuordnungen werden als nicht verifiziert oder nicht übereinstimmend zurückgewiesen. Abbildung 37 (Seite 108) ist eine schematische Darstellung der Methode.

Dieses Verfahren führt, verglichen mit Methoden ohne NOE-Validierung, zwar zu weniger zugeordneten Spuren, die Zuverlässigkeit der getroffenen Aussagen ist dafür hoch. Die erforderliche Vorarbeit ist die Erstellung von Peaklisten für die einzelnen Spuren. Außerdem müssen die Kreuzsignale im NH/Hα-Bereich des NOESY-Spektrums gesucht und ebenfalls zu einer Liste zusammengefaßt werden. Dies kann mit automatisierten Peak-Picking-Verfahren verhältnismäßig schnell durchgeführt werden. Nach Ermittlung des Aminosäuretyps müssen die Resultate zunächst mit der Sequenz in Übereinstimmung gebracht werden. Die anschließende sequentielle Zuordnung der klassifizierten Spuren erfolgt dann in wenigen Minuten.

Die erhaltenen Ergebnisse bilden eine robuste Arbeitshypothese, die die Grundlage für eine anschließende manuelle Auswertung darstellt. Der Aminosäuretyp der einzelnen Spuren ist nun bekannt, eine sequentielle Zuordnung liegt zum Teil vor. Zusätzlich sind bereits einige NOE-Signale als sequentielle NOEs identifiziert. Noch nicht zugeordnete Spuren können nun im Ausschlußverfahren oder durch NOE-Signale außerhalb des NH/Hα-Bereichs identifiziert werden.



Zuordnung durch 4 neuronale Netze pro Kodierung und Aminosäure sowie das Inkrementsystem

Spur	*Standard*	*Kompakt*	*Bitstring*	*Inkrement*
1	Val 3	Val 3	Val 5	Val 3
2	Glu 4	Glu 7	Glu 7	Glu 4
..	..	..	..	..

NOE Analyse mit 2 Algorithmen

Spur	*Standard NOE 1*	*Standard NOE 2*	*Kompakt NOE 1*	*Kompakt NOE 2*	*Bitstring NOE 1*	*Bitstring NOE 2*	*Inkrement NOE 1*	*Inkrement NOE 2*
1	Val 3	Val 3	Val 3	Val 3	Val 3	Val 3	Val 3	Val 3
2	Glu 4	Glu 4	Glu 7	Glu 4	Glu 7	Glu 7	Glu 4	Glu 7
..	..	..	..	..	..	..	..	..

Vergleich der Ergebnisse und endgültige Zuordnung

Spur	*Standard NOE 1*	*Standard NOE 2*	*Kompakt NOE 1*	*Kompakt NOE 2*	*Bitstring NOE 1*	*Bitstring NOE 2*	*Inkrement NOE 1*	*Inkrement NOE 2*	***Zuordnung***
1	Val 3	Val 3	Val 3	Val 3	Val 3	Val 3	Val 3	Val 3	**Val 3**
2	Glu 4	Glu 4	Glu 7	Glu 4	Glu 7	Glu 7	Glu 4	Glu 7	**???**
..	..	..	..	..	..	..	..	..	..

Abbildung 37: **Schematische Darstellung der optimalen Zuordnungsmethode. Die verwendeten neuronalen Netze sind auf einzelne Aminosäuren spezialisiert. Pro Seitenkette werden vier Vorhersagen getroffen und der Mittelwert der berechneten chemischen Verschiebungen dient als Ausgangspunkt für die Suche im Spektrum. Die Zuordnungen werden mit NOE-Daten abgeglichen.**

Die Anwendung dieser Methode auf das Spektrum des im Arbeitskreis dargestellten Glycopeptids *j-v3-10* wird in Abbildung 39 gezeigt. Das Glycopeptid besteht aus 29 Aminosäuren und ist an zwei Positionen mit einem Chitobiosylrest glycosyliert. Die Primärstruktur ist in Abbildung 38 dargestellt.

AcNH-His-Leu-Asn*-Glu-Ser-Val-Glu-Ile-Asn-Thr-Thr-Arg-Pro-Ser-Asn*-Asn-Thr-Arg-Lys-Ser-Ile-His-Ile-Gly-Pro-Gly-Arg-Ala-Phe-$CONH_2$

Abbildung 38: **Primärsequenz des Glycopeptids *j-v3-10*. Die mit einem Stern markierten Asparaginreste sind mit einem Chitobiosylrest glycosyliert. Korrekt zugeordnete Aminosäuren sind türkis hinterlegt, die falsche Zuordnung von Threonin 10 grau.**

Im NH-Bereich des TOCSY-Spektrums sind 27 Aminosäurespuren sichtbar. Von diesen 27 Spuren können elf zugeordnet werden, das entspricht einer absoluten Erkennungsrate von 37 %. Die Zuordnung von zehn Spuren ist dabei korrekt, die relative Erkennungsrate beträgt somit 91 %. Die Spur, die Threonin 10 zugeordnet wurde, gehört korrekterweise zu Threonin 17. In beiden Fällen liegt vor dem Threonin ein Asparaginrest in der Sequenz. Da die NOE-Validierung nur Kontakte zu der in der Sequenz vorherliegenden Aminosäure berücksichtigt, kann solch ein Fehler nicht erkannt werden.

Die getroffene Zuordnung kann nun als Ausgangspunkt für die weitere manuelle Analyse des Spektrums dienen. Hierbei werden auch weiterführende NOE-Kontakte berücksichtigt. Der aufgetretene Fehler kann somit schnell gefunden und korrigiert werden.

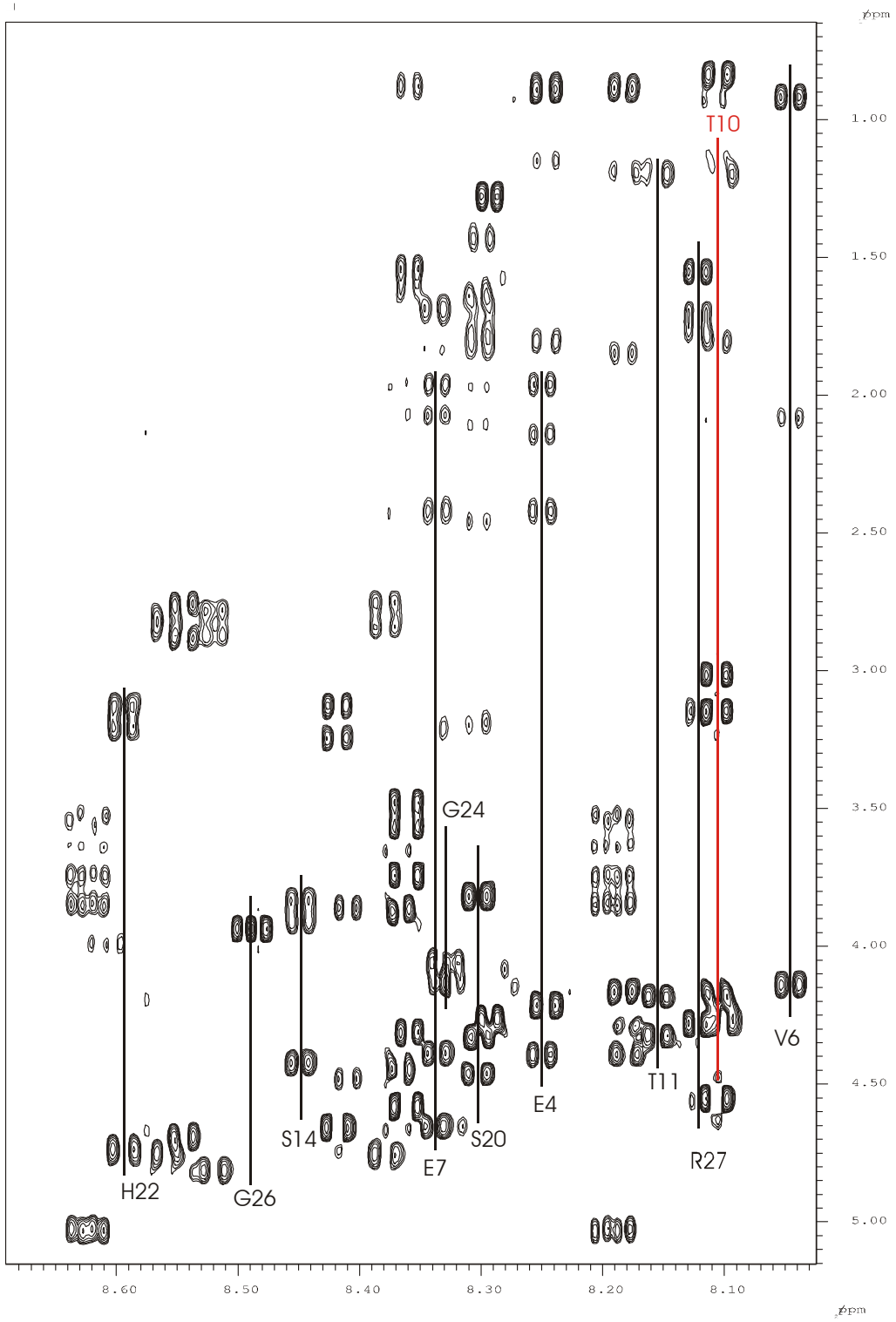


Abbildung 39: **Ausschnitt aus dem TOCSY-Spektrum eines 29 Aminosäuren langen Glycopeptids (H_2O/D_2O 9:1, 300 K, pH=3.5). Eingezeichnet sind die Spuren, die mit der im Text erläuterten Zuordnungsmethode gefunden wurden. Mit Ausnahme der rot markierten Spur für Threonin 10 sind alle Zuordnungen korrekt. Die absolute Erkennungsrate beträgt in diesem Fall 37 %, die relative Erkennungsrate 91 %.**

5 Zusammenfassung

Die NMR-spektroskopische Strukturaufklärung von Peptiden erfordert die Analyse komplexer, mehrdimensionaler Spektren. Hierfür müssen als erstes die Signale in den Spektren eindeutig zugeordnet werden. Typischerweise wird die Interpretation im NH-Bereich von TOCSY-Spektren begonnen, da hier eine relativ gute Dispersion vorliegt. Jede Aminosäure eines Peptids, mit Ausnahme von Prolin, zeigt eine charakteristische Spur. Auf dieser sind im Regelfall ausgehend von dem NH-Signal alle weiteren Signale der betreffenden Aminosäure zu finden. Sowohl die Lage der NH- als auch der Hα-Signale sind stark sequenzabhängig. Da die manuelle Zuordnung der Signale einen wesentlichen Zeitfaktor bei der Interpretation der Spektren darstellt, ist es von Interesse, diesen Vorgang zu automatisieren.

In dieser Arbeit wurde untersucht, in wie weit künstliche neuronale Netze in der Lage sind, diese Zuordnung durchzuführen oder zu erleichtern. Ziel war es, aus den automatisch generierten Listen der Peaks im TOCSY Spektrum einen Vorschlag für den Typ der Aminosäure und für die Position im Peptid zu erhalten.

Zunächst sollte aus diesen Peaklisten automatisch der Typ der Aminosäuren bestimmt werden. Hierzu wurden neuronale Netze entworfen, deren Eingabeneuronen die chemische Verschiebung repräsentieren. Da hier nur die Spuren im NH Bereich ausgewertet wurden, ist mit 650 Eingabeneuronen der Bereich von 0 bis 6.5 ppm abgebildet worden. Zum Training der neuronalen Netze wurden Daten aus der *BioMagResBank,* die die NMR Spektren von ca. 1700 Proteinen und Peptiden enthält, verwendet.

Zur Bestimmung des Aminosäuretyps wurden verschiedene neuronale Netze getestet. Der Hauptunterschied dieser Netze lag in dem Verfahren zur Erzeugung der Trainingsmuster. Ein Ansatz, bei dem die Trainingsdaten auf der beobachteten statistischen Verteilung der jeweiligen chemischen Verschiebung beruhten, ergab nur eine

Erkennungsrate von 35 %. Aminosäuren, deren Seitenketten sehr ähnliche NMR Spektren liefern, wie z.B. Asparaginsäure und Asparagin, wurden daraufhin in Gruppen zusammengefaßt, wodurch sich insgesamt zwölf Klassen ergaben. Hierdurch konnte die Erkennungsrate auf 60 % erhöht werden. In einem weiteren Ansatz wurden die Muster direkt aus den Spektren der Datenbank erzeugt. Dabei wurden aus einzelnen Mustern durch geringe Variation der Signale weitere Muster erzeugt. Hier wurden Erkennungsraten von 40 % bzw. 65 % bei Gruppierung der Aminosäuren erreicht. Der beste Ansatz benutzte die in der Datenbank abgelegten Spektren, denen künstlich eine größere, dreiecksförmige Linienbreite von 0.05 ppm gegeben wurde. Faßt man auch hier ähnliche Aminosäuren zusammen, so liefern die neuronalen Netze Erkennungsraten von 80 % bis 90 %, je nach präsentiertem Peptid.

Als zweiter Schritt bei einer automatischen Interpretation ist die Zuordnung der jeweiligen Aminosäure zu der sequentiellen Position notwendig. Hierfür wurden drei verschiedene Varianten getestet, um die Aminosäuresequenz auf der Eingabeschicht eines neuronalen Netzes abzubilden. Es zeigte sich, daß die besten Ergebnisse mit neuronalen Netzen erzielt werden, die auf die Vorhersage der chemischen Verschiebungen einzelner Aminosäuren spezialisiert sind. Die beste Kodierung für die Struktur der jeweiligen Seitenkette war eine Abfolge einzelner Bits, die die funktionellen Gruppen respektive Atome kodieren. Die Standardabweichung σ für die auftretenden Fehler bei der Vorhersage eines Testdatensatzes lag hier bei 0.57 ppm für Amidprotonen bzw. 0.38 ppm für Hα-Protonen. Um inkohärente Vorhersagen zu behandeln, wurde ein Zuordnungsalgorithmus, der die Vorhersagen von vier mit unterschiedlichen Mustern trainierten Netzen benutzt, erarbeitet. Zusätzlich zu den neuronalen Netzen wurde aus den verfügbaren Daten ein Inkrementsystem zur Berechnung der chemischen Verschiebungen von amidischen und Hα-Protonen entwickelt. Dieses Inkrementsystem kann die chemischen Verschiebungen mit einer ähnlichen Genauigkeit berechnen wie die oben erwähnten neuronalen Netze (σ_{NH} = 0.53 ppm, $\sigma_{H\alpha}$ = 0.37 ppm).

Diese Genauigkeit ist für eine akkurate Vorhersage der Kreuzsignale allerdings nicht ausreichend. Die vorhergesagte Position eines Signals kann jedoch als Startpunkt für eine Suche im Spektrum herangezogen werden. Da der Typ der gesuchten Aminosäure bekannt ist, kann das Kreuzsignal, das dieser Aminosäure entspricht und am dichtesten an der Vorhersage liegt, der entsprechenden Seitenkette in der Sequenz zugeordnet werden. Die beste Methode benutzt zur Berechnung jeweils vier auf einzelne Aminosäuren spezialisierte Netze und das Inkrementsystem. Damit können ca. 25 % der Signale zugeordnet werden (absolute Erkennungsrate). Von diesen Zuordnungen sind im Durchschnitt 61 % korrekt (relative Erkennungsrate).

Um die relative Erkennungsrate zu verbessern, wurden zusätzlich NOE-Daten verwendet. Diese geben weitere Informationen über Konnektivitäten innerhalb der Signale eines Spektrums. Dabei wurde nur der NH/Hα-Bereich in den entsprechenden NOESY-Spektren betrachtet und nur sequentielle NOEs berücksichtigt. Nach Tests zeigte sich, daß durch die zusätzliche Information falsche Zuordnungen weitgehend vermieden werden können. Signalzuordnungen, die durch Einsatz neuronaler Netze und der NOE-Information gewonnen werden, wurden akzeptiert, wenn mindestens 60 % der Methoden dieselbe Vorhersage lieferten. Dadurch steigt die absolute Erkennungsrate auf 31 % und die relative Erkennungsrate auf 91 %.

Neuronale Netze stellen somit ein Hilfsmittel dar, den Aminosäuretyp einer Spur automatisch zu bestimmen und für etwa ein Drittel der Aminosäuren auch deren Position in der Sequenz festzulegen. Mit diesen Eckpunkten ist eine weitere manuelle Zuordnung der Spektren wesentlich leichter und schneller durchführbar.

6 Summary

NMR based structure determination of peptides requires the analysis of complex, multidimensional spectra. First of all, the signals in the spectra must be unambiguously assigned. The NH region in TOCSY-spectra is usually the starting point for theses tasks due to the comparatively good dispersion in this area. Each amino acid other than proline gives rise to one characteristic trace in this region. Within this trace all signals belonging to one amino acid can be found emanating from the NH Signal. The chemical shifts of both NH and Hα protons depend strongly on the peptide sequence. Since the manual assignment of the signals is the major bottleneck in spectra analysis, an automation of this process is highly desirable.

The ability of artificial neural networks to perform or, failing this, facilitate this assignment was investigated in this work. Automatically generated peak lists from TOCSY spectra were to be used to obtain a sound proposal for the type of amino acid and the position within the sequence.

In the first step the type of amino acid should be determined automatically. To this end, artificial neural networks were designed which represent chemical shifts on their input layer. Since only traces within the NH region were analysed only the spectral range from 0 to 6.5 ppm was mapped to an input layer consisting of 650 neurons. Data for the training of neural networks was obtained from the *BioMagResBank*, which holds spectra of about 1700 proteins and peptides.

Various neural networks were tested to determine the amino acid type. The main difference between these networks was the method used in generating the patterns for training of the network. One approach, which used the observed statistical distribution of chemical shifts for the creation of patterns, resulted in recognition rates of 35 %. Some amino acids, e.g. aspartic acid and asparagine, give rise to nearly identical traces. Utilizing a classification of the 20 amino acids into subclasses

containing such similar residues, the recognition rate could be raised to 60 %. In another approach the patterns were generated out of the spectra deposited in the database. Here several additional patterns were generated out of one dataset by varying the chemical shift values by small amounts. The recognition rates of these networks were 40 % respectively 65 % when the aforementioned classification of amino acids was used. The best approach was to use the deposited spectra while artificially broadening the lines to 0.05 ppm with a superimposed triangle. Here the best recognition rates lie between 80 and 90 percent, depending on the presented peptide.

The second step in an automated interpretation is the sequential assignment of the amino acids. Three different coding schemes for the mapping of an amino acid sequence to the input layer of a neural network were developed for this task. Neural Networks, which were specialised for the prediction of chemical shifts of single amino acids, proved to perform best. The best representation of the structure of amino acids was a sequence of bits representing functional groups or atoms in the sidechain. The standard deviation σ of the error occurring when predicting a testset was 0.57 ppm for amidic protons and 0.38 ppm for Hα-protons. To treat incoherent predictions, an algorithm which averages the results of four different networks was developed. In addition to the neural networks an increment system for the fast calculation of chemical shifts was derived from the available data. The performance of this increment system was similar to the neural networks described before (σ_{NH} = 0.53 ppm, $\sigma_{H\alpha}$ = 0.37 ppm).

The precision of these methods is not sufficient for an accurate prediction of spectra. However, the predicted position of a crosspeak can serve as a starting point for a search within the spectrum. Since the type of the sought amino acid is known, the appropriate crosspeak which is nearest to the prediction can be assigned to the corresponding residue in the sequence. The best method uses four specialized neural networks and the increment system for each amino acid. With this combination, about 25 % of all signals can be assigned (absolute recognition rate). Out of

these assignments an average of 61 % are correct (relative recognition rate).

Additional NOE-data was used to increase the relative recognition rate. This data provides information about connectivities between signals in a spectrum. Only the NH/Hα-area was taken into account and all signals were taken to be sequential NOEs. Using this additional information, most wrong assignments could be prevented. Assignments, which were based on different neural nets and NOE-validations, were accepted when at least 60 % of all used methods yielded the same result. Thereby, the absolute recognition rate could be raised to 31 %. The relative recognition rate reached 91 % under this conditions.

Neural Networks thus can aid in the automatic determination of amino acid type and the sequential assignment of about one third of the TOCSY traces. The resulting assignments can be used as a vantage point for further analysis of a spectrum, which can thus be accomplished easier and faster.

7 Experimenteller Teil

7.1 Verwendete Hard- und Software

Die neuronalen Netze standen in Form eines von Dr. Helge Kränz in Fortran77 entwickelten und optimierten Programms[106] zur Verfügung. Der Quellcode wurde zur Anpassung an die eingesetzten Rechnertypen mit dem MIPSpro Fortran77 Compiler (Version 7.2.1) neu übersetzt.

Sämtliche Operationen zur Vorbereitung der Eingabedaten, der Analyse der Ausgabedaten und der NOE-Validierung erfolgten in der Programmiersprache PERL (Practical Extraction and Report Language)[107,108]. Um bestimmte Abschnitte innerhalb der Datensätze der *BioMagResBank* schnell zu finden und weiter zu verarbeiten, wurde das Modul *Text::DelimMatch*[109] von Norman Walsh benutzt. Für statistische Funktionen bei der Auswertung wurde das Modul *Statistics::Descriptive*[110] von Colin Kuskie eingesetzt. Die Programme zur Entwicklung des Inkrementsystems wurden ebenfalls in PERL geschrieben.

Training und Test der neuronalen Netze erfolgten auf einem Parallelrechner der Firma SGI (Power Challenge, 18 MIPS R10000 195 MHz Prozessoren) oder SGI Octane Workstations (2 MIPS R10000 250 MHz Prozessoren). Die Verarbeitung der Datensätze der *BioMagResBank* und die erste Auswertung der Ergebnisse sowie die Entwicklung des Inkrementsystems wurde ebenfalls auf SGI Octane Workstations durchgeführt.

Alle angesprochenen Programme und Skripte, sowie der Quellcode des neuronalen Netzes, sind als elektronischer Anhang auf CD beigelegt.

Graphische und statistische Auswertung und Aufbereitung erfolgte auf handelsüblichen PCs mit den Softwarepaketen Microcal Origin und Microsoft Excel.

7.2 Mustererzeugung

Zur Erstellung der Muster für Trainings- und Testdatensätze wurden Daten aus der *BioMagResBank*[79] verwendet. Dazu wurden im März 1999 alle damals verfügbaren Datensätze als Textdateien abgerufen. Insgesamt standen 1357 Einträge zur Verfügung. In diesen Datensätzen sind unter anderem die chemischen Verschiebungen in Form von Signallisten abgelegt. Um diese Informationen in die entsprechenden Muster zu übersetzen, wurde das Programm *cdb.perl* entwickelt. Mit Hilfe dieses Programms konnten Muster zur Signalvorhersage erzeugt werden. Die benötigten Datenstrukturen sind in den Modulen *nmrdata.pm*, *pdbdata.pm* und *pattern.pm* definiert, welche von *cdb.perl* benutzt werden. Die beschriebenen Kodierungsverfahren und Parameter wurden im Modul *codetable.pm* implementiert und konnten beliebig eingesetzt und modifiziert werden. Die Erzeugung eines Mustersatzes nahm im Regelfall nur wenige Minuten in Anspruch.

Auch für die in Abschnitt 3.2.1 beschriebene statistische Methode wurde ein entsprechendes Programm (*gentracepat.perl*) geschrieben. Dies ermöglichte die Erzeugung einer beliebigen Anzahl von Mustern, die den statistischen Randbedingungen entsprachen.

Die Erzeugung von Mustern zu Spurzuordnung aus realen Daten wurde mit dem Programm *bmrtotrace_fuzzy.perl* durchgeführt. Die benötigten Datenstrukturen sind im Modul *spinsys.pm* definiert.

7.3 Training

Für das Training der neuronalen Netze wurde ein standardisiertes Protokoll verwendet. Alle Gewichte und Schwellwerte wurden zu Beginn mit zufälligen Werten zwischen –0.05 und 0.05 initialisiert. Die anfängliche Lernrate wurde auf 2.0 gesetzt und alle 500 Zyklen um 0.02 verringert. Das Momentum wurde während des Trainings konstant auf einem Wert von 0.5 gesetzt. Als Abbruchkriterium wurde ein RMSD-Wert von 0.001 festgelegt. Wurde dieser nicht erreicht, so wurde nach maximal

50000 Zyklen das Training beendet. Abweichungen von diesem Protokoll sind bei der Schilderung der Ergebnisse für die einzelnen Netze gesondert aufgeführt.

Netz	*Eingabe-neuronen*	*Versteckte Neuronen*	*Ausgabe-neuronen*	*Muster im Trainingssatz*
AS-s100	650	50	20	2000
AS-s200	650	50	20	4000
AS-s400	650	50	20	8000
AS-s12out	650	50	2	2400
AS-sCAr	650	50	2	2400
AS-sEQ	650	50	2	2400
AS-sDN	650	50	2	2400
AS-sIL	650	50	2	2400
AS-sKR	650	50	2	2400
AS-b5	650	100	20	7990
AS-b10	650	100	20	7990
AS-b20	650	100	20	7990
AS-b12out	650	50	12	7990
AS-bCar	650	50	2	1990
AS-bDN	650	50	2	800
AS-bEQ	650	50	2	800
AS-bIL	650	50	2	800
AS-bKR	650	50	2	800
AS-g5	650	100	20	8000
AS-g10	650	100	20	8000
AS-g20	650	100	20	8000

Tabelle 28: Parameter der neuronalen Netze zur Bestimmung des Aminosäuretyps. Die Netze unterscheiden sich in der Art, in der die Muster erzeugt bzw. auf der Eingabeschicht dargestellt werden.

Da neuronale Netze zur Bearbeitung verschiedener Problemstellungen benutzt wurden, ist auch die Architektur der einzelnen Netze unterschiedlich. So unterscheiden sich die Netze in der Anzahl der Neuronen in den verschiedenen Schichten und in der Anzahl der benutzten Muster für das Training. In Tabelle 24 sind diese Werte für die Netze zur Aminosäuretypbestimmung zusammengefasst.

Für die Signalvorhersage wurden ebenfalls verschiedene Netze trainiert, die sich in der Art der Aminosäurekodierung unterschieden. Ein weiteres Unterscheidungskriterium war die Anwendung auf einzelne Aminosäuren oder alle der 20 verschiedenen Standardaminosäuren. Die Architektur dieser Netze ist in Tabelle 29 aufgezeigt.

Netz	*Eingabe-neuronen*	*Versteckte Neuronen*	*Ausgabe-neuronen*	*Muster im Trainingssatz*
SEQ-st	189	50	1	11284
SEQ-comp	54	15	1	11284
SEQ-bit	216	60	1	11284
SEQ-bit5h	216	5	1	11284
SEQ-bit15h	216	15	1	11284
SEQ-bit30h	216	30	1	11284
SEQ-st-sp*	189	25	1	67 – 816
SEQ-comp-sp*	54	10	1	67 – 816
SEQ-bit-sp*	216	30	1	67 – 816

Tabelle 29: Parameter für neuronale Netze zur Vorhersage der Lage des NH/Hα-Kreuzsignals. *Diese Netze waren auf einzelne Aminosäuren spezialisiert. Die genaue Anzahl der Trainingsmuster für diese Netze ist abhängig vom Typ der Aminosäure und in Tabelle 21 angegeben.

In einem weiteren Versuch wurden die Trainingsdaten in jeweils vier gleich große Sätze aufgeteilt und damit vier neuronale Netze trainiert. Damit sollte eine Möglichkeit geschaffen werden, genaue von ungenaueren Vorhersagen unterscheiden zu können. Waren die Vorhersagen der vier

Netze sehr ähnlich, so wurde angenommen, daß die berechneten chemischen Verschiebungen mit einem kleinerem Fehler behaftet waren als im Fall von stark voneinander abweichenden Vorhersagen. Die Architektur der hierfür verwendeten Netze ist in Tabelle 30 dargestellt.

Netz	*Eingabe-neuronen*	*Versteckte Neuronen*	*Ausgabe-neuronen*	*Muster im Trainingssatz*
SEQ-st-sp4*	189	50	2	217 – 1574
SEQ-comp-sp4*	54	20	2	217 – 1574
SEQ-bit-sp4*	216	40	2	217 – 1574
SEQ-st4	189	50	2	15210
SEQ-comp4	54	20	2	15210
SEQ-bit4	216	40	2	15210

Tabelle 30: **Parameter für neuronale Netze zur Vorhersage der Lage des NH/Hα-Kreuzsignals bei Aufteilung der Daten auf vier Netze. Angegeben ist die gesamte Anzahl an zur Verfügung stehenden Mustern. Für jedes Netz wurde ein Viertel dieser Datensätze zum Training verwendet.**
***Diese Netze waren auf einzelne Aminosäuren spezialisiert. Die genaue Anzahl der Trainingsmuster für diese Netze ist abhängig vom Typ der Aminosäure und in Tabelle 22 angegeben.**

7.4 Auswertung

Die Ergebnisse des neuronalen Netzes werden von der Software in Textdateien abgelegt. In diesen Dateien sind die Ausgabewerte als Zahlenwerte eingetragen.

Für die Erkennung des Aminosäuretyps war es nötig, daß Ausgabeneuron mit dem höchstem Wert zu ermitteln. Hierfür wurde das Skript *getaas.perl* geschrieben. Auch die besprochene Zusammenfassung ähnlicher Aminosäuren zu Gruppen und die Berechnung der Erkennungsrate über alle Muster wurden hier implementiert.

Zur Vorhersage der chemischen Verschiebung mußten die Ausgabewerte der neuronalen Netze wieder in die ppm-Skala umgerechnet

werden. Diese Umrechnung sowie die statistische Analyse der aufgetretenen Fehler erfolgte mit dem Skript *getvalues.perl*.

Die Zuordnung der vorhergesagten Kreuzsignale und die Validierung durch NOE-Signale erfolgte in dem eigenständigem Programm *noenaat.pl*. Dieses benötigt als Eingabe nur Listen mit Signalen aus NOESY- und TOCSY-Spektren, sowie die vorher getroffene Zuordnung zu Aminosäuren. Die benutzten neuronalen Netze und das Inkrementsystem wurden implementiert. Der Auswertealgorithmus, der vier verschiedene Vorhersagen berücksichtigt, wurde ebenso eingebaut wie der abschließende Vergleich verschiedener Methoden. Mit Hilfe dieses Programms konnten die erarbeiteten Methoden schnell getestet und miteinander verglichen werden.

7.5 Inkrementsystem

Das Inkrementsystem besteht im Kern aus den zwei Programmen *increment.pl* und *calcshift.pl*. Das erste analysiert die Datensätze der *BioMagResBank* und berechnet die benötigten Mittelwerte und Inkremente. Hierbei kann festgelegt werden, wie lang die Sequenzabschnitte sein sollen. Die Ergebnisse werden in Dateien ausgegeben. Das zweite Programm berechnet aus diesen Werten die chemischen Verschiebungen beliebiger Sequenzen unter Berücksichtigung der entsprechenden Anzahl an Inkrementen.

7.6 NMR Parameter

Die entwickelten Methoden wurden unter anderem mit im Arbeitskreis gemessenen Spektren getestet. Die Spektren wurden mit einem DRX500 NMR-Spektrometer (Bruker) aufgenommen und mit der Software XWinNMR prozessiert. Die Erstellung der Signallisten erfolgte mit dem Programm AURELIA und den dort zur Verfügung gestellten automatischen Funktionen zur Signalsuche. Die gefundenen Signale wurden überprüft und subjektiv gut erkennbare Peaks, die von der automatischen Suche nicht erfasst wurden, manuell hinzugefügt. Die Aufnahme- und

Prozessierungsparameter der verwendeten Spektren sind in Tabelle 31 aufgelistet.

Spektrum	*Temperatur*	*Lösungsmittel*	*pH-Wert*	*Apodisierungs-funktion in f1*	*Apodisierungs-funktion in f2*
j-v3-9	300 K	H_2O/D_2O	3.5	sine4 (TOCSY)	sine6 (TOCSY)
		9:1		sine2 (NOESY)	sine2 (NOESY)
j-v3-10	300 K	H_2O/D_2O	3.5	sine4 (TOCSY)	sine6 (TOCSY)
		9:1		sine2 (NOESY)	sine2 (NOESY)
Kin	300 K	H_2O/D_2O	3.5	sine6 (TOCSY)	sine6 (TOCSY)
		9:1		sine4 (NOESY)	sine4 (NOESY)

Tabelle 31: Aufnahme- und Prozessierungsparameter der verwendeten NMR-Spektren. Die Bezeichnung der angegebenen Apodisierungsfunktionen entspricht der Nomenklatur aus XWinNMR.

8 Literatur

[1] T. L. Blundell, L. N. Johnson, *Protein Crystallography*, New York, Academic Press, **1976**.

[2] K. C. Holmes, D. M. Blow, *The Use of X-Ray Diffraction in the Study of Protein and Nucleic Acid Structure*, New York, Wiley-Interscience, **1965**.

[3] L. P. McIntosh, F. W. Dahlquist, *Quart. Rev. of Biophys.* **1990**, *23*, 1-38.

[4] H. Senn, A. Eugster, G. Otting, F. Sutter, K. Wüthrich, *Eur. Biophys. J.* **1987**, *14*, 301-306.

[5] K. Wüthrich, *NMR of Proteins and Nucleic Acids*, New York, J. Wiley & Sons, **1986**.

[6] A. D. Kline, W. Braun, K. Wüthrich, *J. Mol. Biol.* **1986**, *189*, 377-382.

[7] K. Wüthrich, *Science* **1989**, *243*, 45-50.

[8] W. R. Croasmun, R. M. K. Carlson, *Two-Dimensional NMR Spectroscopy*, Weinheim, VCH Verlagsgesellschaft GmbH, **1994**.

[9] A. Kumar, R. R. Ernst, K. Wüthrich, *Biochem. Biophys. Res. Comm.* **1980**, *95*, 1-6.

[10] A. Kumar, G. Wagner, R. R. Ernst, L. Wüthrich, *Biochem. Biophys. Res. Comm.* **1980**, *96*, 1156-1163.

[11] H. Oschkinat, C. Griesinger, P. J. Kraulis, O. W. Sörensen, R. R. Ernst, A. M. Gronenborn, G. M. Clore, *Nature* **1988**, *332*, 374-376.

[12] L. Braunschweiler, R. R. Ernst, *J. Magn. Reson.* **1983**, *53*, 521-528.

[13] D. S. Wishart, B. D. Sykes, F. M. Richards, *J. Mol. Biol.* **1991**, *222*, 311-333.

[14] A. Pastore, V. Saudek, *J. Magn. Reson.* **1990**, *90*, 165-176.

[15] D. S. Wishart, B. D. Sykes, F. M. Richards, *Biochemistry* **1991**, *31*, 1647-1651.

[16] D. S. Wishart, B. D. Sykes, *Meth. Enzym.* **1994**, *239*, 363-392.

[17] J. Jeener, B. H. Meier, P. Bachmann, R. R. Ernst, *J. Chem. Soc.* **1979**, *71*, 4546-4553.

[18] D. Neuhaus, M. Williamson, *The Nuclear Overhauser Effect*, Weinheim, VCH Verlagsgesellschaft GmbH, **1989**.

[19] K. Wüthrich, G. Wider, G. Wagner, W. Braun, *J. Mol. Biol.* **1982**, *155*, 311-319.

[20] M. Billeter, W. Braun, K. Wüthrich, *J. Mol. Biol.* **1982**, *155*, 321-346.

[21] D. E. Rumelhart, J. L. McClelland, *Parallel Distributed Processing*, Cambridge, MIT Press, **1986**, Vol. 1.

[22] J. Dayhoff, *Neural Network Architectures*, New York, Van Nostrand Reinhold, **1997**.

[23] E. Rich, K. Knight, *Artificial Intelligence*, McGraw-Hill, **1991**.

[24] J. Gasteiger, J. Zupan, *Angew. Chem.* **1993**, *105*, 510-536.

[25] L. P. Zhang, L. M. Li, Z. Chi Z, *Int. J. of Speech Technology* **1998**, *2*, 241-248.

[26] C. Chiu, M. A. Shanblatt, *Int. J. of Neural Systems* **1995**, *6*, 79-89.

[27] S. Kurogi, *Biological Cybernetics* **1991**, *64*, 243-249.

[28] J. L. Elman, D. Zipser, *J. Acoustical Soc. Am.* **1988**, *83*, 1615-1626.

[29] S.W. Yu, *Asia-Pacific Financial Markets* **1999**, *6*, 341-354.

[30] P. R. Lajbcygier, J. T. Connor, *Int. J. of Neural Systems* **1997**, *8*, 457-471.

[31] J. A. Burns, G. M. Whitesides, *Chem. Rev.* **1993**, *93(8)*, 2583-2601.

[32] B. G. Sumpter, D. W. Noid, *Annu. Rev. Mater. Sci.* **1996**, *26*, 223-277.

[33] M. E. Munk, M. S. Madison, E. W. Robb, *Microchim. Acta* **1991**, *11*, 505-514.

[34] E. W. Robb, M. E. Munk, *Microchim. Acta* **1990,** *1*, 131-155.

[35] U. M. Weigel, R. Herges, *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 723-731.

[36] H. Schulz, M. Derrick, D. Stulik, *Anal. Chim. Acta* **1995**, *316*, 145-159.

[37] M. Novic, J. Zupan, *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 454-466.

[38] B. Meyer, T. Hansen, D. Nute, P. Albersheim, A. Darvill, W. York, J. Sellers, *Science* **1991**, *251*, 542-544.

[39] J. P. Radomski, H. v. Halbeek, B. Meyer, *Nat. Struct. Biol.* **1994**, *1*, 217-218.

[40] V. Kvasnicka, *J. Math. Chem.* **1991**, *6*, 63-76.

[41] D. L. Clouser, P. C. Jurs, *Anal. Chim. Acta* **1994**, *295*, 221-231.

[42] L. S. Anker, P. C. Jurs, *Anal. Chem.* **1992**, *64*, 1157-1164.

[43] S. A. Corne, J. Fisher, A. P. Johnson, W. R. Newell, *Anal. Chim. Acta* **1993**, *278*, 149-158.

[44] B. J. Hare, J. H. Prestegard, *J. Biomol. NMR* **1994**, *4*, 35-46.

[45] S. A. Corne, *Data Handl. Sci. Technol.* **1996**, *18*, 407-421.

[46] N. Quian, T. J. Sejnowski, *J. Mol. Biol.* **1988**, *202*, 865-884.

[47] D. G. Kneller, F. E. Cohen, R. Langridge, *J. Mol. Biol.* **1990**, *214*, 171-182.

[48] B. Rost, C. Sander, *Proc. Natl. Acad. Sci. USA* **1993**, *90*, 7558-7562.

[49] J. E. Hansen, O. Lund, J. O. Nielsen, S. Brunak, J.-E. S. Hansen, *Proteins: Struct. Funct. & Genet.* **1996**, *25*, 1-11.

[50] T. N. Petersen, C. Lundegaard, M. Nielsen, H. Bohr, J. Bohr, S. Brunak, G. P. Gippert, O. Lund, *Proteins: Struct. Funct. & Genet.* **2000**, *41*, 17-20.

[51] M. Tusar, J. Zupan, J. Gasteiger, *J. Chim. Phys.* **1992**, *89*, 1517-1529.

[52] J. Zupan, M. Novic, J. Gasteiger, *Chem. Intell. Lab. Syst.* **1995**, *27(2)*, 15-17.

[53] M. Tusar, F. X. Rius, J. Zupan, *Mitteilungsblatt der GDCh-Fachgruppe "Chemie-Information-Computer"*, **1991**, *19*, 72-84.

[54] I. V. Tetko, V. V. Kovalishyn, D. J. Livingstone, *J. Med. Chem.* **2001**, *44*, 2411-2420.

[55] F. R. Burden, M. G. Ford, D. C. Whitley, D. A. Winkler, *J. Chem. Inf. Comp. Sci.* **2000**, 40, 1423-30.

[56] T. Aoyama, Y. Suzuki, H. Ichikawa, *J. Med. Chem.* **1990**, *33*, 905-908.

[57] T. Aoyama, Y. Suzuki, H. Ichikawa, *J. Med. Chem.* **1990**, *33*, 2583-2590.

[58] Q. Liu, S. Hirono, I. Moriguchi, *Quant. Struct.-Act. Relat.* **1992**, *11*, 318-324.

[59] M. Wiese, *Quant. Struct.-Act. Relat.* **1991**, *10*, 369-371.

[60] B. Adler, K. Ammon, S. Dobers, M. Winterstein, H. Ziesmer, *Chem. Tech.* **1992**, *44(11-12)*, 363-367.

[61] A. Breindl, B. Beck, T. Clark, R. C. Glen, *J. Mol. Model.* **1997**, *3*, 142-155.

[62] A. F. Duprat, T. Huynh, G. Dreyfus, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 586-594.

[63] J. C. Westermann, *Diplomarbeit*, Universität Hamburg **2001**.

[64] L. M. Egolf, M. D. Wessel, P. C. Jurs, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 947-956.

[65] M. D. Wessel, P. C. Jurs, *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 68-76.

[66] H. Kränz, V. Vill, B. Meyer, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 1173-1177.

[67] R. Schröder, H. Kränz, V. Vill, B. Meyer, *J. Chem. Soc. Perk. Trans.* **1996**, *2*, 1-4.

[68] D. W. Elrod, G. M. Maggiora, R. G. Trenary, *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 477-484.

[69] D. W. Elrod, G. M. Maggiora, R. G. Trenary, *Tetrahedron Comput. Methodol.* **1990**, *3*, 163-174.

[70] T. Kohonen, *Biol. Cybern.* **1982**, *43*, 59-69.

[71] T. Kohonen, *Self-Organization and Associative Memory*, Berlin, Springer Verlag, **1989**.

[72] J. Gasteiger, X. Li, A. Uschold, *J. Mol. Graph.* **1994**, *12(2)*, 90-107.

[73] S. Anzali, G. Barnickel, M. Krug, J. Sadowski, M. Wagener, J. Gasteiger, J. Polanski, *J. Comp.-Aid. Mol. Des.* **1996**, *10(6)*, 521-534.

[74] D. E. Rumelhart, G. E. Hilton, R. J. Williams, *Nature* **1986**, *333*, 533-536.

[75] W. P. Jones, J. Hoskins, *Byte* **1987 (Oct.)**, 155-162.

[76] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Washington, Spartan Books, **1962**.

[77] M. Minsky, S. Papert, *Perceptrons*, Cambridge, MIT Press, **1996**.

[78] B. R. Seavy, E. A. Farr, W. M. Westler, J. L. Markley, *J. Biomol. NMR* **1991**, *1*, 217-236.

[79] http://www.bmrb.wisc.edu/pages/

[80] S. L. Dixon, R. T. Koehler, *J. Med. Chem.* **1999,** *42, 2887-2900.*

[81] P. Willett, V. A. Winterman, *Quant. Struct.-Act. Relat.* **1986**, *5, 18-25*.

[82] T. L. South, P. R. Blake, D. R. Hare, M. F. Sumers, *Biochemistry* **1991**, *30, 6342-6349*.

[83] H. Tamaoki, Y. Kobayashi, S. Nishimura, T. Ohkubo, Y. Kyogoku, K. Nakajima, S. Kumagaye, *Protein Eng.* **1991**, *4 (5), 509-518*.

[84] K. L. Constantine, V. Ramesh, L. Banyai, M. Trexler, L. Patthy, M. Llinas, *Biochemistry* **1991**, *30, 1663-1672*.

[85] Y. Chen, S. M. Pitzenberger, V. M. Garsky, P. K. Lumma, G. Sanyal, J. Baum, *Biochemistry* **1991**, *30, 11625-11636*.

[86] J. P. Simorre, A. Caille, D. Marion, M. Ptak, *Biochemistry* **1991**, *30, 11600-11608*.

[87] X. Gao, W. Buckhart, *Biochemistry* **1991**, *30*, *7730-7739*.

[88] A. Aumelas, L. Chiche, E. Mahe, D. Le-Nguyen, P. Sizun, P. Berthault, B. Perly, *Neurochem. Int.* **1991**, *18 (4)*, *471-475*.

[89] J. G. Omichinski, G. M. Clore, E. Appella, K. Sakaguchi, A. M. Gronenborn, *Biochemistry* **1990**, *29*, *9324-9334*.

[90] W. Klaus, T. Dieckmann, V. Wray, D. Schomburg, E. Wingender, H. Mayer, *Biochemistry* **1991**, *30*, *6936-6942*.

[91] Y. Feng, A. J. Wand, S. G. Sligar, *Biochemistry* **1991**, *30*, *7711-7717*.

[92] A. Padilla, A. Cave, J. Parello, *J. Mol. Biol.* **1988**, *204*, *995-1017*.

[93] S. C. Brown, L. Mueller, P. W. Jeffs, *Biochemistry* **1989**, *28*, *593-599*.

[94] P. N. Barlow, M. Baron, D. G. Norman, A. J. Day, A. Willis, R. B. Sim, I. D. Campbell, *Biochemistry* **1991**, *30*, *997-1004*.

[95] M. H. Werner, D. Wemmer, *Biochemistry* **1991**, *30*, *3356-3364*.

[96] W. J. Fairbrother, J. Cavanagh, H. J. Dyson, A. G. Palmer, S. L. Sutrina, J. Reizer, M. H. Saier, P. E. Wright, *Biochemistry* **1991**, *30*, *6896-6907*.

[97] M. Kochoyan, T. F. Havel, D. T. Nguyen, C. E. Dahl, H. T. Keutmann, M. A. Weiss, *Biochemistry* **1991**, *30*, *3371-3386*.

[98] P. Strop, G. Wider, K. Wüthrich, *J. Mol. Biol.* **1983**, *166*, *641-667*.

[99] P. K. Hammen, E. B. Waygood, R. E. Klevit, *Biochemistry* **1991**, *30*, *11842-11850*.

[100] B. J. Stockman, A. M. Krezel, J. L. Markley, K. G. Leonhardt, N. A. Straus, *Biochemistry* **1990**, *29*, *9600-9609*.

[101] J.-H. Lee, K. Bang, J.-W. Jung, I.-A. Ahn, S. Ro, W. Lee, *Bull. Kor. Chem. Soc.* **1999**, *20*, *301-306*.

[102] P. L. Yeagle, C. Danis, G. Choi, J. L. Alderfer, A. D. Albert, *Molecular Vision* **2000**, *6*, *125-131*.

[103] J. Tost, *Diplomarbeit*, Universität Hamburg **1999**.

[104] C. Seeberger, E. Mandelkow, B. Meyer, *Biochemistry* **2000**, *39*, *12558-12567*.

[105] Y. Fezoui, P. J. Connolly, J. J. Osterhout, *Protein Science* **1997**, *6*, *1869-1877*.

[106] Kränz, Helge, *Dissertation*, Universität Hamburg **1997**.

[107] http://www.perl.com

[108] http://www.cpan.org

[109] http://www.cpan.org/modules/by-module/Text/

[110] http://www.cpan.org/modules/by-module/Statistics/

Inhalt der beim Fachbereich Chemie hinterlegten CD

Verzeichnis „Mustererzeugung“

- cdb3.1.perl: Erzeugt aus Textdateien der BioMagResBank Musterdateien für neuronale Netze zur Signalvorhersage
- nmrdata.pm: Modul mit Klassen zur Speicherung spektroskopischer Daten
- pdbdata.pm: Modul mit Klassen zur Speicherung von Strukturdaten aus pdb-Dateien
- pattern.pm: Modul mit Klassen zur Mustererzeugung
- codetable.pm: Modul mit verschiedenen Kodierungsmöglichkeiten für Aminosäuren
- gentracepat.perl: Programm zur statistischen Erzeugung von Mustern für die Aminosäurebestimmung
- bmrtotrace_fuzzy.perl: Erzeugt Muster mit breiter Kodierung zur Aminosäuretypbestimmung
- aadata.pm: Modul mit Klassen zur statistischen Verteilung von chemischen Verschiebungen
- spinsys.pm: Modul mit Klassen zur Speicherung einzelner TOCSY-Spuren

Verzeichnis „Inkrementsystem“

- increment.pl: Programm zur Erzeugung der Inkremente aus Daten der BioMagResBank
- calcshift.pl: Programm zur Ermittlung chemischer Verschiebungen unter Verwendung von Inkrementen

Verzeichnis „Auswertung“

- getvalues.perl: Berechnet Fehlerverteilungen und andere statistische Größen aus den Ausgabedateien des neuronalen Netzes

- getaas.perl: Bestimmt die vorhergesagte Aminosäure aus Ausgabedateien des neuronalen Netzes
- noenaat.pl: Programm zur automatischen Zuordnung von vorhergesagten Verschiebungen zu Kreuzsignalen unter Berücksichtigung von NOE-Daten

Verzeichnis „nn“

Quellcode des neuronalen Netzes.

Verzeichnis „nndoc“

Dokumentation des neuronalen Netzes.

Mein Dank gilt:

- meiner Frau Birgit und meinen Töchtern Lea und Jana, ohne deren Unterstützung und Motivation diese Arbeit nicht entstanden wäre
- Helge Kränz, der vor allem in der Anfangsphase eine unschätzbare Hilfe darstellte
- Jörg Dojahn, der stets Ruhe und Gelassenheit ausstrahlte und trotz unterschiedlicher Themenstellung ähnliche Probleme zu bewältigen hatte wie ich auch
- Oliver Schuster, mit dem man immer entspannt diskutieren konnte
- Moriz Mayer, dem Erfinder des „Eisgolfens“
- Florian Ende, der konsequent an seinen Gewohnheiten (Geldautomat) festhielt und dem Tag somit eine feste Struktur gab
- Heiko Möller, der das einzige richtige Komma in der Arbeit gefunden hat
- Axel Neffe, der die restlichen richtigen Kommas gesetzt hat
- Boris Kröplien und Jan Westermann, für das flachsimpeln über die besten goldenen Gegenstände
- Attila Coksezen, der endlich ein Gegner und nicht nur ein Opfer war
- Birgit Claasen und Robert Meinecke, die mir den NOE und andere Mysterien der NMR-Spektroskopie nahegebracht haben
- Jutta Tost, für die schönen Spektren
- dem restlichen AK Meyer, in dem immer eine sehr nette Arbeitsatmosphäre herrschte
- der Deutschen Forschungsgemeinschaft, für die Finanzierung
- den Mitgliedern des Graduiertenkollegs 464, für die Erweiterung des Horizontes