Alignmentfreie Analyse von Proteinsequenzen mit Verfahren des maschinellen Lernens

Dissertation
zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades
"Dr. rerum naturalium"
an der Georg-August-Universität Göttingen

vorgelegt von
Thomas Lingner
aus Wolgast

Göttingen, 2008

Referent: Prof. Dr. B. Morgenstern
Korreferent: Prof. Dr. S. Waack
Tag der mündlichen Prüfung: 6.10.2008

Danksagung

Als erstes möchte ich mich bei Prof. Dr. Burkhard Morgenstern dafür bedanken, dass er mir die Promotion in seiner Abteilung ermöglicht hat. Trotz vieler Verpflichtungen hat er sich stets die Zeit genommen, mir bei Problemen zu helfen und somit auch wesentlichen seelischen Beistand geleistet. Bei Prof. Dr. Stephan Waack möchte ich mich für wertvolle Tipps und Anregungen während der "Committee Meetings" und für interessante Diskussionen bedanken. Besonders danke ich Dr. Peter Meinicke, der mit sehr viel Engagement meine Arbeit betreute und mich durch seine fachliche und didaktische Kompetenz auf dem Gebiet des maschinellen Lernens motiviert und begeistert hat. Weiterhin danke ich allen Kollegen der Abteilung Bioinformatik für die großartige Unterstützung und die gute Arbeitsatmosphäre während der letzten drei Jahre.

Bei meiner Familie und meinen Freunden möchte ich mich für die seelische Unterstützung bedanken. Mit eurer Zuversicht und eurem Beistand habt ihr mir Sicherheit und Selbstvertrauen für diesen Lebensweg gegeben. Schließlich möchte ich mich bei Melanie bedanken, die während der letzten Jahre weit entfernt, aber trotzdem immer für mich da war.

Zusammenfassung

In den letzten Jahren ist die Anzahl bekannter Proteinfamilien und Proteinsequenzen aufgrund zahlreicher Genomprojekte exponentiell gestiegen. Die funktionale Charakterisierung dieser Sequenzen ist eine große Herausforderung, da klassische experimentelle Labormethoden zeitlich und finanziell sehr aufwändig sind. Daher werden rechnerbasierte Methoden verwendet, um die Funktion eines Proteins vorherzusagen oder um evolutionäre Verwandtschaftsverhältnisse von Sequenzen zu analysieren. Weit verbreitet sind in diesem Zusammenhang alignmentbasierte Methoden, welche unbekannte Sequenzen mittels ähnlicher Sequenzen in gut annotierten Datenbanken charakterisieren. Rechentechnisch sind alignmentbasierte Methoden für große Sequenzmengen jedoch sehr aufwändig.

Zur Zeit liefern diskriminative Methoden hervorragende Ergebnisse in Bereichen wie z.B. der Proteinklassifikation oder der Detektion entfernter Homologien. Bei alignmentfreien Verfahren dieser Kategorie werden alle Beispielsequenzen in einen einheitlichen Vektorraum abgebildet, um ein diskriminatives Modell in diesem Raum zu lernen und anzuwenden. Die gelernten diskriminativen Merkmale sind interpretierbar, d.h. sie können z.B. wichtige biochemische Eigenschaften einer Sequenzmenge widerspiegeln.

In dieser Arbeit werden zwei neue Methoden zur alignmentfreien Repräsentation und Analyse von Proteinsequenzen vorgestellt. Die Methoden sind in Kombination mit geeigneten Verfahren des maschinellen Lernens zur Detektion entfernter Homologien und zur Proteinklassifikation auf großen Sequenzmengen verwendbar. Die Evaluation der Methoden auf einem weit verbreiteten Testdatensatz zur Detektion entfernter Homologien demonstriert ihre Leistungsfähigkeit sowie die rechentechnische Effizienz und zeigt, wie die Methoden zur biologischen Interpretation gelernter Merkmale genutzt werden können. Weiterhin werden die Methoden auf einem im Rahmen dieser Arbeit erstellten umfassenden Testdatensatz zur Proteinfunktionsvorhersage mit einem angepassten Verfahren des maschinellen Lernens evaluiert. Auch dieser Ansatz zeigt hervorragende Ergebnisse und unterstreicht damit die generelle Eignung der Methoden zur Untersuchung verschiedener Probleme auf dem Gebiet der Proteinsequenzanalyse.

Inhaltsverzeichnis

Kapitel 1

Einführung

Proteine sind die Bausteine des Lebens: Sie erfüllen in den Zellen eines Organismus lebenswichtige Funktionen wie etwa Energieumwandlung, Nährstofftransport, Muskelbewegung und Replikation des Erbmaterials. Dementsprechend können sich die Abwesenheit eines Proteins oder sein fehlerhafter Aufbau negativ auf die Gesundheit oder gar Lebensfähigkeit eines Organismus auswirken. Obwohl Proteine schon seit vielen Jahren systematisch erforscht werden, werden immer wieder neue Typen – d.h. Proteinfamilien als Repräsentanten neuer Basisfunktionen – als auch neue "Angehörige" schon bekannter Proteinfamilien entdeckt [1,2]. Die Zuordnung funktionaler Eigenschaften zu neu entdeckten Proteinen und die Identifikation neuer Proteine bestimmter funktionaler Kategorien sind wichtig für die medizinische Forschung und pharmazeutische Therapie. So spielt die Suche nach möglichen "Targets" (Wirkstoff-Zielverbindungen) eine wichtige Rolle bei der Medikamentenentwicklung. Aber auch in der industriellen Anwendung – vor allem in der Biotechnologie – sind Proteine von großer Bedeutung. Hier sorgen z.B. Enzyme für die Beschleunigung chemischer Prozesse.

Ein bestimmtes (organismusspezifisches) Protein besteht aus einer charakteristischen Abfolge von Aminosäuren (Aminosäure- oder auch Protein*sequenz*), wobei 20 verschiedenartige Aminosäuren zum Aufbau von Proteinen beitragen können. Das Beispiel in Abbildung 1.1 zeigt die Aminosäuresequenz zweier Proteine der Familie der `14-3-3`-Proteine – Proteine, die andere Proteine binden – im sogenannten Einbuchstabencode, d.h. jede Aminosäure wird durch einen bestimmten Buchstaben des Alphabets repräsentiert. Die Proteinsequenz bestimmt (unter normalen Bedingungen) eindeutig die räumliche Struktur des Proteins, da die Faltung eines Proteins während der Proteinbiosynthese immer in identischer Weise

```
>Q6PC29|143G1_DANRE 14-3-3 protein gamma-1
>Danio rerio (Zebrafish) (Brachydanio rerio)
MVDREQLVQKARLAEQAERYDDMAAAMKSVTELNEALSNEERNLLSVAYKNVVGARRSSW
RVISSIEQKTSADGNEKKIEMVRAYREKIEKELETVCQDVLNLLDNFLIKNCGETQHESK
VFYLKMKGDYYRYLAEVATGEKRAAVVESSEKSYSEAHEISKEHMQPTHPIRLGLALNYS
VFYYEIQNAPEQACHLAKTAFDDAIAELDTLNEDSYKDSTLIMQLLRDNLTLWTSDQQDD
EGGEGNN

>Q6UFZ3|143G1_ONCMY 14-3-3 protein gamma-1
>Oncorhynchus mykiss (Rainbow trout) (Salmo gairdneri)
MVDREQLVQKARLAEQAERYDDMAAAMKSVTELNEALSNEERNLLSVAYKNVVGARRSSW
RVISSIEQKTSADGNEKKMEMVRAYREKIEKELETVCRDVLNLLDNFLIKNCNETQHESK
VFYLKMKGDYYRYLAEVATGEKRVGVVESSEKSYSEAHEISKEHMQPTHPIRLGLALNYS
VFYYEIQNAPEQACHLAKTAFDDAIAELDTLNEDSYKDSTLIMQLLRDNLTLWTSDQZDD
EGGETNN
```

Abbildung 1.1: Zwei Proteinsequenzen der Familie der `14-3-3`-Proteine. Zeilen, die mit einem "`>`" beginnen, dienen als Bezeichner der Sequenz. Die Proteine tragen dieselbe Funktionsbezeichnung (erste Bezeichnungszeile), stammen jedoch aus verschiedenen Organismen (zweite Bezeichnungszeile).

auf Grundlage der Aminosäurensequenz erfolgt [3]. Mit der räumlichen Struktur sind bestimmte funktionale Eigenschaften des Proteins assoziiert, z.B. der Typ eines zu bindenden Moleküls oder der Aufenthaltsort des Proteins in der Zelle.

Während die Aminosäuresequenz eines Proteins als Buchstabenkette spezifiert werden kann, ist die Funktion eines Proteins schwieriger zu definieren [4, 5]. So hängt die Beschreibung der Funktion eines Proteins oft vom Kontext ab, z.B. ob proteinspezifische biochemische Eigenschaften oder aber die Rolle eines bestimmten Proteins bei der Interaktion mit anderen Proteinen bei einer mutationsbedingten Krankheit untersucht werden. Die funktionale Charakterisierung von Proteinen resultiert üblicherweise in Annotationstexten und wissenschaftlichen Artikeln, welche vielfältige Aspekte der Funktionalität mit wechselndem Vokabular beschreiben.

Dennoch gibt es für die Charakterisierung von Proteinfunktionen explizite und implizite Klassifikationsschemata, welche die Anforderungen unterschiedli-

cher Untersuchungsrichtungen berücksichtigen. Weit verbreitete Beispiele für explizite Funktionsklassifikationen mit kontrolliertem Vokabular und definierten Beziehungen zwischen den Termen sind die "Enzyme Commission"-Klassifikation für Enzyme (EC, [6]) und die "Gene Ontology" (GO, [7]). GO bietet kontrollierte Vokabulare für drei Aspekte von Proteinfunktionen: die Funktion auf molekularer Ebene (z.B. Katalyse), den biologischen Prozess (z.B. die Rolle innerhalb eines bestimmten metabolischen Pfades) und den Aufenthaltsort in der Zelle (z.B. Zytoplasma). Während EC streng hierarchisch aufgebaut ist, sind die Terme in GO in einer gerichteten azyklischen Graphstruktur repräsentiert, d.h. ein Term kann auch mehreren Obertermen angehören.

Implizite Klassifikationsschemata basieren meist auf einer Zusammenfassung evolutionär oder funktionell verwandter Proteinsequenzen in Proteinfamilien oder Proteindomänenfamilien.[1] Zum Beispiel fasst die "Structural Classification Of Proteins" (SCOP, [8]) strukturell ähnliche Proteindomänen je nach Verwandtheitsgrad hierarchisch zusammen. Dadurch ergibt sich implizit auch eine funktionelle Ordnung, die z.B. für Annotationszwecke genutzt werden kann. Allerdings ist die Struktur von weniger als 1% der Proteine bekannt [9]. Pfam [10] ist eine Datenbank von Proteindomänenfamilien, die viele der bekannten Sequenzen abdeckt und zahlreiche familienspezifische Annotationen von Experten enthält. Somit kann auch Pfam als implizites Funktionskategorienschema für Proteine verwendet werden. Die Repräsentation der Familien in Pfam ist nichthierarchisch ("flach"). Pfam wird mittlerweile routinemäßig bei der Annotation neu sequenzierter Genome verwendet.

Bedingt durch inzwischen über 700 vollständige und weitere ca. 2800 noch nicht abgeschlossene Genomprojekte (siehe auch `http://www.genomesonline.org/gold_statistics.htm`) ist die Anzahl der bekannten Proteinsequenzen in den letzten Jahren rapide gestiegen. Zusätzlich werden duch Metagenomikprojekte wie z.B. [2, 11] Sequenzen gewonnen, die von bisher nicht kultivierbaren Organismen unterschiedlichster phylogenetischer Herkunft stammen. Allein in [2] wurden über 6 Millionen neue Proteinsequenzen identifiziert. Abbildung 1.2 zeigt die exponentielle Größenentwicklung der UniProtKB/TrEMBL-Datenbank [12]. Die darin enthaltenen Proteinsequenzen sind mittels rechnerbasierter Methoden vorläufig annotiert worden, jedoch steht die experimentelle Überprüfung dieser Annotation noch aus.

[1] "Domäne" bezeichnet üblicherweise eine funktionale Untereinheit eines Proteins.

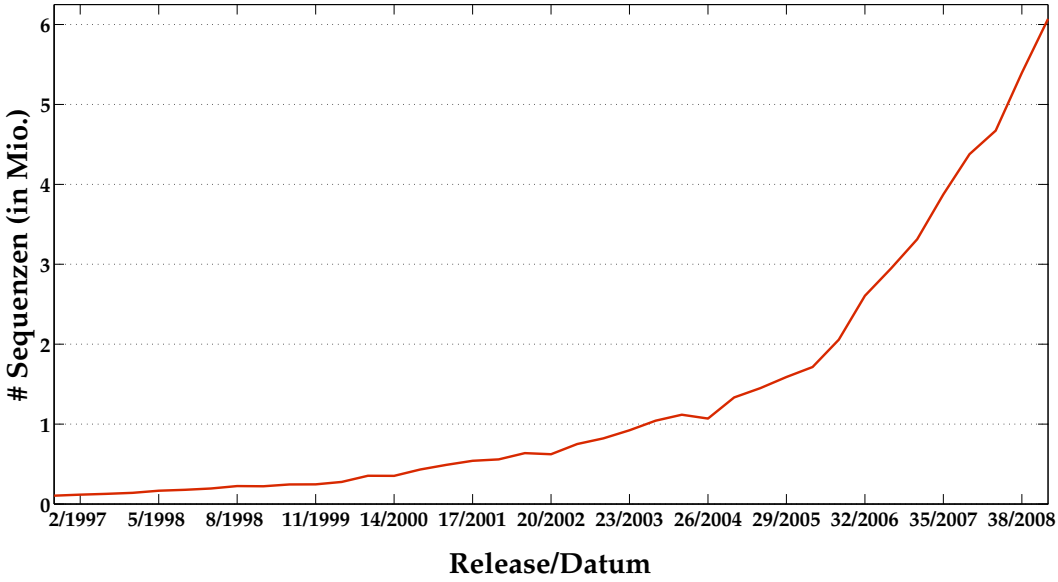

Abbildung 1.2: Anzahl der Proteinsequenzen in der UniProtKB/TrEMBL-Datenbank im Verlauf der letzten 12 Jahre (Angaben von `http://www.expasy.org/txt/old-rel/` und `ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/docs/relnotes.htm`).

1.1 Rechnerbasierte Proteinsequenzanalyse

Die funktionale Charakterisierung von Proteinen mittels klassischer experimenteller Methoden im Labor ist zeitlich und finanziell sehr aufwändig und wurde bisher – gemessen an der Menge der bekannten Sequenzen – nur für verhältnismäßig wenige Sequenzen durchgeführt [4, 9]. Daher werden rechnerbasierte Methoden verwendet, um die Funktion eines Proteins vorherzusagen oder um evolutionäre Verwandtschaftsverhältnisse von Sequenzen zu analysieren. Hierbei können die Methoden entsprechend der verwendeten Information über die zu annotierenden Proteine unterschieden werden: Außer den Ansätzen, die sich lediglich auf die Aminosäuresequenz stützen, existieren Methoden, welche Vorhersagen aufgrund der 3D-Struktur, Genexpressionsdaten, Protein-Protein-Interaktionsnetzwerken oder Literatur über die entsprechenden Proteine durchführen [5]. Für die letztgenannten Ansätze ist jedoch zusätzliches Wissen oder sind zusätzliche Experimente notwendig, daher konzentriert sich diese Arbeit auf rein sequenzbasierte Methoden.

1.1.1 Paarweise Alignmentmethoden

Weit verbreitete Methoden der rechnerbasierten Proteinsequenzanalyse bestimmen die paarweise Ähnlichkeit von Sequenzen mithilfe von Sequenzabgleichen ("Alignments"), um unbekannte Sequenzen mittels ähnlicher Sequenzen in gut annotierten Datenbanken zu charakterisieren. Die meistbenutzten Verfahren in diesem Zusammenhang sind FASTA [13] und BLAST [14], wobei sich letzteres zum Quasi-Standard entwickelt hat. Abbildung 1.3 zeigt das BLAST-Alignment der beiden Beispielsequenzen aus Abbildung 1.1. Einige Aminosäuren sind verschieden, was bei gemeinsamem evolutionären Ursprung auf evolutionsbedingte Mutationen schließen lässt. Bis auf sehr wenige Sequenzpositionen weisen die beiden Sequenzen jedoch eine hohe Aminosäureidentität (97 %) auf.

Die hohe Ähnlichkeit der beiden Sequenzen in Abbildung 1.3 deutet auf einen gemeinsamen evolutionären Ursprung (Homologie) hin.[2] Die offenbare Verwandtschaft kann dazu benutzt werden, um eventuell bestehende Annotationen der charakterisierten Sequenz ("Sbjct" in Abbildung 1.3) auf die zu untersuchende Sequenz ("Query" in Abbildung 1.3) zu übertragen ("Annotationstransfer"). Die biologische Motivation dieses homologiebasierten Annotationstransfers ist die Vermutung, dass zwei evolutionär eng verwandte Sequenzen wahrscheinlich eine ähnliche, wenn nicht sogar die gleiche Funktion erfüllen.

Andererseits beinhaltet der homologiebasierte Annotationstransfer drei schwerwiegende Probleme. **1) falscher Annotationstransfer:** Wie oben erwähnt, deutet eine enge evolutionäre Verwandtschaft auf eine funktionelle Ähnlichkeit hin, aber dies kann auch zu Fehlschlüssen führen. Zum Beispiel können homologe Proteine einem Genduplikationsereignis innerhalb eines Organismus entstammen (Paralogie). In diesem Fall besitzt das neue Protein den Freiheitsgrad, eine andere Funktion auszuführen als das Ursprungsprotein [9]. Auf Fehlschlüssen basierende Annotationen können durch den homologiebasierten Annotationstransfer schnell zur Fortpflanzung dieser Fehler führen [4].

2) geringe Sensitivität: Für homologe Sequenzen mit geringer Sequenzähnlichkeit ($<$ 50 % Aminosäureidentität) sind paarweise Alignmentmethoden nicht sensitiv genug, d.h. die Ähnlichkeit kann nicht mehr zuverlässig festgestellt werden. Die Folge ist, dass diese Sequenzen mit paarweisen Alignmentmethoden nicht

[2] Die Wahrscheinlichkeit, dass zwei Zufallsproteinsequenzen mit 247 Residuen 240 identische Aminosäuren aufweisen beträgt unter dem einfachsten denkbaren Wahrscheinlichkeitsmodell mit unabhängigen und gleichwahrscheinlichen Aminosäuren nur $(\frac{1}{20})^{240} * (\frac{19}{20})^{7} = 3.95 * 10^{-313}$.

```
Score =  477 bits (1227),  Expect = 6e-133
Identities = 240/247 (97%), Positives = 243/247 (98%), Gaps = 0/247 (0%)

Query  1    MVDREQLVQKARLAEQAERYDDMAAAMKSVTELNEALSNEERNLLSVAYKNVVGARRSSW
            MVDREQLVQKARLAEQAERYDDMAAAMKSVTELNEALSNEERNLLSVAYKNVVGARRSSW
Sbjct  1    MVDREQLVQKARLAEQAERYDDMAAAMKSVTELNEALSNEERNLLSVAYKNVVGARRSSW

Query  61   RVISSIEQKTSADGNEKKIEMVRAYREKIEKELETVCQDVLNLLDNFLIKNCGETQHESK
            RVISSIEQKTSADGNEKK+EMVRAYREKIEKELETVC+DVLNLLDNFLIKNC ETQHESK
Sbjct  61   RVISSIEQKTSADGNEKKMEMVRAYREKIEKELETVCRDVLNLLDNFLIKNCNETQHESK

Query  121  VFYLKMKGDYYRYLAEVATGEKRAAVVESSEKSYSEAHEISKEHMQPTHPIRLGLALNYS
            VFYLKMKGDYYRYLAEVATGEKR  VVESSEKSYSEAHEISKEHMQPTHPIRLGLALNYS
Sbjct  121  VFYLKMKGDYYRYLAEVATGEKRVGVVESSEKSYSEAHEISKEHMQPTHPIRLGLALNYS

Query  181  VFYYEIQNAPEQACHLAKTAFDDAIAELDTLNEDSYKDSTLIMQLLRDNLTLWTSDQQDD
            VFYYEIQNAPEQACHLAKTAFDDAIAELDTLNEDSYKDSTLIMQLLRDNLTLWTSDQ+DD
Sbjct  181  VFYYEIQNAPEQACHLAKTAFDDAIAELDTLNEDSYKDSTLIMQLLRDNLTLWTSDQZDD

Query  241  EGGEGNN
            EGGE NN
Sbjct  241  EGGETNN
```

Abbildung 1.3: Paarweises Alignment der beiden Sequenzen aus Abbildung 1.1. Das Alignment wurde mit BLAST bl2seq (`http://blast.ncbi.nlm.nih.gov/bl2seq`) unter Benutzung der Standardparameter erstellt. Die oberen beiden Zeilen spiegeln die Ähnlichkeit der Sequenzen in verschiedenen Maßen wider (siehe auch [14]).

homologiebasiert annotiert werden können [4]. Dies betrifft insbesondere Metagenomikprojekte, da hier Sequenzen von Organismen unbekannten evolutionären Ursprungs und bisher nicht beobachteter phylogenetischer Divergenz anfallen. In diesem Fall ist die Detektion entfernter Homologien ("remote homology detection", Homologie von Sequenzen mit sehr geringer Sequenzähnlichkeit von < 30 %) unabdingbar.

3) schlechte Skalierbarkeit: Mit dem exponentiellen Wachstum der Proteinsequenzdatenbanken steigt auch der Aufwand für die homologiebasierte Annotation exponentiell, da eine zu charakterisierende Sequenz mit allen bekannten Sequenzen verglichen werden muss. Zwar wächst die Anzahl der bekannten funk-

tionellen Kategorien nicht mit derselben Geschwindigkeit, jedoch kann wegen der geringen Sensitivität nicht nur auf Sammlungen mit einzelnen ausgewählten Beispielsequenzen zurückgegriffen werden. Noch problematischer wird die schlechte Skalierbarkeit, wenn der Vergleich jeder Sequenz mit jeder anderen notwendig ist ("all-against-all"), z.B. bei Clusteranalysen zur Erstellung redundanzreduzierter Sequenzdatenbanken [15]. In [2] wurde für die systematische Ähnlichkeitsanalyse aller zu dieser Zeit bekannten Proteinsequenzen insgesamt über eine Million CPU-Stunden benötigt. Die quadratische Abhängigkeit der algorithmischen Komplexität von der Menge der Sequenzen macht die Anwendung paarweiser Alignmentmethoden für viele Probleme sehr schwierig oder sogar unmöglich.

1.1.2 Profilbasierte Ansätze

Ebenso gebräuchlich wie paarweise Alignmentmethoden sind Ansätze, welche statistische Eigenschaften von Proteinfamilien (oder andersartig zusammengehörigen Sequenzen) in Modellen repräsentieren. Diese Profile werden üblicherweise nichtdiskriminativ gewonnen, d.h. ausschließlich auf Basis bekannter Beispiele der Proteinfamilie.

PSI-BLAST [16] stellt eine Mischform aus paarweisen Alignmentmethoden und profilbasierten Ansätzen dar. Hier werden die signifikanten Treffer einer initialen BLAST-Datenbanksuche (mit einer Sequenz) zu einer "Profilsequenz" zusammengefasst. Im nächsten Schritt wird diese Profilsequenz zur Datenbanksuche verwendet – dadurch werden auch Sequenzen mit geringerer Ähnlichkeit zur Ursprungssequenz detektiert. Dieses Prinzip kann iterativ mit einer festgelegten Anzahl von Suchschritten durchgeführt werden oder bis es keine neuen (relevanten) Treffer mehr gibt. Das Verfahren kann als "Modensuche" im Sequenzraum bezeichnet werden und eignet sich besser zur Detektion entfernter Homologer als BLAST.

Weitere Ansätze dieser Kategorie basieren auf Profil-Hidden-Markov-Modellen (PHMM, [17–19]). PHMMs repräsentieren unterschiedlich konservierte Regionen einer Menge von Sequenzen durch Zustände eines probabilistischen generativen Modells. Zur Konstruktion der Modelle werden multiple Alignments der Sequenzen benutzt, auf Grundlage derer die Wahrscheinlichkeitsparameter durch Auszählen der Beobachtungen gewonnen werden. Die Annotation einer unbekannten Sequenz erfolgt dann durch Alignierung dieser Sequenz gegen alle Modelle einer Modelldatenbank. Weil auch kurzreichweitige Abhängigkeiten zwischen Sequenzpositionen in den Modellen repräsentiert werden, sind PHMMs zur De-

tektion entfernter Homologer besser geeignet als paarweise Methoden und PSI-BLAST [17,20].

Eine sehr bekannte und vielbenutzte Modelldatenbank von PHMMs stellt Pfam [10] dar. Pfam ist eine von Experten zusammengestellte und gut annotierte Sammlung multipler Alignments und korrespondierender PHMMs von Proteindomänenfamilien. Mit der HMMER Software (`http://hmmer.janelia.org/`) können diese Modelle verwendet werden, um neue Sequenzen einer oder mehreren der über 10000 Domänenfamilien zuzuordnen. Dies ist jedoch durch die notwendigen Alignments sehr zeitaufwändig, so dass verschiedene Methoden entwickelt wurden, um den Suchprozess zu beschleunigen. Eine Möglichkeit stellt die Verwendung von Parallelisierung und Hardware-Beschleunigung dar (z.B. [21]), für verschiedene alignmentbasierte Verfahren wird außerdem Spezialhardware angeboten, welche die Suche um das ca. 500-fache beschleunigt (`http://www.timelogic.com/seqcruncher.html`). Andererseits kann durch eine Vorfilterung mit recheneffizienten Verfahren die Menge der Modelle reduziert werden, die mit HMMER untersucht werden muss (z.B. `http://www.microbesonline.org/fasthmm/`). Offenbar soll die nächste Version von HMMER (HMMER3, geplant für 2009) eine Beschleunigung des Suchprozesses durch Integration von Vorabfiltern (`ftp://selab.janelia.org/pub/software/hmmer/2.4i/NOTES`) bei gleichzeitig gesteigerter Sensitivität bieten [22].

1.1.3 Ansätze mit Methoden des maschinellen Lernens

Seit einigen Jahren werden vermehrt Methoden des maschinellen Lernens zur Proteinklassifikation (einen Überblick geben [23–25]) und insbesondere zur Detektion entfernt verwandter Sequenzen verwendet (z.B. [26–33]). Diese Ansätze arbeiten in der Regel diskriminativ, d.h. zusätzlich zu den bekannten Sequenzen einer Proteinfamilie, welche als positive Lernbeispiele fungieren, werden Sequenzen anderer – nicht verwandter – Familien als negative Beispiele verwendet. In den resultierenden diskriminativen Modellen (Diskriminanten) werden nach dem Lernprozess die Unterschiede zwischen den Familien explizit repräsentiert. In vergleichenden Studien wurde gezeigt, dass diskriminative Methoden bei der Detektion entfernter Homologer den paarweisen Alignmentmethoden und auch den nichtdiskriminati-

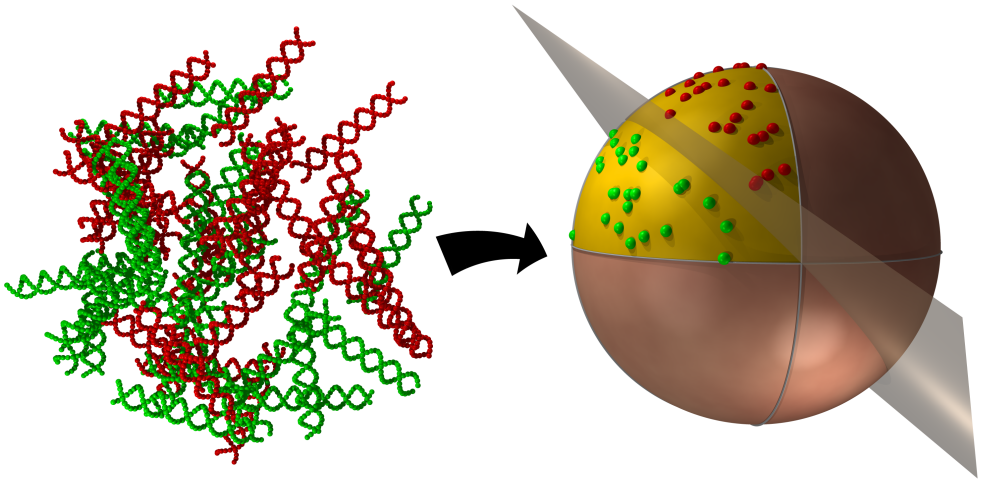

Abbildung 1.4: Schema der Abbildung von biologischen Sequenzen in einen Merkmalsraum und Trennung der verschiedenen Mengen (repräsentiert durch verschiedene Farben) durch eine Hyperebene. Abbildung mit freundlicher Genehmigung von Dr. Tobias Glasmachers.

ven PHMMs überlegen sind (z.B. [26, 29]).[3]

Um Methoden des maschinelles Lernen auf Proteinsequenzen anwenden zu können, bedarf es einer geeigneten Repräsentation der Proteinsequenzen in einem Vektorraum ("Merkmalsraum") und eines Lernverfahrens, mit welchem die Diskriminante in diesem Vektorraum berechnet werden kann. Abbildung 1.4 veranschaulicht das Prinzip der Abbildung von Sequenzen in einen Merkmalsraum und die dortige Trennung zweier Beispielmengen mittels einer linearen Diskriminante. Die gelernte Diskriminante kann in diesem Fall als Vektor von diskriminativen Gewichten für die einzelnen Dimensionen des Merkmalsraums gesehen werden.

Während als Lernverfahren heutzutage üblicherweise Support-Vektor-Maschinen (SVM, [34]) zum Einsatz kommen, unterscheiden sich die Ansätze in der Repräsentation der Proteinsequenzen. Eine einfache, aber dennoch leistungsfähige Proteinsequenzrepräsentation wurde in [27] vorgeschlagen: das k-mer-Spektrum. Beim k-mer-Spektrum werden die Häufigkeiten von Teilsequenzen der Länge k in einer Sequenz gezählt. Jedem der (für Proteinsequenzen) 20^k verschiedenen k-mere entspricht eine Dimension im Merkmalsraum des k-mer-Spektrums. In Kombination mit einer Normierung der resultierenden Merkmalsvektoren auf eine einheit-

[3] Die Eignung der Evaluationsszenarien wird später diskutiert.

liche Länge können damit auch die von Natur aus verschieden langen Proteinsequenzen in einen einheitlichen Vektorraum abgebildet werden.

Andererseits wird beim diskriminativen Lernen oft gar nicht auf eine explizite Repräsentation der Sequenzen zurückgegriffen. Stattdessen wird die Ähnlichkeit zweier Sequenzen mittels sogenannter Kernfunktionen ("Kerne") berechnet und ein kernbasiertes Lernverfahren zum Training der Diskriminante verwendet [35]. Ein Sequenzkern berechnet das innere Produkt zweier Datenelemente in einem Merkmalsraum, wobei dessen Dimensionen keine intuitive Bedeutung haben müssen ("abstrakter" Merkmalsraum). Dies ermöglicht das Lernen in komplexen und hochdimensionalen Merkmalsräumen. Beispielsweise wird in [31] die Summe der Scores[4] der lokalen Alignments zweier Sequenzen als Ähnlichkeitsmaß und damit als Grundlage für den Sequenzkern verwendet ("Local-Alignment-Kernel"). In diesem Fall entsprechen die Dimensionen des assoziierten Merkmalsraums der Ähnlichkeit einer Sequenz zu allen theoretisch möglich Sequenzen, also einer "abstrakten" Ähnlichkeit. Dadurch ist ein intuitiver Bezug gelernter diskriminativer Gewichte auf Sequenzmerkmale nicht möglich. Auch die k-mer-Spektrum-Methode wurde in Form eines Sequenzkerns eingeführt. Die Kernfunktion zweier Sequenzen ist in diesem Fall das Skalarprodukt der mit den Sequenzen assoziierten Merkmalsvektoren. Kernbasierte Methoden zur Detektion entfernter Homologer haben sich als besonders leistungsfähig herausgestellt ([31–33]).

Ein gravierender Nachteil kernbasierter Methoden ist die schlechte Skalierbarkeit. Zur Berechnung einer Diskriminante für N Trainingsbeispiele müssen $O(N^2)$ Kernfunktionen berechnet werden.[5] Weiterhin erfordert die Anwendung der gelernten Diskriminanten – z.B. zur Charakterisierung unbekannter Sequenzen – in der Regel die Berechnung der Ähnlichkeit jedes Testbeispiels zu jedem Trainingsbeispiel, d.h. für die Klassifikation neuer Sequenzen müssen $O(N)$ Kernfunktionen berechnet werden.[6] Für Probleme, bei denen Tausende oder Millionen Sequenzen anfallen, ist diese Testmethode daher sehr zeitaufwändig. Dementsprechend erfolgte die Evaluation kernbasierter Methoden nur auf Datensätzen, welche speziell auf die beschränkte Skalierbarkeit zugeschnitten sind. Beispielsweise beinhaltet der oft verwendete Datensatz in [29] lediglich 4352 Sequenzen aus 54 Pro-

[4] Der Score bezeichnet hier die Qualität eines Alignments.

[5] Üblicherweise erfolgt die Speicherung der Kernfunktionen in Form einer $N \times N$ Kernmatrix. Dies ist selbst auf Hochleistungsrechenanlagen nur für maximal $N \approx 10^4$ Beispiele praktikabel.

[6] Bei Support-Vektor-Maschinen werden nur zu Support-Vektoren korrespondierende Trainingsbeispiele benötigt.

tein(super)familien. Dies impliziert auch eine sehr eingeschränkte praktische Anwendbarkeit der gelernten Diskriminanten auf ebendiese 54 Kategorien.

Andererseits kann für explizite Repräsentationsmethoden mit moderater Dimensionalität – wie z.B. das k-mer-Spektrum für $k \leq 3$ – die Diskriminante im Merkmalsraum für die schnelle Klassifikation verwendet werden [27]. Dabei kann die Diskriminante auch auf Grundlage der kernbasiert gelernten Gewichte berechnet werden. Die Anwendung einer Diskriminante im Merkmalsraum zur Klassifikation neuer Sequenzen erfordert lediglich die Transformation der Testsequenz(en) in den Merkmalsraum und die Berechnung des Skalarprodukts aus diskriminativem Gewichtsvektor und Merkmalsvektor(en).

Ein weiterer Vorteil expliziter Repräsentationsmethoden gegenüber Methoden mit abstraktem Merkmalsraum besteht in der Interpretierbarkeit der gelernten diskriminativen Gewichte. Sofern die Merkmalsraumdimensionen bedeutungsvollen Sequenzeigenschaften entsprechen, kann die Diskriminante zur Analyse dieser herangezogen werden. Beim k-mer-Spektrum beispielsweise deuten hohe positive diskriminative Gewichte auf Überrepräsentiertheit entsprechender k-mere in den Sequenzen einer Proteinfamilie hin. Bei Repräsentation in einem abstraktem Merkmalsraum entstehen im Lernprozess lediglich diskriminative Sequenzgewichte. Hier ist – wie im Beispiel des Local-Alignment-Kernels angedeutet – nicht klar, inwieweit diese Sequenzgewichte von Nutzen für weitere Analysen sind.

Wie bereits erwähnt, kann mit kernbasierten Methoden auch in komplexen und hochdimensionalen Merkmalsräumen diskriminativ gelernt werden. Dies impliziert auch die Möglichkeit, beliebig viele Parameter im Ähnlichkeitsmaß zu verwenden. Beispielsweise beinhaltet der oben erwähnte Local-Alignment-Kernel mehrere Parameter für die Konstruktion und Bewertung der Alignments und weitere Parameter zur Transformation des Ähnlichkeitsmaßes in einen validen Kern.[7] Können diese Parameter nicht auf Grundlage der Trainingsdaten bestimmt werden, bezeichnet man sie als *Hyperparameter*. Eine Vielzahl von Hyperparametern aber bedeutet, dass eine sorgfältige Messung des Einflusses der Parameter auf die Performanz des Ansatzes notwendig ist. Stehen wenig Lernbeispiele zur Verfügung oder wird – wie im Datensatz aus [29] – auf eine Validierungsmenge verzichtet und somit die Parameter direkt bezüglich der Testdaten optimiert, so besteht die Gefahr der Überanpassung ("overfitting"). Viele kernbasierte Ansätze mit zahlreichen Hyperparametern (z.B. [31, 33]) wurden auf dem Datensatz bezüglich

[7] Valide sind in diesem Zusammenhang sogenannte Mercer-Kernel [36].

der Testbeispiele optimiert und erzielten eine hervorragende Performanz. Vor diesem Hintergrund stellt sich die Frage, inwieweit die Ansätze und insbesondere die eingestellten Parameter auf andere Probleme anwendbar sind.

1.2 Ziele der Arbeit

Das Hauptziel dieser Arbeit besteht darin, die im vorherigen Abschnitt erwähnten Vorteile merkmalsbasierter Methoden des maschinellen Lernens für die diskriminative Analyse von Proteinsequenzen – insbesondere zur Detektion entfernter Homologien und zur Proteinfunktionsvorhersage – nutzbar zu machen und umfassend zu evaluieren. Dazu sollen leistungsfähige alignmentfreie Repräsentationsmethoden für Proteinsequenzen entwickelt werden, welche ohne langwierige (Re-)Evaluation der Parameter auf unterschiedliche Probleme der Sequenzanalyse anwendbar sind. Dies erfordert eine Beschränkung der Methoden auf wenige, idealerweise biologisch bedeutungsvolle Parameter. Weiterhin sollen die mithilfe dieser Methoden gelernten diskriminativen Merkmale intuitiv interpretierbar sein und somit Anhaltspunkte für spezifischere experimentelle Untersuchungen liefern. Damit sich aus dieser hinweisgebenden Analyse ein signifikanter Zeit- und Aufwandsvorteil gegenüber Labormethoden ergibt, müssen die Methoden eine effiziente Anwendung der gelernten Modelle ermöglichen und sollten mit geringem Einarbeitungsaufwand von vielen Forschern benutzbar sein.

Die Evaluation der Methoden bezüglich ihrer Vorhersageperformanz, Interpretierbarkeit und rechentechnischen Effizienz im Vergleich mit den derzeit leistungsfähigsten Ansätzen soll auf einem weit verbreiteten Testdatensatz zur Detektion entfernter Homologien erfolgen. Als Beleg für die Praxistauglichkeit wird außerdem die Evaluation auf einem Testdatensatz verfolgt, der die Reichhaltigkeit der bekannten Proteinsequenzen und die Probleme bei der Proteinfunktionsvorhersage widerspiegelt. Mangels Verfügbarkeit adäquater Testdatensätze ist dazu im Rahmen der Arbeit die Erstellung eines solchen Testdatensatzes notwendig.

Kapitel 2

Ergebnisse und Diskussion

Im Rahmen der vorliegenden Arbeit sind zwei Proteinsequenzrepräsentationsmethoden untersucht worden [37, 38], welche eine alignmentfreie Analyse von Proteinsequenzen erlauben. Die Methoden wurden auf einem weit verbreiteten Testdatensatz zur Detektion entfernt verwandter Sequenzen [29] evaluiert und zeigten hervorragende Ergebnisse. Weiterhin wurde im Rahmen dieser Arbeit ein Testdatensatz zusammengestellt [39], welcher die vielfältigen Aspekte der Proteinfunktionsvorhersage berücksichtigt. Zur Evaluation der beiden Proteinsequenzrepräsentationsmethoden wurde ein Verfahren des maschinellen Lernens an die Anforderungen dieses Testdatensatzes angepasst. Im Folgenden werden die Repräsentationsmethoden und der Testdatensatz vorgestellt sowie deren Eigenschaften und die Evaluationsergebnisse diskutiert.

2.1 Oligomerdistanzhistogramme

Der Begriff "Oligomer" steht hier für eine sehr kurze Proteinsubsequenz der Länge $k = 1, \ldots, 3$. Der Merkmalsraum der Oligomerdistanzhistogramme (ODH, [37]) zur Repräsentation von Proteinsequenzen besteht aus insgesamt $(20^k)^2$ Histogrammen entsprechend aller verschiedenen k-mer-Paare. Jedes dieser Histogramme repräsentiert die Häufigkeit eines bestimmten k-mer-Paars für verschiedene Abstände der k-mere in einer Sequenz, als Abstand zählt die Differenz der Anfangspositionen der k-mere.[1] Die Vorkommenshäufigkeit eines k-mer-Paars wird für jede Distanz separat gezählt (entspricht Histogrammintervallbreite 1), d.h. es werden kei-

[1] In den Histogrammen wird auch der Spezialfall identischer Anfangspositionen (Distanz 0) berücksichtigt.

ne Distanzen zusammengefasst. Die Distanzhistogramme können in Vektorform übereinander "gestapelt" werden und bilden somit den ODH-Merkmalsraum. Die mit den Sequenzen assoziierten Merkmalsvektoren werden zur besseren Vergleichbarkeit unterschiedlich langer Sequenzen auf gleiche (euklidische) Länge normiert.

Um einen einheitlichen Merkmalsraum für alle Proteinsequenzen zu erhalten, entspricht die größte Distanz in jedem Histogramm der Maximaldistanz der längsten Sequenz in einer Sequenzsammlung. In Abhängigkeit von k und der Maximaldistanz ergibt sich somit ein hochdimensionaler Merkmalsraum, z.B. umfasst der ODH-Merkmalsraum für Trimere ($k = 3$) bei Verwendung einer Maximaldistanz von $D = 1000$ mehr als $6.4 * 10^{10}$ Dimensionen. Jedoch sind nicht alle Sequenzen so lang, dass sie Distanzen nahe der Maximaldistanz aufweisen. Außerdem nimmt in Proteinen die Distanzkonserviertheit mit dem Abstand der Aminosäuren in der Sequenz aufgrund evolutionsbedingter Insertionen und Deletionen ab. Daher wurde in [39] die Beschränkung der Maximaldistanz für ODHs eingeführt. Dies erlaubt die Definition von ODH-Merkmalsräumen mit moderater Dimensionalität zur Verwendung mit Lernverfahren für große Datenmengen. Stehen wie beim Testdatensatz in [29] nur wenige Lernbeispiele zur Verfügung, können ODHs auch mit kernbasierten Lernmethoden verwendet werden. Die Kernfunktion zweier Sequenzen ist in diesem Fall – wie beim k-mer-Spektrum – das Skalarprodukt ihrer (normierten) Merkmalsvektoren. Dies ermöglicht auch die Verwendung hochdimensionaler ODH-Merkmalsräume, also z.B. die Verwendung längerer Oligomere oder sehr hoher Distanzen. In diesem Fall bietet sich die spärliche Repräsentation der Merkmalsvektoren an, da nur verhältnismäßig wenige Dimensionen – entsprechend der relevanten Sequenzmerkmale – einen von 0 verschiedenen Wert haben.

Bestimmte Dimensionen von Oligomerdistanzhistogrammen haben einen interessanten Bezug zum k-mer-Spektrum: für Monomere ($k = 1$) entsprechen die mit der Distanz 0 (Distanz 1) assoziierten ODH-Dimensionen dem Monomerspektrum (Dimerspektrum). Im ODH-Merkmalsraum für Dimere ($k = 2$) sind sogar die Merkmale des Dimer-, Trimer- und Tetramerspektrums enthalten (Distanzen 0, 1 und 2).

Bei der Repräsentation einer Sequenz mit der k-mer-Spektrum-Methode geht die Positionsinformation der jeweiligen Merkmale komplett verloren. Bei alignmentbasierten Ansätzen hingegen ist die Positionsinformation zentraler Bestandteil der im Alignment korrespondierenden Sequenzregionen. Die ODH-Methode kann in diesem Zusammenhang als "Zwischenrepräsentation" bezeich-

net werden, da durch die Verwendung großer Positionsdifferenzen relative Positionsinformation modelliert wird. Dies führt auch zu einer impliziten Längenmodellierung der Sequenzen, welche eine zusätzliche Information bei der Beschreibung von Proteinsequenzen darstellt.

Aufgrund mehrerer Nachfragen internationaler Forscher haben wir die ODH-Methode als MATLAB®-Toolbox unter `http://www.gobics.de/thomas/ODH` bereitgestellt.

2.1.1 Performanz

Die Leistungsfähigkeit der ODHs wurde auf einem weit verbreiteten Testdatensatz zur Analyse entfernt verwandter Sequenzen ([29], siehe auch Abschnitt 1.1.3) evaluiert. Dabei zeigte sich, dass ODHs sowohl nichtdiskriminativen Ansätzen (z.B. PSI-BLAST und PHMMs) als auch vielen diskriminativen Ansätzen (z.B. k-mer-Spektrum) bezüglich der Detektionsleistung überlegen sind. Lediglich alignmentbasierte Methoden (z.B. der Local-Alignment-Kernel, [31]) zeigten eine bessere Performanz. Allerdings wurden die teilweise zahlreichen Hyperparameter vieler Ansätze (z.B. [31, 33, 40]) auf dem Testdatensatz optimiert, was einen objektiven Vergleich der Leistungsfähigkeit erschwert. ODHs ohne Beschränkung der Maximaldistanz (wie sie in [37] eingeführt wurden) weisen mit der Oligomerlänge k nur einen Hyperparameter auf, der sich zudem auch auf sehr wenige sinnvolle Werte beschränken lässt.

Die beste Performanz der distanzbasierten Repräsentation in [37] wurde für $k = 1$ (also Monomerdistanzhistogramme) festgestellt. Für Dimerdistanzhistogramme wird die Performanz nur unwesentlich schlechter, doch für Trimerdistanzhistogramme bricht die Detektionsleistung stark ein. Dieses Phänomen kann mit der Performanzentwicklung für das k-mer-Spektrum für längere k-mere verglichen werden: Da nur noch sehr wenige k-mere (bzw. k-mer-Paare mit einem bestimmten Abstand) in zwei verschiedenen Sequenzen übereinstimmen, geht die Ähnlichkeit dieser Sequenzen – die hier als Skalarprodukt der korrespondierenden Merkmalsvektoren formulierbar ist – gegen Null [30].

In einer weiteren Untersuchung zur Feststellung der Eignung verschiedener Repräsentationmethoden für Proteinsequenzen für die Proteinfunktionsvorhersage erzielten ODHs wesentlich bessere Ergebnisse als das k-mer-Spektrum ([39], siehe auch Abschnitt 2.3). Hierbei stellte sich die Beschränkung der Maximaldistanz als geeignetes Mittel zur Begrenzung der Dimensionalität des Merkmalsraums her-

aus.

In einem kürzlich erschienenen Artikel [41] wurde die Kombination verschiedener Sequenzkerne zur Detektion entfernt verwandter Sequenzen evaluiert. Hier stellten sich die Monomerdistanzhistogramme noch vor dem Local-Alignment-Kernel als höchstgewichtete Methode heraus. Die Kombination von Local-Alignment-Kernel und Monomerdistanzhistogrammen erzielte eine hervorragende Performanz.

2.1.2 Interpretierbarkeit der Merkmale

Eine herausstechende Eigenschaft der Oligomerdistanzhistogramme ist die Interpretierbarkeit der gelernten diskriminativen Gewichte im Merkmalsraum. Ein hohes positives Gewicht einer Dimension deutet auf eine Diskriminativität des assoziierten Merkmals hin, beispielsweise ein gehäuftes Vorkommen von Alanin und Serin mit einem Abstand von 4 Sequenzpositionen in den positiven Lernbeispielen. Dabei können zur besseren visuellen Erfassung die diskriminativen Anteile bestimmter Oligomerpaare (für alle Distanzen) bzw. bestimmter Distanzen (für alle Oligomerpaare) zusammengefasst werden, um weniger spezifische Muster zu identifizieren. Abb. 2.1 zeigt die für alle Monomerpaare zusammengefassten diskriminativen Gewichte aus einem Experiment des oben erwähnten Testdatensatzes (Abb. aus [37], jedoch hier in Farbe). Dabei sind die Monomerpaare in einer Matrix gegeneinander aufgetragen, was die Paaridentifikation besonders einfach macht. Mit der intuitiv verständlichen Farbskala können so sehr schnell Oligomerpaare identifiziert werden, die charakteristisch für die (mit den positiven Lernbeispielen assoziierte) Proteinfamilie sind.

Über eine Analyse der Sequenzpositionen entsprechend der diskriminativsten Merkmale lassen sich zudem charakteristische Sequenzregionen bzw. Sequenzpositionen der untersuchten Proteinfamilie abbilden. Diese Identifikation biologisch bedeutungsvoller Merkmale impliziert einen sehr spezifischen Hinweis für weitere experimentelle Untersuchungen, z.B. strukturelle Analysen, und kann somit Zeit und Aufwand sparen.

2.1.3 Rechentechnische Effizienz

Bei der Evaluation der Oligomerdistanzhistogramme in [37] wurden die Diskriminanten mit einer kernbasierten Variante von Support-Vektor-Maschinen bestimmt.

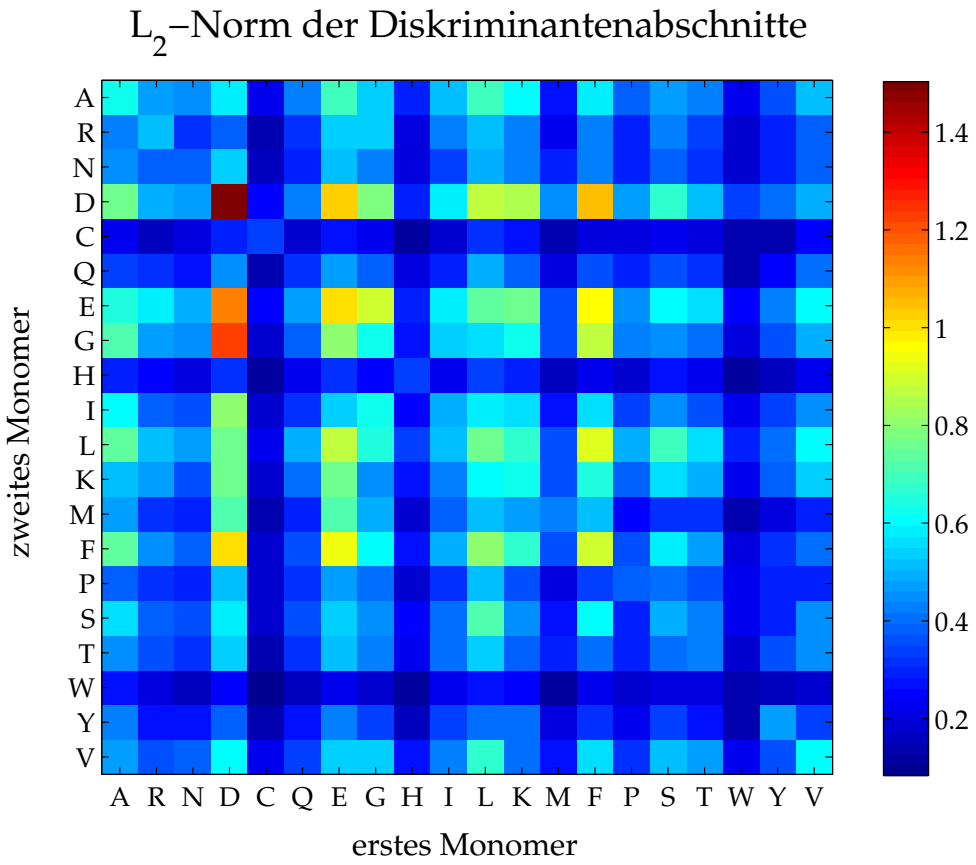

Abbildung 2.1: Matrixdarstellung der zusammengefassten diskriminativen Gewichte für Monomerdistanzhistogramme entsprechend Experiment 51 aus [37] (entspricht SCOP-Familie 1.41.1.5). Jedes Matrixelement entspricht der euklidischen Norm der Dimensionen des diskriminativen Gewichtsvektors, welche mit dem Distanzhistogramm des Monomerpaares assoziiert sind. Die Werte sind entsprechend der Farbskala auf der rechten Seite kodiert.

Die Berechnung der erforderlichen Kernmatrix für die über 4000 Lernbeispiele konnte durch Ausnutzung der expliziten Repräsentation der Merkmalsvektoren und Anwendung effizient implementierter Matrixalgebra von – im zeitgünstigsten Fall für $k = 1$ – einigen Minuten für die konventionelle Berechnungsweise auf wenige Sekunden reduziert werden. Dabei spielt die Merkmalsextraktion – also die Transformation der Proteinsequenzen in die numerische ODH-Repräsentation – eine wesentliche Rolle. Werden alle Abstände zwischen Oligomeren in den Sequenzen betrachtet, so wächst der Berechnungsaufwand für die Extraktion quadratisch

mit der Länge der Sequenz(en). Wird die Maximaldistanz jedoch beschränkt, so hängt der Aufwand nur noch linear von der Sequenzlänge (und der Maximaldistanz) ab.

In 1.1.3 wurde erwähnt, dass bei expliziten Repräsentationsmethoden die Diskriminante im Merkmalsraum zur schnellen Klassifikation von unbekannten Sequenzen genutzt werden kann. Bei der Evaluation der Oligomerdistanzhistogramme in [37] konnte eine Beschleunigung der Klassifikation um mehr als das 1000-fache gegenüber dem Local-Alignment-Kernel festgestellt werden. Der Aufwand zur Berechnung einer einzelnen Kernfunktion ist bei beiden Methoden von algorithmischer Komplexität $O(L^2)$. Andererseits ist für die Klassifikation einer neuen Sequenz mit der ODH-Methode nur die Transformation der Sequenz in den Merkmalsraum (ebenfalls $O(L^2)$) und die Berechnung des Skalarprodukts aus Diskriminante und Merkmalsvektor notwendig. Die alignmentbasierte Methode hingegen erfordert im betrachteten Szenario mangels expliziter Repräsentation die Berechnung von durchschnittlich über 1000 Kernfunktionen zur Berechnung des Klassifikations-Scores einer Sequenz. Dies fällt besonders ins Gewicht, wenn viele Proteinfamilien für die Klassifikation infrage kommen, da der Aufwand sich entsprechend der Anzahl der Kategorien vervielfacht.

2.2 Wortkorrelationsmatrizen

Die Grundlage dieser Repräsentationsmethode ist ein Sequenzkern, der die Ähnlichkeit zweier Sequenzen mittels der durchschnittlichen Wortähnlichkeit beider Sequenzen misst. Dabei bezieht sich der Begriff "Wort" hier auf eine Subsequenz der Länge $k = 1, \ldots, 10$, wobei diese Wörter innerhalb der Sequenzen um $k - 1$ Positionen überlappen. Die Wortähnlichkeit wurde in [38] als Quadrat der Summe der übereinstimmenden Aminosäuren (an identischen Wortpositionen) in beiden Wörtern definiert. Mit diesem Wortähnlichkeitsmaß und der dazugehörigen Wortrepräsentation wird in [38] gezeigt, dass die Kernfunktion durch Anwendung algebraischer Transformationen auf eine Darstellbarkeit der einzelnen Sequenzen als Wortkorrelationsmatrizen (WKM) führt. Dabei enthält eine WKM die kumulierten Wortähnlichkeiten aller Wörter einer Sequenz. Durch Vektorisierung dieser Matrizen – also durch "Stapeln" der einzelnen Spalten – ergibt sich eine explizite Vektorrepräsentationsmethode für Proteinsequenzen. Im korrespondierenden Merkmalsraum entspricht eine Dimension dann der Häufigkeit zweier bestimmter Amino-

säuren an bestimmten Wortpositionen in allen Wörtern einer Sequenz. Auch hier kann – wie bei der ODH-Methode – durch Normierung der Merkmalsvektoren die unterschiedliche Länge der Sequenzen berücksichtigt werden.

Der Merkmalsraum der WKM weist – bei Wahl der Wortlänge gemäß obiger Begriffseinführung – eine moderate Dimensionalität auf, da die Anzahl der verschiedenen Korrelationen quadratisch von der Wortlänge k abhängt. Da Wortkorrelationsmatrizen symmetrisch sind, werden zudem nur die Einträge der oberen Dreiecksmatrix benötigt. Der WKM-Merkmalsraum für Wortlänge $k = 3$ ($k = 10$) umfasst somit nur 1830 (20100) Dimensionen.

Der WKM-Merkmalsraum weist einen interessanten Bezug zum ODH-Merkmalsraum auf: Für eine Wortlänge k enthält der WKM-Merkmalsraum den Monomerdistanzhistogramm-Merkmalsraum mit der Maximaldistanz $k - 1$. Merkmale der Monomerdistanzhistogramme sind dabei mehrfach auf den Diagonalen einer Wortkorrelationsmatrix vertreten, z.B. erscheint das Aminosäurespektrum bei einer Wortlänge $k = 3$ dreimal auf der Hauptdiagonalen. Jedoch unterscheidet sich die Häufigkeit der Merkmale entsprechend der verschiedenen Wortpositionen in einer Sequenz. Der WKM-Merkmalsraum kann somit in gewisser Hinsicht als Verallgemeinerung des ODH-Merkmalsraums angesehen werden.

Der WKM-Merkmalsraum für Wortlängen $k \geq 2$ enthält mit der obigen Definition des Wortähnlichkeitsmaßes auch den Merkmalsraum des k-mer-Spektrums für $k = 1, 2$. Für $k = 1$ entsprechen die Merkmalsräume beider Methoden der (relativen) Aminosäurehäufigkeit. Im Gegensatz zum k-mer-Spektrum enthält der WKM-Merkmalsraum zu einer Wortlänge k jedoch auch die Merkmalsräume, welche mit kleineren Wortlängen assoziiert sind. Damit kann das in 2.1.1 geschilderte Problem abnehmender exakter Übereinstimmungen von k-meren elegant umgangen werden.

2.2.1 Performanz

Die Evaluation der WKM-Methode auf dem Testdatensatz zur Detektion entfernt verwandter Sequenzen [29] zeigte ähnlich gute Ergebnisse wie die ODH-Methode [38]. Die beste Performanz wurde für die Wortlänge $k = 6$ gemessen, jedoch zeigte die Evaluation, dass die Leistungsfähigkeit der WKM-Methode sowohl für kürzere als auch für längere Wörter nicht wesentlich schlechter ist. Die Wortlänge ist

der einzige Parameter dieser Methode.[2] Daher lassen diese Ergebnisse vermuten, dass eine umfassende Neuevaluation dieses Parameters bei Anwendung auf andere Probleme nicht notwendig ist.

Im vorigen Abschnitt wurde angedeutet, dass der WKM-Merkmalsraum "rekursiv" aufgebaut ist. Prinzipiell können beim kernbasierten Lernen durch Kombination der Kernmatrizen des k-mer-Spektrums entsprechend verschieden großer k unterschiedliche Merkmalsräume integriert werden. In [38] zeigte sich jedoch, dass diese "explizite" Kombination verschiedener Merkmalsräume der konzeptionellen Integration bei der WKM-Methode leistungsmäßig unterlegen ist.

2.2.2 Interpretierbarkeit der Merkmale

Die WKM-Methode ermöglicht eine umfangreiche Interpretation gelernter diskriminativer Merkmale. Nach dem Lernen der Diskriminante kann der diskriminative Gewichtsvektor[3] in Form einer diskriminativen Wortkorrelationsmatrix dargestellt werden. Dies ermöglicht die Identifikation wichtiger Paare von Aminosäuren an bestimmten Wortpositionen in den Sequenzen der untersuchten Proteinfamilie. Abb. 2.2 zeigt die diskriminative WKM-Repräsentation eines Experiments aus [38] für die Wortlänge $k = 3$. Mithilfe dieser Abbildung kann man z.B. leicht die familienspezifisch überrepräsentierten gleichzeitigen Vorkommen der Aminosäure Glutamin (Q) an Wortposition 1 und 3 feststellen.

Weiterhin lässt sich bei der WKM-Methode durch "Scoring" (Gewichtung) eines einzelnen Wortes mit dem diskriminativen Gewichtsvektor (in WKM-Darstellung) die Diskriminativität dieses Wortes berechnen [38]. Jedem Wort kann so ein diskriminativer Wort-Score zugewiesen werden, wobei hohe positive und negative Wort-Scores auf eine Diskriminativität (z.B. Überrepräsentiertheit) des Wortes in den Sequenzen der untersuchten Proteinfamilie hindeuten. Die diskriminativsten Wörter einer Proteinfamilie können so z.B. durch Analyse der Aminosäureeigenschaften zur Bestimmung biologisch bedeutungsvoller Motive herangezogen werden.

Durch Berechnung aller aufeinanderfolgenden Wort-Scores der überlappenden Wörter einer Sequenz ergibt sich ein sequenzspezifisches Score-Profil. Dieses Profil kann dazu benutzt werden, um charakteristische oder diskriminative Regionen

[2] Andere Wortähnlichkeitsmäße oder Wortrepräsentation werden hier nicht betrachtet.

[3] Bei kernbasiertem Lernen muss dieser zuerst aus den sequenzspezifischen Gewichten und den Merkmalsvektoren gewonnen werden.

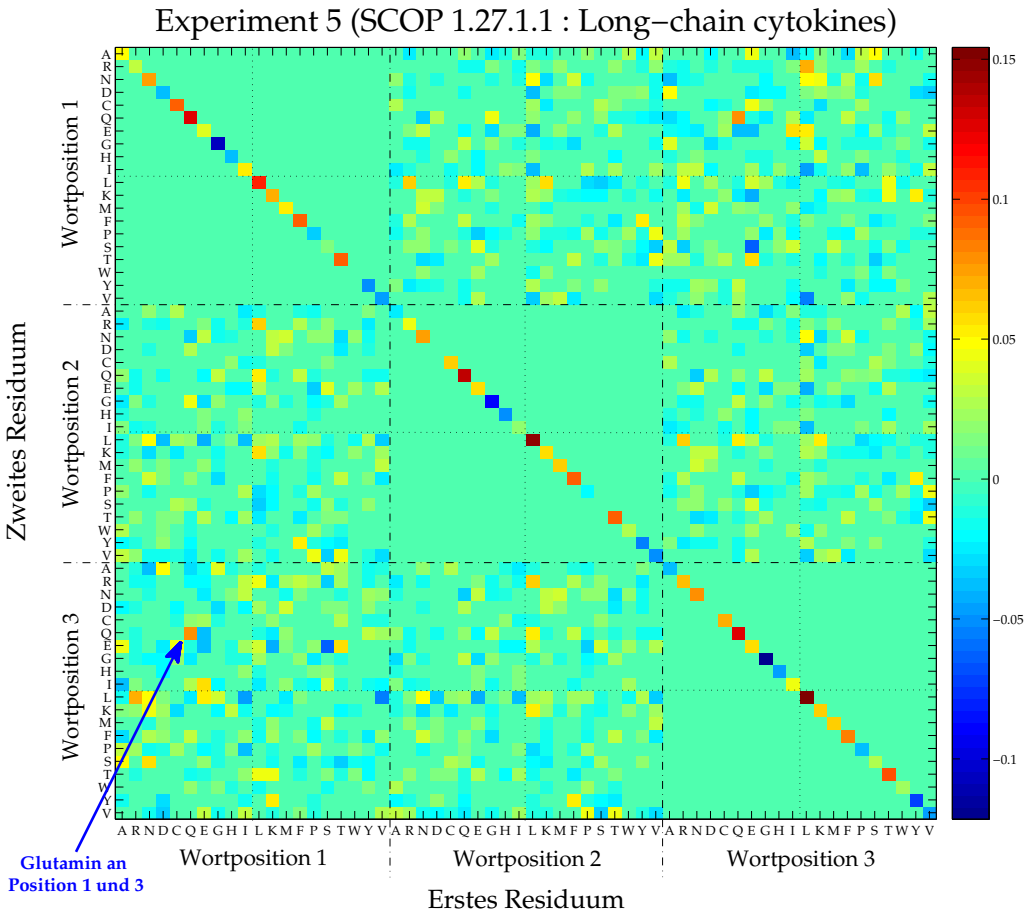

Abbildung 2.2: Diskriminativer Gewichtsvektor entsprechend Experiment 5 aus [38] (entspricht SCOP-Familie 1.27.1.1) in der Wortkorrelationsmatrix-Darstellung ($k = 3$). Die Farbwerte der Gewichte sind entsprechend der Farbskala auf der rechten Seite kodiert.

in den Sequenzen der untersuchten Proteinfamilie zu identifizieren. Zur besseren Interpretierbarkeit können diese Score-Profile visualisiert werden (siehe Abb. 2.3). Die Analyse der Score-Profile ermöglicht auch die Verfeinerung der aufgrund der diskriminativen Wörter gewonnenen biologisch bedeutungsvollen Motive. Beispielsweise können diskriminative Wörter, welche oft in ähnlichen Sequenzregionen vorkommen und ein ausgedehntes lokales Profilmaximum bilden, zu längeren Motiven zusammengefasst werden.

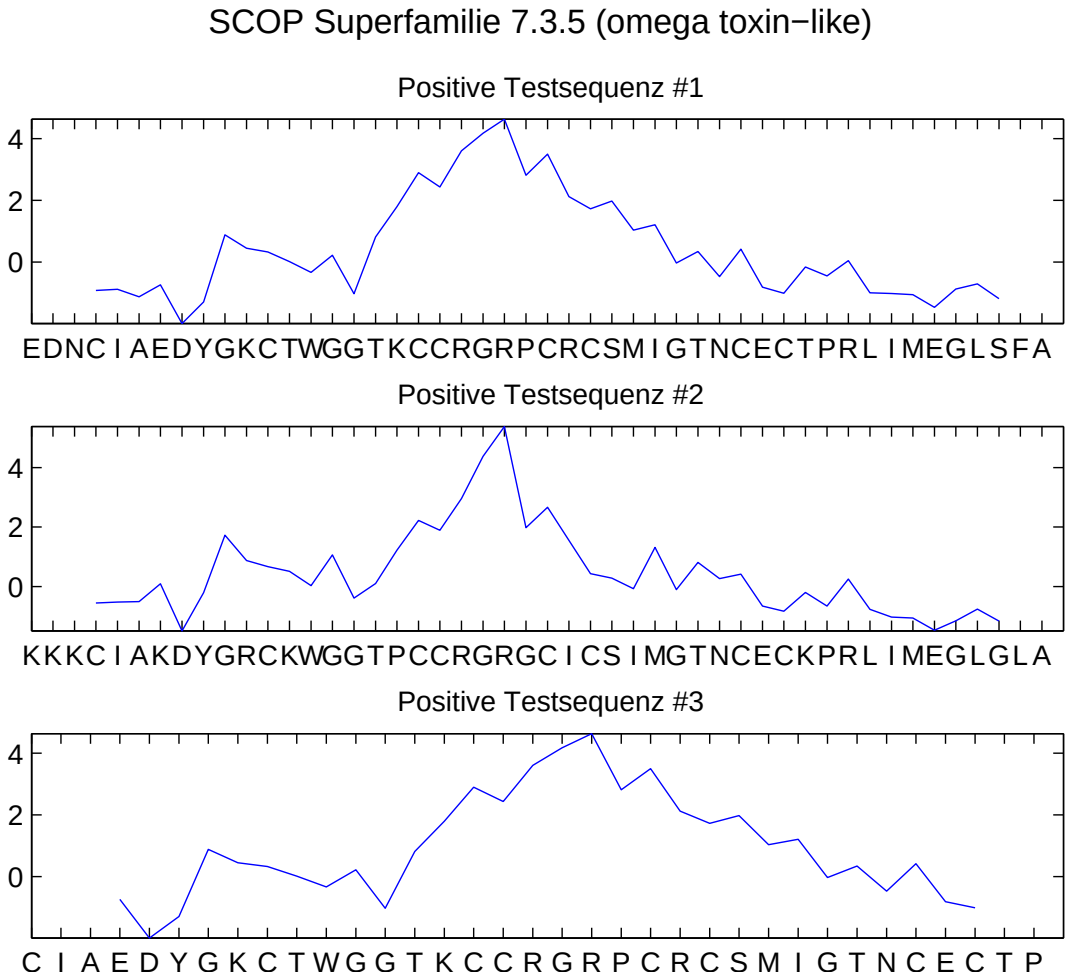

Abbildung 2.3: Score-Profile (der diskriminativen Wort-Scores) entsprechend der ersten drei Testsequenzen eines Experiments aus [38] unter Verwendung der Wortlänge $k = 6$. Die einzelnen Wort-Scores zur Erstellung des Score-Profils sind hier um die Wortposition 4 zentriert.

2.2.3 Rechentechnische Effizienz

Die WKM-Methode weist bezüglich der rechentechnischen Effizienz ähnlich positive Eigenschaften auf wie die ODH-Methode. Auch hier ist im Fall des kernbasierten Lernens eine schnelle Berechnung der Kernmatrix durch Transformation der Sequenzen in den Merkmalsraum und Anwendung des Matrixprodukts möglich. Hierbei kommt die Zerlegung der Kernfunktion zweier Sequenzen in sequenzspezifische WKMs besonders zur Geltung. Da die originäre Definition des Sequenzkerns die Berechnung der Wortähnlichkeiten aller Wörter einer Sequenz zu allen

anderen Wörtern der anderen Sequenz erfordert, hängt der Aufwand zur Berechnung der Kernfunktion quadratisch von der Sequenzlänge ab.[4] Der Aufwand zur Berechnung der WKM einer Sequenz hängt dagegen nur linear von der Sequenzlänge ab. Zwar hängt der Aufwand nun quadratisch von der Wortlänge k ab, jedoch ist diese (bei sinnvoller Wahl) sehr viel kleiner als die Sequenzlänge. In [38] konnte auf diese Weise der Berechungsaufwand der Kernmatrix für 1000 exemplarische Sequenzen und Verwendung der Wortlänge $k = 5$ von fast 10 Minuten auf ca. 3 Sekunden reduziert werden.

Auch die Anwendbarkeit der Diskriminante im Merkmalsraum zur schnellen Klassifikation neuer Sequenzen ist analog zur ODH-Methode möglich. Weiterhin kann bei der Verwendung der oben erwähnten Wortrepräsentation der Klassifikations-Score einer neuen Sequenz ohne explizite Transformation in den Merkmalsraum durch direkte Inspektion der Wörter berechnet werden. Dabei entspricht der Score einer Sequenz der Summe der diskriminativen Gewichtsvektorelemente, welche mit vorkommenden Aminosäurepaaren an bestimmten Wortpositionen (aller Wörter) assoziiert sind. In [38] konnte gezeigt werden, dass die Klassifikation mit der WKM-Methode eine Beschleunigung um das ca. 10000-fache gegenüber der Klassifikation mit dem Local-Alignment-Kernel ermöglicht. Für die Analyse großer Sequenzmengen kann dieser Faktor entscheidend für die praktische Durchführbarkeit sein.

2.3 Proteinfunktionsvorhersage

Aufgrund der zahlreichen Aspekte des Begriffs "Proteinfunktion" (siehe Abschnitt 1) existieren verschiedene Herangehensweisen für die Proteinfunktionsvorhersage. Homologiebasierte Ansätze stützen sich auf die Suche mit (paarweisen) Alignmentmethoden und den Transfer der Annotation, z.B. in Form von GO-Kategorien (z.B. [42], für einen Überblick siehe [5]). Hierbei gelten jedoch die in Abschnitt 1.1.1 erwähnten Probleme des homologiebasierten Annotationstransfers. PHMM-Modelldatenbanken wie Pfam werden standardmäßig zur Annotation von Genomen [43] und Metagenomen [2] benutzt, sind aber sehr rechenaufwändig. Neuere (alignmentfreie) Ansätze auf Grundlage von Methoden des maschinellen Lernens sind teilweise sehr recheneffizient, berücksichtigen aber oft nur Teilaspek-

[4] Aus Gründen der Vereinfachung wird hier von einer gleichen (durchschnittlichen) Länge der Sequenzen ausgegangen.

te zur Gewinnung von Hinweisen auf die Funktion, z.B. die Vorhersage von Faltungsmustern von Proteinen (z.B. [44,45]) oder die Detektion (entfernt) homologer Sequenzen (z.B. [30,40,46]). Bei der Evaluation dieser Ansätze werden üblicherweiser nur wenige Kategorien verwendet, z.B. 54 SCOP-Superfamilien im weit verbreiteten Testdatensatz aus [29] oder 46 Enzymfamilien in [47]. Des Weiteren sind die Testdatensätze oft auch nicht repräsentativ, z.B. besteht die SCOP-Datenbank – die vielen Testdatensätzen zugrunde liegt – fast ausschließlich aus Proteinen mit nur einer Domäne.[5] Während die Testdatensätze noch zur Evaluation der Methoden geeignet sind, sagen die Ergebnisse der Evaluation wenig über die praktische Nützlichkeit der Ansätze aus. Die geringe Abdeckung funktionaler Kategorien führt dazu, dass bei der Annotation großer Sequenzmengen (z.B. Genomen) nur diese Kategorien detektiert werden können. Zudem wurden die Hyperparameter der Methoden oft auf Basis der Testdatensätzen eingestellt und erfordern somit bei anderen Datenmengen eine erneute, im Allgemeinen aufwändige Anpassung.

Die Pfam-Datenbank [10] weist eine sehr hohe Abdeckung funktionaler Vielfalt auf und wird beständig und sorgfältig erweitert [50]. Obwohl Pfam und HMMER inzwischen standardmäßig zur Annotation benutzt werden, wurde Pfam bisher nocht nicht in vollem Maße für Evaluationszwecke verwendet. Um die Nützlichkeit der in den vorigen Abschnitten beschriebenen Repräsentationsmethoden für die Proteinfunktionsvorhersage zu evaluieren, wurde daher in [39] ein Testdatensatz erstellt, welcher eine rigorose Evaluation auf einem großen Teil der Pfam-Datenbank ermöglicht. Die Proteinfunktionsvorhersage wird dabei durch ein Klassifikationsproblem gemäß der Pfam-Proteinfamilien realisiert.

Der Testdatensatz spiegelt viele Aspekte der Proteinfunktionsvorhersage wider und stellt hohe Ansprüche an das verwendete Lernverfahren und die zur Evaluation verwendeten Gütemaße. So weist Pfam (und somit auch der Testdatensatz) enorme Größenunterschiede der Proteinfamilien auf.[6] Dies erfordert für solch "unbalancierte" Kategorien geeignete Methoden des maschinellen Lernens und spezielle Gütemaße zur Evaluation der Methoden. Weiterhin kann ein Protein aus mehreren Domänen bestehen und folglich in diesem Datensatz unter Umständen mehreren Familien gleichzeitig zugeordnet werden. Dementsprechend kom-

[5] Der Grund dafür ist, dass SCOP auf der Strukturdatenbank PDB [48] basiert. Da kleine Proteine leichter strukturell bestimmbar sind und oft nur eine Domäne enthalten, sind diese in der PDB- und SCOP-Datenbank überrepräsentiert [49].

[6] Die Größenunterschiede begründen sich durch die unterschiedliche natürliche Reichhaltigkeit und unterschiedlich fortgeschrittene Aufklärung der Proteinfamilien.

men nur Lernverfahren (und Evalutionsgütemaße) in Betracht, welche für dieses "Multilabel-Problem" ausgelegt oder dementsprechend erweiterbar sind.

Von den 9318 Proteindomänenfamilien mit insgesamt 217445 von Experten für repräsentativ befundenen Sequenzen in Pfam 22.0 (veröffentlicht im Juli 2007) wurden in [39] 4423 Familien für eine 5-fach Kreuzvalidierung verwendet. Diese große Anzahl von Kategorien und Sequenzen impliziert bei bisherigen Ansätzen zur diskriminativen Klassifikation große Nachteile bezüglich der rechentechnischen Durchführbarkeit. Üblicherweise werden diskriminative Multiklassen-Probleme mit M Klassen gelöst, indem M Diskriminanten mit der "Eine-gegen-den-Rest" (one-against-all) Strategie gelernt werden [51]. Dies ist bei Tausenden von Kategorien nur unter Einbezug massiver Parallelisierung – bei entsprechender Ausstattung – praktikabel. Die hinlänglich verwendeten kernbasierten Methoden sind bei Größenordnungen von 10^5 Sequenzen ebenfalls nur mit Hochleistungsrechnern praktisch verwendbar.

Die in den vorigen Abschnitten vorgestellten Repräsentationsmethoden sind in Verbindung mit einem recheneffizienten Lernverfahren besonders zur Analyse großer Sequenzmengen geeignet. Die "Regularized-Least-Squares"-Methode (RLSQ, [52]) ist ein mathematisch simpler, aber dennoch leistungsfähiger Ansatz zum effizienten Lernen von großen Beipielmengen in moderat dimensionierten Merkmalsräumen. In vergleichenden Studien hat sich die RLSQ-Methode als ähnlich leistungsfähig wie die weit verbreiteten SVMs herausgestellt [53]. In [39] wurde die RLSQ-Methode auf die Anforderungen des Pfam-Testdatensatzes angepasst, so dass alle Diskriminanten gleichzeitig unter Berücksichtigung verschieden umfangreicher Kategorien gelernt werden können. Dabei werden die Diskriminanten direkt im Merkmalsraum gelernt, so dass im Gegensatz zu kernbasierten Lernmethoden keine anschließende Transformation notwendig ist, um die Diskriminante zur schnellen Annotation neuer Sequenzen zu verwenden. Außerdem ist die Lernmethode in [39] direkt für die Lösung von Multilabel-Problemen geeignet und stützt sich bei der Vorhersage nur auf die Verwendung effizient implementierter Matrixalgebra.

Die angepasste RLSQ-Lernmethode ist mit verschiedenen Repräsentationsmethoden verwendbar, wobei diese eine moderate Dimensionalität aufweisen müssen, damit das Training rechentechnisch effizient lösbar ist. In [39] wurden das k-mer-Spektrum für $k = 1, 2, 3$ und die Monomerdistanzhistogramme für die Maximaldistanzen $D = 10, 20, 30$ auf dem Pfam-Testdatensatz evaluiert. Hierfür wur-

den spezielle, für unbalancierte Multilabel-Probleme geeignete Performanzmaße verwendet.[7] Bei den Ergebnissen zeigten die Monomerdistanzhistogramme durchgängig eine bessere Performanz als die verschiedenen Realisierungen des k-mer-Spektrums. Außerdem stellte sich heraus, dass das Monomerspektrum, welches eine beachtliche Performanz auf dem Testdatensatz in [29] zeigte, nicht zur Proteinfunktionsvorhersage auf dem Pfam-Testdatensatz geeignet ist. Eine mögliche Erklärung dafür liefert der nur 20 Dimensionen umfassende Merkmalsraum des Monomerspektrums. Offenbar ist die Dimensionalität zu gering, um Tausende von Proteinfamilien linear zu trennen. Andererseits kann die gute Performanz des Monomerspektrums auf dem Testdatensatz aus [29] auch auf der angesprochenen Verzerrtheit des SCOP-Datensatzes beruhen.

Für längere Oligomere eignet sich das k-mer-Spektrum zunehmend für die in [39] untersuchte Klassifikation von Proteinsequenzen in Pfam-Proteinfamilien. Allerdings sind k-mer-Spektrum-Merkmalsräume mit $k > 3$ aufgrund der hohen Dimensionalität nicht mehr mit dem vorgestellten Lernverfahren verwendbar. Die WKM-Methode dagegen weist selbst für größere k-mere eine moderate Dimensionalität auf, daher wurde auch diese Methode auf dem Pfam-Testdatensatz evaluiert.[8] In Tabelle 2.1 ist die Performanz der WKM-Methode auf dem Pfam-Testdatensatz für verschiedene Wortlängen k abgebildet. Im Vergleich zum Trimerspektrum ist die Coverage der WKM-Methode schon für $k > 4$ besser, jedoch sind Wörter mit $k > 9$ (und somit große Merkmalsräume) nötig, um eine ähnlich hohe ROC50-Performanz wie die Spektrum-Methode zu erzielen.[9] Insgesamt reicht die Performanz beider Methoden jedoch nicht an die Leistungsfähigkeit der ODH-Methode heran, was auf die Relevanz der Wahl eines geeigneten Merkmalsraums zur Repräsentation der Sequenzen hindeutet. Der ODH-Merkmalsraum für Monomere mit beschränkter Maximaldistanz bietet hier eine geeignete Möglichkeit, große Mengen von Proteinsequenzen für die Klassifikation in viele funktionale Kategorien zu

[7] Dazu gehört z.B. die sogenannte "Coverage" (Abdeckung) – ein Maß für die Anzahl an Kategorien, die man mit der untersuchten Methode im Mittel berücksichtigen muss, um alle wahren Kategorien für ein Beispiel in der Vorhersage einzuschließen.

[8] Da die Wortkorrelationsmatrix-Methode zum Zeitpunkt der Einreichung des Artikels zur Proteinfunktionsvorhersage noch nicht veröffentlicht war, fehlen die Ergebnisse der WKM-Methode in [39].

[9] Die ROC50-Performanz kann als Maß für die Spezifität einer Methode bezeichnet werden. Da innerhalb der Pfam-Proteinfamilien oftmals längere Oligomere ($k \geq 3$) konserviert sind, ermöglicht die exakte Repräsentation von Trimeren mit der 3-mer-Spektrum-Methode spezifischere Vorhersagen.

Methode	d	Coverage		One-error	ROC	ROC50
		mean	median			
WKM ($k = 1$)	210	452.42	243.8	0.95	0.925	0.046
WKM ($k = 2$)	820	221.0	63.4	0.86	0.975	0.421
WKM ($k = 3$)	1830	155.3	34.4	0.79	0.983	0.588
WKM ($k = 4$)	3240	124.1	21.4	0.73	0.987	0.679
WKM ($k = 5$)	5050	104.1	14.0	0.69	0.990	0.735
WKM ($k = 6$)	7260	92.8	10.2	0.65	0.991	0.767
WKM ($k = 7$)	9870	83.2	7.6	0.62	0.992	0.790
WKM ($k = 8$)	12880	75.4	5.6	0.59	0.992	0.809
WKM ($k = 9$)	16290	69.6	4.4	0.57	0.993	0.823
WKM ($k = 10$)	20100	65.1	4.0	0.55	0.993	0.834
Spektrum ($k = 3$)	8000	116.7	4.8	0.57	0.987	0.827
MDH ($D_{max} = 30$)	12020	41.6	1.2	0.37	0.995	0.894

Tabelle 2.1: Performanz der WKM-Methode auf dem Pfam-Testdatensatz aus [39] für verschiedene Wortlängen $k = 1, \ldots, 10$ im Vergleich zum k-mer-Spektrum für $k = 3$ und den Monomerdistanzhistogrammen (MDH) mit Maximaldistanz $D = 30$. Die erste Spalte kennzeichnet die Methode und den verwendeten Parameter, die zweite Spalte gibt die Dimensionalität des zugehörigen Merkmalsraums an. Die Spalten 3-7 stehen für verschiedene Performanzindizes, welche in [39] ausführlich beschrieben sind.

repräsentieren. Dabei ist anzumerken, dass der Parameter für die Maximaldistanz auf neuen Datenmengen nicht neu evaluiert werden muss, da aufgrund der Tendenz bei der Performanz einfach die größte Maximaldistanz gewählt werden kann, welche mit der zur Verfügung stehenden Rechenanlage praktikabel ist. Ähnliches gilt für den Wortlängenparameter der WKM-Methode. Im Gegensatz zum Problem der Detektion entfernter Homologien gibt es hier bezüglich der Performanz offenbar keine sinnvolle obere Schranke für k innerhalb des getesteten Intervalls. Eine Erklärung dafür liefert die gegenüber den SCOP-Superfamilien engere evolutionäre Verwandtschaft der Sequenzen innerhalb der Pfam-Proteinfamilien, welche eine stärkere Konserviertheit längerer Sequenzregionen impliziert.

Mit dem Testdatensatz in [39] kann die prinzipielle Eignung einer Methode zur Proteinfunktionsvorhersage – innerhalb des Pfam-Klassifikationsschemas – gemes-

sen werden. Allerdings sind die dort vorgestellten Methoden nicht ohne weiteres zur praxisgerechten Klassifikation von Proteinsequenzen in funktionale Kategorien verwendbar, da die Kategorien bei der Vorhersage lediglich entsprechend des Vorhersage-Scores sortiert werden. Bei der Multilabel-Klassifikation reicht jedoch die Auswahl des Treffers mit dem höchsten Vorhersage-Score nicht aus, da so bei Beispielen mit mehreren Funktionen zwangsläufig eine geringere Sensitivität erzielt wird. Zur adäquaten Vorhersage wäre zusätzlich eine Methode zur Ermittlung der Anzahl der relevanten Kategorien notwendig, z.B. eine Kalibrierung der Vorhersage-Scores zur Bestimmung eines Score-Schwellwerts.

Andererseits ist ein Abschneiden der Trefferliste (also der nach Vorhersage-Score sortierten Kategorien) in der Praxis oft nicht notwendig, da eine manuelle Inspektion der Treffer (z.B. bezüglich der Konsistenz) meist unerlässlich ist und eine Sortierung somit eine wesentliche Aufwandserleichterung darstellt. Weiterhin kann mit spezifischeren, aber auch rechenaufwändigeren alignmentbasierten Methoden wie z.B. HMMER die Signifikanz der Treffer bis zu einer festen – z.B. anhand der "Coverage" vorher ermittelten – Anzahl M evaluiert werden. Damit müssen nicht mehr alle, sondern nur noch die mit den M höchstrangigen Kategorien assoziierten Modelle mittels Alignments evaluiert werden. Auf diese Weise stellt die Sortierung eine "Zielmengenreduktion" der Anzahl notwendiger Alignments dar, was eine effiziente Verwendung der alignmentbasierten Methoden erlaubt. In [39] wurde gezeigt, dass mit den Monomerdistanzhistogrammen mit beschränkter Maximaldistanz und Verwendung des angepassten RLSQ-Verfahrens eine Beschleunigung der HMMER-gestützten Proteinfunktionsvorhersage um das ca. 100-fache möglich ist. Dies kann insbesondere bei Metagenomanalysen in Verbindung mit schnellen Methoden zur Genvorhersage (z.B. [54]) eine Reduktion des Zeitaufwands von mehreren Wochen auf wenige Stunden bedeuten.

Kapitel 3

Fazit und Ausblick

Die im vorherigen Abschnitt diskutierten Evaluationsergebnisse haben gezeigt, dass die Ziele der Arbeit erfüllt wurden. Die beiden vorgestellten Repräsentationsmethoden für Proteinsequenzen – Oligomerdistanzhistogramme und Wortkorrelationsmatrizen – haben sich bei der Evaluation zweier unterschiedlicher Probleme (Detektion entfernter Homologien und Proteinfunktionsvorhersage) als leistungsfähige Ansätze zur Analyse von Proteinsequenzen herausgestellt. Dabei ermöglichen beide Methoden eine intuitive Interpretation der gelernten diskriminativen Merkmale und unterstützen somit die Untersuchung charakteristischer Eigenschaften, z.B. von Proteinfamilien. Die Beschränkung auf wenige Methodenparameter macht die Ansätze für viele Probleme ohne Anpassung nutzbar und die explizite Repräsentation der Merkmale und Diskriminanten in einem Vektorraum erschließt die effiziente Anwendung auf große Beispielmengen.

Der vorgestellte Testdatensatz zur Proteinfunktionsvorhersage innerhalb des Pfam-Klassifikationsschemas ermöglicht die umfassende Evaluation von Methoden des maschinellen Lernens auf Proteinsequenzdaten in einem biologisch relevanten Setup. Dabei spiegelt der Testdatensatz viele Probleme wider, die in bisherigen Ansätzen zur Evaluation von Methoden zur Proteinfunktionsvorhersage nur unzureichend vertreten sind. Die Praxistauglichkeit der hier vorgestellten Repräsentationsmethoden für Proteinsequenzen wurde auf diesem Datensatz evaluiert und belegt.

Die in dieser Arbeit vorgestellten Ansätze bieten viele Möglichkeiten für weitere Untersuchungen. So wurden die Repräsentationsmethoden zunächst unter Verwendung einfacher Aminosäuremerkmale untersucht, wobei verschiedene Aminosäuren mit unterschiedlichen Dimensionen in einem Merkmalsraum assoziiert

sind. Da Aminosäuren aber aufgrund evolutionärer Mutationen oder bezüglich der biochemischen Eigenschaften zu Gruppen zusammengefasst werden können, bietet sich bei den Repräsentationsmethoden die Verwendung alternativer Alphabete zur Definition der Merkmalsräume an. Allerdings existieren bei der Untersuchung von Aminosäuremerkmalen auch verschiedene anwendbare Alphabete, so dass mit diesem zusätzlichen Methodenparameter umfassende Evaluationen notwendig sind.

Bei den Oligomerdistanzhistogrammen ist weiterhin die Zusammenfassung mehrerer Distanzen zu "Distanzgruppen" denkbar, so dass die Histogramme weniger Einträge aufweisen und gleichzeitig eine "Abstandsunsicherheit" repräsentieren können. Dies impliziert jedoch auch mindestens einen zusätzlichen Hyperparameter, was die Gefahr der Überanpassung an ein bestimmtes Problem und Nichtübertragbarkeit auf andere Probleme beinhaltet.

Die Wortkorrelationsmethode lässt sich außer mit den aus alternativen Alphabeten resultierenden Wortrepräsentationen auch mit Aminosäure-Substitutionsmatrizen verwenden. Diese können in Form einer zentralen Transformationsmatrix direkt im Wortähnlichkeitsmaß verwendet werden. Somit könnte – unter Einführung eines weiteren Hyperparameters – auf evolutionäre Besonderheiten des zu untersuchenden Problems eingegangen werden.

Da die vorgestellten Repräsentationsmethoden alignmentfrei sind und somit komplementär zu alignmentbasierten Methoden, kann eine kombinierte Verwendung beider Methodenprinzipien hilfreich für die Verifikation von weniger signifikanten Ergebnissen einer Methode sein. Prinzipiell können beide Repräsentationsmethoden nach Anpassung der Alphabete und Reevaluation der Parameter auch für die Analyse von DNA- oder RNA-Daten verwendet werden.

Um den vorgestellten Ansatz zur Proteinfunktionsvorhersage zu einer eigenständigen Anwendung auszubauen, ist die Erweiterung um eine Methode zur Bestimmung der Anzahl der relevanten Funktionsklassen aus den Vorhersage-Scores erforderlich. Hierfür ist eine umfassende Evalution verschiedener Methoden zur Score-Kalibrierung und der Ermittlung von Score-Schwellwerten notwendig. Ein anderer Ansatz, welcher dem Vorgehen bei der manuellen Überprüfung von Trefferlisten ähnelt, ist die statistische Auswertung von Treffern entsprechend ihres Vorhersage-Scores. Hierbei kann z.B. die Häufigkeit bestimmter mit den Treffern assoziierter Annotationskategorien entsprechend der Treffergewichte (also ihrer Vorhersage-Scores) verwendet werden. Dabei erfordert diese Erweiterung eine

Evaluation unter Beteiligung biologischer Experten, da die Inspektion der Trefferlisten umfangreiches Wissen über Proteinfamilien und ihre Funktionen erfordert.

Auch der Testdatensatz selbst kann erweitert werden, z.B. indem nicht nur die von Experten zur Definition der Pfam-Familien verwendeten Sequenzen, sondern auch alle Sequenzen, welche signifikante Treffer zu den mit den Familien assoziierten PHMM-Modellen darstellen, berücksichtigt werden. Dies impliziert jedoch ein Wachstum der Sequenzmenge um eine Größenordnung. Weiterhin kann der Testdatensatz mit Erscheinen neuer Versionen der Pfam-Datenbank aktualisiert werden, um die höchstmögliche Abdeckung bezüglich bekannter Proteinfamilien zu gewährleisten.

Anhang A

Artikel 1

BIOINFORMATICS ORIGINAL PAPER Vol. 22 no. 18 2006, pages 2224–2231
doi:10.1093/bioinformatics/btl376

Sequence analysis

Remote homology detection based on oligomer distances

Thomas Lingner* and Peter Meinicke

Abteilung Bioinformatik, Institut für Mikrobiologie und Genetik, Georg-August-Universität Göttingen, Goldschmidtstr. 1, 37077 Göttingen, Germany

Received on March 30, 2006; revised on June 20, 2006; accepted on July 5, 2006
Advance Access publication July 12, 2006
Associate Editor: Christos Ouzounis

ABSTRACT

Motivation: Remote homology detection is among the most intensively researched problems in bioinformatics. Currently discriminative approaches, especially kernel-based methods, provide the most accurate results. However, kernel methods also show several drawbacks: in many cases prediction of new sequences is computationally expensive, often kernels lack an interpretable model for analysis of characteristic sequence features, and finally most approaches make use of so-called hyperparameters which complicate the application of methods across different datasets.

Results: We introduce a feature vector representation for protein sequences based on distances between short oligomers. The corresponding feature space arises from distance histograms for any possible pair of *K*-mers. Our distance-based approach shows important advantages in terms of computational speed while on common test data the prediction performance is highly competitive with state-of-the-art methods for protein remote homology detection. Furthermore the learnt model can easily be analyzed in terms of discriminative features and in contrast to other methods our representation does not require any tuning of kernel hyperparameters.

Availability: Normalized kernel matrices for the experimental setup can be downloaded at www.gobics.de/thomas. Matlab code for computing the kernel matrices is available upon request.

Contact: thomas@gobics.de, peter@gobics.de

1 INTRODUCTION

Protein homology detection is a central problem in computational biology. The objective is to predict structural or functional properties of proteins by means of homologies, i.e. based on sequence similarity with phylogenetically related proteins, for which these properties are known.

For proteins with high sequence similarity according to >80% identity at the amino acid level, homologies can easily be found by pairwise sequence comparison methods like BLAST (Altschul *et al*., 1990) or the Smith–Waterman local alignment algorithm (Smith and Waterman, 1981). However, in many cases these methods fail because more subtle sequence similarities, so-called remote homologies, have to be detected.

Recently, many approaches challenged this problem with increasing success. The corresponding methods are usually based on a suitable representation of protein families and can be divided into two major categories: on one hand protein families can be represented by generative models which provide a probabilistic measure of association between a new sequence and a particular family. In this case, so-called profile hidden markov models (e.g. Krogh *et al*., 1994, Park *et al*., 1998) are usually trained in an unsupervised manner using only known example sequences of the particular family. On the other hand discriminative methods can be used to focus on the differences between protein families. In that case kernel-based support vector machines are usually trained in a supervised manner using example sequences of the particular family as well as counter-examples from other families. Recent studies (Jaakkola *et al*., 2000, Liao and Noble, 2002, Leslie *et al*., 2004) have shown that an explicit representation of sequence differences between different protein families is important for remote homology detection and that kernel methods can significantly increase the detection performance as compared with generative approaches.

A kernel computes the inner product between two data elements in some abstract feature space, usually without an explicit transformation of the elements into that space. Using learning algorithms which only need to evaluate inner products between feature vectors, the 'kernel trick' makes learning in complex and high-dimensional feature spaces possible. Kernels for remote homology detection provide different ways for evaluation of position information in protein sequences. Many approaches, like spectrum (Leslie *et al*., 2002) or motif (Ben-Hur and Brutlag, 2003) kernels, do not consider position information since feature vectors are merely based on counting occurrences of oligomers or certain motifs in a particular sequence.

Other kernels are based on the concepts of pairwise alignment and therefore they provide a biologically well-motivated way to consider position-dependent similarity between a pair of sequences. In recent studies on benchmark data, position-dependent kernels showed the best results (Saigo *et al*., 2004).

Despite their state-of-the-art performance, recent alignment-based kernels show a significant disadvantage concerning the interpretability of the resulting discriminant model. Unlike spectrum or motif kernels, alignment-based kernels do not provide an intuitive insight into the associated feature space for further analysis of relevant sequence features which have been learnt from the data. Therefore these kernels do not offer additional utility for researchers interested in finding the characteristic features of protein families. Furthermore alignment-based kernels generally require the evaluation of all relevant kernel functions for classification of new sequences. Therefore in case of a large number of relevant kernel functions detection of homologies in large databases is computationally demanding. As another disadvantage of recent

*To whom correspondence should be addressed.

alignment-based kernels one may view the incorporation of hyperparameters which by definition cannot be optimized on the training set because they control the generalization performance of the approach. For the realization of the local alignment kernel, (Saigo *et al*., 2004) used a total number of three kernel parameters. While the dependence of the performance on one particular parameter was evaluated on the test data, the remaining two parameters were fixed in an ad hoc manner. Also other approaches, e.g. Dong *et al*. (2006) and Rangwala and Karypis (2005) comprise several hyperparameters which were optimized using the test data. It is often overlooked that the extensive use of hyperparameters bares the risk of adapting the model to particular test data. This fact complicates a fair comparison of different methods and the application of the method to different data because new data are likely to require readjustment of these parameters.

We here introduce an intuitively interpretable feature space for protein sequences which obviates the tuning of kernel hyperparameters and allows for efficient classification of new sequences. In this feature space sequences are represented by histograms for counting the occurrences of distances between short oligomers. These so-called oligomer distance histograms (ODH) provide the basis of our new representation which will be detailed in the following sections.

2 METHODS

Proteins are basically amino acid sequences of variable length and different steric constitution. Therefore absolute position information in terms of a direct comparison between residues at the same sequence position cannot be used with unaligned sequences in general. Therefore several methods for remote homology detection do not take into account any position information at all. A well-known example is the spectrum kernel (Leslie *et al*., 2002) which only counts the occurrences of K-mers in sequences. Obviously, a considerable loss of information may result from this restriction. Recently, several kernels based on the concepts of local alignment have been proposed to overcome the restriction of position-independent kernels. These alignment-based kernels actually consider position information within pairwise sequence comparisons and the results so far indicate that these kernels provide the state-of-the-art within the field of remote homology detection (Saigo *et al*., 2004).

In the context of promoter prediction it has been shown that characteristic distances between motifs associated with transcription factor binding sites provide useful information for the recognition of promoters (Ma *et al*., 2004). Now, the idea is that this kind of relative position information based on distances between motifs or oligomers may also provide a suitable representation for unaligned protein sequences.

2.1 Distance-based feature space

Our feature space for representation of protein sequences is based on histograms for counting distances between oligomers. For each pair of K-mers there exists a specific histogram counting the occurrences of that pair at certain distances. These distance histograms are 'naive' histograms with unit bin width and without any averaging or aggregation of neighboring bins. This implies, that all possible distances have their own bin. Consequently every bin gives rise to one particular feature space dimension. Finally the total feature space arises from the collection of all histograms from any possible pair of K-mers.

More specifically for the alphabet $\mathcal{A} = \{A, R, \ldots, V\}$ of amino acids we consider all K-mers $m_i \in \mathcal{A}^K$ with index $i = 1, \ldots, M$ according to an alphabetical order. For distinct K-mers m_i and m_j we distinguish between pairs (m_i, m_j) and (m_j, m_i) because we want to represent the order of oligomers occurring at a certain distance: for the pair (m_i, m_j) we only consider cases where oligomer m_i occurs before m_j. For a maximum sequence length $L_{\max}$ we have to consider a maximum distance $\mathrm{D} = L_{\max} - K$ between K-mers. Then we can build the M^2 distance histogram vectors of a sequence S according to

$$\mathbf{h}_{ij}(S) = [h_{ij}^0(S), h_{ij}^1(S), \ldots, h_{ij}^D(S)]^{\mathrm{T}}, \tag{1}$$

where T indicates transposition. In this representation an entry h_{ij}^d counts the occurrences of pair (m_i, m_j) at distance d. The distance is measured between the starting letters of K-mers. Note that h_{ij}^0 counts the occurrences of pair (m_i, m_j) at zero-distance. For $i = j$ this implies that the corresponding histogram vectors also count the number of K-mer occurrences in the sequence. Therefore the feature space associated with the above-mentioned spectrum kernel is completely contained in our representation, i.e. it actually is a subspace of the distance-based feature space. To realize the representational power of the distance-based feature space it is instructive to consider the simplest case of monomer distances: not only the feature space of the spectrum kernel for $K = 1$ is included in that representation, but also dimer counts ($d = 1$) and trimer counts ($d = 2$) according to a central mismatch are contained in the distance-based feature vectors.

The overall feature space transformation Φ of a sequence S is simply achieved by stacking all histogram vectors:

$$\Phi(S) = [\mathbf{h}_{11}^{\mathrm{T}}(S), \mathbf{h}_{12}^{\mathrm{T}}(S), \ldots, \mathbf{h}_{MM}^{\mathrm{T}}(S)]^{\mathrm{T}}. \tag{2}$$

For the final representation we normalize the feature vectors to have unit Euclidean length, in order to improve comparability between sequences of different length. In general, the resulting feature space dimensionality will be huge: e.g. for dimers with a maximum sequence length of $L_{\max} = 1000$ residues we have 400^2 histograms of length 999 which results in $\sim 1.6 \times 10^8$ dimensions. For trimers the distance-based feature space already comprises $\sim 6.4 \times 10^{10}$ dimensions. Though the feature space is very high-dimensional, the amount of memory required for the storage of the feature vectors can considerably be decreased if the sparse nature of these vectors is utilized. A sequence $S = s_1, \ldots, s_L \in \mathcal{A}^L$ contains a total number of $L - K + 1$ overlapping K-mers. For the maximum distance $L - K$ occurring in that sequence we obtain only one non-zero histogram entry concerning the oligomers $s_1, \ldots, s_K$ and $s_{L-K+1}, \ldots, s_L$. For smaller distances $L - K - \mathrm{q}$ in general we obtain at most $\mathrm{q} + 1$ non-zero entries. In total we get at most $1 + 2 + \cdots + (L - K + 1) = (L - K + 2) \cdot (L - K + 1)/2$ non-zero entries. This 'sparseness' allows for an explicit representation in terms of sparse vectors: e.g. considering dimer distances, for a sequence of length $\mathrm{L} = 400$ we have to compute at most 79 800 histogram entries. In technical terms, this corresponds to a minimum sparseness of 99.95% and a maximum allocation of 0.05%, respectively.

The feature space transformation of a sequence S can efficiently be realized by systematic evaluation of all pairwise K-mer occurrences in S. The following pseudocode shows a simple procedure for computation of a suitably initialized featureVector array and indicates the characteristic $O(L^2)$ complexity of the systematic evaluation scheme. The array indList contains the $L - K$ indices of the oligomers—e.g. index 0 for the first dimer $m_1 = AA$, index 1 for $m_2 = AR$ and so on—occurring at successive sequence positions. The list can be computed beforehand with algorithmic complexity $O(L)$. `M` and `D` correspond to the number of possible K-mers and the maximum distance, respectively.

```
for firstPos = 1 to length(indList)
    for secondPos = firstPos to length(indList)
        indJ = (M*D) * indList[firstPos]
        indK = D * indList[secondPos]
        indDist = secondPos - firstPos
        featureVector[indJ + indK + indDist] += 1
    end
end
```

2.2 Kernel-based training

While the explicit feature space representation is well-suited for analysis of relevant sequence characteristics (see section 'Results') it is not appropriate for the training of classifiers owing to the huge dimensionality. For that purpose a kernel-based representation of the discriminant function f is more suitable. Using the kernel function $k(\cdot\ ,\ \cdot)$ and sequence-specific weights $\alpha_1, \ldots, \alpha_N$ the discriminant function (with additive constant omitted) can be expressed by

$$f(S) = \mathbf{w}^{\mathrm{T}} \cdot \Phi(S) = \sum_{i=1}^{N} \alpha_i \cdot k(S, S_i), \tag{3}$$

according to the primal and dual representation of the discriminant (Schölkopf and Smola, 2002), respectively. In our case we first compute a sparse matrix of all feature vectors:

$$\mathbf{X} = [\Phi(S_1), \ldots, \Phi(S_N)]. \tag{4}$$

Then the $N \times N$ kernel matrix $\mathbf{K}$ with entries $k_{ij} = k(S_i, S_j)$ which contains all inner products on the training set can efficiently be computed by the sparse matrix product:

$$\mathbf{K} = \mathbf{X}^{\mathrm{T}}\mathbf{X}. \tag{5}$$

The above-mentioned normalization of feature vectors to unit length can then efficiently be realized by scaling the entries k_{ij} of the kernel matrix:

$$k'_{ij} = \frac{k_{ij}}{\sqrt{k_{ii} \cdot k_{jj}}}. \tag{6}$$

The normalized kernel matrix in turn can be used for training of kernel-based classifiers, e.g. support vector machines, which require optimization of the weights α_i. After training the discriminant weight vector in feature space can be computed by

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i \cdot \frac{\Phi(S_i)}{\sqrt{k_{ii}}}. \tag{7}$$

This weight vector can be used for fast classification of new sequences and for interpretation of the discriminant as we will show in the following section.

3 EXPERIMENTS AND RESULTS

In order to evaluate the performance of our method, we used a common dataset for protein remote homology detection (Liao and Noble, 2002). This set has been used in many studies of remote homology detection methods (Liao and Noble, 2002, Saigo *et al*., 2004, Leslie *et al*., 2004) and therefore it provides good comparability with previous approaches. The evaluation on this dataset requires to solve 54 binary classification problems at the superfamily level of the SCOP-hierarchy [Structural Classification Of Proteins, Murzin *et al*. (1995)]. In total, a subset of 4352 SCOP sequences was used to build the dataset. Each superfamily is represented by positive training and test examples which have been drawn from families inside the superfamily and by negative training and test examples which were selected from families in other superfamilies. Thereby the number of negative examples is much larger than that of the positive ones. In particular this situation gives rise to highly 'unbalanced' training sets.

To test the quality of our feature space representation based on distances between K-mers we utilize kernel-based support vector machines (SVM). Kernel methods in general require the evaluation of a kernel matrix including all inner products between training examples. To speed up computation we pre-calculated a complete kernel matrix based on all 4352 sequences for each oligomer length $K \in \{1, 2, 3\}$. Then for every experiment we extracted the required entries according to the setup of Liao and Noble (2002). In the evaluation we tested our method for monomer, dimer and trimer distances. All kernel matrices used for the evaluation can be downloaded in compressed text format from www.gobics.de/thomas.

Table 1. Classification results of oligomer distance histograms using monomers (K=1), dimers (K=2) and trimers (K=3) in comparison with local alignment (LA-eig) kernel (Saigo *et al*., 2004), SVM pairwise (Liao and Noble, 2002), mismatch string kernel (Leslis *et al*., 2004) and Fisher kernel (Jaakkola *et al*., 2000)

Method	Average ROC	Average ROC50	Average mRFP
Monomer-dist.	0.919	0.508	0.0664
Dimer-dist.	0.914	0.453	0.0659
Trimer-dist.	0.844	0.290	0.1352
LA-eig ($\beta = 0.5$)	0.925	0.649	0.0541
Pairwise	0.896	0.464	0.0837
Mismatch (5:1)	0.872	0.400	0.0837
Fisher	0.773	0.250	0.2040

For best comparability with other representations, we used the publicly available Gist SVM package (http://svm.sdsc.edu/) in order to exclude differences owing to particular realizations of the kernel-based learning algorithm. As described in Jaakkola *et al*. (2000) the Gist package implements a soft margin SVM which can be trained using a custom kernel matrix. Besides an activation of the 'diagonal factor' option in order to cope with the unbalanced training sets, we used the SVM entirely with default parameters.

To measure the detection performance of our method on the test data, we calculated the area under curve with respect to the receiver operating characteristics (ROC) and the ROC50 score, which is the area under curve up to 50 false positives. Besides these ROC scores we also computed the median rate of false positives (mRFP). The mRFP is the fraction of false positive examples, which score equal or higher than the median score of true positives. Consequently, smaller values are better than larger ones.

The results of our performance evaluation in terms of averaged values over 54 experiments are summarized in Table 1. For comparison with other approaches also the results published in Saigo *et al*. (2004) are shown in the table. The rates indicate that our method performs well for monomers ($K = 1$) and dimers ($K = 2$) with a slight decrease of the ROC scores for dimers. Owing to the extremely sparse feature space, for trimers the detection performance decreases significantly. While the length of the sequences and thus the number of possible oligomer pairs remains constant, the feature space dimensionality grows by orders of magnitude. This implies a nearly diagonal kernel matrix according to vanishing similarity between different protein sequences. Among all compared methods only the local alignment kernel yields a performance which is slightly better than that of the distance-based representations for monomers and dimers.

Figure 1 summarizes the relative performance of the compared methods. For each method the associated curve shows the number of superfamilies that exceed a given ROC score threshold ranging from 0 to 1. For oligomer distance histograms we used the

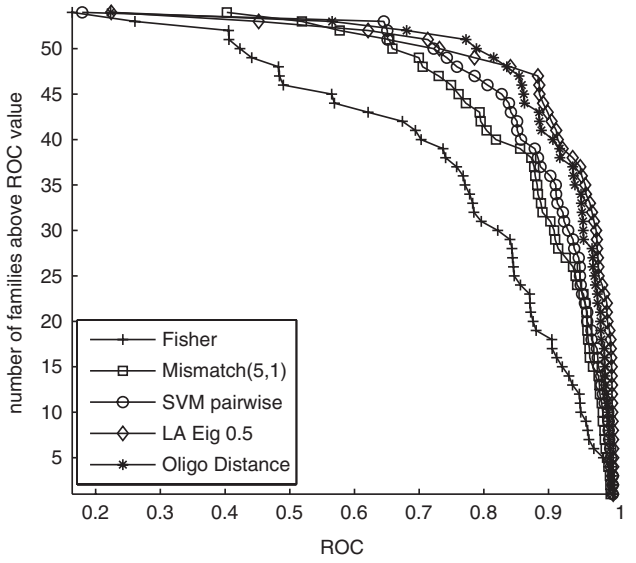

Fig. 1. ROC score distribution for different methods (see text), depending on the number of superfamilies (*y*-axis) above a given ROC score threshold (*x*-axis). For oligomer distance histograms (Oligo Distance) the performance curve for monomers is shown.

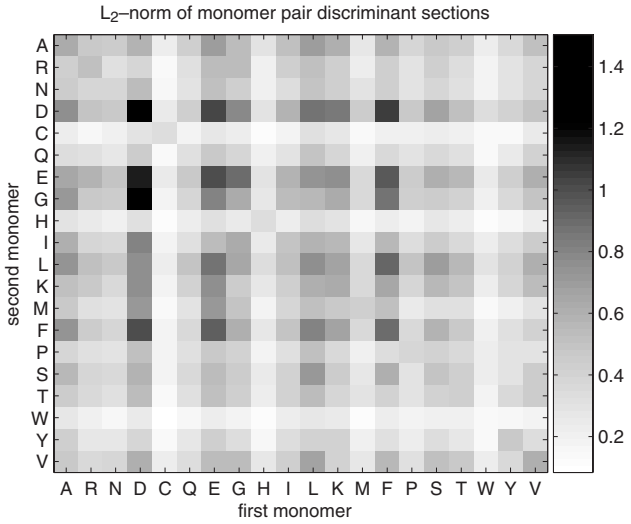

Fig. 2. Discriminative power (L_2-norm) of discriminant subvectors for all possible combinations of monomers in sequences from experiment 51; amino acid letters are used according to IUPAC one-letter code. The adjacent color bar shows the mapping of L_2-norm values.

representation based on monomers, which showed a slightly better ROC performance than the dimer-based representation. While the LA-eig kernel is slightly better for the higher ROC scores >0.85, our representation shows an improved performance for a decreasing score threshold with a higher number of included superfamilies. In particular for ROC scores between 0.7 and 0.85 the distance histograms outperform the compared methods.

During kernel-based training for monomer distance histograms on average 749 (26.3%) training examples turned out to be support vectors. In order to compare our results with the best alignment-based kernel, we also measured the support vector ratio of the local alignment kernel using the publicly available kernel matrices and the SVM parameters of (Saigo *et al*., 2004). The results revealed a significantly higher average number of support vectors ($\bar{N}_{SV} = 1330/47.1\%$). Note that for kernel-based classification all sequences which correspond to support vectors have to be evaluated in terms of kernel functions with regard to the new candidate sequence [see Equation (3)]. However, according to Section 2 this is not necessary for our approach since the discriminant can be calculated in feature space so that the calculation of the classification score reduces to a feature space transformation of the new sequence and the calculation of one sparse dot product with algorithmic complexity $O(L^2)$. Therefore the speed-up which can be achieved with our method in comparison with the local alignment kernel classifier ($O(\bar{N}_{SV} * L^2)$) is more than a factor 1000.

For kernel-based learning also the cost for computation of the kernel matrix has to be considered. For the worst case in terms of the most dense feature space, namely monomer distance histograms, this (largely sparse) procedure required 341 s (71 s for sequence transformation plus 270 s for the matrix product according to Section 2) on a standard PC. This is ∼20 times faster than the method presented in Saigo *et al*. (2004): running the author-provided program on the same machine we measured a CPU time of 6794 s (1 h 53 min) to calculate the pairwise similarity matrix which still requires some additional processing to obtain the final kernel matrix.

3.1 Discriminant visualization and interpretation

One of the main advantages of our representation is the possibility to compute (sparse) feature vectors of the sequences in order to visualize the resulting discriminant after kernel-based training.

According to the above results, already for monomers ($K = 1$) oligomer distance histograms yield a good performance and a rich representation with high discriminative power of the included features. The discriminative power of an oligomer pair (m_j, m_k) can be measured by the L_2-norm of the discriminant subvector associated with histogram vector $\mathbf{h}_{jk}$. As an example, for experiment 51 [corresponding to the superfamily of proteins containing an EF-hand motif (Yap *et al*., 1999)] of the above SCOP setting the L_2-norm of all 400 histogram vectors of monomer pairs is depicted in the 20×20 image in Figure 2. According to the darkest spots in the image, for experiment 51 the four most discriminative pairs are (D, D), (D, G), (D, E) and (F, D), indicating the importance of amino acid D (aspartic acid).

Figure 3 shows the discriminant weights of the four most discriminative monomer pairs for experiment 51 after kernel-based training as described above. As one might expect, long distances are less important for discrimination, indicated by the decay of the absolute value of the discriminant weights for increasing distances. Only the weights of the first 101 distances ($L_{\max} = 994$) are shown in Figure 3 in order to improve visibility of the more important weights.

Oligomer distances with large positive discriminant weights can be interpreted as characteristic features occurring in sequences from the corresponding family. The upper left picture shows the discriminant subvector of pair (D, D) where the peak at

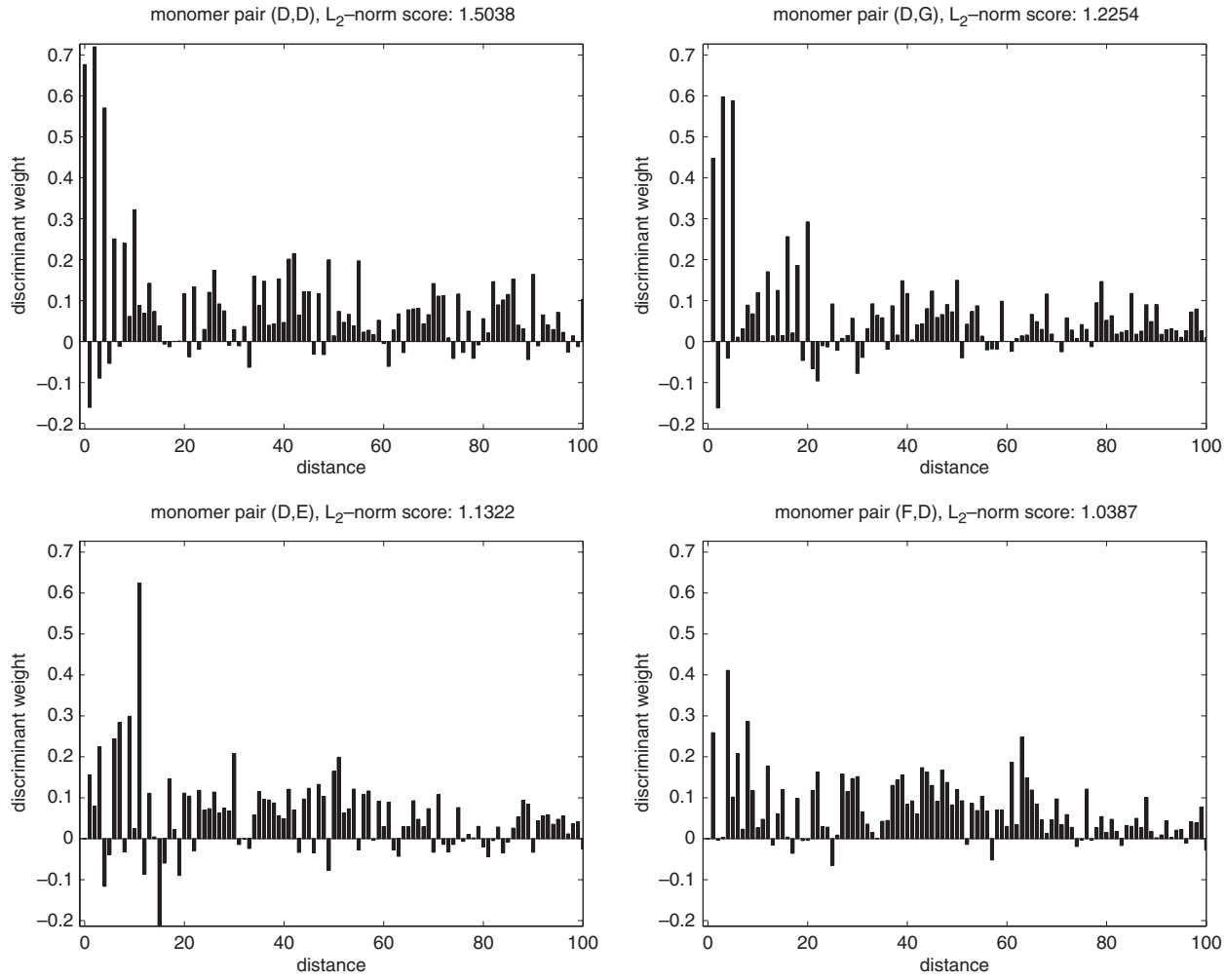

Fig. 3. Discriminant weights of the most discriminative monomer pairs for experiment 51; amino acid letters are used according to IUPAC one-letter code. Only the first 101 distances of each oligomer pair are shown (see text).

zero-distance shows the importance of aspartic acid frequency for discrimination. The picture also shows a comb-shaped structure of discriminant values for short distances. This structure indicates that even distances ($d = 2, 4, 6, \ldots$) at that range more frequently occur in positive training sequences than in counter-examples from the negative training set. On the other hand negative weights indicate that odd distances, e.g. for dimer *DD* frequencies, seem to occur more often in counter-examples. This characteristic distance distribution of aspartic acid can be clearly identified in the multiple alignment of sequences containing the above-mentioned EF-hand calcium-binding domain and the corresponding PROSITE pattern. The discriminant subvector of pair (D, G) shows a similar structure for small distances, but with even distances providing negative evidence. Note that discriminant values for pairs of differing monomers always have zero-weight at zero-distance because all histogram vectors contain zero counts at the associated positions. The other two bar plots in Figure 3 also show noticeable peaks for certain distances: e.g. with respect to pair (D, E), a high positive value for distance 11 and a high negative value for distance 15, or with respect to (F, D), high positive values for distances 1 and 4, respectively. In contrast, small values for pair (F, D) for distances 2 and 3 indicate that the corresponding occurrences are not discriminative. The increased density of high values at distances in the range 40–70 residues for pair (F, D) suggests relevance of longer distances for discrimination.

For an exemplary analysis of the discriminative features, Figure 4 shows the occurrences of selected features in sequences which correspond to the positive support vectors of the model. A sequence is symbolized by a rectangle whose width corresponds to the sequence length. Each feature occurrence is visualized by an arrow line whose horizontal position corresponds to the position of occurrence in the sequence, while the length of the line segment indicates the distance between the associated monomers. We selected two exemplary features suggested by analysis of the discriminant: in Figure 3 the discriminant subvector of pair (D, E) shows a large positive weight for distance 11. In Figure 4 the occurrence of the corresponding feature is depicted by the longer arrow lines between pair-specific residues. Another significant discriminant

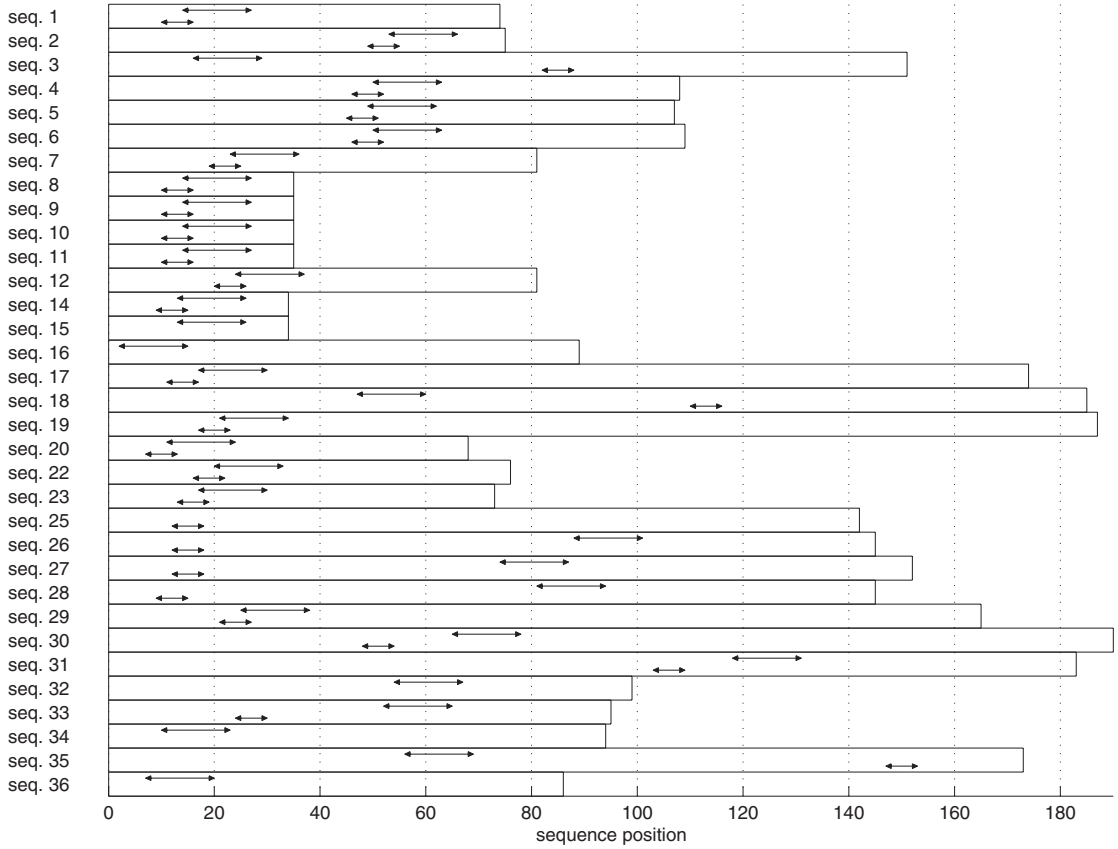

Fig. 4. Visualization of selected discriminant features for positive training sequences from experiment 51 corresponding to support vectors (see text). Long arrow lines represent the occurrence distribution of monomer pair (*D*, *E*) at distance 11, short arrow lines that of pair (*F*, *D*) at distance 4.

peak can be observed for pair (*F*, *D*) at distance 4, which corresponds to the shorter lines in Figure 4. These two features can be interpreted on the basis of biological knowledge: the EF-hand calcium-binding domain [PROSITE pattern PS00018 (Hulo *et al*., 2006)] shows a strong conservation of aspartic acid (*D*) and glutamic acid (*E*) at a distance of 11 residues where both amino acids are part of a loop between two alpha helices in the protein. In EF-hand-like proteins the leading alpha helix often contains a phenylalanin (*F*) at distance 4 ahead of the loop start which arises from the typical helical hydrogen bond structure. In Figure 4 this property can be matched with the feature occurrences. Many of the sequences—mostly from the family of Calmodulin-like proteins (ID 1.41.1.5, sequences 7–31)—show the above-mentioned characteristic amino acid distribution between sequence position 0 and 40. Others sequences show this feature combination at later sequence positions and often only the helical or the loop structure alone can be identified.

4 DISCUSSION AND CONCLUSION

We introduced a novel approach to remote homology detection based on oligomer distance histograms (ODH) for feature space representation of protein sequences. Although the ODH feature space provides a position independent representation of sequences, in comparison with other position independent approaches, like spectrum or mismatch kernels, additional information is extracted from the data by means of the distance histograms. The results show that this additional information is relevant for discrimination. Although the feature space of the ODH and other counting kernels like spectrum or mismatch kernels can formally be viewed as a special case of a general motif kernel, as for instance proposed in Ben-Hur and Brutlag (2003), it is obvious that restriction of the 'motif space' is necessary in order to make learning possible. Otherwise whole sequences could be used as motifs and the resulting representation would be too flexible to provide generalization. Therefore prior knowledge about relevant protein motifs in terms of conserved segments in multiple sequence alignments has been used in Ben-Hur and Brutlag (2003) to restrict the set of possible motifs. In contrast our approach as well as the spectrum or mismatch kernel do not require any domain knowledge in order to realize learnability. In Dong *et al*. (2006) the authors showed that on the above benchmark dataset the knowledge-based motif kernel of Ben-Hur and Brutlag (2003) is clearly outperformed by the local alignment kernel with a detection performance similar to the SVM pairwise method which is included in our performance comparison in Section 3.

Because the distance-specific representation of all pairwise *K*-mer occurrences gives rise to rather high-dimensional feature

vectors, the sparseness of these vectors has to be utilized in order to keep the approach feasible. Then sparse matrix algebra can be used for efficient computation of the kernel matrix which in turn can be used for kernel-based training of classifiers. Although the theoretical algorithmic worst-case complexity of our approach for computation of the kernel value for two sequences S_1 and S_2 equals that of the local alignment kernel ($O(L^2)$ for $L_1 \approx L_2$), we showed that our method is significantly faster.

Using standard SVMs, we showed that the prediction performance of our distance-based approach is highly competitive with state-of-the-art methods within the field of remote homology detection. Although the local alignment kernel of Saigo *et al*. (2004) yields slightly better results, it should be noted that its performance depends on a continuous kernel parameter (β). Because the performance can significantly decrease for non-optimal values of that hyperparameter (Saigo *et al*., 2004), in practice a time-consuming model selection process would be necessary with that method to achieve optimal results. Furthermore the local alignment kernel involves two additional parameters which, however, have not been evaluated for their influence on the performance (Saigo *et al*., 2004). In contrast, the homogeneity of ROC values for monomer and dimer distances underlines the good generalization performance of our representation which obviates the tuning of any hyperparameters.

Another advantage of our approach arises from the explicit feature space representation: the possibility to calculate the discriminant weight vector in feature space allows for fast classification of new data. In contrast kernel-based methods without an explicit feature space need to evaluate kernel functions of all relevant training sequences with regard to the new candidate sequence. This is in general time-consuming for problems with a large number of support vectors. We showed that in the remote homology detection setup an explicit discriminant weight vector can result in a speed-up of more than factor 1000. The explicit representation also automatically implies positive semidefinite kernel matrices which are required for kernel-based training. In contrast, the local alignment kernel arises from a similarity matrix which has to be transformed in order to be positive semidefinite. In Saigo *et al*. (2004) two transformation methods have been proposed which were evaluated in terms of the resulting test set performance. However, it remains unclear how these methods apply to classification of new sequences in practice.

With respect to other position independent approaches, like spectrum or mismatch kernels, ODHs considerably improve the detection performance while preserving the favorable interpretability of the former approaches in terms of an explicit feature space representation. The advantage of interpretable features has also been realized by other researchers: in Kuang *et al*. (2005) profile-based string kernels were used to extract 'discriminative sequence motifs' which can be interpreted as structural features of protein sequences. On a similar dataset the method also provides state-of-the-art performance. However, the performance of the approach depends on two kernel parameters, an additional smoothing parameter and the number of PSI-BLAST iterations for profile extraction.

As we showed, also ODHs allow the user to analyze the learnt model for identification of the most discriminative features. These features, which correspond to pairs of oligomers occurring at characteristic distances, may in turn reveal biologically relevant properties of the underlying protein families. In contrast, the best position-dependent approaches, like local alignment kernels, do not provide an intuitive insight into the learnt model. Without an explicit transformation into some meaningful feature space these approaches lack an interpretability of the discriminant in terms of discriminative sequence features. Furthermore, local alignment kernels involve several hyperparameters which complicate the evaluation and application of the proposed method. Besides the oligomer length K, ODHs do not require the specification of any kernel parameters and therefore our approach obviates a time-consuming optimization which moreover could increase the risk of fitting the data to the test set. In our experimental evaluation ODHs based on monomers and dimers both showed a good generalization behavior. We found the trimer-based representation to break down, because obviously the corresponding feature vectors become too sparse. A similar behavior can be observed for the K-mer counting spectrum kernel if K becomes too large. On the widely used SCOP dataset considered here, the spectrum kernel breaks down for $K = 4$ (Leslie *et al*., 2004). The authors in Leslie *et al*. (2004) therefore proposed to allow mismatches in order to increase the number of non-zero counts. The best resulting mismatch-kernel ($K = 5$, one mismatch) significantly improves the performance of the spectrum kernel. Therefore also the ODH performance may be increased by the incorporation of mismatches. Many other strategies for further improvement of the performance are conceivable: e.g. the set of oligomers may be restricted in a suitable way, as well as the range of possible distances. In Meinicke *et al*. (2004) position-dependent oligo kernels for sequence analysis were introduced where a smoothing parameter is used to represent positional variability. In a similar way, distance variability could be realized with oligomer distance histograms by means of histogram smoothing techniques. Although these extensions may considerably improve the detection performance, we are aware of several hyperparameters which would have to be included into the representation. We think it is an important advantage of our method that it does not require any parameter tuning in order to achieve state-of-the-art performance.

ACKNOWLEDGEMENTS

The work was partially supported by BMBF project MediGrid (01AK803G).

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al*. (1990) Basic local alignment search tool. *J. Mol. Biol*., **215**, 403–410.

Ben-Hur,A. and Brutlag,D. (2003) Remote homology detection: a motif based approach. *Bioinformatics*, **19** (Suppl. 1), i26–i33.

Dong,Q. *et al*. (2006) Application of latent semantic analysis to protein remote homology detection. *Bioinformatics*, **22**, 285–290.

Hulo,N. *et al*. (2006) The PROSITE database. *Nucleic Acids Res*., **34**, D227–D230.

Jaakkola,T. *et al*. (2000) A discriminative framework for detecting remote protein homologies. *J. Comput. Biol*., **7**, 95–114.

Krogh,A. *et al*. (1994) Hidden Markov models in computational biology. Applications to protein modeling. *J. Mol. Biol*., **235**, 1501–1531.

Kuang,R. *et al*. (2005) Profile-based string kernels for remote homology detection and motif extraction. *J. Bioinform. Comput. Biol*., **3**, 527–550.

Liao,L. and Noble,W.S. (2002) Combining pairwise sequence similarity and support vector machines for remote protein homology detection. In *Proceedings of the*

Sixth Annual International Conference on Research in Computational Molecular Biology, pp. 225–232.

Leslie,C. *et al*. (2002) The spectrum kernel: A string kernel for SVM protein classification. *Pac. Symp. Biocomput*., 566–575.

Leslie,C. *et al*. (2004) Mismatch string kernels for discriminative protein classification. *Bioinformatics*, **20**, 467–476.

Ma,X. *et al*. (2004) Predicting polymerase II core promoters by cooperating transcription factor binding sites in eukaryotic genes. *Acta Biochim. Biophys. Sin*., **36**, 250–258.

Meinicke,P. *et al*. (2004) Oligo kernels for datamining on biological sequences: a case study on prokaryotic translation initiation sites. *BMC Bioinformatics*, **5**, 169.

Murzin,A.G. *et al*. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol*., **24**, 536–540.

Park,J. *et al*. (1998) Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J. Mol. Biol*., **284**, 1201–1210.

Rangwala,H. and Karypis,G. (2005) Profile-based direct kernels for remote homology detection and fold recognition. *Bioinformatics*, **21**, 4329–4247.

Saigo,H. *et al*. (2004) Protein homology detection using string alignment kernels. *Bioinformatics*, **20**, 1682–1689.

Schölkopf,B. and Smola,A.J. (2002) *Learning with Kernels*. MIT Press, Cambridge, MA.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol*., **147**, 195–197.

Weston,J. *et al*. (2005) Semi-supervised protein classification using cluster kernels. *Bioinformatics*, **21**, 3241–3247.

Yap,K.L. *et al*. (1999) Diversity of conformational states and changes within the EF-hand protein superfamily. *Proteins*, **37**, 499–507.

Anhang B

Artikel 2

BMC Bioinformatics

Research article

Word correlation matrices for protein sequence analysis and remote homology detection

Thomas Lingner* and Peter Meinicke

Address: Department of Bioinformatics, Institute of Microbiology and Genetics, Georg-August-University Göttingen, Göttingen, Germany

Email: Thomas Lingner* - thomas@gobics.de; Peter Meinicke - pmeinic@gwdg.de

* Corresponding author

Published: 3 June 2008

BMC Bioinformatics 2008, **9**:259 doi:10.1186/1471-2105-9-259

Received: 20 February 2008

Accepted: 3 June 2008

This article is available from: http://www.biomedcentral.com/1471-2105/9/259

Abstract

Background: Classification of protein sequences is a central problem in computational biology. Currently, among computational methods discriminative kernel-based approaches provide the most accurate results. However, kernel-based methods often lack an interpretable model for analysis of discriminative sequence features, and predictions on new sequences usually are computationally expensive.

Results: In this work we present a novel kernel for protein sequences based on average word similarity between two sequences. We show that this kernel gives rise to a feature space that allows analysis of discriminative features and fast classification of new sequences. We demonstrate the performance of our approach on a widely-used benchmark setup for protein remote homology detection.

Conclusion: Our word correlation approach provides highly competitive performance as compared with state-of-the-art methods for protein remote homology detection. The learned model is interpretable in terms of biologically meaningful features. In particular, analysis of discriminative words allows the identification of characteristic regions in biological sequences. Because of its high computational efficiency, our method can be applied to ranking of potential homologs in large databases.

Background

Advances in large-scale sequencing have led to a vast amount of protein sequences that have to be classified into structural and functional classes. Because experimental determination is time consuming and expensive, several computational methods based on sequence similarity were introduced to automatically annotate sequences by homology transfer. For close homologs, i.e. sequences with a similarity of more than 80% at the amino acid level, this can be done by pairwise comparison methods like the Smith-Waterman local alignment algorithm [1] or BLAST [2]. However, these methods often fail in cases where sequence similarity is low. In the so-called "twilight-zone", the detection of remote homologies still remains a challenging task in computational biology.

Remote homology detection methods are often based on a statistical representation of protein families and can be divided into two major categories: first, profile-based methods provide a non-discriminative approach to family-specific representation of sequence properties. The corresponding generative models are usually trained using only known example sequences of the particular family [3,4]. Second, discriminative methods provide a super-

vised approach [5-8] to representing sequence properties that explicitly model the differences between protein families. In this case, training requires example sequences from the particular protein family and counterexamples from the other protein families.

Discriminative methods often measure the similarity of two sequences by means of a kernel function. A sequence kernel computes the inner product of sequence representatives in some abstract feature space, often without explicit transformation of the sequences into that space. Using learning algorithms that only need to evaluate inner products between feature space elements, the "kernel trick" makes learning in complex and high dimensional feature spaces possible. Recent studies [7-14] have shown that discriminative kernel methods can significantly increase the detection performance as compared with profile-based methods.

Kernel methods in general require the evaluation of N^2 kernel functions for training the discriminant function on a set of N sequences. Since this requirement is computationally demanding even for a few thousand sequences, the use of kernel-based approaches for large-scale discriminative learning is problematic. Testing the trained model is also expensive since it involves kernel computations between test examples and N training examples.

However, in some cases evaluation of the discriminant can be computed rather efficiently if an explicit representation of the discriminant in feature space is possible. For example, the Spectrum kernel [9] measures the similarity between two sequences by counting the occurrences of all K-length subsequences ("K-mers") in these sequences. The method has been shown to provide considerable speed-up of the evaluation using the discriminant in the K-mer feature space. However, the use of the Spectrum kernel for longer K-mers is problematic, because of the decreasing number of perfect matches. Several methods based on inexact matches have been introduced to tackle this problem [15]. These methods count the occurrences of nearly matching K-mers by means of a binary match function that is invariant with respect to changes within a specified "mutation neighborhood". For example, the Mismatch kernel [8] defines a mapping to the K-mer feature space via a (K, m)-"mismatch neighborhood", i.e. the occurrence of a particular K-mer i contributes to all feature space dimensions associated with K-mers that differ from i by at most m mismatches. Recently, Oligomer Distance Histograms [14] have been introduced for protein sequence representation and remote homology detection. Here, the similarity between two sequences is measured by counting the occurrences of all K-mer pairs for all distances. Oligomer Distance Histograms are highly competitive with state-of-the-art methods for remote homology detection and provide an explicit feature space. All these feature-based methods allow for fast classification of new sequences. Furthermore, they do not require prior knowledge about sequence properties in terms of relevant motifs or structural information. By analysis of the discriminative features, these methods can even help to find new motifs or other interesting sequence properties.

In contrast, motif kernels [7] evaluate the occurrences of known motifs from an existing motif database, i.e. the number of matching motifs in a pair of sequences is used to define a kernel. As another example, profile kernels [11] use probabilistic profiles as produced by PSI-BLAST to define "positional mutation neighborhoods", i.e. profile-defined mappings to the K-mer feature space. Here, the profiles originate from an initial homology search of training examples, therefore this method can also be viewed as a homology-based kernel. Based on prior knowledge, motif kernels and profile kernels also provide an explicit representation of the discriminant, and thus allow for interpretation in the associated feature space and fast classification of new sequences.

Currently, alignment-based kernels show the best detection performance on widely-used homology detection setups [10,12]. For example, in [10] the authors derive the similarity measure between two sequences from the sum of their local alignment scores. This similarity measure requires additional transformation in order to provide a valid kernel. However, these methods show a significant disadvantage concerning the *interpretability* of the resulting discriminant model. In contrast to methods that are based on a meaningful vector space representation of the sequences, alignment-based kernels do not provide direct inspection of the associated feature space. With this limitation it is difficult to identify the relevant sequence properties that have been learned from the data. Therefore, these kernels do not offer additional utility for researchers interested in finding the characteristic features of protein families. In principle, the same holds for kernel methods that involve certain kinds of nonlinear transformations, like Gaussian (RBF) kernels do, because the learned discriminant parameters, i.e. the sequence-specific weights after kernel-based training, cannot be associated with particular sequence properties. This considerably complicates the interpretation of these "black box" classification models.

As an additional drawback, several kernel methods incorporate *hyperparameters* that have to be carefully adjusted before training. For example, the authors of [10] used a total number of 3 kernel parameters, two of which were fixed in an ad-hoc manner. The dependence of the performance on the third parameter was evaluated on the test data in this particular setup. Other approaches, e.g. [12] and [13] also comprise several hyperparameters that were

chosen to provide maximum performance on the test data. The extensive use of hyperparameters increases the risk of overfitting when no dedicated validation data set is used. In this case, the application of the method to different data is difficult because new data are likely to require the readjustment of these parameters.

In this work, we present an alignment-free feature space representation for protein sequences, which is based on the average pairwise similarity of short subsequences ("words"). First, we show that this similarity measure defines a valid kernel function between two sequences. We then provide some further analysis of the associated sequence representation, which gives rise to a well interpretable feature space in terms of "word correlation matrices" (WCMs). Finally, we demonstrate the performance of this representation on a widely-used benchmark setup for protein remote homology detection. In addition, we show how the resulting discriminants can be analyzed to gain insight into particular sequence properties.

Methods

From Average Word Similarity to Word Correlation Matrices

We first define a sequence similarity measure based on average word similarity. Consider two sequences S, $\tilde{S}$, represented by two lists of words W, $\tilde{W}$ containing all consecutive overlapping K-length words w_i, $\tilde{w}_j$ occurring in the respective sequence(s). With some word similarity function $s(w, \tilde{w})$ measuring the similarity between words w and $\tilde{w}$ we compute the *average word similarity* between sequences S, $\tilde{S}$ by

$$k(S,\tilde{S}) = \frac{1}{n\tilde{n}} \sum_{i=1}^{n} \sum_{j=1}^{\tilde{n}} s(w_i, \tilde{w}_j) \tag{1}$$

where n and $\tilde{n}$ denote the number of K-length words in the sequences. In particular we are interested in word similarity functions that provide a positive semidefinite sequence similarity measure, i.e. that provide valid sequence *kernels*. We here propose a simple realization of the word similarity function that not only results in a valid sequence kernel but also implies a feature space of moderate dimensionality. Consider an alphabet $\mathcal{A}$ and a binary vector encoding of K-length words $\mathbf{x} \in \{0,1\}^{K|\mathcal{A}|}$. The *i - th* letter of a word only yields a non-zero entry in vector dimension $K \times (i - 1) + j$ if that letter matches the j-th element of the alphabet. Let $\mathbf{z} \in \{0, 1\}^{20}$ be an amino acid indicator vector, i.e. a 20-dimensional vector that contains only one non-zero entry for the vector dimension associated with a particular amino acid. With this definition and ^{T} indicating vector (matrix) transposition, a word vector for protein sequences corresponds to a stacking of particular amino acid indicator vectors $\mathbf{x} = [\mathbf{z}_1^T, ..., \mathbf{z}_K^T]^T$ for K different word positions. With the two word vectors $\mathbf{x}$, $\tilde{\mathbf{x}}$ of the words w, $\tilde{w}$ our word similarity is computed by the squared dot product

$$s(w,\tilde{w}) = (\mathbf{x}^T\tilde{\mathbf{x}})^2. \tag{2}$$

Note that this measure corresponds to the squared number of matching letters occurring at the same position in both words. In terms of the Hamming distance $h(w, \tilde{w})$ between words, it is equal to $(K - h(w, \tilde{w}))^2$. We shall now show that this formulation gives rise to a valid sequence kernel $k(S, \tilde{S})$ if used in Equation (1). Further we will consider the dimensionality of the associated feature space, which will be shown to grow quadratically with the word length K. We now write the above sequence similarity in terms of the word vectors $\mathbf{x}_i$ and $\tilde{\mathbf{x}}_j$ of S and $\tilde{S}$, respectively:

$$k(S,\tilde{S}) = \frac{1}{n\tilde{n}} \sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{x}_i^T\tilde{\mathbf{x}}_j)^2 \tag{3}$$

$$= \frac{1}{n\tilde{n}} \sum_{i=1}^{n} \sum_{j=1}^{\tilde{n}} (\mathbf{x}_i^T\tilde{\mathbf{x}}_j)(\tilde{\mathbf{x}}_j^T\mathbf{x}_i) \tag{4}$$

$$= \frac{1}{n\tilde{n}} \sum_{i=1}^{n} \sum_{j=1}^{\tilde{n}} \mathrm{tr}(\mathbf{x}_i\mathbf{x}_i^T\tilde{\mathbf{x}}_j\tilde{\mathbf{x}}_j^T) \tag{5}$$

$$= \mathrm{tr}\left(\underbrace{\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^T}_{S-specific} \underbrace{\frac{1}{\tilde{n}}\sum_{j=1}^{\tilde{n}} \tilde{\mathbf{x}}_j\tilde{\mathbf{x}}_j^T}_{\tilde{S}-specific} \right) \tag{6}$$

where *tr* denotes the trace function, i.e. the sum of diagonal elements. With matrix $\mathbf{X}_S$ containing all word vectors $\mathbf{x}_i$ of sequence S as columns, we define the sequence-specific *word correlation matrix* (WCM) as

$$\mathbf{C}(\mathbf{X}_S) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^T = \frac{1}{n}\mathbf{X}_S\mathbf{X}_S^T \tag{7}$$

With the abbreviations $\mathbf{C} \equiv \mathbf{C}(\mathbf{X}_S)$ and $\tilde{\mathbf{C}} \equiv \mathbf{C}(\mathbf{X}_{\tilde{S}})$ we can finally write the kernel as

$$k(S, \tilde{S}) = \mathrm{tr}(\mathbf{C}\tilde{\mathbf{C}}) = \mathrm{vec}(\mathbf{C})^T \mathrm{vec}(\tilde{\mathbf{C}}). \quad (8)$$

The *vec* function converts a matrix to a vector by stacking the matrix columns successively, i.e. the upper right element in a 2 × 2 matrix contributes to the third vector dimension. From this we see that the sequence kernel corresponds to a dot product in a particular feature space which arises from vectorized WCMs. In the following, we use

$$\Phi = \mathrm{vec}(\mathbf{C}) \quad (9)$$

to denote the feature space representative of a sequence.

WCM feature space

The particular primary structure of a protein is commonly characterized by a sequence of amino acids. The IUPAC one-letter abbreviation code for 20 naturally occurring amino acids gives rise to an alphabet $\mathcal{A} = \{A, R, N, ..., V\}$ with $|\mathcal{A}| = 20$. For a protein sequence S and a given word length K, every dimension in the WCM feature vector Φ corresponds to the number of occurrences of two particular amino acids at specific positions within all words of length K in S. For example, the first feature space dimension counts the occurrences of Alanine (A) at the first position of all words. The second dimension corresponds to the number of occurrences of Alanine *and* Arginine at the first position. If the binary **z**-vector encoding as defined in the previous section is used, this dimension always contains a zero value, because different amino acids cannot occur at the same word position by definition. However, this dimension can be useful in combination with word encoding schemes that take into account amino acid substitutions. As a last example, the 21st dimension in our WCM feature space corresponds to the number of occurrences of Alanine at the first and second position of all words, i.e. the frequency of the dimer *AA*.

Interestingly, the features of the WCM representation correspond to features of special realizations of Oligomer Distance Histograms [14]: for a particular word length K the WCM features correspond to features of Monomer Distance Histograms when only distances up to K - 1 are taken into account. For a particular distance D, Monomer Distance Histograms contain the number of occurrences of all amino acid pairs whose sequence positions differ by D. A feature in the WCM feature space contains the number of occurrences of two amino acids at distance D at particular positions within the same word. Because of overlapping words in a sequence, a particular feature associated with a dimension in the Monomer Distance Histogram feature space is counted at most K times and added to different WCM feature space dimensions according to specific word positions. On the other hand, the first and last K - 1 words in a sequence have less overlap with other words than words inside the sequence, such that features of words at the beginning and at the end of a sequence are counted less than K times. Therefore, long words and short sequences would result in more different features as compared with the Monomer Distance Histogram feature space. In total, the WCM feature space comprises $(K|\mathcal{A}|)^2$ dimensions, and thus grows quadratically with the word length. Because of the symmetry of the WCM, it is sufficient to consider the upper (or lower) triangular matrix, which can be used to reduce the dimensionality of the feature space to $\frac{K|\mathcal{A}|(K|\mathcal{A}|+1)}{2}$. Furthermore, off-diagonal elements of entries belonging to the same word position can be disregarded if amino acid indicator vectors are used. In this case, the feature space reduces to $K|\mathcal{A}| + \frac{K|\mathcal{A}|(K|\mathcal{A}|-1)}{2}$ dimensions.

Kernel matrix computation

For kernel-based training with a set of N sequences, the $N \times N$ matrix of pairwise kernel functions between all sequences has to be computed. Doing this directly according to Equation (3) requires $\frac{N(N-1)}{2}$ evaluations of all $L\tilde{L}$ word similarity values between two sequences of length L and $\tilde{L}$, respectively. Therefore, the overall algorithmic time complexity of this method is $O(N(N - 1)L\tilde{L}K|\mathcal{A}|)$. With $L \approx \tilde{L}$ and $|\mathcal{A}| = const.$ this simplifies to $O(N^2L^2K)$. In particular, for long sequences this can be computationally demanding.

However, in most cases the kernel matrix can be efficiently calculated using the feature space representatives Φ of the sequences as defined in Equation (9). After transformation of all sequences into the WCM feature space, their representatives can be stored in a matrix $\mathbf{M} = [\Phi_1, ..., \Phi_N]$. Then, the kernel matrix $\mathbf{K}$ can be computed by the matrix product

$$\mathbf{K} = \mathbf{M}^T\mathbf{M}. \quad (10)$$

Using the same simplifications as above, the feature-based computation of the kernel matrix involves N sequence

transformations of complexity $O(LK^2)$ and the evaluation of the matrix product involving the $LK^2 \times N$ matrix **M**, which is of theoretical complexity $O(N^2LK^2)$. Therefore, the overall time complexity of this method is $O(N^2LK^2)$. In contrast to the direct kernel computation, the computational complexity only grows linearly with the length of the sequences but quadratically with the word length.

The theoretical overall time complexity formulas indicate that for $L > K$ the feature-based method is preferable for calculation of the kernel matrix. In general, K has to be chosen to be significantly smaller than L in order to obtain reasonable sequence similarity values. Feature-based calculation is much more efficient than the direct computation for moderate word length K. However, the memory requirements to store all feature vectors grows quadratically with the word length K.

We compared the required time for computation of the kernel matrix using 1000 protein sequences with an average length of 118.6 amino acids. The feature-based calculation using a word length of $K = 5$ ($K = 10$) took 3.09 (7.51) seconds on an AMD Opteron 870 processor with 2GB RAM. Thereby 1.83 (3.62) seconds were used for the transformation of the sequences into the 5050 (20100) dimensional feature space and 1.26 (3.89) seconds were used for the computation of the matrix product. In contrast, the direct calculation of the kernel matrices took 583 and 927 seconds, respectively.

Discriminant function in feature space

After kernel-based training, the learned sequence-specific weights can be used to calculate the discriminant weight vector in WCM feature space for better interpretation and fast computation of the discriminant.

Let $\alpha = [\alpha_1, ..., \alpha_N]^T$ be the weight vector of a set of N sequences after kernel-based training and **M** be the matrix of sequence representatives. Then, the discriminant weight vector **w** in feature space can be computed according to

$$\mathbf{w} = \mathbf{M}\alpha. \tag{11}$$

The magnitude of an entry in **w** reflects the discriminative power of the corresponding feature. This can be used to identify relevant features or feature combinations for a given set of sequences. For better interpretability, the discriminant weight vector can be remapped to the WCM space, which provides a convenient visualization of the discriminant.

The discriminant weight vector in feature space can also be used to identify discriminative words in a set of sequences. The discriminative power of a particular word in terms of a word score $score(\mathbf{x})$ can be computed with the discriminant weight vector **w** and the word vector **x** according to

$$score(\mathbf{x}) = \mathbf{x}^T\mathbf{W}\mathbf{x} \tag{12}$$

where **W** is the WCM space representation associated with **w**, i.e. $vec(\mathbf{W}) = \mathbf{w}$. High absolute word score values indicate importance of w for discrimination between positive and negative example sequences. These discriminative words can be interpreted biologically in terms of short "motifs", i.e. conserved sequence regions within a set of related sequences. Scores with a low magnitude usually correspond to words that do not contribute significantly to the discrimination, e.g. words that occur in positive and in negative example sequences. Discriminative word scores can also be used to detect discriminative regions within sequences by means of score profiles. A score profile of a sequence S is the sequence of word scores for all overlapping words of S. Discriminative regions of S correspond to global or local maxima (minima) of the score profile of S. In Figure 1, five exemplary word score profiles are shown.

For fast classification of a new sequence S with the discriminant weight vector in WCM feature space, the classification score can be efficiently computed according to

$$Score(S) = \mathbf{w}^T\Phi\ . \tag{13}$$

The score computation involves transformation of the sequence to the feature space with complexity $O(LK^2)$ and the calculation of the dot product for at most $(K|\mathcal{A}|)^2$ vector elements. Using the same simplification as in the previous section, the overall computational complexity of classification with the feature space discriminant is $O(LK^2)$. In contrast, for kernel-based classification of S the evaluation of N kernel functions

$$Score(S) = \sum_{i=1}^{N} \alpha_i k(S_i, S) \tag{14}$$

according to N training sequences is necessary. Note that only kernels with a non-zero α_i (support vectors) need to be considered. With L^2K computations for evaluation of a single kernel function the overall complexity for kernel-based classification is $O(NL^2K)$. This indicates that for large N the feature-based computation of the classification score can be faster by orders of magnitude.

Results

In order to evaluate our approach, we considered a widely-used benchmark data set for remote homology detection [6] based on the SCOP database [16]. In the cor-

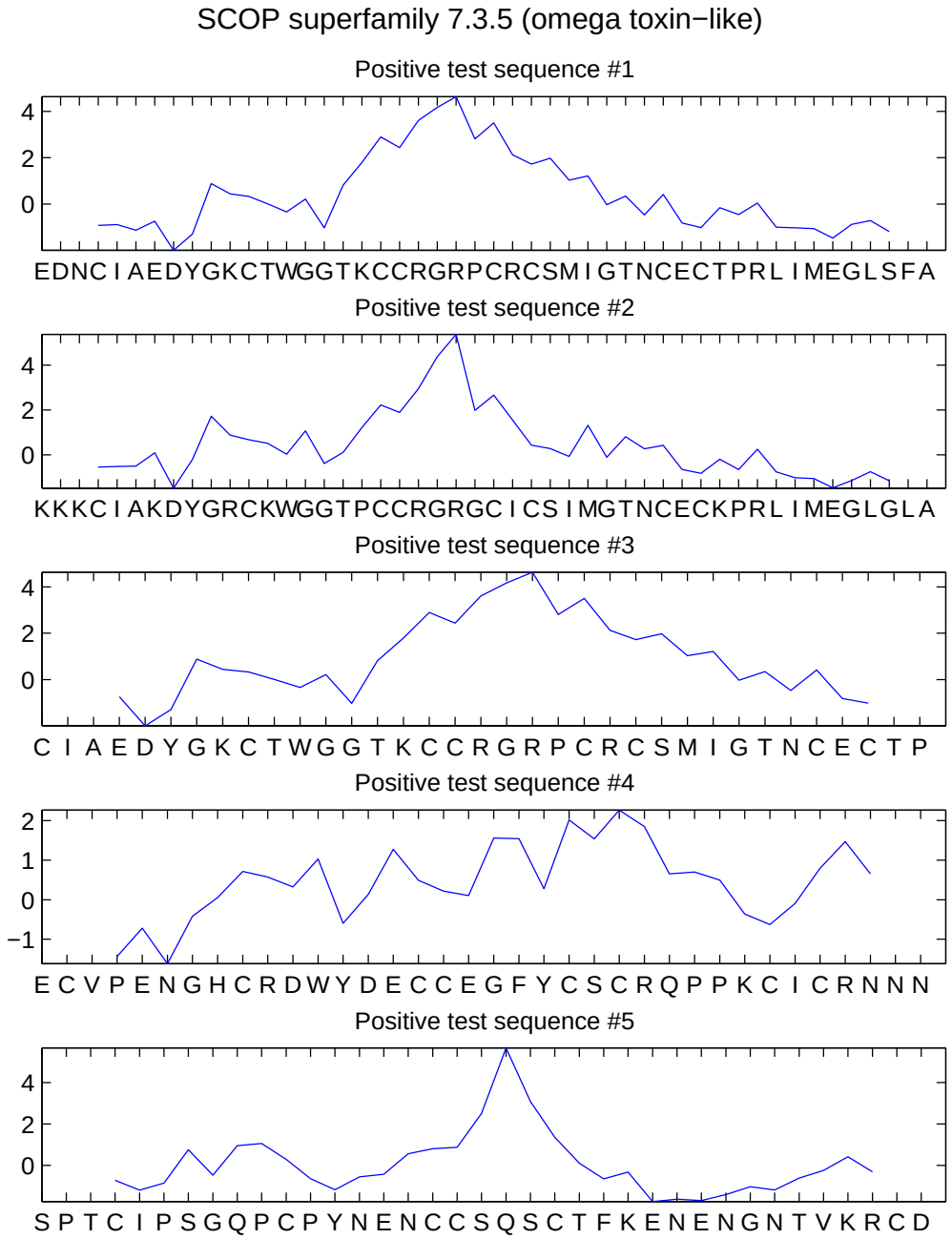

Figure 1
Word score profiles for positive test sequences of SCOP superfamily 7.3.5. Word score profiles of the first 5 positive test sequences associated with experiment 1 (SCOP superfamily 7.3.5: omega toxin-like) using word length $K = 6$. Amino acid sequences are mapped to the x-axis while the y-axis corresponds to discriminative word scores. Word score values are centered at position 4 of the overlapping words. See Equation (12) in section "Discriminant function in feature space" for details about calculation of word scores.

responding setup, remote homology detection is simulated by holding out all sequences of a particular SCOP family from a given superfamily in order to use these members as positive test examples. Positive training examples were selected from the remaining families in the same SCOP superfamily. Negative training and test examples have been drawn from disjoint sets of folds outside the fold of the target (test) family. In that way, every detection experiment involves a specific set of negative examples. According to the considered subset of SCOP families there are 54 binary classification problems at the superfamily level of the SCOP hierarchy. In this setup, the number of negative examples for each experiment is much larger than that of the positive ones. In particular, this situation gives rise to highly "unbalanced" training sets. In total, the setup consists of 4352 sequences from the SCOP 1.53 database.

To test the quality of our representation based on average word similarity, we utilize kernel-based support vector machines (SVM). Kernel methods in general require the evaluation of a kernel matrix including all inner products between training examples. To speed up computation, we pre-calculated the kernel matrices based on all 4352 sequences for different K and extracted the experiment-specific entries according to the setup of [6]. In the evaluation we tested our method for words of length K = 1, .., 10, whereby the entries of $\mathbf{K} = [k_{ij}]$ were normalized according to

$$k'_{ij} = \frac{k_{ij}}{\sqrt{k_{ii} \cdot k_{jj}}}. \qquad (15)$$

All kernel matrices used for the evaluation can be downloaded in compressed text format from [17]. Instead of the GIST support vector machine that was used in the original setup, we apply a MATLAB® implementation of the soft margin SVM with quadratic loss function as described in [18] for kernel-based training. The first reason is that we observed convergence problems of the GIST SVM in some cases. The second reason is that the direct implementation is considerably faster since the GIST package requires to create large experiment-specific data files containing the training and test kernel matrices. For reasons of comparability to the setup in [6], we used the same constant offset parameter (o = 10) for the kernel matrix and fixed the scaling parameter of the diagonal factor to a constant value (q = 1). While the offset parameter is added to all entries of the kernel matrix, the diagonal factor only affects the diagonal elements in order to cope with the unbalanced data sets [19]. With the diagonal factor q and the median of the diagonal elements m, $\frac{N^+}{N} qm$ and $\frac{N^-}{N} qm$ are added to diagonal elements for positive and negative examples, respectively. For training of the SVM we use the normalized kernel as defined in Equation (15) without any further transformations.

Besides from the unbalanced training sets, the setup in [6] also provides unbalanced test sets. In this case, widely-used performance metrics like predictive accuracy are not applicable [19]. Furthermore, homology search usually requires the analysis of an ordered list of potential homologs rather than hard classification. To measure the detection performance of our method on the test data, we calculated the area under curve with respect to the receiver operating characteristics (ROC) and the ROC50 score, which is the area under curve up to 50 false positives. Besides this, we also computed the median rate of false positives (mRFP). The mRFP is the ratio of false positive examples, which score equal or higher than the median score of true positives.

The results of our performance evaluation are summarized in Table 1 in comparison with other approaches. In order to exclude differences due to different implementation of the L_2-SVM, we recalculated the detection performance for all approaches. For the Spectrum method, we also performed experiments with combined kernel matrices using word length sets $\hat{K} = \{1, 2\}$, $\hat{K} = \{1, 2, 3\}$ and $\hat{K} = \{1, 2, 3, 4\}$. For this purpose, we calculated the average kernel matrix element over different word lengths. The performance indices in the table correspond to average ROC/ROC50 and mRFP values over all 54 experiments. Furthermore, the average number of support vectors is given in the fifth column of the table. Support vectors are data examples with a non-zero weight after kernel-based training and have to be considered for kernel-based classification of new sequences. Therefore, a lower number of support vectors in general decreases the storage requirements and the computational demands for kernel-based evaluation of the discriminant. In addition, most SVM training schemes benefit from a smaller number of support vectors in terms of decreasing computation time.

The performance values indicate that the WCM approach is well-comparable with other state-of-the-art methods. While the local alignment kernel and monomer distance histograms show better ROC and ROC50 performance, our new approach outperforms other feature-space based methods as well as the SVM pairwise kernel.

As described in the previous section, an explicit discriminant weight vector can be calculated in WCM feature space (see Equation (11)). Therefore, the weight vector

Table 1: Overview of detection performance for several methods.

Method	avg. ROC	avg. ROC50	avg. mRFP	avg. # SV
WCM_1	0.8705	0.3153	0.1065	1798
WCM_2	0.8926	0.3814	0.0833	1673
WCM_3	0.8964	0.4040	0.0813	1628
WCM_4	0.9013	0.4257	0.0801	1604
WCM_5	0.9032	0.4413	0.0795	1591
WCM_6	0.9044	0.4473	0.0778	1591
WCM_7	0.9036	0.4454	0.0785	1600
WCM_8	0.9024	0.4470	0.0801	1607
WCM_9	0.9018	0.4516	0.0815	1614
WCM_{10}	0.9012	0.4528	0.0830	1620
LA-eig	0.9348	0.6614	0.0489	2640
ODH Monomer	0.9135	0.4554	0.0729	1601
SVM pairwise	0.9008	0.3986	0.0810	2355
Mismatch (5,1)	0.8852	0.3815	0.0949	2943
Spectrum (3)	0.8239	0.2939	0.1535	2350
Spectrum {1,2}	0.8919	0.3913	0.0798	1560
Spectrum {1,2,3}	0.8957	0.4094	0.0766	1711
Spectrum {1,2,3,4}	0.8981	0.4180	0.0769	1882

Performance evaluation results of the word correlation approach (WCM_K) using several word lengths K = 1, ..10 in comparison to local alignment kernel (LA-eig) [10], Monomer Distance Histograms (ODH Monomer) [14], SVM pairwise [6], Mismatch string kernel [8], Spectrum kernel [9] and the combination of Spectrum kernels for different word lengths (see section "Results").

can be visualized in WCM space for identification of discriminative features. Figure 2 shows the WCM discriminant of superfamily 7.3.5 (omega toxin-like) according to experiment 1 after kernel-based training using word length K = 6. Rows and columns correspond to particular amino acids at particular word positions for the first and second word occurrence, respectively. Elements with values in the range between 10% of the largest negative and 10% of the largest positive discriminant value were set to zero to reduce the noise in the visualization. Large positive values indicate that for detection of SCOP family 7.3.5.2 (Spider toxins) the corresponding feature is overrepresented in positive training sequences as compared with the negative training sequences. Table 2 shows a list of the 10 most discriminative words for the positive training sequences associated with superfamily 7.3.5 after kernel-based training (see section "Methods"). This table allows to identify the most discriminative features of a particular superfamily in biologically meaningful terms. For an exemplary analysis of globally important features, Table 3 shows the 10 most discriminative features of four experiments associated with families from the SCOP class "All alpha proteins". This class contains protein domains whose structure is essentially formed by alpha helices. The features in Table 3 correspond to particular dimensions in the word correlation feature space in terms of an amino acid pair at particular word positions.

Discussion

Table 1 indicates that the best ROC performance for the WCM approach on the SCOP benchmark setup is achieved using word length K = 6. For longer words, the ROC performance gradually decreases but still remains comparable with the other methods. However, the ROC50 performance for longer words increases and nearly achieves the ROC50 performance of the Oligomer Distance Histogram method for monomers. While prediction scores of all test examples are used for computation of the ROC performance, the ROC50 performance takes into account only prediction scores up to 50 false positive examples. This corresponds to an evaluation of the ROC curve in regions where a maximum number of 50 false positive examples are allowed for computation of specificity. Therefore, the results indicate that longer words yield more specific predictions. However, as compared with the local alignment kernel method [10] the WCM method performs inferior in terms of ROC and ROC50 scores. On the contrary, the detection performance of this approach depends on several hyperparameters. Table 1 shows that the performance of the WCM approach does not depend critically on the word length K. This obviates the tuning of this method parameter for different setups. However, longer words may be more suitable to identify biologically meaningful features or regions within sequences than short words.

Comparison to closely related approaches

Surprisingly, our WCM approach for K = 1 (WCM_1) outperforms the K-mer Spectrum method for K = 3 (Spectrum (3)) in terms of ROC and ROC50 performance. Technically, the WCM_1 feature space corresponds to the feature space of the Spectrum (1) method, i.e. the amino acid composition. This feature space comprises only 20 dimensions, and thus allows for fast and memory efficient representation and classification of sequences. This suggests that this simple approach could be useful for large-scale remote homology detection. In [9], the authors applied the Spectrum method to a similar remote homology detection setup as described here (see also [5]). However, the authors limit the evaluation of detection performance to the Spectrum (3) and Spectrum (4) method, respectively. Thereby, the Spectrum (3) method outperformed the Spectrum (4) method in terms of ROC50 performance. Figure 3 shows a comparison of the ROC performance for the Spectrum method and the WCM approach using word length K = 1, .., 6. It is clearly visible that the performance of the Spectrum rapidly decreases for growing word length while the performance of our method continuously increases. This results from the fact that the WCM feature space for a word length K > 1 completely includes the WCM feature space for shorter words. In contrast, the Spectrum feature space associated with a particular word length does not include the feature

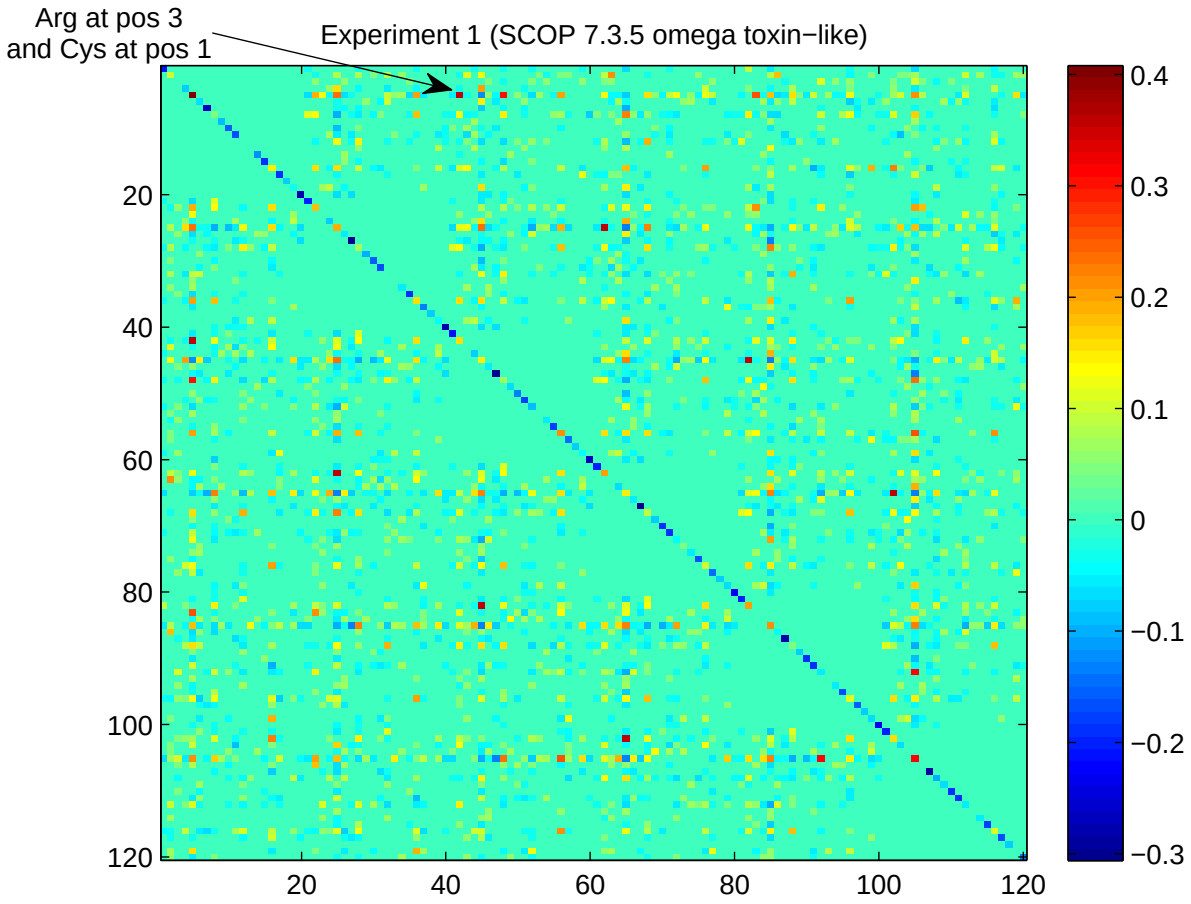

Figure 2
Discriminant of SCOP superfamily 7.3.5 in the WCM space. Word correlation matrix representation of the discriminant weight vector of superfamily 7.3.5 (omega toxin-like) after training using $K = 6$ (see text). Rows and columns correspond to occurrences of amino acids at two particular word positions for the first and second occurrence, respectively. Red (blue) matrix elements represent large positive (negative) discriminant weight values according to the color bar on the right hand side.

space for shorter words by definition. The results indicate that the Spectrum method is rather unsuitable for use of longer words. This can be traced back to the fact that the number of exact matches rapidly decreases for growing word length. This results in very small values for the similarity between two non-identical sequences. Therefore, the incorporation of inexact matches as in [8] is necessary for use with longer words. In [15], the authors present several string kernels that are based on inexact matching of *K*-mers. These methods realize inexact matching by a so-called "mismatch" or "mutation neighborhood" using a binary match function with specific invariance properties. In that case, a particular *K*-mer is mapped to several dimensions in the feature space of the *K*-mer Spectrum. The similarity of two *K*-mers can then be calculated as the dot product in this feature space. However, this feature space grows exponentially with *K* and is difficult to interpret in terms of biological sequence features. Furthermore, classification with the discriminant in this feature space for large *K* is demanding in terms of memory requirements. In contrast, the WCM method is based on a more "continuous" similarity measure between two words (see also equation (2)) rather than on a binary match criterion. The corresponding feature space only grows quadratically with *K* and each feature space dimension directly corresponds to a biologically meaningful sequence feature. In addition, the WCM approach allows for memory efficient classification with the discriminant in feature space.

Table 2: Ordered list of discriminative words for experiment 1.

#	Score	Word	Count
1	7.066	CCSGSC	3
2	6.930	CCSRKC	2
3	6.419	CRSGKC	4
4	5.451	CCRSCN	2
5	5.354	GRSGKC	1
6	5.215	CSRKCN	2
7	5.142	GRGSRC	1
8	4.979	CSGRGS	1
9	4.812	CCTGSC	4
10	4.789	SYNCCR	2

List of 10 most discriminative words for positive training sequences of experiment 1 according to SCOP superfamily 7.3.5 using word length $K = 6$. Words are sorted according to their word score. The first and second column correspond to rank and score of a word, respectively. The third column contains the word as amino acid sequence in IUPAC one-letter code. In the fourth column, the number of occurrences of a particular word in the positive training sequences are shown.

Another possibility to deal with the decrease of exact matches for longer words is the combination of Spectrum kernel matrices based on different word lengths. Table 1 shows that the results for the Spectrum method using combined kernel matrices up to a maximum word length are only slightly inferior as compared with the WCM approach using the respective maximum word length. Note that the WCM approach does not require to identify a suitable combination of different kernels to achieve good prediction performance.

Interpretation of discriminative features

The WCM feature space is useful for identification of discriminative features that have been learned from the data. In Figure 2, the discriminant weight vector is visualized in the WCM feature space that allows to analyze discriminative features in terms of the corresponding sequence properties. For example, the highlighted matrix element in Figure 2 indicates that for positive training sequences of superfamily 7.3.5 the occurrence of Cysteine (C) at the first word position in combination with Arginine (R) at the third word position is highly discriminative. This feature may not be detected in the sequences associated with this superfamily if only unsupervised methods, e.g. motif finders are used. The reason is, that the combination can only be observed in few cases but nevertheless occurs more often than in protein sequences from unrelated families. Therefore, our discriminative approach can help to identify features that are likely to be overlooked by unsupervised methods. These features can readily be used for analysis of more specific biological properties of the particular protein family.

Table 2 shows a list of the 10 most discriminative words in positive training sequences of superfamily 7.3.5 (omega toxin-like) after kernel-based training. Some of these words are very similar, e.g. words no. 1, 2, 4 and 9 begin with two Cysteine residues and words no. 1, 2 and 9 end with a Cysteine, too. Word no. 10 also shows two successive Cysteine residues, but in this case at word positions 4 and 5. The last column of Table 2 contains the number of occurrences of a particular word in the set of positive training sequences. It can be seen that this number is not directly related to the discriminative word score in the second column. This indicates that discriminative learning and unsupervised counting of words produce motifs with different meanings. The most discriminative word (CCSGSC) can easily be identified in the multiple alignment of the Omega-toxin family in the Pfam database [20]. The figure in Additional file 1 shows the full alignment of this family, which is a member of the omega toxin-like superfamily according to experiment 1 in the remote homology detection setup. In two sequences, the word exactly matches the subsequence and in 5 of the 6 remaining sequences the word only differs by one amino acid. In this case, exact word matches cannot

Table 3: Ordered list of discriminative features.

#	Family 1.27.1.1	Family 1.27.1.2	Family 1.36.1.2	Family 1.36.1.5
1	Leu@5, Leu@5	Leu@6, Leu@6	Thr@1, Val@5	Ala@1, Lys@5
2	Leu@6, Leu@6	Leu@5, Leu@5	Thr@2, Val@6	Ala@2, Lys@6
3	Leu@1, Leu@1	Leu@1, Leu@1	Val@1, Ser@2	asp@2, asp@2
4	Leu@2, Leu@2	Leu@2, Leu@2	Val@2, Ser@3	asp@3, asp@3
5	Leu@4, Leu@4	Leu@4, Leu@4	Val@5, Ser@6	asp@1, asp@1
6	Leu@3, Leu@3	Leu@3, Leu@3	Val@4, Ser@5	asp@4, asp@4
7	Leu@1, Leu@5	Leu@1, Leu@5	Val@3, Ser@4	asp@6, asp@6
8	Leu@2, Leu@6	Leu@2, Leu@6	Val@2, Thr@6	asp@5, asp@5
9	Glu@6, Glu@6	Glu@1, Glu@1	Val@1, Thr@5	Ala@1, Leu@2
10	gly@1, gly@1	Glu@2, Glu@2	Ser@1, Thr@4	Ala@2, Leu@3

List of 10 most discriminative features for four superfamilies associated with the SCOP class "All alpha proteins". Features are sorted in descending order according to their absolute discriminative weight (not shown). The first column corresponds to the rank of a feature and the succeeding columns contains the description of the feature in the word correlation feature space in terms of a pair of amino acids (in IUPAC three-letter code) at particular word positions. Features that are associated with negative discriminative weights are printed with lowercase first letters.

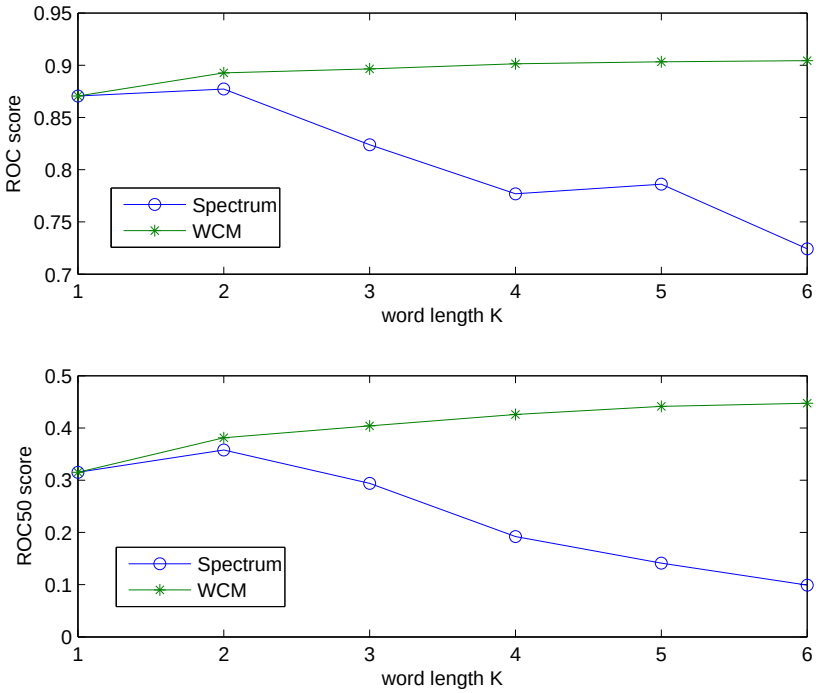

Figure 3
Comparison of ROC and ROC50 performance for Spectrum method and WCM method. The figure shows the mean ROC and ROC50 performance over 54 experiments for the Spectrum method and the word correlation method (WCM) using word length K = 1, .., 6.

capture the conserved region of the sequences. In contrast, the WCM method is able to capture this similarity in terms of high scoring words. Figure 1 shows score profiles of the first 5 positive test sequences associated with experiment 1 using word length K = 6. All score profiles have a global maximum that corresponds to a discriminative sequence region. For example, in sequence no. 5 the score maximum corresponds to the word CCSQSC, which is very similar to the most discriminative word in the training sequences. This indicates that score profiles may be used to identify characteristic sequence regions.

Table 1 shows that after kernel-based training the average number of support vectors of the WCM approach is significantly lower than that of the local alignment kernel and the Mismatch and Spectrum kernel methods. This may suggest that WCMs might be a more concise and accurate representation of globally important protein features such as secondary structure elements. Table 3 shows the most discriminative features of four protein families from the SCOP class "All alpha proteins". In the protein families 1.27.1.1 and 1.27.1.2 (long-chain/short-chain cytokines), the occurrences of Leucine at word position 1 and 5 (2 and 6) are among the top ten discriminative features. Similarly, in the protein families 1.36.1.2 and 1.36.1.5 (phage repressors/bacterial repressors) the occurrences of Valine at word position 1 and Threonine at word position 5 as well as the occurrences of Alanine at word position 1 and Lysine at word position 5 belong to the top ten discriminative features. This indicates that the characteristic distance of 4 residues between linked amino acids in an alpha helix provides a discriminative sequence feature in these families.

Computational efficiency

In section "Methods", we pointed out that our WCM approach is very efficient in terms of computation time requirements for feature extraction from sequences. The feature-based calculation of the 4352 × 4352 kernel matrix for the WCM approach using word length K = 6 takes 31.62 seconds. This is by orders of magnitude faster than the computation of the kernel matrix for the local alignment kernel method, which nearly takes 2 hours. However, feature-based computation of the kernel matrix

can also be applied to the Spectrum method. For $K = 1$ ($K = 3$), the calculation only requires 6.9 (10) seconds. For classification of new sequences with alignment-based kernel methods all kernel functions between the test sequences and support vector sequences, i.e. sequences with a non-zero weight after kernel-based training, have to be evaluated. For example, for classification of a new sequence with the local alignment kernel on average 2640 kernel function evaluations need to be computed. Using the software provided by the authors of [10], evaluation of a single kernel function requires on average 0.36 *ms* CPU time. In total, this yields 0.95 *s* for classification of a single sequence.

For classification of new sequences with the WCM approach, the discriminant weight vector in feature space can be used instead of the kernel-based evaluation. This dramatically reduces the computational effort for classification, because only transformation of the new sequence to a WCM feature vector and calculation of the dot product of that vector with the discriminant weight vector are necessary. If indicator vectors are used for amino acid representation, the score of a sequence can be computed by summing up all weight vector entries according to the number of occurrences of the associated pair of amino acids at two particular word positions in the sequence. We implemented a fast MATLAB® version of this scoring procedure that requires on average 0.09 *ms* for scoring of a single sequence in the SCOP setup using word length $K = 6$. This is more than 10000 times faster than scoring with the local alignment kernel and implies a different category of computation time requirements for ranking of potential homologs in a large database. For example, the UniProt Protein Knowledgebase [21] release 12.8 contains 5678599 protein sequences, which could be potential targets in a homology detection task. In this case, scoring with the local alignment kernel would require more than 60 days on a single machine. Although not directly comparable in terms of detection performance, the feature-based scoring with the WCM approach takes less than 9 minutes. For comparison with the Spectrum method, we also implemented a fast procedure that scores a protein sequence using a feature space discriminant as produced by the Spectrum kernel method. For $K = 1$ ($K = 3$), scoring of the UniProt database takes about 4 (10) minutes. In principle, the computational cost for classification of new sequences with alignment-based kernels grows linearly with the number of training sequences. Therefore, the application of these methods to large-scale classification setups is problematic, too. In contrast, the computational cost for classification with the feature-based methods only grows linearly with the number of feature space dimensions. Therefore, our method is suitable for large-scale classification setups. In particular, the WCM approach could be very useful to reduce the number of target sequences or target families. This reduced set may then be further investigated with more specific alignment-based methods.

Conclusion

In this work, we presented a new approach for protein sequence representation based on word correlation matrices (WCM). WCMs arise from a sequence kernel defined by average pairwise word similarity between two sequences. The approach shows comparable detection performance to state-of-the-art methods for protein remote homology detection. Our method includes a single kernel parameter that specifies the word length. We showed, that the detection performance does not critically depend on this parameter. Our results indicate, that for remote homology detection the word length parameter can be fixed to $K = 6$ for time and memory efficiency. Our protein sequence representation is associated with an explicit feature space in terms of word correlations. The discriminant weight vector in feature space can be used for fast classification of new sequences and intuitive interpretation of discriminative features.

In general, the basic word similarity measure can be defined in other ways than presented in this work. For example, in the definition of the word similarity measure (Equation (2) in section "Methods") a word substitution matrix can be inserted between the word vectors to include prior knowledge about the similarity of particular words. On the other hand, such substitution matrices are usually problem-specific, i.e. they should depend on the application. Furthermore, the substitution matrix has to be positive semidefinite so that the similarity measure still implies a valid sequence kernel.

Like other explicit feature space methods, our representation approach can be combined with different feature selection techniques. This would be useful in cases where a small set of relevant features has to be identified. Finally, the WCM approach is not limited to protein sequences, but can also be used for DNA or RNA sequence representation. In this case, the word length possibly has to be chosen larger to obtain meaningful features. The investigation of these possibilities will be part of future work.

Authors' contributions

TL did the experimental evaluation and drafted parts of the manuscript. PM designed the method and drafted parts of the manuscript. Both authors read and approved the final manuscript.

Additional material

Additional file 1

Pfam full alignment of the Omega-toxin family (PF06357). *The file pfamAln.png contains a screenshot from the Pfam website (see [22]) which shows the multiple alignment of all member sequences of the Omega-toxin family (Pfam ID PF06357). The Omega-toxin family belongs to the omega toxin-like superfamily. Some of the discriminative words in Table 2 can be identified in the sequences (see text).*
Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2105-9-259-S1.png]

Acknowledgements

This work was partially supported by Federal Ministry of Research and Education project "MediGRID" (BMBF 01AK803G).

References

1. Smith TF, Waterman MS: **Identification of common molecular subsequences.** *J Mol Biol* 1981, **147:**195-197.
2. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: **Basic local alignment search tool.** *J Mol Biol* 1990, **215(3):**403-410.
3. Krogh A, Brown M, Mian IS, Sjolander K, Haussler D: **Hidden Markov models in computational biology. Applications to protein modeling.** *J Mol Biol* 1994, **235(5):**1501-1531.
4. Park J, Karplus K, Barrett C, Hughey R, Haussler D, Hubbard T, Chothia C: **Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods.** *J Mol Biol* 1998, **284(4):**1201-1210.
5. Jaakkola T, Diekhans M, Haussler D: **Using the Fisher kernel method to detect remote protein homologies.** *Proc Int Conf Intell Syst Mol Biol* 1999:149-158.
6. Liao L, Noble WS: **Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships.** *J Comput Biol* 2003, **10(6):**857-868.
7. Ben-Hur A, Brutlag D: **Remote homology detection: a motif based approach.** *Bioinformatics* 2003, **19(Suppl 1):**26-33.
8. Leslie CS, Eskin E, Cohen A, Weston J, Noble WS: **Mismatch string kernels for discriminative protein classification.** *Bioinformatics* 2004, **20(4):**467-476.
9. Leslie C, Eskin E, Noble WS: **The spectrum kernel: a string kernel for SVM protein classification.** *Pac Symp Biocomput* 2002:564-575.
10. Saigo H, Vert JP, Ueda N, Akutsu T: **Protein homology detection using string alignment kernels.** *Bioinformatics* 2004, **20(11):**1682-1689.
11. Kuang R, Ie E, Wang K, Wang K, Siddiqi M, Freund Y, Leslie C: **Profile-based string kernels for remote homology detection and motif extraction.** *J Bioinform Comput Biol* 2005, **3:**527-550.
12. Rangwala H, Karypis G: **Profile-based direct kernels for remote homology detection and fold recognition.** *Bioinformatics* 2005, **21(23):**4239-4247.
13. Dong QW, Wang XL, Lin L: **Application of latent semantic analysis to protein remote homology detection.** *Bioinformatics* 2006, **22(3):**285-290.
14. Lingner T, Meinicke P: **Remote homology detection based on oligomer distances.** *Bioinformatics* 2006, **22(18):**2224-2231.
15. Leslie C, Kuang R: **Fast String Kernels using Inexact Matching for Protein Sequences.** *J Mach Learn Res* 2004, **5:**1435-1455.
16. Murzin AG, Brenner SE, Hubbard T, Chothia C: **SCOP: a structural classification of proteins database for the investigation of sequences and structures.** *J Mol Biol* 1995, **247(4):**536-540.
17. **gobics.de: Thomas Lingner** [http://www.gobics.de/thomas/]
18. Chapelle O: **Training a Support Vector Machine in the Primal.** *Neural Comp* 2007, **19(5):**1155-1178.
19. Cohen G, Hilario M, Sax H, Hugonnet S, Geissbuhler A: **Learning from imbalanced data in surveillance of nosocomial infection.** *Artif Intell Med* 2006, **37:**7-18.
20. Finn R, Mistry J, Schuster-Böckler B, Griffiths-Jones S, Hollich V, Lassmann T, Moxon S, Marshall M, Khanna A, Durbin R, Eddy S, Sonnhammer E, Bateman A: **Pfam: clans, web tools and services.** *Nucleic Acids Res* 2006, **34:**D247-251.
21. UniProtConsortium: **The universal protein resource (UniProt).** *Nucleic Acids Res* 2008, **36:**D190-195.
22. **PFAM: Family: Omega-toxin (PF06357)** [http://pfam.janelia.org/family/alignment/downahtml?acc=PF06357&alnType=full&viewer=html]

Anhang C

Artikel 3

Fast Target Set Reduction for Large-Scale Protein Function Prediction: A Multi-class Multi-label Machine Learning Approach

Thomas Lingner and Peter Meinicke

Department of Bioinformatics, Institute for Microbiology and Genetics, University of Göttingen, Goldschmidtstr. 1, 37077 Göttingen, Germany
thomas@gobics.de, pmeinic@gwdg.de

Abstract. Large-scale sequencing projects have led to a vast amount of protein sequences, which have to be assigned to functional categories. Currently, profile hidden markov models and kernel-based machine learning methods provide the most accurate results for protein classification. However, the prediction of new sequences with these approaches is computationally expensive. We present an approach for fast scoring of protein sequences by means of feature-based protein sequence representation and multi-class multi-label machine learning techniques. Using the Pfam database, we show that our method provides high computational efficiency and that the approach is well-suitable for pre-filtering of large sequence sets.

Keywords: protein classification, large-scale, multi-class, multi-label, Pfam, homology search, metagenomics, target set reduction, protein function prediction, machine learning.

1 Introduction

In the last decade, the number of known protein sequences has rapidly increased, mainly due to several genome sequencing projects. Currently, large-scale shotgun sequencing as used in metagenomics produces large amounts of protein sequences with unknown phylogenetic origin [1]. For further analysis of these protein sequences, they have to be classified in terms of structural and functional properties. Because experimental determination is time-consuming and expensive, several computational methods have been proposed for protein function prediction and protein classification [2,3].

Widely-used methods for functional assignment are based on annotation transfer from homologues by means of pairwise sequence alignments. Here, heuristic approaches such as BLAST [4] are used to search well-annotated databases for similar protein sequences. However, these methods require the evaluation of all pairwise alignments, which is computationally demanding for large sets of sequences. Furthermore, pairwise alignment methods often fail when sequence similarity is below 60% residue identity. Finally, this kind of annotation transfer may be erroneous and may further lead to propagation of erroneous annotation [2,3].

K.A. Crandall and J. Lagergren (Eds.): WABI 2008, LNBI 5251, pp. 198–209, 2008.

Another widely-used approach for protein classification is based on models that statistically represent properties of a set of related sequences, e.g. protein families. For example, the Pfam database [5] provides a comprehensive collection of manually curated multiple alignments of protein domain families. Each family is represented by a Profile Hidden Markov Model (PHMM, [6]) that has been constructed from the multiple alignment. The PHMMs can be used to classify new sequences into the Pfam categorization scheme, e.g. with the HMMER package (`http://hmmer.janelia.org/`). The threshold for assignment of a sequence to a particular model in Pfam is chosen to detect only verified family members. Because of the resulting highly specific models and the amount of manually verified annotation, Pfam and HMMER provide a highly valuable combination for automatic functional assignment of protein sequences. However, PHMMs may fail in the detection of remote homologues, i.e. homologues with a sequence similarity below 30% residue identity [7]. A major drawback of the PHMM approach is that for classification, every candidate sequence has to be aligned to all models. This is computationally demanding and several methods have been proposed to tackle this problem, e.g. by means of parallelization or hardware-acceleration [8]. Another approach is to use a "pre-filter", i.e. a computationally more efficient algorithm that reduces the set of candidate sequences or models (e.g. `http://www.microbesonline.org/fasthmm/`). For example, the Pfam website (`http://pfam.janelia.org/help`) provides a perl script that uses the BLAST method for pre-filtering. According to the author of the script, a speed-up factor of 10 may be achieved, accompanied by a slightly reduced sensitivity.

As an alternative, computationally more efficient alignment-free methods could be used as pre-filters. Recently, several feature-based machine learning approaches have been proposed for protein classification (for an overview see [9,10,11]) and detection of remote homologues [12,13,14,15]. Many of these discriminative methods have been shown to provide state-of-the-art classification performance. Therefore, feature-based approaches in general could be used for pre-filtering, too. However, so far the evaluation of these methods has been limited to detection of members of a particular protein family or problems with few exemplary categories and a relatively small number of sequences.

In that context, alignment-based kernel methods have been shown to outperform other approaches for remote homology detection [16,17]. In comparison with fast feature-based methods, kernel-based approaches are computationally expensive [15] and thus unsuitable for large-scale classification problems. As a consequence, the application of kernel-based methods has been limited to evaluation on small-scale setups, e.g. on the widely-used setup described in [7] including 54 classes (superfamilies).

If the classification problem does involve many functional categories, discriminative methods require a multi-class evaluation setup. The most common way to handle a multi-class problem is to split it up into isolated two-class one-against-all problems in order to utilize binary classification methods [18,19]. However, for large-scale multi-class problems, discriminative training and hyperparameter search for several thousand classifiers is computationally demanding. To our

knowledge, the application of discriminative approaches in computational biology has been limited to small-scale setups so far.

In protein classification, an example may belong to several functional categories, e.g. multi-domain proteins consist of several functional regions. This gives rise to so-called "multi-label" learning problems. Multi-class multi-label machine learning methods have first been applied within the research field of text categorization [20]. Recently, multi-label methods have also been applied in bioinformatics, for example for analysis of gene expression profiles [21,22], protein subcellular localization prediction [23] and also for protein classification [24]. Usually, multi-label classification can be splitted into a ranking of examples which provides a correctly ordered list of categories and a set size prediction method, which determines the number of relevant classes.

In this work, we present an approach for ranking of protein sequences that is based on feature-based protein sequence representation methods and multi-class multi-label machine learning techniques. Based on an evaluation setup involving a large part of the Pfam database, we show that our approach can be used as a computationally efficient pre-filter for protein function prediction methods.

2 Methods

2.1 Machine Learning Approach

In order to realize a multi-label ranking scheme within a highly imbalanced large-scale setup with $M = 4423$ classes, we implemented an extension of the so-called "regularized least squares classifiers" [25]. For computational efficiency, we utilize a linear one-against-all approach with simultaneous training of all M discriminants.

For ranking of a sequence, we compute the corresponding feature vector $\mathbf{x} \in \mathbb{R}^d$ and the linear discriminant of class i based on the weight vector $\mathbf{w}_i$ to obtain a score from the corresponding scalar product:

$$s_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x} \tag{1}$$

where the upper ^{T} indicates vector (matrix) transposition. The sorting of all M scores s_i in descending order then produces the ranks of the classes, i.e. the first element in the sorted list with the highest score indicates the highest rank (M).

For the training of M discriminants we use N feature vectors $\mathbf{x}_j$ and the corresponding binary indicator vectors $\mathbf{y}_j$ for representation of the labels. The i-th dimension of $\mathbf{y}_j$ only has a non-zero value ($= 1$) if sequence j is associated with class i. The discriminant weight vectors $\mathbf{w}_i$ are represented as columns in the $d \times M$ weight matrix $\mathbf{W}$ and the inverse class sizes are collected in the balancing vector $\mathbf{b} = [1/n_1, \ldots, 1/n_M]^T$ with n_i counting the number of sequences associated with class i. For optimization of the weight matrix we minimize the regularized squared error criterion:

$$E(\mathbf{W}) = \sum_{j=1}^{N} \mathbf{b}^T \mathbf{y}_j \|\mathbf{W}^T \mathbf{x}_j - \mathbf{y}_j\|^2 + \lambda \|\mathbf{W}\|_F^2 \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Minimization of the above cost function tries to approximate the indicator vectors by means of the discriminant scores. Thus, the training forces a discriminant to produce relatively high scores only if examples are associated with the corresponding class. In addition, the λ-weighted regularization term bounds the squared norm of the weight vectors. A suitable value of λ then provides a compromise between a low approximation error and good generalization. The balancing vector $\mathbf{b}$ down-weights the contribution of examples, which are exclusively associated with overrepresented classes.

Now, consider the N feature vectors $\mathbf{x}_j$ as column vectors in a $d \times N$ training matrix $\mathbf{X}$ and the binary indicator vectors $\mathbf{y}_j$ as columns in an $M \times N$ label matrix $\mathbf{Y}$. Using the $N \times N$ diagonal matrix $\mathbf{D} = \mathrm{diag}(\mathbf{Y}^T \mathbf{b})$ of example-specific balancing weights, the minimizer of the regularized error can be written as:

$$\hat{\mathbf{W}} = (\mathbf{X}\mathbf{D}\mathbf{X}^T + \lambda \mathbf{I})^{-1} \mathbf{X}\mathbf{D}\mathbf{Y}^T \tag{3}$$

where $\mathbf{I}$ is the $d \times d$ identity matrix. The above solution requires inversion of a $d \times d$ matrix, which practically limits the approach to feature spaces with a moderate dimensionality. However, on current computers, spaces with approximately 10^4 dimensions can be used without problems. For ranking of a set of test examples represented by columns of a matrix $\mathbf{X}_{\mathrm{test}}$, the $M \times N$ score matrix $\mathbf{S}$ of all examples can be efficiently computed using the matrix product

$$\mathbf{S} = \hat{\mathbf{W}}^T \mathbf{X}_{\mathrm{test}} \,. \tag{4}$$

The j-th column of $\mathbf{S}$ contains all M discriminant scores of the j-th example.

2.2 Protein Sequence Representation

In order to apply our machine learning approach to the protein ranking problem, we have to represent all amino acid sequences in a common vector space. Here, we describe two exemplary protein sequence representation methods that have been used for protein classification and remote homology detection and which can be associated with an interpretable feature space of moderate dimensionality.

***k*-mer Spectrum.** The k-mer Spectrum of a sequence S counts the occurrences of all k-length words in S and can be represented as a vector in the k-mer Spectrum feature space. According to the number of different k-mers, for protein sequences this feature space comprises 20^k dimensions. The dimension associated with a particular k-mer K counts the occurrences of K in a sequence.

For large values of k, the k-mer Spectrum feature space becomes very large. For example, the trimer ($k = 3$) Spectrum feature space comprises a moderate number of 8000 dimensions, whereas the feature space for $k = 5$ consists of $3.2 * 10^6$ dimensions. On the other hand, the algorithmic complexity to calculate

the feature space representative $\mathbf{x}$ of a sequence of length L is only $O(L)$. The k-mer Spectrum for small values of k is a time and memory efficient protein sequence representation method and thus provides a suitable vector space for large-scale machine learning problems. In order to provide comparability with respect to sequences of different length, the feature vectors are normalized to Euclidean unit length.

For $k = 1$ and $k = 2$, the k-mer Spectrum corresponds to the amino acid composition and dipeptide composition of a sequence, respectively. Both methods have been successfully applied to protein classification [10,11]. In comparative studies, the dipeptide composition has been shown to perform similarly to other feature-based sequence representation methods [9]. The k-mer Spectrum has also been used for remote homology detection in [12], where it has been shown to outperform unsupervised approaches.

Oligomer Distance Histograms. In [15], Oligomer Distance Histograms (ODH) have been introduced for protein sequence representation and remote homology detection, where they have been shown to outperform the k-mer Spectrum. In the distance-based feature space, for each pair of k-mers (oligomers) there exists a specific histogram counting the occurrences of that pair at certain distances. For the set of $M = 20^k$ different k-mers, the feature vector of a sequence S comprises M^2 distance histogram vectors. Considering a maximum sequence length L_{max}, each histogram vector contains the counts of two k-mers at distances $d = 0, \ldots, L_{max} - k + 1$. For $k > 1$ and large L_{max} the ODH representation gives rise to a high dimensional feature space. For example, the feature space for $k = 3$ (trimers) and a maximum sequence length of $L_{max} = 1000$ comprises about $6.4 * 10^{10}$ dimensions. In this case, our feature-based machine learning approach cannot be applied.

On the other hand, in [15] the best detection performance has been achieved for $k = 1$ (monomers). Furthermore, sequences with less than L_{max} residues do not contribute to feature space dimensions associated with this distance. Therefore, a restriction of the maximum distance provides a suitable means to reduce the feature space dimensionality of ODHs. As an example, the ODH feature space for monomer pairs with a maximum distance of $D_{max} = 10$ residues comprises only 4020 dimensions, which is about half the number of dimensions required for the feature space associated with the trimer Spectrum. In that way, the ODH representation allows a fine-grained control of the feature space size, which linearly depends on the maximum distance. Note that the distance-based feature space for monomers incorporates the amino acid composition ($D = 0$), the dipeptide composition ($D = 1$) and trimer counts according to a central mismatch ($D = 2$).

The systematic evaluation of all pairwise k-mers in a sequence of length L to calculate the feature space representative $\mathbf{x}$ is of algorithmic complexity $O(L^2)$. If the maximum distance is restricted to D_{max}, the algorithmic complexity reduces to $O(D_{max}L)$. As the Spectrum feature vectors, ODH feature vectors are normalized to unit length.

2.3 Performance Measures

For evaluation of our approach, we use different performance indices. First, we measure the so-called *coverage* to analyze the ranking performance [22]. We define the *minimal set size* of a test example as the cardinality of the minimal set that includes all ranks above or equal to the minimal rank of a true category. Then, the coverage is defined as the average minimal set size over all examples reduced by 1. This measure is well-suited to evaluate our approach in terms of a pre-filtering for target set reduction. In this case, the coverage reflects the number of functional categories that have to be considered in a subsequent function prediction stage. Therefore, a smaller coverage allows a higher possible speed-up of the prediction. The coverage measure is widely used in multi-label learning problems. Besides the "classical" mean coverage, we also consider the median coverage, i.e. the median of the above minimal set size over all test examples. In addition, we use the so-called *one-error* to measure the ranking performance. The one-error evaluates how often on average the top-ranked category is not a true class of a particular example. Therefore, a one-error value of 0 is desirable, which means that all examples would have a correct functional category assignment for the highest rank.

In addition, we measure the average area under ROC and ROC50 curve over all families to measure the detection performance w.r.t. potential family members. The ROC curve reflects the dependency of the the false positive rate (1−specificity) on the true positive rate (sensitivity) w.r.t. to variation of the classification threshold. The ROC50 score is the area under curve up to 50 false positives. ROC scores are particularly useful for analysis of rank-ordered lists and imbalanced problems [26] and are widely used in the evaluation of remote homology detection performance [7,15,16].

3 Experimental Setup

In order to evaluate our approach, we developed a test setup based on the Pfam database [5]. Pfam is a widely-used, comprehensive and well-annotated collection of protein domain families. Pfam 22.0 (released in July 2007) consists of 9318 families in the Pfam-A section, i.e. the manually curated part of the database. Each family comprises a seed alignment, which contains selected representative sequences of the family. For our setup, we use these seed sequences, which in total add up to 217445 examples in Pfam 22.0.

The seed alignments of Pfam are based on protein domains, i.e. functional regions of protein sequences. When functional categories have to be assigned to unannotated sequences, the boundaries of these domains are unknown. Therefore, we consider the complete protein sequences associated with the domains. Furthermore, multi-domain proteins include several domains, which may realize different functions. Therefore, multi-domain proteins may be associated with several Pfam families. In our setup, the label indicator vector of a protein sequence refers to all its relevant classes, i.e. Pfam families that have a significant match to this sequence.

For training of the discriminant weight vectors we use the regularized least squares approach described in section "Methods". For this purpose, we calculated the feature vectors for the k-mer Spectrum method using $k = 1, 2, 3$ and for the ODH method using $D_{max} = 10, 20, 30$. Higher values for k and D_{max} result in feature spaces that complicate the training due to high dimensionality. We evaluated different values of the regularization parameter $\lambda = \{10^m | m = -4, -3, \ldots, 4\}$ by means of 5-fold cross-validation. In order to provide a sufficiently high number of training and test examples for each class, we only consider Pfam families with at least 10 different seed sequences. In total, our setup consists of 147003 protein sequences from 4423 Pfam families. The size of the families varies from a minimum of 10 to a maximum of 1670 example sequences and thus gives rise to a highly imbalanced classification problem.

All methods were implemented using the Matlab® programming language. Files containing sequences and labels associated with our evaluation setup can be downloaded from `http://www.gobics.de/thomas/data/pfam/`

4 Results and Discussion

The results of our performance evaluation are summarized in table 1. As a main result, the ODH method clearly outperforms the k-mer Spectrum in terms of all performance indices. In particular, the coverage (columns 3 and 4) and the one-error values (column 5) are substantially better for the ODH method. In terms of ROC/ROC50 values (columns 6 and 7), all methods except for the monomer Spectrum show a good performance. On the other hand, the coverage and one-error values are more suitable measures for the utility of a particular method for target set reduction on multi-label problems (see also section "Performance Measures"). Although not tested here, we expect the combination of Pfam PHMMs and HMMER to perfectly classify the sequences. This would implicate a value of 1 for ROC/ROC50 values and 0 for the one-error. The mean coverage is expected to correspond to the average number of true categories over all examples reduced by 1, which is approximately 0.6 in our test setup.

The results in table 1 also show that for the ODH method the highest maximum distance $D_{max} = 30$ produces better results than smaller maximum distances. Similarly, higher values of k for the Spectrum method provide better detection performance and less coverage. Note that the ODH method for $D_{max} = 10$ performs better than the Spectrum method for $k = 3$, although the feature space associated with the ODH method comprises only about half the number of dimensions as compared to that of the Spectrum method. This indicates that the choice of a suitable feature mapping is more important than just a high dimensionality of the feature space. In [15], the ODH method was introduced without restriction of the maximum distance and clearly outperformed the Spectrum method on a small-scale benchmark setup for protein remote homology detection. Our results in this work indicate that the introduction of a maximum distance for the ODH method is a suitable means for application of this method to large-scale problems. In column 8, the best regularization

Table 1. Performance results of different methods. The first column indicates the method, where "Spectrum" refers to the k-mer Spectrum and "ODH" refers to Oligomer Distance Histograms using $k = 1$ with maximal distance D_{max} (see also section "Methods"). Column 2 denotes the dimensionality of the feature space associated with a particular method. Columns 3-7 show the different performance indices as described in section "Methods" in terms of average values over 5 cross-validation test folds. "ROC" and "ROC50" refer to the average area under ROC/ROC50 curve over all families with at least 10 positive test examples in each fold. In column 8, the best regularization parameter λ associated with a particular method is shown.

Method	d	Coverage		One-error	ROC	ROC50	best λ
		mean	median				
Spectrum ($k = 1$)	20	452.4	243.8	0.95	0.925	0.046	0.001
Spectrum ($k = 2$)	420	214.9	65.0	0.85	0.978	0.563	0.001
Spectrum ($k = 3$)	8000	116.7	4.8	0.57	0.987	0.827	1
ODH ($D_{max} = 10$)	4020	65.3	3.8	0.55	0.993	0.840	0.001
ODH ($D_{max} = 20$)	8020	47.6	2.0	0.43	0.995	0.881	0.001
ODH ($D_{max} = 30$)	12020	41.6	1.2	0.37	0.995	0.894	0.01

parameter λ w.r.t the coverage over all 5 cross-validation test folds is shown. The values indicate that the regularization parameter has to be chosen larger for feature spaces with higher dimensionality. On the other hand, our results only showed a small variation of the performance for a broad range of λ-values.

Column 3 of table 1 shows that on average only about 42 of 4423 families are required to detect all functional categories of a particular test example with the ODH method for maximum distance $D_{max} = 30$. If the approach is used as a pre-filter for protein function prediction, this corresponds to a speed-up factor of 106. This is one order of magnitude higher than the speed-up that is usually achieved with alignment-based methods. Furthermore, the one-error for the ODH method with $D_{max} = 30$ indicates that for 63% of the test examples the class associated with the highest rank implies the correct assignment to a functional category.

However, the total speed-up also depends on the computational efficiency of the target set reduction method. Therefore, we measured the running time for classification with different methods on an AMD Opteron 870 workstation. Table 2 shows that the ranking with the monomer Spectrum ($k = 1$) took 585 seconds for all categories and 147003 examples. As another example, the ranking with the ODH method using $D_{max} = 30$ took 3380 seconds. On average, 23 ms are required for ranking of a single sequence with this more sensitive method as compared with the monomer Spectrum method. This is about 1270 times faster than functional assignment of a single sequence with the HMMER package using 4423 models on the same hardware (on average 33 seconds per sequence). In general, the feature extraction process and the calculation of the matrix product can easily be parallelized, which further reduces the required running time. Therefore, our approach provides a suitable means for target set reduction on huge sequence collections, which are routinely analyzed in metagenomics [1,27].

Table 2. Runtime comparison of different methods for sequence scoring. The first column denotes the method, whereby "Spectrum" refers to the k-mer Spectrum and "ODH" refers to Oligomer Distance Histograms using $k = 1$. Column 2 denotes the dimensionality of the feature space associated with a particular method. Columns 3 and 4 show the runtime in seconds for feature extraction and matrix multiplication during the scoring process for all 147003 sequences (see also section "Machine Learning Approach"). Columns 5 and 6 denote the total runtime in seconds for scoring and the average runtime in milliseconds for scoring of a single sequence, respectively.

Method	d	$t_{featExt}$ (s)	$t_{matMult}$ (s)	t_{total} (s)	t_{avg} (ms)
Spectrum ($k = 1$)	20	575	10	585	4.0
Spectrum ($k = 2$)	400	622	80	702	4.8
Spectrum ($k = 3$)	8000	643	1500	2143	14.6
ODH ($D_{max} = 10$)	4020	1027	800	1827	12.4
ODH ($D_{max} = 20$)	8020	1082	1525	2607	17.7
ODH ($D_{max} = 30$)	12020	1130	2250	3380	23.0

Table 1 shows that mean and median coverage of a particular method differ significantly. This results from the shape of the minimal set size distribution for the test examples. Figure 1 shows the dependency of the corresponding fraction of sequences on a given minimal set size in terms of curve plots for different methods. It is clearly visible that for the ODH method a large number of test examples show a low minimal set size, while only few examples show a very high minimal set size. If one is willing to accept a significant loss of sensitivity, e.g. for a coarse estimation of functional profiles in metagenomics, the target set reduction allows a considerable speed-up.

We also analyzed whether the performance of our approach critically depends on the family size or on the number of associated classes of an example. Therefore, we calculated the correlation coefficient of the minimal set size and the number of true categories of a particular example using the results from the ODH method with maximum distance $D_{max} = 30$. The low correlation coefficient of -0.0031 indicates that examples with many functional categories do not lead to higher minimal set sizes than examples with few categories. Since only multi-domain proteins have more than one assigned category, this result indicates that our approach is suitable for ranking of single-domain and multi-domain proteins, as well. Furthermore, we computed the correlation coefficients between ROC/ROC50 values and the family size. The low correlation values (ROC: 0.1288, ROC50: 0.046) indicate that the performance does not critically depend on family size. Note that we measured the ROC/ROC50 performance only for families with at least 10 positive test examples in each test fold.

In this work, we limited the evaluation setup to Pfam families with at least 10 different seed sequences. In practice, our approach is also suitable to be used with the complete Pfam database for learning of 9318 family-specific discriminants. As a draft study, we evaluated Pfam families with 2 to 9 seed sequences using 2-fold cross-validation. The corresponding data set consists of 4750 families with 22944 sequences in total. Here, the ODH method using $D_{max} = 30$ achieves a

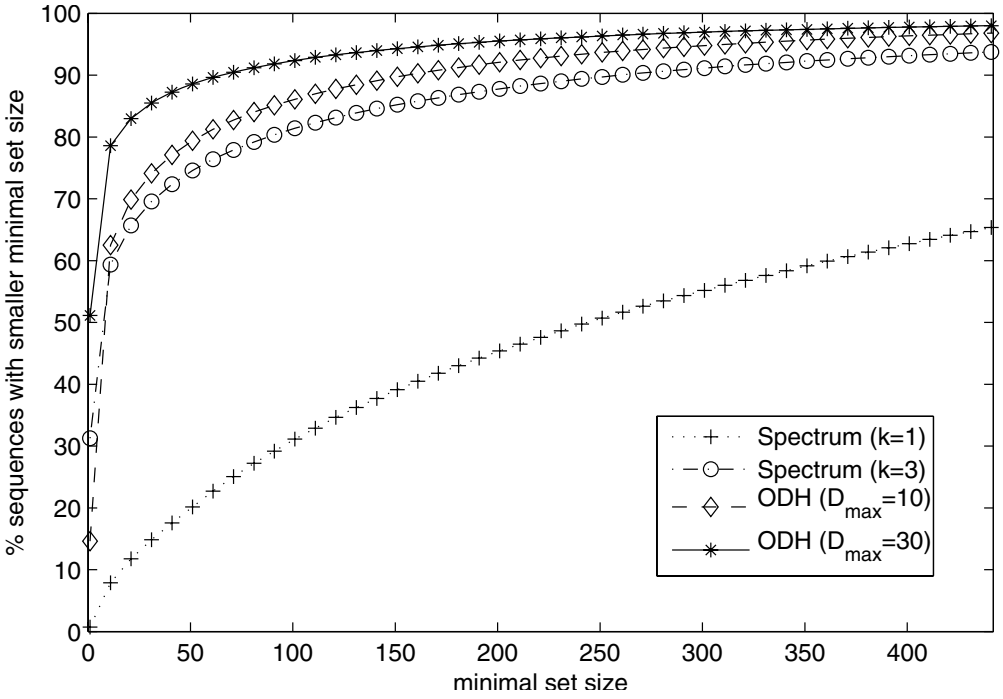

Fig. 1. Coverage curves for different methods. The coverage curve of a particular method plots the relative number of examples that have a minimal set size below a given threshold. Only minimal set size values up to 442 (10% of the classes) are shown.

mean coverage of 32.4 and a one-error of 0.26. The performance of the Spectrum method for $k = 3$ is inferior, resulting in a mean coverage of 94.6 and a one-error of 0.54. A direct comparison to the results in table 1 is not possible, because the average number of true categories over all example sequences is much lower for the small families (1.07 instead of 1.6). Furthermore, the results of the two cross-validation folds show large differences. One possible reason is that for the smallest families only one example sequence is used for training and testing. In practice, sequences from the Pfam full alignments [5] could be used to extend the training set of small families.

In contrast to PHMMs, feature-based approaches allow the interpretation of discriminative sequence features that have been learned from the data. Therefore, these methods are not only useful for target set reduction, but they could also be used for analysis of biologically meaningful properties of the sequences. In [15] we showed how discriminative features of the ODH method could be interpreted in a remote homology detection context.

5 Conclusion

In this work, we presented an approach for large-scale ranking of protein sequences for function prediction. Our method is based on explicit representation

of sequences in feature spaces that allow fast scoring and efficient training of discriminants. We developed a setup for evaluation of multi-label machine learning techniques, which is based on a large part of the Pfam database. We showed that our approach can be used for fast target set reduction in terms of a ranking of functional categories. The reduced target set can be analyzed with highly accurate but computationally more expensive methods. As a main result, we showed that the ranking performance critically depends on the choice of a suitable feature space for representation of protein sequences.

Although our approach worked well for most of the Pfam families and sequences, for some categories the performance was very low. In this work, we showed that the detection performance does not critically depend on family size. First results also indicate that functional categories can be predicted for multi-domain proteins and single-domain proteins with similar accuracy. A detailled analysis of the reasons for the low performance in a few cases will be part of future work. In order to use our approach as a stand-alone protein classification method, our ranking has to be extended by a so-called "set size predictor" [21], which estimates the number of relevant functional categories from the discriminant scores. For this purpose, we are currently investigating several set size prediction methods.

Acknowledgments. This work was partially supported by the Federal Ministry of Research and Education project "MediGRID" (BMBF 01AK803G).

References

1. Yooseph, S., et al.: The Sorcerer II Global Ocean Sampling expedition: expanding the universe of protein families. PLoS Biol. 5, 16 (2007)
2. Friedberg, I.: Automated protein function prediction–the genomic challenge. Brief. Bioinformatics 7, 225–242 (2006)
3. Pandey, G., Kumar, V., Steinbach, M.: Computational approaches for protein function prediction. Technical Report TR 06-028, Department of Computer Science and Engineering, University of Minnesota, Twin Cities (2006)
4. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. J. Mol. Biol. 215(3), 403–410 (1990)
5. Finn, R., et al.: Pfam: clans, web tools and services. Nucleic Acids Res. 34, D247–251 (2006)
6. Eddy, S.R.: Profile hidden Markov models. Bioinformatics 14(9), 755–763 (1998)
7. Liao, L., Noble, W.S.: Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. J. Comput. Biol. 10(6), 857–868 (2003)
8. Walters, J.P., Meng, X., Chaudhary, V., Oliver, T.F., Yeow, L.Y., Schmidt, B., Nathan, D., Landman, J.I.: MPI-HMMER-Boost: Distributed FPGA Acceleration. VLSI Signal Processing 48(3), 223–238 (2007)
9. Ong, S., Lin, H., Chen, Y., Li, Z., Cao, Z.: Efficacy of different protein descriptors in predicting protein functional families. BMC Bioinformatics 8, 300 (2007)
10. Strope, P., Moriyama, E.: Simple alignment-free methods for protein classification: a case study from G-protein-coupled receptors. Genomics 89, 602–612 (2007)

11. Han, L., Cui, J., Lin, H., Ji, Z., Cao, Z., Li, Y., Chen, Y.: Recent progresses in the application of machine learning approach for predicting protein functional class independent of sequence similarity. Proteomics 6, 4023–4037 (2006)
12. Leslie, C., Eskin, E., Noble, W.S.: The spectrum kernel: a string kernel for SVM protein classification. In: Pac. Symp. Biocomput., pp. 564–575 (2002)
13. Ben-Hur, A., Brutlag, D.: Remote homology detection: a motif based approach. Bioinformatics 19 (suppl. 1), 26–33 (2003)
14. Leslie, C.S., Eskin, E., Cohen, A., Weston, J., Noble, W.S.: Mismatch string kernels for discriminative protein classification. Bioinformatics 20(4), 467–476 (2004)
15. Lingner, T., Meinicke, P.: Remote homology detection based on oligomer distances. Bioinformatics 22(18), 2224–2231 (2006)
16. Saigo, H., Vert, J.P., Ueda, N., Akutsu, T.: Protein homology detection using string alignment kernels. Bioinformatics 20(11), 1682–1689 (2004)
17. Rangwala, H., Karypis, G.: Profile-based direct kernels for remote homology detection and fold recognition. Bioinformatics 21(23), 4239–4247 (2005)
18. Rifkin, R., Klautau, A.: In Defense of One-Vs-All Classification. Journal of Machine Learning Research 5, 101–141 (2004)
19. Jensen, L.J., Gupta, R., Staerfeldt, H., Brunak, S.: Prediction of human protein function according to Gene Ontology categories. Bioinformatics 19, 635–642 (2003)
20. Schapire, R., Singer, Y.: Boostexter: A system for multiclass multi-label text categorization (1998)
21. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) NIPS, pp. 681–687. MIT Press, Cambridge (2001)
22. Zhang, M.L., Zhou, Z.H.: A k-nearest neighbor based algorithm for multi-label classification. The IEEE Computational Intelligence Society 2, 718–721 (2005)
23. Lee, K., Kim, D., Na, D., Lee, K., Lee, D.: PLPD: reliable protein localization prediction from imbalanced and overlapped datasets. Nucleic Acids Res. 34, 4655–4666 (2006)
24. Diplaris, S., Tsoumakas, G., Mitkas, P.A., Vlahavas, I.P.: Protein classification with multiple algorithms. In: Bozanis, P., Houstis, E.N. (eds.) PCI 2005. LNCS, vol. 3746, pp. 448–456. Springer, Heidelberg (2005)
25. Rifkin, R., Yeo, G., Poggio, T.: Regularized Least Squares Classification. In: Advances in Learning Theory: Methods, Model and Applications NATO Science Series III: Computer and Systems Sciences, vol. 190, pp. 131–153. IOS Press, Amsterdam (2003)
26. Cohen, G., Hilario, M., Sax, H., Hugonnet, S., Geissbuhler, A.: Learning from imbalanced data in surveillance of nosocomial infection. Artif. Intell. Med., 7–18 (2006)
27. Hoff, K., Tech, M., Lingner, T., Daniel, R., Morgenstern, B., Meinicke, P.: Gene prediction in metagenomic fragments: a large scale machine learning approach. BMC Bioinformatics 9, 217 (2008)

Lebenslauf

Persönliche Daten

Name:	Thomas Lingner
Geburtsdatum:	30.04.1977
Geburtsort:	Wolgast
Staatsangehörigkeit:	deutsch

Schulische Ausbildung

1984 – 1986	Grundschule Stendal
1986 – 1992	Comenius-Gymnasium Stendal
1992 – 1996	Gymnasium Leopoldinum Detmold Abschluss: Abitur

Wehrdienst

07/1996 – 06/1997	Luftlandemörserkompanie 270, Wildeshausen

Berufliche Ausbildung

07/1997 – 06/1999	Klinikum Lippe-Detmold Ausbildung zum Bürokaufmann IHK-Abschluss: Kaufmannsgehilfe

Studium

10/1999 – 09/2005	Naturwissenschaftliche Informatik, Universität Bielefeld Schwerpunkt: Mustererkennung Diplomarbeit: “Neue Ansätze zum maschinellen Lernen von Alignments” Abschluss: Diplom
seit 10/2005	Doktorand in der Abteilung Bioinformatik, Universität Göttingen Promotionsthema: “Alignmentfreie Analyse von Proteinsequenzen mit Verfahren des maschinellen Lernens”

Berufliche und studentische Tätigkeiten

04/2000 – 03/2002	Netzwerk- und Systemadministration bei Netzwerk Lippe GmbH, Detmold
01/2002 – 09/2005	Systemadministration und Redaktionsassistenz bei AKP-Redaktion (Alternative Kommunalpolitik), Bielefeld
04/2002 – 09/2005	studentische Hilfskraft in der Abteilung Neuroinformatik, Universität Bielefeld