Georg-August-Universität
Göttingen
Zentrum für Informatik

ISSN 1612-6793
Nummer ZFI-BM-2005-38

Masterarbeit

im Studiengang „Angewandte Informatik“

Maschinelles Lernen zur Vorhersage von Protein-Protein-Interaktionen

Nico Pfeifer

am Institut für
Mikrobiologie und Genetik

Bachelor- und Masterarbeiten
des Zentrums für Informatik
an der Georg-August-Universität Göttingen

31. Oktober 2005

Georg-August-Universität Göttingen
Zentrum für Informatik

Lotzestraße 16-18
37083 Göttingen
Germany

Tel. +49 (5 51) 39-1 44 02
Fax +49 (5 51) 39-1 44 03
Email office@informatik.uni-goettingen.de
WWW www.informatik.uni-goettingen.de

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Göttingen, den 31. Oktober 2005

Masterarbeit

Maschinelles Lernen zur Vorhersage von Protein-Protein-Interaktionen

Nico Pfeifer

31. Oktober 2005

Betreut durch Dr. Peter Meinicke
Abteilung Bioinformatik
Institut für Mikrobiologie und Genetik
Georg-August-Universität Göttingen

Inhaltsverzeichnis

1. Einleitung

Nahezu jeder biologische Prozess in einer Zelle wird beeinflusst von Proteinen.
Sie sind zum Beispiel beteiligt an der Regulation metabolischer Netzwerke, der DNA Replikation und der Proteinsynthese. Die Fähigkeit durch maschinelle Verfahren vorhersagen zu können, welche Proteine miteinander interagieren, würde zu einem besseren Verständnis dieser Prozesse führen und die direkte Manipulation der Selbigen erleichtern.

Die Fähigkeit eines Proteins, ein anderes Protein zu binden, wird durch seine dreidimensionale Struktur bestimmt [JON96], was auch in dieser Arbeit in Kapitel 2 erläutert wird. Da allerdings die Strukturvorhersage von Proteinen sehr schwierig und noch nicht zufriedenstellend gelöst ist, existieren verschiedene Verfahren, die andere Informationen nutzen, um Protein-Protein-Interaktionen vorherzusagen. Die Bekanntesten dieser Verfahren werden in Kapitel 3 vorgestellt. Die Idee, dass interagierende Proteine co-evolvieren und dieser Grad an Co-Evolution gemessen werden kann, ist die Grundlage des *Mirrortree*-Verfahrens, das in Kapitel 3.4 vorgestellt wird. Dieses anerkannte Verfahren dient in dieser Arbeit als Ausgangspunkt für die Semi *Supervised Kernel Regression* (*SSKR*), eine Erweiterung der *Unsupervised Kernel Regression* (*UKR*) [MEI05], die im Rahmen dieser Masterarbeit für die Vorhersage von Protein-Protein-Interaktionen realisiert wurde. Außerdem dient das *Mirrortree*-Verfahren als Referenz um die Performanz der *SSKR* zu messen.
Hierzu werden zuerst drei verschiedene Datensätze erstellt. Die Generierung dieser Datensätze sowie die Datenbanken, aus denen die Informationen extrahiert werden, sind in Kapitel 5 dargestellt. Die Ergebnisse der Auswertung des Performanzvergleichs (Kapitel 6.4) zeigen, dass die *UKR* und insbesondere die *SSKR* für die Vorhersage von Protein-Protein-Interaktionen besser geeignet sind als das *Mirrortree*-Verfahren.
Eine biologische Anwendung der Methoden wäre beispielsweise, für ein bestimmtes Protein potenzielle Interaktionspartner vorherzusagen. Für diese Kandidaten könnte dann die Interaktion im Labor überprüft werden. Dadurch könnten die aufwendigen Screenings für eine viel größere Zahl von potenziellen Kandidaten vermieden werden. Dies würde zu einer Kosten- und Zeitersparnis führen.

Da das Thema der Arbeit das maschinelle Lernen zur Vorhersage von Protein-Protein-Interaktionen ist, sollen hier kurz einige Begriffe des maschinellen Lernens erläutert werden. Eine tiefergehende Einführung ist zum Beispiel in [TIB01] nachzulesen.

Maschinelle Lernverfahren lassen sich in drei grundsätzliche Kategorien einteilen[1]. Die erste Kategorie ist die des **überwachten Lernens**. Hierbei ist eine Stichprobe der Daten gegeben, die oft mit $\{ (x_1, y_1), (x_2, y_2), ..., (x_n, y_n) \}$ dargestellt wird, wobei die x_i oft als Eingabe oder Input bezeichnet werden und die y_i als Label. Sowohl die x_i als auch die y_i können mehrere Dimensionen enthalten. Die Aufgabe beim maschinellen Lernen ist es, durch die Stichprobe eine spezielle Funktion zu lernen. Diese Funktion ermöglicht es, für neue Beispiele, für die die Labels unbekannt sind, die aber der gleichen Verteilung zugrunde liegen, das „echte" Label so gut wie möglich vorherzusagen. Bei den Labels werden drei unterschiedliche Fälle unterschieden. Sind die $y_i \in \{ -1, 1 \}$, so wird das Lernen als *binäre Klassifikation* bezeichnet. Existieren mehrere diskrete Werte oder Klassen für die Labels ($y_i \in \{ 1, 2, 3, ..., m \}$), so wird dies als *Multiklassen Klassifikation* bezeichnet. *Regression* liegt vor, wenn für die Labels reelle Zahlen erlaubt sind. In diesem Fall wird versucht, die Regressionsfunktion $f(x) = E(Y \mid X = x)$ so gut wie möglich zu approximieren.
Die zweite Kategorie ist die des **semi-überwachten Lernens**. Hierbei sind nach [SHA04] die Labels der Stichprobe nur teilweise bekannt und werden zum Beispiel beim transduktiven Lernen verwendet, um die Labels der restlichen ungelabelten Daten vorherzusagen.
Die dritte Kategorie ist die des **unüberwachten Lernens**, bei der die Stichprobe überhaupt keine Labels enthält. Die Aufgabe ist hier, die Daten in Gruppen aufzuteilen, die bestimmte Eigenschaften der Verteilung repräsentieren.

Die in dieser Arbeit vorgestellte *UKR*-Methode (siehe 4.2) ist dem unüberwachten Lernen zuzuschreiben, wobei die *SSKR* (siehe 4.3) dem **semi-überwachten Lernen** zuzurechnen ist.

[1] Die vierte Kategorie des Verstärkungslernens (Reinforcement-Learning) wird in dieser Arbeit nicht behandelt

2. Biologische Grundlagen

2.1. Proteine

Definition: Ein Protein ist ein lineares Polymer, das aus monomeren Untereinheiten, den Aminosäuren, zusammengesetzt ist.

Proteine sämtlicher Lebewesen, vom Bakterium bis zum Homo sapiens, werden aus dem selben Satz von 20 Aminosäuren aufgebaut. Die hohe Zahl an verschiedenen Bausteinen ermöglicht es, sehr unterschiedliche Proteine herzustellen. Diese Vielseitigkeit führt dazu, dass sie in fast allen biologischen Prozessen entscheidende Funktionen einnehmen. Hier sei erwähnt, dass Proteine unter anderem als Katalysatoren wirken, Bewegung ermöglichen, Immunität verleihen, Nervenimpulse übermitteln und Wachstum sowie Differenzierung kontrollieren.

Da Proteine eine Vielzahl an funktionalen Gruppen enthalten können wie zum Beispiel Alkohole, Carbonsäuren, Carboxamide, Thiole, Thioeter und eine Reihe von alkalischen Gruppen, ermöglicht dies ein breites Spektrum an Proteinfunktionen. Die chemische Reaktivität dieser Gruppen ist zum Beispiel von zentraler Bedeutung für die Funktion von speziellen Proteinen, den Enzymen, die spezifische Reaktionen in biologischen Systemen katalysieren.

Alle Aminosäuren besitzen ein zentrales C Atom (C_α), eine Aminogruppe (NH_3^+), ein Wasserstoffatom (H), eine Carboxylgruppe (COO^-) und einen für die Aminosäure charakteristischen Rest (R), wie in Fig. 2.1 zu sehen ist.

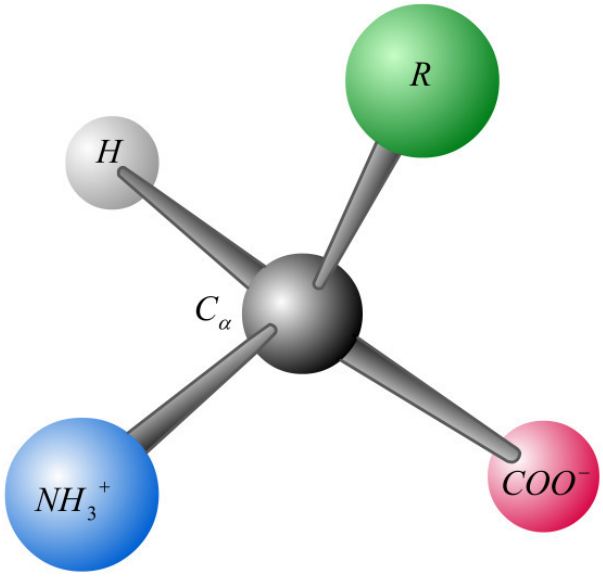


Fig. 2.1 Eine Aminosäure

Über Peptidbindungen werden die Aminosäuren verknüpft, indem die α-Carboxylgruppe der einen Aminosäure mit der α-Aminogruppe der anderen Aminosäure verbunden wird. Bei dieser Reaktion wird Wasser frei. Eine solche Verknüpfung ist in Fig. 2.2 zu sehen.

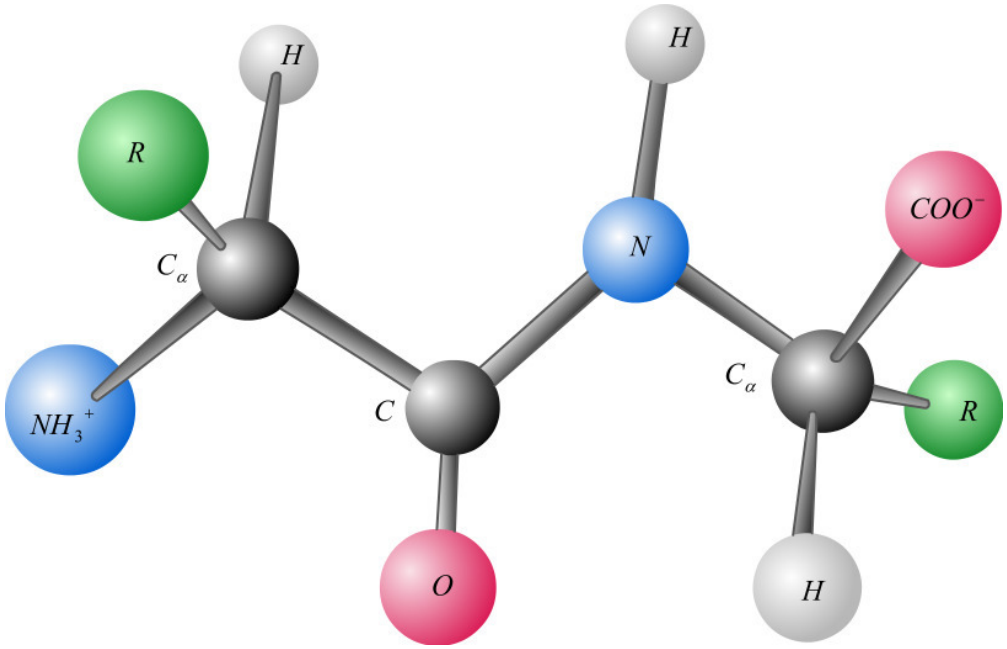


Fig. 2.2 Zwei Aminosäuren sind über eine Peptidbindung verknüpft

2.2. Proteinbindungen

2.2.1. Kovalente Bindungen

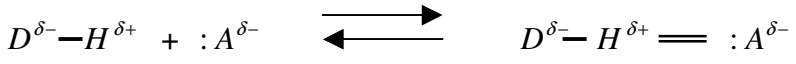
Unter einer kovalenten Bindung versteht man eine Bindung, bei der zwei Atome sich ein oder mehrere Elektronenpaare teilen. Dies ist die stärkste Art von Bindung zwischen zwei Atomen. Zum Beispiel hat die typische kovalente Bindung zwischen zwei Kohlenstoffatomen eine Bindungsenergie von 356 $kJ\ mol^{-1}$. Die Peptidbindung aus 2.1 ist ein Beispiel für eine kovalente Bindung.

2.2.2. Ionenbindung

Die elektrostatischen Anziehungskräfte, die dazu führen, dass gegensinnig geladene Ionen zusammengehalten werden, bezeichnet man als Ionenbindung. Positiv geladene Ionen werden als Kationen bezeichnet. Sie entstehen dadurch, dass Atome mit einer geringen Anzahl an Valenzelektronen (Metalle) diese unter gewissen Umständen abgeben. Negativ geladene Ionen bezeichnet man als Anionen. Diese entstehen aus Atomen, denen an der Edelgaskonfiguration der Valenzelektronen ein oder zwei Elektronen fehlen (Nichtmetalle), da sie eine Tendenz dazu haben, die fehlenden Elektronen aufzunehmen. Ionenbindungen sind deutlich schwächer als kovalente Bindungen. Zum Beispiel hat die elektrostatische Wechselwirkung zwischen zwei Atomen mit entgegengesetzten einfachen Ladungen, die in Wasser 0,3 nm voneinander entfernt sind, einen Energiegehalt von 5,9 $kJ\ mol^{-1}$.

2.2.3. Wasserstoffbrückenbindungen

Wasserstoffatome, die an ein Atom kovalent gebunden sind, können unter gewissen Umständen eine schwächere zweite Verbindung, die auch als Wasserstoffbrückenbindung bezeichnet wird, mit einem anderen Atom eingehen. Dies ist dann der Fall, wenn die beiden Atome negative Partialladungen besitzen. Das kovalent gebundene Atom wird dann als Donor (D) bezeichnet und das andere Atom, zu dem die Wasserstoffbrückenbindung besteht, als Akzeptor (A). In biologischen Systemen sind Donoren und Akzeptoren in der Regel Stickstoff- oder Sauerstoffatome.



Die dicke Linie steht für eine kovalente Bindung. Die beiden dünneren Linien für eine Wasserstoffbrückenbindung. Die beiden Punkte links von dem Akzeptor sollen anzeigen, dass der Akzeptor ein zusätzliches Elektronenpaar aufgenommen hat. Wie Ionenbindungen sind auch Wasserstoffbrückenbindungen deutlich schwächer als

kovalente Bindungen. Der Energiegehalt einer solchen Bindung beträgt zwischen vier und 13 $kJ\ mol^{-1}$.

2.2.4. Van-der-Waals-Wechselwirkungen

Da sich die Verteilung der Elektronenladung rund um einen Atomkern im Laufe der Zeit ändert, bedeutet dies, dass die Ladungsverteilung zu keinem Zeitpunkt genau symmetrisch ist. Die Asymmetrie der Elektronenladung führt bei Nachbaratomen durch elektrostatische Wechselwirkungen zu einer komplementären, ebenfalls asymmetrischen, Elektronenverteilung. Dadurch entstehen Anziehungskräfte zwischen den beiden Atomen. Je näher sich die beiden Atome zueinander befinden, umso größer wird diese Anziehungskraft. Kommen sich beide Atome zu nahe, treten allerdings wieder andere Effekte ein, die zu einer Abstoßung der beiden Atome führen. Dies geschieht dadurch, dass sich die Elektronenwolken der beiden Atome dann überlappen. Der Abstand von dem Punkt, an dem die Anziehungskraft der Atome durch die Van-der-Waals-Kräfte am größten ist und noch keine Abstoßung durch andere Effekte auftritt, wird als Van-der-Waals-Kontaktdistanz bezeichnet.
Die Energie einer Van-der-Waals-Wechselwirkung beträgt zwischen zwei und vier $kJ\ mol^{-1}$. Sie ist daher geringer als kovalente Bindungen, Ionenbindungen und Wasserstoffbrückenbindungen.

2.3. Proteinstruktur

Es gibt vier verschiedene Abstraktionsebenen, auf denen Proteine beschrieben werden können. Die Abfolge der Aminosäuren eines einzelnen Proteins wird als *Primärstruktur* bezeichnet. Die verschiedenen möglichen Aminosäuren mit ihren Abkürzungen sind in Tab. 2.1 zu sehen.

Aminosäure	Abkürzung	Buchstabensymbol
Alanin	Ala	A
Arginin	Arg	R

Asparagin	Asn	N
Asparaginsäure	Asp	D
Cystein	Cys	C
Glutamin	Gln	Q
Glutaminsäure	Glu	E
Glycin	Gly	G
Histidin	His	H
Isoleucin	Ile	I
Leucin	Leu	L
Lysin	Lys	K
Methionin	Met	M
Phenylalanin	Phe	F
Prolin	Pro	P
Serin	Ser	S
Threonin	Thr	T
Tryptophan	Trp	W
Tyrosin	Tyr	Y
Valin	Val	V

Tab 2.1 Aminosäuren und ihre Kurzschreibweisen

Regelmäßige Strukturen, zu denen sich Polypeptidketten falten können, werden als Sekundärstrukturelemente bezeichnet. Die zwei wichtigsten unter ihnen sind α-Helices und β-Faltblätter. Sie sind in Fig. 2.3 und Fig. 2.4 zu sehen. Die Abfolge aller Sekundärstrukturelemente einer Polypeptidkette wird als *Sekundärstruktur* bezeichnet.

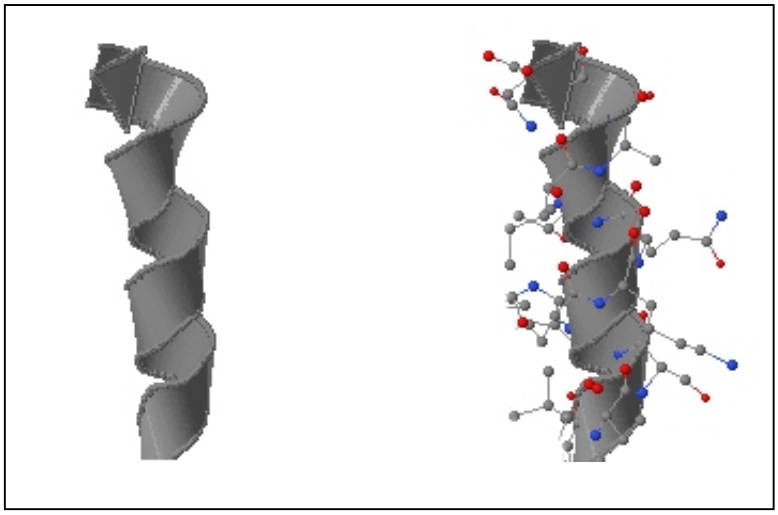


Fig 2.3 Sekundärstrukturelement: α-Helix

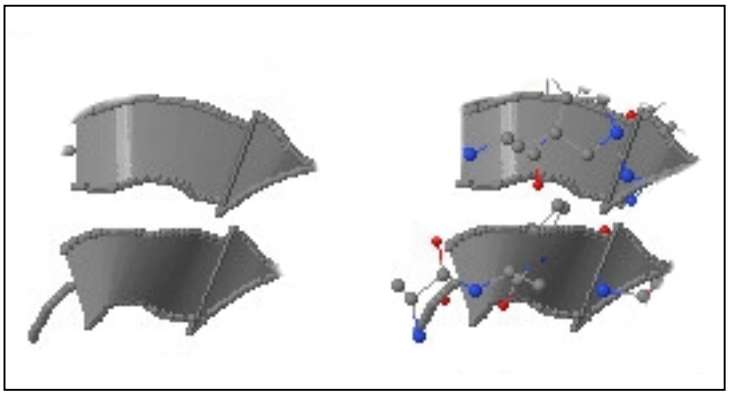


Fig 2.4 Sekundärstrukturelement: β-Faltblatt

Auf der nächsten Abstraktionsebene wird die Gesamtanordnung der Polypeptidkette, die als *Tertiärstruktur* bezeichnet wird, betrachtet. Die Tertiärstruktur eines Proteins ist in Fig. 2.5 zu sehen.

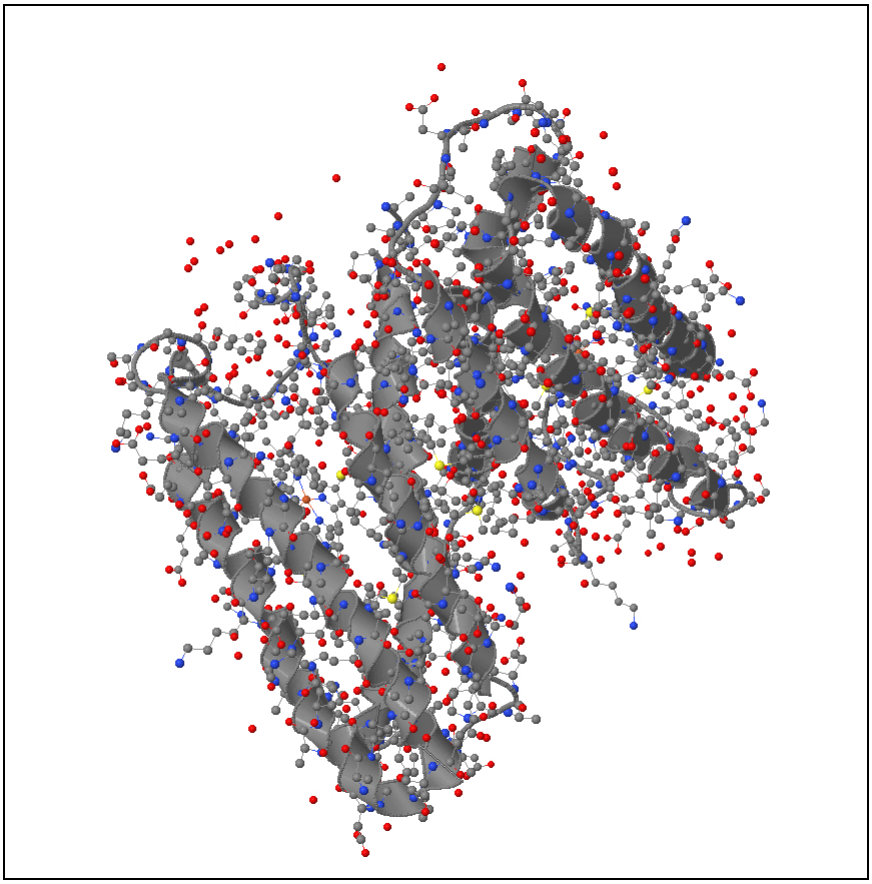


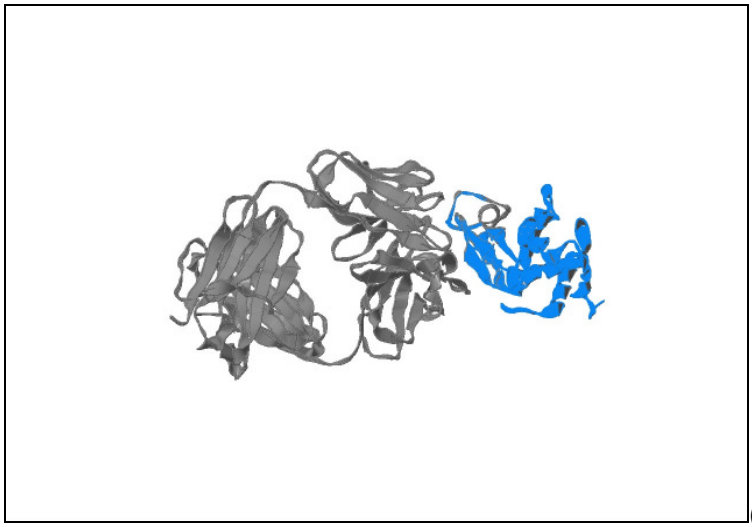
Fig. 2.5 Tertiärstruktur: Xanthin-Guanin Phosphoribosyltransferase

Proteine, die mehrere Polypeptidketten besitzen, können in Bezug auf die Anordnung der einzelnen Polypeptidketten, die dann als Untereinheiten bezeichnet werden,

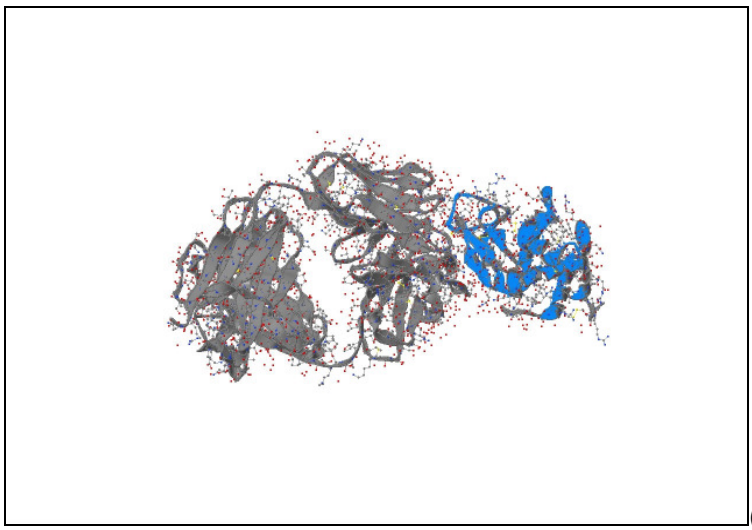
beschrieben werden. Die *Quartärstruktur* beschreibt die Anordnung dieser Untereinheiten und die Art ihrer Wechselwirkungen untereinander.

2.4. Grundlagen von Protein-Protein-Interaktionen

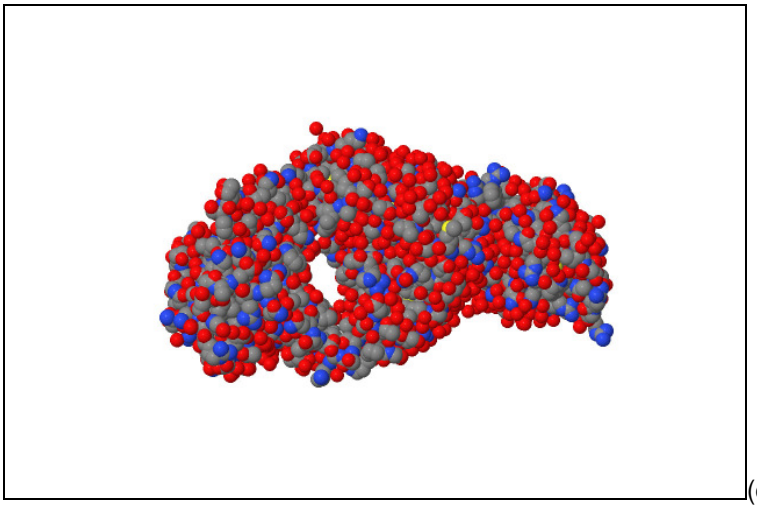
Die biologischen Eigenschaften eines Proteins werden von der Fähigkeit zu physikalischem Kontakt mit anderen Molekülen bestimmt. Zum Beispiel binden Antikörper an die Oberflächenproteine von Bakterien, um sie für die Zerstörung zu markieren. Die Stärke dieser Bindung variiert von Protein zu Protein, aber sie ist immer sehr spezifisch. Das bedeutet, dass jedes Protein nur eine geringe Anzahl anderer Moleküle binden kann. Die Substanz, die von dem Protein gebunden wird, wird als Ligand bezeichnet.
Die Fähigkeit eines Proteins, sehr selektiv und mit hoher Affinität einen Liganden binden zu können, ist abhängig von einer Reihe von nichtkovalenten Bindungen. Diese Bindungen sind Ionenbindungen (siehe 2.2.1), Wasserstoffbrückenbindungen (siehe 2.2.2) und Van-der-Waals-Kräfte (siehe 2.2.3). Da jede einzelne Bindung sehr schwach ist, ist es für eine stabile Verbindung zwischen Protein und Ligand erforderlich, dass sehr viele dieser Bindungen geknüpft werden. Dies erfordert, dass die Form der betreffenden Interaktionspartner sehr ähnlich ist. Sie passen zusammen wie Schlüssel und Schloss.
Das Prinzip kann anhand folgender Grafiken veranschaulicht werden: In Fig. 2.4 (a) sieht man zwei Proteine. Das graue Protein ist ein Antikörper mit leichter und schwerer Kette, der ein zweites Protein – hier blau dargestellt – bindet, das Lysozym genannt wird. Fig. 2.4 (b) und Fig. 2.4 (c) zeigen die gleichen Proteine in detaillierteren Darstellungen. Besonders in Fig. 2.4 (c) kann man erkennen, dass die Form der leichten und schweren Kette des Antikörpers perfekt in die Form des Lysozyms passt.



(a)



(b)



(c)

Fig. 2.4 Antikörper-Protein-Komplex: HYHEL5-Lysozym (a) Darstellung ohne Atome und ohne Bindungen: Das Lysozym ist in hellblau dargestellt, der Antikörper mit leichter und schwerer Kette in grau. (b) Darstellung mit Atomen und Bindungen: Das Lysozym ist in hellblau dargestellt, der Antikörper mit leichter und schwerer Kette in grau. (c) Darstellung mit Kugeln, die die Van-der-Waals-Kontaktdistanz (siehe 2.2.4) symbolisieren

2.5. Homologe Proteine

Stammen zwei Proteine von einem gemeinsamen Vorfahren ab, so werden sie als *homologe* Proteine bezeichnet. Sie können in zwei Klassen aufgeteilt werden. Die erste ist die Klasse der *Paralogen*. Dies sind homologe Proteine, die innerhalb einer Art auftreten. Sie besitzen oft unterschiedliche biochemische Funktionen. Die zweite Klasse ist die Klasse der *Orthologen*. Dies sind homologe Proteine in verschiedenen Organismen. Ihre biochemische Funktion ist oft sehr ähnlich oder sogar gleich.

2.6. Yeast Two-Hybrid Verfahren zum Nachweis von Protein-Protein-Interaktionen

Die Idee des Yeast Two-Hybrid Verfahrens [ALB02] ist es, die Interaktion zweier Proteine mit Hilfe eines zweiteiligen Transkriptionsaktivators nachzuweisen. Dies ist in Fig. 2.5 schematisch dargestellt. Zudem wird ein Reportergen verwendet. Reportergene sind Gene, die dazu verwendet werden, um bestimmte Veränderungen der Transkription nachzuweisen. Ein Beispiel für ein Reportergen ist das Chloramphenicoltransacetylasegen (cat). Das Proteinprodukt dieses Gens vermittelt eine Resistenz gegen das Antibiotikum Chloramphenicol. Können Zellen auf einem Nährmedium wachsen, das Chloramphenicol enthält, so ist dies ein Indiz dafür, dass das cat-Gen transkribiert wurde.
Der Transkriptionsaktivator benötigt, um die Transkription des Reportergens anzuschalten, zwei Teile. Der eine Teil besteht aus einer DNA Bindedomäne und der andere Teil ist eine Transkriptions-Aktivierungsdomäne. Mit Hilfe rekombinanter DNA Technologie wird die DNA, die ein bestimmtes Zielprotein kodiert, mit der DNA der DNA Bindedomäne verknüpft. Wird dieses Konstrukt in eine Hefezelle gebracht, so produziert die Zelle ein Fusionsprotein, das aus der DNA Bindedomäne und dem Zielprotein besteht. Das entstandene Protein bindet an die regulatorische Region eines Reportergens.
Für alle Proteine, für die Interaktionen mit dem Zielprotein nachgewiesen werden sollen, wird folgendermaßen vorgegangen. Die DNA, die die Aktivierunsdomäne kodiert, wird

mit der DNA des zu untersuchenden Bindungspartners fusioniert. Dann wird das Konstrukt in eine separate Hefezelle eingeführt und die Zelle produziert das Fusionsprotein. Bindet das Protein an das Zielprotein, das schon über die DNA Bindedomäne mit der regulatorischen Region des Reportergens verbunden ist, so aktiviert die Transkriptions-Aktivierungsdomäne die Transkription des Reportergens. Für jede Hefezelle wird folglich genau ein potenzieller Bindungspartner getestet. Wird in einer Hefezelle die Transkription des Reportergens festgestellt, so ist dies ein Indiz dafür, dass das Zielprotein mit dem in dieser Zelle getesteten Bindunspartner interagiert, da vermutlich durch die Interaktion der beiden Proteine die Transkription des Reportergens aktiviert wurde.

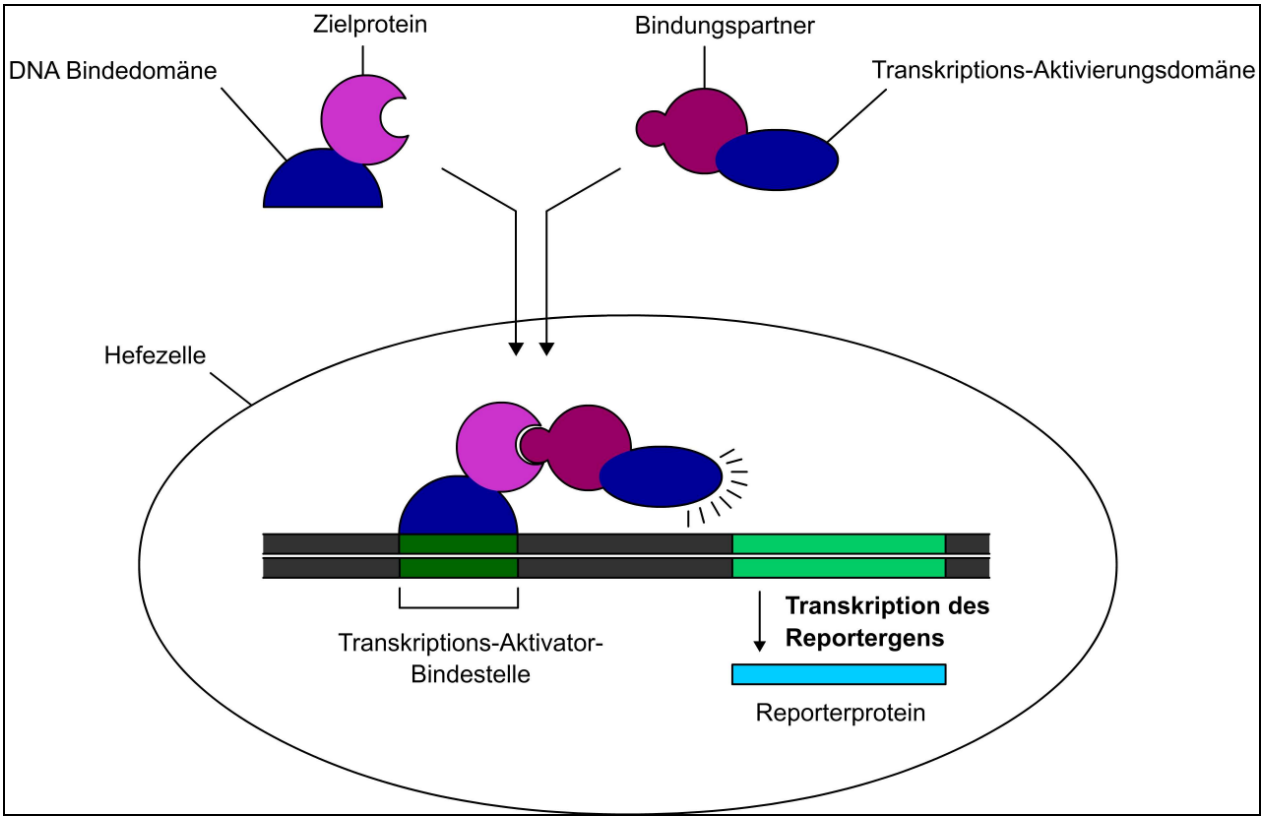


Fig. 2.5 Das Yeast Two-Hybrid Verfahren: Durch Bindung des Bindungspartners an das Zielprotein wird die Transkription des Reportergens aktiviert

3. Aktuelle Verfahren zur Vorhersage von Protein-Protein-Interaktionen

3.1. Phylogenetic Profiles - Methode

Die Grundannahme des Verfahrens von Pellegrini et. al [PEL99] ist, dass Proteine, die in funktionaler Beziehung zueinander stehen, korreliert evolvieren. Das heißt, dass funktional abhängige Proteine in einem neuen Organismus entweder alle konserviert bleiben oder alle während der Evolution verloren gehen.
Um diese korrelierte Evolution bei Proteinen messen zu können, wird zuerst definiert, was unter einem *phylogenetic profile* zu verstehen ist.

Definition: Sei n die Anzahl der verschiedenen betrachteten Organismen. Dann ist das *phylogenetic profile* eines Proteins ein Vektor der Länge n, bei dem an Position i eine eins steht, wenn zu dem betrachteten Protein in Organismus i ein orthologes Protein existiert und eine null andernfalls.

Haben zwei Proteine ein ähnliches phylogenetisches Profil - ähnlich meint hier, dass sich die Vektoren an höchstens einer Stelle unterscheiden - so wird für sie eine funktionale Beziehung vorhergesagt.
In [PEL99] dient als Ausgangsgenom der Untersuchung das Genom von Escherichia Coli. Zusätzlich werden 16 weitere, voll sequenzierte, Genome aus der TIGR Datenbank verwendet. Ein Protein gilt hier als homologes Protein, wenn es per BLAST [ALT90] einen Alignment Score erhält, der signifikant ist.

Obwohl durch dieses Verfahren die Evidenz für eine funktionale Beziehung hergestellt werden kann, ist dies nicht gleichbedeutend mit einem direkten physikalischen Kontakt der Proteine, da es bei biochemischen Prozessen in der Zelle oft mehrere Zwischenschritte gibt, bei denen die Schritte zwar voneinander abhängen, aber nicht jedes beteiligte Protein mit jedem anderen in direktem physikalischen Kontakt steht. Nichtsdestotrotz kann das Verfahren Anhaltspunkte für Protein-Protein-Interaktionen liefern.

Die größten Einschränkungen dieses Verfahrens sind zum einen, dass es nur für komplett sequenzierte Genome anwendbar ist, da ansonsten nicht mit absoluter Sicherheit angenommen werden kann, dass für ein Protein in dem nicht komplett sequenzierten Organismus kein homologes Protein existiert. Zum anderen kann das Verfahren nicht für die essentiellen Proteine angewendet werden, da sie in nahezu jedem Organismus vorkommen.

3.2. Conservation of Gene Neighborhood – Methode

Die Grundannahme der *Conservation of Gene Neighborhood* - Methode ist, dass die Gene der Proteine, die in funktionaler Beziehung zueinander stehen, in bakteriellen Genomen oft räumlich nah beieinander liegen und dass eine Evidenz für diese funktionale Beziehung noch stärker wird, wenn die räumliche Nähe in verschiedenen Organismen konserviert ist. Die Vorhersage einer funktionalen Beziehung zweier Proteine wird anhand von Fig 3.1 verdeutlicht.

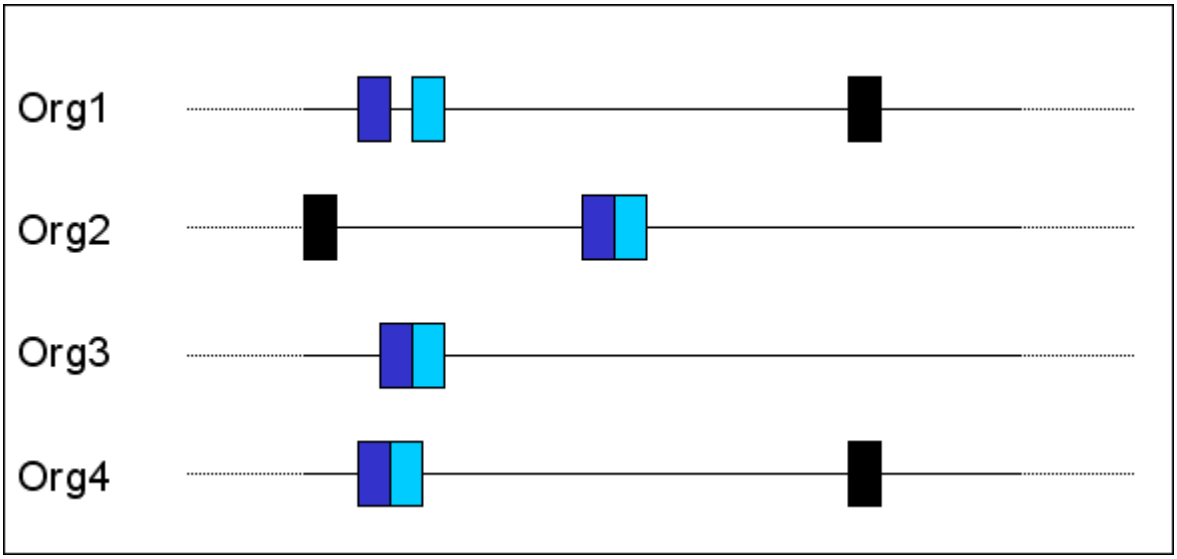


Fig. 3.1: Gene neighborhood – Methode: Jede waagerechte Linie steht für das Genom eines Organismus, jedes Rechteck für das Gen eines Proteins. Für die beiden blauen Proteine wird in diesem Fall eine funktionale Beziehung vorhergesagt, da ihre Nähe in verschiedenen Organismen erhalten ist.

Auch bei diesem Verfahren sei erwähnt, dass eine funktionale Beziehung zweier Proteine nicht gleichbedeutend mit direktem physikalischem Kontakt ist, dass sie aber ein guter Indikator für eine Protein-Protein Interaktion ist [DAN98].
Ein Problem dieses Verfahrens ist, dass es direkt nur für Bakterien angewendet werden kann, da hier die Genreihenfolge bedeutender ist als bei Archaen und Eukaryoten.

3.3. Gene Fusion – Methode

Die Grundannahme bei der *Gene Fusion* – Methode ist, dass, wenn zu zwei Proteinen in einem Organismus i in einem anderen Organismus j ein Fusionsprotein vorhanden ist, das eine hohe Ähnlichkeit zu beiden Proteinen aufweist, die beiden Proteine aus i potenzielle Interaktionspartner sind. Die Vorhersage einer funktionalen Beziehung zweier Protein mit Hilfe der *Gene Fusion* – Methode wird anhand von Fig 3.2 verdeutlicht.

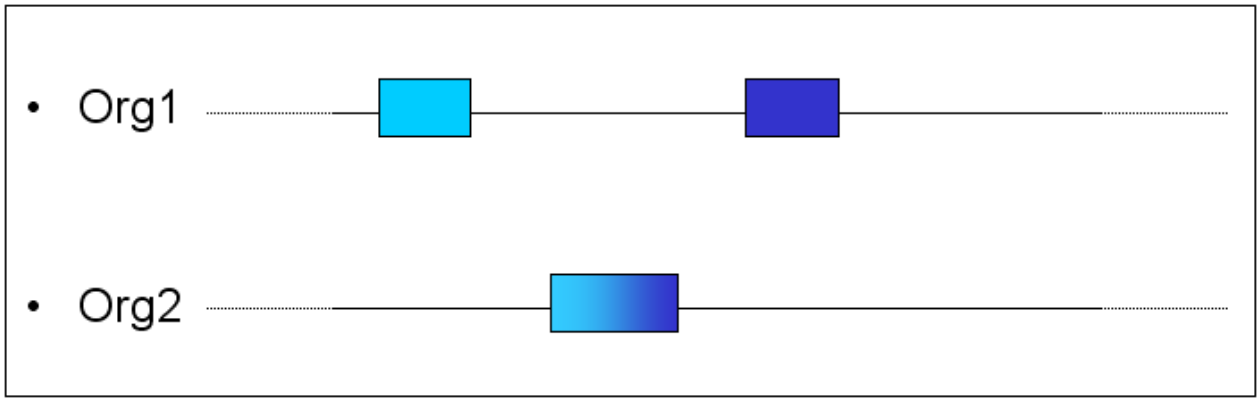


Fig 3.2 Die waagerechten Linien stehen für verschiedene Genome, die Rechtecke für die Gene von Proteinen. Für die beiden blauen Proteine aus Organismus eins würde in diesem Fall eine funktionale Beziehung vorhergesagt werden, da ein Fusionsprotein in einem anderen Organismus existiert, dass zu beiden Proteinen eine Ähnlichkeit aufweist (symbolisiert durch die gleichen Farben)

Die Fusionsproteine werden mit Hilfe von rekursiven Sequenzsuchen und multiplen Sequenzalignments gefunden und die beiden Proteine, die dem Fusionsprotein entsprechen, werden als potenzielle Interaktionspartner vorhergesagt [ENR99].

3.4. Mirrortree – Methode

Die Grundlage des *Mirrortree* – Verfahrens ist die Beobachtung, dass die phylogenetischen Bäume von Liganden und Rezeptoren, z.B. bei Insulin, eine höhere Ähnlichkeit aufweisen, als unter normalen evolutionären Bedingungen zu vermuten wäre [FRY96].
Das Ausgangsgenom in [PAZ01] ist Escherichia Coli. Zusätzlich werden 14 voll sequenzierte Bakteriengenome verwendet. Die homologen Sequenzen werden per

BLAST [ALT90] extrahiert mit einem Cutoff-Value von $P(N) < 1 \times 10^{-5}$ und anschließend mit ClustalW [HIG92] aligniert. Da nicht für jedes Protein in jedem Organismus ein homologes Protein existiert, für die weitere Evaluation aber auch nicht Proteinpaare mit zu kleinen phylogenetischen Bäumen betrachtet werden sollen, wird die Mindestzahl an Organismen mit homologen Proteinen, die pro Proteinpaar erreicht werden muss, auf elf gesetzt. Diese Schwelle wurde durch empirische Tests ermittelt. Erfüllt ein potenzielles Proteininteraktionspaar diese Mindestzahl nicht, so wird es nicht weiter betrachtet. Existieren für ein Escherichia Coli Protein in einem anderen Organismus mehrere homologe Proteine, so wird das Protein ausgewählt, das die größte Ähnlichkeit aufweist. Für die übrigbleibenden potenziellen Interaktionspartner werden mit Hilfe der McLachlan Aminosäure-Homologie-Matrix Distanzmatrizen erstellt [MCL71]. Danach wird für jedes potenzielle Interaktionspaar der lineare Korrelationskoeffizient r berechnet [PRE92].

$$r = \frac{\sum_{i=1}^{n} (R_i - \overline{R})(S_i - \overline{S})}{\sqrt{\sum_{i=1}^{n} (R_i - \overline{R})^2} \sqrt{\sum_{i=1}^{n} (S_i - \overline{S})^2}}$$

Hierbei ist n die Anzahl der Elemente in der Distanzmatrix, R_i das ite Element der ersten Distanzmatrix und S_i das ite Element der zweiten Distanzmatrix. $\overline{R}$ und $\overline{S}$ sind die jeweiligen Mittelwerte der Distanzmatrizen. Proteinpaare werden als Proteininteraktion vorhergesagt, wenn ihr Korrelationskoeffizient r einen vorher festgelegten Schwellwert übertrifft.

3.4.1. Erweiterung der Mirrortree - Methode

Sato et. al [SAT05] postulieren, dass das *Mirrortree*-Verfahren verbessert werden kann, indem die phylogenetische Information entfernt wird, die durch die immer gleiche Auswahl der Organismen für die homologen Proteine in der Distanzmatrix enthalten ist. Dazu wird mit drei verschiedenen Vorgehensweisen ein Richtungsvektor im Datenraum konstruiert, der die phylogenetische Information repräsentiert. Durch Projektion aller Daten in einen Raum, der orthogonal zum Richtungsvektor ist, wird dann die phylogenetische Information aus den Daten entfernt.
Als Datengrundlage dient Sato et. al eine Sammlung von 13 Paaren interagierender Proteine, für die ein physikalischer Kontakt in der DIP (siehe 5.1.1) vorhanden ist. Eine

zusätzliche Bedingung für die Proteine ist, dass keins der Proteine mehr als einen Interaktionspartner im Datensatz besitzen darf.
Die orthologen Proteine zur Erstellung der Distanzmatrizen (siehe 3.4) werden aus 40 bakteriellen Spezies mit Hilfe der KEGG/KO Datenbank [KAN04] extrahiert. Zur Berechnung der Distanzmatrizen wird jeweils zuerst mit Hilfe von MAFFT [KAT05] ein multiples Alignment erzeugt und die paarweisen Distanzen berechnet, die wiederum in PROTDIST aus dem PHYLIP Paket [FEL04] verwendet werden, um die Matrizen mit den genetischen Distanzen zu erstellen.
Wie bereits erwähnt, wird mit drei verschiedenen Vorgehensweisen ein Richtungsvektor berechnet, der die phylogenetische Information der Organismen repräsentieren soll. Hierbei repräsentiert der Richtungsvektor alle Einträge der oberen Dreiecksmatrix der Distanzmatrix. Im Folgenden wird die geordnete Abfolge dieser Einträge als phylogenetischer Vektor eines Proteins bezeichnet.
Die erste Vorgehensweise zur Erstellung des Richtungsvektors ist folgendermaßen: Zu jedem beteiligten Organismus wird die 16 S ribosomale RNA (rRNA) extrahiert und eine Distanzmatrix berechnet. Die 16 S rRNA wird verwendet, weil jeder der verwendeten Organismen eine Kopie des Gens für die 16 S rRNA enthält und die Sequenz als guter Indikator für die phylogenetischen Beziehungen der Organismen angesehen wird. Als Quelle für die rRNAs dient die KEGG/GENES Datenbank [KAN04] und das *Ribosomal Database Project-II Release 9* [GUS05]. Die paarweisen Distanzen werden mit Hilfe des *F84 Scoring Table* [KIS89] und dem DNADIST Modul des PHYLIP Pakets [FEL04] berechnet.
Bei der zweiten Vorgehensweise wird ein Durchschnittsvektor aller phylogenetischen Vektoren berechnet, wobei die einzelnen Vektoren vorher auf Standardabweichung eins normiert werden.
Bei der dritten Methode wird eine Korrelationsmatrix erstellt, bei der der Eintrag (i, j) der Korrelation zwischen dem normierten phylogenetischen Vektor des Proteins i und dem normierten phylogenetischen Vektor des Proteins j entspricht (siehe auch 6.1). Für die Korrelationsmatrix wird eine Hauptkomponentenanalyse (siehe 4.1) durchgeführt und der Richtungsvektor entspricht dann der normierten ersten Hauptkomponente (siehe 4.1.3).
Der erhaltene Richtungsvektor $\vec{u}$ wird im folgenden verwendet, um die Daten in einen Unterraum zu projizieren, der orthogonal zum Richtungsvektor ist. Hierzu wird der Projektionsoperator

$$P = I - \vec{u}\vec{u}^T$$

folgendermaßen verwendet. Sei $\vec{v}_i$ der phylogenetische Vektor eines Proteins. Dann ist

$$\vec{\varepsilon}_i = P\,\vec{v}_i = \vec{v}_i - \vec{u}\vec{u}^T\vec{v}_i$$

die Projektion des Vektors in dem zu $\vec{u}$ orthogonalen Unterraum. Sei

$$\vec{\varepsilon}_i^{\,*} = \frac{\vec{\varepsilon}_i - \vec{\mu}}{\sqrt{Var(\vec{\varepsilon}_i)}}$$

, dann wird die Korrelation r zwischen Protein i und Protein j in dem Unterraum berechnet durch

$$r_{i,j} = \vec{\varepsilon}_i^{\,*T}\,\vec{\varepsilon}_j^{\,*}$$

Schließlich wird wie beim normalen *Mirrortree*-Verfahren ein Schwellenwert festgelegt. Ist die Korrelation zwischen zwei Proteinen größer als dieser Schwellenwert, so wird für die beiden Proteine eine Interaktion vorhergesagt, andernfalls nicht.
Zur Wahl des Richtungsvektors (rRNA, Mittelwert, PCA) ist zu erwähnen, dass alle drei vorgestellten Methoden bessere Ergebnisse liefern, als das *Mirrortree*-Verfahren in Bezug auf die Spezifität. Das heißt unter allen Proteininteraktionen, die Vorhergesagt werden ist der Anteil der richtig erkannten Interaktionen bei diesen drei Verfahren größer. Zusätzlich wird auch eine Korrelation zwischen den verschiedenen Richtungsvektoren untersucht und festgestellt, dass der Betrag der linearen Korrelation sehr groß ist (größer gleich 0,95). Dies wird als Rechtfertigung verwendet, dass anstelle des ersten Richtungsvektors, bei dem zusätzlich die Information über die rRNAs der beteiligten Organismen benötigt werden, auch der zweite oder dritte Richtungsvektor (Mittelwert bzw. PCA) verwendet werden kann, um die phylogenetische Information aus den Daten zu entfernen und so das *Mirrortree*-Verfahren zu verbessern.

3.5. Vorhersage von Protein-Protein-Interaktionen mit Hilfe von Supportvektormaschinen

Die Idee des Verfahrens von Bock et. al [BOC01] ist, Protein-Protein-Interaktionen mit Hilfe von Supportvektormaschinen [BUR98] vorherzusagen. Hierbei werden die Proteine nur durch ihre Primärstruktur (siehe 2.3) und weitere Merkmale wie Hydrophobizität, Ladung und Oberflächenspannung repräsentiert.
Als Datengrundlage für die Positivbeispiele dient bei diesem Ansatz die DIP (siehe 5.1.1), die zum Zeitpunkt der Veröffentlichung der Methode 2664 Protein-Protein-Interaktionen beinhaltete. Die Negativbeispiele werden mit Hilfe des Programms Shufflet [COW99] generiert. Dieses Programm dient zur Randomisierung von Sequenzen unter Beibehaltung von k-mer Häufigkeiten. Die Negativbeispiele sind somit ähnlicher zu echten Proteinsequenzen als vollkommen zufällig generierte Proteinsequenzen.
Der Merkmalsvektor eines Proteins wird folgendermaßen konstruiert: Für die drei verwendeten Merkmale Hydrophobizität, Ladung und Oberflächenspannung wird jeweils ein Merkmalsvektor erstellt, der die Eigenschaften der Residuen in Bezug auf das entsprechende Merkmal repräsentiert. Jeder dieser Vektoren wird dann auf ein Einheitsintervall abgebildet (damit Proteine verschiedener Längen verglichen werden können). Der Merkmalsvektor eines Proteins ist schließlich die Konkatenation aller seiner einzelnen Merkmalsvektoren.
Für ein Protein-Protein-Interaktionspaar werden die beiden Merkmalsvektoren der betroffenen Proteine konkateniert. Dies wird für die Positivbeispiele mit echten Interaktionspartnern aus der DIP durchgeführt und für die Negativbeispiele mit Paaren der mit Shufflet randomisierten Proteinsequenzen.
Die Performanz des Verfahrens wird folgendermaßen gemessen: Die 2664 Positivbeispiele werden zufällig auf zwei Mengen aufgeteilt. Die eine Menge wird zum Training verwendet, die andere zum testen der Performanz. In der Beschreibung der Methode gehen Bock et. al nicht direkt auf die Anzahl der Negativbeispiele ein, aber anhand der angegebenen Insgesamtzahl der Interaktionen für Training und Test lässt sich vermuten, dass jeweils ca. 760 Negativbeispiele für Training und Test verwendet wurden.
Um Zufallseffekte zu beschränken, wird das Verfahren zehn Mal hintereinander durchgeführt und die Ergebnisse werden gemittelt.
Das Hauptproblem dieser Methode ist es, dass nicht sicher behauptet werden kann, dass die SVM nicht einfach nur die Trennung zwischen echten und randomisierten

Proteinsequenzen lernt. In der Veröffentlichung wurde auf dieses Problem hingewiesen und es wurden weitere Studien in Bezug auf dieses Thema angekündigt. Da hierzu allerdings bis heute nichts veröffentlicht wurde, ist davon auszugehen, dass nicht gezeigt werden konnte, dass die Methode auch für ein echtes Szenario anwendbar ist. Zusätzlich ist fraglich, ob die Merkmalsvektoren auf ein Einheitsintervall skaliert werden sollten, damit kleine Proteine auch mit großen Proteinen interagieren können, da hierdurch die Merkmalsfunktionen sehr stark verfälscht werden. Des weiteren ist nicht klar, ob die Trennung zwischen Interaktionspaaren und Nicht-Interaktionspaaren sinnvoll ist, weil es wahrscheinlich Informationen gibt, die verwendet werden können um echte Interaktionspaare vorherzusagen, es aber vermutlich keine Informationen gibt, durch die mit Sicherheit festgestellt werden kann, dass zwei Proteine nicht interagieren. Hier wäre also ein transduktiver Lernansatz besser, bei dem nur die Informationen der Positivbeispiele verwendet werden. Dieser Ansatz wird in dieser Masterarbeit verfolgt.

4. Verfahren zur Merkmalsextraktion

Es existieren sehr viele verschiedene Verfahren zur Merkmalsextraktion. Hierbei ist das Hauptziel, bestimmte Merkmale aus den Daten zu extrahieren, die für eine bestimmte Aufgabe gut geeignet sind. In dieser Arbeit wird die Merkmalsextraktion zur Vorhersage von Protein-Protein-Interaktionen verwendet. Hier soll exemplarisch die Hauptkomponentenanalyse dargestellt werden, da sie ein weit verbreitetes Verfahren zur Merkmalsextraktion darstellt. Eine Übersicht über weitere Verfahren ist in [TIB01] nachzulesen.

4.1. Hauptkomponentenanalyse

Bei der Hauptkomponentenanalyse geht es darum eine lineare Projektion der Daten zu finden, die die maximale Ausdehnung der Verteilung repräsentiert. Wie in Fig. 4.1 zu sehen ist, bedeutet dies gleichzeitig, dass für diese Projektion der quadratische Abstand zu den Datenpunkten minimal ist. Man erhofft sich dadurch eine Repräsentation niedrigerer Dimensionalität der Daten, in der wichtige Eigenschaften erhalten bleiben und besser interpretierbar, bzw. deutlicher sichtbar sind.

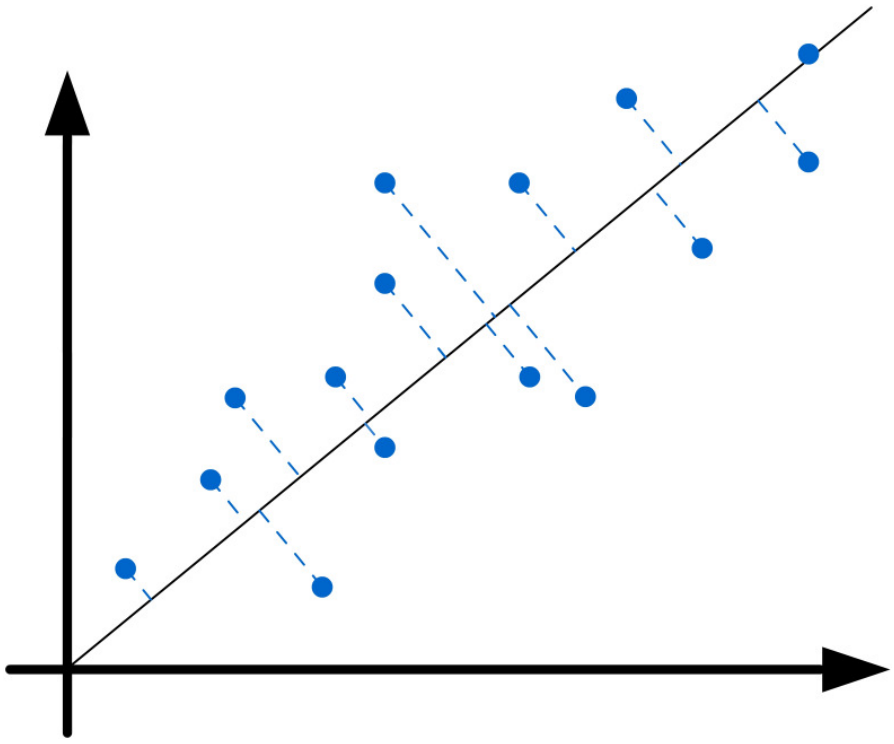


Fig. 4.1 Projektion der Daten auf die Gerade, die der maximalen Ausdehnung der Daten entspricht

Da bei der Hauptkomponentenanalyse eine Projektion der Daten vollzogen wird, soll hier kurz erläutert werden, was unter einer Projektion eines Datenpunktes $\vec{x}$ auf eine Gerade $\lambda\vec{v}$ durch den Ursprung mit $\|\vec{v}\|=1$ verstanden wird.

4.1.1. Projektion von $\vec{x}$ auf die Gerade $\lambda\vec{v}$

Zur Vereinfachung der Rechnungen wird angenommen, dass die Gerade $g:\lambda\vec{v}$ durch den Ursprung geht und dass $\|\vec{v}\|=1$.

Wird $\vec{x}$ auf eine Gerade projiziert, so geschieht das in der Weise, dass ein Punkt $\lambda_{\vec{x}}\vec{v}$ auf der Gerade gefunden wird, der zu dem Datenpunkt den minimalsten quadratischen Abstand besitzt. In Fig. 4.2 ist zu sehen, dass dieser Abstand genau dann minimal ist, wenn die Gerade durch die Punkte $\vec{x}$ und $\lambda_{\vec{x}}\vec{v}$ senkrecht zu g ist.

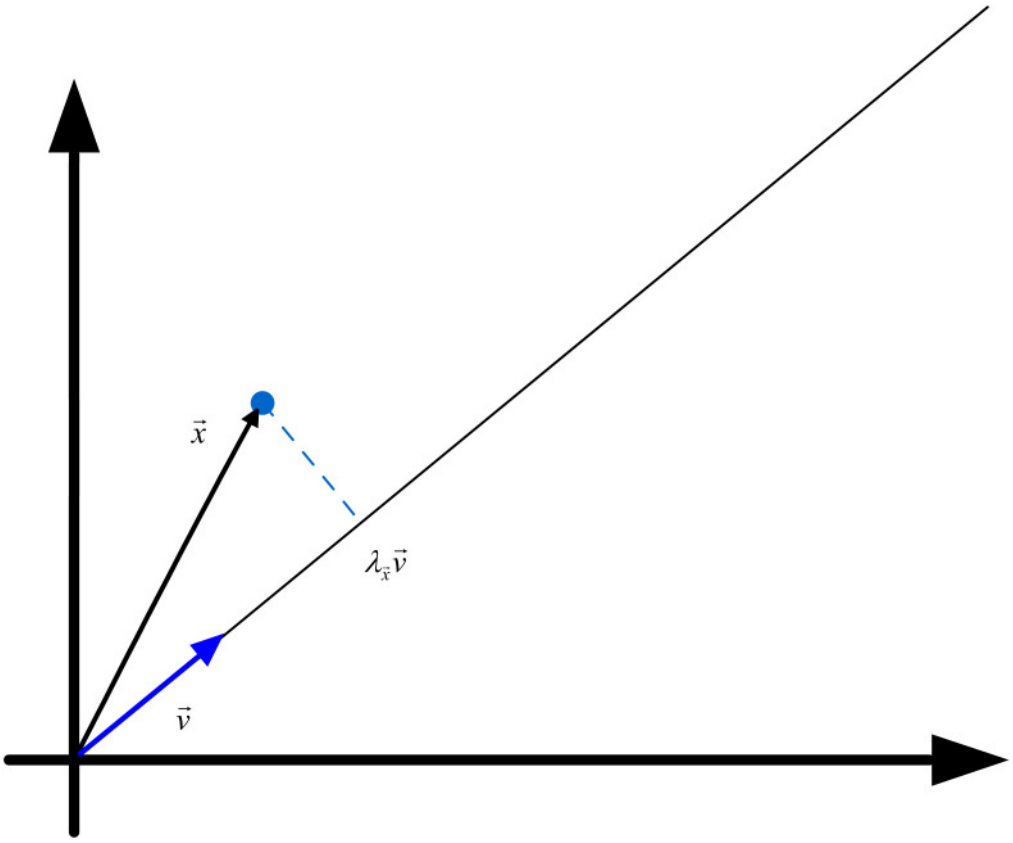


Fig. 4.2 Projektion von $\vec{x}$ auf die Gerade $\lambda\vec{v}$

Gesucht ist der Index $\lambda_{\vec{x}}$, für den gilt, dass $\left\|\vec{x}-\lambda_{\vec{x}}\vec{v}\right\|^2$ minimal ist unter allen möglichen λ aus den reellen Zahlen. Dieser Index wird auch Projektionsindex genannt, weil er die Repräsentation des Datenpunktes im reduzierten eindimensionalen Koordinatensystem der Geraden darstellt.

$$\left\|\vec{x}-\lambda_{\vec{x}}\vec{v}\right\|^2 = \min_{\lambda}\left\|\vec{x}-\lambda\vec{v}\right\|^2$$

Einige Umformungen zeigen, dass für ein Minimum gelten muss: $\lambda = \vec{x}^T\vec{v}$. Der gesuchte Projektionsindex $\lambda_{\vec{x}}$ ist also gegeben durch das Skalarprodukt $\vec{x}^T\vec{v}$.

4.1.2. Maximierung der Varianz des Projektionsindex

Bei der Hauptkomponentenanalyse wird nach der Richtung $\vec{v}$ einer Geraden, bzw. Achse gesucht, so dass die Varianz des Projektionsindex $\lambda_{\vec{x}} = \vec{x}^T\vec{v}$ maximal wird.

Sei $\{\vec{x}_1,\vec{x}_2,...,\vec{x}_N\}$ eine Stichprobe, die nullpunktzentriert ist. Das bedeutet, dass $\sum_{i=1}^{N}\vec{x}_i = \vec{0}$ und deswegen auch $\left(\sum_{i=1}^{N}\vec{x}_i^{\,T}\right)\vec{v} = \sum_{i=1}^{N}\vec{x}_i^{\,T}\vec{v} = \frac{1}{N}\sum_{i=1}^{N}\vec{x}_i^{\,T}\vec{v} = 0$. Das heißt, der geschätzte Erwartungswert für $\lambda_{\vec{x}} = \vec{x}^T\vec{v}$ ist gleich null. Hierdurch vereinfacht sich die Formel zur Bestimmung der geschätzten Varianz von $\lambda_{\vec{x}}$ auf die Formel

$$F(\vec{v}) = \frac{1}{N}\sum_{i=1}^{N}\left(\vec{x}_i^{\,T}\vec{v}\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\vec{v}^T\vec{x}_i\vec{x}_i^{\,T}\vec{v} = \vec{v}^T\left(\frac{1}{N}\sum_{i=1}^{N}\vec{x}_i\vec{x}_i^{\,T}\right)\vec{v} = \vec{v}^T\hat{C}\vec{v} = \frac{\vec{w}^T\hat{C}\vec{w}}{\vec{w}^T\vec{w}}$$

Hierbei steht $\hat{C}$ für die geschätzte Kovarianzmatrix. Außerdem ist $\vec{v} = \frac{\vec{w}}{\left\|\vec{w}\right\|}$.

Zur Maximierung der Varianz, wird also $F(\vec{v})$ maximiert. Einige Umformungen zeigen, dass die Vektoren $\vec{w}$ die Funktion $F(\vec{v})$ maximieren, für die gilt:

$$\hat{C}\vec{w} = \frac{\vec{w}^T\hat{C}\ \vec{w}}{\vec{w}^T\vec{w}}\ \vec{w}$$

Somit erfüllen alle Eigenvektoren der Kovarianzmatrix die notwendige Bedingung für ein Maximum, da alle Vektoren $\vec{w}$, die die letzte Gleichung erfüllen per Definition Eigenvektoren sind.

Für Eigenvektoren $\vec{u}_i$ mit $\|\vec{u}_i\| = 1$ gilt

$$M \cdot \vec{u}_i = \nu_i \cdot \vec{u}_i \Leftrightarrow \nu_i = \frac{\vec{u}_i^{\,T} \cdot M \cdot \vec{u}_i}{\vec{u}_i^{\,T} \vec{u}_i} \Leftrightarrow \nu_i = \vec{u}_i^{\,T} \cdot M \cdot \vec{u}_i$$

Folglich maximiert der Eigenvektor $\vec{u}_i = \vec{w}$ zum größten Eigenwert ν_i von $\hat{C}$ die Funktion $F(\vec{w}) = \frac{\vec{w}^T \hat{C}\, \vec{w}}{\vec{w}^T \vec{w}}$.

Somit verläuft die Achse, die die Varianz des Projektionsindex maximiert in Richtung des ersten Eigenvektors der geschätzten Kovarianzmatrix. Diese Achse wird in der Literatur auch oft als erste Hauptkomponente bezeichnet.

4.1.3. Hauptkomponenten

Die Richtung der ersten Hauptkomponente einer Verteilung entspricht der Achse, die in Richtung der maximalen Varianz der Daten verläuft. Wie in 4.1.2 gezeigt wurde maximiert der Eigenvektor zum größten Eigenwert der geschätzten Kovarianzmatrix die Varianz des Projektionsindex. Folglich ist dieser Eigenvektor eine Schätzung für die erste Hauptkomponentenrichtung. Sei X die Datenmatrix und $\vec{u}_1$ der Eigenvektor zum größten Eigenwert der geschätzten Kovarianzmatrix, dann ist die geschätzte erste Hauptkomponente, $\vec{z}^{(1)}$, gegeben durch:

$$\vec{z}^{(1)\,T} = \vec{u}_1^{\,T} \cdot \mathrm{X}$$

Somit enthält $\vec{z}^{(1)}$ die Projektionsindizes der Variablen aus X auf den Eigenvektor $\vec{u}_1$. Ferner ist die geschätzte Varianz der ersten Hauptkomponente gleich dem größten Eigenwert, denn:

$$\sigma_1^{\,2} = \frac{1}{n} \sum_{i=1}^{n} \vec{u}_1^{\,T} \vec{x}_i \vec{x}_i^{\,T} \vec{u}_1 = \vec{u}_1^{\,T} \hat{C}\, \vec{u}_1 = \nu_1$$

Analog ist die ite Hauptkomponente gegeben durch:

$$\vec{z}^{(i)\,T} = \vec{u}_i^{\,T} \cdot X$$

mit geschätzter Varianz $\sigma_i^2 = \nu_i$.

Die Summe $\sum_{i=1}^{d} \sigma_i^2$ wird daher auch oft als Gesamtvarianz der Verteilung bezeichnet.

4.2. Unsupervised Kernel Regression

4.2.1. Kern-Regression

Die Idee der Kern-Regression ist, den *k-Nearest-Neighbor-Average* Schätzer [HAS01]

$$\hat{f}(\vec{x}) = Ave(\vec{y}_i \mid \vec{x}_i \in N_k(\vec{x}))$$

für die Regressionsfunktion $E(Y \mid X = \vec{x})$ dahingehend zu verändern, dass weiter entfernte Nachbarn weniger zur Vorhersage des Labels beitragen. Hierbei ist N_k die Menge der k nächsten Nachbarn von $\vec{x}$, gemessen in euklidischer Distanz und $Ave(S) = \frac{1}{\#S}\sum_{i=1}^{\#S} S_i$. Diese Veränderung ist unter anderem sinnvoll, da die Funktion des *k-Nearest-Neighbor-Average* Schätzers, wie in Fig. 4.1 zu sehen ist, nicht stetig ist. Hierzu wird unter anderem der *Nadaraya-Watson Kernel* [NAD64]

$$\hat{f}(\vec{x}_0) = \frac{\sum_{i=1}^{N} K_\lambda(\vec{x}_0, \vec{x}_i)\vec{y}_i}{\sum_{i=1}^{N} K_\lambda(\vec{x}_0, \vec{x}_i)}$$

als Schätzer verwendet, wobei als Kern zum Beispiel ein radialer Epanechnikow Kern seine Anwendung findet.

$$K_\lambda(\vec{x}_0, \vec{x}) = D\left(\frac{\|\vec{x} - \vec{x}_0\|}{\lambda}\right) \text{ mit } D(t) \propto \begin{cases} (1-t^2), & falls\ |t| \le 1 \\ 0, & sonst \end{cases}$$

Wie man an der Kernfunktion sieht, bestimmt das λ darüber, wie weit ein Punkt entfernt sein darf um noch einen Wert ungleich null zu erhalten. λ wird daher als Kernbreite bezeichnet. Je größer die Kernbreite ist, umso mehr tragen weiter entfernt liegende Punkte zur Schätzung bei.
Für den Vergleich des *k-Nearest-Neighbor-Average* Schätzers mit dem *Nadaraya-Watson Kernel* in Fig. 4.1 wurden Zufallsdatenpunkte erzeugt mit $y = \cos(4 * x) + \varepsilon$,

wobei das Rauschen ε eine normalverteilte Zufallsvariable mit Mittelwert 0 und Varianz 1/3 ist. Die Kernbreite ist 0,15 und k ist gleich 25.

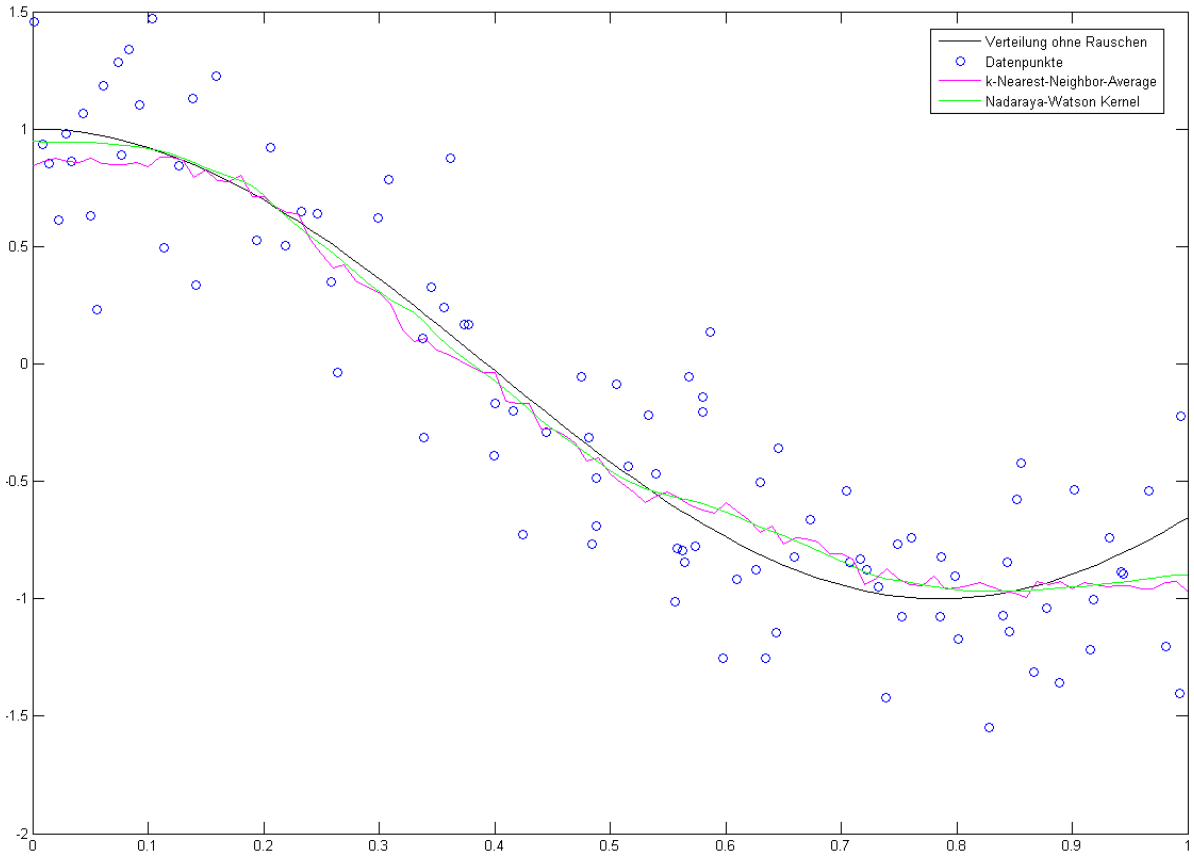


Fig 4.1 Vergleich des *k-Nearest-Neighbor-Average* Schätzers mit dem *Nadaraya-Watson Kernel* Schätzer: Die schwarze Kurve entspricht der ursprünglichen Verteilung der Daten ohne Rauschen. Die magentafarbene Kurve entspricht dem *k-Nearest-Neighbor-Average* Schätzer und die Grüne dem *Nadaraya-Watson Kernel* Schätzer.

4.2.2. Verfahren

Die Grundidee der *Unsupervised Kernel Regression* (*UKR*) [MEI05] ist, den *Nadaraya-Watson Kernel* Regressionsschätzer

$$\hat{f}(\vec{x}_0) = \frac{\sum_{i=1}^{N} K_\lambda(\vec{x}_0, \vec{x}_i)\vec{y}_i}{\sum_{i=1}^{N} K_\lambda(\vec{x}_0, \vec{x}_i)}$$

(siehe 4.2.1) für das unüberwachte Lernen von Funktionen zu generalisieren. Die fehlenden Eingabevariablen $\{\vec{x}_1, \vec{x}_2, ..., \vec{x}_N\}$, $\vec{x}_i \in \mathrm{R}^q$, werden als Parameter betrachtet, die zusammen mit der Regressionsfunktion gelernt werden. Sie dienen als niedrig dimensionale latente Repräsentationen der Datenpunkte $\vec{y}_i \in \mathrm{R}^d$. Um die Komplexität der *UKR*-Mannigfaltigkeit zu beschränken seien alle latenten Variablen aus einem kompakten Teilraum des R^d [Mei05]. Im Folgenden soll mit $\mathrm{X} = [\vec{x}_1, \vec{x}_2, ..., \vec{x}_N]$ die Matrix mit den latenten Datenpunkten als Spaltenvektoren und die Matrix $\mathrm{Y} = [\vec{y}_1, \vec{y}_2, ..., \vec{y}_N]$ mit einer unabhängigen und identisch verteilten Stichprobe der Datenraumverteilung bezeichnet werden. Mit diesen Festlegungen sieht die *UKR*-Funktion folgendermaßen aus

$$f(\vec{x}; \mathrm{X}) = \sum_{i=1}^{N} \frac{K(\vec{x} - \vec{x}_i)}{\sum_{j=1}^{N} K(\vec{x} - \vec{x}_j)} \vec{y}_i = \mathrm{Y} b(\vec{x}; \mathrm{X}) \quad ,$$

wobei der N-Vektor $b(\vec{x}; \mathrm{X})$ die latenten kernbasierten Basisfunktionen enthält. Für i = 1 bis N gilt folglich:

$$b_i(\vec{x}; \mathrm{X}) = \frac{K(\vec{x} - \vec{x}_i)}{\sum_{j=1}^{N} K(\vec{x} - \vec{x}_j)}$$

Die Summe aller Basisfunktionen ist gleich eins, da jede Basisfunktion mit $\sum_{j=1}^{N} K(\vec{x} - \vec{x}_j)$ normiert wird.

Analog zum mittleren quadratischen Fehler beim überwachten Fall, wird als Qualitätsmaß zur Bewertung der latenten Variablen der mittlere Rekonstruktionsfehler im Datenraum gemessen. Sei Y eine d x N dimensionale Matrix, die pro Spalte einen Datenvektor enthält und von N q-dimensionalen latenten Variablen $\vec{x}_i$ rekonstruiert wurde, dann ist der mittlere Rekonstruktionsfehler

$$R(\mathrm{X}) = \frac{1}{N} \sum_{i=1}^{N} \left\| \vec{y}_i - f(\vec{x}_i; \mathrm{X}) \right\|^2 = \frac{1}{N} \left\| \mathrm{Y} - \mathrm{YB}(\mathrm{X}) \right\|_F^2$$

, wobei $\| \ \|_F$ für die Frobenius-Norm steht und $\mathrm{B}(\mathrm{X}) = [b(\vec{x}_1;\mathrm{X}), b(\vec{x}_2;\mathrm{X}),..., b(\vec{x}_N;\mathrm{X})]$ die Werte der latenten Basisfunktionen enthält.

4.2.3. UKR im Merkmalsraum

Die Fehlerfunktion $\mathrm{R}(\mathrm{X})$ kann auch so umgeformt werden, dass nur innere Produkte der Datenpunkte verwendet werden.

$$R(\mathrm{X}) = \frac{1}{N}\left\|\mathrm{Y} - \mathrm{YB}(\mathrm{X})\right\|_F^2 = \frac{1}{N}\left\|\mathrm{Y}\overline{\mathrm{B}}\right\|_F^2 = \frac{1}{N} tr\left(\overline{\mathrm{B}}^T \mathrm{Y}^T \mathrm{Y}\overline{\mathrm{B}}\right) = \frac{1}{N} tr\left(\overline{\mathrm{B}}^T \mathrm{G}\overline{\mathrm{B}}\right)$$

In diesem Fall ist $\overline{\mathrm{B}} = \mathrm{I} - \mathrm{B}(\mathrm{X})$. $\mathrm{G} = \mathrm{Y}^T\mathrm{Y}$ steht für die Grammatrix, deren Einträge die Form $\mathrm{G}_{ij} = \left\langle \vec{y}_i, \vec{y}_j \right\rangle$ haben. Das heißt, dass an jedem Eintrag (i, j) der Grammatrix das Skalarprodukt von Spalte i und Spalte j der Matrix Y steht. Wird die Grammatrix durch eine Kernmatrix K ersetzt, die eine implizite Abbildung ϕ der Datenpunkte in einen Merkmalsraum ausführt, da für sie gilt

$$k\left(\vec{y}_i, \vec{y}_j\right) = \left\langle \phi(\vec{y}_i), \phi(\vec{y}_j) \right\rangle \quad ,$$

so kann die Merkmalsraum-*UKR* ausgeführt werden ohne dass die inneren Produkte im Merkmalsraum explizit ausgerechnet werden müssen.

$$R(\mathrm{X}) = \frac{1}{N}\left\|\mathrm{Y}\overline{\mathrm{B}}\right\|_F^2 = \frac{1}{N} tr\left(\overline{\mathrm{B}}^T \mathrm{G}\overline{\mathrm{B}}\right) = \frac{1}{N} tr\left(\overline{\mathrm{B}}^T K(\mathrm{Y})\overline{\mathrm{B}}\right)$$

4.2.4. UKR im latenten Raum

Zusätzlich zu den Methoden zur Bestimmung der Matrix X in [MEI05] existiert eine weitere Variante, die in [MEM03] vorgestellt wird. Die Idee bei diesem Verfahren ist, die Rolle der latenten Variablen und der beobachtbaren Variablen zu vertauschen und die Minimierung des mittleren Rekonstruktionsfehlers im latenten Raum der $\vec{x}_i$ zu verfolgen. Der *Nadaraya-Watson Kernel* Schätzer sieht dann folgendermaßen aus:

$$\vec{x} = \hat{f}(\vec{y}_0) = \frac{\sum_{i=1}^{N} K_\lambda(\vec{y}_0, \vec{y}_i)\vec{x}_i}{\sum_{i=1}^{N} K_\lambda(\vec{y}_0, \vec{y}_i)}$$

Der mittlere Rekonstruktionsfehler im latenten Datenraum ist dann

$$R^{lat}(\mathrm{X}) = \frac{1}{N}\sum_{i=1}^{N} \left\| \vec{x}_i - f(\vec{y}_i; \mathrm{Y}) \right\|^2 = \frac{1}{N} \left\| \mathrm{X} - \mathrm{XB(Y)} \right\|_F^2$$

Weitere Umformungen zeigen, dass

$$\frac{1}{N} \left\| \mathrm{X} - \mathrm{XB(Y)} \right\|_F^2 = \frac{1}{N} \left\| \mathrm{X(I - B(Y))} \right\|_F^2 = \frac{1}{N} tr\left(\mathrm{X\left(I - B - B^T + BB^T \right) X^T} \right)$$

mit $\mathrm{B(Y)} = \mathrm{B}$. Zur Minimierung von $R^{lat}(\mathrm{X})$ reicht es folglich nach [HOR94] eine Eigenwert-Dekomposition zu berechnen, da $\mathrm{I - B - B}^T + \mathrm{BB}^T$ quadratisch und positiv-semidefinit ist. Der Eigenvektor zum zweitkleinsten Eigenwert minimiert die Fehlerfunktion in nicht trivialer Weise. Der Eigenvektor zum kleinsten Eigenwert, der den Wert null hat, gehört zu der trivialen Lösung, bei der alle Einträge von X gleich sind (im folgenden Beispiel gleich c), da dann folglich

$$\frac{1}{N} \left\| \mathrm{X} - \mathrm{XB(Y)} \right\|_F^2 = \frac{1}{N} \left\| c \cdot 1 - \mathrm{XB(Y)} \right\|_F^2 = \frac{1}{N} \left\| c \cdot 1 - c \cdot 1 \right\|_F^2 = 0$$

$\mathrm{XB(Y)} = c \cdot 1$ gilt für die beschriebene Matrix X, da für jede Spalte von $\mathrm{B(Y)}$ die Summe aller Elemente 1 ist (siehe 4.2.2).

4.3. Semi Supervised Kernel Regression

Die Semi Supervised Kernel Regression (*SSKR*) wurde in dieser Arbeit zur Vorhersage von Protein-Protein-Interaktionen realisiert.
Die Idee bei der *SSKR* ist, die Abbildung in den latenten Datenraum so auszurichten, dass interagierende Proteine nah zueinander abgebildet werden und Proteine, für die

keine Interaktion bekannt ist, weit entfernt voneinander. Hierzu wird in dieser Arbeit die Menge $E_{Training}$ (siehe 6.3) verwendet und die Performanz der Methode wird mit der Menge E_{Test} (siehe 6.3) gemessen. Für die Bestimmung von X wird eine leicht modifizierte Variante der *UKR im latenten Raum* (siehe 4.2.4) verwendet.
Für jede Interaktion aus $E_{Training}$ wird ein Repräsentant ausgewählt. Die Wahl dieses Repräsentanten ist beliebig. Der mittlere Rekonstruktionsfehler im latenten Datenraum

$$R^{lat}(\mathrm{X}) = \frac{1}{N}\sum_{i=1}^{N}\left\|\vec{x}_i - f(\vec{y}_i;\mathrm{Y})\right\|^2 = \frac{1}{N}\left\|\mathrm{X}-\mathrm{XB}\right\|_F^2$$

wird dann ersetzt durch

$$R^{lat}(\tilde{\mathrm{X}}) = \frac{1}{N}\left\|\tilde{\mathrm{X}}\mathrm{S}-\tilde{\mathrm{X}}\tilde{\mathrm{B}}\right\|_F^2$$

Hierbei enthält $\tilde{\mathrm{X}}$ nur die Repräsentanten der Interaktionen aus $E_{Training}$ und alle Proteine, die nicht in eine Interaktion aus $E_{Training}$ involviert sind. S ist die „Selektormatrix“, die eine Spalte aus $\tilde{\mathrm{X}}$ selektiert. Die Selektormatrix ist folgendermaßen definiert. In jeder Spalte befindet sich nur eine eins und die restlichen Einträge sind null. Die eins befindet sich an der Position, die dem Index der Spalte aus $\tilde{\mathrm{X}}$ entspricht, die selektiert werden soll. Das heißt, dass für die Interaktion (i, j) aus $E_{Training}$ S in den Spalten i und j an der Stelle eine eins besitzt, die dem Index des Repräsentanten entspricht. Ein Besipiel für das Prinzip der Selektormatrix ist folgendes:

$$\tilde{\mathrm{X}}\cdot\mathrm{S} = [\vec{x}_1,\vec{x}_2,\vec{x}_4]\cdot\begin{pmatrix}1&0&0&0\\0&1&1&0\\0&0&0&1\end{pmatrix} = [\vec{x}_1,\vec{x}_2,\vec{x}_2,\vec{x}_4]$$

Hier wäre die einzige Interaktion aus $E_{Training}$ die Interaktion (2, 3) und $\vec{x}_2$ der Repräsentant dieser Interaktion.
Für jede Interaktion (i, j) aus $E_{Training}$ enthält $\tilde{B}$ in der Zeile, die dem Repräsentanten der Interaktion entspricht, die Summe der Zeilen i und j aus B. Folglich gilt auch für jede

Spalte von $\tilde{\mathrm{B}}$, dass die Spaltensumme gleich eins ist. Ein Beispiel für die Erstellung von $\tilde{B}$ ist in Fig. 4.2 dargestellt.

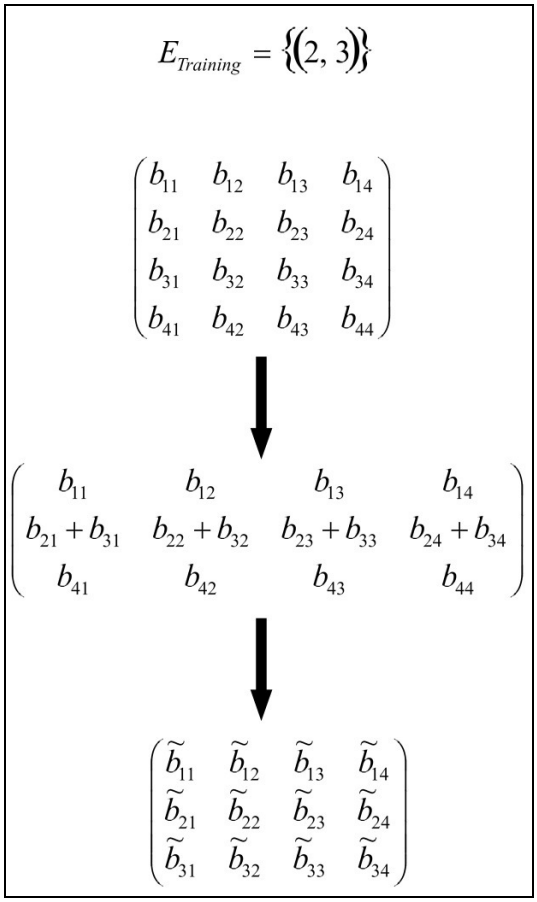


Fig. 4.2 Konstruktion von $\tilde{\mathrm{B}}$

$\tilde{\mathrm{X}}$ wird durch die Optimierung des minimalen Rekonstruktionsfehlers im latenten Datenraum bestimmt.

$$R^{lat}\left(\tilde{\mathrm{X}}\right) = \frac{1}{N}\left\|\tilde{\mathrm{X}}\mathrm{S} - \tilde{\mathrm{X}}\tilde{\mathrm{B}}\right\|_F^2 = \frac{1}{N}\left\|\tilde{\mathrm{X}}\left(\mathrm{S} - \tilde{\mathrm{B}}\right)\right\|_F^2 = \frac{1}{N} tr\left(\tilde{\mathrm{X}}\ \left(\mathrm{S} - \tilde{\mathrm{B}}\right)\left(\mathrm{S} - \tilde{\mathrm{B}}\right)^T \tilde{\mathrm{X}}^T\right)$$

$$= \frac{1}{N} tr\left(\tilde{\mathrm{X}}\ \left[\mathrm{S}\mathrm{S}^T - \mathrm{S}\tilde{\mathrm{B}}^T - \tilde{\mathrm{B}}\mathrm{S}^T + \tilde{\mathrm{B}}\tilde{\mathrm{B}}^T\right] \tilde{\mathrm{X}}^T\right)$$

Da zwischen $\tilde{\mathrm{X}}$ und $\tilde{\mathrm{X}}^T$ eine quadratische, positiv semi-definite Matrix steht, löst der Eigenvektor zum kleinsten Eigenwert der Matrix das Minimierungsproblem. Dadurch, dass dieser Vektor allerdings zu der trivialen Lösung gehört, bei der alle Einträge von $\tilde{\mathrm{X}}$

gleich sind (siehe 4.2.4), ist der Eigenvektor zum zweitkleinsten Eigenwert der Vektor, der die gesuchte Lösung repräsentiert.
Da bei einer Reduktion auf eine Dimension sehr viel Information verloren gehen kann, ist es auch denkbar mehrere Eigenvektoren zu verwenden. Hierzu werden die weiteren Eigenvektoren zu den nächstkleineren Eigenwerten herangezogen. Ist also zum Beispiel eine Repräsentation mit vier Dimensionen erwünscht, so werden die Eigenvektoren zum zweitkleinsten, drittkleinsten, viertkleinsten und fünftkleinsten Eigenwert als Repräsentation im latenten Datenraum verwendet.

5. Datengrundlage

5.1. Datenbanken für Protein-Protein-Interaktionen

5.1.1. DIP: Database of Interacting Proteins

Die Database of Interacting Proteins (DIP) [SAL04] wurde an der UCLA entwickelt, um binäre Protein-Protein-Interaktionen zu speichern, die aus einzelnen wissenschaftlichen Veröffentlichungen extrahiert wurden. Mittlerweile werden die Protein-Interaktions-Informationen teilweise auch automatisch extrahiert. Der aktuelle Datenbankzustand ist in Tab. 5.1 aufgelistet.

Anzahl der Proteine	18672
Anzahl der Organismen	109
Anzahl der Interaktionen	53463
Anzahl an Experimenten, die eine bestimmte Interaktion beschreiben	59858

Tab. 5.1 DIP-Datenbankzustand am 31.7.2005 [http://dip.doe-mbi.ucla.edu/dip/Stat.cgi]

Die DIP bietet unterschiedliche Dateien zum Download an. Zum einen gibt es Interaktionen, die durch High-Throughput-Messungen (zum Beispiel durch das Yeast Two-Hybrid Verfahren (siehe 2.6)) gefunden wurden für die Organismen *C. elegans*, *D. melanogaster* und *S. cerevisiae*. Zum anderen gibt es eine Datei mit allen Interaktionen, die in der DIP zu finden sind. Für Organismen mit mehr als 200 bekannten Interaktionen gibt es organismusspezifische Dateien. Am 31.7.2005 war dies der Fall für *C. elegans*, *D. melanogaster*, *S. cerevisiae*, *E. coli*, *H. pylori*, *H. sapiens* und *M. musculus*.
Es gibt drei verschiedene Dateiformate: ein eigenes reines Textformat, ein eigenes XML Format (XIN) und ein gemeinsam mit der BIND [BAD03] und der MINT Datenbank [ZAN02] entwickeltes XML Format (PSI MI [http://psidev.sourceforge.net/mi/xml/doc/user/]), das seit dem Release vom 31.7.2005 zur Verfügung steht.

5.1.2. BIND: Biomolecular Interaction Network Database

Die Biomolecular Interaction Network Database [BAD03] ist eine Sammlung von Informationen über molekulare Interaktionen. Die Informationsquellen sind zum einen High-Throughput-Messungen und zum anderen manuell gesammelte Informationen aus der wissenschaftlichen Literatur.
Die Datenbank unterscheidet zwischen drei verschiedenen Verbindungstypen: Interaktionen zwischen zwei Molekülen, molekulare Komplexe aus einer oder mehreren Interaktionen und sogenannte *Pathways*, die aus einer Folge von Interaktionen bestehen.
Die BIND sammelt zudem nicht nur Interaktionsdaten für Proteine sondern auch für DNA, RNA, Liganden, molekulare Komplexe und Gene. Der aktuelle Datenbankzustand ist in Tab. 5.2 zu sehen.

Gesamtzahl der Protein-Protein-Interaktionen	80422
Low-Throughput Einträge (manuell aus der Literatur extrahiert)	10674
High-Throughput Einträge	42164
Importierte Datenbankeinträge	27584

Tab. 5.2 BIND Datenbankzustand vom 22.8.2005 in Bezug auf Protein-Protein-Interaktionen [http://bind.ca/Action?pg=15000]

Es gibt bei der BIND drei verschiedene Dateiformate: NCBI ASN.1 Format, ein eigenes XML Format (BIND) und ein gemeinsam mit der DIP [SAL04] und der MINT Datenbank [ZAN02] entwickeltes XML Format (PSI MI [http://psidev.sourceforge.net/mi/xml/doc/user/]).

5.1.3. MINT: Molecular INTeraction

In der MINT Datenbank [ZAN02] sind Informationen über Interaktionen zwischen biologischen Molekülen gespeichert. Das Hauptaugenmerk liegt hierbei auf experimentell nachgewiesenen Protein-Protein-Interaktionen, wobei hauptsächlich Informationen zu Proteomen von Säugetieren gesammelt werden. Die Informationen über die Interaktionen werden aus der wissenschaftlichen Literatur per Hand extrahiert. Der aktuelle Datenbankzustand ist in Tab. 5.3 zu sehen.

Anzahl an Interaktionen in der MINT Datenbank	
Anzahl an Interaktionen mit zwei Partnern aus Säugetieren	6644
Anzahl an Interaktionen mit zwei Partnern aus Caenorhabditis elegans	4499
Anzahl an Interaktionen mit zwei Partnern aus Drosophila melanogaster	20651
Anzahl an Interaktionen mit zwei Partnern aus Hefe	13232
Gesamtzahl an Interaktionen	51957

Tab. 5.3 MINT Datenbankzustand vom 21.8.2005 in Bezug auf Interaktionen [http://mint.bio.uniroma2.it/mint/statistics/statistics.php]

Es gibt zwei verschiedene Dateiformate: ein reines Textformat (flat file) und ein gemeinsam mit der DIP [SAL04] und der BIND [BAD03] entwickeltes XML Format (PSI MI [http://psidev.sourceforge.net/mi/xml/doc/user/]).

5.1.4. STRING: Search Tool for the Retrieval of Interacting Genes/Proteins

Die STRING-Datenbank diente am Anfang ihrer Entstehung dem Zweck, aufgrund von Nachbarschaften von Genen auf ihre Funktion zu schließen. Deswegen stand STRING auch zuerst für: *Search Tool for Recurring Instances of Neighbouring Genes* [SNE00]. Mittlerweile bietet die Datenbank Informationen über interagierende Proteine an und STRING steht daher für *Search Tool for the Retrieval of Interacting Genes/Proteins*.
Die STRING-Datenbank verwendet für die Extraktion von Protein-Protein-Interaktionsinformationen verschiedene Quellen. Es werden Protein-Protein-Interaktionen aus der Literatur gewonnen und von anderen Datenbanken importiert. Außerdem werden Informationen aus der Analyse von Microarray-Daten gewonnen. Bei dieser Analyse wird untersucht, welche Gene co-reguliert sind. Dazu werden Expressionsdaten unter diversen Bedingungen ausgewertet. Diese Daten werden vom ArrayProspector Server [JEN04] importiert. Schließlich werden auch Informationen über Protein-Protein-Interaktionen durch systematischen Genomvergleich gewonnen. Dabei werden die Phylogenetic Profiles - Methode (siehe 3.1), die Conservation of Gene Neighborhood - Methode (siehe 3.2) und die Gene Fusion - Methode (siehe 3.3) verwendet.
Von der COG Datenbank [TAT03] (Clusters of Orthologous Genes) wurden Cluster importiert, die aus Proteinen mit ihren zugehörigen homologen Proteinen bestehen, also aus Paralogen des eigenen Organismus und Orthologen aus einem anderen

Organismus. Diese Cluster wurden während der Erstellung des Datensatzes (siehe 5.2) verwendet, um die Orthologen eines Proteins zu erhalten.

5.2. Datengenerierung

5.2.1. Datengenerierung durch Integration aller Protein-Protein-Interaktionsinformationen aus der DIP, BIND und MINT-Datenbank

Das Ziel der Datengenerierung war es, einen Datensatz für Escherichia coli zu generieren, der alle Interaktionen, die in der DIP (siehe 5.1.1), BIND (siehe 5.1.2) und MINT Datenbank (siehe 5.1.3) vorhanden sind, enthält. Die Interaktionen werden in einer Interaktionsmatrix gespeichert. Zusätzlich werden die entsprechenden Proteinsequenzen von Escherichia coli, über Referenzen der Interaktionsdatenbanken, aus der Proteinsequenzdatenbank UniProt und aus Flatfiles extrahiert, die von den Datenbanken angeboten wurden. Dies ist in Fig. 5.1 zu sehen. Zusätzlich sollten für jedes dieser Proteine alle orthologen Sequenzen (siehe 2.5) aus der STRING-Datenbank (siehe 5.1.4) extrahiert werden. Existieren zu einem Protein von Escherichia coli mehrere homologe Proteine in einem anderen Organismus, so wird das Protein mit größter Ähnlichkeit ausgewählt. Dies wurde unter anderem auch beim *Mirrortree* Verfahren (siehe 3.4) so durchgeführt und deckt sich mit der biologischen Tatsache, dass Proteine, die eine sehr ähnliche Sequenz aufweisen auch oft die gleiche Funktion erfüllen.

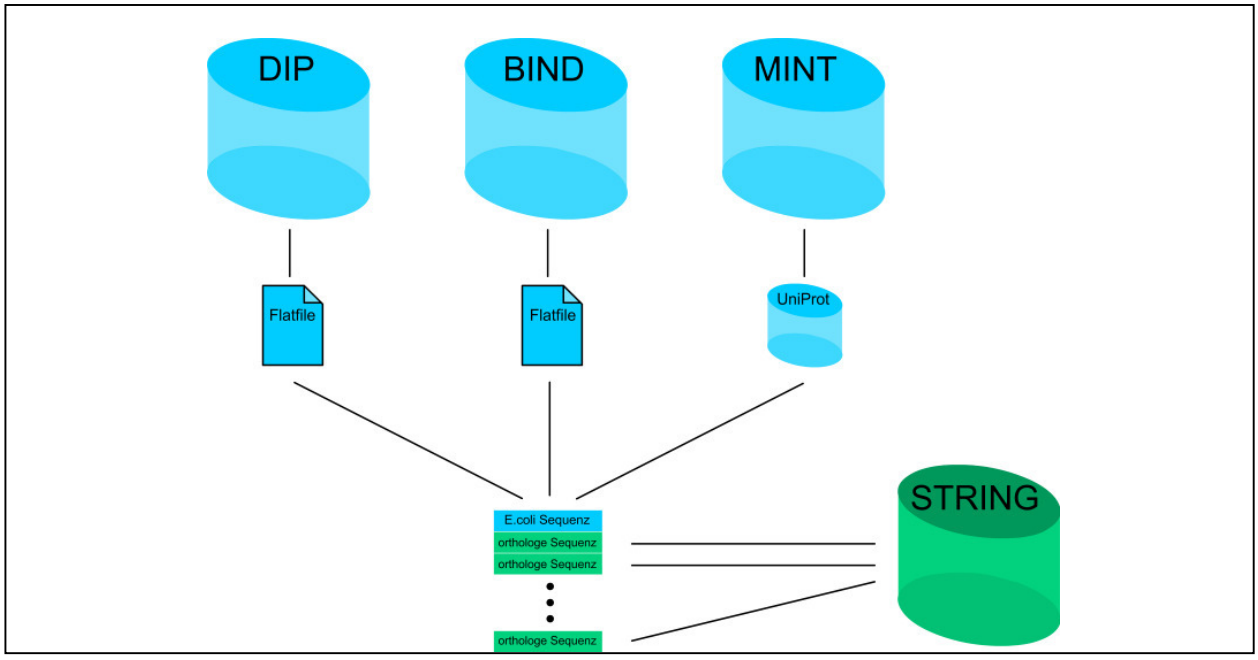


Fig. 5.1 Generierung der Proteinsequenzcluster für Escherichia coli

Die Generierung der Interaktionsmatrix ist in Fig. 5.2 dargestellt. Jedes Protein aus Escherichia Coli, für das eine Interaktion in einer der drei Datenbanken vorhanden ist, bekommt eine eindeutige Clusternummer zugewiesen. Nun wird mit Hilfe aller Interaktionsdaten eine Interaktionsmatrix erstellt, die an der Stelle (i, j), bzw. (j, i) eine eins enthält, wenn für Protein i mit Protein j eine Interaktion bekannt ist und eine null andernfalls.

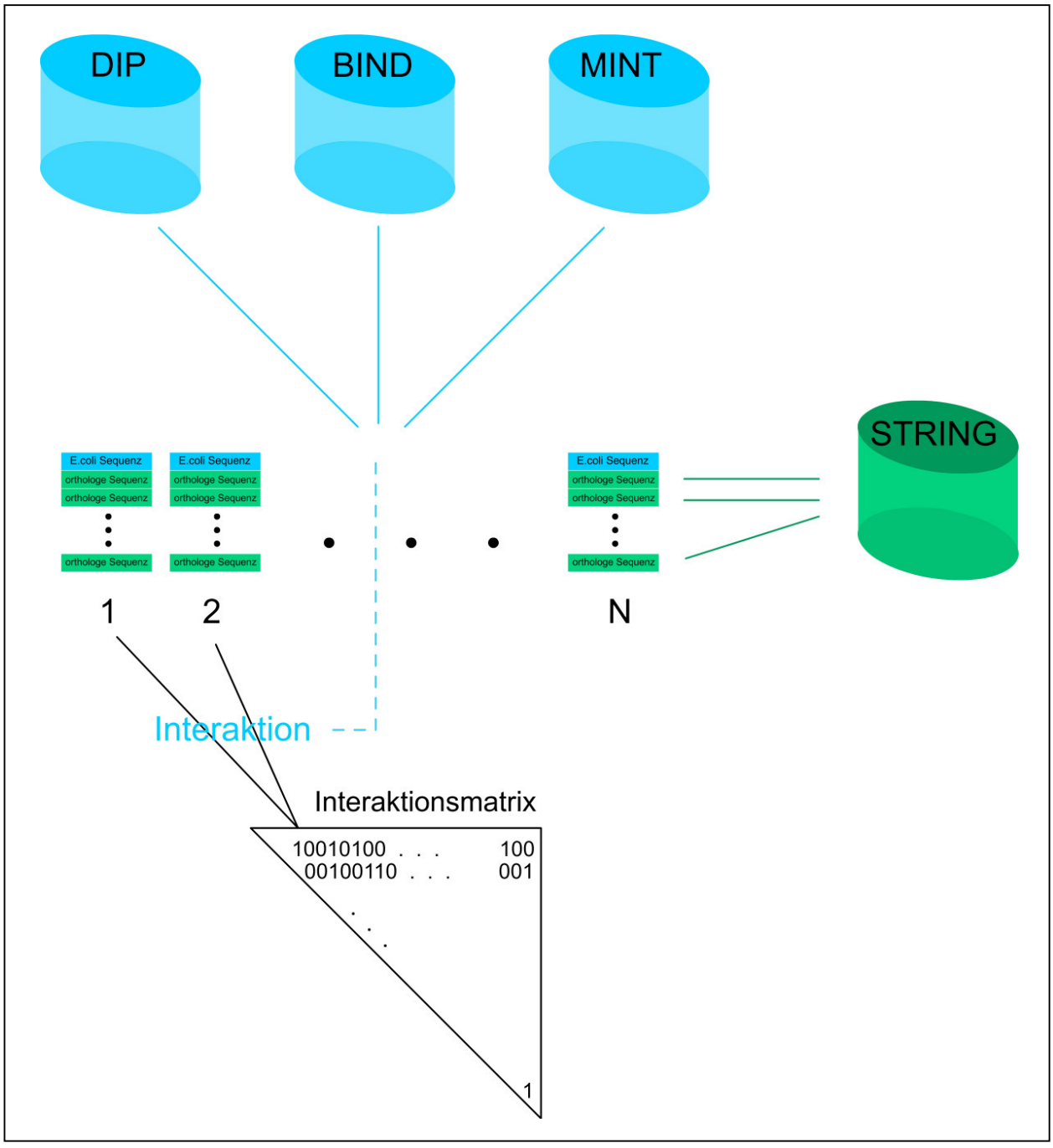


Fig. 5.2 Generierung der Interaktionsmatrix

Für jedes Proteincluster wird danach eine Distanzmatrix D erstellt. Sie gibt an, wie weit die Sequenzen des Clusters, in Bezug auf die Sequenzähnlichkeit, voneinander entfernt sind. D(i,j) gibt folglich die Distanz zwischen dem Protein i und dem Protein j des

Clusters an. Um die paarweisen Distanzen zu berechnen wird der ClustalW-Algorithmus verwendet [HIG92]. Da die Distanzen symmetrisch sind, ist es ausreichend, die obere Dreiecksmatrix von D zu berechnen. Die Generierung einer Distanzmatrix ist in Fig. 5.3 zu sehen.

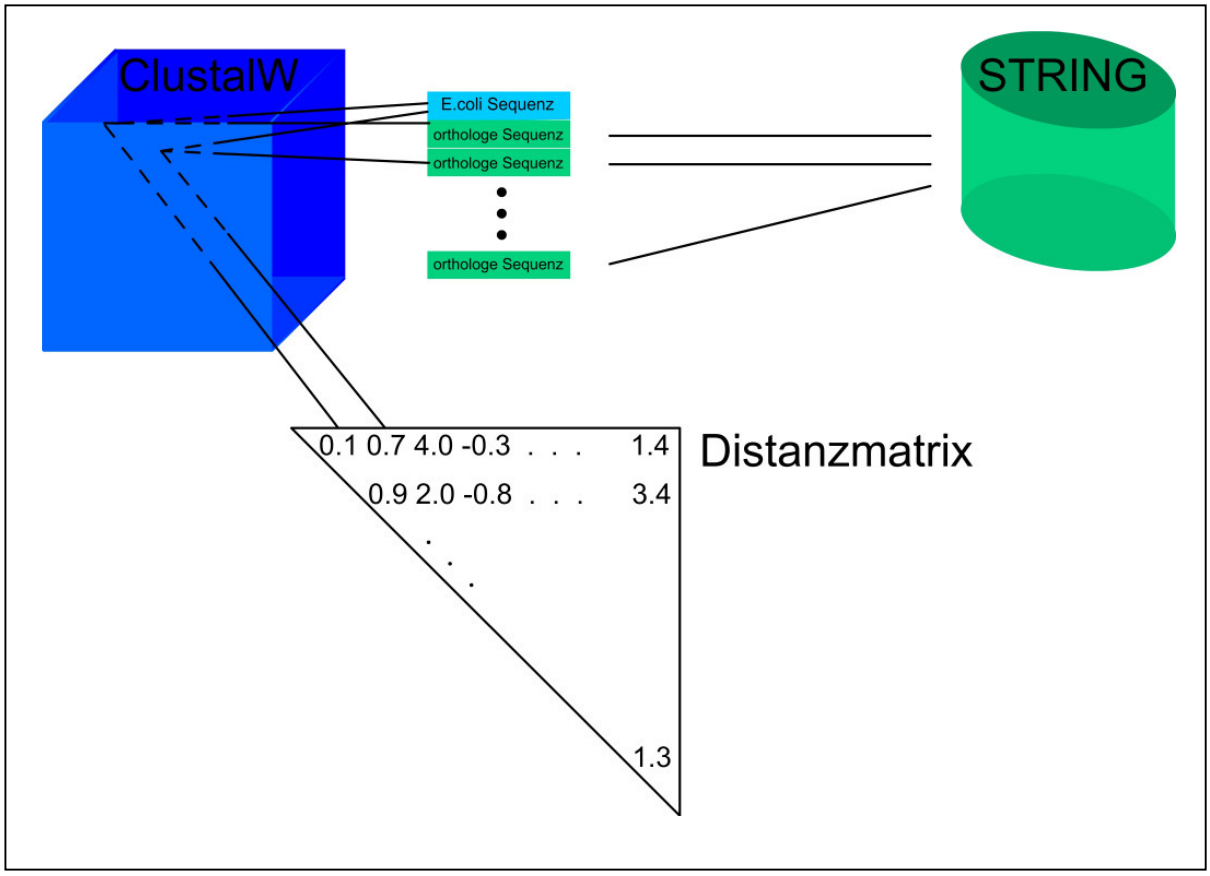


Fig. 5.3 Generierung einer Distanzmatrix: Die Sequenzen werden paarweise per ClustalW alignt und ihre Distanz wird berechnet.

Mit dieser generellen Vorgehensweise wurden zwei unterschiedliche Datensätze erstellt. Bei dem einen Datensatz ist die Bedingung, dass für jedes verwendete Protein aus Escherichia coli für jeden Organismus, aus einer vorher festgelegten Menge von Organismen, ein orthologes Protein aus der STRING-Datenbank extrahiert werden kann. Erfüllt ein Protein von Escherichia Coli diese Bedingung nicht, so wird der entsprechende Proteincluster entfernt. Für die Wahl der Organismen wurden die Organismen gewählt, für die die meisten orthologen Proteine zu Escherichia Coli in der STRING-Datenbank existieren, wobei jeweils immer nur ein Stamm ausgewählt wurde, falls es zu einem Organismus mehrere Stämme gab (z.B. Bacillus subtilis, Bacillus anthracis). Der Datensatz wird im Folgenden auch als *DBM1*-Datensatz bezeichnet. Die ausgewählten Organismen sind in Tab. 5.4 aufgeführt.

Spezies_id	Name
181661	Agrobacterium tumefaciens (Cereon)
1423	Bacillus subtilis
375	Bradyrhizobium japonicum
29461	Brucella suis
155892	Caulobacter crescentus
727	Haemophilus influenzae
1642	Listeria innocua
381	Mesorhizobium loti
182710	Oceanobacillus iheyensis
747	Pasteurella multocida
287	Pseudomonas aeruginosa
305	Ralstonia solanacearum
602	Salmonella typhimurium LT2
70863	Shewanella oneidensis
623	Shigella flexneri
382	Sinorhizobium meliloti
672	Vibrio vulnificus
92829	Xanthomonas axonopodis
2371	Xylella fastidiosa
632	Yersinia pestis CO92

Tab. 5.4 Verwendete Organismen für die Bestimmung homologer Proteine für den *DBM1*-Datensatz

Bei dem anderen Datensatz ist die Bedingung etwas schwächer formuliert. Hier müssen für jedes mögliche Proteinpaar jeweils orthologe Proteine aus mindestens zehn gleichen Organismen existieren (Bsp. Interaktion zwischen Protein A und Protein B: orthologes Protein zu Protein A in Organismus 1, orthologes Protein zu Protein A in Organismus 5,..., orthologes Protein zu Protein A in Organismus 41 - orthologes Protein zu Protein B in Organismus 1, orthologes Protein zu Protein B in Organismus 5,..., orthologes Protein zu Protein B in Organismus 41). Erfüllt ein Protein diese Bedingung nicht, wird der ganze zugehörige Proteincluster mit allen involvierten Interaktionen entfernt. Im Folgenden wird dieser Datensatz auch als *DBM2*-Datensatz bezeichnet. Die 50 möglichen bakteriellen Organismen sind in Tab. 5.5 aufgeführt. Auch bei diesem Datensatz wurde pro Organismus nur ein Stamm verwendet.

Spezies_id	Name
180835	Agrobacterium tumefaciens (Wash.)
63363	Aquifex aeolicus
1423	Bacillus subtilis
1679	Bifidobacterium longum
139	Borrelia burgdorferi
375	Bradyrhizobium japonicum
29459	Brucella melitensis
98794	Buchnera aphidicola (Schiz.g.)
197	Campylobacter jejuni
155892	Caulobacter crescentus
813	Chlamydia trachomatis
115711	Chlamydophila pneumoniae AR39
1097	Chlorobium tepidum
1488	Clostridium acetobutylicum
1718	Corynebacterium glutamicum
1299	Deinococcus radiodurans
76856	Fusobacterium nucleatum
727	Haemophilus influenzae
85962	Helicobacter pylori 26695
1360	Lactococcus lactis
173	Leptospira interrogans
1639	Listeria monocytogenes
381	Mesorhizobium loti
1769	Mycobacterium leprae
2097	Mycoplasma genitalium
491	Neisseria meningitidis B
103690	Nostoc sp. PCC 7120
182710	Oceanobacillus iheyensis
747	Pasteurella multocida
287	Pseudomonas aeruginosa
305	Ralstonia solanacearum
781	Rickettsia conorii
601	Salmonella typhi CT18

70863	Shewanella oneidensis
623	Shigella flexneri
382	Sinorhizobium meliloti
1282	Staphylococcus epidermidis
1309	Streptococcus mutans
1902	Streptomyces coelicolor
1148	Synechocystis sp. PCC 6803
119072	Thermoanaerobacter tengcongensis
2336	Thermotoga maritima
160	Treponema pallidum
218496	Tropheryma whipplei TW0827
134821	Ureaplasma parvum
666	Vibrio cholerae
164609	Wigglesworthia brevipalpis
340	Xanthomonas campestris
2371	Xylella fastidiosa
632	Yersinia pestis CO92

Tab. 5.5 Verwendete Organismen für die Bestimmung orthologer Proteine für den *DBM2*-Datensatz

Dadurch, dass die Anforderungen an die Proteine des ersten Datensatzes höher sind, wurden mehr Proteincluster entfernt als bei dem zweiten Datensatz. Da für beide Datensätze die Informationen der DIP, BIND, und MINT-Datenbank verwendet werden, wird der erste Datensatz mit *DBM1* und der Zweite mit *DBM2*.

Der *DBM1*-Datensatz enthält 85 Proteine, die an 90 Interaktionen beteiligt sind und der *DBM2*-Datensatz enthält 150 Proteine, die an 148 Interaktionen beteiligt sind.
Um ein realistischeres Szenario zu kreieren wurden bei beiden Datensätzen zusätzliche Proteine mit ihren zugehörigen Orthologen hinzugefügt, die Escherichia Coli – Sequenzen entsprechen, für die keine Interaktionen mit anderen Proteinen des Datensatzes bekannt sind. Für diese Proteine gelten die gleichen Kriterien, wie für die Positivbeispiele, nur dass die Escherichia Coli – Proteinsequenzen direkt aus der STRING-Datenbank extrahiert wurden.

Insgesamt beinhaltet der *DBM1*-Datensatz 925 Proteine und 90 Interaktionen und der *DBM2*-Datensatz 1010 Proteine und 148 Interaktionen. Für den DBM2-Datensatz hätten

auch noch mehr Negativbeispiele hinzugefügt werden können, allerdings wurde darauf verzichtet, um die beiden Datensätze und die unterschiedlichen Erstellungsmethoden besser vergleichen zu können.

5.2.2. Datengenerierung durch Protein-Protein-Interaktionsinformationen aus der STRING-Datenbank

Ein weiterer Datensatz für Interaktionen zwischen Escherichia coli Proteinen, der lediglich mit Hilfe der STRING-Datenbank (siehe 5.1.4) erstellt wurde, wurde ähnlich zu 5.2.1 generiert. Der einzige Unterschied ist, dass die Information über die Protein-Protein-Interaktionen in diesem Fall nicht aus der DIP (siehe 5.1.1), BIND (siehe 5.1.2) oder MINT Datenbank (siehe 5.1.3) stammt, sondern aus der STRING Datenbank. Wie in 5.1.4 beschrieben befinden sich in dieser Datenbank auch Informationen über Protein-Protein-Interaktionen, die aus der Literatur extrahiert wurden. Zur jeweiligen Gewichtung der Evidenz der Interaktion befindet sich in der Datenbank für jede dieser Interaktionen ein Textmining-Score.
Zur Generierung des Datensatzes wurden nur die Interaktionen verwendet, die den höchsten Textmining-Score in der STRING-Datenbank besitzen. Die Proteinclustergenerierung ist in Fig. 5.4 zu sehen.

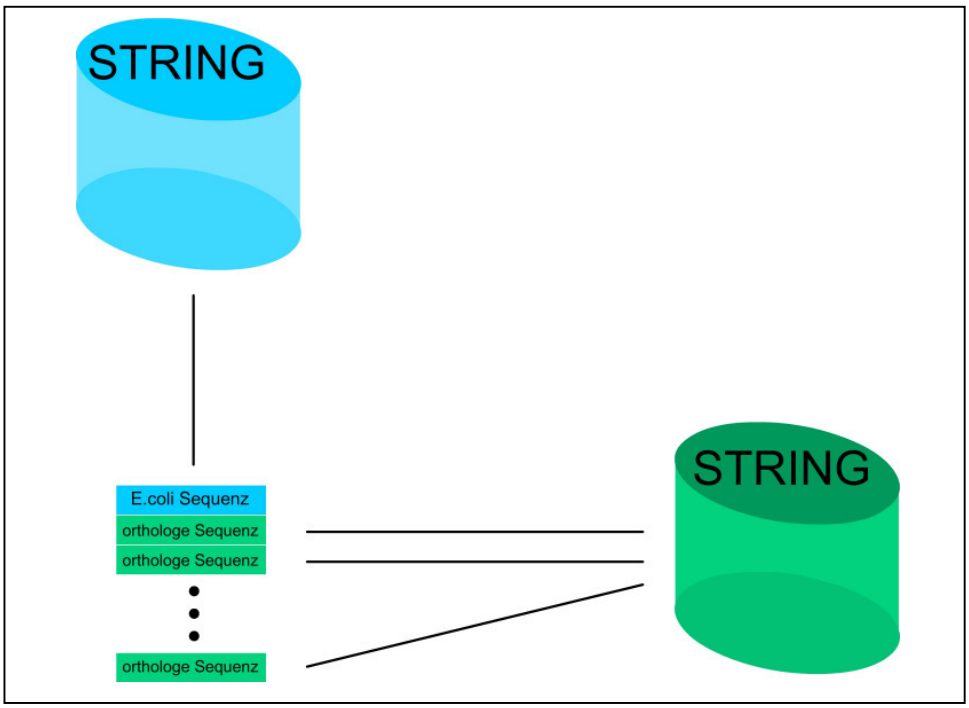


Fig.5.4 Generierung der Proteincluster

Für jedes Protein, das in einer Interaktion vorkommt, die den höchsten Textmining-Score besitzt wird ein Proteincluster erzeugt. Dieser enthält die Proteinsequenz mit allen homologen Sequenzen, die unter den bakteriellen Organismen der STRING-Datenbank zu finden sind (siehe Tab.5.5). Die Distanzmatrizen wurden analog zu den anderen beiden Datensätzen erstellt (siehe 5.2.1) und die Erstellung der Interaktionsmatrix ist in Fig. 5.5 zu sehen.
Da für diesen Datensatz lediglich Protein-Protein-Interaktions-Informationen aus der STRING-Datenbank verwendet wurden, wird dieser Datensatz im Folgenden auch als *STRING*-Datensatz bezeichnet. Die Anforderung an die einzelnen Proteincluster sind die gleichen wie beim *DBM2*-Datensatz.

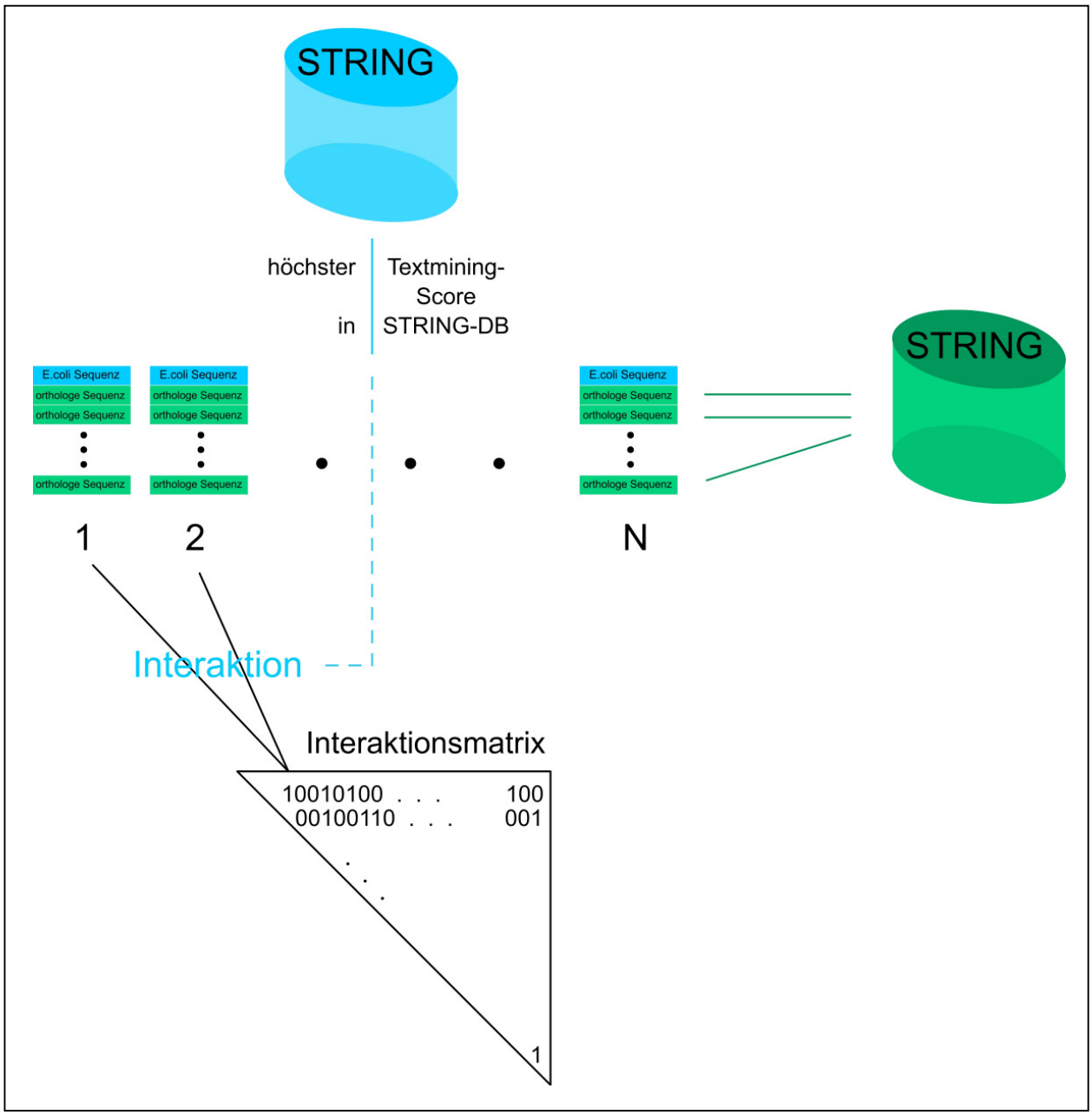


Fig. 5.5 Erstellung der Interaktionsmatrix

Der *STRING*-Datensatz beinhaltet 612 Proteincluster von Proteinen, die an 670 Interaktionen beteiligt sind sowie 1583 weitere Proteincluster von Proteinen, für die keine Interaktionen mit anderen Proteinen des Datensatzes bekannt sind.

6. Anwendung der UKR und SSKR zur Vorhersage von Protein-Protein-Interaktionen

6.1. Repräsentation der Proteine

Für jedes mögliche Proteinpaar des Datensatzes (siehe 5.2) werden der lineare Korrelationskoeffizient r und der P-Value berechnet. Für r_{jk} gilt:

$$r_{jk} = \frac{\sum_{i=1}^{n} (S_{ij} - \bar{S}_j)(S_{ik} - \bar{S}_k)}{\sqrt{\sum_{i=1}^{n} (S_{ij} - \bar{S}_j)^2} \sqrt{\sum_{i=1}^{n} (S_{ik} - \bar{S}_k)^2}}$$

Hierbei ist S die Matrix in der spaltenweise die Distanzmatrizen der Proteine als Vektoren aufgeführt sind. r_{jk} ist der Korrelationskoeffizient zwischen der Distanzmatrix von Protein j und der Distanzmarix von Protein k. Die Anzahl der Elemente in der Distanzmatrix ist gleich n. $\bar{S}_j$ und $\bar{S}_k$ sind die Mittelwerte der jeweiligen Distanzmatrizen. Der P-Value p_{jk} steht für die Wahrscheinlichkeit, dass r_{jk} zufällig diesen Wert annimmt, ohne dass eine tatsächliche Korrelation von j und k vorliegt. Hierzu wird eine t-Statistik verwendet mit n – 2 Freiheitsgraden.
Mit den P-Values wird eine N x N – Matrix erstellt. N ist die Anzahl der verwendeten Proteine des Ausgagnsorganismus.

6.2. Parameter der SSKR zur Vorhersage von Protein-Protein-Interaktionen

Wie in 4.3 erwähnt sind die Basisfunktionen gegeben durch $b_i(\vec{y}; \mathrm{Y}) = \frac{K(\vec{y} - \vec{y}_i)}{\sum_{j=1}^{N} K(\vec{y} - \vec{y}_j)}$.

Dann ist für eine feste Datenmatrix Y die Matrix B der Basisfunktionen gegeben durch $\mathrm{B} = \mathrm{B}(\mathrm{Y}) = [b(\vec{y}_1; \mathrm{Y}), b(\vec{y}_2; \mathrm{Y}), ..., b(\vec{y}_N; \mathrm{Y})]$. Da die Summe aller Basisfunktionen gleich eins ist, können die Basisfunktionen als eine Dichtefunktion aufgefasst werden.

In 6.1 wird erwähnt, dass der P-Value p_{ij} für die Wahrscheinlichkeit steht, dass die Korrelation r_{ij} zwischen der Distanzmatrix von Protein i und der Distanzmatrix von Protein j zufällig diesen Wert annimmt. Der Wert $1-p_{ij}$ kann folglich als ein probabilistisches Wahrscheinlichkeitsmaß dafür aufgefasst werden, dass eine hohe Korrelation zwischen i und j vorhanden ist. Je höher dieser Wert ist, desto höher ist die Wahrscheinlichkeit, dass Protein i und Protein j interagieren. Zur Vorhersage von Protein-Protein-Interaktionen wird folglich aus der Basisfunktion $b_{ij} = \frac{K(\vec{y}_i - \vec{y}_j)}{\sum_{l=1}^{N} K(\vec{y}_l - \vec{y}_j)}$ die Basisfunktion $b_{ij} = \frac{1-p_{ij}}{\sum_{l=1}^{N}(1-p_{lj})}$.

Wie in 3.4 erläutert, ist die Messung der Korrelation zwischen den Distanzmatrizen der Proteine die Grundlage des Mirrortee-Verfahrens. Da die Verwendung von $b_{ij} = \frac{1-p_{ij}}{\sum_{l=1}^{N}(1-p_{lj})}$ sehr ähnlich zur Messung der Korrelation ist, kann das *SSKR*-Verfahren zur Vorhersage von Protein-Protein-Interaktionen als eine Erweiterung des *Mirrortree*-Verfahrens angesehen werden.

In 4.3 wurde dargestellt, dass die Eigenvektoren zur Matrix $\left(SS^T - S\tilde{B}^T - \tilde{B}S^T + \tilde{B}\tilde{B}^T\right)$ den Ausdruck $\frac{1}{N} tr\left(\tilde{X}\left[SS^T - S\tilde{B}^T - \tilde{B}S^T + \tilde{B}\tilde{B}^T\right]\tilde{X}^T\right)$ minimieren. Weiter wurde ausgeführt, dass die Anzahl der Dimensionen der latenten Repräsentation der Daten durch die Anzahl der verwendeten Eigenvektoren (im Folgenden als q bezeichnet) gewählt werden kann. Um jegliche Information zu nutzen, wurden in dieser Arbeit alle Eigenvektoren verwendet, die $\left(SS^T - S\tilde{B}^T - \tilde{B}S^T + \tilde{B}\tilde{B}^T\right)$ auf nicht triviale Weise minimieren, also alle Eigenvektoren außer dem Eigenvektor zum kleinsten Eigenwert. Sei n die Anzahl der Zeilen von $\tilde{X}$, dann ist q = n – 1.

6.3. Generierung der Trainings- und Testmenge

Da Proteine im Datensatz existieren, die mehrere Interaktionen besitzen, gibt es kleine Interaktionsnetzwerke, bei denen mehrere Proteine über eine Reihe von Interaktionen verbunden sind. Seien die Proteine die Knoten eines Graphen und die Interaktionen die Kanten. Dann lässt sich eine Zusammenhangskomponente des Datensatzes wie in Fig. 6.1 darstellen.

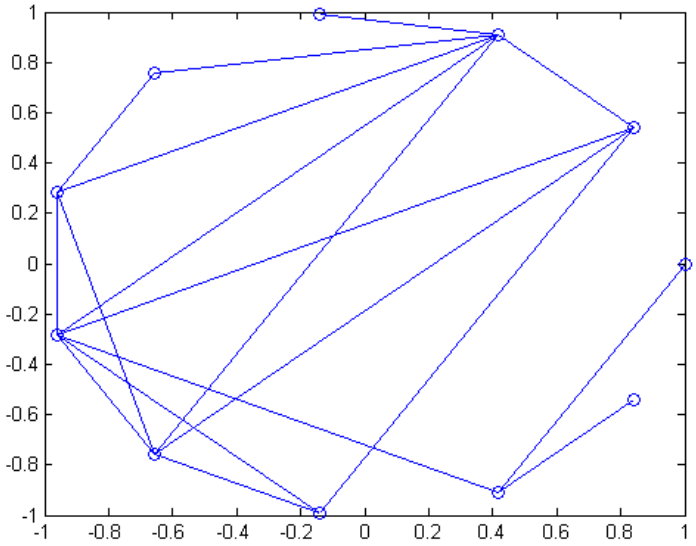


Fig. 6.1 Darstellung eines Interaktionsnetzwerks: Die Proteine sind als Knoten dargestellt und die bekannten Interaktionen als Kanten

Sollen bestimmte Interaktionen zum Training verwendet und auf dem Rest der Interaktionen die Performanz des Verfahrens getestet werden, so muss auf jeden Fall vermieden werden, dass triviale Eigenschaften gelernt werden. Ein Beispiel für das Lernen einer trivialen Eigenschaft wäre, wenn die Kanten (i, j) und (j, k) im Trainingsset und Kante (i, k) im Testset wären. Das Testset wird daher mit folgenden Algorithmen aufgebaut.

Random Tree – Algorithmus

Eingabe: Ein zusammenhängender Graph G mit Kanten $(i, j) \in E$ und Knoten $V = \{1, 2, ..., n\}$

Ausgabe: Eine Menge E_{Tree}, die alle Baumkanten des Zufallsbaums enthält

1. Füge alle Knoten aus V der Menge V_{unused} hinzu und initialisiere die Mengen V_{used}, E_{tree} und E_{temp} mit der leeren Menge.
2. Ziehe zufällig einen Knoten v aus der Menge V_{unused}, entferne v aus V_{unused} und füge v der Menge V_{used} hinzu.
3. Weise E_{temp} alle Kanten von v zu.
4. Entferne aus E_{temp} alle Kanten (v,k) , bzw. (k,v) für die gilt, dass $k \in V_{used}$

5. Falls E_{temp} leer ist, so gehe zu 2.,
   andernfalls ziehe zufällig eine Kante (v,k) , bzw. (k,v) aus E_{temp} und füge sie der Menge E_{tree} hinzu. Entferne k aus V_{unused} und füge ihn V_{used} hinzu.
6. Falls E_{tree} weniger als n – 1 Kanten enthält, gehe zu 2.,
   andernfalls ist der Algorithmus beendet.

E_{Tree} enthält nach Beendigung des Algorithmus alle Kanten des Zufallsbaums. Ein möglicher Baum ist in Fig. 6.2 und Fig. 6.3 abgebildet, wobei hier die Baumkanten rot dargestellt sind.

Component Partitioning – Algorithmus

Eingabe: Ein zusammenhängender Graph G mit Kanten $(i, j) \in E$

Ausgabe: $E_{Training}$: Die Menge der Kanten, die zum Training verwendet werden kann,
E_{Test} : Die Menge der Kanten, die zum Testen verwendet werden kann

1. Initialisiere $E_{Training}$ und E_{Test} mit der leeren Menge
2. Finde einen Zufallsbaum mit Hilfe des *Random Tree – Algorithmus* im Interaktionsnetzwerk und weise alle seine Kanten der Menge M zu.
3. Solange M nicht leer ist, gehe zu 4, andernfalls gehe zu 7
4. Suche zufällig eine Kante (i, j) aus M aus und füge sie $E_{Training}$ hinzu.
5. Entferne alle Kanten aus M, die i oder j beinhalten und weise sie E_{Test} zu.
6. Gehe zu 3.
7. Ende.

Die Trainingsinteraktionen befinden sich in der Menge $E_{Training}$ und die Testinteraktionen in der Menge E_{Test}. Die ausgewählten Trainingskanten eines Zufallslaufs auf der vorher dargestellten Zusammenhangskomponente sind in Fig. 6.4 zu sehen.

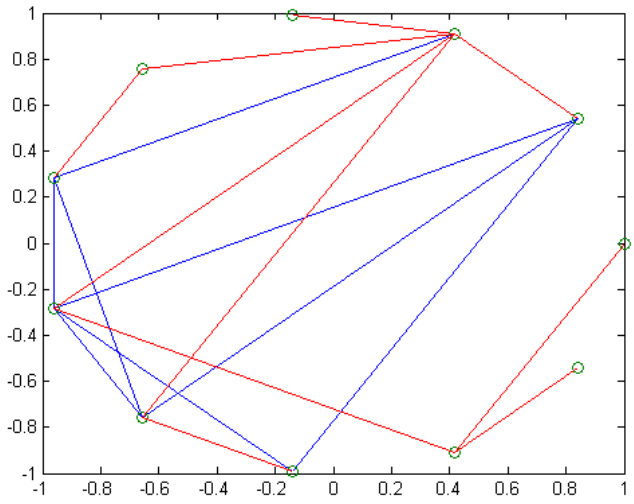


Fig. 6.2 Ein zufällig generierter Baum in dem Interaktionsnetzwerk (Baumkanten in rot)

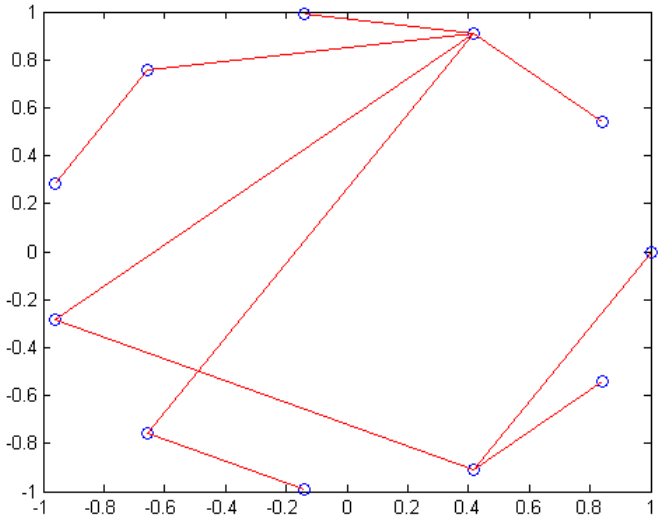


Fig. 6.3 Ein zufällig generierter Baum

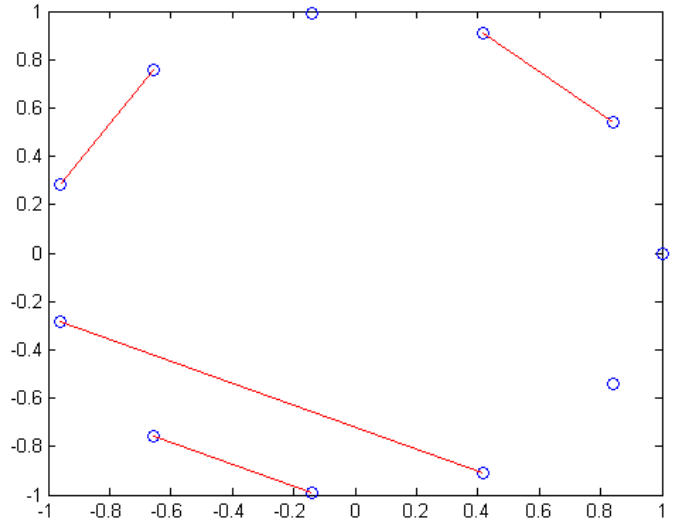


Fig. 6.4 Zufällig aus dem Baum ausgewählte Kanten

Für den *DBM1*-Datensatz (siehe 5.2.1) gibt es 9, für den *DBM2*-Datensatz 16 und für den *STRING*-Datensatz 67 Komponenten, für die mit dem *Component Partitioning – Algorithmus* Trainingsinteraktionen ausgewählt werden. Für Komponenten mit zwei Proteinen entscheidet ein Zufallstest mit 40 Prozent Erfolgswahrscheinlichkeit darüber, ob die jeweilige Interaktion in die Trainingsmenge aufgenommen werden soll oder nicht. Die Schwelle von 40 Prozent wurde gewählt, da durch empirische Tests herausgefunden wurde, dass auf den Datensätzen ca. 40 Prozent der Kanten einer Zusammenhangskomponente mit mehr als drei Knoten als Trainingskanten ausgewählt werden.

Das Erstellen der Trainingspartition kann folgendermaßen zusammengefasst werden:

1. Bestimme für jede Komponenten mit mehr als zwei Proteinen mit Hilfe des *Component Partitioning* Algorithmus eine zufällige Auswahl von Trainings- und Testinteraktionen ($E_{Training}$ und E_{Test})
2. Ziehe für jede Komponente mit zwei Proteinen eine gleichverteilte Zufallsvariable aus dem Intervall [0,1]. Ist die Zufallszahl kleiner als 0,4, so füge die Interaktion zwischen den beiden Proteinen der Menge $E_{Training}$ hinzu.
   Ist die Zufallszahl größer oder gleich 0,4 so füge die Interaktion der Menge E_{Test} hinzu.

6.4. Ergebnisse der SSKR zur Vorhersage von Protein-Protein-Interaktionen

6.4.1. Auswertungsschema

Ein mögliches Anwendungsszenario für die Vorhersage von Protein-Protein-Interaktionen wäre zum Beispiel, dass zu einem bestimmten Protein mögliche Interaktionspartner vorhergesagt werden sollen.
Um die Performanz der *SSKR* (siehe 4.3) im Vergleich zum *Mirrortree*-Verfahren (siehe 3.4) in Bezug auf diese Aufgabe zu messen, wird folgendes Szenario verwendet.

Für jede Interaktion (i, j), die sich in der Menge E_{Test} befindet (siehe 6.3), wird untersucht, wie viele Proteine im latenten Datenraum näher am Protein i sind als das Protein j, für die aber keine Interaktion mit dem Protein i bekannt ist. Je weniger Proteine dieses Kriterium erfüllen, umso besser funktioniert das Verfahren. Da die vorgestellte Relation nicht invariant bezüglich der Wahl des Proteins ist, wird die gleiche Auswertung auch für Protein j durchgeführt.

Durch die hohe Zahl an potenziellen Interaktionspartnern (z.B. 925 bei dem kleinsten verwendeten Datensatz) ist nicht zu erwarten, dass der echte Interaktionspartner der nächste Nachbar ist. In dem Anwendungsszenario wäre es aber zum Beispiel auch schon ein Erfolg, wenn mit der Methode mehrere potenzielle Interaktionspartner vorhergesagt werden könnten, die mit relativ hoher Wahrscheinlichkeit echte Interaktionspartner sind. Das Nachweisexperiment wäre viel billiger, wenn man sich zum Beispiel auf 40 potenzielle Interaktionspartner beschränken könnte, anstatt, wie beim STRING-Datensatz (siehe 5.2.2), 2194 Proteine testen zu müssen.

Von daher werden die K-Nachbarschaften eines Proteins im latenten Raum ausgewertet. Hierbei wird für eine Interaktion (i, j) unter einer i-ten K-Nachbarschaft und einer j-ten K-Nachbarschaft unterschieden. Das Protein j befindet sich in einer i-ten K-Nachbarschaft, wenn höchstens K – 1 Proteine näher am Protein i sind als das Protein j. i-te K-Nachbarschaft und j-te K-Nachbarschaft sind nicht zwingend gleich, wie Fig. 6.5 verdeutlicht. Sei das hell-grüne Protein mit i bezeichnet und das rote Protein mit j. Für K gleich drei befindet sich i in einer j-ten K-Nachbarschaft, aber j befindet sich für diese Wahl von K nicht in einer i-ten K-Nachbarschaft.

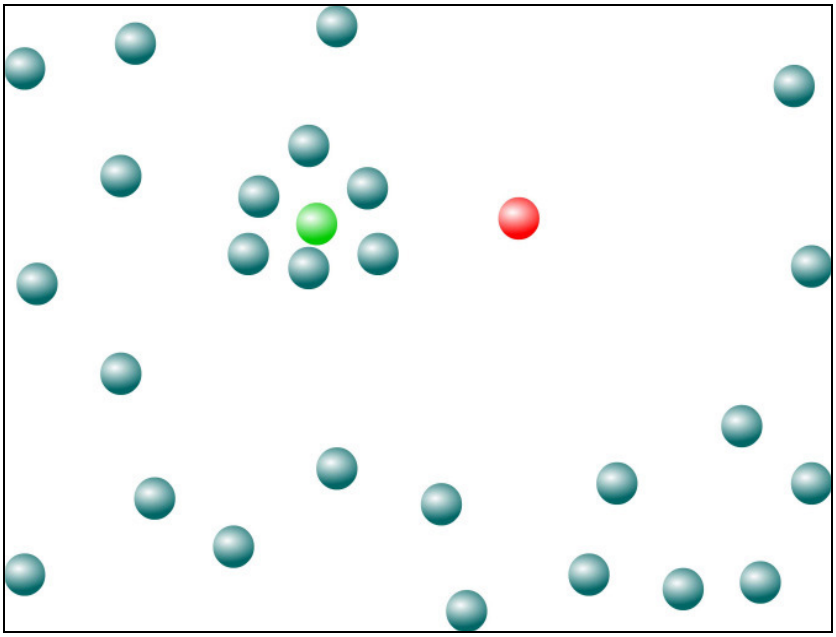


Fig. 6.5 Verdeutlichung der i-ten K-Nachbarschaft

Für die Beispielinteraktion in Fig. 6.6, bei der die Interaktion durch die rote Linie dargestellt ist, ist der echte Interaktionspartner in einer i-ten K-Nachbarschaft für K größer gleich elf (linker Interaktionspartner = i).

Die Auswertung wird für 50 zufällige $E_{Training}$, E_{Test} - Partitionen durchgeführt und der Quotient aus der Anzahl der Erfolge und der gesamten Anzahl der Tests wird gemessen. Hierbei meint *Anzahl der Tests*, die Anzahl der Interaktionen in E_{Test} mal zwei, da, wie bereits beschrieben, für jedes Protein jeder Interaktion (i, j) ein separater Test durchgeführt wird (i-te K-Nachbarschaft und j-te K-Nachbarschaft).

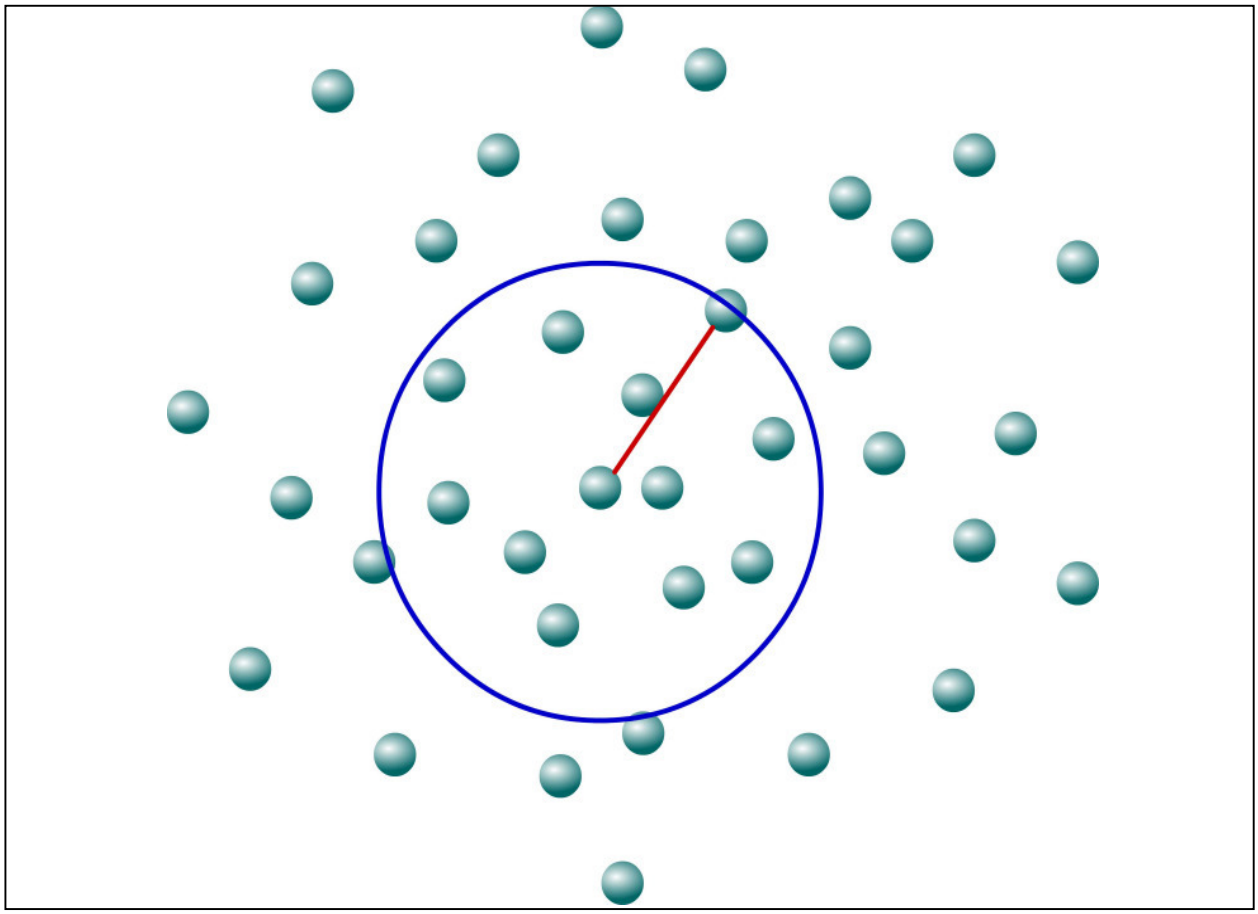


Fig. 6.6 Beispiel für eine i-te K-Nachbarschaft einer Interaktion: die Kreise stellen Proteine dar und die rote Linie symbolisiert eine Interaktion zwischen den beiden verbundenen Proteinen. Der blaue Kreis dient zur besseren Visualisierung der Nachbarschaften. Für K größer gleich 11 ist der rechte Interaktionspartner in einer i-ten K-Nachbarschaft (linker Interaktionspartner = i).

6.4.2. Auswertung für den *DBM1*-Datensatz

Die Ergebnisse nach dem Evaluationsschema (siehe 6.4.1) für den *DBM1*-Datensatz (siehe 5.2.1) sind in Fig. 6.7 dargestellt. Es fällt auf, dass gerade für den Bereich, der für die biologische Anwendung interessant ist (k = 20, 21, ..., 50), das *SSKR*-Verfahren (siehe 4.3) deutlich bessere Ergebnisse liefert als das *Mirrortree*-Verfahren (siehe 3.4). Zusätzlich scheint der positive Effekt, der durch die Einbringung von Wissen über die Interaktionen in der *SSKR* auftritt, am stärksten zu sein für kleinere Nachbarschaften (k kleiner als 30). Dies bedeutet, dass die Erweiterung der *UKR*-Methode gerade für diesen Anwendungsfall sehr sinnvoll ist. Zum Beispiel ist für k = 25 die Rate des *SSKR*-Verfahrens mehr als doppelt so hoch wie die des *Mirrortree*-Verfahrens.

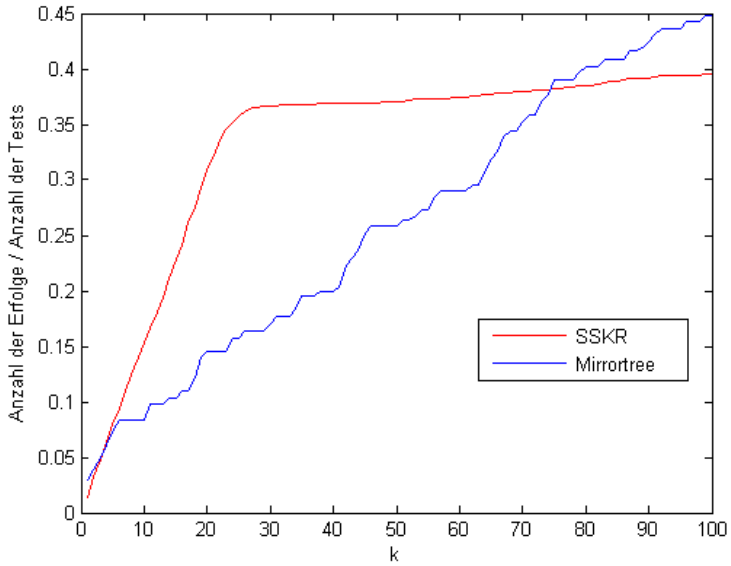


Fig. 6.7 Vergleich der Performanz zwischen der *SSKR* und dem *Mirrortree*-Verfahren (*DBM1*-Datensatz)

Die Auswertung in Fig. 6.8 zeigt, dass das normale *UKR*-Verfahren für bestimmte Datensätze schlechter sein kann, als das *Mirrortree*-Verfahren. Bei diesem Verfahren wurde die *UKR im latenten Raum* (siehe 4.2.4) zur Bestimmung der Repräsentationen der Proteine im latenten Datenraum verwendet. Da bei der Minimierung des Rekonstruktionsfehlers in keiner Weise Wissen über Interaktionen eingeht, ist es möglich, dass in dieser latenten Repräsentation zwar wichtige Merkmale der Proteine repräsentiert werden, sodass der Rekonstruktionsfehler klein ist, aber interagierende Proteine nicht nah zueinander abgebildet werden. Die Ergebnisse dieser Auswertung unterstreichen, dass die Einbringung von zusätzlichem Wissen vorteilhaft für das

maschinelle Lernen sein kann, was besonders im Vergleich der *SSKR* mit der normalen *UKR* deutlich wird.

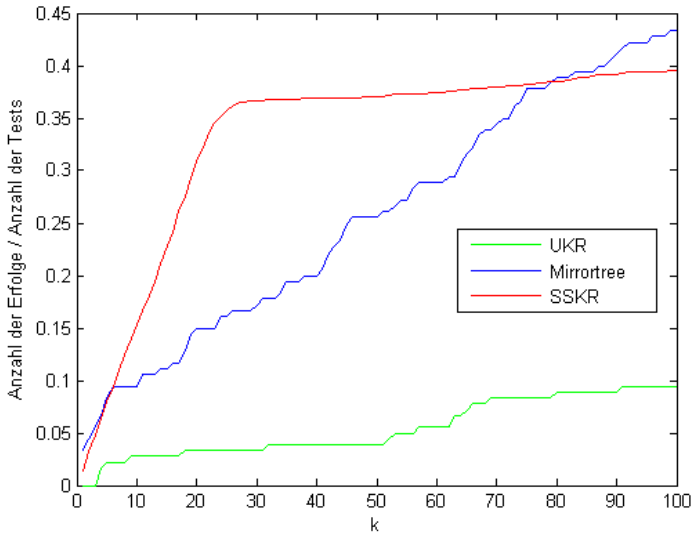


Fig. 6.8 Vergleich der Performanz zwischen *SSKR*, *UKR* und dem *Mirrortree*-Verfahren (*DBM1*-Datensatz)

6.4.3. Auswertung für den *DBM2*-Datensatz

Die Ergebnisse nach dem Evaluationsschema (siehe 6.4.1) für den *DBM2*-Datensatz (siehe 5.2.1) sind in Fig. 6.9 dargestellt. Für diesen Datensatz ist zu konstatieren, dass das *SSKR*- Verfahren (siehe 4.3) grundsätzlich bessere Raten produziert, als das *Mirrortree*-Verfahren (siehe 3.4). Für k größer gleich 20 wird dieser Performanzunterschied am deutlichsten sichtbar. So ist zum Beispiel für k = 40 die Performanz der *SSKR* bei 0,31 und die des *Mirrortree*-Verfahrens bei 0,12.

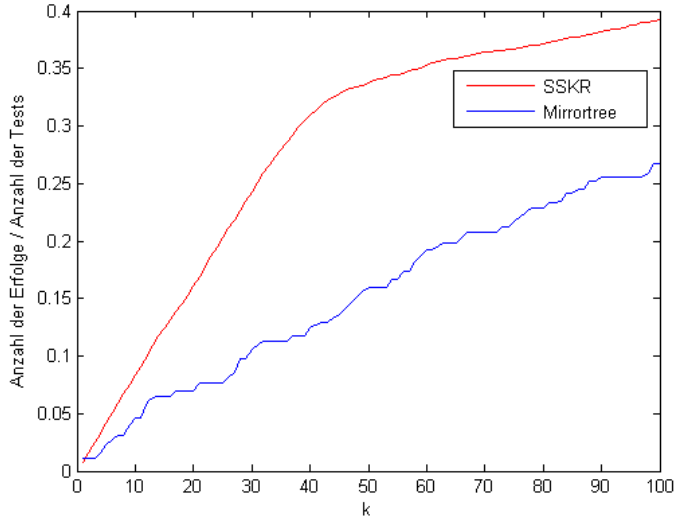


Fig. 6.9 Vergleich der Performanz zwischen *SSKR* und dem *Mirrortree*-Verfahren (*DBM2*-Datensatz)

Beim Vergleich der *UKR im latenten Raum* (siehe 4.2.4) mit dem *Mirrortree*-Verfahren in Fig. 6.10 ist zu beobachten, dass das *UKR*-Verfahren leicht bessere Raten erreicht als das *Mirrortree*-Verfahren. Es ist davon auszugehen, dass die latente Repräsentation der Daten der *UKR im latenten Raum* die Proteine, die interagieren, nah zueinander abbildet und so den minimalen Rekonstruktionsfehler minimiert. Es wird also genau die Eigenschaft der Verteilung der Daten gelernt, die eine Vorhersage von Protein-Protein-Interaktionen ermöglicht.

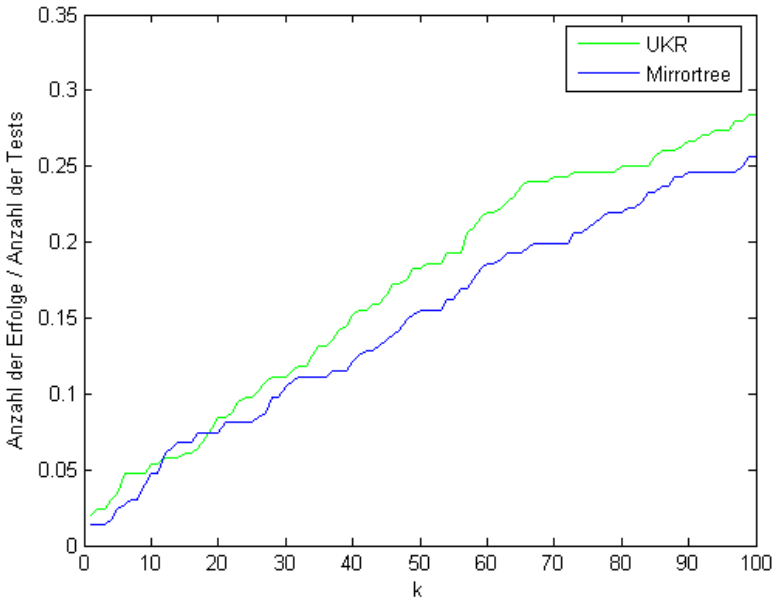


Fig. 6.10 Vergleich der Performanz zwischen *UKR* und dem *Mirrortree*-Verfahren (*DBM2*-Datensatz)

Beim Vergleich der *SSKR* mit den anderen beiden Verfahren in Fig. 6.11 fällt auf, dass der Performanzgewinn der *SSKR* besonders im Bereich von k = 20 bis k = 50 auftritt. Dies ist genau der Bereich, der für eine mögliche Anwendung interessant wäre, da die relativ kleine Anzahl an potenziellen Interaktionspartnern noch relativ kostengünstig im Labor nachgewiesen werden könnte.

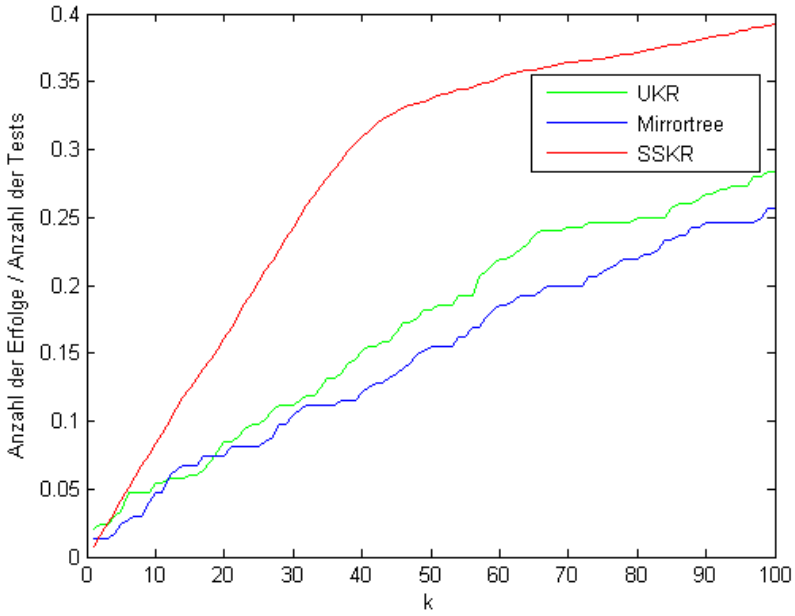


Fig. 6.11 Vergleich der Performanz zwischen *UKR*, *SSKR* und dem *Mirrortree*-Verfahren (*DBM2*-Datensatz)

6.4.4. Auswertung für den *STRING*-Datensatz

Die Ergebnisse nach dem Evaluationsschema (siehe 6.4.1) für den *STRING*-Datensatz (siehe 5.2.2) sind in Fig. 6.12 dargestellt. Grundsätzlich fällt im Vergleich zu der Auswertung des *DBM1*- und des *DBM2*-Datensatzes (siehe 6.4.2, 6.4.3) auf, dass die Performanz auf dem STRING-Datensatz schlechter ist. Dies liegt sehr wahrscheinlich daran, dass dieser Datensatz größer ist, als die anderen beiden Datensätze, da der *DBM1*-Datensatz 925 Proteine enthält, der *DBM2*-Datensatz 1010 und der *STRING*-Datensatz 2195. Dadurch, dass der Datensatz mehr als doppelt so groß ist, gibt es auch doppelt so viele potenzielle Interaktionspartner. Die Raten der Auswertungen sind daher nicht direkt vergleichbar.
Was aber auch bei diesem Datensatz auffällt ist, dass das *SSKR*-Verfahren in dem für eine Anwendung interessanten Bereich von k höhere Raten erreicht als das *Mirrortree*-Verfahren.

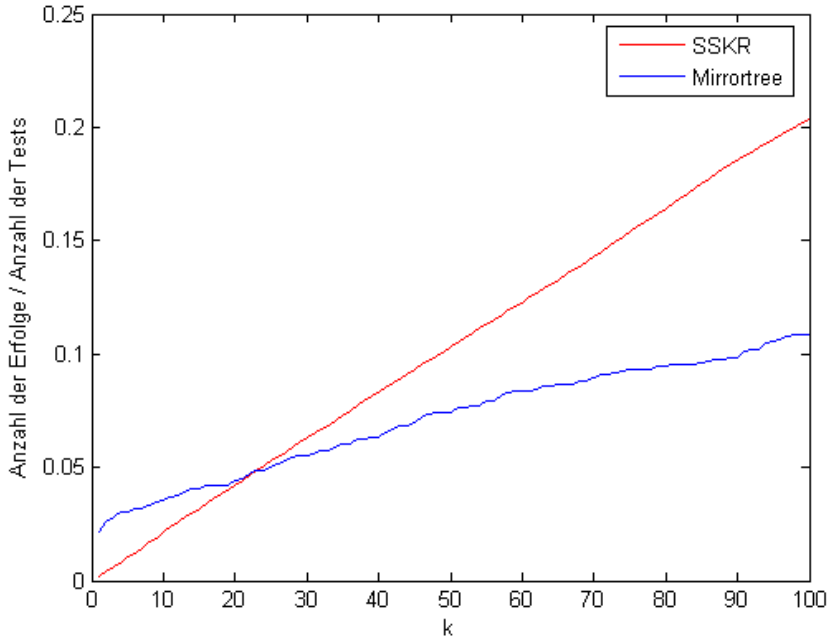


Fig. 6.12 Vergleich der Performanz zwischen *SSKR* und dem *Mirrortree*-Verfahren (*STRING*-Datensatz)

Ähnlich zu 6.4.3 produziert das *UKR*-Verfahren auch beim *STRING*-Datensatz leicht bessere Raten, als das *Mirrortree*-Verfahren, wie in Fig. 6.13 ersichtlich ist. Es ist davon auszugehen, dass auch hier relevante Eigenschaften der Datenverteilung gelernt wurden, die helfen Protein-Protein-Interaktionen vorherzusagen.

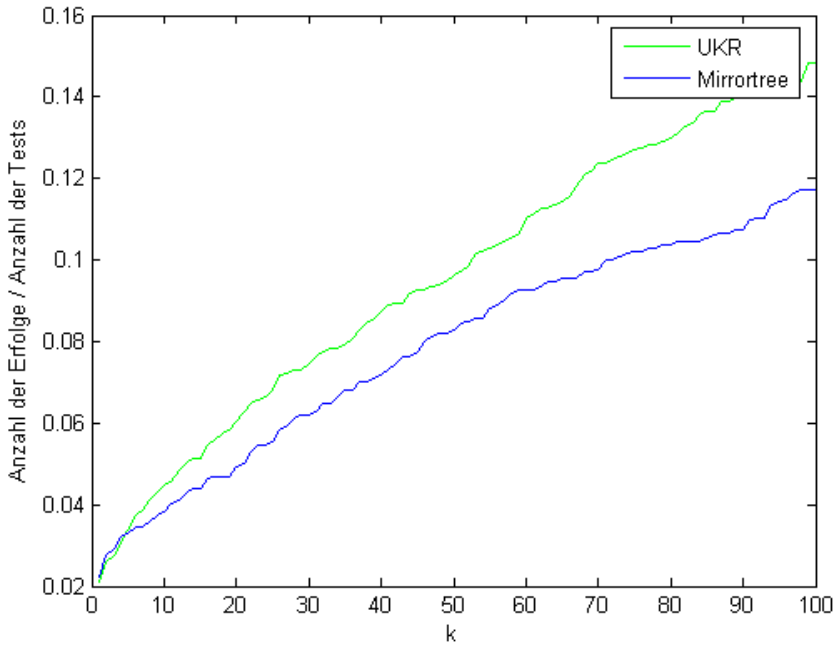


Fig. 6.13 Vergleich der Performanz zwischen *UKR* und dem *Mirrortree*-Verfahren (*STRING*-Datensatz)

Beim Vergleich aller drei verwendeten Methoden (*SSKR*, *UKR*, *Mirrortree*) in Fig. 6.14 fällt auf, dass für k kleiner 43 das *UKR*-Verfahren die besten Raten erreicht. Hierzu ist allerdings anzumerken, dass die Raten aller drei Verfahren in diesem Bereich von k sehr niedrig sind, was wahrscheinlich in der Größe des Datensatzes begründet ist. Für größere k produziert die *SSKR* bessere Raten, als die anderen Verfahren. Dieser Effekt scheint für größere k immer stärker ins Gewicht zu fallen.
Aufgrund der relativ schlechten Raten für kleinere Werte von k, legt die Auswertung nahe, dass für eine Anwendung, bei der für ein Protein potenzielle Interaktionspartner vorhergesagt werden sollen, eine größere Wahl von k notwendig wäre, als bei den anderen beiden Datensätzen. Da aber selbst eine Wahl von k = 100 bedeuten würde, dass nur 100 potenzielle Interaktionspartner im Labor nachgewiesen werden müssten, wäre dies trotzdem ein erheblicher Kosten- und Zeitgewinn, da, ohne biologisches Wissen einzubringen, ansonsten alle 2194 potenziellen Interaktionspartner des Proteins untersucht werden müssten.

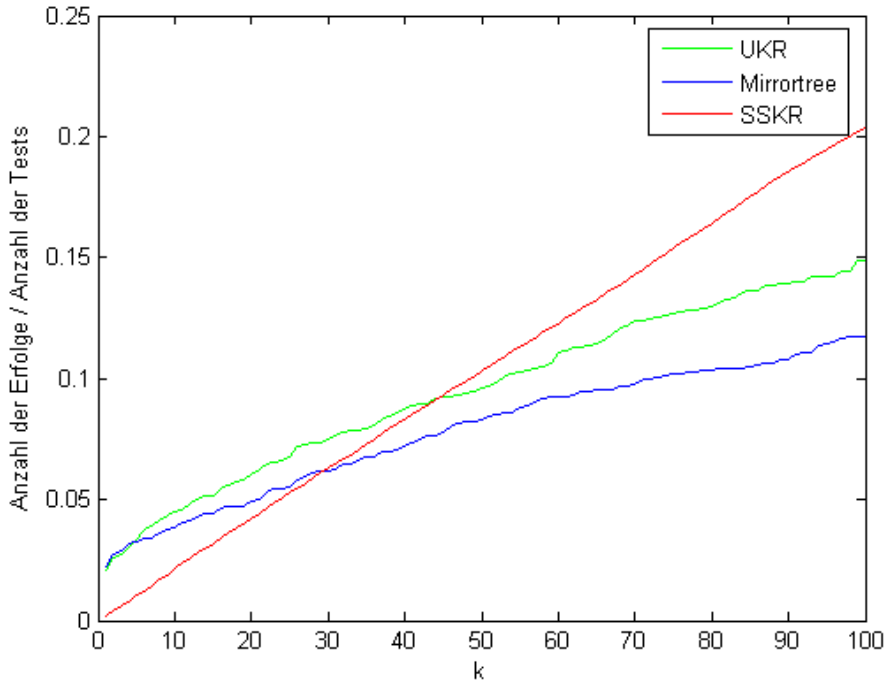


Fig. 6.14 Vergleich der Performanz zwischen *UKR*, *SSKR* und dem *Mirrortree*-Verfahren (*STRING*-Datensatz)

7. Fazit und Ausblick

Die Vorhersage von Protein-Protein-Interaktionen ist nach wie vor eine schwierige Aufgabe und bei weitem noch nicht zufriedenstellend gelöst. Sehr gute Vorhersagen sind vermutlich erst möglich, wenn die 3-D-Strukturvorhersage von Proteinen zufriedenstellend gelöst ist und diese Informationen zur Vorhersage von Interaktionen zwischen den Proteinen verwendet werden können.
Nichtsdestotrotz existieren vielversprechende Verfahren, die aufgrund anderer Informationen als der 3-D-Struktur eines Proteins, hilfreiche Vorhersagen über Protein-Protein-Interaktionen treffen können, wie in Kapitel drei gezeigt wurde. Das *UKR*-Verfahren (siehe 4.2) und insbesondere das *SSKR*-Verfahren (siehe 4.3), welche in dieser Arbeit zum ersten Mal für die Vorhersage von Protein-Protein-Interaktionen verwendet wurden, scheinen für die Problemstellung sehr gut geeignet zu sein, wie die Auswertung in 6.4 zeigt.
Besonders auf den beiden größeren Datensätzen waren die beiden Verfahren besser als das etablierte *Mirrortree*-Verfahren (siehe 3.4). Diese beiden Datensätze sind insbesondere mit Hinblick auf eine Anwendung der Methode zur Vorhersage von Protein-Protein-Interaktionen wichtig. Bei dem dritten, kleineren Datensatz müssen für jedes Protein aus jedem der verwendeten anderen Organismen homologe Proteine existieren. Dies ist gerade mit Hinblick auf eine Anwendung schwierig, da sich die Organismen gerade durch die Unterschiede in ihrer Proteinausstattung unterscheiden. Die Methode bleibt daher auf eine kleinere Teilmenge der Proteine eines Organismus beschränkt. Da für die anderen beiden Datensätze schwächere Restriktionen in Bezug auf die orthologen Proteine angewendet wurden, ist die Zahl an Proteinen eines Organismus deutlich größer, für die die Methode verwendet werden kann und besser funktioniert als das *Mirrortree*-Verfahren.
Die *SSKR* könnte folglich dazu verwendet werden, um für einen Organismus für ein Protein bisher unbekannte Interaktionspartner zu finden. Hierzu würde die Methode eine kleine Menge von Kandidaten vorhersagen, für die dann die Interaktion im Labor überprüft werden müsste. Außerdem könnte die Methode wahrscheinlich verbessert werden, wenn weitere Informationen (siehe Kapitel 3) mit einbezogen würden. Denkbar wäre zum Beispiel eine Verwendung der Positionsinformationen der Gene der Proteine.

Anhang A

A.1. Literaturverzeichnis

[ALB02] Alberts, B, Johnson, A., Lewis, J., Raff, M., Roberts, K. and Walter, P.: **Molecular Biology of the Cell.** *Garland Publishing; 4th edition (March, 2002)*, III.8

[ALT90] Altschul, S.F., Gish, W., Miller, W., Myers, E.W. und Lipman, D.J.: **Basic Local alignment search tool.** *J. Mol. Biol.*, **215**, 403-410

[BAD03] Bader, G.D., Betel, D., Hogue, C.W.: (2003) **BIND: the Biomolecular Interaction Network Database.** *Nucleic Acids Res.* **31**(1):248-50 PMID: 12519993

[BER03] Berg, J.M., Tymoczko, J.L. und Stryer, L. (2003): **Biochemie** *Spektrum Akademischer Verlag*

[BUR98] Burges, C.J.: **A tutorial on support vector machines for pattern recognition.** *Data mining and knowledge discovery, U. Fayyad, Ed. Kluwer Academic,* **1998,** pp. 1 - 43

[BOC01] Bock, J. R. und Gough, D. A.: **Predicting protein-protein interactions from primary structure.** *Bioinformatics 2001*, **17**: 455 - 460

[COW99] Coward, E.: **Shufflet: shuffling sequences while conserving the k-let counts.** *Bioinformatics*, **15**: 1058 - 1059

[DAN98] Dandekar, T., Snel, B., Huynen, M., Bork, P.: **Conservation of gene order: a fingerprint of proteins that physically interact.** *Trends Biochem Sci* 1998, **23**:324-328.

[ENR99] Enright A.J., Iliopoulos I., Kyrpides N.C., Ouzounis C.A.: **Protein interaction maps for complete genomes based on gene fusion events.** *Nature* 1999, **402**:86-90.

[FEL04] Felsenstein, J., (2004): **PHYLIP (Phylogeny Inference Package) version 3.6.** *Distributed by the author.* Department of Genome Sciences, University of Washington, Seattle.

[FRY96] Fryxell, K.J.: The coevolution of gene family trees. *Trends Genet.*, 12, 364-369

[GUS05] Gustafson,A., Allen,E., Givan,S., Smith,D., Carrington,J.C. und Kasschau,K.D. (2005): **ASRP: the Arabidopsis Small RNA Project Database.** *Nucleic Acids Research*, **33**, D637-D640

[HAS01] Hastie, T., Tibshirani, R. and Friedman, J. H.: **The Elements of Statistical Learning**.
Springer-Verlag, 2001.

[HIG92] Higgins, D.G., Bleasby, A.J., und Fuchs, R.: **CLUSTAL W: improved software for multiple sequence alignment.** *Comput. Appl. Biosci.* **8**, 189-191

[JON96] Jones,S. und Thornton,J.M. (1996) **Principles of protein–protein interactions.** *Proc. Natl Acad. Sci. USA*, **93**, 13–20.

[JEN04] Jensen,L.J., Lagarde,J., von Mering,C. and Bork,P. (2004) **ArrayProspector: a web resource of functional associations inferred from microarray expression data.** *Nucleic Acids Res.*, 32, W445–W448.

[KAN04] Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y., und Hattori, M. (2004): **The KEGG resource for deciphering the genome.** *Nucleic Acids Research*, **32**, D277 - D280

[KAT05] Katoh, K., Kuma, K., Toh, H. und Miyata, T. (2005): **MAFFT version 5: improvement in accuracy of multiple sequence alignment.** *Nucleic Acids Research*, **33**, 511 – 518

[KIS89] Kishino,H. and Hasegawa,M. (1989): **Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in hominoidea.** *J. Mol. Evol.*, **29**, 170-179.

[MCL71] McLachlan, A.D.: **Tests for comparing related aminoacid sequences.** *J. Mol. Biol*. **61**, 409-424

[MEI05] Meinicke, P., Klanke, S., Memisevic, R. und Ritter, H. (2005): **Principal Surfaces from Unsupervised Kernel Regression**, IEEE Transactions on pattern analysis and machine intelligence, **27**, 9:1379-1391

[MEM03] Memisevic, R. (2003): **Unsupervised Kernel Regression for Nonlinear Dimensionality Reduction.** *Diplomarbeit an der Technischen Fakultät der Universität Bielefeld*

[NAD64] Nadaraya, E. A. (1964): **On estimating regression.** *Theory Prob. Appl.* **9**:141-142.

[PAZ01] Pazos F., Valencia A.: **Similarity of phylogenetic trees as indicator of protein-protein interaction.** *Protein Eng* 2001, **14**:609-614.

[PEL99] Pellegrini M., Marcotte E.M., Thompson M.J., Eisenberg D., Yeates T.O.: **Assigning protein functions by comparative genome analysis: protein phylogenetic profiles.** *Proc Natl Acad Sci USA* 1999, **96**:4285-4288.

[PRE92] Press, W.H., Teukolsky, S.A., Vetterling W.T. und Flannery B.P.: **Numerical Recipies in C: theArt of Scientific Computing. 2nd edn.**, *Cambridge University Press*

[SAL04] Salwinski L., Miller C.S., Smith A.J., Pettit FK, Bowie JU, Eisenberg D (2004) **The Database of Interacting Proteins: 2004 update.** *NAR* **32 Database issue**:D449-51

[SAT05] Sato, T., Yamanishi, Y., Kanehisa, M., Hiroyuki, T.: **The inference of protein-protein interactions by co-evolutionary analysis is improved by excluding the information about the phylogenetic relationships.** *Bioinformatics 2005*, **21**: 3482 – 3489

[SHA04] Shawe-Taylor, J., Cristianini, N. (2004): **Kernel methods for pattern analysis.** *Cambridge University Press*

[SNE00] Snel, B., Lehmann, G., Bork, P. und Huynen, M. A.: **STRING: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene.** *Nucleic Acids Research* **28**, 18:3442-3444

[TAT03] Tatusov, R.L., Fedorova, N.D., Jackson, J.D., Jacobs, A.R., Kiryutin, B., Koonin, E.V., Krylov, D.M., Mazumder, R., Mekhedov, S.L., Nikolskaya, A.N., Rao, B.S., Smirnov, S., Sverdlov, A.V., Vasudevan, S., Wolf, Y.I., Yin, J.J., Natale, D.A.: **The COG database: an updated version includes eukaryotes.** *BMC Bioinformatics*. 2003 Sep 11;4:41

[ZAN02] Zanzoni A., Montecchi-Palazzi L., Quondam M., Ausiello G., Helmer-Citterich M. and Cesareni G. **MINT: a Molecular INTeraction database.** (2002) *FEBS Letters*, 513(1);135-140.

[ZEE03] Zeeck, A., Fischer, S.C., Grond, S. und Papastavrou, I. (2003): **Chemie für Mediziner** *Urban & Fischer Verlag*

A.2. Hilfsmittel

Die benötigten Klassen und Funktionen wurden in Java und MATLAB programmiert. Die Abbildungen wurden von mir mit Flash, JMol [http://jmol.sourceforge.net/] und Matlab erstellt.

Danksagung

Mein besonderer Dank gilt Herrn Dr. Peter Meinicke für seine ausführliche Betreuung und die Möglichkeit ein interessantes Thema im Rahmen dieser Masterarbeit bearbeiten zu dürfen. Weiterhin möchte ich Dr. Rainer Merkl für seine Unterstützung und Motivation danken.
Besonders bedanke ich mich auch bei meinen Eltern und meiner Freundin Ina für ihre Unterstützung und Geduld.

Göttingen, den 31. Oktober 2005 Nico Pfeifer