

Användandet av olika jämförelsemetoder för att upptäcka proteiner som har liknande tredimensionell struktur

Exjobbssrapport i Biomedicinsk teknik, 2D1025, Nada, KTH av Magnus Nilsson

Sammanfattning

Samtidigt som antalet sekvenserade genom växer, ökar också antalet proteiner med okänd tredimensionell struktur och funktion. Att experimentellt bestämma ett proteins struktur är svårt och tidskrävande, därför är datoriserade metoder högst önskvärda. Existerar ett annat protein med liknande sekvens, kan den okända proteinsekvensen jämföras mot ett bibliotek av kända strukturer. Det blir dock svårare att förutsäga strukturen då inga kända proteiner på sekvensnivå existerar, vilket var fallet i detta arbete. Proteindata erhöles från olika typer av parvisa jämförelser. Klassificeringen, lika eller inte lika, utfördes av en supportvektormaskin. Övervakad inlärning tillämpades, eftersom svaret redan var känt från en strukturdatabas (SCOP, A. G. Murzin *et al.*, 1995). Detta arbete visar att om parametrar, såsom indata och maskininställningar, optimeras kan supportvektormaskiner mycket väl vara ett bra val då det gäller att förutbestämma proteiners tredimensionella struktur.

Abstract

The use of different alignments to detect proteins that share the same fold

As the number of sequenced genomes grows, the number of proteins whose structure and function is unknown increases rapidly. To experimentally determine a protein structure is difficult and time-consuming, therefore computer methods for protein structure prediction are highly desirable. In general the approach in protein fold recognition is a comparison of protein sequences with unknown folds to a library of determined protein structures. The method developed in this work is based on pairwise sequence alignments between protein models, to generate input data to a Support Vector machine. The SV machine can perform binary classification, i.e. if the structures are equal or not. Supervised learning is used, since the answer is known from a protein structure database (SCOP, A. G. Murzin *et al.*, 1995). This work shows that if the input parameters and machine arguments are optimized, SV machines can perform well in protein fold recognition.

Förord

Denna rapport är en sammanställning från ett examensarbete i Biomedicinsk teknik, Nada, KTH i Stockholm. Arbetet är utfört vid Stockholm Bioinformatics Center (SBC), Stockholms universitet (SU).Handledare och examinator vid Nada, KTH, var professor Anders Lansner. Extern handledare vid SBC, SU, var docent Arne Elofsson.

Stockholm Bioinformatics Center

Bioinformatik är ett nyckelområde inom den framtida forskningen avseende molekylär vetenskap. Därför erhöLL i december 1999 Stockholms universitet (SU), Kungl tekniska högskolan (KTH) och Karolinska institutet (KI) totalt 40 miljoner kronor från Stiftelsen för strategisk forskning (SSF) för att under fem år bygga upp en nationell kärnverksamhet med fokusering på bioinformatik – Stockholm Bioinformatics Center (SBC).

Tack till

Emil Olovsson och Jesper Lundström (samtida exjobbare), samt Arne Elofsson och Anders Lansner.

Innehåll

1 Inledning.....	4
2 Bakgrund.....	5
2.1 Proteiner.....	5
2.1.1 Struktur.....	5
2.1.2 Funktion.....	5
2.2 Introduktion till bioinformatik.....	6
2.2.1 Resurser för proteininformation.....	7
2.2.2 Informationskällor för genom.....	8
2.2.3 Analys av DNA-sekvenser.....	8
2.2.4 Parvis sekvensjämförelse.....	9
2.2.5 Multipel sekvensjämförelse.....	10
2.3 Strukturigenkänning.....	12
2.3.1 Olika metoder.....	12
2.3.2 Evolutionär information.....	13
2.3.3 GenTHREADER.....	14
2.4 Utvärdering av energier.....	14
2.5 SCOP.....	16
2.6 Kort om supportvektormaskiner.....	17
2.6.1 Artificiella neuronnät och andra lärande system.....	17
2.6.2 Generella användningar.....	18
2.6.3 Idén med supportvektormaskiner.....	18
2.6.4 Lärande maskiner.....	19
2.6.5 Den linjära supportvektormaskinen.....	19
2.6.6 Större problem.....	20
2.6.7 Nackdelar.....	21
3 Metoder och material.....	22
3.1 SVM-light.....	22
3.1.1 Parametrar hos SVM-light.....	22
3.2 Tränings- och testdata.....	23
3.2.1 Mathews korrelationskoefficient.....	25
4 Resultat.....	26
4.1 Optimering.....	26
4.1.1 Utvärdering av olika dataparametrar.....	26
4.1.2 Utvärdering av olika inställningar hos SVM-light.....	27
4.2 Jakten på det bästa MC-värdet.....	29
4.3 Jämförelse med data genererade från andra metoder.....	30
5 Diskussion.....	33
5.1 Val av dataparametrar.....	33
5.2 Andra parametrar att variera.....	34
5.3 Slutsatser.....	35
6 Referenser.....	36

1 Inledning

Samtidigt som antalet sekvenserade genom växer, ökar också antalet proteiner med okänd tredimensionell struktur och funktion. Att experimentellt bestämma ett proteins struktur är tidskrävande, därför är datoriserade metoder högst önskvärda. Under de senaste åren har olika metoder för strukturigenkänning (eng. *fold recognition*) förbättrats avsevärt. Vanligtvis jämförs då en proteinsekvens med okänd struktur mot ett bibliotek av kända strukturer. Detta kan göras eftersom det endast finns ett begränsat antal naturliga strukturer.

Vid strukturigenkänning sker ofta en uppdelning mellan proteiner som har ett gemensamt evolutionärt ursprung och proteiner som inte har det, men har konvergerat under evolutionen och därför har en liknande struktur. Eftersom proteiner med en gemensam förfader ofta har hög sekvenslikhet är det mycket lättare att bestämma strukturen för ett okänt protein då strukturen hos en släkting är känd. Proteiner med hög sekvenslikhet har nämligen nästan alltid liknande struktur. Strukturen är dessutom mer konserverad än sekvensen, varför detta är högst befogat. Snarare än att bestämma hur en sekvens kommer vecka sig i tre dimensioner är metodiken att förutbestämma hur väl en speciell struktur kommer att passa sekvensen.

Detta arbete är fokuserat på den lite svårare delen, nämligen den där det inte existerar några nära kända släktingar. Stockholm Bioinformatics Center (SBC) har i tidigare studier visat att det inte finns någon metod som generellt fungerar bättre än någon annan då det gäller olika proteinjämförelser. Här användes data från flera olika metoder, bland annat olika energivärden, längden på jämförelsen och ett värde från den parvisa jämförelsen. Syftet är att en supportvektormaskin (SV-maskin) skall klassificera olika jämförelsedata för att på bästa möjliga sätt förutsäga om de jämförda proteinerna är strukturellt lika eller inte. Svaret är känt, eftersom klassificeringen redan finns i en databas [17]. Problemet ligger i att optimera indata till SV-maskinen samt att ställa in parametrarna hos denna så bra som möjligt.

Rapporten behandlar först bakgrunden till arbetet, såsom proteiner, bioinformatik, strukturigenkänning och SV-maskiner. Därefter följer en presentation av de olika metoder och det material som användes. Vidare presenteras resultaten, samt en diskussion om vad som skulle kunna göras för att eventuellt förbättra metoden ytterligare. Arbetet har hämtat inspiration hos en metod för automatisk strukturigenkänning kallad GenTHREADER [1].

2 Bakgrund

2.1 Proteiner

I alla levande organismer finns proteiner. Dessa bildas av 20 olika aminosyror, vilka kallas de naturligt förekommande aminosyrorerna. Aminosyror är karboxylsyror med den funktionella gruppen amin. Vad som skiljer dessa aminosyror åt är de sidogrupper som sitter på kolatomen i mitten, α -kolatomen. Sidogruppen kan vara allt från en enstaka väteatom till långa kedjor eller ringar. Några aminosyror innehåller ytterligare en aminogrupp eller en karboxylgrupp i sidogruppen, varför de då kan vara neutrala, sura eller basiska. Hos peptider har aminosyrorerna förenats med peptidbindningar. Vid 50 eller fler aminosyror kallas peptiden för protein.

2.1.1 Struktur

Proteinstrukturer kan definieras på flera olika nivåer. En aminosyrasekvens kallas för en primär struktur. Med den sekundära strukturen avses den konformation peptidkedjan har beroende på de intramolekylära krafterna, främst vätebindningar mellan olika NH- och C=O-grupper. Två vanliga konformationer är α -strukturen, där peptidkedjan bildar en spiral, och β -strukturen, där kedjan bildar veckade flak. Den tertiära strukturen beskriver hur hela proteinet veckar sig i proteinets naturliga tillstånd, samt hur eventuellt prostetiska grupper (en metalljon eller atomgrupp adderad till proteinet) är orienterade. Förutom aminosyrasekvensen är strukturen mycket beroende av pH och temperatur. Om proteinet består av flera polypeptidkedjor som samverkar intramolekylärt, tillkommer även kvartär struktur.

2.1.2 Funktion

Proteiner har många olika funktioner. De kan bilda strukturelement och stödjande vävnad i form av till exempel hår, hud och bindväv. En annan typ är globulära proteiner, även kallade biologiskt aktiva proteiner, vilka bildar kompakta nystan, så kallade miceller. I varje protein av denna typ har peptidkedjan en alldeles bestämd veckning. Varje aminosyraenhet ligger på en viss plats i nystanet. Det inre, som är helt kompakt och alltså utan hålrum, består huvudsakligen av aminosyraenheter med hydrofoba sidokedjor. På nystanets utsida finns främst de aminosyror som har hydrofila sidokedjor. Dessa små nystan stabiliseras av den hydrofoba effekten, som är den viktigaste sammanhållande kraften hos de globulära proteinerna. På grund av denna micellbildning, med de hydrofoba delarna inåt, kan

proteinmolekylerna fungera i cellens vattenmiljö trots att de innehåller så många hydrofoba grupper. Också hos de globulära proteinerna består peptidkedjan i större eller mindre utsträckning av α - eller β -struktur. Det finns även loopar, vilka varken är α - eller β -struktur. De globulära proteinernas funktion bestäms till stor del av deras tredimensionella struktur, tertiärstrukturen. Biologiskt aktiva proteiner är till exempel transportproteiner, antikroppar, hormoner och enzymer.

2.2 Introduktion till bioinformatik

Termen bioinformatik används för att omfatta nästan all datoranvändning inom biologisk vetenskap, men myntades i mitten på 1980-talet för analys av biologisk sekvensdata hos olika proteiner. Med sekvens avses de aminosyraenheter som proteinet är uppbyggt av. Kvantiteten känd sekvensdata är mångfaldigt större än känd proteinstrukturdata, det vill säga tertiärstrukturen eller den tredimensionella strukturen. Vidare bidrar olika genomprojekt till att sekvensdatabaserna fördubblas i storlek varje år, samtidigt som mängden av kända tredimensionella strukturer inte alls växer lika explosionartat.

En stor utmaning inom bioinformatiken är att analysera floran av sekvensdata för att kunna förstå den information som ges i form av proteinstruktur, funktion och evolution. Två vanliga tillvägagångssätt inom bioinformatiken är mönsterigenkänning (eng. *pattern recognition*) och strukturigenkänning (eng. *fold recognition*), men det finns många fler metoder. Stora framsteg har gjorts med metoder för mönsterigenkänning eftersom det finns referensdatabaser innehållande sekvensmönster och veckningsmallar mot vilka det protein som skall bli undersökt kan jämföras.

Otillräcklig förståelse av proteinveckningsproblemet, det vill säga hur den linjära sekvensen bestämmer den slutliga tredimensionella veckningen, utgör ett stort hinder vid försök att förutsäga den tredimensionella strukturen hos proteinet direkt från dess sekvens. Om endast sekvensen var känd, och därur den tredimensionella strukturen kunde bestämmas, skulle detta vara en av de största bedrifterna någonsin inom naturvetenskapen [2].

Homologi är ett centralt koncept inom bioinformatiken, sekvenser sägs vara homologa om de har en gemensam förfader. Det fundamentala inom sekvensanalys är upptäckten av homologa släktskap med hjälp av databassökningar. Inom homologin delas proteinerna upp i undergrupper såsom ortologa, proteinerna har samma funktion inom olika arter, eller paraloga, proteinerna har olika funktioner inom samma art. Den tredimensionella strukturen

är vanligtvis bättre bevarad under evolutionen än den en- dimensionella strukturen, den primära, varför avlägset homologa proteiner kan ha en liknande, eller väldigt lika, tredimensionell struktur.

Termen analogi används i sammanhang med liknande proteinveckningar som inte delar någon synlig sekvenslikhet eller proteiner som delar grupper av aminosyraenheter med samma tredimensionella geometri, men annars inte har någon sekvens- eller strukturlikhet. Sådana förhållanden är troligtvis ett resultat från en evolutionär konvergensprocess. Sökningar för att finna tre- dimensionella likheter kan fortgå med minskad säkerhet mot *the Twilight Zone* (mindre än 20% sekvenslikhet), där resultaten upphör att vara statistiskt signifikanta. Om det är möjligt skall olika analysmetoder för proteinstrukturbestämning användas, och resultatet skall tas i beaktande tillsammans med övrig tillgänglig biologisk information [3].

2.2.1 Resurser för proteininformation

Databaser används för att lagra de väldiga mängderna information från olika genomprojekt. Det finns flera olika typer av databaser, men för vanlig analys av sekvensdata från protein är primära, sekundära och sammansatta databaser de viktigaste. Primära databaser innehåller sekvensdata från nukleinsyror eller proteiner. SWISS-PROT är en vanlig proteinsekvensdatabas. Sammansatta databaser använder olika primära källor och olika kriterium då de fusioneras. Sekundära databaser innehåller mönsterdata, vilka är speciella diagnostiska signaturer för olika proteinfamiljer. Dessa signaturer (mönster) visar på de mest bevarade egenskaperna från multipla sekvensjämförelser (många proteinsekvenser jämförs mot varandra) vilka ofta är avgörande för strukturen eller funktionen av ett protein.

Olika sekvensanalysmetoder har gett upphov till olika mönster- databaser: för utforskning av enstaka konserverade amino- syrasträngar (eng. *regular expressions*), flera återkommande aminosyrasträngar (eng. *fingerprints*) eller fullständiga domänjämförelser (eng. *hidden Markov models*). Det finns många olika sekundära databaser, men till och med totalsumman av mönsterdata från dem är för liten för att strukturbestämma de uppskattade 1000–10000 proteinfamiljerna. Två manuellt kom- menterade sekundära databaser är PROSITE och PRINTS. De sekundära databaserna har tillsammans bildat en förenad databas av proteinfamiljer, känd som InterPro. Förhoppningen är att InterPro kommer att göra sekvensanalys mer rättfram i framtiden [3].

2.2.2 Informationskällor för genom

De huvudsakliga databaserna för nukleinsyrasekvenser är GenBank, EMBL och DDBJ. Dessa samlar ihop en stor del av den totala mängden sekvensdata rapporterad från hela världen, och utväxlar nya och uppdaterade sekvenser dagligen. GenBank är indelat i mindre, diskreta divisioner. Detta underlättar snabba, specifika sökningar genom restriktiva förfrågningar till speciella underavdelningar av GenBank. Under 1992–1997 växte innehållet av EST-data, små delsekvenser, och STS-data, korta DNA-sekvenser, i GenBank tiofalt. Dock var den sekvensinformation som bidrog genom sådan partiell data fortfarande mindre än den av högre sekvenskvalitet.

Förutom de allsidiga databaserna för DNA-sekvenser finns det ett antal mer specialiserade genomiska resurser. Dessa så kallade specialistdatabaser (eng. *boutique databases*) fokuserar på art-specifika genom och på speciella tekniker för sekvensering. Tillgången av genomdata och verktyg för att behandla denna data på Internet är enorm och har haft en otrolig genomslagskraft på möjligheten för vetenskapsmän att presentera och sprida undersökningsresultat [3].

2.2.3 Analys av DNA-sekvenser

Sekvensjämförelser är mer känsliga på proteinnivå än på nukleinsyrenivå därför att den genetiska koden då är reducerad till en unik mängd aminosyror, vilket innebär att information som relaterar direkt till evolutionära processer går förlorad. DNA-sekvensdatabaser inkluderar genomisk sekvensdata och innehåller därför en blandning av datatyper som inte kan behandlas lika (bland annat introner, exoner, mRNA och cDNA). Detta påverkar sättet på hur sökningen skall utföras.

Närvaron av introner och exoner i eukaryota gener kan ge uppkomst till genprodukter av olika längd, alla exoner kanske inte används vid transkriptionen. Resulterande proteiner är kända som skarvade varianter (eng. *spliced forms*). En del tillgänglig DNA-data kommer från *Expressed Sequence Tags*, ESTs, vilka är partiella sekvenser. EST-produktion är mycket automatiserad och resultaten är ofta förenade med mångtydiga och missade baser. Detta ger upphov till svårigheter vid sekvenstolkningen. Olika tillvägagångssätt för att etablera EST-bibliotek har utvecklats för akademiska och kommersiella syften [3].

2.2.4 Parvis sekvensjämförelse

Databasförfrågningar kan ske som textförfrågningar eller sökningar avseende sekvensjämförelse. För att identifiera evolutionära släktskap mellan en nyligen bestämd sekvens och en känd genfamilj måste utsträckningen av delad likhet bedömas. Det lättaste sättet att jämföra två sekvenser är att jämföra dem genom att infoga gap, hålrum, för att få dem i ett register. Sedan räknas de matchande teckenpositionerna och en enkel skalär jämförelsepoäng (eng. *output score*) utdelas. Poäng vid datajämförelse är mest exakt för långa, olika sekvenser med disparata längder. Straffpoäng ges för att minimera antalet gap och längden på gap. Matriser används för att ge poäng både för identiska och liknande aminosyraenheter. Identitetsmatriser är sparsamma, mest nollor, och har därför liten diagnostisk genomslagskraft. Likhetsmatriser viktar icke-identiska matchningar som stämmer överens över stora evolutionära avstånd. Sådana matriser är brusiga därför att de förstärker både slumpmatchningar och svaga signaler. Att skilja biologiska lågpoängssignaler från höga poängsbrus är en viktig utmaning inom sekvensanalys [3].

Poäng i *the Dayhoff Mutation Data Matrix* är baserad på idén om poängaccepterad mutation (eng. PAM – *the Point Accepted Mutation*). Ett evolutionärt avstånd på 250 PAMs ger likhetspoäng ekvivalent tills 20% av matchningarna återstår mellan två sekvenser (*the Twilight Zone*, ingen statistisk signifikans kvarstår). Därför är inställningen 250 PAMs ofta använd som grundmatris i jämförelseprogram. Om ett program jämför två sekvenser, är detta inget bevis på att släktskap existerar dem emellan. Statistiska värden används för att indikera nivån av tillförlitlighet som skall tas i beaktande vid jämförelsen [3].

En enkel metod för att jämföra två sekvenser är DotPlot, se *Figur 1*. DotPlot visar en graf i vilken sekvenserna ligger på x- respektive y-axeln och kors/prickar ritas vid alla positioner där identiska rester är observerade. För identiska sekvenser leder detta till att man erhåller en obruten diagonal linje samtidigt som liknande sekvenser ger upphov till en bruten linje. Jämförelsemodeller kallas modeller som reflekterar olika biologiska perspektiv. En jämförelsemodell är därför inte mer rätt eller fel än en annan [3].

av dem skall därför bli betraktad som representativ för korrekthet eller för någon speciell standard [3].

En multipel jämförelse kan definieras som en tvådimensionell tabell i vilken varje rad representerar individuella sekvenser och kolumnerna är aminosyraenheternas position. En enhetsposition inom en ojämförd sekvens är benämnd den absoluta positionen medan den jämförda aminosyraenhetens position kallas relativ position. Tiden det tar att beräkna en jämförelse ökar exponentiellt med antalet sekvenser som skall jämföras. Resultatet av automatiska jämförelseprogram kräver dock nästan alltid manuell finputsning och därför har jämförelseeditorer blivit fundamentala verktyg inom bioinformatiken. Samtida multipla jämförelsemetoder jämför alla sekvenser inom en grupp på en gång och de är därför väldigt tidskrävande. De arbetar bäst på små grupper av sorterade sekvenser [3].

Framåtskridande multipla jämförelsemetoder jämför sekvenser i par, följande grenarna i ett familjetråd, ett så kallat fylogenetiskt träd. De mest lika jämförs först och mer långväga relaterade sekvenser adderas sedan. Genom att utforska troliga evolutionära släktskap kan sådana metoder hantera mer realistiska dataserier på ett tids- och kostnadsmässigt mer effektivt sätt. Det finns mångtaliga databaser för jämförelser på Internet.

Jämförelser producerade av rent automatiserade, speciellt iterativa, metoder skall iakttagas med försiktighet, speciellt i fall då sekvenslikheten är låg [3]. De resulterar ofta i övernitiska infogningar av gap och kan producera feljämförelser [3]. Olika algoritmer har utvecklats för att söka i primära sekvensdatabaser användandes jämförelsebaserade datastrukturer. En ny hybridtillämpning, innefattande element av både parvisa och multipla jämförelsemetoder är *Position-Specific Iterated BLAST*, eller PSI-BLAST.

2.3 Strukturigenkänning

Samtidigt som antalet sekvenserade genom växer, ökar också antalet proteiner med okänd tredimensionell struktur och funktion. Likheter och släktskap mellan proteiner kan variera inom ett brett spektrum, från i stort sett identiska sekvenser till proteiner som endast delar en liknande tredimensionell struktur. Genom att bestämma om sekvenser är relaterade till kända proteiner, kan information erhållas om deras strukturella, funktionella och evolutionära egenskaper. Eftersom struktur- och funktionsbestämmelse på intet sätt är trivialt, inte ens för ett enskilda protein, så är det bästa sättet för att förstå dessa sekvenser att relatera dem till andra proteiner med kända egenskaper. Detta kan ske genom sökningar i databaser. Att förbättra dessa algoritmer är en av de grundläggande utmaningarna inom bioinformatiken idag. Vidare används olika tillvägagångssätt då det gäller att finna de rätta algoritmerna för att hitta likheter – en metod som är utmärkt för att finna sekvenslikhet kanske inte är speciellt bra på att hitta strukturella likheter, och vice versa [6].

Vid strukturigenkänning av proteiner (eng. *protein fold recognition*) är målet att hitta vilken struktur den nya sekvensen troligtvis kommer att anta. Det gäller därför att hitta likheter mellan tre-dimensionella strukturer, även om det inte finns någon signifikant sekvenslikhet. Snarare än att bestämma hur en sekvens kommer vecka sig i tre dimensioner är metodiken att förutbestämma hur väl en speciell struktur kommer att passa sekvensen [6].

2.3.1 Olika metoder

Det finns många proteiner med liknande struktur där ingen uppenbar homologt existerar. Metoder som är utvecklade för att identifiera detta strukturella släktskap kallas ofta för strukturigenkänningsmetoder (eng. *fold recognition* eller *threading*). Grovt kan de delas in i två kategorier: sekvensbaserade metoder och strukturella metoder. De strukturbaserade metoderna skiljer sig från de andra eftersom de inte direkt använder någon sekvensinformation för att förutsäga om två proteiner delar samma struktur eller inte. Istället används en energifunktion (se *Utvärdering av energier*) som beskriver hur väl en okänd sekvens matchar den kända strukturen. Energifunktionen är ofta erhållen från en databas av kända proteinstrukturer och kan till exempel beskriva omgivningen runt varje enskild aminosyraenhet eller möjligheten att finna två aminosyraenheter på ett visst avstånd från varandra. Vissa metoder är baserade på sekvensinformation, andra på multipla jämförelser, en del på strukturell information eller kombinationer av sekvens- och strukturinformation. En av de bästa metoderna är att använda gömda

Markovmodeller (eng. *hidden Markov models*), som använder multipel jämförelse för att bygga en sekvensprofil [7].

Ett annat tillvägagångssätt är påträdnings (eng. *threading*), där sekvenser av okänd struktur läggs in direkt på *backbone*-koordinaterna, det vill säga proteinet utan sidokedjor, av en känd proteinstruktur [8]. Varje modell utvärderas sedan med hjälp av en energifunktion. Två av orsakerna till att *threading* visar så goda resultat tros vara att de hydrofoba interaktionerna i proteinets inre tas med i beräkningarna samt att den sekundära strukturen identifieras genom metoden [9]. Proteiner som har liknande veckning har per definition också liknande sekundär struktur [10]. Eftersom den sekundära strukturen kan bestämmas direkt från proteinsekvensen med en noggrannhet på 70% idag [11], så inkorporerar många strukturigenkänningsmetoder idag sekundära strukturförutsägelser för att förbättra resultatet från igenkänningsmetoden.

2.3.2 Evolutionär information

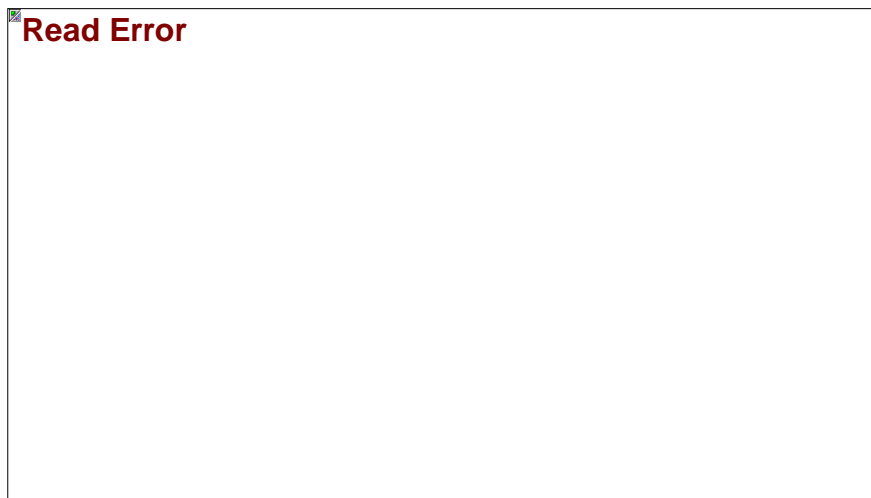
Även i en grupp av proteiner med ett gemensamt evolutionärt ursprung så kan ett par av sekvenser skilja sig markant i sammansättningen. I många år har det antagits att inkluderandet av evolutionär information i en multipel sekvensjämförelse hjälper till då det gäller att upptäcka långväga släktskap. Det är dock först ganska nyligen som det bevisats genom stora och omfattande undersökningar [10]. I dessa studier visades att inkluderandet av evolutionär information gör att tre gånger så många avlägsna släktingar upptäcks (vid samma antal falskt positiva jämförelser).

Det finns flera, ganska olika, sätt att inkludera evolutionär information. En möjlighet är att starta från en familj som redan är jämförd och sedan leta efter flera medlemmar till samma familj. Ett annat sätt är använda ett iterativt tillvägagångssätt med början från en enda sekvens, söka efter alla sekvenser som är relaterade till denna och sedan använda multipel sekvensjämförelse. Från denna jämförelse tar en ny iteration vid och denna procedur fortlöper tills konvergens uppnås. Ett tredje sätt är att benämna två proteiner som relaterade om de hittas av en sökalgoritm direkt eller genom ett tredje protein. Alla de nämnda metoderna har olika fördelar. Den direkta sökmetoden är snabbast och den iterativa är långsammast [10].

2.3.3 GenTHREADER

En metod för att kunna förutsäga den tredimensionella strukturen på automatisk väg är David T. Jones metod GenTHREADER, vilken är både snabb och pålitlig enligt upphovsmannen. Att bestämma strukturen hos ett enkelt proteom (alla gener hos en prokaryot) går på en dag [1].

Metoden använder sig av tre steg: först parvis jämförelse, sedan beräknas två olika energitermer och till sist utvärderas varje modell i ett artificiellt neuronnet. Sex parametrar används som indata till nätverket. Dessa är: en jämförelsepoäng, jämförelselängden, längden på de två jämförda sekvenserna, en löslighetsenergi och en parenergi. Energierna beräknas från modeller av det aktuella proteinet. Samtliga inparametrar ges som numeriska tal. Se *Figur 2*.



Figur 2. GenTHREADERs arkitektur. Bild från [1].

2.4 Utvärdering av energier

När ett okänd sekvens jämförs mot ett protein med känd struktur ger jämförelsen en grov bedömning av strukturen hos det okända proteinet. Om denna uppskattning av strukturen används för att utveckla en enkel modell av proteinet kan infogningen av den föreslagna strukturen beräknas. Ett vanligt sätt att göra en sådan bedömning är att beräkna kunskapsbaserade energier [12], [13] & [14]. Koordinaterna för den kända proteinstrukturen kombineras med jämförelsen och överförs till den okända sekvensen. På detta

sätt kan ett energivärde beräknas och den föreslagna modellen kan jämföras mot andra modeller.

Många olika modeller har använts för denna typ av beräkningar, det är inte alls uppenbart hur den idealiska energifunktionen skall se ut. För att en funktion skall vara framgångsrik måste den kunna upptäcka skillnader mellan inkorrekta och korrekta delstrukturer. Det är på intet sätt säkert att en mer komplex funktion fungerar bättre än en enkel. En enkel potential som ger en grov uppskattning av verkligheten kan gott och väl fungera tillfredsställande. Energi-funktionen som använts här är utvecklad av Park & Levitt [15].

Funktionen är en avståndsberoende kontaktpotential, liknande van der Waals energifunktion. Strukturen representeras på ett förenklat sätt, där aminosyraenheterna består av två interaktionscentra, dessa två centra är α -kolatomen och partikel som representerar hela sidokedjan. Alla α -katomer betraktas som energetiskt ekvivalenta, sidokedjorna är däremot distinkta. Energin för en konformation är beräknad enligt *Ekvation 1*.

$$E = \sum_{i=1}^N \sum_{j=i+4}^N \frac{A_{ij}}{r_{ij}^8} - \frac{B_{ij}}{r_{ij}^4} + \sum_{i=1}^N \sum_{j=i+4}^N \frac{A_{\alpha\alpha}}{r_{\alpha_i\alpha_j}^8} - \frac{B_{\alpha\alpha}}{r_{\alpha_i\alpha_j}^4} + \sum_{i=1}^N \sum_{j=1}^{i-3} \frac{A_{i\alpha}}{r_{i\alpha_j}^8} - \frac{B_{i\alpha}}{r_{i\alpha_j}^4} + \sum_{i=1}^N \sum_{j=1}^{i+3} \frac{A_{i\alpha}}{r_{i\alpha_j}^8} - \frac{B_{i\alpha}}{r_{i\alpha_j}^4}$$

där

$$A_{ij} = -\epsilon_{ij} (R_{ij}^a)^8 \qquad B_{ij} = -2 \epsilon_{ij} (R_{ij}^a)^4$$

Ekvation 1. Energi ekvationen som beskriver interaktionerna mellan endast α -katomerna, endast sidokedjorna samt mellan både α -katomerna och sidokedjorna.

Kontaktenergierna är representerade av α_{ij} -värdena, och kontaktavstånden av R_{ij} -värdena. Dessa finns tabellerade i [15] och härledda i [16]. Värdena r_{ij} beskriver de geometriska avstånden mellan två aminosyraenheter och är beräknade med x-, y- och z-koordinaterna. De fyra termerna i energi ekvationen beskriver interaktionerna mellan α -katomerna, sidokedjorna samt mellan α -katomerna och sidokedjorna. En av de viktigaste faktorerna inom strukturigenkänning är hydrofobiciteten i det inre av proteinet, med andra ord lösligheten hos proteinet. Detta refererar David T. Jones till som löslighetsenergin (eng. *solvation energi*), vilken är representerad i ekvationen av de två sista termerna, se [1].

2.5 SCOP

SCOP, *the Structural Classification of Proteins*, är en hiarkisk databas som innehåller samtliga proteiner från PDB, *the Protein DataBank* [17] & [18]. SCOP innehåller detaljerade beskrivningar av strukturella och evolutionära släktskap mellan proteiner där den tredimensionella strukturen har bestämts. Kvaliteten på databasen anses hög eftersom klassificeringen är gjord manuellt av SCOPs skapare, Alexei Murzin. Det manuella framtagandet av data har en fördel i och med att detta gör SCOP oberoende av någon speciell sekvens- eller strukturalgoritm.

Proteinerna är klassificerade på tre nivåer – nämligen familjenivå, superfamiljenivå och veckningsnivå (eng. *fold*). Proteiner på familjenivå har ett klart evolutionärt samband. De har antingen sekvensidentiteter på minst 30%, eller lägre sekvensidentiteter, men väldigt lika funktion och struktur. Superfamiljnivån innehåller familjer med ett troligt evolutionärt gemensamt ursprung. De har låg sekvensidentitet, men liknande strukturella och funktionella egenskaper. Proteiner placerade på veckningsnivå har stora strukturella likheter med i stort sett samma sekundärstruktur. Vidare har sekundärstrukturen samma arrangemang med liknande topologiska bindningar [17].

Flera studier under senare år har ändrat synen på vilka de bästa metoderna är för att upptäcka släktskap mellan proteiner. Dessa studier skiljer sig på vissa detaljer, men de har en sak gemensam, nämligen att de använder SCOP-klassificeringarna som en måttstock för att utvärdera prestandan från olika igenkänningsmetoder. Innehållet i SCOP är till stor del klassificerat och bestämt manuellt. Detta gör SCOP objektivt avseende någon speciell sekvens- eller strukturjämförelsealgoritm. SCOP är därför idealisk för att jämföra olika sådana metoder.

Jämförelser som fungerade som dataparametrar till SV-maskinen ansågs lika om de var lika på foldnivå, annars olika. Kvaliteten på resultatet kontrollerades med Mathews korrelationskoefficient, se *Mathews korrelationskoefficient*.

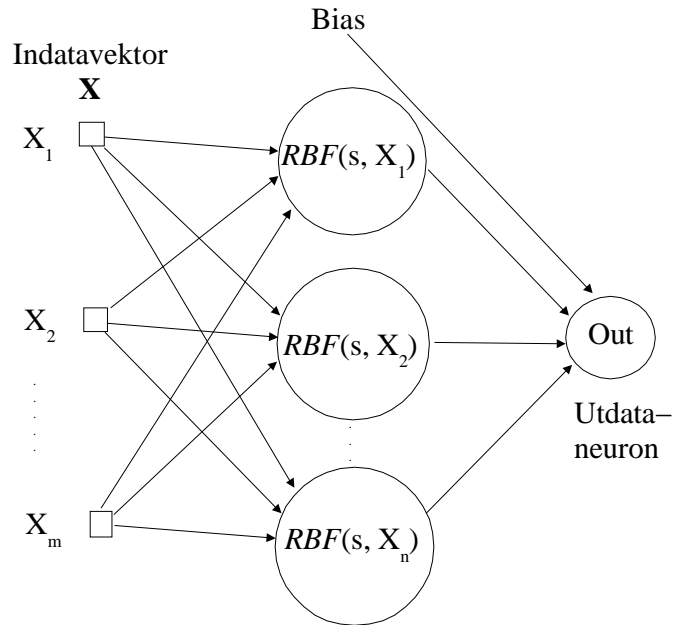
2.6 Kort om supportvektormaskiner

Neuronnät, biologiska såväl som konstgjorda, tränas med hjälp av representativa mönster och har den önskvärda egenskapen att de kan generalisera [19]. Vidare råder dem emellan intern representation, vilket innebär att minnet sprids ut – andra delar tar över vid förluster. Viktade kopplingar tros ligga bakom minnet hos biologiska neuronnät. Detta försöker efterliknas också då konstgjorda neuronnät byggs.

2.6.1 Artificiella neuronnät och andra lärande system

Artificiella neuronnät (ANN), vilket även innefattar supportvektormaskiner (SV-maskiner), är lärande maskiner vilka hämtat inspiration från verkliga neuronnät. SV-maskiner dock i mindre utsträckning än traditionella ANN. Ett neuron är en nervcell, vilken består av en cellkropp (*soma*), utskott (*axon*) och mottagardelar (*dendrit*). Neuronnät är kapabla till massivt parallella processer, vilket gör att indata kan bearbetas väldigt snabbt. De kan samla in och omarbete inkommande signaler för att skicka denna information vidare till andra neuron via speciella kopplingar (*synapser*). På ett liknande sätt är det önskvärt att ett ANN också skall kunna lära sig komplexa samband mellan multipla variabler och sedan reducera komplexiteten till ett enda utvärde.

Strukturen hos ett enkelt neuronnät är följande: indatanoder fungerar som ett inlager (identitetsavbildning), vikter mellan indatalagret och ett utdatalager (ändrar insignal till utsignal, en eller flera noder) fungerar som minne. Utdatalagret ändrar indata med en funktion (eng. *squashing function*), vilken kan vara allt ifrån linjär till sigmoid. Eventuellt kan ett eller flera lager med gömda noder (eng. *hidden layers*) existera, se *Figur 3*.



Indatalager med dimensionen m Gömt lager med n inre-
kernelprodukter

Figur 3. Arkitekturen hos en supportvektormaskin med en radialbasfunktion som kernelfunktion, liksom i detta arbete. Radialbasfunktioner är lokala och radiellt symmetriska. Bild från [14].

2.6.2 Generella användningar

Vad kan då artificiella neuronnät göra? De kan bland annat utföra klassificering och förutsägelse. Detta utnyttjas i detta arbete. Meningen är att SV maskinen skall, på bästa möjliga sätt, tala om utifall två protein har en strukturell likhet eller inte.

I andra fall används ANN till bland annat larmanordningar, textigenkänning, simuleringar, reglersystem och optimering [19].

2.6.3 Idén med supportvektormaskiner

Supportvektormaskiner (SV-maskiner) är lärande maskiner som kan utföra binär klassificering och icke-linjär regression.

Huvudidén med SV-maskiner är att konstruera ett hyperplan, vilket fungerar som en separerande yta, på ett sådant sätt att marginalen av separation mellan de positiva och negativa exemplen är maximerad.

SV-maskiner transformerar den n -dimensionella indatarymden till en högdimensionell egenskapsrymd [20]. I denna högdimensionella rymd konstrueras en linjär klassificerare. Maskinen följer en princip med rötterna inom statistisk inlärningsteori. Mer exakt, SV-maskinen är en approximerad implementation av *strukturell riskminimering* [21].

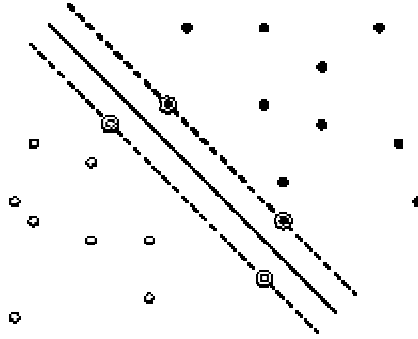
2.6.4 Lärande maskiner

Inom datavetenskapen använder sig de flesta system av induktivt lärande – utifrån en mängd givna exempel hittas gemensamma egenskaper. Några problem som bidrog till utvecklandet av SV-maskiner var bland annat problemen med att sätta rätt viktning (eng. *bias*), svårigheter med kapacitetskontroll och överinlärning. För att erhålla bäst generalisering gäller det att hitta den rätta balansen mellan komplexiteten hos träningsmängden och kapaciteten hos maskinen [19]. Termen kapacitet kan beskrivas som möjligheten för maskinen att lära sig en träningsmängd utan några fel. Centralt vid konstruktionen av en SV-maskin är den inre kernelprodukt (kernelfunktionen bildar ett hyperplan) mellan en supportvektor s och en vektor x dragen från indatarymden [19].

Supportvektorerna består av en del av träningsdata framtaget med algoritmen. Beroende på hur den inre kernelprodukten är framtagen, kan konstruktion ske av olika inlärningsmaskiner, vilka är karakteriserade av en icke linjär avgränsningsyta som ger dem dess egenskaper. Speciellt användas SVM-inlärningsalgoritmen för att konstruera följande tre typer av maskiner: polynomiala inlärningsmaskiner, radialbasfunktionsnätverk och tvålagars-perceptroner (med ett gömt lager) [19].

2.6.5 Den linjära supportvektormaskinen

Givet en träningsmängd som en sekvens av l specificerade punkter i \mathbb{R}^n , så är uppgiften att hitta det optimala hyperplan som separerar dem. Antag att vi har ett hyperplan som separerar träningsdata. Dess ekvation är då $w \cdot x + b = 0$, där w är normalen till hyperplanet. Låt d_+ (d_-) vara det kortaste avståndet från hyperplanet till det närmsta positiva (negativa) exemplet. Marginalen för det separerade hyperplanet är då $(d_+) + (d_-) = 2d_+$ och för det linjärt separerbara fallet kommer supportvektoralgoritmen helt enkelt att försöka hitta det hyperplan med den största marginalen. För det linjärt separerbara fallet är alltså $d_+ = d_- = 1 / |w|$ och marginalen blir då $2 / |w|$, således hittas det optimala hyperplanet genom att minimera $|w|$ [20]. Se *Figur 4*.



Figur 4. En typisk lösning för det tvådimensionella fallet. Supportvektorerna, träningsdata, är de punkter som ligger närmast avgränsningsytan och därför är svårast att klassificera. Den solida linjen är hyperplanslösningen, marginalen är avståndet mellan de två parallella streckade linjerna. De positiva och negativa exemplen som är märkta med cirklar kallas supportvektorer. Bild från [20].

Notera att antalet supportvektorer normalt är litet jämfört med storleken på träningsmängden. Hyperplanslösningen är endast definierad av supportvektorerna, om all övrig träningsdata avlägsnas så kommer inte hyperplanet att påverkas [20].

2.6.6 Större problem

Problem med optimering av supportvektorer kan lösas analytiskt endast då mängden träningsdata är litet, eller för de separerbara fall då det är känt på förhand vilka av träningsdata som kommer att bli supportvektorer [20]. För större problem finns flera olika tekniker som används. Ett enkelt tillvägagångssätt kan vara att först notera de optimala villkoren som lösningen måste uppfylla, sedan definiera en startegi för att nå optimalitet genom att uniformt flytta hyperplanet mot maximal marginal och till sist bestämma en uppdelningsalgorithm så att endast delmängder av träningsdata behöver bli behandlade samtidigt [20].

Begränsat datorminne kan vara ett hinder vid större problem. I sådana fall används en uppdelningsalgorithm (eng. *decomposition*). Alla uppdelningsalgoritmer tenderar att anta att antalet supportvektorer är litet jämfört med storleken på träningsdata. Utmaningen är därför att identifiera supportvektorerna före eller under tiden som det optimala hyperplanet eftersöks. SV-maskiner har en intressant egenskap: både tränings- och testfunktionerna beror endast på

kernelfunktionen [20]. Träningen av en SV-maskin är ett kvadratisk programmeringsproblem som är attraktivt av två orsaker: för det första kommer ett globalt extremum av felytan garanterat att hittas, där felet härör från skillnaden mellan den önskade responsen och utdata från maskinen, och för det andra kommer beräkningen utföras effektivt. Det är dessutom mycket viktigt att använda en lämplig inre kernelprodukt, maskinen beräknar automatiskt alla viktiga nätverksparametrar efter val av kernelfunktion [20].

2.6.7 Nackdelar

Trots SV-maskinens elegans och kraftfullhet, har den sina begränsningar [20]. En begränsning är val av kernelfunktion: när kernelfunktionen är vald finns bara en parameter kvar att använda sig av, nämligen straffpoängen, eller kostnadsfaktorn, då SV-maskinen gör fel. Maskinen går dessutom långsamt och det krävs stora tränings- respektive testmängder för att nå ett gott resultat. Långsamheten beror främst på att det inte finns någon kontroll över antalet datapunkter valda av inlärningsalgoritmen för att användas som supportvektorer. Vidare kan ingen fördel erhållas genom att inkorporera tidigare kunskap om problemen då SV-maskinen skall designas [20].

3 Metoder och material

I detta arbete användes en supportvektormaskin (SV-maskin), vilken skulle utföra binär klassificering av olika proteinmodeller. För att kunna mäta prestandan hos SV-maskinen beräknades ett värde kallat *Mathews korrelationskoefficient*. Många olika parametrar varierades, såsom inställningar hos SV-maskinen och olika indataparametrar till denna.

3.1 SVM-light

Supportvektormaskinen som användes i detta arbete laddades ned från en webbplats [22] & [23]. Maskinen heter SVM-light och är en implementation av *Vapnik's Support Vector Machine* [24]. Några egenskaper hos SVM-light är snabb optimeringsalgoritm, korrekt hanterande av flera tusen supportvektorer samt möjligheten att använda tiotusentals träningsexempel. SVM-light är, enligt skaparen och författaren till handledningen, Thorsten Joachims, speciellt optimerad för mönsterigenkänning [23].

3.1.1 Parametrar hos SVM-light

En parametrar som kan varieras hos SVM-light är kostnadsfaktorn j , genom vilken viktning ges åt att positiva, eller negativa, träningsdata klassificeras rätt. Vidare kan olika kernelfunktioner användas (se *Supportvektormaskiner*), såsom linjära, polynomiala, sigmoida eller radialbasfunktioner. Användaren kan även använda en egendefinierad kernelfunktion. Dessa olika funktioner har dessutom var för sig olika parametrar som kan ställas in. I detta arbete användes en radialbasfunktion, se *Ekvation 2*. Övriga funktioner fungerade inte tillfredsställande då SV maskinen skulle utföra klassificeringen. Antingen blev resultatet dåligt eller så utfördes träningen otillfredsställande.

$$RBF = e^{-g*(|a*x-b|)^2}$$

Ekvation 2. Radialbasfunktionen vilken användes som kernelfunktion i SVM-light. Exponentialfaktorn g varierades mellan 10^{-5} och 10^{-7} , a och b är konstanter.

Kostnadsfaktorn j varierades mellan 0,5, negativa exempel (TN) rätt favoriseras, och 2,0, positiva exempel (TP) rätt favoriseras. Större

värden på j tolererades inte av SVM-light. Då j sätts till 1,0 görs ingen viktning. Dessa parametrar var de enda två som ändrades under körningarna. Övriga parametrar bidrog inte till bättre resultat, varför dessa uteslöts. De olika variationerna av j och g beror på att inget entydigt svar gavs på vilken parameterinställning som var bättre än någon annan generellt. Det verkar som vissa inställningar fungerar bättre med viss indata och andra inställningar bättre med andra indata. Om exponentialfaktorn g och kostnadsfaktorn j sattes utanför dessa intervall ($g = [10^{-5}, 10^{-7}]$, $j = [0,5, 2,0]$) fungerade inte SVM-light, eller så blev resultatet dåligt.

3.2 Tränings- och testdata

När ett artificiellt neronnät eller en supportvektormaskin används behövs två olika grupper av data. En av de två grupperna används som träningsdata, och den andra används som testdata för att utvärdera prestandan hos nätverket.

De undersökta proteinerna var från gruppen PDB40d, SCOP version 1.39 [17]. Inga jämförda proteiner hade mer än 40% sekvensidentitet. Strukturer där aminosyraenheter saknades togs inte med i beräkningarna. Detta gav totalt 1,6 miljoner parvisa jämförelser av 1260 olika protein. Tränings- och testdata delades konsekvent upp i två lika stora grupper (det vill säga lika många tränings- som testdata). Lika många positiva som negativa exempel användes både för träning och test. Detta var ej representativt för de olika jämförelserna, negativa jämförelser (olika strukturer) var klart överrepresenterade, endast drygt 12200 jämförelser var positiva. Då stora filer, eller filer med för många negativa exempel, användes uppstod problem med SV maskinen varför träningsexemplen presenterades på ovanstående sätt. Tränings- och testdata presenterades som n -dimensionella vektorer för SV maskinen, n antog värden mellan ett (1) och 13. De olika dataparametrarna var de som anges i *Tabell 1*.

<i>Term</i>	<i>Betydelse</i>
1	längden på jämförelsen (stopp-start)
2	antal jämförda aminosyraenheter
3	längden på det ena proteinet
4	längden på det andra proteinet
5	värdet från den parvisa jämförelsen
6	total energi
7	energi för interaktioner mellan α -kolatomer
8	energi för sidokedjor
9	energi för sidokedjor mot α -kolatomer
10	andel sekundärstruktur som är lika
11	antal aminosyraenheter i ett gap
12	antalet aminosyraenheter i en α -helix eller β -sheet i ett gap
13	antalet gap i jämförelsen mellan proteinerna

Tabell 1. Dataparametrar till SVM-light. Kolumnen "Term" refereras till senare.

Förutom parametrarna i *Tabell 1* angavs också om proteinmodellerna var lika eller inte, enligt SCOP. Det rörde sig alltså om övervakad inläring. Den totala energin är summan av samtliga energier.

När SV-maskiner och neurala nätverk används, önskas så få paramerar som möjligt för bättre kontroll över förloppet. Därför kördes SV-maskinen med många olika parameteruppsättningar för att hitta en bra kombination av inställningar och data. Detta gjordes med 1000 positiva och 1000 negativa tränings- respektive testdata, med de nämnda värdena på kostnadsfaktorn j och exponentialfaktorn g . En koefficient (värdet från Mathews korrelationskoefficient) användes som kontroll av prestandan hos SV-maskinen. Parametrar som inte påverkade resultatet nämnvärt plockades bort, såväl data som parametrar hos SV-maskinen, se [22].

Det verkade som om sorteringen av data spelade en viss roll, sämre prestanda erhöles hos SV-maskinen då sorteringen var mindre strikt. Därför användes sortering enligt en striktare metod i samtliga fall. Hela datamängden sorterades efter lika eller inte lika (+1 respektive -1) och efter längden på den parvisa jämförelsen. I samtliga fall presenterades först de positiva exemplen och sedan de negativa. Varannan jämförelse gick till träningsmängden och varannan till

testmängden efter sorteringen för att få en jämn fördelning av data. Data var således jämnt fördelat i tränings- respektive testmängderna.

Av de 1,6 miljoner jämförelserna valdes endast en bråkdel. Om stora mängder data behandlades i SV-maskinen blev resultatet inte bättre, snarare sämre varför cirka 2000 tränings exempel och 2000 testdata användes i de flesta fallen. Användes till exempel 12000 data vardera för träning och testning istället för endast 2000 blev resultaten sämre. Dessutom ökar tiden exponentiellt vid varje körning då mer träningsdata används. I vissa fall fick SV-maskinen bearbeta data i flera dygn.

3.2.1 Mathews korrelationskoefficient

Mathews korrelationskoefficient (MC) är ett direkt mått på SV-maskinens prestanda och indikerar noggrannheten på klassificeringen (lika eller inte lika) [25]. Ett MC-värde på ett (1) erhålls om förutsägelsen av maskinen är helt rätt, det vill säga 100% korrekthet. Om det erhållna MC-värdet är noll (0) är resultatet inte bättre än en slumpvis gissning. Om värdet minus ett (-1) erhålls är klassningen helt fel, all data som skulle höra till den ena delmängden hamnade i den andra delmängden och vice versa, se *Ekvation 3*.

$$MC = \frac{((TP * TN) - (FP * FN))}{\sqrt{(TN + FN) * (TN + FP) * (TP + FN) * (TP + FP)}}$$

Ekvation 3. Mathews korrelationskoefficient. TP står för sant positiva (True Positives, sådana modeller som är lika enligt SCOP), TN står för sant negativa (True Negatives, modellerna är olika strukturellt enligt SCOP), FP står för falskt positiva (False Positives, SCOP har klassificerat modellerna som olika, men SV-maskinen har klassat dem som lika) och slutligen står FN för falskt negativa (False Negatives, modellerna är lika, men SV-maskinen har klassat dem som olika).

MC använder ett tröskelvärde för att separera korrekta från inkorrekta klassificeringar. I samtliga fall användes tröskelvärdet noll (0), vilket innebär att ingen speciell vikt gavs åt någon klassificering.

4 Resultat

Data formaterades med hjälp av programmeringsspråket Perl, som är utmärkt då det gäller att bearbeta stora datamängder. De första körningarna med data i SVM-light (SV-maskinen) var från en metod som innefattar både förutsägelse av sekundärstrukturen och en profil skapad av PSI-BLAST. Data framtagna med tre andra metoder fanns också tillgängliga, se *Jämförelse med data genererad från andra metoder*.

4.1 Optimering

Då en SV-maskin används finns tyvärr inga inställningar eller dataparametrar som i förväg kan sägas fungera bättre i kombination än andra. Testkörningar är det enda sättet att mäta prestandan och korrektheten.

4.1.1 Utvärdering av olika dataparametrar

För att kunna diskriminera mellan olika kombinationer av data kördes SV-maskinen med olika datakombinationer, men med konstanta parametrar hos SVM-light. Efter en första sällning testades de kombinationer som verkade mest lovande. För dessa olika kombinationer kontrollerades MC-värdena för att kunna avgöra vilka som verkade ge ett gott resultat. I samtliga fall användes värdena $g = 10^{-6}$ och $j = 1$. 2000 positiva och 2000 negativa träningssexempel samt 2000 positiva och 2000 negativa testdata användes, se *Tabell 2*.

<i>Data</i>	<i>MC-värde</i>
5 (referensvärde)	0,665
5, 6	0,694
5, 6, 7	0,725
5, 6, 7, 8	0,702
5, 6, 7, 8, 9	0,700
5, 6, 7, 8, 9, 11	0,692
5, 6, 7, 8, 9, 11, 12	0,683
5, 6, 7, 8, 9, 11, 12, 13	0,835
1, 5, 6, 7	0,835
1, 5, 6, 7, 8	0,835
1, 5, 6, 7, 8, 9	0,835
1, 5, 6, 7, 8, 9, 11	0,835

Tabell 2. För förklaring till kolumnen "Data", se Tabell 1. Kolumnen "MC-värdet" är värdet beräknat med Mathews korrelationskoefficient.

Som synes spelar vissa data mindre roll, eller till och med försämrar MC-värdet från testdata. Dessa data ignorerades på så vis att de inte inkluderades i framtida körningar av SV maskinen. Som referens till övriga MC-värden användes värdet då endast poängen från den parvisa jämförelsen inkluderades (MC = 0,665), se *Tabell 3*. Eftersom bättre resultat erhöles då flera dataparametrar inkluderas bidrar dessa parametrar till att lättare skapa det hyperplan som skall utföra den binära klassificeringen. Fler dimensioner underlättar linjär separerbarhet, varför detta ej var oväntat. Dock, ett så litet antal parametrar som möjligt är önskvärt, varför de bästa kombinationerna testades med olika värden på SV-maskinens inställningar för att reducera antalet parametrar, se *Jakten på det bästa MC-värdet*.

4.1.2 Utvärdering av olika inställningar hos SVM-light

Olika dataparametrar (tränings- och testdata) och olika inställningar hos SV maskinen, kostnadsfaktorn j och exponentialfaktorn g , användes för att hitta de kombinationer som gav de bästa resultaten. Värdet på Mathews korrelationskoefficient (MC-värdet) användes som ett mått på prestandan hos SV-maskinen. Till att börja med kördes SV-maskinen med 1000 positiva och 1000 negativa tränings- och testdata (4000 totalt). Antalet är ej exakt, detta spelar

dock ingen roll då det gäller MC-värdet. *Tabell 3* visar variationen i resultat med samma dataparametrar, men olika inställningar på kostnadsfaktorn j och exponentialfaktorn g på SV-maskinen.

$MC(g, j)$	$j = 0,5$	$j = 1$	$j = 1,5$	$j = 2$
$g = 10^{-5}$	0,837	0,852	0,862	0,851
$g = 5 \times 10^{-5}$	0,846	0,850	0,850	0,842
$g = 10^{-6}$	0,820	0,835	0,840	0,828
$g = 5 \times 10^{-6}$	0,833	0,851	0,857	0,845
$g = 10^{-7}$	0,736	0,770	0,768	0,758
$g = 5 \times 10^{-7}$	0,786	0,823	0,817	0,806

Tabell 3. Olika värden på kostnadsfaktorn j och exponentialfaktorn g , samt de MC-värden som erhöles då dessa varierades. Data som användes var 1, 5, 6, och 7, se Tabell 1.

Den bästa funktionen var radialbasfunktionen, användes någon av de övriga funktionerna blev resultatet dåligt eller så fungerade inte kombinationen kernelfunktion och data alls. Eftersom det är omöjligt att kontrollera alla kombinationer av inställningar hos SVM-light och dataparametrar är det möjligt att någon annan kombination skulle fungera bättre.

Om diskriminering görs för MC-värden mindre än 0,85 och inställningarna hos SVM-light kontrolleras blir resultatet att i 33% av de bästa körningarna var kostnadsfaktorn $j = 1$ (ingen viktning), i 60% var $j = 1,5$ (viktning åt att klassificera positiva modeller rätt) och i 7% av de högsta MC-värdena var $j = 2$. Exponentialfaktorn g var vid de flesta av de bästa körningarna inställd på 10^{-5} eller 5×10^{-6} , se *Tabell 4*.

j	<i>Andel MC > 0.85</i>	g	<i>Andel MC > 0.85</i>
1	33%	10^{-5}	47%
1,5	60%	5×10^{-5}	13%
2,0	7%	5×10^{-6}	40%

Tabell 4. Inställningar hos SVM-light då de högsta MC-värdena erhöles.

Det bästa MC-värdet, $MC = 0,862$, erhöles då g sattes till 10^{-5} , detta värde samt då g sattes till 5×10^{-6} fungerade generellt bäst vid de olika testkörningarna. Varför till exempel $g = 5 \times 10^{-5}$ eller $g = 10^{-6}$ gav sämre MC-värde förblir en obesvarad fråga. Kostnadsfaktorn j bidrog till bättre MC-värden då viktning åt att positiva träningsexempel skulle favoriseras ($j > 1$) eller då ingen viktning alls användes ($j = 1$).

4.2 Jakten på det bästa MC-värdet

Ett program som automatiskt loopade igenom olika kombinationer av dataparametrar och inställningar hos SV-maskinen konstruerades. Totalt kördes 25 olika tränings- och testdatafiler (2000 träningsexempel och 2000 testdata) med 18 olika inställningar hos SV-maskinen. De bästa resultaten, MC-värde större än 0,85, redovisas i *Tabell 5*. Dessa resultat är betydligt bättre än då endast värdet från den parvisa jämförelsen användes som indataparameter till SV maskinen.

<i>Data</i>	<i>Inställningar SVM-light</i>	<i>MC-värde</i>
1, 5, 6, 7	$g = 10^{-5}, j = 1$	0,852
1, 5, 6, 7	$g = 10^{-5}, j = 1,5$	0,862
1, 5, 6, 7	$g = 10^{-5}, j = 2$	0,851
1, 5, 6, 7	$g = 5 \times 10^{-5}, j = 1,5$	0,850
1, 5, 6, 7	$g = 5 \times 10^{-5}, j = 1$	0,851
1, 5, 6, 7	$g = 5 \times 10^{-6}, j = 1,5$	0,857
1, 5, 6, 7	$g = 5 \times 10^{-6}, j = 1$	0,852
1, 5, 6, 7, 8	$g = 10^{-5}, j = 1,5$	0,854
1, 5, 6, 7, 8	$g = 5 \times 10^{-6}, j = 1,5$	0,855
1, 5, 6, 7, 8	$g = 5 \times 10^{-6}, j = 1$	0,850
1, 5, 6, 7, 8, 9	$g = 10^{-5}, j = 1,5$	0,852
1, 5, 6, 7, 8, 9	$g = 5 \times 10^{-6}, j = 1,5$	0,850
1, 5, 6, 7, 8, 9	$g = 10^{-6}, j = 1$	0,851
1, 5, 6, 7, 8, 9, 11	$g = 10^{-5}, j = 1,5$	0,854
1, 5, 6, 7, 8, 9, 11	$g = 5 \times 10^{-5}, j = 1,5$	0,853

Tabell 5. För förklaring till kolumnen "Data", se Tabell 1. "Inställningar SVM-light" visar viktningen för att positiva respektive negativa exempel favoriseras, kostnadsfaktorn j , och exponentialfaktorn g som ingår i radialbasfunktionen som delar upp data binärt. "MC-värde" är värdet beräknat med Mathews korrelationskoefficient.

Det framgår med stor tydlighet att bättre MC-värde *inte* erhålls då vissa dataparametrar tas med vid körningen av SV-maskinen. Således bör endast längden på den parvisa jämförelsen, värdet från den parvisa jämförelsen, den totala energin och energin för interaktioner mellan α -kolatomer användas som data för att minimera antalet parametrar. Dessa parametrar tillsammans utgjorde den bästa kombinationen av de som utvärderades.

4.3 Jämförelse med data genererade från andra metoder

Fyra olika metoder för strukturigenkänning hade använts av SBC innan detta examensarbete påbörjades, nämligen global jämförelse, global jämförelse av sekvenserna mot en profil skapad av PSI-BLAST, strukturbestämning av sekundärstrukturen och en metod som beaktar både sekundärstrukturen och profilen skapad av PSI-BLAST. För de fyra metoderna jämfördes alla proteiner mot varandra. Eftersom den metod som innefattar både sekundärstruktur och profil ansågs vara den som skulle fungera bäst var det denna som användes i första hand. Dock testades SVM-light med körningar från samtliga metoder, samt en mix av dem alla. De inställningar som användes hos SVM-light var de som hade fungerat bäst i tidigare körningar, se *Tabell 4*. Som dataparametrar användes de som gett bäst resultat hittills. 1000 positiva och 1000 negativa tränings- och testdata användes för samtliga metoder. Alla parvisa jämförelser fanns representerade i samtliga metoder.

Då tränings- och testdata togs fram söktes efter parvisa jämförelser mellan protein som fanns representerade vid samtliga metoder, vilket inte var något villkor vid framtagandet av de bästa dataparametrarna och inställningarna då metoden som inkluderar både en profil och sekundärstruktur användes. Resultatet från dessa körningar visar att det var svårare för SV-maskinen att hitta något mönster i data då dessa jämförelser gjordes än då data framtoogs för optimering. I de fallen valdes varannan jämförelse till träningsdata och varannan jämförelse till testdata, sorterade efter lika respektive inte lika samt efter längden på jämförelsen. På samma sätt gjordes vid dessa körningar, men det var inte samma jämförelser som var representerade dessa gånger som i de körningar som användes för val av parametrar och kernelfunktion samt inställningar hos denna. Tränings- respektive testdata verkar ha varit mer lika, eller i alla fall lättare att klassificera, vid körningarna för optimering än då samtliga jämförelser skulle finnas representerade hos alla metoder, därav bättre MC-värde.

Resultaten var alltså sämre än de som erhöles från tidigare körningar. Metoden som inkluderar både försutsägning av sekundärstrukturen och profilen från PSI-BLAST gav högst värden. De inställningar hos SVM-light som användes var $g = 10^{-5}$ och $g = 5 \times 10^{-6}$ samt $j = 1$ och $j = 1,5$ eftersom dessa hittills hade gett de bästa resultaten. För vilka data som inkluderades, se *Tabell 6*. Det är möjligt att andra parameterinställningar skulle givit andra resultat för de andra metoderna, detta är lite av dilemmat med SV-maskiner – det finns inga inställningar och dataparametrar som på förhand kan sägas fungera bättre än andra. Testkörningar är den enda metoden för att finna de bästa parameterinställningarna. Se *Tabell 6* för det bästa samt det sämsta resultatet från varje metod.

<i>Metod</i>	<i>Inställningar SVM-light</i>	<i>MC-värde</i>	<i>Referensvärde</i>
Global jämförelse	$g = 10^{-5}, j = 1$	0,695	0,523
Global jämförelse	$g = 5 \times 10^{-6}, j = 1$	0,650	0,529
Sek. struktur	$g = 10^{-5}, j = 1$	0,760	0,428
Sek. struktur	$g = 5 \times 10^{-6}, j = 1,5$	0,743	0,382
PSI-BLAST	$g = 10^{-5}, j = 1$	0,704	0,473
PSI-BLAST	$g = 5 \times 10^{-6}, j = 1,5$	0,636	0,451
PSI-B. och sek. strukt.	$g = 10^{-5}, j = 1$	0,775	0,440
PSI-B. och sek. strukt.	$g = 5 \times 10^{-6}, j = 1,5$	0,744	0,354

Tabell 6. De bästa och de sämsta resultaten från körningar då data genererad på olika sätt användes. De inställningar hos SVM-light som givit bäst resultat då metoden som inkluderade både modellen från PSI-BLAST och sekundärstrukturen användes. Data var längden på den parvisa jämförelsen, värdet från den parvisa jämförelsen, den totala energin och energin för interaktioner mellan α -kolatomer. Dessa data hade visat sig fungera bäst vid tidigare körningar. Som referens till MC-värdena användes värdet från de olika metoderna då endast poängen från den parvisa jämförelsen inkluderades.

Av ovanstående tabell framgår att inkludering av sekundärstrukturen verkar spela en betydande roll. Inkludering även av profilen skapad av PSI-BLAST verkar förbättra resultatet ytterligare. Inkludering av endast profilen förbättrar inte resultatet. Som referens kan nämnas de MC-värden, som är tabellerade i *Tabell 6*, då endast värdet från den parvisa jämförelsen användes som dataparameter. Skillnaden är markant då övriga dataparametrar också tas med i

tränings- respektive testmängderna. Notera att då endast värdet från den parvisa jämförelsen används, så är inte det högsta MC-värdet nödvändigtvis erhållet med samma inställningar som gav det högsta MC-värdet med de fyra dataparametrarna omnämnda i *Tabell 1*.

Till sist gjordes en körning i SVM-light med samtliga fyra metoder representerade. 500 positiva och 500 negativa tränings- och testdata presenterades från varje metod. Resultatet var inte bättre än då endast en metod användes, MC-värdet antog värden runt 0,7 i de bästa fallen. SVM-light blev troligtvis konfunderad av all information och kunde inte hitta något mönster att lära sig av för att sedan utnyttja detta till klassificeringen, se *Tabell 7*.

<i>Metod</i>	<i>Inställningar SVM-light</i>	<i>MC-värde</i>	<i>Referens-värde</i>
Mix av samtliga	$g = 10^{-5}, j = 1$	0,708	0,326
Mix av samtliga	$g = 5 \times 10^{-6}, j = 1$	0,670	0,360
Mix av samtliga	$g = 10^{-5}, j = 1,5$	0,708	0,377
Mix av samtliga	$g = 5 \times 10^{-6}, j = 1,5$	0,647	0,359

Tabell 7. Samtliga metoder presenterades för SVM-light. Resultatet var inte bättre än de övriga resultaten. Som referensvärde användes värdet då endast poängen från den parvisa jämförelsen inkluderades.

Valet av inställningar hos SV-maskinen är uppenbarligen viktigt i samtliga fall. De bästa inställningarna varierar från gång till gång, varför flera olika inställningar bör testas för olika data.

5 Diskussion

Då data presenterades på ett ofördelaktigt sätt, eller då SV-maskinen var felaktigt inställd, blev resultaten mindre bra. Det är således av yttersta vikt att finna den bästa möjliga presentationen av data och de optimala inställningarna hos SV-maskinen. Vidare är det av stor betydelse att rätt mängd data används och att elementen i träningsexemplen samt testdata är av rätt karaktär, det vill säga att de bidrar till att förbättra prestandan hos SV-maskinen. Det verkar vara nödvändigt att avvika från förhållandet mellan hur vanligt det är att proteiner på veckningsnivå är lika naturligt – få proteiner är tredimensionellt lika då den primära strukturen är väldigt olika. Här användes konsekvent lika stora mängder lika som olika tränings- och testdata. Om förhållandet mellan lika respektive olika jämförelser inte anpassas generaliserar SV-maskinen så att samtliga jämförelser anses negativa. Större tränings- respektive testmängder gav inte bättre resultat, snarare sämre. De större mängderna tog längre tid att behandla och SV-maskinen fick svårare att bearbeta data än om mindre mängder användes.

5.1 Val av dataparametrar

När en supportvektormaskin används kan data presenteras för maskinen på många olika sätt. Dataparametrarna kan vara framtagna på olika sätt, olika dimensioner på indata till supportvektormaskinen kan användas, olika mängd av tränings- respektive testdata kan användas och dataparametrar som är bra ena gången behöver nödvändigtvis inte vara det en annan gång. I detta arbete användes de dataparametrar som redovisats under *Parametrar hos SVM-light*. Dessa fungerade bäst, men det är mycket möjligt att andra dataparametrar skulle kunnat bidra till bättre resultat. Dessutom verkade det som om det var av betydelse hur data presenterades, då tränings- och testdata var valda så att de båda delmängderna var relativt lika erhöles bättre resultat. Tilläggas skall att det rörde sig om 1,6 miljoner parvisa jämförelser, varför urvalet till endast ett par tusen data skulle kunna göras på många andra sätt.

Om alla fyra metoder för att generera proteinmodeller, vilka jämfördes mot varandra, användes i en kombination klassade SV-maskinen inte dessa bättre än om varje metod användes enskilt. Högst MC-värde, ett mått på SV maskinens prestanda, erhöles med den metod som innefattade både en profil skapad av PSI-BLAST och information om sekundärstrukturen. Alltså är den information som ges av modellen och sekundärstrukturen betydande, de ger troligtvis rätt karaktär åt indataparametrarna. De bästa värdena på

korrekt klassificering erhöjls då fyra parametrar skickades med som indata till SV-maskinen, dessa var: längden på jämförelsen, ett värdet från den parvisa jämförelsen, total energi och energi för interaktioner mellan α -koltomer. Således är dessa viktiga egenskaper för att SV-maskinen skall kunna utföra en så bra klassificering som möjligt. Övriga dataparametrar gav inte samma tydliga förbättring varför de kan antas vara mindre karaktäristiska för de olika jämförelserna.

5.2 Andra parametrar att variera

Inställningarna hos SV-maskinen kan varieras i nästan det oändliga. Här ändrades endast två parametrar, det finns hos SVM-light ett femtontal till att variera [19]. Vilka värden som skall sättas på dessa parametrar är på intet vis entydigt – vidare kan de anta i stort sett vilka värden som helst. Det är omöjligt att testa alla möjligheter som ges, samtidigt som olika testkörningar är det enda sättet på att få reda på om prestandan hos SV-maskinen är godtagbar. Förbättringar, eller fler tester, skulle därför kunna vara ett sätt att få bättre MC-värden. Här prövades, förutom exponentialfaktorn g och kostnadsfaktorn j , även ett viktat hyperplan. Detta fungerade sämre än ett oviktat, varför viktning uteslöts ganska omgående. Vilken kernelfunktion som skall användas är inte heller trivialt, att testa är det enda sättet att få fram de bästa resultaten. Kombinationen dataparametrar, funktioner och inställningar kan ställas in i otaliga varianter. Den optimala är svår, om inte omöjlig, att finna. Just problemet med att veta vilka värden som skall sättas på olika parametrar är ett dilemma med SV-maskiner och andra lärande maskiner.

De värden som gavs åt parametrarna hos SV-maskinen var av stor betydelse. Sattes de till vissa värden fungerade inte maskinen eller så blev klassificeringen dålig. Den enda kernelfunktion som fungerade var en radialbasfunktion. En orsak till att den fungerade men inte de andra kan vara att radialbasfunktioner är lokala och radiellt symmetriska, vilket inte de övriga kernelfunktionerna är. Den exponentialfaktor som användes bidrog i högsta grad till hur bra resultatet blev; varför vissa värden fungerade bättre än andra kunde inte klargöras av det utförda arbetet. Om kostnadsfaktorn ändrades så att en liten fördel gavs åt någon av klassningarna bidrog det inte nämnvärt till att förbättra resultaten. Eventuellt förbättrar en favorisering av de positiva exemplen antalet korrekt klassificerade jämförelser. Vid för stor viktning åt någon klass fungerade inte SV-maskinen alls. Detta kan bero på att SV-maskinen då klassificerade alla som den favoriserade klassen, vilket var fallet även om för stor delmängd av tränings- och testdata var positiv eller negativ.

5.3 Slutsatser

Supportvektormaskiner kan, med rätt inställningar och karaktäristiska dataparametrar, användas för att utföra binär klassificering av proteiner. Val av parametrar och kernelfunktion är av yttersta vikt varför noggranna utvärderingar av parametrar samt funktioner bör göras för att erhålla ett så gott resultat som möjligt. Mängden på tränings- respektive testdata är också av betydelse. Vidare ger inkludering av information om sekundärstrukturen avsevärt bättre resultat än om denna inte inkluderas.

Då det finns många parametrar att ändra kan ytterligare optimering av dessa säkert ge ett ännu bättre resultat. Andra jämförelse-algoritmer för att ta fram data till SV maskinen skulle vara intressant att prova. Vidare skulle det vara spännande att köra samma data i ett artificiellt neuronnät, som i [1]. En ambition var att göra det, men tiden räckte inte till. Olika SV-maskiner skulle också kunna testas för att jämföra resultaten, liksom andra typer av lärande maskiner.

6 Referenser

- [1] D. T. Jones: *GenTHREADER: An efficient and reliable protein fold recognition method for genomic sequences*. J. Mol. Biol. 287: 797–815, 1999.
- [3] T. K. Attwood & D. J. Parry–Smith: *Introduction to Bioinformatics*. Prentice Hall, ISBN 0–582–327881, 1999.
- [2] Kisac Bioinformatics: *En kort summering av bioinformatik*. <http://kisac.cmb.ki.se/course/courseKTH.html>, 1999 (1999–01–26).
- [4] S. B. Needleman & C. D. Wunch: *A general method applicable to the search for similarities in the amino acid sequence of two proteins*. J. Mol. Biol. 48: 443–453, 1970.
- [5] T. F. Smiths & M. S. Waterman: *Identification of common molecular subsequences*. J. Mol. Biol. 147: 195–197, 1981.
- [6] A. Elofsson: *A study on how to best align protein sequences*. SBC, 2000.
- [7] J. Hargbo & A. Elofsson: *A study of hidden Markov models that use predicted secondary structures for protein fold recognition*. Proteins: Struct., Funct. & Genet., 36: 68–87, 1999.
- [8] D. T. Jones, W. R. Taylor & J. M. Thornton: *A new approach to protein fold recognition*. Nature: 358: 86–89, 1992.
- [9] C. M. R. Lemer, M. J. Rومان & S. J. Wodak: *Protein structure prediction by threading methods. Evaluation of current techniques*. Proteins: Struct., Func. & Genet., 23: 337–355, 1995.
- [10] E. Lindahl & A. Elofsson: *Identification of related proteins on family, superfamily and fold level*. J. Mol. Biol. 295: 613–625, 2000.
- [11] B. Rost & C. Sander: *Prediction of protein secondary structure at better than 70% accuracy*. J. Mol. Biol. 267: 1026–1038, 1997.
- [12] D. T. Jones: *Progress in protein structure prediction*. Curr. Op. Struct. Biol. 7: 377–387, 1997.
- [13] D. T. Jones & J. M. Thornton: *Potential energy functions for threading*. Curr. Op Struct. Biol. 6: 210–216, 1996.
- [14] H. Flöckner, F. Domingues & M. J. Sippl: *Proteins folds from pair interactions – A blind test in fold recognition*. Proteins: Struct., Funct. & Genet. 1: 129–133, 1997.
- [15] B. Park & M. Levitt: *Energy functions that discriminate x-ray and nearnative folds from well-constructed decoys*. J. Mol. Biol. 258: 367–392, 1996.

- [16] D. A. Hands & M. Levitt: *A lattice model for protein structure prediction at low resolution*. Prot. Natl. Acad. Sci. 89: 2536–2540, 1992.
- [17] A. G. Murzin, S.E. Brenner, T. Hubbard & C. Chothia: *SCOP – A structural classification of proteins database for the investigation of sequences and structure*. J. Mol. Biol. 247: 536–540, 1995.
- [18] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, P. E. Bourne: *The Protein Data Bank*. Nucleic Acids Research, 28 pp. 235–242, 2000.
- [19] S. Haykin: *Neural networks – A comprehensive foundation*, Second edition, Chapter six. Prentice Hall, ISBN 0–13–273350–1, 1999.
- [20] E. Borovikov: *Support vector machines for pattern recognition*.
<http://www.umiacs.umd.edu/users/yab/SVMForPatternRecognition/svm.html>, 1998 (2000–09–01).
- [21] Royal Holloway, University of London, Department of Computer Science: *Support vector machine*.
<http://www.clrc.rhbnc.ac.uk/research/SVM>, 1998 (2000–09–01).
- [22] A. Smola, B. Schölkopf et al.: *Kernel machines*.
<http://www.kernel-machines.org>, 2000 (2000–08–14).
- [23] T. Joachims: *SVM–light, a support vector machine*.
<http://www.kernel-machines.org>, 1999 (2000–08–14).
- [24] V. N. Vapnik: *The nature of statistical learning theory*. Springer, 1995.
- [25] B. Mathews: *Biochemical biophysics*. Acta 405: 442–451, 1975.