TEHNICI DE RECUNOAŞTERE AUTOMATĂ ÎN ANALIZA ŞI DETERMINAREA SIMILARITĂŢII SECVENŢELOR BIOLOGICE

Teză destinată obţinerii
titlului ştiinţific de doctor
la
Universitatea „Politehnica" din Timişoara
în domeniul AUTOMATICA
de către

Ing. Alina Bogan-Marta

Conducă: prof.univ.dr. ing. Nicolae Budişan
prof.univ.dr. ing. Nicolae Robu

Referenţi ştiinţifici: prof.univ.dr. ing. Mircea Petrescu
prof.univ.dr. ing. Mircea Ivănescu
prof.univ.dr. ing. Vladimir Creţu

Ziua susţinerii tezei: 19 ianuarie 2007

Seriile Teze de doctorat ale UPT sunt:

1. Automatică
2. Chimie
3. Energetică
4. Ingineria Chimică
5. Inginerie Civilă
6. Inginerie Electrică
7. Inginerie Electronică şi Telecomunicaţii
8. Inginerie Industrială
9. Inginerie Mecanică
10. Ştiinţa Calculatoarelor
11. Ştiinţa şi Ingineria Materialelor

Universitatea „Politehnica" din Timişoara a iniţiat seriile de mai sus în scopul diseminării expertizei, cunoştinţelor şi rezultatelor cercetărilor întreprinse în cadrul şcolii doctorale a universităţii. Seriile conţin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susţinute în universitate începând cu 1 octombrie 2006.

România, 300159 Timişoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Cuvânt înainte

Teza de doctorat a fost elaborată pe parcursul activităţii mele în cadrul Departamentului de Calculatoare al Universităţii din Oradea în colaborare cu Departamentul de Automatică al Universităţii „Politehnica" din Timişoara, grupul de cercetare în „Recunoaşterea vorbirii" al Universităţii Katolice din Leuven, Belgia şi laboratorul de „Inteligenţă artificială şi analiza informaţiei" al Universităţii Aristotel din Tesalonic, Grecia.

Îi mulţumesc d-lui prof. Nicolae Budişan pentru înţelegere şi sprijin de-a lungul acestor ani de activitate doctorală.

Adresez cele mai sincere mulţumiri d-lui prof. Nicolae Robu pentru răbdare, susţinerea continuă şi îndrumările pe parcursul acestui drum, şi sub încurajările căruia am reuşit să îmi conturez o direcţie în activitatea desfăşurată.

Le mulţumesc colegilor şi supervizorilor din grupurile de cercetare din care am facut parte pentru ajutorul oferit pe timpul perioadelor de activitate desfăşurate în cadrul acestora. De asemenea, îndrept mulţumiri şi recunoştinţă tuturor colegilor şi persoanelor care şi-au oferit sprijinul în realizarea acestei teze.

Nu în ultimul rand le mulţumesc părinţilor pentru înţelegere, ajutor şi dragostea cu care m-au susţinut în realizarea acestui obiectiv.

Timişoara, ianuarie 2007 Alina Bogan-Marta

Părinţilor mei.

Bogan-Marta, Alina

Tehnici de recunoaştere automată în analiza şi determinarea similarităţii secvenţelor biologice

Teze de doctorat ale UPT, Seria 1, Nr. 2, Editura Politehnica, 2006/2007, 212 pagini, 27 figuri, 9 tabele.

ISSN: 1842-5208

ISBN: 978-973-625-415-4

Cuvinte cheie:
secvenţe biologice, algoritmi de determinare a similarităţii, tehnici automate de recunoaştere, entropie, modele lingvistice Markov

Rezumat:

Multe cercetări în inteligenţă bioinformatică sunt focalizate pe algoritmi de învăţare şi sisteme integrate care să permită transformarea secvenţelor biologice, a observaţiilor şi cunoştinţelor în informaţie structurată şi semnificativă pe care biologii o pot interoga, vizualiza şi înţelege.

Această teză de doctorat contribuie la îmbogăţirea metodelor de determinare automată a similarităţii secvenţelor biologice cu propunerea unei noi abordări folosind o strategie statistică. Ea oferă o cale efectivă de capturare a caracteristicilor comune ale secvenţelor comparate evitând dificultăţile întâmpinate de utilizatori în practicile actuale. Performanţa ridicată şi eficienţa computaţională fac din această nouă abordare o alternativă promiţătoare la algoritmii cunoscuti în prezent.

CUPRINS

Abreviar

ADN- aciddezoxiribonucleic
NLA-Normalized Local Alignment
ROC- Receiver Operating Characteristic
DM-Data Mining
SSE-Secundary Structure Elements
PSI-Protein Structure Index
GO-Geene Ontology
DAG- Dyrected Acyclic Graph
USM-Universal Similarity Metric
SVM- Support Vector Machine
LSA-Latent Semantic Analysis
SVD-Singular Value Decomposition
*k*NN- *k* Nearest Neighbour
IMM-Interpolated Markov Models
SMM-Selective Markov Models
NNets-Neural Networks
BNN-Baiesian Neural Networks
LCC-Linear Corelation Coeficient

Lista de tabele

Lista de figuri

INTRODUCERE

Cercetările comunităţii în inteligenţă bioinformatică sunt focalizate pe diverşi algoritmi de învăţare, sisteme de data mining şi sisteme integrate care să permită transformarea secvenţelor biologice, a observaţiilor şi cunoştinţelor în informaţie structurată şi semnificativă pe care biologii o pot interoga, vizualiza şi înţelege. Dintre cele mai importante funcţii computaţionale identificate în acest domeniu de cercetare sunt: identificarea genelor/proteinelor şi clasificarea lor în categorii; metode de comparare a secvenţelor biologice la diferite nivele de detaliu; identificarea regiunilor regulatorii şi a modelelor noi în secvenţa de date (căutarea regiunilor care se repetă, similarităţi şi modele rare care au semnificaţie biologică). Pe baza acestor teme, oamenii de ştiinţă încearcă să dezvolte tehnici specializate şi un mare volum al cercetării este orientat spre algoritmi destinaţi prelucrării informaţiei conţinute în depozite mari de secvenţe. Analiza şi compararea automată a secvenţelor biologice a beneficiat în ultimul deceniu de un efort considerabil din partea cercetării informatice reuşind să substituie cu succes multe din necesităţile impuse de extinderea rapidă a cantităţii de cunoştinţe biologice stocate sub forma bazelor de date de secvenţe.

Pentru căutarea în baze de secvenţe cele mai frecvent folosite metode folosesc algoritmi de determinare a similarităţii secvenţelor care adesea sunt reprezentaţi de proceduri anevoioase şi costisitoare. Astfel, prin această teză de doctorat se contribuie la îmbogăţirea metodelor de determinare automată a similarităţii secvenţelor cu propunerea unei noi abordări. Prin urmare, utilizând o strategie statistică, se oferă o cale efectivă de capturare a caracteristicilor comune ale secvenţelor comparate evitând misiunea mai putin plăcută a utilizatorilor de a fi puşi în dificultate în alegerea de parametri, funcţii adiţionale, matrice de substituţie sau diverse metode de evaluare.

Metoda a fost inspirată de folosirea cu succes a conceptului de entropie pentru domeniul *information retrieval* (tradus liber „regăsirea informaţiei") în modelarea statistică a limbajului pentru sistemele automate de recunoaştere a vorbirii. Performanţa ridicată şi caracterul facil de implementare, împreună cu eficienţa sa computaţională fac din această abordare o alternativă promiţătoare la algoritmii cunoscuţi şi sofisticaţi de comparare şi analiză a secvenţelor.

Aparte de contextul biologic, noua metodă poate fi aplicată şi pentru studiul general al secvenţelor mari de text dar un aspect esenţial îl va reprezenta limitarea impusă de mărimea alfabetului considerat. Astfel, ea îşi poate găsi deocamdată aplicabiliteatea în domeniul biologic unde alfabetul este fie de 4 litere cand este vorba de nucleotidele care formează secvenţe de ADN/ARN fie de 20 dacă sunt vizate secvenţele de proteine, definite printr-un alfabet de 20 amino acizi.

Teza, deservind domeniul biologiei, are un evident character interdisciplinar: biologie, automatizări, matematică, informatică şi calculatoare. Ea însă poate fi mai puţin atribuită doar domeniului biologiei, matematicii, informaticii sau calculatoarelor, fiind cel mai bine încadrată în sfera mai largă a automatizărilor. Această încadrare este motivată prin urmărirea realizării scopului domeniului căreia îi este dedicată fără intervenţia umană („automatos" însemnând „funcţionare de la sine" în limba greacă). Aşa cum automatica circumscrie automatizarea celor mai diferite procese – tehnice, energetice, militare etc.-, tot aşa şi preocupările cuprinse

în teză - analiza automată a similarităţilor secvenţelor biologice- urmăresc realizarea scopului în mod automat, ceea ce o poate înscrie in domeniul specializării de "Automatică", servindu-se de suportul teoretic specific al acesteia.

Teza este structurată în şapte capitole după cum urmează:
În *capitolul 1* se face o introducere a cititorului în domeniul secvenţelor biologice subliniind aspectele necesare înţelegerii modului de reprezentare şi analiză al acestora. Prin această introducere se vine în sprijinul capitolelor următoare pentru a ilustra şi a motiva dezvoltarea metodelor şi algoritmilor pentru sisteme de calcul destinate analizei şi comparării automate a secvenţelor biologice.

Pe parcursul *capitolului 2* este realizată o investigare a metodelor de similaritate a secvenţelor biologice deja cunoscute, urmată de o analiză a stării actuale în dezvoltarea de noi metode de similaritate. Deoarece aliniamentul secvenţelor este cea mai folosită strategie în analiza şi determinarea similarităţii sau disimilarităţii dintre secvenţe sunt evidenţiate cele mai semnificative metode folosite în prezent.

Capitolul 3 conţine o multitudine de tehnici de recunoaştere care şi-au găsit aplicabilitate directă în analiza secvenţelor biologice. Ele au fost întâlnite în procesul de identificare a genelor/proteinelor şi clasificarea lor în categorii, în compararea secvenţelor, identificarea regiunilor regulatorii şi tiparelor noi în date secvenţiale, fiind valorificate în produse comerciale specializate.

Ca o continuitate în utilizarea tehnicilor de recunoaştere, în *capitolul 4* este realizată o investigare a potenţialului lingvistic din perspectiva modelării statistice pentru sistemele de recunoaştere a vorbirii. Astfel, se descriu premisele teoretice de la care porneşte investigarea şi analiza datelor reprezentate textual. De asemenea, se descrie modul de analiză şi rezultatele obţinute cu un set de programe experimentale implementate pentru acest scop. Aceste rezultate se vor constitui ca bază de pornire în promovarea ideii de recunoaştere a conţinutului unui text pe baza evaluării cu ajutorul măsurilor derivate din conceptul de entropie.

În *capitolul 5* se propune o nouă metodă de evaluare a similarităţii secvenţelor biologice inspirată din rezultatele experimetale descrise în capitolul anterior. Prin urmare, cu scopul identificării secvenţelor similare din baze de secvenţe biologice este propusă noua metodă de comparare. Ea este aplicată secvenţelor de proteine făcând uz de principiul de evaluare al entropiei folosind modele Markov lingvistice. Pentru testarea şi validarea acesteia au fost propuse două experimente ale căror rezultate sunt analizate şi interpretate.

Capitolul 6 este destinat comparării rezultatelor obţinute aplicând noua strategie de determinare a similarităţii secvenţelor cu cele ale unei alte metode larg utilizate. Astfel, a fost ales programul ClustalW şi vis-a-vis de rezultatele obţinute este făcută o analiză comparativă.

Întreg *capitolul 7* este destinat concluziilor finale subliniind aspectele esenţiale obţinute cu noua metodă precum şi partea de contribuţii personale şi posibile direcţii de dezvoltare.

1. SECVENŢE BIOLOGICE DIN PERSPECTIVA ABORDĂRII BIOINFORMATICE

1.1. Definirea, organizarea şi reprezentarea secvenţelor biologice

Faţa biologiei a fost schimbată de urgenţele geneticii moleculare. Printre cele mai apreciate realizări în direcţia susţinerii progresului în domeniul biologic sunt eforturile secvenţierii la scară largă a părţilor de ADN (cum ar fi proiectul "Human Genome Project") care produce o cantitate imensă de date. Nevoia de înţelegere a datelor a devenit însă şi mai presantă. Cererile pentru analize sofisticate ale secvenţelor biologice conduc înainte domeniul nou creat şi într-o continuă expansiune al cercetării în biologie moleculară, computaţională sau bioinformatică.

Pentru a avea o imagine generală asupra a ceea ce reprezintă secvenţele biologice şi modul lor de reprezentare şi gestionare, în acest capitol sunt descrise noţiuni necesare înţelegerii domeniului abordat pe parcursul acestei lucări.

Secvenţa de ADN codifică informaţia necesară pentru vieţuitoare de a trăi şi a se reproduce. Determinarea secvenţei este prin urmare, utilă în cercetare pentru a şti de ce şi cum trăiesc organismele. Datorită caltăţii sale de *cheie a vieţii*, cunoştinţele despre ADN devin practic folositoare în orice domeniul de cercetare biologică.

În genetică şi biochimie procesul de secvenţiere presupune determinarea structurii primare (sau secvenţei primare) a unui fragment de ADN. Mai exact, în terminologia genetică, secvenţierea ADN este procesul prin care se determină ordinea nucleotidelor unui fragment de ADN dat. Astfel, se determină structura primară a secvenţei respective. În mod curent, apropape toate secvenţierile de ADN sunt executate utilizând tehnica dezvoltată de Frederick Sanger ce presupune utilizarea terminaţiei secvenţei specifice a unei reacţii in vitro a sintezei ADN folosind substraturi modificate de nucleotide[1] [WIKI'05b].

Prin urmare, în urma secvenţierii se obţine o înşiruire de simboluri de forma :

```
1   ggatccccta cccagctggg accctcccca acccccttct tcctcaccct ctgcacgaga
61  acgaagttct tcgtcaccag ctttttggta aggatggagg aggggaggcg tacgtgaggg
...
```

unde, fiecare simbol reprezintă una dintre cele patru nucleotide, după cum este ilustrat în figura 1.1-1.

---

[1] Nucleotidele sunt unităţi structurale ale ARN, AND şi câţiva cofactori. În celulă ei joacă roluri importante în producerea de energie, metabolism şi semnalizare [WIKI'05a].

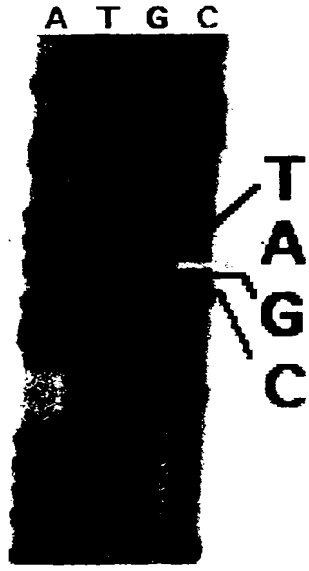


Figura 1.1- 1. Parte a unui gel de secvenţiere marcat radioactiv

Fiecare nucleotidă conţine o moleculă de fosfat, una de zahar (dezoxiriboza) şi una dintre cele patru molecule denumite baze (adenina (A), guanina (G), citozina (C) sau timina(T)). Seria acestor patru molecule determină codul genetic.

Anumite segmente de ADN care conţin instrucţiunile pentru producerea proteinelor specifice organismului se numesc *gene*. Genele sunt situate în anumite segmente de-a lungul ADN-ului uman, împachetate bine în structuri numite cromozomi (oamenii de ştiinţă apreciază că ADN-ul uman conţine circa 30000 gene). Unele dintre ele sunt responsabile de formarea proteinelor structurale care determină în final trăsăturile fizice, cum ar fi ochii căprui sau parul ondulat al unei persoane. Altele, oferă instrucţiuni pentru ca organismul să producă substanţe chimice importante, numite enzime. În funcţie de codurile unei anumite gene, chiar o mică eroare în structura ADN-ului poate însemna probleme grave pentru întregul organism. Spre exemplu, uneori, o eroare într-o singură genă poate duce la scurtarea vieţii sau la deficienţe fizice. Anumite probleme genetice sunt cauzate de o singură genă, care este prezentă, dar alterată. Unele modificari ale genelor se numesc *mutaţii.* Pentru a detecta gena defectă, specialiştii folosesc tehnici de "screening" sofisticate.

1.1.1. Sinteza proteinelor

Prima menţionare a cuvantului "proteina", care înseamnă în limba greacă "de prim rang", este atestată de o scrisoare trimisă de Jöns Jakob Berzelius (medic şi chimist, 1779-1848) lui Gerhardus Johannes Mulder (biochimist, 1802-1880) pe 10 iulie 1838.

Aşa după cum s-a mai menţionat, ADN-ul poartă instrucţiunile pentru producerea de proteine. O secvenţă de trei baze nucleotide, numite triplet sau

codon, este codul genetic care specifică un anumit amino acid. De exemplu, un triplet GAC (guanina, adenina şi citozina) este codul genetic pentru amino acidul *leucina* iar un triplet CAG (citosina, adenina şi guanina) este codul genetic pentru amino acidul *valina*. O proteină este compusă din mai multe molecule mici numite amino acizi şi structura şi funcţia proteinei este determinată de secvenţa acestora. Secvenţa amino acizilor este la randul ei determinată de succesiunea bazelor nucleotide în ADN. Proteinele, constituite sub formă de lanţuri de amino acizi se pliază[2] într-o structură unică 3-dimensională. Forma în care se pliază în mod natural o proteină este cunoscută ca stare nativă şi este determinată de secvenţa de amino acizi. Biochimiştii se referă la patru aspecte distincte ale structurii proteinelor. Acestea sunt: structura primară, secundară, terţiară şi quaternară.

1. **Structura primară,** reprezintă secvenţa formată din înşiruirea de amino acizi.

Exemplu de secvenţă de proteină descrisă prin structura sa primară:

```
1 madaldlapl laagwshdpe rgvlsktfgf etfvaafgfm trvalwaekw nhhpdwsnsy
61 gtvevsltsh daggIterdl klarkidela aa
```

(Rhodobacter sphaeroides 2.4.1, din baza de proteine NCBI, nr de acces: ABA80080.)

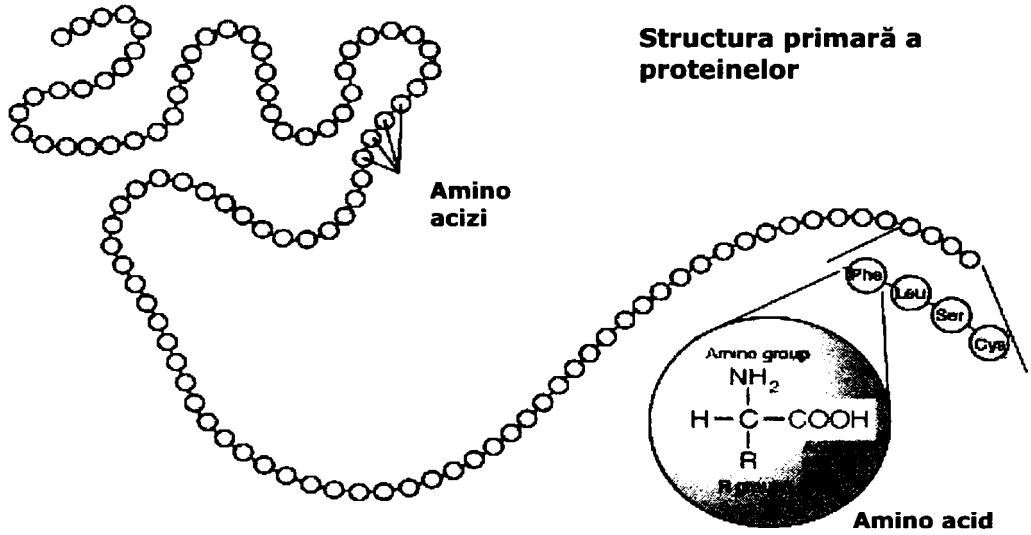


Figura 1.1.1-1. O reprezentare sugestivă a structurii primare.

[2] Plierea proteinelor este procesul prin care structura unei proteine îşi asumă forma funcţională sau conformaţia. Toate moleculele proteinei sunt lanţuri de amino-acizi ramificate heterogen. Prin spiralare şi plieri într-o formă 3D specifică ele sunt în stare să îşi execute funcţia biologică [WIKI'05c].

2. **Structura secundară**, descrie forma generală (3D) a regiunilor locale. Ea poate include regiuni ale helixului alfa (alfa helixes), fâşii beta (beta sheets), răsuciri (turns) şi spire aletoare (random coil) sau ceva mai puţine structuri comune.

Structura de alfa-helix. Peptidele[3] sunt răsucite în jurul unui cilindru imaginar şi stabilizate de legăturile de hidrogen.

Fâşiile plate beta. Amino acizii adoptă conformaţia unei foi de hârtie iar structura este stabilizată de legături de hidrogen în diferite fâşii de polipeptide. Se observă că unele dintre fâşii sunt paralele şi altele antiparalele.

Alte părţi ale structurii nu sunt foarte stabile şi adoptă o formă răsucită aleator.

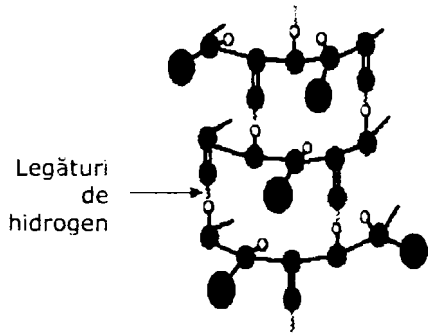


Figura 1.1.1-2. Reprezentarea simbolică a structurii de alfa-helix

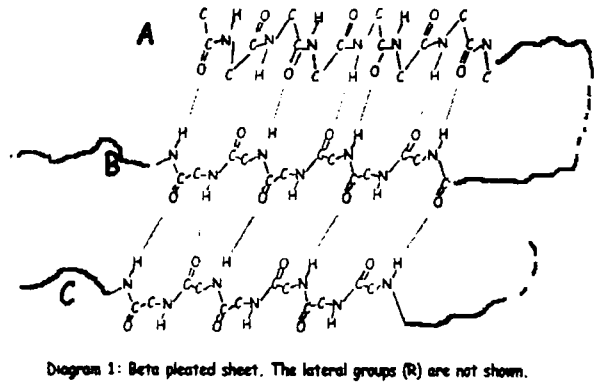


Diagram 1: Beta pleated sheet. The lateral groups (R) are not shown.

Figura 1.1.1-3. Reprezentarea simbolică a fâşiilor plate beta

[3] Doi sau mai mulţi amino acizi legaţi printr-o legătură numită "legătură peptidică" (peptide bond). O proteină fiind formată din amino acizi legaţi în acest mod este uneori referită ca polipeptidă. Unele proteine conţin mai mult decât un singur lanţ de polipeptide [WIKI'05d].

Figura 1.1.1-4. Reprezintă structura 3D a unei proteine (Myoglobin). Alfa helix-urile sunt prezentate colorate, rasucirile aleatoare în alb, nu există fâşii beta (beta sheets).

3. **Structura terţiară,** reprezintă forma generală a unei singure molecule de proteine; relaţia spaţială a motivelor[4] din structura secundară. Conventional, structura terţiară este dedusă prin cristalografie sau rezonanţa magnetică nucleară. Domeniul de studiu al structurii terţiare al proteinei este cunoscut ca *biologie structurală*. Ea poate fi considerată drept împachetarea în structura 3D a unui alt aspect al lanţului (side chain packing in the 3D structure).

[4] Într-o moleculă biologică reprezentată ca un lanţ, neramificată, cum ar fi o proteină sau o fâşie de ARN, un motiv structural este un element structural 3D sau o pliere în cadrul lanţului care apare de asemenea într-o varietate de alte molecule. În contextul proteinelor, termenul este adesea folosit pentru domeniul structural, cu toate că un domeniu nu trebuie să fie un motiv şi în cazul în care conţine un motiv, nu trebuie să fie compus doar dintr-unul singur.

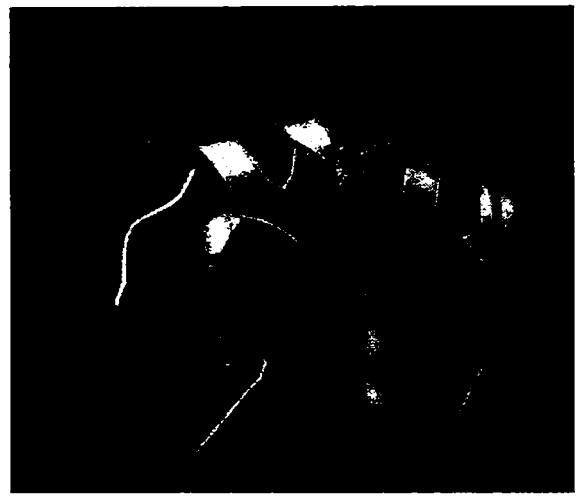


Figura 1.1.1-5. Reprezentarea 3D a unei proteine conţinând structuri mai complexe

Amino acizii care sunt foarte distanţi unul faţă de altul în structura primară pot fi apropiaţi în cea terţiară datorită plierii lanţului şi stabilizării lui prin interacţiuni ionice, legături de hidrogen, forţe de dispersie (Van Der Waals), poduri de sulfuri (*sulphur bridges*)

Notă : Unul din scopurile bioinformaticii este de a prezice conformaţia nativă a unei proteine din secvenţa sa primară.

4. **Structura quaternară,** reprezintă forma sau structura care rezultă din unificarea mai multor molecule de proteine care în acest context funcţionează ca parte a unui ansamblu mai mare de proteine sau complex de proteine. Cu alte cuvinte, această structură este aranjamentul subgrupurilor de polipeptide în cadrul proteinelor complexe compuse din două sau mai multe subunităţi. Este de menţionat faptul că numai proteinele cu mai mult de un singur lanţ au o structură quaternară.

Un exemplu de reprezentare în care este evidenţiată structura quaternară poate fi observat în figura 1.1.1- 6.

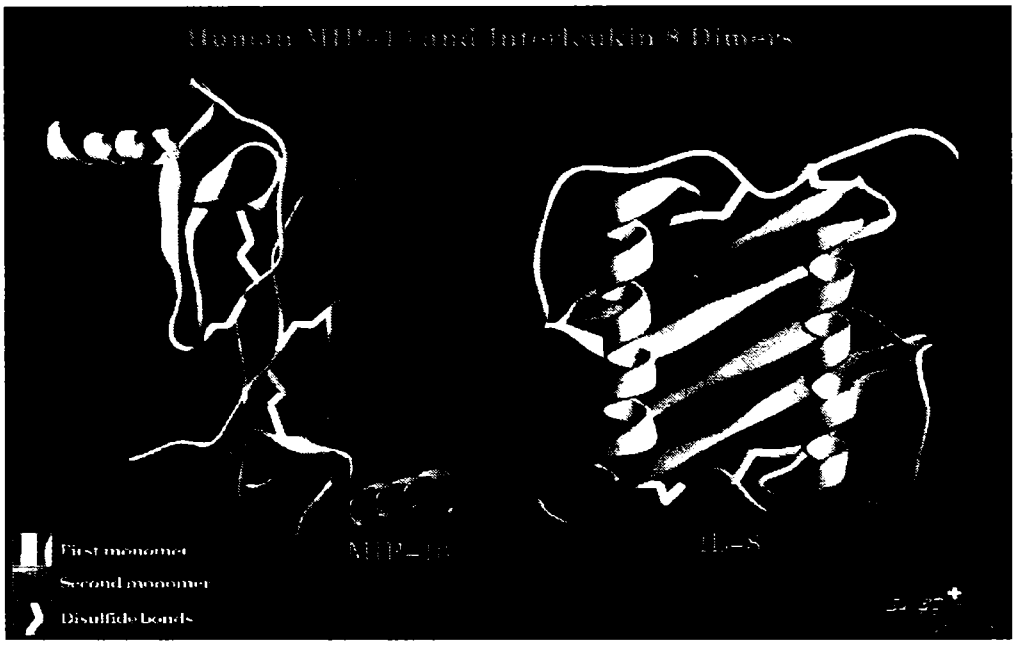


Figura 1.1.1- 6. Reprezentare a structurii quaternare

Proteinele sunt implicate practic în fiecare funcţie executată de o celulă. Prin inginerie genetică cercetătorii pot altera secvenţa şi prin umare structura proteinei. Dintre efectele deficienţelor în proteine sunt simptomele de oboseală, rezistenţa la insulină, pierderea părului, a pigmenţilor din păr, a masei musculare, reducerea temperaturii corporale şi neregularităţi în funcţiile hormonale. De asemenea şi excesul poate crea probleme cum ar fi supra reacţia sistemului imunitar, disfuncţii ale ficatului datorate excesului de reziduuri, scurtarea oaselor datorată gradului crescut de aciditate în sânge.

Notă : Imaginile sunt preluate din materialul biologic documentar realizat pentru proiectul EU Biopattern: Computational Intelligence for biopattern analysis in Support of eHealthcare, Network of Excellence Project No. 508803.

1.1.2. Clasificarea structurală a proteinelor

Conform celor menţionate în [MURZ'95], aproape toate proteinele au similarităţi structurale cu alte proteine şi în multe cazuri împart o origine evoluţionară comună. Cunoaşterea acestor relaţii aduce contribuţii importante biologiei moleculare şi altor domenii de ştiinţă înrudite. Înţelegerea structurii şi evoluţiei proteinelor este o problemă centrală care joacă un rol important în interpretarea secvenţelor produse de proiectele genomice, şi prin urmare, înţelegerea evoluţiei şi dezvoltării.

Având în vedere creşterea exponenţială în numărul proteinelor ale căror structuri au fost determinate prin cristalografie cu raze X şi spectroscopie, se remarcă existenţa unui corpus larg de informaţie disponibilă şi care creşte rapid. Structurile proteinelor pot fi clasificate într-o varietate de moduri înrudite între ele: similaritate funcţională, similaritate evoluţionară şi similaritate a modului de pliere (fold similarity). Cîteva dintre regulile luate în considerare sunt bazate pe următoarele aspecte:

1. Clasificarea familiilor de proteine la modul general este definită de *funcţia lor* care provine din biochimia experimentală.
2. Similaritatea funcţională şi clasificarea familiilor poate fi dedusă din similaritatea secvenţelor.
3. Dacă există o identitate a secvenţelor mai mare de 25% se presupune o oarecare omologie a structurii. Spre exemplu, lanţurile A şi B ale hemoglobinei au o omologie de 45%, iar diferit de micile inserări ele au o structură 3D identică.
4. Există o distribuţie Gauss a structurilor similare cu un maxim la o valoare în jur de 9% identitate a secvenţelor. Aceasta înseamnă că similaritatea secvenţelor poate exista şi nu poate fi detectată prin metodele standard de comparare a secvenţelor. Evoluţia divergentă poate merge o cale lungă şi încă să păstreze plierea.
5. Omologia structurii, singură, inferă un tip de aranjamete din punct de vedere energetic favorabile dar nu neaparat o relaţie funcţională –evoluţie convergentă.
6. Este foarte dificil să se distingă evoluţia divergentă de cea convergentă.
7. S-au depus eforturi pentru a clasifica proteinele pe baza structurii.

Pentru a facilita înţelegerea şi accesul la această informaţie a fost construită baza de proteine SCOP[5], care furnizează descrierea detaliată şi cuprinzătoare a relaţiilor evoluţionare şi structurale ale proteinelor a căror structură tridimensională a fost determinată. Metoda folosită la clasificare este în esenţă bazată pe inspectarea vizuală şi compararea structurilor cu ajutorul diverselor programe specializate. Clasificarea este făcută manual, pe nivele ierarhice, după cum urmează.

Familia conţine un grup de proteine înrudite evoluţionar. În primul rând toate proteinele care au o identitate reziduală de 30% şi mai mare, apoi proteinele cu o identitate mai mică a secvenţelor dar ale căror funcţii şi structuri sunt foarte simiare.

Superfamilia este constituită din familii ale căror proteine au o identitate a secvenţelor scăzută dar ale căror structuri şi, în multe cazuri, caracteristici funcţionale sugerează o probabilă origine evoluţionară comună.

Plierea comună. Superfamiliile şi familiile sunt definite ca având o pliere comună dacă proteinele lor au aceeaşi structură secundară majoră în acelaşi aranjament cu aceleaşi conexiuni topologice.

[5] SCOP: Structural Classification of Proteins. http://scop.berkeley.edu/

Clasa. Pentru comoditatea utilizatorilor, diferite plieri au fost grupate în clase. Majoritatra plierilor sunt asociate la una dintre cele cinci clase structurale pe baza structurilor secundare din care sunt compuse.

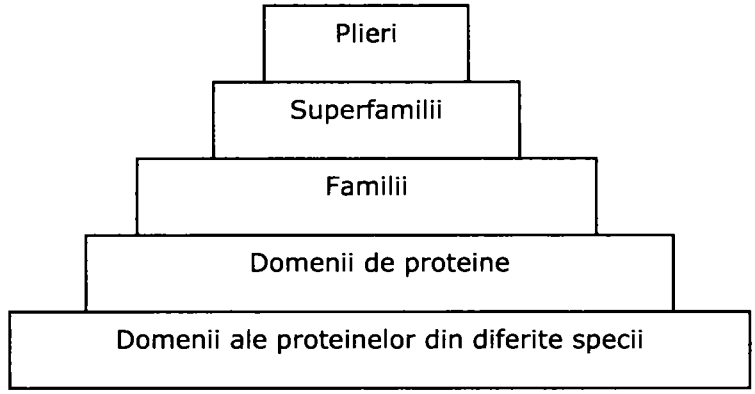


Figura 1.1.2-1 Reprezentarea ierarhiei folosite în clasificarea proteinelor în baza de proteine SCOP. Unitatea de clasificare este de obicei domeniul proteinei.

Exemplu: **cytochrome c2 [a.9.3.1]**

- **Class:** toate alfa proteinele
- **Fold:** Citocrom c
  nucleu: 3 helixuri; fâşie pliată, deschidere
- **Superfamily:** Citocrom c
  hema legată covalent completează nucleul
- **Family:** monodomeniul Citochrom c
- **Protein:** Citocrom c2

Pentru studiul intensiv al secvenţelor biologice, în prezent se estimează că există în jur de 500 de baze de date publice şi comerciale [WIKI'05e]. Acestea conţin informaţii despre secvenţe de nucleotide ale genelor sau secvenţe de amino acizi ale proteinelor. Mai mult, ele stochează informaţii despre funcţia, structura, localizarea în cromozomi, efecte clinice ale mutaţiilor precum şi facilitează găsirea similarităţii secvenţelor biologice. Astfel, pot fi găsite diverse baze de secvenţe după cum vor fi menţionate în subcapitole următoare.

1.1.3. Formate de reprezentare ale secvenţelor biologice

Formatele secvenţelor sunt, la modul simplu, felul în care secvenţele de amino acizi sau ADN sunt înregistrate într-un fişier pe calculator. Programe diferite aşteaptă adesea şi formate diferite. Prin urmare, pentru a executa cu succes o funcţie este important să se înţeleagă cum arată diverse formate şi care este structura lor de bază. Formatele utilizate sunt foarte flexibile dar nu pot face faţă la prea multă inconsistenţă.

În general, secvenţele sunt descrise sub formă de text ASCII. Ele sunt un aranjament de caractere impus, simboluri şi cuvinte cheie care specifică modul de reprezentare pentru: secvenţă, nume de identificare, comentarii, etc. şi indicii despre locul unde ar trebui să caute programul pentru a le găsi. În general nu există caractere ascunse neprintabile în vre-un format cunoscut de secvenţe. Toate formatele standard de secvenţe pot fi tipărite sau simplu vizualizate prin afişarea simplă a fişierului/filei care le conţin.

Există cel puţin cateva formate de secvenţe utilizate în prezent la scară largă, unele mai mult altele mai puţin folosite. Ele au fost proiectate astfel încât să fie capabile să reţină secvenţa de date şi alte informaţii suplimentare despre secvenţă. De când au început să se folosească programele software pentru a scrie şi citi secvenţe, aproape fiecare pachet de analiză a secvenţelor şi-a inventat propriul ei format. Majoritatea colecţiilor de secvenţe care îndrăznesc să se numească baze de date şi-au stocat datele în propriul format. O secvenţă nu presupune vre-un tip de identificare dar în mod sigur aceasta ar ajuta. Cele mai multe formate de secvenţe includ cel puţin o formă de nume de identificare (ID), de obicei plasat undeva la începutul formatului de secvenţă.

Pentru a fi posibilă utilizarea aceluiaşi set de secvenţe în diverse scopuri executate de programe variate s-au implementat convertoare de secveţe dintr-un format în altul. Un asemenea convertor este disponibil la EMBL-EBI (European Bioinformatics Institute http://www.ebi.ac.uk/cgi-bin/readseq.cgi), şi se numeşte "READSEQ", tool de conversie al biosecvenţelor. EMBOSS, The European Molecular Biology Open Source Software Suite, este un pachet de Open Source software de înaltă calitate pentru analiza secvenţelor. Lista completă de formate de secvenţe suportată de aplicaţiile EMBOSS cu descrierea detaliilor poate fi accesată la adresa oficială sau http://www.ebi.ac.uk/clustalw/#...formats, iar unul dintre cele mai cunoscute este formatul Fasta, prezentat în cele ce urmează iar alte câteva exemple sunt menţionate în ANEXA 1.

Formatul Fasta (Pearson) este unul dintre cele mai utillizate pentru secvenţele biologice şi a fost propus de David J. Lipman şi William R. Pearson în 1985 în articolul "Rapid and sensitive protein similarity searches" cu numele de FASTA. Programul original a fost proiectat pentru căutarea similarităţii proteinelor, descris în 1988 ("Improved Programs for Biological Sequence Comparison"). În prezent, în bioinformatică, formatul FASTA este un format de fişier folosit pentru schimbul informaţiei între baze de date cu secvenţe genetice. Aşa după cum este descris mai jos, o secvenţă în format FASTA începe cu o descriere biologică într-un singur rând, urmată de rânduri de secvenţă. Linia de descriere este diferenţiată de secvenţa de date prin simbolul semnului mai mare (">") pe prima poziţie. Cuvântul care urmează acestui simbol este identificatorul secvenţei şi restul liniei este o descriere opţională. De obicei se recomandă să nu fie spaţiu între ">" şi prima literă a identificatorului. De asemenea, este de preferat ca toate liniile de text să fie mai scurte de 80 de caractere. Secvenţele se încheie cand o nouă linie începe cu ">".

Un exemplu de format FASTA :

>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPWGQMSFWGATVITNLFSAIPYI
GTNLV

EWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPYYTI
KDFLG
LLILILLLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVPNKLGGVLALFLSI
VIL
GLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEYPYTIIGQMASILYFSIILAFLP
IAGX
IENY

Sau o altă variantă exprimată la modul general:

>SEQUENCE_1
;comment line 1 (optional)
MTEITAAMVKELRESTGAGMMDCKNALSETNGDFDKAVQLLREKGLGKAAKKADRLAAEG
LVSVKVSDDFTIAAMRPSYLSYEDLDMTFVENEYKALVAELEKENEERRRLKDPNKPEHK
IPQFASRKQLSDAILKEAEEKIKEELKAQGKPEKIWDNIIPGKMNSFIADNSQLDSKLTL
MGQFYVMDDKKTVEQVIAEKEKEFGGKIKIVEFICFEVGEGLEKKTEDFAAEVAAQL
>SEQUENCE_2
;comment line 1 (optional)
;comment line 2 (optional)
SATVSEINSETDFVAKNDQFIALTKDTTAHIQSNSLQSVEELHSSTINGVKFEEYLKSQI
ATIGENLVVRRFATLKAGANGVVNGYIHTNGRVGVVIAAACDSAEVASKSRDLLRQICMH

Institutul Naţional de Cercetări în Biotehnologie al SUA a încercat să definească un standard pentru antetul secvenţelor stocate. Oricum, nu se oferă o descriere definitivă a acestui format FASTA dar o încercare de a se crea un asemenea format este urmatoarea, unde pentru fiecare bază de date se propune o codificare :

GenBank	gi|*gi-number*|gb|*accession*|*locus*
EMBL Data Library	gi|*gi-number*|emb|*accession*|*locus*
DDBJ, DNA Database of Japan	gi|*gi-number*|dbj|*accession*|*locus*
NBRF PIR	pir||*entry*
Protein Research Foundation	prf||*name*
SWISS-PROT	sp|*accession*|*name*
Brookhaven Protein Data Bank (1)	pdb|*entry*|*chain*
Brookhaven Protein Data Bank (2)	*entry*:*chain*|PDBID|CHAIN|SEQUENCE
Patents	pat|country|number
GenInfo Backbone Id	bbs|number
General database identifier	gnl|database|identifier
NCBI Reference Sequence	ref|accession|locus
Local Sequence identifier	lcl|identifier

Secvenţele se aşteaptă să fie reprezentate în standard-ul IUB/IUPAC (International Union of Biochemistry (IUB)/ Union of Pure and Applied Chemistry (IUPAC)) pentru amino acizi şi coduri pentru acizi nucleici, cu următoarele excepţii: literele mici sunt acceptate şi transformate în litere mari, o singură linioară poate fi folosită pentru a reprezenta un character lipsă (gap character) şi în secvenţele de amino acizi "U" şi "*" sunt caractere acceptabile (a se vedea în cele două tabele din ANEXA 1). Înainte de a începe o interogare, orice cifră din secvenţa de căutare (query sequence) trebuie fie înlăurată fie înlocuită de codul de literă corespunzator

(e.g. N pentru acid nucleic necunoscut sau X pentru reziduu de amino acid necunoscut).

1.1.4. Baze de date de secvenţe

Secvenţele biologice rezultate în urma procesului de secvenţiere sunt stocate în diverse baze de date respectând cerinţele de reprezentare şi organizare proprii fiecărei baze. Astfel, există o varietare de „depozite" de secvenţe. În general ele sunt organizate în baze de nucleotide, baze de proteine sau baze specializate fie pe o anumită genă fie pe o anumită specie sau proces evolutiv. Câteva dintre cele mai cunoscute vor fi menţionate în continuare.

Pentru nucleotide:

a)EMBL Nucleotide DB, European Molecular Biology Laboratory. Baza de nucleotide (cunoscută ca EMBL-Bank) constituie resursa europeană principală de secvenţe. Sursele principale de ADN şi ARN sunt obţinute în urma unor cercetări individuale, proiecte de secvenţiere genomică sau aplicaţii autorizate. (http://www.ebi.ac.uk/emdl/index.html.)

b)Genbank (National Center for Biotechnology Information), este o bază de date a Institutului National de Certătari în Biotehnologie din SUA care serveşte ca arhivă pentru toate secvenţele publice de ADN pentru un numar mai mare de 100,000 de organisme diferite. (http://www.ncbi.nih.gov/Genbank/index.html.)

c)DDBJ (DNA Database of Japan). Aceasta este singura bancă de ADN din Japonia care este certificată oficial să colecteze secvenţe de ADN de la cercetători şi de a genera numere de acces internaţional recunoscute. Există un schimb de date cu EMBL/EBI si Gen Bank/NCBI. (http://www.ddbj.nig.ac.jp/Welcome-e.html.)

d)Entrez este baza de nucleotide care colecţionează secvenţe din diferite surse incluzând GenBank, RefSeq, şi PDB. Numărul de componente continuă creşterea cu o rată exponenţială.
(http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Nucleotide)

Pentru proteine :

a)Swiss-Prot, este o bază de secvenţe de proteine care încearcă să furnizeze un nivel ridicat de adnotări (cum ar fi descrierea funcţiei proteinei, structura domeniului ei, modificarile post translaţionale, variante, etc), un nivel minim de exces şi un nivel ridicat de integrare cu al altor baze de date de secvenţe. (http://us.expasy.org/sprot/)

b)PIR-PSD (Protein Information Resource – Protein Sequence Database), este o bază adnotată ce conţine peste 283 000 de secvenţe acoperind întreaga arie taxonomică a proteinelor. International Protein Sequence Database-PIR, conţine informaţii privind toate tipurile de proteine care apar natural şi ale căror secvenţe sunt cunoscute. Un obiectiv major al bazelor de secvenţe este de a furniza date cuprinzătoare, non redundante, organizate unic pe baza omologiei şi taxonomiei. (http://pir.georgetown.edu/pirwww/search/textpsd.shtml.)

c)NCBI (National Center for Biotechnology Information), cuprinde date compilate din diverse surse incluzand: Swiss-Prot, Protein Information Resource (PIR),Protein Data Bank (PDB), Protein Resource Foundation (PRF) in Japan. (http://www.ncbi.nih.gov/)

d)Uniprot (Universal Protein Resource), este cel mai cuprinzător catalog de informaţii despre proteine. Este un depozit central de secvenţe de proteine şi funcţii creat unind informaţia conţinută în alte baze de proteine: Swiss-Prot, TrEMBL, and PIR. (http://www.uniprot.org)

e)SCOP (Structural Classification of Proteins) îşi doreşte să ofere o descriere detaliată şi cuprinzătoare a relaţiilor structurale şi evoluţionare dintre proteine a căror structură 3D este cunoscută.(http://scop.berkeley.edu/)

Baze specializate:

Profilate pe genele producătoare de cancer:
-CGAP Cancer Genes **,** http://cgap.nci.nih.gov/Genes/GeneFinder,
-Tumor Gene Database, http://condor.bcm.tmc.edu/ermb/tgdb/tgdf.html.
-Entrez-Cancer Cromosomes,
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=cancerchromosomes.
-OMIM (Online Mendelian Inheritance in Man),este o bază de date de gene umane, tratamente genetice şi dezordini ereditare disponibile la National Center for Biotechnology Information (NCBI) şi care pot fi accesate pe web.
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=OMIM.
- LSDBs (Locus Specific Database),
http://www.genomic.unimelb.edu.au/mdi/dblist/glsdb.html.
-P53 Databases, International Agency for Research on Cancer (Lyon), France (IARC), http://www.iarc.fr/p53/, TP53 este o genă responsabilă pentru tumoare.
- BRCA1&2 database, http://research.nhgri.nih.gov/bic/.

Baze de secvenţe cu mutaţii:
- **H**uman **G**ene **M**utation **D**atabase (**HGMD**),
http://archive.uwcm.ac.uk/uwcm/mg/hgmd0.html
- **H**uman **G**enome **V**ariation database (**HGV**), http://hgvbase.cgb.ki.se/

Baze de secvenţe profilate pe o anumită specie:
- **GDB** (Human Genome Organization), http://wehih.wehi.edu.au/gdb/

1.2. Elemente în analiza secvenţelor biologice

Pentru înţelegrea modului de relaţionare dintre secvenţele biologice, în cele ce urmează vor fi descrise pe scurt elementele de interes major.

1.2.1. Omologia secvenţelor în genetică

În genetică este folosită omologia cu referire la secvenţele de proteine sau ADN, însemnând că secvenţele respective au strămoşi comuni. Omologia secvenţelor poate indica de asemenea o funcţie comună iar din punct de vedere al gradelor de comparaţie ea este o calitate de forma totul-sau-nimic; nu există grade de omologie. Regiunile de secvenţe care sunt omoloage pot fi numite regiuni conservate, consensus sau canonice şi reprezintă cea mai comună alegere a bazei sau amino acidului la fiecare poziţie.

Omologia proteinelor şi ADN-ului este adesea concluzionată pe baza similarităţii secvenţelor, şi asta în special în bioinformatică. Spre exemplu, în general, dacă două sau mai multe secvenţe au o secventa de ADN aproape identică este probabil ca ele să fie omoloage. Oricum este posibil ca similaritatea secvenţelor să nu provină din apartenenţa la un ancestor comun şi secvenţele scurte pot fi similare accidental sau secvenţele pot fi similare deoarece ambele au fost selectate pentru a fi legate la o anume proteină cum ar fi factorul de transcripţie[6]; asemenea secvenţe sunt similare dar nu omoloage. Există mulţi algoritmi pentru a clasifica secvenţele de proteine în familii de secvenţe care sunt mulţimi mutual omoloage.

1.2.2. Motiv al unei secvenţe

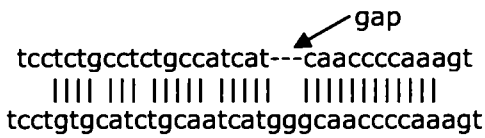
Motiv al unei secvenţe este numită o secvenţă model de nucleotide sau amino acizi care este destul de răspândită şi are sau se presupune a avea o semnificaţie biologică. Motivele au fost descoperite prin studierea genelor similare la diferite specii.

În cadrul unei secvenţe sau bază de date de secvenţe, cercetătorii caută şi găsesc motivele folosind tehnici bazate pe utilizarea calculatoarelor. Asemenea tehnici aparţin disciplinei numite bioinformatică.

[6] Transcripţia (in genetică) este procesul prin care o secvenţă de ADN este copiată din punct de vedere enzimatic de către un ARN polimerizat pentru a produce un ARN complementar. Sau, în alte cuvinte, transferul informaţiei genetice de la ADN la ARN. În cazul proteinelor, codificarea ADN, transcripţia este începutul procesului care în mod ultimativ conduce la translatarea codului genetic în proteine sau peptide funcţionale.

1.2.3. Aliniament

Respectând definirile din [WIKI'05f], aliniamentul secvenţelor este un aranjament de două sau mai multe secvenţe, subliniind similaritatea lor. Secvenţele sunt presărate cu gap-uri (care sunt amino acizi sau secvenţe de amino acizi lipsă şi sunt de obicei marcate cu linii) astfel că oricând este posibil, coloanele rezultate în urma alinierii conţin caractere identice sau similare din secvenţele implicate:



De obicei se studiază evoluţia secvenţelor care provin de la un ancestor comun, în special secvenţele biologice cum ar fi cele de proteine sau ADN. Nepotrivirile din aliniament corespund mutaţiilor iar gap-urile corespund inserărilor sau ştergerilor. Aliniamentul secvenţelor poate fi folosit pentru studiul evoluţiei limbajului sau similarităţii dintre texte. Termenul de aliniament al secvenţelor poate de asemenea să se refere la procesul de construire al unui asemenea aliniament sau de căutare a unor aliniamente semnificative într-o bază de secvenţe posibil neânrudite.

Aliniament pentru perechi de secvenţe

Metodele de aliniament ale perechilor de secvenţe sunt preocupate de găsirea intervalelor de aliniament local sau global ale secvenţelor de proteine (amino acizi) sau ADN (acizi nucleici). În mod frecvent, scopul acestora este de a găsi omologii sau înrudiri ale unei gene sau produs al unei gene într-o bază de date de exemple cunoscute. Această informaţie este utilă în a răspunde unei mari varietăţi de întrebări biologice. Cea mai importantă aplicaţie a intervalului de aliniament este identificarea secvenţelor cu structuri sau funcţii necunoscute. O altă utilizare importantă o reprezintă studiul evoluţiei moleculare.

Aliniamentul multiplu

Aliniamentul multiplu este o extensie a aliniamentului perechilor de intervale pentru a incorpora mai mult de două secvenţe. Metodele de aliniament multiplu încearcă să alinieze toate secvenţele dintr-un set specificat. Aliniamentele ajută la identificarea regiunilor comune dintre secvenţe. Există câteva abordări pentru crearea aliniamentelor multiple de secvenţe, una dintre cele mai populare fiind strategia de aliniament progresiv utilizată de familia de programe Clustal (a se vedea în capitolul 3-5).

Notă: Aliniamentul multiplu al secvenţelor este o problemă dificilă din punct de vedere computaţional şi este clasificată ca o problemă *NP-hard* (Non-deterministic Polynomial-Time hard).

Aliniamentul global

Aliniamentul global dintre secvenţe este un aliniament în care toate caracterele din secvenţele comparate participă la aliniament. Aliniamentele globale sunt cel mai utile pentru găsirea secvenţelor celor mai înrudite. Deoarece aceste secvenţe sunt de asemenea uşor de identificat prin metode de aliniament local, aliniamentul global este oarecum discriminat ca tehnică.

Aliniamentul local

Metodele de aliniament local caută regiuni înrudite în cadrul secvenţelor – cu alte cuvinte ele pot să consiste dintr-un subset de caractere aflat în fiecare secvenţă. Spre exemplu, poziţiile 20-40 ale unei secvenţe. A pot fi aliniate cu poziţiile 50-70 ale altei secvenţe B.
Aceasta este o tehincă mai flexibilă decât aliniamentul global şi are avantajul că regiuni înrudite care apar într-o ordine diferită în cele două proteine pot fi identificate ca înrudite. Aceasta nu este posibil însă cu o metodă de aliniament global.

Aliniamentul structural

Un mod posibil de a valida aliniamentul secvenţelor este de a utiliza aliniamentele structurale unde este posibil. Aliniamentul structural este o formă de aliniament care
încearcă să stabilească echivalenţe între două sau mai multe structuri de proteine pe baza plierilor lor. Deoarece structura proteinelor este mai conservată în timpul evoluţiei decât secvenţa aşa după cum afirma biologul Cyrus Chothia şi cercetătorul Arthur Lesk în 1986 [LESK'86], aliniamentele structurale pot fi mai de încredere pentru distanţe evoluţionare lungi, când secvenţele diverg aşa de mult încât simpla comparare a secvenţelor nu poate detecta similaritatea lor. Oricum, cu toate că aliniamentele structurale sunt utile, ele sunt complet artifactuale.

1.2.4. Similaritatea secvenţelor biologice

Similaritatea a două sau mai multe secvenţe biologice poate fi privită din perspectiva asemănării funcţionale care la randul ei, este strîns legată de asemănarea structurală a secvenţelor biologice. Una dintre cele mai importante măsuri de similaritate este distanţa editată dintre amino acizii a două secvenţe.
La prima vedere, decizia că două secvenţe sunt similare nu este diferită de decizia conform căreia două şiruri de text sunt similare. O mulţime de metode pentru analiza secvenţelor biologice este, prin urmare, înrădăcinată în ştiinţa calculatoarelor unde există o literatură extensivă în metode de comparare a şirurilor de caractere. Conceptul de aliniament e crucial. Evoluând, secvenţele acumulează inserţii si ştergeri precum şi substituiri. Prin urmare, înainte de a evalua similaritatea a două secvenţe în general se începe prin căutarea unui aliniament plauzibil între ele [DURB'98].

Apropape toate metodele de aliniament găsesc cel mai bun aliniament dintre două şiruri urmând unele scheme/planuri de calcul. Deoarece se doreşte un algoritm de calcul care să ofere cel mai probabil aliniament biologic având scorul cel mai mare, se ia în considerare faptul că evoluţiile biologice au istorii evoluţionare, structuri 3D pliate, şi alte caracteristici care limitează evoluţia secvenţei lor primare. Prin urmare, ca un adaos la mecanica aliniamentului şi a algoritmilor de comparaţie, sistemul de calcul însuşi cere o atenţie separată putând fi foarte complex.

Pe baza similarităţii şi/sau omologiei se construiesc familii de secvenţe care partajează proprietăţi comune. Ele sunt constituite fie prin analiza manuală a specialiştilor fie utilizand metode computaţionale.

1.3. Concluzii

Progresul rapid în cercetare legat de secvenţe genomice/proteomice ale diverselor organisme, motivează nevoia de conceptualizare, analiză şi în final înţelegere a informaţiei stocate de aceste secvenţe. Datorită modului lor de reprezentare textuală s-a deschis calea către analiza asistată de calculator, îmbinând cunoştinţele biologice cu cele matematice şi informatice pentru a eficientiza găsirea de răspunsuri la diverse întrebări biologice.

Dintre multiplele avantajele ale cooperării între aceste domenii ale ştiinţei sunt:

- Păstrarea informaţiei biologice sub forma organizată a bazelor de secvenţe;
- Posibiliteatea de explorare a unei cantităţi imense de informaţii în timp scurt;
- Extragerea automată de caracteristici pe baza comparaţiei secvenţelor, operaţiune de altfel dificilă şi mare consumatoare de timp, prin analiza cu metode biologice sau manuale.

2. STADIUL ACTUAL AL TEHNICILOR DE SIMILARITATE PENTRU SECVENŢELE BIOLOGICE

2.1. Metode de determinare a similarităţii secvenţelor biologice

În definirea noţiunii de similaritate, mai întâi depindem de noţiunile de similaritate intuitive şi umane. Astfel, similaritatea este exact ceea ce credem noi că este [WATE'84]. Pe parcursul acestui capitol se va încerca ajungerea la o noţiune formală a ceea ce determină ceva să fie mai similar cu un lucru decât cu altul.

Considerând secvenţele ACTCCG şi ACACCG, ele par a fi similare. De fapt se observă că simpla înlocuire a lui T cu un A este suficientă pentru a obţine a doua secvenţă. Pentru definirea similarităţii, este utilă mai întâi introducerea noţiunii de „distanţă" dintre două şiruri. Astfel, distanţa dintre două şiuri este zero dacă ele sunt identice şi creşte dacă ele devin mai disimilare [WATE'84].Un mod de a defini distanţa dintre două şiruri este de a se observa cantitatea de schimbări care este necesară de a se aplica unei secvenţe pentru a o obţine pe cealaltă. În cazul în care trebuie să înlocuim T cu A din exemplul anterior, distanţa dintre cele două secvenţe va fi 1 (dacă înlocuirea poate fi considerată unitatea de măsură). Se poate merge mai departe la prezentarea altor modificări cum sunt „inserarea" şi „ştergerea". Inserarea are loc atunci când se inserează una sau mai multe litere în secvenţă (la anumite poziţii) iar ştergerea atunci când ştergem anumite litere aflate în anumite poziţii. Înlocuirea însăşi poate fi considerată ca o ştergere urmată de o inserţie dar în termeni de complexitate computaţională nu creează vreo problemă în a considera înlocuirea drept o unitate de schimb [WATE'84].

2.1.1. Distanţă de editare şi similaritate

Această *distanţă de editare* este definită în general ca numărul minim de schimbări efectuate asupra unei secvenţe pentru a o aduce exact în forma celeilalte. Spre exemplu, şirul ACCTGA devine AGCTA prima dată prin înlocuirea celei de-a doua litere, C, cu G şi apoi ştergând G din a cincea poziţie. Astfel, distanţa de editare dintre cele două şiruri este 2.

În [WATE'76] este făcută următoarea formulare: fie S setul de cuvinte finite dintr-un alfabet, incluzând cuvântul vid şi fie $\tau = \{ T \mid T : S \rightarrow S \}$ un set de transformări care include transformările identice. Interesul este în transformările $T_1, T_2, \ldots, T_k$ din τ astfel încât

$$T_1 \circ T_2 \circ \ldots \circ T_k (a) = b. \qquad (2.1.1\text{-}1)$$

Când ponderi nenegative sunt asociate transformărilor, atunci suma minimă a ponderilor $T_1 \circ T_2 \circ \ldots \circ T_k (a) = b$ poate fi văzută ca distanţa de la o secvenţă *a* la o

alta *b*. Aceasta este descrisă în mod explicit în [WATE'76] şi, cu o simetrizare evidentă, este obţinut un spaţiu metric. Cazuri interesante apar când τ este restricţionat la seturi specifice de transformări şi ponderi specifice. Spre exemplu, este important în ce mod există sau nu algoritmi eficienţi.

Pentru mutaţii, inserţii şi ştergeri, aliniamentul corespunzător sumei minime de ponderi ar trebui să fie afişat atâta timp cât fiecare element al secvenţei nu este în mai mult de un singur eveniment evoluţionar. Lăsând d(x,y) să fie ponderea substituţiei y pentru x, chiar având d(x, x)=0 pentru toţi x, nu face din suma aliniamentului minim de ponderi o metrică. Dacă d(*.*) este o metrică pe mulţimea de litere, atunci Sellers în [SELL'74b] arată că rezultă o metrică pe S. În acest caz poate fi afişat aliniamentul.

O aboradare a lui Needelman şi Wunch în [NEED'70], prezintă un algoritm pentru maximizarea numărului de potriviri minus numărul de inserări şi ştergeri. Acesta este referit ca un criteriu de similaritate maximă în timp ce, cel amitit anterior este un criteriu al distanţei minime. Legarea conceptului de distanţă cu cel de similaritare este important în psihologie [SHEP'80], iar relaţia dintre ele este de asemenea importantă în acest context. Uneori, este folosit d(x, y)<0 iar minimul rezultat referit ca distanţă. În [WATE'84], distanţa este rezultatul minimizării ponderilor nenegative, cu d(x, x)=0. Situaţii cu d(x, y)<0 pot fi numite *similaritate negativă*.

Fiecare substituţie sau pereche (a_i, b_j) într-un aliniament corespunde uneia cu k = 1,2,...,16 perechi (A,A), (A,T), ..,(G,G) şi are o similaritate $\alpha_k \geq 0$ sau o distanţă $\beta_k \geq 0$.

Observaţie: Operaţiile de inserţie şi ştergere sunt referite adeseori cu un singur cuvânt: ***indels.***

Indels au fiecare o pondere dată *w* pentru similaritate sau *x* pentru distanţă. Aliniamentul similarităţii maxime coincide cu aliniamentul distanţei minime dacă şi numai dacă

$$\beta_i = \max_{1\leq j\leq 16}(\alpha_j) - \alpha_i \quad \text{pentru i = 1, 2, ..., 16}$$

şi

$$x = \left(\max_{1\leq j\leq 16} \alpha_j\right)/2 + w. \qquad (2.1.1\text{-}2)$$

Un rezultat mai general decât acesta poate fi găsit în [SMIT'81]. Problema legării *distanţei* de *similaritate* presupune existenţa unui criteriu general pentru similaritate care să poată fi complementară unei distanţe metrice. Aceasta însă ramâne o chestiune deschisă, abordată separat de diverşi algoritmi.

2.1.1.1. Similaritate versus distanţă

Elementele matricelor de scoruri folosite pentru estimarea similarităţii secvenţelor (vor fi tratate mai pe larg în secţiunea 2.1.2) specifică ponderea care trebuie asociată unei comparaţii date prin:

- Costul înlocuirii unui reziduu cu un altul (distanţa); sau o măsură a similarităţii pentru înlocuire.
- Distanţa este în mod natural folosită pentru reconstruirea arborelui filogenetic; similaritatea este folosită pentru căutarea în baze de date.
- Logica algoritmului nu se schimbă: maximizarea similarităţii este fundamental aceeaşi cu minimizarea unei distanţe.
- Matricele de distanţă şi similaritate sunt inter-convertibile prin unele transformări matematice în conformitate cu aplicaţia dată.

2.1.2. Matrice de substituţie

În scopul evaluării gradului de similaritate al secvenţelor aliniate sunt necesare planuri de calcul care de cele mai multe ori se bazează pe matrice de ponderi obţinute din analiza comportamentelor ce formează secvenţele. Alegerea uneia dintre matrice poate influienţa rezultatul analizei iar ca un alt aspect, ele reprezintă o teorie particulară a evoluţiei.

Un biolog cu o bună intuiţie pentru proteine ar putea inventa un set de 210 termeni de calcul pentru toate perechile posibile de amino acizi dar este extrem de util de a avea o teorie ghid pentru semnificaţia scorului. Pentru exemplificare, vor fi prezentate derivarea de scoruri de substituţie dintr-un model probabilistic. Fie o pereche de secvenţe x şi y, de lungime *n* şi respectiv *m* iar x_i este al *i*-lea simbol în x şi y_j al *j*-lea simbol al lui y. Aceste simboluri vin din dintr-un alfabet A; în cazul ADN alfabetul va fi format din 4 baze {A, G, C, T} şi în cazul proteinelor din 20 amino acizi. Simbolurile din acest alfabet A vor fi notate cu litere mici ca *a*, *b*. Secvenţele care vor fi aliniate simbolic se consideră a nu avea goluri („*gap*"-uri), astfel vor avea lungime egală.

Fiind dată o prereche de secvenţe aliniate se doreşte asocierea unui scor acestui aliniament, care să dea o măsură a probabilităţii relative ca secvenţele să fie înrudite sau complet divergente. Aceasta se realizează de regulă folosind modele care asociază o probabilitate unui aliniament pentru fiecare caz; prin urmare, se consideră raportul dintre cele două probabilităţi.

Modelul independent sau **aleator** R este cel mai simplu. El presupune că o litera *a* apare independent cu o frecvenţă q_a, şi prin urmare probabilitatea celor două secvenţe este tocmai produsul probabilităţilor fiecărui amino acid:

$$P(x, y \mid R) = \prod_i q_{x_i} \prod_j q_{y_j} . \qquad (2.1.2\text{-}1)$$

În **modelul alternativ** de potrivire M, perechile de reziduuri aliniate apar cu o probabilitate de legătură q_{ab}. Această valoare q_{ab} poate fi considerată drept probabilitatea ca reziduurile *a* şi *b* sunt fiecare derivate independent dintr-un reziduu original necunoscut *c* în strămoşul lor comun (*c* poate fi la fel ca *a* şi/sau *b*). Aceasta dă o probabilitate întregului aliniament de

$$P(x, y \mid R) = \prod_i q_{x_i y_i} . \qquad (2.1.2\text{-}2)$$

Raportul acestor două probabilităţi este cunoscut ca *grad de asociere* numit în literatura de specalitate şi *„**odds ratio**”*:

$$\frac{P(x,y \mid M)}{P(x,y \mid R)} = \frac{\Pi_i p_{x_i y_i}}{\Pi_i q_{x_i} \Pi_i q_{y_i}} = \prod_i \frac{p_{x_i y_i}}{q_{x_i} q_{y_i}}. \qquad (2.1.2\text{-}3)$$

Pentru a ajunge la un sistem aditiv de calcul, se ia logaritmul acestui raport numit *„log-odds ratio”*:

$$S = \sum_i s(x_i, y_i), \qquad (2.1.2\text{-}4)$$

unde

$$s(a,b) = \log\left(\frac{p_{ab}}{q_a q_b}\right) \qquad (2.1.2\text{-}5)$$

este log din raportul de probabilitate al perechii de reziduuri (*a*,*b*) care apare ca pereche aliniată, opusă uneia nealiniate.

Aşa cum s-a dorit, ecuaţia 2.1.2-4 este o sumă de scoruri individuale *s(a,b)* pentru fiecare pereche de reziduuri aliniată. Valorile *s(a,b)* pot fi aranjate într-o matrice. Pentru proteine, spre exemplu, ele formează o matrice de 20×20, cu $s(a_i,a_j)$ în poziţia *i*, *j* în matrice, unde a_i, a_j sunt al *i*-lea şi al *j*-lea amono acizi (în numerotaţii). Aceasta este cunoscută ca *matricea de scoruri* sau *matricea de substituţie*. Un exemplu de matrice obţinute în acest fel sunt matricele PAM şi BLOSSUM.

Observaţii:
Când se consideră matrice de scoruri se foloseşte convenţia conform căreia matricele au indici numerici ce corespund liniilor şi coloanelor matricei. Astfel, M_{11} se referă la intrarea din prima linie şi prima coloană. În general, M_{ij} se va referi la conţinutul corespunzător liniei *i* şi coloanei *j*. Pentru aliniamentul secvenţelor, se asociază fiecărei litere din alfabet valori numerice.

Urmând abordarea lui Wheeler în [WHEE'96], dacă alfabetul este A={A; C; G; T}, atunci A=1, C=2, etc. Astfel la M_{12} se va afla scorul dintre A şi C . Deoarece vor fi considerte diverse matrice de scoruri, ele vor fi distinse folosind diferite litere ca R_{ij} pentru matricea de înlocuiri, S_{ij} pentru matricea log odds[7], şi aşa mai departe.

Distanţa dintre două secvenţe este calculată ca suma de diferenţe dintre secvenţe. Diferenţele se pot datora inserărilor şi ştergerilor de nucleotide sau amino acizi (în funcţie de tipul secvenţelor comparate). În acest scop au fost dezvoltate tabele de frecvenţă numite Percent Accepted Mutations sau matrice PAM care măsoară ratele de mutaţii ale amino acizilor în cadrul familiilor de proteine.

[7] Matricele **log-odds** sumarizează înlocuirile observate care au avut loc în timp ce s-au conservat proprietăţile esenţiale a multor familii de proteine.

Există mai multe tipuri de matrice PAM (PAM 250[8], PAM 45) [DAYH'78]. Un alt tip de matrice des utilizate sunt BLOSUM (Blocks Substitution Matrix) care sunt matrice de substituţie derivate din observarea frecvenţelor înlocuirilor de amino acizi în regiunile foarte conservate ale aliniamentelor locale fară discontinuităţi. Datele pentru scorurile de substituţie în aceste matrice provin de la aproximativ 2000 blocuri de segmente de secvenţe aliniate caracterizând mai mult de 500 de grupuri de proteine înrudite [HENI'92].

2.1.2.1. Matrice de substituţie pentru nucleotide

1. Matricea identitate (similaritate)

	A	**T**	**C**	**G**
A	1	0	0	0
T	0	1	0	0
C	0	0	1	0
G	0	0	0	1

Pentru elementele în linia i pe coloana j: $S_{ij} = 1$, $i=j$; $S_{ij} = 0$, $i \neq j$

2. Matricea BLAST (similaritate)

	A	**T**	**C**	**G**
A	5	-4	-4	-4
T	-4	5	-4	-4
C	-4	-4	5	-4
G	-4	-4	-4	5

3. Matricea de Tranziţie/Transversie

	A	**T**	**C**	**G**
A	0	5	5	1

[8] Numărul 250 în PAM250 corespunde unei medii de 250 înlocuiri de amino acizi per 100 reziduuri dintr-un se de date de 71 de secvenţe aliniate [DAYH'78]. Cu cât este mai mare numărul matricii cu atât este mai mare distanţa evoluţională dintre secvenţele comparate.

T	5	0	1	5
C	5	1	0	5
G	1	5	5	0

Folosirea unei matrice de Tranziţie/Transversie reduce "zgomotul" în comparaţii de secvenţe distante ca înrudire.

2.1.2.2. Matrice de substituţie pentru proteine

a) Matricea identitate

$R_{ij} = 1,\ i=j$
$S_{ij} = 0,\ i \neq j$

b) Matricea Codului Genetic, ce presupune un scor bazat pe numărul minim de schimbări de baze necesar pentru convertirea unui amino acid în altul (matricea distanţă conţine costurile mutaţiilor de amino acizi [WHEE'96]).
c) Caractaeristici Fizoco/chimice, care încearcă să cuantifice unele atribute fizice sau chimice ale rezuduurilor şi asignează arbitrar ponderi pe baza similarităţii reziduurilor în proprietatea aleasă (ex. Matricea hidrofobicităţii [WHEE'96] construită din date hidofile (M. Levitt, J. Mol. Biol. 104, 59 [1976]) derivate de George et al. 1990, şi prezentate în lucrarea: Mutation Data Matrix and Its Uses, Methods in Enzymology 183, 333.)

2.1.2.3. Matricea "Log odds"

$$S_{ij} = \log \frac{q_{ij}}{p_i p_j} \qquad (2.1.2.3\text{-}1)$$

S este raportul a două probabilităţi: probabilitatea ca două reziduuri, i şi j sunt aliniate prin descendenţă evoluţionară şi probabilitatea ca ele să fie aliniate din întâmplare.

- q_{ij} sunt frecvenţele cu care se observă alinierea reziduurilor i şi j în secvenţe cunoscute ca înrudite. Ele sunt derivate dintr-o "matrice a probabilităţilor de tranziţie."
- p_i şi p_j sunt frecvenţele de apariţie a reziduurilor *i* şi *j* în setul de secvenţe.
- ex., PAM250, BLOSUM62 şi altele.

2.1.2.4. Matricea PAM(250)

Paşii parcurşi:
1. Se aliniază secvenţe care sunt cel puţin 85% identice. În plus, se minimizează ambiguităţile în aliniamente şi numărul mutaţiilor identice.

2. Se reconstruieşte arborele filogenetic şi se inferă secvenţele originale (cu rol de ancestor). Au fost folosiţi 71 arbori conţinând 1 572 schimbări.
3. Punctarea înlocuirilor acceptate prin selecţia naturală, în toate comparaţiile de perechi de secvenţe (fiecare A_{ij} este frecvenţa cu care amino acidul *j* a fost înlocuit de amino acidul *i* în toate comparaţiile).
4. Se calculează mutabilitatea amino acizilor, $\boldsymbol{m_{ij}}$, adică tendinţa unui aminoacid *j* de a fi înlocuit;
5. Se combină datele de la 3. şi 4. penntru a produce o matrice a probabilităţii mutaţiilor pentru o distanţă evoluţionară PAM, în conformitate cu formula

$$M_{ij} = \frac{m_j A_{ij}}{\sum_i A_{ij}} \qquad (2.1.2.4\text{-}1)$$

$$M_{jj} = 1 - m_j$$

6. Se calculează Matricea *log odds* pentru scorurile de similaritate:

Se împarte fiecare element al matricei de mutaţii M, la frecvenţa apariţiilor fiecărui reziduu:

$$R_{ij} = \frac{M_{ij}}{f_i} \qquad (2.1.2.4\text{-}1)$$

R este o matrice a înrudirilor matricei de asociere (a *Relatedness Odds Matrix)*, f_i este frecvenţa reziduului *i*. Matricea *log odds*, S_{ij} este calculată din matricea înrudirilor de asociere , R_{ij} simplu, luând *log* din fiecare R_{ij}.
Familii diferite de proteine manifestă rate PAM diferite.

2.1.2.5. Proprietăţi ale Matricei Probabilităţilor de Mutaţii

1. Suma lui $\boldsymbol{m_{ij}}$ pentru orice coloană *j*, este unu (trivial). A se nota faptul că probabilitatea cu care un amino acid se va modifica este de ordinul 1% pentru fiecare amino acid. Probabilitatea că el va rămâne acelaşi este de ordinul 99% pentru fiecare amino acid.
2. Matricea Probabilităţilor de Mutaţii (The Mutation Probability Matrix), M1, *defineşte* un element de schimbare evoluţionară: anume, 1 PAM (Accepted Point Mutation per 100 residues). Matricea poate fi folosită pentru a simula evoluţia prin folosirea unui generator de numere aleatoare pentru a selecta soarta fiecărui reziduu în secvenţă în conformitate cu probabilitatea dată în tabel.
3. Aplicarea succesivă a M1 pe o secvenţă rezultă în 2, 3, 4... PAM de modificări evoluţionare.
4. Matricea conţine informaţii compoziţionale, atâta timp cât ea depinde de frecvenţele relative ale amino acizilor din baza de secvenţe din care sunt deduse potrivirile. Pentru cazurile extreme se obţin:

a. Elementele matricei 0 PAM sunt 1 pentru M_{ii} şi 0 pentru M_{ij};
b. Elemetele ∞ ale matricei PAM abordează o compoziţie a amino acizilor asimptotică.

Presupuneri în modelul PAM:

1. Înlocuirile în fiecare poziţie depind numai de amino acidul din acea poziţie şi probabilitatea dată de tabel (model Markov).
2. Secvenţele comparate au compoziţia medie a amino acizilor.

Surse de eroare în modelul PAM

1. Multe secvenţe se abat de la compoziţia medie.
2. Înlocuirile rare nu se observă prea frecvent pentru a rezolva acurateţea probabilităţilor relative (pentru 36 de perechi nu s-a observat nici o înlocuire!).
3. Erorile în 1PAM sunt intensificate în explorarea la 250 PAM.
4. Procesul Markov este o reprezentare imperfectă a evoluţiei: Secvenţele întrudite ca distanţe au de obicei insule (blocuri) de reziduuri conservate. Aceasta implică faptul că înlocuirea nu este egal probabilă pe întreaga secvenţă.

2.1.2.6. Matricea BLOSUM (Blocks Substitution Matrix)

Iniţiatorii acestui tip de matrice sunt Steven Henikoff şi Jorja G. Henikoff (1992) în lucrarea "Amino acid substitution matrices from protein blocks" apărută în Proceedings of Natural Academic Sciences în 1989.

1. Datele de start sunt blocuri conservate extrase dintr-o bază specializată de blocuri şi conţin:
   - Secvenţe aliniate, fără goluri (ungapped sequences);
   - Prezintă un grad de similaritate foarte variată dar măsurile sunt luate astfel încât să evite limitarea secvenţei de test cu secvenţele foarte înrudite care apar frecvent.
2. Înregistrările înlocuirilor sunt făcute direct prin consemnarea tuturor perechilor de reziduuri aliniate f_{ij} .

Frecvenţa observată a fiecărei perechi este: $q_{ij} = f_{ij}$ ***/ (numărul total al perechilor de reziduuri)***

   - Aceasta include cazul în care $i = j$ (i.e. nici o înlocuire observată);
   - Frecvenţa aşteptată a fiecărei perechi este în esenţă produsul frecvenţelor fiecărui reziduu din setul de date.
3. Secvenţele similare dintr-un bloc, deasupra unui prag procentual de similaritate, sunt grupate iar membrii ai grupului contribuie fracţional la întregistrarea finală. Această etapă:
   - Reduce numărul perechilor identice (AA, SS, TT, etc., potriviri) în înregistrările finale;
   - Este într-o oarecare măsură analogă cu creşterea distanţei PAM .
   - Dacă pragul de grupare este 80%, matricea finală este BLOSUM 80.

- Gruparea la 62% reduce numărul de blocuri care contribuie la tabel cu 25% - încă 1.25 x 10^6 perechi contributante!
- Cea mai puţin frecventă înlocuire a perechilor de amino acizi a fost observată de 2369 ori!

2.1.2.7. Noi matrice de scoruri

O actualizare a matricei PAM folosind metoda lui Dayhoff este propusă de autori în [DAVI'92] unde sunt marcate 59 190 mutaţii acceptate în 16 130 secvenţe.

O altă matrice de scoruri rezultată din aliniamentul întregii baze de date SWISS-PROT pentru care s-au identificat 1.7 ×10^6 potriviri din secvenţe prin deducţie de 6.4 la 100 PAM este propusă de Gonnet et al. în [GONN'92].

Alte matrice specializate de scoruri

- În [LEE'94] se propune crearea unei matrice "step-matrix" bazată numai pe blocuri aliniate de receptori cuplaţi de G-Proteine în care elementele matricei
- sunt proporţionale cu raritatea substituţiei. Constituie un avantaj excelent în construirea unei filogenii coerente a larg divergenţilor receptori ai G-proteinei.
- Matrice pentru detectarea mutaţiilor "frame shift"care duc la noi secvenţe codificate (sau care provin din erori de secvenţiere) [CLAV'93].
- Matrice de scoruri create din substituţii observate de reziduuri găsite în medii structurale similare 3D[BORD'91].

2.2. Starea actuală în dezvoltarea unor noi metode de similaritate

În această secţiune sunt prezentate o varietate de metode oferite publicului pentru exprimarea similarităţii dintre secvenţele biologice. Datorită versatilităţii acestor tehnici se va evita impunerea vreunui model de organizare a prezentării lor. În schimb ele vor fi descrise succesiv în funcţie de ordinea lor cronologică, incluzând ideile principale, datele folosite, rezultatele şi unele concluzii.

O abordare interesantă este întâlnită în [SJOL'96], unde este descrisă o metodă pentru detectarea omologiei slabe dintre proteine utilizând modelarea mixturilor Dirichlet. Densităţile incorporate sunt proiectate pentru a fi combinate cu fecvenţele observate ale amino acizilor, în scopul formării de estimări ale probabilităţilor aşteptate ale amino acizilor pentru fiecare poziţie într-un profil HMM sau orice alt model statistic. Backgroud-ul teoretic al acestei lucrări a fost stabilit în publicaţii anterioare [BERG'85], [SANT'89].

În [KATT'00], au fost selectate un set de 80 000 de proteine din baza de date SWISS-PROT şi analizate pentru repetiţiile în tandem folosind tehnica alunecarii ferestrelor ("sliding window techique"). Modelele observate că se repetă au fost analizate în privinţa impactului pe care îl au asupra structurii generale şi funcţiei

proteinelor. Rezultatele obţinute au fost organizate sub forma unei baze de date[9] public accesibilă.

Autorii în [ARSL'01] afirmă că "algoritmul Smith-Waterman găseşte aliniamentul local cu scor maximal dar este incapabil să găsească aliniamentul local cu gradul maxim de similaritate" precum şi procentul maxim de potriviri. De asemenea, este menţionat faptul că încă nu există un algoritm eficient care să poată oferi un răspuns la întrebarea dacă două secvenţe partajează un fragment suficient de lung având un grad de similaritate mai mare de 70%. Ca rezultat, uneori aliniamentele locale oferă un "mozaic" de fragmente bine conservate, conectate artificial de fragmente slab conservate sau neânrudite. O remarcă similară a fost făcută de Zhang în [ZHAN'99]. El spune că asemenea cazuri pot conduce la probleme în compararea secvenţelor genomice lungi şi predicţia comparativă a genelor. Pornind de la această motivaţie, autorii din [ARSL'01] propun un nou algoritm de comparaţie al secvenţelor numit aliniament local normalizat (NLA) care este capabil să raporteze regiuni cu grad maxim de similaritate. Problema este formulată sub următoarea formă: scorul $s(I, J)$ al aliniamentului local ce implică subşirurile I şi J poate fi ajustat prin împărţirea lui $s(I, J)$ cu lungimea totală a regiunilor aliniate, $s(I, J)/(|I| + |J|)$. Problema NLA este astfel de a găsi subşirurile I şi J care maximizează $s(I, J)/(| I | + | J |)$ dintre toate subşirurile I şi J cu $| I | + | J | \geq T$, unde T este un prag pentru lungimea generală minimă a lui I şi J. Cu scopul de a fi important din punct de vedere biologic pentru această problemă, autorii au introdus aici un obiectiv puţin diferit sub forma $s(I, J)/L(| I | + | J | + L)$, pentru un parametru dat L. Ideea a fost de a controla normalizarea variind pe L şi au implementat algoritmul lui Dinkelbach [DINK'67] cunoscut ca programare fracţională, ce foloseşte o metodă parametrică drept tehnică de optimizare. Noua metodă s-a dovedit a fi de 3-5 ori mai înceată decît algoritmul standard Smith-Waterman şi sunt alţi parametri care influenţează rezultatele. Se concluzionează că nu o singură alegere a lui L elimină toate efectele nedorite şi relevă toate aliniamentele importante în acelaşi timp. Totuşi, autorii argumentează că există o valoare a lui L cu care poate fi detectat, folosind algoritmul NLA, un aliniament important dacă are un scor normalizat suficient de mare.

În [ESKI'01], este găsită o motivaţie biologică pentru folosirea unui model mixt de strămoşi comuni pentru a estima probabilitatea de distribuţie din familiile de proteine. Autorii consideră metoda a fi mai bună decât mixturile Dirichlet deoarece ea este "simplu de calculat pentru alfabete mari". Rezultatele experimentale sunt evaluate folosind metoda ROC (folosită în secţiunea 5.5) şi ele sunt comparabile cu cele obţinute cu metodele anterioare. Munca fundamentală care descrie utilizarea mixturilor de "ancestori" comuni în estimarea probabilităţilor evenimentelor discrete în secvenţele biologice este oferită în [JAAK'00].

O metodă mai puţin relevantă dar care utilizează conţinutul secvenţelor de proteine este dezvoltată în [SARK'02]. Ideea principală este de a extrage relaţiile de interacţiune dintre proteine din texte stiinţifice (ex. abstractele din MEDLINE) înrudite cu descrierea proteinelor (ca şi cum ar fi selectate de către un expert).

[9] http://www.ncl-india.org/trips

Prin modificarea unui tool destinat procesării limbajului natural, autorii construiesc o analiză bazată pe reguli pentru regăsirea datelor text necesare găsirii proteinelor similare, cu similaritatea exprimând noţiunea de atribute funcţionale comune. Sunt selectate chiar şi proteinele care nu sunt similare structural dar sunt înrudite prin faptul că provoacă tipuri similare de dezordini neuronale. Rezultatul principal al acestui studiu a constat în faptul că, rulând peste 70 de propoziţii conţinând 40 *predicaţii* marcate, au fost identificate 27 relaţii de interacţiune a proteinelor dintre care 18 corecte. Prin urmare, procentajul nefavorabil (recall-ul) a fost 45% şi precizia de 67%. Concluzia finală a acestui studiu este că toolurile lingvistice nu vor putea niciodată substitui cercetarea medicală.

În [WU'03] autorii propun o nouă tehnică pentru compararea perechilor de secvenţe biologice pe baza unor modele mici asociate cu regiuni bine conservate în unele secvenţe de proteine. Aceste modele sunt folosite ca termeni index în regăsirea informaţiei (information retrieval) fără a considera gap-urile. În acest sens, ei crează grupuri de amino acizi similari dând fiecărui grup un cod. Pentru fiecare secvenţă de patru amino acizi sunt create toate modelele posibile iar similaritatea bazată pe aceste modele este măsurată folosind oricare din următoarele scoruri:

a) $S_1(p_1, p_2) = c \times Match(p_1, p_2)|/(|Pattern(p_1)|+|Pattern(p_2)|)$,
unde $Match(p_1, p_2)$ este setul de modele partajat de secvenţele p_1 şi p_2; c este un factor normalizator care este folosit când se ignoră lungimea proteinelor;

b)$S_2(p_1, p_2) = c \times Match(p_1, p_2)| /(|Pattern(p_1)|+|Pattern(p_2)|)-$
$|Length(p_1)+Length(p_2)|$, când cele două secvenţe de proteine se cer a avea aceeaşi lungime.

Această metodă a fost aplicată pentru a construi arbori filogenetici, pentru gruparea proteinelor (proteins clustering) şi predicţia structurii secundare. Concluzia este că predicţia structurii secundare folosind modele se pare să depăşească alte metode existente; este uşor de implementat şi are o senzitivitate şi specificitate relativ mare. În plus, mecanismul de codificare reduce numărul fragmentelor candidate pentru a fi verificate.

În [CAMO'03] scopul este de a găsi similarităţile în structura bazelor de date de proteine extrăgând vectori de caracteristici de triplete de elemente din structura secundară (SSE) şi aceşti vectori de caracteristici sunt indexaţi folosind o structură multidimensională de indecşi. Această structură de indecsi este folosită pentru a înlătura automat din domeniul de căutare toate proteninele potenţial nepromiţătoare. Pentru restul de proteine este folosit un tool de căutare de aliniamente (numit VAST) pentru efectuarea aliniamentului perechilor [MADE'95]. Tehnica este numită de autori PSI (Protein Structure Index) şi include un pas de modelare statistică pentru luarea deciziei de a considera sau nu proteinele drept candidaţi în căutarea subsecventă. Experimentele au fost efectuate comparând performaţele lui PSI şi VAST pe aceleaşi date. Acestea arată că noua tehnică este mai rapidă în timp ce se menţine aceeaşi senzitivitate.

Un tip diferit de măsuri de similaritate, ca tooluri pentru explorarea ontologică a genelor, este folosit în [LORD'03]. Pe baza ontologiei genelor (GO), descrieri ca funcţia moleculară, procesul biologic şi componenta celulară sunt

comparate via trei măsuri de similaritate semantică. Tipul de date este adnotat şi provine din setul de date GO[10] disponibil accesului public. GO reprezintă secvenţele adnotate din cadrului unui graf aciclic (DAG - Dyrected Acyclic Graph) ce constă dintr-un număr de termeni reprezentaţi ca noduri în graf şi conectaţi de muchii reprezentând relaţiile dintre ei. Termenii pot avea "părinţi" multipli precum şi mai mulţi "copii". Acest mod de manipulare a secvenţelor poate fi folosit pentru a reprezenta toate bazele de date [LORD'03]. În această lucrare experimentele au fost realizate numai pentru acele asocieri care au fost identificate între termenii GO şi proteinele din SWISS-PROT-Human. Toate măsurile testate se bazează pe conţinutul de informaţie al fiecărui termen, care este definit ca numărul de apariţii în corpus al fiecărui termen sau termen copil şi este exprimat ca probabilitate. Fiecare măsură foloseşte conţinutul de informaţie al "părinţilor" partajaţi ai celor doi termeni comparaţi. Remarcile concluzive sugerează că toate cele trei măsuri dovedesc o corelare puternică dintre similaritatea secvenţelor şi similaritatea semanitică a funcţiei moleculare şi nici una nu se dovedeşte a avea o superioritate clară faţă de celelalte.

O abordare diferită a similarităţii este utilizată în [KRAS'04]. Autorii propun folosirea metricii universale de similaritate (USM) pentru structurile proteinelor. Este menţionat faptul că această metodă a fost introdusă şi îmbunătăţită de [LI'01] şi [LI'03] respectiv, şi se bazează pe conceptul de complexitate Kolmogorov. Pe scurt, citez "complexitatea Kolmogorov este o măsură obiectivă a cantităţii de informaţie conţinute întru obiect dat" şi se menţionează că o măsură apropiată este complexitatea Kolmogorov condiţionată a lui o_1 fiind dat o_2, descrisă de relaţia:

$$K(o_1|o_2) = \min\{|P|, P \text{ a program and } U(P, o_2) = o_1\}, \qquad (2.2\text{-}1)$$

unde U este o Maşină Turing Universală, necesară pentru a produce un anume obiect [LI'97]. Această relaţie este folosită pentru definirea distanţei informaţiei (ID-Information Distance) dintre două obiecte după cum urmează [KRAS'04]:

$$ID(o_1|o_2) = \max\{K(o_1|o_2), K(o_2|o_1)\}. \qquad (2.2\text{-}2)$$

Similaritatea structurală dintre perechile de proteine a fost aplicată pe patru seturi de date diferite şi toate distanţele dintre perechi sunt stocate întro matrice de similarităţi care este apoi alimentată de un tool de grupare deja implemetat şi public accesibil[11]. După experimete extensive, concluzia a fost că USM este robust, poate diferenţia între familii de proteine şi sub-familii dar nu oferă vre-un indiciu despre locaţia exactă a (di)similarităţii de-a lungul celor două secvenţe.

Folosind abordarea SVM (Support Vector Machines), metodele discriminative sunt luate în considerare de către autorii [SAIG'04], care le dovedesc a fi cele mai eficiente metode pentru problema recunoaşterii superfamiliilor în clasificarea structurală a proteinelor în baza de date SCOP. Ei explică performanţa ridicată a SVM ca un rezultat al funcţiilor nucleu (kernel-functions) folosite pentru a cuantifica

[10] http://www.godatabase.org/dev
[11] http://www2.biology.ualberta.ca/jbrzusto/cluster.php.

similaritatea dintre secvenţe (ei propun noţiunea de nucleu pentru şirul care este adaptat secvenţelor biologice şi este numit nucleu de aliniament local). Aceste funcţii nucleu măsoară similaritatea dintre două secvenţe prin însumarea scorurilor obţinute din aliniamente locale ale secvenţelor ţinând cont de gap-uri. Experimentele au fost efectuate folosind 4352 proteine din baza de date Astral[12], grupate în familii şi super-familii. Urmând procedura de marcaj din [LIAO'03] pentru fiecare familie, domeniile proteinelor din cadrul familiei au fost considerate exemple pozitive de **testare** şi domeniile proteinelor din cadrul superfamiliei dar dinafara familiei au fost considerate exemple pozitive de **antrenare**. Exemple negative au fost luate dinafara clasei secvenţelor pozitive şi au fost împărţite aleator în seturile de testare şi antrenare. Evaluarea metodei a fost făcută folosind scorurile furnizate de caracteristicile ROC şi rezultatele de concluzie ale autorilor au fost că această metodă depăşeşte performanţele altor metode considerate „state of the art" şi care folosesc SVM.

Analiza semantică latentă (LSA) este o metodă bazată pe tehnica SVD (Singular Value Decomposition). Ea este o abordare extrem de utilă în procesarea limbajului natural pentru generarea sumarelor (rezumatelor), compararea documentelor, generarea de tezaure lingvistice şi mai departe pentru regăsirea informaţiei (information retrieval) [BELL'00], [LAND'98], [BOGA'03a]. Modul în care LSA capturează relaţiile conceptuale din text, pe baza distribuţiei cuvintelor în documente, autorii în [GANA'04] îl folosesc pentru capturarea propensităţilor (tendinţelor) structurii secundare în secvenţele de proteine, folosind diferite vocabulare. Ei consideră documentele $d_1, d_2, \ldots d_{N1}$ a fi segmente nesuprapuse de proteine pentru care sunt cunoscute categoriile structurale $C_1, C_2, \ldots, C_{N1}$ şi $t_1, t_2, \ldots, t_{N2}$ segmente de test nesuprapuse cu lungime cunoscută pentru care trebuie prezisă structura secundară. Structura secundară a datelor de test este prezisă folosind ca metodă un model de clasificare *k*NN (*k* Nearest Neighbour). Pentru fiecare segmet de test t_i, similaritatea cosinus a lui t_i este calculată faţă de toate segmentele de antrenare $d_1, d_2, \ldots d_{N1}$ şi sunt identificate cele *k* segmente având similaritatea cu

[12] www.cs.columbia.edu/compbio/svmpairwise

similaritatea lui t_i maximă. Aceste *k* segmete sunt *k*NN al lui t_i. Categoria prezisă a lui t_i este categoria structurală căreia îi aparţin cei mai mulţi *k*NN. Acest proces este repetat pentru fiecare dintre segmentele de test. În experimente, autorii consideră ca vocabular cei 20 de amino acizi, grupurile chimice şi tipurile de amino acizi. Ca documente ei folosesc structurile de: helix, fâşii şi răsuciri (helix, strand and coil structures) identificate în structura secundară a fiecărei proteine. După metoda LSA a fost folosită metoda VSM (Vector Space Model) pentru aceleaşi modele şi date. Subsecvenţele de proteine au fost tratate ca documente în matricea vocabular/documente. Rezultatele sunt comparate folosind măsurile: precizie şi recall preulate din teoria prelucrării/regăsirii informaţiei (information retrieval theory). Autorii concluzionează că alfabetele distincte diferă prin cantitatea de informaţie pe care o poartă. Astfel, helixurile şi fâşiile (helices and sheets) au fost cel mai bine clasificate folosind LSA cu tipurile de amino acizi ca vocabulare, în timp ce răsucirile (coils) sunt caracterizate cu o mai mare precizie când amino acizii sunt folosiţi ca vocabulare şi pentru analiză se foloseşte metoda SVM.

2.3. Tehnici de realizare a aliniamentului de secvenţe biologice

În cele ce urmează vor fi considerate metode clasice de aliniament şi anume, cei mai larg aplicaţi algoritmi în cercetarea secvenţelor de gene/proteine. Pe baza numărului de secvenţe implicate în aliniament, aceste metode sunt împărţite în algoritmi pentru compararea de perechi de secvenţe sau algoritmi pentru compararea de secvenţe multiple. O mare parte din temele acestui capitol urmează descrierea făcută de unul dintre fondatorii metodeleor de aliniament al secvenţelor biologice, şi anume Michael S. Waterman [WATE'84].

Combinatorica aliniamentelor

Fie secvenţele $a= a_1, a_2, \ldots, a_n$, şi $b= b_1, b_2, \ldots, b_m$. Un aliniament poate fi produs prin creşterea lungimii fiecărei secvenţe cu inserarea lui Δ. Dacă lungimea unui asemenea aliniament este *L*, atunci acesta ar putea fi scris

$$\begin{matrix} a_1^* & a_2^* & & a_L^* \\ b_1^* & b_2^* & \ldots & b_L^* \end{matrix}$$

unde subsecvenţa $a^*(b^*)$ de elemente neegale cu Δ este $a(b)$.
Interesul este în calcularea numărului de aliniamente. Dacă acesta nu este prea mare, atunci o căutare directă este posibilă pentru căutarea aliniamentelor optimale. După cum se va putea observa însă, numărul creşte foarte repede.

Un aliniament a lui $a = a_1, a_2, \ldots, a_n$, cu $b = b_1, b_2, \ldots, b_m$ poate sfârşi în unul din următoarele trei moduri:

$$\begin{matrix} \ldots a_n & \ldots a_n & \ldots \Delta \\ \ldots \Delta & \ldots b_m & \ldots b_m \end{matrix}$$

(alinierea lui Δ peste Δ este eliminată deoarece nu contribuie cu vreo informaţie). Dacă *f(n,m)* calculează configuraţii generate de aliniamente terminate recursiv ca mai sus, atunci

$$f(n,m) = f(n-1,m) + f(n-1, m-1) + f(n,m-1) .$$

Numerele generate de această ecuaţie recursivă sunt cunoscute ca numere Stanton-Cowan [STAN'70], unde ele provin din calcularea volumului unei sfere de rază *m* în *n* dimensiuni folosind metrica Lee. Asimptotele sunt date pentru o generalizare a acestor numere de către H.T. Laquer [LAQU'81]. El arată că

$$f(n,n) \approx \left(1+\sqrt{2}\right)^{2n+1}\sqrt{n.} \qquad (2.3.-1)$$

După o examinare atentă, s-a observat că *f(n,m)* supracalculează aliniamentele sau cel puţin o definire rezonabilă a aliniamentelor. Spre exemplu, cele două aliniamente

$$\begin{matrix} A\Delta \\ \Delta T \end{matrix} \quad \text{şi} \quad \begin{matrix} \Delta A \\ T\Delta \end{matrix}$$

pot să nu fie distincte într-un sens biologic. Pentru a proiecta o recursivitate care nu calculează dublu aceste ştergeri „tandem", fie *g(n,m)* numărul unor asemenea aliniamente. Dacă un aliniament se termină într-un Δ există trei posibilităţi:

$$\begin{matrix} \ldots a_{n-1}\, a_n \\ \ldots b_m\, \Delta \end{matrix} \qquad \begin{matrix} \ldots a_{n-1}\, a_n \\ \ldots \Delta\, \Delta \end{matrix} \qquad \begin{matrix} \ldots \Delta\, a_n \\ \ldots b_m\, \Delta; \end{matrix}$$

şi dacă un aliniament se termină în b_m, există trei posibilităţi:

$$\begin{matrix} \ldots a_n\ \Delta \\ \ldots b_{m-1}\, b_m \end{matrix} \qquad \begin{matrix} \ldots \Delta\, \Delta \\ \ldots b_{m-1}\, b_m \end{matrix} \qquad \begin{matrix} \ldots a_n\Delta. \\ \ldots \Delta\, b_m \end{matrix}$$

Prin urmare,

$$g(n, m) = g(n-1, m) + g(n, m-1) + g(n-1, m-1) - g(n-1, m-1)$$

sau

$$g(n,m) = g(n-1, m) + g(n, m-1) ,$$

o revenire a caracterului mai simplu decât recursivitatea Stanton-Cowan.
Condiţiile de limită pentru recursivitatea adoptată de [WATE'84] trebuie considerate în aceasta fază.
Evident,

$$g(0, 0) = g(1,0) = g(0, 1) = 1,$$

astfel soluţia explicită poate fi scrisă

$$g(n,m) = \left(\frac{n+m}{n}\right). \qquad (2.3\text{-}2)$$

Cu formula lui Stirling,

$$g(n,n) \approx 2^{2n} / \left(4\sqrt{n\pi}\right) \qquad (2.3\text{-}3)$$

pentru n = 1000, g(n, n) > 10^{600} şi examinarea directă a tuturor aliniamentelor este imposibilă. Acesta este unul din motivele care a dus la dezvoltarea de algoritmi eficienţi.

2.3.1 Metode vizuale

Există o metodă importantă de analiză a secvenţelor care este cel mai bine descrisă de expresia „doar uită-te la...". Un analist poate observa o succesiune ca GCGCGC... şi să găsească o caracteristică de interes fără vreo metodă matematică sau computaţională sofisticată. Este posibil să se formuleze o problemă analitică dintr-o asemenea observaţie: "Cât de probabilă este observaţia unui asemenea eveniment în secvenţe aleatoare?" Un biolog este puţin probabil surprins, informat sau impresionat dacă i se spune că apariţia succesiunii GCGC... a fost neobişnuită. În schimb, problema funcţiei regiunii GC captează atenţia biologului. Doar privirea secvenţelor poate fi utilă.

Metoda „dot matrix" tradusă ca „matricea punct" este o metodă vizuală larg răspânită care în general foloseşte calculatorul. În acest caz, este formată o matrice M = (m_{ij}), unde m_{ij} = 0 dacă al *i*-lea element al secvenţei *a* este diferit de al *j*-lea element al secvenţei *b* altfel m_{ij} =1. Execuţia potrivirilor exacte arată ca diagonale de elemente 1. Matricea punct a fost descoperită independent de câteva ori, spre exemplu Maizel şi Lenk (1981), Novotny (1982), Harr et al.(1982), Jagadeeswaran şi McGuire (1982), Gibbs şi McIntzre(1970).

Multe dintre aceste implementări filtrează potrivirile pentru a afişa doar execuţia unui element sau mai multe. Este posibilă combinarea unor metode mai sofisticate pentru filtrarea potrivirilor.

Metoda *matricei punct* este utilă în localizarea regiunilor de mare potrivire dintre două secvenţe. Este normal astfel de a fi interesaţi de probabilitatea distribuţiei celei mai lungi potriviri dintre două secvenţe. Această distribuţie va permite analistului să localizeze potrivirile semnificative.

2.3.2. Algoritmi pentru aliniamentul perechilor de secvenţe

Problema obţinerii aliniamentului global optimal dintre două secvenţe permiţând gap-uri a fost rezolvată la început folosind un algoritm de programare dinamică. Metoda produsă este cunoscută în analiza secvenţelor biologice ca algoritmul Needleman-Wunch [NEED'70], după numele celor doi autori. Ideea principală este de a construi un aliniament optimal folosind soluţii anterioare pentru aliniamente optimale ale unor subsecvenţe mai mici. Algoritmul a fost îmbunătăţit mai departe de Sellers [SELL'74] şi mai târziu optimizat de O. Gotoh [GOTO'82].

Cu intenţia de a căuta cel mai bun aliniament local util pentru aflarea dacă două proteine împart un domeniu comun, pentru a compara secţiuni extinse ale secvenţelor genomice de ADN sau pentru a compara două secvenţe foarte divergente, a fost dezvoltat algoritmul Smith-Waterman [SMIT'81a].

O metodă alternativă aliniamentului, bazată pe algoritmi de programare dinamică, foloseşte modelul stărilor automatului finit (Finite State Automata) şi utilizează tooluri interactive construite automat [SEAR'95].

Metodelele de aliniament amintite sunt considerate corecte în privinţa unui scor optimal. Ele se bazează pe crearea dinamică a unor matrice şi ca rezultat complexitatea timp este de ordinul O(*nm*), (i.e. produsul lungimii secvenţelor) [DURB'98]. Aceasta înseamnă că pentru o bază de date mare de proteine/secvenţe, timpul de execuţie poate deveni considerabil de mare. Pentru ajustarea acestei probleme, au fost introduse tehnicile euristice. În aceste tehnici există un compromis între acurateţe şi eficienţă [LIAO'03]. Dintre cele mai cunoscute metode sunt BLAST [ALTS'90], [ALTS'97] şi FASTA [PEAR'88]-[PEAR'98], care au fost dezvoltate pentru detectarea omologiilor în baze de date mari cu ajutorul scorurilor obţinute din aliniamente locale. Ambele au fost incorporate în pachete de programe accesibile public[13,14].

În toate tehnicile menţionate mai sus se cere o atenţie suplimentară în alegerea modelului de calcul al scorului de similaritate atâta timp cât scorul total asociat unui aliniament va fi o sumă de termeni corespunzând diferenţelor dintre reziduurile perechilor aliniate plus termenii corespunzători execuţiei operaţiilor de inserare/ştergere. Schema de calcul incorporată este bazată pe una dintre matricele de substituţie disponibile (fiecare având propriile caracteristici). Cele mai populare dintre ele sunt PAM, BLOSSUM, GONNET şi matricea de identitate ADN. Semnificaţia scorurilor obţinute este calculată făcând comparaţii cu un model Baiesian sau folosind abordarea clasică a comparării statistice a valorilor extreme ale scorurilor de potrivire [DURB'98].

Pentru o analiză mai detaliată vor fi prezentate în continuare metodele cele mai importante de aliniere a perchilor de secvenţe.

Problematica acestui subcapitol ar putea fi descrisă ca cea a „găsirii numărului cel mai mic" de paşi care să schimbe o secvenţă în cealaltă. Dacă cele două secvenţe ar fi formate din acizi nucleici: $a = a_1, a_2, \ldots, a_n$, şi $b = b_1, b_2, \ldots, b_m$, unde a_i şi b_j sunt unul dintre nucleotidele adenina (A), timina (T), guanina (G) sau citozina (C). Aceste secvenţe sunt cuvinte finite formate din cele patru litere ale alfabetului folosit iar paşii menţionaţi mai sus corespund evenimentelor evoluţionare ce pot altera secvenţa. Cel mai simplu set de evenimente evoluţioniste constă din mutaţii, unde o literă (sau mai multe) este substituită de o alta şi inserări sau ştergeri unde o literă sau mai multe sunt inserate respectiv şterse.

Dacă a = AATAG, atunci o substituire a lui T pentru a_2 = A transformă a în b:

$$a = \text{AATAG} \rightarrow b = \text{ATTAG}.$$

[13] http://fasta.bioch.virginia.edu/

[14] http://www.ncbi.nlm.nih.gov/Class/BLAST/blast_course.short.html

Această corespondenţă este arătată de obicei prin reprezentarea într-un aliniament:

a: AATAG;
b: ATTAG.

Dacă a_4=A este şters, transformarea în *c* = AATG este reprezentată de

a: AATAG;
c: ATTΔG,

unde ştergerea lui a_4=A este indicată de un „Δ" inserat în *c*. Prin corespondenţă, inserarea lui C între a_3 şi a_4 este reprezentată prin

a: AATΔAG
b: AATCAG.

Kruskal şi Sankoff [KRUS'83] plasează aceste probleme într-un context general. O altă formulare generală este în [WATE'76] care a mai fost amintită şi presupune următoarele: Fie S setul de cuvinte finite dintr-un alfabet, incluzând cuvântul vid şi fie $\tau = \{ T \mid T : S \rightarrow S \}$ un set de transformări care include transformările identice. Interesul este în transformările $T_1, T_2, \ldots, T_k$ din τ astfel încât

$$T_1 \circ T_2 \circ \ldots \circ T_k (a) = b. \qquad (2.3.2\text{-}1)$$

Pe baza acestei exprimări pot fi formulate probleme variate. Pentru problema celui mai mic număr de paşi trebuie căutat *k* minim unde τ = {mutaţii de tipul unei singure litere, inserţii şi ştergeri}. Funcţia obiectiv poate fi schimbată iar setul de transformări mărit. Există un echilibru între realitatea biologică şi algoritmii computaţionali.

Unul dintre algoritmii timpurii de comparare a şirurilor s-a datorat lui Levenstein [LEVE'66] cu toate că munca sa nu a influienţat dezvoltările menţionate în continuare. Lucrările lui Fitch şi Margoliash [FITC'67] şi [FITC'69] au adus problema comparării secvenţelor în atenţia unui mare număr de persoane.

2.3.3. Metode fundamentale de programare dinamică

Un algoritm bazat pe programare dinamică furnizează o soluţie elegantă O(n^2), *n* fiind lungimea celei mai mari dintre cele două secvenţe. Programarea dinamică este o tehnică algoritmică ce foloseşte cunoştinţe învăţate din subprobleme anterioare în calcularea soluţiei unei probleme mai mari.
La început se apleleză la intuiţie în spatele programării dinamice pentru calcularea aliniamentului cel mai bun (în termenii distanţei minime).
Până acum, secvenţele $a = a_1, a_2, \ldots, a_n$, şi $b = b_1, b_2, \ldots, b_m$ au fost folosite pentreu a genera aliniamente,

$$a_1^* a_2^* \dots a_L^*$$
$$b_1^* b_2^* \dots b_L^*,$$

unde subsecvenţele de elemente *a** şi *b** neegale cu Δ sunt secvenţele originale. Pentru cazul similarităţii şi distanţei, ecuaţia de revenire pentru *f(n,m)* (din introducerea secţiunii 2.3) este modificată pentru a oferi un mod eficient pentru calcularea similarităţii şi distanţei.

Din punct de vedere istoric, aceste metode au început în biologie cu Needleman şi Wunsch [NEED'70] care au prezentat algoritmul descris în cele ce urmează. După aceea, Sankoff [SANK'72] şi Sankoff şi Sellers [SANK'73] găsesc metode de programare dinamică pentru aliniamente optimale cu un număr dat de indels. Sellers [SELL'74a,b] vine cu algoritmi pentru calcularea distanţei D(a,b), problemă ridicată de Ulam [ULAM'72]. Gordon [GORD'73] şi Delcoigne şi Hansen [DELC'75] au contribuţii utile în compararea secvenţelor.Toate aceste metode sunt incluse într-o clasă de tehnici cunoscute ca programare dinamică ce a fost introdusă de Richard Bellman. În Byers şi Waterman [BYER'84] poate fi găsită o discuţie generală pe această temă.

Considerând alfabetul {A, C, G, T, Δ}, lărgit pentru a include Δ, fie o funcţie de pondere *s(a, b)* definită pe perechi de litere din alfabet. O funcţie de tipul folosit de Needelman-Wunch este

$$s(a,b) = \begin{cases} 1 & \text{daca } a = b, \\ 0 & \text{daca } a \neq b,\ a \neq \Delta,\ b \neq \Delta, \\ -1.5 & \text{daca } a \neq b \text{ si unul dintre } a,b = \Delta. \end{cases} \qquad (2.3.3\text{-}1)$$

Aici potrivirile primesc pondere pozitivă iar indels primesc ponderi negative. Următoarele declaraţii apar în [SMIT'81b] unde dovezile merg de-a lungul ecuaţiei pentru *f(n,m)*, aşa cum a fost menţionat în secţiunea 2.3.
La început se defineşte unde se ia maximul peste toate aliniamentele.

$$S(a,b) = \max \sum_{k=1}^{L} s\left(a_k^*, b_k^*\right) \qquad (2.3.3\text{-}2)$$

Algoritmul Needleman-Wunsch
Fie

$$S_{0j} = \sum_{k=1}^{j} s(\Delta, b_k), \qquad S_{00} = 0 \qquad \text{si} \quad S_{i0} = \sum_{k=0}^{i} s(a_k, \Delta).$$

Dacă $S_{ij} = (a_1 a_2 \dots a_i, b_1 b_2 \dots b_j)$, atunci

$$S_{ij} = \max\{S_{i-1,j} + s(a_i, \Delta),\ S_{i-1,j-1} + s(a_i, b_j),\ S_{i,j-1} + s(\Delta, b_j)\}$$

Dacă o funcţie distanţă iniţială *d* este specificată pe {A, C, G, T, Δ}, atunci este obţinut un rezultat similar. O asemenea funcţie este

$$d(a,b) = \begin{cases} 1 & \text{daca } a = b \\ 0 & \text{daca } a \neq b. \end{cases}$$

Se defineşte, ca mai sus,

$$D(a,b) = \min \sum_{k=1}^{L} d(a_k^*, b_k^*) \qquad (2.3.3\text{-}3)$$

Algoritmul Sellers

Fie $D_{0j} = \sum_{k=1}^{j} d(\Delta, b_k)$, $D_{00} = 0$ si $D_{i0} = \sum_{k=0}^{i} d(a_k, \Delta)$

Dacă $D_{ij} = S(a_1 a_2 \ldots a_i, b_1 b_2 \ldots b_j)$, atunci

$$D_{ij} = \min\{D_{i-1,j} + d(a_i, \Delta), D_{i-1,j-1} + d(a_i, b_j), D_{i,j-1} + d(\Delta, b_j)\}$$

Fiecare dintre cei doi algoritmi au timpul computaţional proporţional cu

$$\sum_{i=1}^{n} \sum_{j=1}^{m} 1 = nm. \qquad (2.3.3\text{-}4)$$

Spaţiul cerut pentru calcularea *D(a,b)* este min{*n*,*m*} dar *D(a,b)* este câteodată cu valoarea în afara setului asociat de aliniamente optime.

Există două tehinci de a produce aliniamente. Prima este de a salva pointeri la fiecare (*i*,*j*) pentru a arăta care dintre $D_{i-1,j}$; $D_{i-1,j-1}$; $D_{i,j-1}$ sunt folosiţi în calcularea lui $D_{i,j}$ optim. Pointerii sunt salvaţi în timpul calculului înainte (forward) astfel, în timpul parcurgerilor înapoi (tracebacks), aceştia pot fi urmaţi pentru a produce un aliniament optimal. În cazul în care există *optimi* multipli, pointerii nefolosiţi pot fi păstraţi şi în această manieră (breath-first search with stacking), pot fi produse toate aliniamentele optimale. În al doilea caz, dacă pointerii nu sunt salvaţi, recalculând care dintre $D_{i-1,j}$; $D_{i-1,j-1}$; $D_{i,j-1}$ rezultă în $D_{i,j}$ va fi simplu de realizat dacă matricea *D* este salvată. În oricare din cele două cazuri, spaţiul de stocare necesar este *O*(*nm*).

2.3.4. Extensii ale metodelor de bază

Cele mai importante transformări ale evoluţiei tratate în compararea secvenţelor sunt mutaţiile unei singure baze şi cele determinate de *indels*. Până acum au fost tratate doar situaţiile indels.

Metodele clasice de comparare a secvenţelor de AND au presupus că secvenţele pot fi numai "mutate" prin operaţii care acţionează individual asupra nucleotidelor i.e. substituţii, inserări şi ştergeri. Mai recent însă, studii adiţionale au considerat rearanjarea evenimentelor genomice la scară largă cum ar fi inversiunile, transpoziţiile, şi translocările. În continuarie se va extinde cazul pentru indels

mai lungi şi se va descrie o problemă a inversiunii lungi de segmente ale unei secvenţe.

În timp ce indels a multe baze, fie 100, ar putea fi suma a 100 de baze indels, explicaţia probabilă este că a fost un singur eveniment. Într-un studiu de aliniament de parametri, Fitch şi Smith [FITC'83] arată că pentru anume secvenţe, sunt necesare indels mai lungi pentru a obţine aliniamentul corect. Ponderile indels-urilor mai lungi nu ar trebui să fie socotite ca sumă de indels singulare.

Fie x_k ponderea aleasă pentru un indel de k litere, $k \geq 1$. Următoarele rezultate apar în [WATE'76]. Dacă $x_1 \leq x_2 \leq \ldots$ şi d este o metrică în mulţimea elementelor secvenţelor, atunci D este o metrică pe mulţimea secvenţelor.

Algoritmul Waterman-Smith-Byers

Fie $D_{0j} = x_i$, $D_{oj} = x_j$ $D_{00} = 0$, si $D_{ij} = D(a_1a_2\ldots a_j, b_1b_2\ldots b_j)$

Atunci,

$$D_{ij} = \min\left\{D_{i-1,j-1} + d(a_i, b_j), \min_{k\geq 1}\{D_{i,j-k} + x_k\}, \min_{k\geq 1}\{D_{i-k,j} + x_k\}\right\}$$

Un algoritm corespunzător va calcula un S generalizat.
Este important a se nota faptul că timpul de calcul este mărit cu

$$\sum_{i=1}^{n}\sum_{j=1}^{m}(ij) = O(nm^2 + n^2m) \qquad (2.3.4\text{-}1)$$

Un algoritm $O(n^3)$, pentru două secvenţe de lungime $n = 1000$, este un preţ semnificativ de plătit pentru multiple indels. O abordare pentru a evita aceasta este de a presupune x_k liniar. Acesta este,

$$x_k = a + bk..$$

Pentru calculele structurii secundare această presupunere este expluatată în [WATE'76] şi [KANE'82]. Gotoh în [GOTO'82] (1982) derivă un algoritm înrudit pentru x_k liniar cu timp de execuţie $O(nm)$. Taylor [TAYL'84] vine cu mai multe rezultate pe linia deschisă de Gotoh. Desigur, după x_1 şi x_2, un asemenea x_k se comportă mai mult ca un indel singur. Este prin urmare de dorit să se extindă algoritmul, în mod special să prelucreze funcţiile indel astfel încât

$$x_k = a + b\log(k).$$

Aceasta a fost realizată în [WATE'84] şi are aplicaţie în problemele de structură secundară unde indels sunt analoge salturilor interioare şi salturilor ramificate. Algoritmul pentru x_k *concav* are timp de execuţie *O(nm).*

Problema includerii inversiunilor este foarte interesantă. Interschimbatrea a două litere adiacente este o transformare propusă cu ajutorul computerelor. Wagner [WAGN'83] a arătat că această transformare poate fi inclusă cu timpul de calcul $O(\alpha\, n\, m\, 4^{\alpha})$, unde

$$\alpha \leq \min\{4 \max d(a, b), 2x_1\}/y+1,$$

unde *y* este costul transpunerii. Includerea inversiunilor lungi pare să fie o activitate dificilă. În mod cert, aceste inversiuni au loc în secvenţele de ADN şi ar trebui incluse. O problemă relevantă pentru biologie este de a include inversiuni într-un algoritm unde există un singur cost al inversiunii plus distanţa dintre segmente. Richard et al. [REIC'73], Wong et al. [WONG'74] şi Cohen et al. [COHE'75] produc o serie de algoritmi de programare dinamică motivaţi de maşina Turing şi teoria informaţiei. Aceşti algoritmi pot fi văzuţi ca şi cazuri speciale ale algoritmilor generali de programare dinamică [WATE'84].

Notă: O exemplificare a modului de aplicare al acestor algoritmi dinamici este ilustrată în ANEXA 2.

2.3.5. Algoritmul de programare dinamică a lui Ukkonen

În timp ce algoritmul de programare dinamică $O(n^2)$ este mult mai rapid decât forţa brută a algoritmului $O(2^{2n})$, el devine productiv pentru un *n* foarte mare. E. Ukkonen [UKKO'83], [UKKO'84] vine cu algoritmi semnificativ mai rapizi pentru calcularea distanţei dintre două secvenţe. Algoritmii, evidenţiaţi în cele ce urmează, calculează distanţa *s* dintre secvenţe de lungime *n* şi *m*, de-a lungul aliniamentului, în timp şi spţiu de stocare de $O(s \cdot min\{n, m\})$ (cu spaţiu de stocare de $O(s^2)$ în unele cazuri). Dacă nu este dorit nici un aliniament, stocarea este de $O(s)$. Cel mai rău caz de comportament este echivalent cu algoritmul standard, când pentru *s* cu valoare mică îmbunătăţirea este dramatică. Dacă nu este dorită nici o distanţă mai mare de un prag *t* atunci timpul nu este mai mare de $O(t \cdot min\{n, m\})$.

Un algoritm similar în esenţă cu cel al lui Ukkonen este propus de J.W. Ficket [FICK'84], dar acesta din urmă nu este aşa de eficient în termeni de timp. (Ambii autori tratează cazul pentru indels singulare.)

Ukkonen [UKKO'83] prezintă algoritmul său pentru neportiviri, indels singulare, şi transpuneri de 2 litere. Mai jos sunt aceleaşi rezultate pentru nepotriviri şi indels multiple. Fie ca mai sus:

$$D_{ij} = \min\left\{D_{i-1,j-1} + d(a_i, b_j), \min_{k\geq 1}\{D_{i,j-k} + x_k\}, \min_{k\geq 1}\{D_{i-k,j} + x_k\}\right\},$$

şi se presupune $d(a,b) = 1$ doar dacă $a = b$. Ukkonen demonstrează o lemă cheie care se arată a fi adevărată pentru cazul general de indels multiple.

Lema U.: Pentru toţi (i, j), $D_{ij} - 1 \leq D_{i-1,\, j-1} \leq D_{ij}$.

Demonstraţie: Demonstraţia este prin inducţie pe $i+j$. Partea stângă reiese imediat din recursivitate. Dacă $D_{ij} = D_{i-1,\, j-1} + d(a_i, b_j)$ atunci urmrează $D_{ij} \leq D_{i-1,j-1}$. Altfel, fără a pierde caracterul de generalitate, se presupune $D_{ij} = D_{i-k,\, j} + x_k$. Ipoteza inducţiei implică $D_{i-k,j} \geq D_{i-(k+1),\, j-1}$ astfel încât $D_{i-k,j} \geq D_{i-(k+1),\, j-1} + x_k \geq D_{i-1,\, j-1}$.

Lema U care este elementară, este cheia metodei lui Ukkonen. Ea declară că $D_{i,i+c}$ este o funcţie a lui *i* nedescrescătoare. Aceasta implică o structură pentru matricea $D_{i,j}$: ea este de forma unei văi cu ridicături de-a lungul liniilor constante $i-j$. Cea mai joasă ridicătură este $D_{00} = 0$. Mai jos, atenţia este focalizată pe limitele schimbărilor când $D_{i,i+c} = k$ se schimbă cu $D_{i+1,\ i+1+c} = k+1$.

Se presupune că toate indels au costul de 1, i.e. $x_k = k$. Ideea de bază a algoritmului lui Ukkonen [UKKO'83] este de a începe la D_{00} =0 şi a se extinde de-a lungul $j - i = 0$ până când $D_{ii} = 1$. În general, vor exista $2k + 1$ limite ale regiunii $D_{ij} \leq k$. Fiecare limită $j - i = c$ este extinsă până când $D_{ij} = k + 1$ (pentru $i - j = c$). Extinderea limitei la $k + 1$ poate fi determinată din limitele pentru k, $k - 1$, ...şi teste ale $a_i = b_j$. Această procedură este urmată până când este atins $D_{n,m}$. Dacă $D_{n,m} = s$, este clar că nu au fost calculate mai mai mult de $(2s + 1)\min\{n, m\}$ intrări. Este suficient doar să se stocheze aceste limite astfel spaţiul necesar de stocare este $O(s^2)$.

2.3.6. Aliniamente aproape optimale

Cu toate că algoritmii localizează aliniamentele optimale, ponderile sunt determinate de utilizator. Calibrarea ponderilor prin aliniamente deja cunoscute a fi corecte poate fi făcută dar nici un set de ponderi nu poate fi considerat a fi absolut corect. Chiar dacă ponderile sunt corect alese, constrângeri biologice necunoscute pot determina ca aliniamentul adevărat să fie diferit de cel optimal, generat de computer.

Prin urmare, o problemă naturală este de a găsi toate aliniamentele din cadrul distanţei optimale. Motivaţia vine din faptul că aliniamentul corect ar trebui să fie lângă cel optimal şi că biologul sau cel care analizează secvenţele să îl poată recunoaşte. Această problemă a fost soluţionată în [WATE'83] şi primeşte un tratament mai general în Byers şi Waterman [BYER'84].

În mod formal, problema este de a găsi toate aliniamentele cu distanţa sau scorul de aliniament pâna la ε al distanţei $D_{n,m}$ dintre cele două secvenţe a şi b. Se presupune că toţi $D_{n,m}$ sunt calculaţi şi stocaţi.

La poziţia (i, j) se presupune o parcurgere înapoi (traceback) de la (n, m) la $(0, 0)$ ce poate rezulta într-un aliniament cu scor mai mic sau egal cu $D_{n,m} + \varepsilon$. Scorul de la (n, m) la (i, j) dar fără a include (i,j), este de $T_{i,j}$. $T_{i,j}$ este suma posibilului aliniament non-optimal de a atinge (i,j). De la (i,j), ca de obicei, sunt posibili trei paşi: $(i\text{-}1,j)$, $(i\text{-}1,j\text{-}1)$ şi $(i,j\text{-}1)$. Fiecare pas este într-un aliniament dorit dacă şi numai dacă

$$\begin{aligned} T_{i,j} + d(a_i, \Delta) + D_{i-1,j} &\leq D_{n,m} + \varepsilon \\ T_{i,j} + d(a_i, b_j) + D_{i-1,j-1} &\leq D_{n,m} + \varepsilon \\ T_{i,j} + d(\Delta, b_i) + D_{i-1,j} &\leq D_{n,m} + \varepsilon. \end{aligned}$$

Aliniamentele multiple aproape optimale pot fi produse prin gruparea direcţiilor neexplorate.

Un studiu al senzitivităţii aliniamentului de secvenţe la ponderi şi multiple indels a fost reralizat de Fitch şi Smith [FITC'83].

2.3.7. Metoda regiunilor

O examinare a matricelor punct (abordate în secţiunea 2.3.1) poate sugera construirea unei liste cu regiuni de potrivire. Atunci când un aliniament este doar o submulţime ordonată a unei asemenea liste, algoritmii pot fi divizaţi pentru găsirea aliniamentelor optimale. Această abordare trebuie să fi fost continuată de câteva grupuri independente deoarece este importantă găsirea de regiuni lungi de potrivire. Pentru predicţia structurii secundare, Studnicka et al. [STUD'78] urmează o asemenea direcţie. Martinez [MART'80], [MART'83] vine cu un algoritm mai matematic, care de fapt adaptează algoritmul său pentru structura secundară în aliniament de secvenţe. O încercare de descriere a acestui algoritm va fi dată în cele ce urmează.

Pentru început este necesară o listă *L* de regiuni de potrivire. O regiune *R* este definită a fi un triplet de forma (*w*; *i*, *j*), care înseamnă o potrivire a cuvântului *w* care începe la $a_i = b_j$. În alte cuvinte, dacă l = |*w*| este lungimea cuvântului,

$$a_i = b_j \ , \ a_{i+1} = b_{j+1} \ , \ \ldots\ldots\ , \ a_{i+l-1} = b_{j+l-1} \ .$$

Pentru a obţine o asemenea listă, Martinez [MART'83] mai întâi concatenează secvenţele într-una singură *S*. El foloseşte tehnica sortării rapide a lui *S* folosind ordinea lexicografică. O primă sortare grupează toate elementele egale ale lui *S* împreună. A doua sortare operează asupra fiecărui asemenea grup de elemente egale şi grupează împreună elemente care sunt urmate de elemente egale în secvenţa originală *S*. La sfârşitul celei de-a *k* sortări, două elemente ale lui *S* permutat vor aparţine la acelaşi grup dacă şi numai dacă locaţiile lor *i* şi *j* în şirul original *S* sunt astfel încât elementele la locaţiile *i*+l şi *j*+l sunt egale pentru l = 0,1,...*k*-1.

După cum este arătat în Martinez [MART'83], viteza acestei proceduri de sortare pentru a genera regiuni este, în cazul aşteptat, de ordinul *NlogN*, unde *N* este lungimea totală a secvenţelor concatenate. Procedura este, prin urmare, comparabilă în viteză cu metoda standard utilizată de calculatoare pentru construirea „arborilor poziţionali" (position trees) pentru identificarea substringurilor comune a două sau mai multe secvenţe, aşa după cum o descrie Aho, Hopcroft şi Ullman [AHO'74], dar are avantajul implementării simple.

Pentru a ilustra conceptul arborelui de poziţie fie *a* = AATAATGCS, unde *S* semnalează finalul secvenţei. Pentru fiecare *i*, *i* =1 la 8, fie subşirul *S* cel mai scurt subşir care începe la *i* care nu apare altundeva în *a*. Acest subşir este spus că identifică pe *i*. Spre exemplu, poziţia *i* = 4 este identificată de AATG. Aceste subşiruri de identificare sunt organizate într-un arbore de poziţii care reprezintă informaţia:

Poziţia	Identificând subşirul
1	AATA
2	ATA
3	TA
4	AATG
5	ATG
6	TG
7	G

8 CS
9 S
.

Cele *n* noduri terminale ale arborelui de poziţie pentru $a = a_1, a_2, ..., a_n$ constă din 1, 2, ..., *n*. Secvenţa de etichete pe muchii de la rădăcină spre nodul terminal *i* este subşirul identificator pentru poziţia *i*. Arborele de poziţie pentru lungimea secvenţei 8 de mai sus este reprezentat în Figura 2.3.7-1. Două secvenţe (sau mai multe) pot fi procesate simultan pentru a da un arbore de poziţie unde cea mai lungă potrivire poate fi găsită simplu.

O altă metodă pentru găsirea rapidă de regiuni poate să se bazeze pe conceptul de „hashing" aşa cum este folosit în problemele clasice de căutare lexicografică. Cea mai timpurie referinţă este Dumey [DUME'56]. Descrisă mai detaliat în Dumas şi Nino [DUMA'82], idea de bază a acestui concept este de a asocia cu fiecare poziţie a unei secvenţe echivalentul numeric al lui *k*–mer începând cu acea poziţie. Echivalentul numeric este obţinut cu privire la alfabetul secvenţei ce defineşte baza unui sistem de numere. Astfel, un alfabet format din patru numere dă un sistem de numere bazei patru, şi pentru *k* fix există 4^k numere posibile. Aceste numere pot fi folosite pentru a identifica poziţii ale unui vector de mărime 4^k de liste ale poziţiilor locaţiilor din secvenţă la care apare *k*-merul corespunzător. Această metodă este folosită de Wilbur şi Lipman [WILB'83] în realizarea căutarilor rapide în baze de date, şi aparent la fel de către Karlin et al. [KARL'83] pentru găsirea repetiţiilor exacte. Vectorul poate fi construit într-un timp de ordinul *N*. Cele mai lungi repetiţii, şi astfel regiuni, sunt găsite prin simpla îmbinare a tuturor repetiţiilor de lungime *k*, iar viteza pare să fie de ordinul *N log N* (sau ordinul *N* ?).

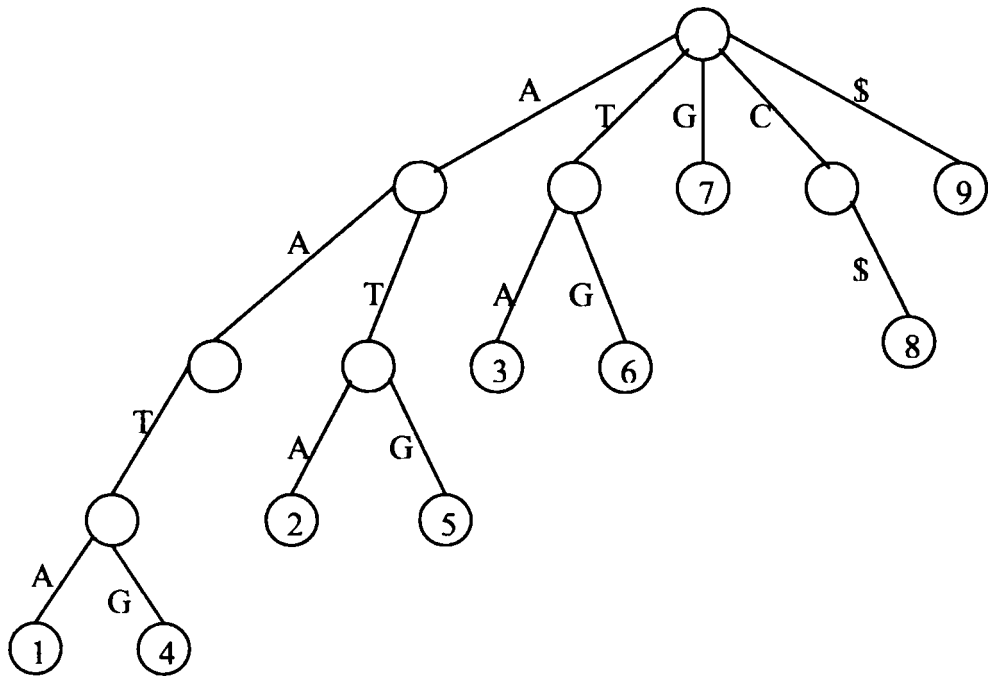


Figura 2.3.7-1. Arborele de poziţie pentru a=AATAATGC.

Fiind dată lista regiunilor găsite de oricare dintre aceste metode, ele vor fi utilizate pentru găsirea aliniamentelor optimale. Două regiuni $R_1 = (w_1; i_1, j_1)$ şi $R_2 = (w_2; i_2, j_2)$ se spune că satisfac $R_1 < R_2$ dacă $i_1 + |w_1| - 1 < i_2$ şi $j_1 + |w_2| - 1 < j_2$. $R_1 \leq R_2$ înseamnă că există $i_2 - i_1 - |w_1|$ baze ale lui a şi $j_2 - j_1 - |w_1|$ baze a lui b între regiuni, şi acel R_1 este stânga lui R_2. Trebuie dată o formulă pentru ponderea acestor baze care nu se potrivesc. Dacă sunt x baze din a şi y baze din b, fie $z(x,y)$ această pondere. Dacă nepotrivirile costă 1 şi indels costă δ cu $1 < 2\delta$, atunci o alegere rezonabilă pentru $z(x,y)$ este

$$z(x,y) = |x-y|\delta + \min\{x,y\}.$$

Formula poate fi divizată pentru alte situaţii şi $z(x,y) = x + y$ poate fi folosită.

Fiecare regiune R poate fi considerată a fi la sfârşit stânga al unui aliniament optimal A(R), începând cu R şi continuând cu a_n şi b_m. Fie D(A(R)) un scor al unui asemenea aliniament. Optimizarea urmează din
$D(A(R)) = \min\{w(k - i - |w|, l - j - |w|) + D(A(R^*)) : R=(w: i, j) < R^*=(w^*: k, l)\}$.
Algoritmii generali sunt cunoscuţi cu timpul de execuţie $O(|L|^2)$.
O implementare a acestui algoritm a fost realizată de către Martinez şi Sobel şi descrisă în Martinez [MART'83]. Ei produc de asemenea un aliniament aproape optimal prin adaptarea ideilor din 2.3.6.

2.3.8. Localizarea segmentelor lungi de potrivire

Problemele din această secţiune se centrează pe căutarea de segmente cu grad ridicat de similaritate care fac parte din două secvenţe biologice. Există două abordări de bază: căutarea pentul potriviri *lungi exacte* sau pentru potriviri *lungi inexacte*. Noi dezvoltări în teoria probabilităţilor asistă aceste căutări.

2.3.8.1. Potriviri lungi exacte

Prima metodă eficientă pentru localizarea potrivirilor exacte a fost dată de Korn et al. [KORN'77]. Abordarea lor utilizează conceptul de arbore de poziţii discutat deja. Aceeaşi metodă poate fi folosită pentru a găsi repetiţii lungi într-o secvenţă fixată. Repetiţiile vor partaja ramuri în arborele de poziţii. Spre exemplu, în Figura 2.3.7-1, poziţia 1 este identificată de AATA în timp ce poziţia 4 este identificată de AATG, care implică repetiţia „lungă" AAT. Korn et al. aranjează subşirurile de identificare într-o ordine lexicografică pentru a găsi aceste repetiţii printre celelalte lucruri. Aceasta este o aplicaţie directă şi utilă a ştiinţelor computaţionale moderne pentru analiza secvenţelor biologice şi cu precadere a celor de ADN.

Algoritmul dat în Aho et al. [AHO'74] pentru construirea arborilor de poziţii are cel mai dezavantajos timp de execuţie de $O(n^2)$ pentru o secvenţă de lungime n. Timpul de execuţie este proporţional cu numărul de vârfuri (vertices=noduri) ale arborelui. Oricum dacă literele cuvântului sunt independente şi identic distribuite atunci timpul de execuţie aşteptat este $O(n)$. De asemenea ei punctează existenţa unui algoritm care rulează în $O(n)$ pentru toate intrările.

În lucrările lui Karlin et al. [KARL'83] şi [KARL'84] se realizează analiza secvenţelor prin localizarea repetiţiilor directe lungi. Tehnica lor foloseşte o metodă de *hashing* şi localizează toate repetiţiile directe exacte.

O determinare teoretică a distribuţiei statistice a similarităţii a fost dată. Dacă $M(n, m)$ este lungimea celei mai lungi regiuni de potrivire exactă dintre două secvenţe, atunci

$$E(M(n, m)) = log((1-p)nm+\gamma/\lambda - 1/2 + r_1(n, m)+ o(1))$$

şi

$$\sigma^2(n, m) = \pi^2/6\lambda^2 + 1/12 + r_2(n, m) + o(1),$$

unde $p = P$ (două potriviri aleatoare de nucleotide), $log = log_{1/p}$, $\gamma = 0.577...$ este constanta Euler-Mascheroni, $\lambda=log(1/p)$ şi $r_1(n, m)$ şi $r_2(n, m)$ sunt mici. Se observă că $\sigma (n, m)$ este esenţial independentă de n şi m. Acest rezultat este din Arratia et al. [ARRA'84a]. Karlin et al. [KARL'84] declară un rezultat ce diferă doar printr-o constantă de acesta. De asemenea se poate consulta Arratia şi Waterman [ARRA'84b] pentru legi înrudite ale numerelor mari. Potrivirile dintre secvenţe sunt considerate semnificative dacă depăşesc $E(M(n, m)) + 2\ \sigma$.

Collins şi Coulson [COLL'84] vin cu un algoritm de procesare paralelă pentru a produce matricea punct a tuturor potrivirilor de lungime mai mare sau egală cu un prag fixat. Implementarea acestora acceptă secvenţe de până la 49152 baze şi este un indicator al rezultatelor aplicării noii tehnologii acestor probleme. Rezultatele probabilităţilor anterioare pot fi folosite pentru fixarea pragului.

2.3.8.2. Potriviri lungi inexacte cu ajutorul programării dinamice

Se consideră problema localizării segmentelor similare dintre două secvenţe fără a cere segmentelor să fie identice. Primul care a considerat această problemă a fost Sellers [SELL'79], [SELL'80]. El defineşte un interval I pentru a care seamănă cel mai bine cu b la modul general dacă $D(I, b) \leq D(J, b)$ pentru toate segmentele J ale lui a. Sunt necesare ambele matrice: „forward" şi „backward". Una dintre probleme este găsirea segmentelor de potrivire dorite din multele produse. Sankoff şi Kruskal [SANK'83] estimează că n^4 potriviri rezultă din secvenţe de lungime n. Goad şi Kanehisa [GOAD'82] modifică tehnica lui Sellers. Erickson şi Sellers [ERIC'83], discută pe larg această metodă şi dau două aplicaţii non-triviale. În acest articol, Sellers rafinează analiza şi vine cu un alt algoritm pentru găsirea „celor mai bune segmente". El foloseşte conceptul de densitate de potrivire a lui Goad şi Kanehisa [GOAD'82] pentru găsirea celor mai lungi segmente ale unei densităţi de potivire recomandate. Similaritetea calculează

AAAA A

versus

AAAA A

ca patru potriviri versus una, atata timp cât distanţa este zero în fiecare caz. Pentru a filtra porţiuni ale secvenţei cu potrivire negativă, se defineşte H_{ij} a fi similaritatea maximă a două segmente care se termină în a_i şi b_j, sau zero, indiferent care este mai mare.

Se defineşte:

$$H_{ij} = \max \{0, S(a_x\, a_{x+1} \ldots a_j\, ,\, b_y\, a_{y+1} \ldots b_j) : 1 \le x \le i \text{ şi } 1 \le y \le j\}.$$

În acest articol, Seller sugerează faptul că cele mai bune segmente sunt în esenţă segmente de maximă similaritate care:
(1) au similaritate non-negativă;
(2) au scoruri cel puţin la fel de largi ca orice alt segment cu căi care se intersectează şi
(3) au scoruri cel puţin la fel de mari ca nişte valori izolate.
Smith şi Waterman [SMIT'81a,b] au recomandat o procesare secvenţială a matricei *H*, găsind aliniament cu cele mai mari, apropiat de cele mai mari, etc. valori de similaritate cu direcţii neintersectate. După stabilirea algoritmului pentru *H*, [WATE'84] vine cu un nou şi mai complet algoritm pentru găsirea de aliniamente satisfăcând recomandările (1), (2) şi (3) ale lui Seller.

Algoritmul Smith – Waterman

Fie $H_{i0} = H_{0j} = 0$ pentru $1 \le i \le n$ şi $1 \le j \le m$. Atunci

$$H_{ij} = \max\left\{ H_{i-1,j-1} + s(a_i, b_j),\ \max_{1 \le k \le i}\{H_{i-k,j} - x_k\},\ \max_{1 \le k \le j}\{H_{i,j-k} - x_k\},\ 0 \right\}.$$

Valorile care pot fi folosite pentru ponderi sunt

$$s(x,y) = \begin{cases} 1 & \text{daca } x = y \\ -1/3 & \text{daca } x \neq y \end{cases}$$

şi

$$x_k = 1 + k/3\ .$$

O reducere în timpul de calculare de la $O(n^3)$ la $O(n^2)$ pentru funcţii liniare sau concave de ştergere pot fi obţinute ca în secţiunea 2.3.4.

Când se construieşte matricea *H*, se pun în stivă toate (i, j, Y) cu $Y = H_{ij}$ şi $H_{ij} \ge C$ = valoare extremă (cut-off value). Stiva este ordonată de „>", unde

$(i, j) > (k, l)$ dacă:

(1) $H_{ij} > H_{kl}$ sau
(2) $H_{ij} = H_{ij}$ şi $i + j < k + l$ sau
(3) $H_{ij} = H_{ij}$, $i + j = k + l$ şi $i < k$.

În timpul parcurgerii înapoi, pentru unele intrări în stivă aliniamentele multiple sunt rezolvate în modul următor: dacă două aliniamente multiple se termină în (i, j), şi (k, l) o produce pe aceea care se termină în (i, j), dacă: (1) $i + j < k + l$, sau dacă (2) $i + j = k + l$ atunci şi $i > k$. Odată ce o parcurgere înapoi este completată cu succes, intrările aliniamentelor în matrice sunt multiplicate cu -1, (i.e. aliniamentul produs), şi intrarea corespunzătoare în stivă este înlăturată din stivă. Elementele negative ale lui *H* nu sunt folosite în vre-unul dinte aliniamentele viitoare.

Dacă o intrare în stivă are un element negativ corespunzător al matricei, se înlătură şi se merge mai departe. Dacă o parcurgere înapoi întâlneşte un element negativ al matricei ea nu se va putea continua. Dacă cel mai bun aliniament generat de departe are scorul = $Y \geq C$, atunci (i, j, Y) trebuie înlocuit în stiva ordonată.

Boswell and McLachlan [BOSW'84] sugerează de asemenea folosirea valorilor similarităţii pentru localizarea segmentelor similare. Matricea lor directă (forward) este calculată de către formula

$$F(i, j) = s(a_i, b_j) + \lambda \max \{F_{i-2, j-1} - w_1 , F_{i-1, j-1} , F_{i-1, j-2} - w_1 \}. \qquad (2.3.8.2\text{-}1)$$

Matricea inversă R este găsită prin inversarea secvenţelor. Atunci

$$M(i, j) = F(i, j) + R(i, j) - s(a_i, b_j).$$

Idea este că $M(i, j)$ este suma a $s(a_i, b_j)$ plus căile ponderate cele mai bune care se extind în orice direcţie. Parametrul $\lambda \in (0, 1)$ este un factor geometric de amortizare (damping).

Problema distingerii valorilor statistice semnificative a

$$H^* = \max_{i,j} H_{ij} \qquad (2.3.8.2\text{-}2)$$

este în mod evident importantă.

Arratia et al. [ARRA'84a] a arătat că lungimea $M(n,m)$ a celei mai lungi potriviri întreruptă de k nepotriviri satisface relaţia:

$$E(M(n, m)) = \log(nm) + k \log(nm) + (k+1)\log(1-p) - \log(k!) + k + \gamma/\lambda - 1/2 + r_1(n, m) + o(1)$$

şi

$$\sigma^2(n, m) = \pi^2/6\lambda^2 + 1/12 + r_2(n, m) + o(1),$$

unde $p = P$ (două potriviri aleatoare de nucleotide), $\log = \log_{1/p}$, $\gamma = 0.577...$ este constanta Euler-Mascheroni, $\lambda = \log(1/p)$ şi r_1, r_2 sunt mici. Se poate folosi spre exemplu $H_{ij} \geq E(M(n,m)) + 2\sigma(n,m)$ pentru a decide care H_{ij} sunt de intreres pentru a fi produse. Se observă că $\sigma(n,m)$ este încă odată esenţial independentă de n şi m.

Într-un studiu empiric, Smith et al. [SMIT'84] arată că, pentru

$$s(x, y) = \begin{cases} 1 & \text{daca } x = y \\ -0.9 & \text{daca } x \neq y \end{cases}$$

şi

$$x_k = \begin{cases} 2 & \text{daca } k = 1 \\ \infty & \text{daca } k > 1 \end{cases}$$

valorile lui $E(H^*)$ şi σ sunt

$$E(H^*) = 2.5\ log(nm) - 9$$
$$\sigma = 1.78$$

unde $log = log_{1/p}$ ca mai sus.

Un rezultat asimptotic cum că legea lui $log(n)$ este de acord cu indels precum şi cu nepotrivirile este dată în Arratia şi Waterman [ARRA'84a] (1984). Studiul empiric raportat mai sus este o evidenţă a robusteţii acestei distribuţii.

2.3.8.3. Potriviri lungi inexacte folosind regiuni

Un algoritm pentru găsirea potrivirilor lungi inexacte şi care nu foloseşte programarea dinamică este propus de Korn et al. [KORN'77]. El are câteva dezavantaje serioase care sunt punctate mai jos, dar este o metodă folositoare şi a fost larg răspândită printre analiştii secvenţelor biologice, a se vedea Queen et al. [QUEE'82].

Algoritmul începe la poziţia (i,j) unde $a_i = b_j$ şi $a_{i+1} = b_{j+1}$. Această potrivire de lungime doi este extinsă într-o manieră recursivă, unde regulile pentru extindere sunt:

(1) următoarele baze se potrivesc (i.e. $a_{i+2} = b_{j+2}$),
(2) prin ştergerea a 1, 2 sau 3 baze din secvenţa a există o execuţie a 3 potriviri,
(3) prin ştergerea a 1, 2 sau 3 baze din secvenţa b există o execuţie a 3 potriviri sau
(4) prin nepotrivirea lui a_{i+2} şi b_{j+2}, două dintre următoarele 3 perechi se potrivesc. Programul lor nu caută perechi (i, j) unde (a_i, b_j) sunt deja într-o regiune de identificată.

Sankoff şi Kruskal [SANK'83] remarcă faptul că această metodă, când compară AACAAA şi AAAAA, va găsi AACAAA şi AAΔAAA, dar cu secvenţele inversate nu va găsi această regiune. Aceasta ar putea fi o proprietate nedorită. Sankoff şi Kruskal au remarcat de asemenea că AACCGT şi AACGT vor produce

AAC CGT
şi
AAC CGT

Care au o bază în comun, în loc de

AACCGT AACCGT
şi
AACΔGT AACΔGT

Timpul de execuţie al acestui algoritm pentru două secvenţe de lungime n este proporţional cu n^2. Constanta este mai mare decât aşteptarea unei variabile geometrice aleatoare. Pentru baze egal probabile aceasta înseamnă: constanta va

depăşi 4. De când algoritmii de programare dinamică au o constată de 3, aceasta sugerează putenic folosirea algoritmilor matematici mai riguroşi.
Metoda Wilbur-Lipman descrisă în secţiunea destinată căutării eficiente în baze de secvenţe (3.3.3) este de asemenea o metodă a regiunilor ce poate fi aplicată acestei probleme.

2.3.9. Aliniamentul perechilor folosind modele Markov ascunse(HMMs)

Folosind teoria HMM, a fost dezvoltat un anume HMM care modelează perechi aliniate de secvenţe. Această abordare oferă unele moduri de estimare a acurateţii unui aliniament şi calculează scorul care exprimă similaritatea fără a referi un aliniament particular. HMM-urile specifice se bazează pe conceptul de FSA (Finite State Automata) care sunt folosite pentru aliniamentul perechilor de secvenţe ca descriptori convenabili ai unor algoritmi de programare dinamică mai complecşi. Este executată o transformare în care toate alternativele sunt ponderate probabilistic. Acest punct de vedere oferă posibilitatea de a calcula similaritatea a două secvenţe independent de orice aliniament specific. Algoritmii incluşi în această categorie cuprind: algoritmul Viterbi pentru perechi de HMM, calcularea „înainte" (forward calculation) pentru perechi de HMM şi calcularea „înapoi" (backward calculation) pentru perechi de HMM. O descriere detaliată a acestora poate fi găsită în [DURB'98].

Utilitatea profilelor HMM este extinsă la căutarea familiilor de secvenţe, permiţând celor preocupaţi de biologia computaţională să infere aproape de trei ori mai multe omologii decât oricare alt algoritm simplu pentru perechi de secvenţe [PARK'98].

Dintre toolurile disponibile care folosesc profile de HMM sunt SAM[15] (Sequence Alignment and Modelling System) şi cele accesibile pe serverul HMMER[16], care este o implementare de software specializat pe profile HMM pentru analiza secvenţelor de proteine distribuită gratuit.

În aliniamentul de perechi folosind modele Markov ascunse (HMMs) se presupun 3 stări, în care M corespunde stării de potrivire şi două stări ce corespund inserărilor şi care vor fi notate cu X şi Y în următoarea figură.

[15] http://www.cse.ucsc.edu/research/compbio/HMM-apps/HMM-applications.html
[16] http://bioweb.pasteur.fr/seqanal/motif/hmmer-uk.html

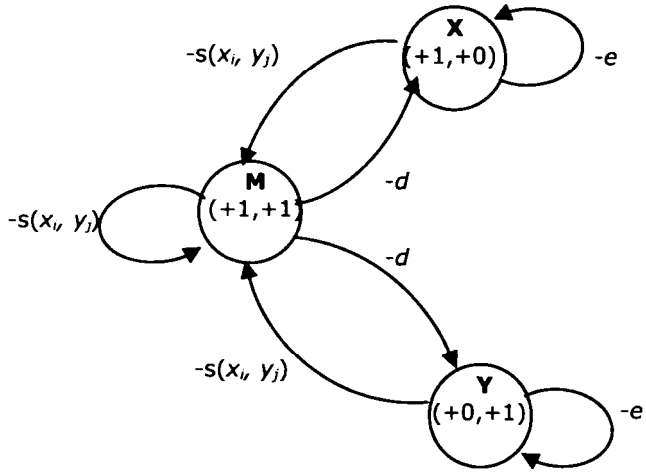


Figura 2.3.9-1. Diagrama unui automat cu stări finite pentru un aliniament cu gap

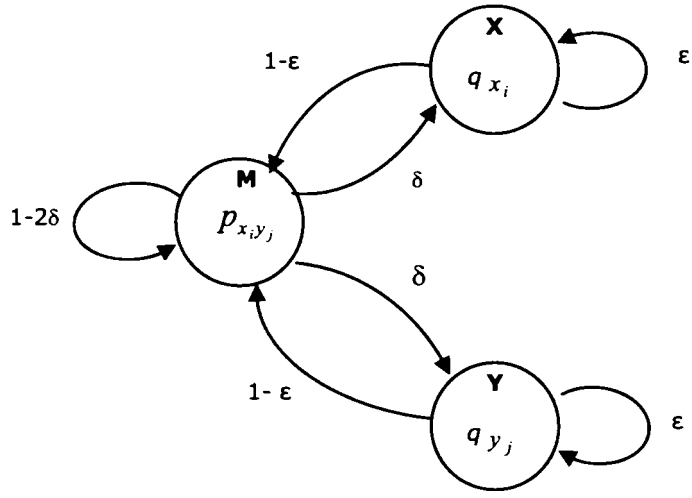


Figura 2.3.9-2 . Modelul probabilistic corespunzător diagramei unui automat cu stări finite pentru un aliniament cu gap din figura anterioară.

Relaţiile de recurenţă pentru actualizarea valorilor acestor stări în matricea de programare dinamică sunt:

$$V^{M}(i,j) = s(x_i, y_j) + \max\begin{cases} V^{M}(i-1, j-1), \\ V^{X}(i-1, j-1), \\ V^{Y}(i-1, j-1); \end{cases} \qquad (2.3.9\text{-}1)$$

$$V^X(i,j) = \max\begin{cases} V^M(i-1,j) - d, \\ V^X(i-1,j) - e; \end{cases}$$

$$V^Y(i,j) = \max\begin{cases} V^M(i,j-1) - d, \\ V^Y(i,j-1) - e. \end{cases}$$

Aceste ecuaţii sunt corespunzătoare aliniamentului global. Pentru aliniamentul local se intervine cu observaţiile adecvate privind modificările necesare.

Aşa după cum se poate observa din figurile anterioare, sunt necesare două seturi de schimbări asupra unui FSA pentru a-l transforma într-un HMM. Mai întâi trebuie date probabilităţi pentru emiterea de simboluri din stări precum şi pentru tranziţii între stări. Spre exemplu, starea M are distribuţia probabilităţii de emisie p_{ab} pentru emiterea unei perechi aliniate a:b, şi stările X şi Y vor avea distribuţii q_a pentru emiterea simbolului a opus unui gap. Deoarece starea X emite simboluri x_i din secvenţa x, se va scrie q_{x_i} în interiorul cercului ce reprezintă starea X. De asemenea, probabilităţile de tranziţie dintre stări, trebuie să satisfacă cerinţa ca suma tuturor probabilităţilor pentru toate tranziţiile care pleacă dintr-o stare să fie unu. Permiţând simetria, există doi parametri liberi pentru probabilităţile de tranziţie între cele trei stări principale. Se notează forma de tranziţie de la M la o stare de inserţie (X sau Y) cu δ, iar probabilitatea rămânerii în starea de inserare cu ε.

Oricum, modelul rezultat din figura a doua nu generează un model complet care să ofere distribuţia de probabilităţi peste toate secvenţele posibile. Pentru aceasta este necesară definirea unei stări de Begin şi End aşa cum o ilustrează Figura 2.3.9-3. Ca efect, acestea formalizează condiţiile de iniţializare şi terminare necesare pentru algoritmii de programare dinamică [DURB'98]. După cum se explică în [DURB'98] aranjamentele mai complexe a stărilor de Begin şi End pot corespunde aliniamentelor locale sau altor tipuri de aliniament. Adăugarea unei stări End explicite introduce nevoia unui alt parametru, probabilitatea unei tranziţii în starea de End, care acum se presupune a fi aceeaşi pentru fiecare din M, X şi Y şi este notată cu τ. Ca efect, aceasta va determina lungimea medie a unui aliniament dintr-un model. Pentru moment, se vor stabili tranziţiile de la starea Begin ca fiind aceleaşi cu cele de la starea M (se putea spune şi că se porneşte în M, dar se recurge la această formă pentru a clarifica faptul că atât iniţializării cât şi terminării i se pot da considerente independente.)

Aceasta metodă ne oferă un model probabilistic foarte apropiat de HMM. Diferenţa constă în faptul că în loc să se emită o singură secvenţă se emite un aliniament de perechi. Modelul astfel construit este referit ca *pereche* HMM pentru a fi distins de alte timpuri standardizate care emit secvenţe singulare.

Aşa cum un HMM standard poate genera o secvenţă, această *pereche* HMM poate genera o pereche aliniată de secvenţe. Aceasta se realizează pornind din starea de Begin şi ciclând asupra următorilor doi paşi: (1) ia următoarea stare în conformitate cu distribuirea probabilităţilor de tranziţie părăsind starea curentă; (2) ia simbolul unei perechi ce va fi adăugată aliniamentului în conformitate cu

distribuţia de emisie în noua stare. Procesul se opreşte când o tranziţie este făcută în starea de End.

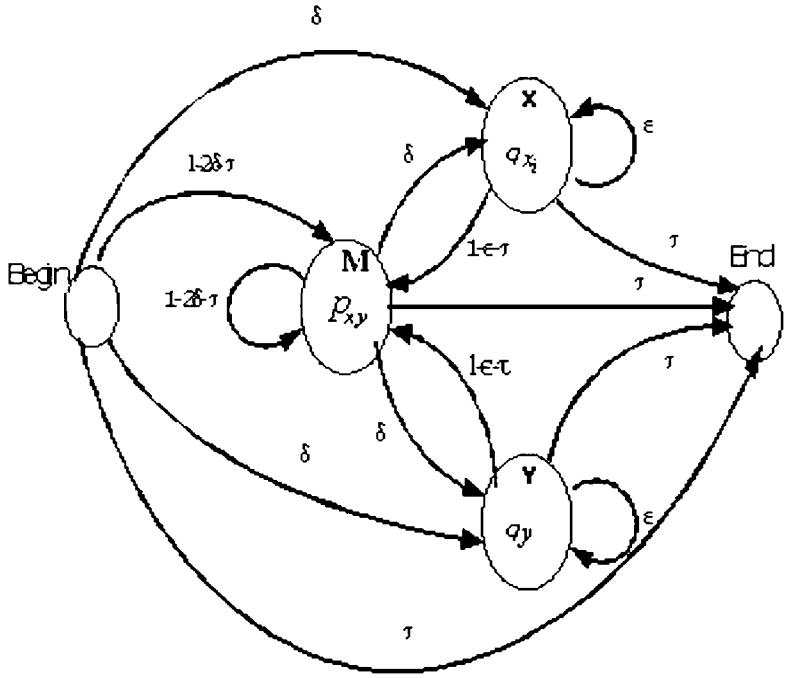


Figura 2.3.9-3. Versiunea probabilistică integrală a figurii anterioare

Deoarece fiecare pas are probabilităţi se poate, de asemenea, păstra probabilitatea totală de generare a unui aliniament particular pe care l-am produs. Acesta este doar produsul probabilităţilor fiecărui pas individual.

2.3.9.1. Cea mai probabilă cale - aliniamentul FSA optimal

Algoritmul Viterbi permite găsirea celei mai probabile căi într-o pereche HMM fiind date secvenţele x şi y. Forma corectă pentru perechea globală HMM din Figura 2.3.9-3 este după cum urmează. Pentru a face ecuaţia mai simplă, se defineşte starea de Begin cu M. De asemenea, se folosesc simboluri cu litere mici $v^*(i,j)$ pentru valorile de probabilitate şi cu litere mari $V^*(i,j)$ scorurile log-odds (care simbolizează valorile probabilistice asociate, calculate de regulă ca logaritm natural sau în baza 2 din $q_{ij}/p_i\ p_j$ (a se vedea detalii în secţiunea 2.1.2.3.) Algoritmul Viterbi în termeni probabilistici este următorul.

Algoritmul Viterbi pentru perechi HMMs

Iniţializare:

$V^M(0, 0)=1$. Toţi ceilalţi $v^*(i, 0)$, $v^*(0,j)$ sunt setaţi la 0.

Recurenţa: $i=1,...,n$, $j=1,...,m$;

$$v^M(i,j) = p_{x_i y_j} \max \begin{cases} (1-2\delta-\tau)v^M(i-1,j-1), \\ (1-\varepsilon-\tau)v^X(i-1,j-1), \\ (1-\varepsilon-\tau)v^Y(i-1,j-1); \end{cases}$$

(2.3.9.1-1)

$$v^X(i,j) = q_{x_i} \max \begin{cases} \delta v^M(i-1,j), \\ \varepsilon v^X(i-1,j); \end{cases}$$

$$v^Y(i,j) = q_{y_j} \max \begin{cases} \delta v^M(i,j-1), \\ \varepsilon v^X(i,j-1); \end{cases}$$

Terminare:

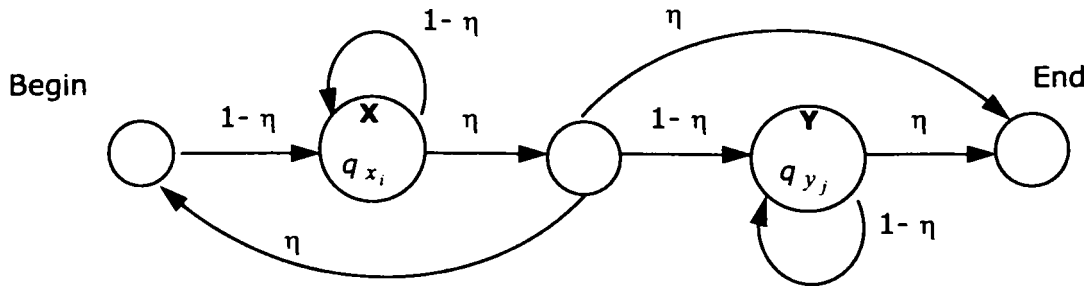
$$v^E = \tau \max\left(v^M(n,m), v^X(n,m), v^Y(n,m)\right)$$

Pentru găsirea celui mai bun aliniament se pastrează pointerii şi se parcurge drumul înapoi. Desigur, pentru obţinerea aliniamentului însuşi se păstrează reziduurile emise la fiecare pas din traiectorie pe timpul parcurgerii înapoi precum şi secvenţa de stări.

Cu toate că este evident faptul că ecuaţiile de recurenţă ale algoritmului Viterbi pentru perechi HMM au aceeaşi formă ca şi cele pentru versiunea automatului cu stări pentru aliniament de perechi, este utilă urmărirea formei exacte de corespondenţă.

Mai întâi trebuie transformate în rapoarte log-odds cu raportare la modelul aleator. De fapt acum se are un model probabilistic complet pentru aliniamentul în cauză şi trebuie de asemenea unul pentru modelul aleator cu o condiţie de terminare corespunzătoare. Pâna acum s-a ignorat faptul că modelul aleator nu poate produce secvenţe de lungime variabilă într-o formă probabilistică adecvată.

În figura de mai jos este un nou model aleator care de asemenea este o pereche HMM.



Stările principale sunt X şi Y, care emit cele două secvenţe, independente una de cealaltă. Fiecare are un salt înapoi la ea însăşi cu probabilitatea (1- η). La fel ca stările Begin şi End există, de asemenea, o stare de silent/tăcere între X şi Y, indicată de un cerc mai mic. Aceasta nu emite vre-un simbol dar este folosită pentru a acumula intrări din ambele stări X şi Y. Definit în acest fel, modelul permite secvenţe x sau y de lungime zero, la fel cum o permite şi modelul de perechi HMM în Figura 2.3.9-3 şi generează o formă simplă pentru distribuţia modelului aleator asupra secvenţelor. Probabilitatea unei perechi de secvenţe *x* şi *y* conform acestui model este

$$\begin{aligned} P(x,y|R) &= \eta(1-\eta)^n \prod_{i=1}^{n} q_{x_i} \eta(1-\eta)^m \prod_{j=1}^{m} q_{x_j} \\ &= \eta^2 (1-\eta)^{n+m} \prod_{i=1}^{n} q_{x_i} \prod_{j=1}^{m} q_{x_j}. \end{aligned} \qquad (2.3.9.1\text{-}2)$$

Acum se doreşte alocarea termenilor în această expresie celor care constituie probabilitatea aliniamentului Viterbi, astfel încât raportul de probabilitate asociat pentru întregul aliniament poate fi exprimat ca un produs al rapoartelor de probabilitate a termenilor individuali (şi corespunzători, astfel încât raportul de probabilitate al aliniamentului este o sumă a termenilor probabilistici). Aceasta se face prin alocarea unui factor (1- η) şi a factorului corespunzător q_a la fiecare reziduu care este emis într-un pas Viterbi. Astfel, tranziţiile de potrivire vor fi alocate $(1-\eta)^2 q_a q_b$ unde *a* şi *b* sunt două reziduuri care se potrivesc, iar stările de inserare $(1-\eta)q_a$, unde *a* este reziduul inserat. Deoarece calea Viterbi trebuie sa ţină cont de toate reziduurile, vor fi folosiţi exact (*n*+*m*) termeni, şi toate din eq. 2.3.9.1-2 cu excepţia factorului iniţial de η^2.

În termeni de probabilităţi asociate, acum se poate calcula un model aditiv cu scorul emis de probabilităţile log-odds şi scorurile de tranziţie log-odds. În practică aceasta este cea mai normală cale de a implementa HMMs. Din aceasta, este posibilă combinarea scorurilor de emitere cu cele de tranziţie după exemplul următor:

$$\begin{aligned} s(a,b) &= \log \frac{p_{ab}}{q_a q_b} + \log \frac{(1-2\delta-\tau)}{(1-\eta)^2}, \\ d &= -\log \frac{\delta(1-\varepsilon-\tau)}{(1-\eta)(1-2\delta-\tau)}, \\ e &= \log \frac{\varepsilon}{1-\tau}, \end{aligned} \qquad (2.3.9.1\text{-}3)$$

pentru a produce scoruri care corespund termenilor standard folosiţi în aliniamentul de secvenţe prin programare dinamică. A se nota: contribuţia lui q_a la *d* şi *e* este inutilă datorită factorilor anulaţi din modelele Viterbi şi aleatoare. De asemenea, pentru anihilarea diferenţelor în tranziţiile ce provin din stările de potrivire şi gap a fost necesară ceva îndemânare pentru exprimarea lui *s* şi *d*. Se întenţionează folosirea lui *s*(*a*, *b*) ca scor pentru fiecare potrivire, indiferent dacă este urmat de o

altă potrivire sau o inserare. Cu scopul de a face această strategie să funcţioneze corect, s-a construit în *d* o ajustare care să corecteze diferenţele în scorul de potrivire când revine dintr-o inserare. Aceasta presupune că termenii matricei de programare dinamică nu mai corespund exact rapoartelor probabilistice de asociere (log-odds ratios) de a fi în aceleaşi stări, cu toate că rezultatul final va fi corect.
Astfel, acum se poate oferi versiunea log-odds a algoritmului de aliniament Viterbi într-o formă care arată ca algoritmul dinamic clasic de aliniere pentru perechi de secvenţe.

Aliniamentul log-odds optimal

Iniţializare:
$V^M(0, 0)=2 \log \eta$, $V^X(0, 0) = V^Y(0, 0)= -\infty$.
Toţi $V^*(i, +1)$, $V^*(-1, j)$ sunt setaţi la $-\infty$.

Recurenţa:
$i = 0,\ldots,n$, $j= 0,\ldots,m$ mai puţin (0, 0);

$$v^M(i,j) = s(x_i, y_j) + \max\begin{cases} V^M(i-1,j-1), \\ V^X(i-1,j-1), \\ V^Y(i-1,j-1); \end{cases} \qquad (2.3.9.1\text{-}4)$$

$$V^X(i,j) = \max\begin{cases} V^M(i-1,j) - d, \\ V^X(i-1,j) - e; \end{cases}$$

$$V^Y(i,j) = \max\begin{cases} V^M(i,j-1) - d, \\ V^X(i,j-1) - e; \end{cases}$$

Terminare:

$$V = \max\left(V^M(n,m), V^X(n,m) + c, V^Y(n,m) + c\right)$$

Acestea sunt identice cu relaţiile 2.3.9-1 cu excepţia constantei $2\log\eta$ la iniţializare, şi constantei $c= log(1-2\delta -\tau) -log (1- \varepsilon - \tau)$ la terminare, care este necesară pentru corectarea ajustării descrise mai sus în *d*.

Procedura, după cum este descrisă, arată cum pentru orice pereche HMM de tipul din Figura 2.3.9-3 se poate deriva un FSA echivalent pentru a obţine cel mai probabil alinmiament. Aceasta ne permite observarea unei interpretări probabilistice riguroase pentru termenii folosiţi în aliniamentul secvenţelor. Pentru a parcurge invers acest proces, adică de la un algoritm de programare dinamică exprimat via FSA la perechi HMM este mai complicat. În acest caz ar fi nevoie în general de un nou parametru λ care să acţioneze ca un factor global de scalare pentru scoruri, şi pentru orice set de scoruri date pot fi constrângeri în alegerea lui η şi τ.

2.3.9.2. Perechi de HMM pentru aliniament local

Modelul prezentat în Figura 2.3.9-3 este potrivit pentru găsirea unei potriviri globale între secvenţe. După cum s-a consatat în [DURB'98], multe dintre căutările cele mai senzitive pentru perechi de secvenţe sunt locale. De obicei, când se introduce un algoritm de aliniament local sau alte variante cum ar fi algoritmi de repetare sau suprapunere se explică în termeni de schimbări în ecuaţiile de actualizare şi condiţii de limită. Toate acestea sunt făcute explicit în formalismul perechilor HMM prin adăugarea de stări şi tranziţii. Astfel se poate defini un model de perechi HMM pentru fiecare variantă. În Figura 2.3.9.2-1 este prezentat un model pentru aliniament local. Acesta arată mai complicat decât modelul global din Figura 2.3.9-3, dar este construit din bucăţi simple şi într-un mod direct.
Un model probabilistic complet trebuie să ţină seama de toate secvenţele *x* şi *y*: nu numai de aliniamentul local dintre x şi y ci şi de secvenţele nealiniate care le flanchează. Prin urmare pot fi adăugate secţiuni de modele suplimentare înainte şi după segmentul de potrivire cu trei stări din Figura 2.3.9-3. Fiecare segment alăturat este o copie a modelului complet fundamental, deoarece secvenţele în regiunile laterale sunt nealiniate. Cei mai mulţi termeni în contribuţiile probabilistice a acestor secţiuni vor fi anulate cu termenii echivalenţi în modelul aleator când se calculează scorurile log-odds ale unei potriviri în comparaţie cu modelul aleator, lăsând numai scorul local de potrivire din partea centrală a modelului, şi unii termeni suplimentari. Modele similare mixte pot fi construite pentru modele de suprapunere şi repetiţie şi hibride (a se vedea cap 2 din [DURB'98]).

Având perechi HMM se poate face mai mult decât oferirea unei alternative raţionale pentru aliniament standard prin programare dinamică. Şi anume, se poate merge la discutarea semnificaţiei potrivirilor. Spre exemplu, atunci când similaritatea este scăzută este dificilă identificarea aliniamentului corect şi testarea semnificaţiei. Astfel, există diverşi algoritmi şi metode care se dezvoltă pe principiile de bază ale perechilor HMM. A se vedea [DURB'98].

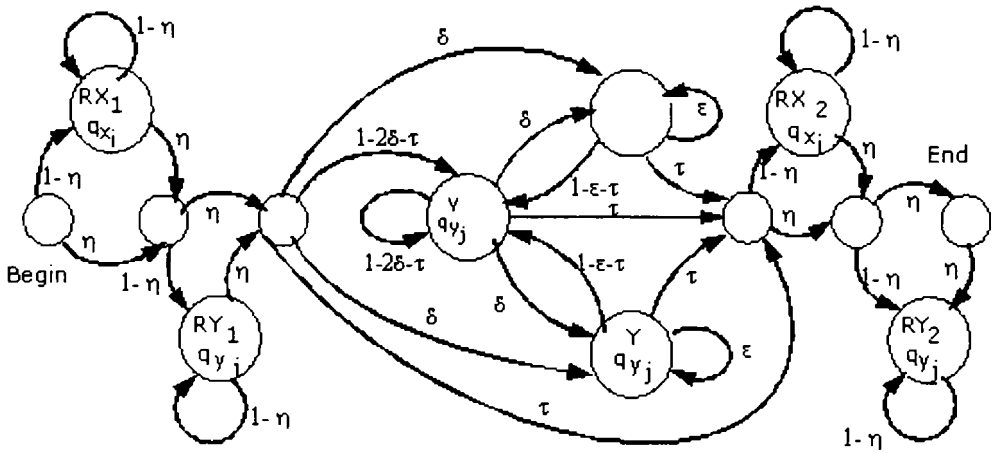


Figura 2.3.9.2-1: O pereche HMM pentru aliniament local. Aceasta este compusă din modelul global (stările M, X şi Y) mărginite de două copii ale modelului aleator (stările RX_1, RY_1 şi RX_2, RX_2).

2.3.10. Aliniament multiplu de secvenţe

Tehnicile pentru aliniamentul multiplu al secvenţelor sunt folosite pentru a recunoaşte/caracteriza familiile de proteine şi a identifica regiuni partajate de omologie de-a lungul a multiple secvenţe (aceasta se întâmplă în general când căutarea unei secvenţe relevă omologii la unele secvenţe). În plus, aceste tehnici sunt importante pentru determinarea secvenţei de consens (***consensus sequence***), adică secvenţa ideală pentru interacţiunea cu o proteină regulatoare a câtorva secvenţe aliniate şi predicţia structurilor secundare şi terţiare a secvenţelor noi. În concluzie, ele sunt considerate ca un pas de bază preliminar în evoluţia analizei moleculare cu scopul construirii arborelui filogenetic pe baza metodelor filogenetice.

Metodele de grupare tradiţionale pot executa automat aliniamente dar ele sunt subordonate unor restricţii. Adesea ele necesită prelucrări suplimentare şi interacţiune cu utilizatorul. În unele cazuri mai dificile este necesară chiar şi validarea vizuală. În [BART'96] au fost identificate diferite categorii de tehnici de aliniament multiplu:

- Extensii ale algoritmuli de programare dinamică pentru perechi de secvenţe;
- Extensii ierarhice a aliniamentului de perechi de secvenţe;
- Metode pentru segmente;
- Metode pentru consens sau 'regiuni'.

Prima categorie îşi are originea în sugestia autorilor algoritmului original de programare dinamică pentru perechi de secvenţe şi anume, că metoda lor poate fi utilizată pentru a alinia mai mult de două secvenţe [NEED'70], [WATE'76]. În practică, aceasta înseamnă extinderea spaţiului de stocare de N ori (unde N este numărul de secvenţe disponibile) şi datorită acestei extensii a fost aplicat la pâna la trei secvenţe. Una dintre aplicaţiile timpurii ale programării dinamice este realizată de Sankoff [SANK'76] care aliniază 9 secvenţe. Algoritmul său construieşte o structură de arbore iar problema alinierii simultane a 9 secvenţe este descompusă în definirea a 7 aliniamente de trei triplete. Aliniamentele sunt executate secvenţial parcurgând arborele de jos în sus. Nu există însă tooluri larg cunoscute care să utilizeze această metodă.

Dintre abordările destinate aliniamentelor multiple, cea mai practică şi mai populară este extensia ierarhică a aliniamentului de perechi. Principiul esenţial constă în faptul că aliniamentele multiple pot fi efectuate prin aplicarea succesivă a metodelor pentru perechi de secvenţe. Paşii obişnuiţi, aşa cum sunt descrişi în [BART'96], sunt:

1. Se compară toate secvenţele după metoda aplicată perechilor. Vor fi $N(N-1)/2$ perechi pentru un număr N de secvenţe date.
2. Se generează o ierarhie reprezentând toate datele aflate sub formă de perechi. Aceasta poate fi prezentată sub forma unui arbore binar sau a unei simple liste ordonate.
3. Se construiesc aliniamentele multiple aliniind la început cea mai « populară » pereche de secvenţe apoi perechea cea mai similară şi aşa mai

departe. Odată ce a fost aliniată o pereche de secvenţe aceasta rămâne fixă. Astfel, pentru un set de secvenţe A,B,C,D, după ce a fost aliniat A cu C şi B cu D, aliniamentul lui A, B, C, D este obţinut prin compararea aliniamentelor lui A şi C cu acela al lui B şi D.

Cel mai popular tool care se încadrează în această categorie este CLUSTAL W şi care este accesibil publicului.

Metodele segment sunt identificate în [BACO'86] unde problema aliniamentului a *N* secvenţe este rezolvată considerând aliniamentul într-o ordine specifică. Mai întâi, secvenţa unu este comparată cu secvenţa doi şi primele M perechi de segmete calculate ca cele mai similare sunt stocate. Următoarea secvenţă este comparată cu segmentele selectate iar cele mai ridicate scoruri dintre cele trei secvenţe sunt reţinute. Continuând acest proces se obţine o listă de M aliniamente (i.e. perechi de segmente) cu cele mai ridicate scoruri dintre cele *N* secvenţe. Barton în [BART'96] consideră că acest algoritm este foarte promiţător pentru identificarea similarităţilor importante dintre segmete scurte de secvenţe. Metoda însă nu furnizează un aliniament general al secvenţelor şi nu consideră gap-urile în mod explicit. În [JOHN'86], autorii modifică algoritmul reducând numărul comparaţiilor segment-cu-segment care trebuie executat progresiv. Ei restricţionează evaluarea aliniamentului dintre segmente într-o "fereastră" specificată anterior. Această abordare poate genera un aliniament complet al secvenţelor considerând gap-uri. Din păcate timpul de execuţie devine nerezonabil de mare pentru secvenţe cu mai mult de 50 reziduuri.

Ultima categorie a aliniamentelor multiple, aşa numita *consensus* sau *metoda regiunilor*, a fost introdusă în [WATE'86]. Aici este prezentat un algoritm care execută aliniament multiplu de secvenţe prin potrivire de cuvinte de lungime dată. Mărmea cuvintelor şi gradul de nepotrivire permis sunt selectate de utilizator. Aliniamentul maximizează funcţia de calcul corespunzătoare. Metoda este bazată pe o extindere nouă a metodelor consensus anterioare şi permit estimarea semnificaţiei statistice.

2.4. Concluzii

Indiferent de obiectivul biologic urmărit prin analiza şi compararea secvenţelor biologice este necesară folosirea de metode şi tehnici algoritmice adecvate. Din diversitatea de metode existente larg cunoscute se conturează două tendinţe:

- Metode de aliniament bazate pe programarea dinamică;
- Metode orientate spre folosirea modelelor Markov ascunse.

Deoarece fiecare dintre aceste direcţii implică o serie de limitări, cum ar fi timpii de execuţie relativ mari, spaţiul de memorie semnificativ pentru prelucrarea unei cantităţi mari de date sau căutarea unor valori adecvate pentru diverşi parametri necesari estimărilor probabilistice, a existat şi încă există o preocupare permanentă de a simplifica aceste metode şi de a găsi metode alternative care să servească aceloraşi scopuri.

Prin urmare s-a realizat şi o investigare a tehnicilor de similaritate propuse relativ recent. Ele vin oarecum ca alternativă la metodele de aliniament fundamentale şi chiar dacă nu se pretinde o acoperire totală a acestora ele pot să ofere o imagine de ansamblu asupra noilor metode propuse/experimentate pentru determinarea similarităţii secvenţelor (reprezentate ca structuri primare în cazul proteinelor).

Dată fiind complexitatea problemei de comparare a secvenţelor atât cu scopul determinării similarităţii cât şi al identificării de segmente cu rol biologic important, s-a ajuns la următaorele concluzii:

1) Pentru determinarea similarităţii sunt necesare fie funcţii pentru evaluarea aliniamentelor în vederea stabilirii distanţei dintre secvenţe, care implică şi folosirea matricelor de substituţie, fie tehnici algoritmice independente de aliniamentul secvenţelor;
2) Aliniamentele multiple folosesc de obicei tehnicile pentru compararea de perechi de secvenţe pe care le incorporează în algoritmi de parcurgere şi evaluare a tuturor secvenţelor implicate în procesul de comparare.
3) Din diversitatea de metode nu poate fi stabilit cu precizie cel mai performant algoritm de determinare a similarităţii întrucât el este dependent de scopul urmărit în compararea secvenţelor. Totuşi, ceea ce se urmăreşte în general este ca timpul şi spaţiul computaţional necesar să fie cât mai redus;
4) Metodele de aliniament şi determinare a similarităţii cunosc o continuă îmbunătăţire având ca scop obţinerea de performanţe ridicate la costuri cât mai mici.
5) Este de remarcat implicarea estimărilor statistice atât în evaluarea rapoartelor de asociere pentru ponderea elementelor în matricele de substituţie cât şi funcţiile de evaluare ale aliniamentelor cu precădere pentru modelele Markov ascunse.

Aceste metode de similaritate se constituie în nuclee care stau la baza diverselor programe comerciale destinate comparării, analizei şi extragerii de informaţii relevante din baze mari de secvenţe.

3. APLICAREA TEHNICILOR DE RECUNOAŞTERE ÎN ANALIZA SECVENŢELOR BIOLOGICE

3.1. Obiective de interes biologic ce implică tehnicile informatice

Cercetările comunităţii în inteligenţă bioinformatică sunt focalizate pe diverşi algoritmi de învăţare, sisteme de data mining şi sisteme integrate care să permită transformarea secvenţelor biologice, a observaţiilor şi cunoştinţelor în informaţie structurată şi semnificativă pe care biologii o pot interoga, vizualiza şi înţelege [KASI'99]. În conformitate cu descrierea făcută de Simon Kasiff, cele mai importante funcţii computaţionale identificate în acest domeniul de cercetare pot fi grupate în următoarele categorii:

- Identificarea genelor/proteinelor şi clasificarea lor în categorii;
- Metode de comparare a secvenţelor biologice la diferite nivele de detaliu;
- Identificarea regiunilor regulatorii şi a modelelor noi în secvenţa de date (căutarea regiunilor care se repetă, similarităţi şi modele rare care au semnificaţie biologică).

Pe baza acestor teme, oamenii de ştiinţă încearcă să dezvolte tehnici specializate şi cel mai mare volum al cercetării corespunzătoare este orientat spre:

- algoritmi pentru învăţare automată destinaţi prelucrării informaţiei conţinute în depozite mari de secvenţe;
- noi metode de declanşare automată a algoritmilor pornind de la date (bootstrapping algorithms);
- integrarea surselor multiple de informare într-un singur sistem de învăţare şi descoperire;
- îmbunătăţirea vitezei şi acurateţii sistemelor probabilistice de raţionament.

În secţiunile următoare sunt prezentate cu precădere tehnici bine cunoscute în domeniul sistemelor de recunoaştere automată, puse în serviciul atingerii obiectivelor biologice mai sus menţionate aşa după cum au fost identificate în lucrări de specialitate relativ recente. Scopul este de a furniza o privire cuprinzătoare.

3.2. Identificarea geneleor/proteinelor şi clasificarea în categorii

Acest subcapitol include o prezentare sumară a algoritmilor aparţinând marii familii a clasificatorilor. Descrierea detaliată a acestor algoritmi poate fi găsită în lucrările citate.

3.2.1. Clasificatorul celor mai apropiaţi *k* vecini (k-NN)

Urmând procedura standard a clasificării, pentru a clasifica o nouă secvenţă (test) fiind dată o mulţime de secvenţe (set de antrenare) cu clasificare cunoscută (şi prin urmare caracterizare funcţională cunoscută) mai întâi trebuie identificaţi primii *k* cei mai apropiaţi vecini de secvenţe de test în setul de antrenare. Acest pas foloseşte o măsură de similaritate aşa încât să ordoneze secvenţele în setul de antrenare şi în final să le localizeze pe *k* cele mai similare. Mai departe, şi bazat pe etichetele claselor celor mai apropiaţi *k* vecini (*k*-NNs), eticheta cea mai frecventă este asociată secvenţei de testare şi cu această simplă schemă (vot majoritar), clasificarea ei este îndeplinită. Cea mai critică parte a practicii acestei clasificări este cea a estimării similarităţii perechilor de secvenţe. În diferite versiuni ale clasificatorului *k*-NN, sunt utilizate funcţii alternative de similaritate şi diferenţele dintre ele constă în modul în care sunt aliniate secvenţele. Pot fi folosite fie scorurile aliniamentului global (care consideră secvenţele aliniate de-a lungul întregii lungimi) sau scoruri ale aliniamentului local (unde numai porţiuni ale celor două secvenţe sunt aliniate). Clasificarea *k*-NN poate fi găsită adesea în lucrări legate de compararea secvenţelor sau predicţia funcţiilor [GANA'04], [MARK'03].

3.2.2. Clasificatori ce folosesc lanţuri sau modele Markov generalizate

Tehnicile bazate pe lanţuri Markov sunt foarte populare în modelarea secvenţelor datorită abilităţii lor de a captura constrângeri secvenţiale existente în date. În [DESP'02] sunt descrise câteva abordări de clasificări înrudite. Ele pot folosi lanţuri Markov simple (ordinul 1), lanţuri Markov de ordin superior, modele Markov interpolate şi modele Markov selective.

În primul caz, secvenţele de antrenare sunt partiţionate în clase diferite pe baza etichetelor asociate. Apoi, un singur lanţ Markov M este construit pentru fiecare din aceste grupuri de secvenţe. O secvenţă de test dată S_r este clasificată calculând la început probabilitatea condiţionată $P(S_r|M)$ a acelei secvenţe fiind generată de fiecare din acele lanţuri Markov. Lanţul Markov ce corespunde probabilităţii maxime este selectat pentru a împrumuta eticheta clasei corespunzătoare secvenţei de test. Pentru problema a două clase aceasta este făcută prin calcularea raţiei *log* a probabilităţii furnizată de funcţia de clasificare: $L(S_r) = \log \frac{P(S_r|M_+)}{P(S_r|M_-)}$, unde M_+ şi M_- sunt lanţurile Markov corespunzătoare clasei de exemple pozitive respectiv negative.

În cazul lanţurilor Markov de grad mai mare, probabilitatea de tranziţie pentru un simbol este calculată pe baza a *k* simboluri precedente, obţinându-se astfel un lanţ Markov de ordinul *k*. În general, aceste modele de ordin superior au o acurateţe a clasificării ridicată comparativ cu modelele de ordin mai mic datorită faptului că ele sunt capabile să capteze secvenţe mai lungi din setul de date cu rol constrictiv dar, de asemenea, pot întâmpina probleme. În principiu, problema alegerii modelelor de ordin ridicat este dependentă de mărimea setului de

antrenare. O variantă a acestor metode sunt modelele Markov interpolate (IMM) unde o serie de lanţuri Markov sunt construite începând de la ordinul 0 până la ordinul *k*. *k* este o variabilă definită de utilizator şi fixată în avans. Motivaţia acestei abordări este că, în ciuda faptului că stările Markov de ordin ridicat capturează un context mai lung, ele au suport redus. Tehnicile IMM au fost dezvoltate iniţial pentru problema prezicerii simbolului următor în secvenţă şi nu pentru clasificarea secvenţelor. Dar, din moment ce ele constituie o metodă alternativă pentru estimarea diverselor probabilităţi condiţionate pot fi folosite, de asemenea, la clasificare.

Modelele Markov selective (SMM) sunt un alt set de tehnici care adresează problema estimării lanţurilor Markov de ordin ridicat şi, ca IMM-urile, ele au fost dezvoltate pentru predicţia simbolului următor în secvenţe. Aceste tehnici construiesc lanţuri Markov de ordin variat după care retează stările non-discriminative din lanţurile Markov de ordin ridicat. Atribuţia principală este de a decide care stări sunt non-discriminative pentru a putea fi înlăturate. Autorii din [DESP'02] au folosit metoda bazată pe pragul simplu de frecvenţă cu scopul de a înlătura stările care nu au loc frecvent.

3.2.3. Clasificarea secvenţelor pe baza caracteristicilor

Clasificarea tradiţională folosind algoritmii de învăţare automată cum sunt arborii de decizie, clasificatorii bazaţi pe reguli şi maşini cu suport vectorial (Support Vector Machines notate presurtat SVM), pot fi folosiţi şi pentru clasificarea mulţimilor de secvenţe dacă se dovedeşte că secvenţele au fost mai întâi modelate într-o formă potrivită respectivului algoritm dar fără a ignora natura lor secvenţială.

Autorii [DESP'02] au dezvoltat un cadru pentru modelarea secvenţelor astfel încât algoritmii tradiţionali de învăţare automată cum ar fi SVM pot fi aplicaţi uşor. Ei consideră N numărul simbolurilor distincte din diferite secvenţe (ex. 4 pentru nucleotide şi 20 pentru amino-acizi) după care casificarea este făcută pe baza funcţiei de clasificare a lanţurilor Markov (deja menţionate). Prin urmare $L(S_i) = u' w$, unde *u* şi *w* sunt vectori de lungime N^2. Fiecare dintre dimensiunile acestor vectori corespunde unei perechi unice de simboluri. Dacă *ab* este prechea de simboluri corespunzătoare dimensiunii *j* a acestor vectori, atunci *u*(*j*) este egal cu numărul care reprezintă frecvenţa apariţiei lui *ab* în secvenţa S_i, iar *w*(*j*) este egală cu logaritmul corespunzător care apare în funcţia de clasificare. Aspectele importante ale acestei transformări constau în faptul că permit abordarea fiecărei secvenţe ca şi un vector într-un spaţiu nou ale cărui dimensiuni sunt constituite din toate perechile, tripletele, etc. de simboluri posibile şi că fiecare din tehnicile bazate pe lanţurile Markov nu sunt altceva decât un clasificator liniar [MITC'97], unde planul de decizie este definit de vectorul *w* care corespunde logaritmului din raportul de incertitudine (log-odds[17]) al estimării probabilităţilor maxime pentru diverse probabilităţi de tranziţie. Acurateţea rezultatelor obţinute sugerează că modelul liniar de clasificare învăţat de SVM este mai bun decât modelele liniare învăţate de abordarea bazată pe lanţurile Markov

[17] Un sistem de notare în care valorile sunt logaritmul probabilităţii relative a unei comparaţii datorate omologiei sau simplei şanse [BIG'05].

3.2.4. Reţele neuronale pentru clasificarea secvenţelor

O aplicaţie interesantă a reţelelor neuronale (NNets) poate fi întâlnită în [WANG'00a], unde sunt extrase caracteristici din date aflate sub forma proteinelor şi folosite în combinaţie cu reţelele neuronale Baesiene (BNN) pentru a clasifica secvenţele de proteine. Într-un mod formal, problema studiată consideră secvenţa de proteină nemarcată/neclasificată S şi o superfamilie *F*. Scopul este de a verifica în ce măsură S aparţine sau nu lui *F*. Aici, atât similaritatea locală cât şi cea globală sunt analizate în mod simultan. Metoda lor include o etapă de extragere a caracteristicilor. În acest pas şi pentru a considera similaritatea globală este folosită o metodă de codificare de forma 2-gram [Wu'00] (cu şi fără substituţie). Aceasta rezultă într-un număr total de 436 modele posibile de 2 litere care ar fi variabilele de intrare în reţeaua neuronală. Pentru a evita „capcana dimensionalităţii", este urmat un pas de selectare a caracteristicilor pe baza unei măsuri probabilistice de distanţă. Cele mai relevante *Ng* caracteristici sunt folosite ca intrări pentru BNN. Separat, pentru a compensa pierderea de caracteristici este inclus un coeficient de corelare liniar (LCC). Pentru a incorpora în schema clasificării informaţia din similaritatea locală este folosită o descriere a lungimii (LS) pentru fiecare secvenţă. Această măsură LS este utilizată ca intrare separată pentru reţeaua neuronală. Astfel, în final, o secvenţă de proteine este reprezentată folosind Ng+2 numere reale. Comparând rezultatele cu cele obţinute folosind BLAST [ALTS'97], ce se bazează pe aliniamentul secvenţelor şi SAM [KARC'98], ce foloseşte modele Markov ascunse, autorii concluzionează că cele trei metode se complementează şi "combinându-le se obţin rezultate mai bune decât dacă s-ar folosi fiecare clasificator separat".

O abordare diferită a clasificării pe baza învăţării supervizate folosind reţelele neuronale a fost găsită în [ANAS'03], unde intenţia este de a demonstra faptul că dacă unii algoritmi deja cunoscuţi sunt corect utilizaţi ei pot duce la soluţii mai bune pentru unele probleme cerute în bioinformatică. Îmbunătăţirile soluţiilor acestor probleme s-au obţinut cu ajutorul reţelelor neuronale feed-forward aplicând scheme mai avansate pentru învăţarea supervizată în reţea. Rezultatele sunt evaluate comparativ cu cele ale altor clasificatori cunoscuţi.

3.3. Compararea secvenţelor

Compararea secvenţelor biologice conduce, în mod indirect, la măsuri de similaritate. Există o varietate de tehnici pe această temă aşa cum au fost menţionate în *captolul 2*. În acest subcapitol însă va fi facută o mai mult o prezentare a modului în care acestea se aplică pentru căutari eficiente în bazele de secvenţe.

Cele mai frecvent folosite metode pentru aflarea similarităţii secvenţelor biologice sunt cele care se bazează pe aliniamentul perechilor de secvenţe (cum ar fi algoritmul Needleman-Wunsch [NEED'70] şi algoritmul Smith-Waterman [SMIT'81b]) sau pe principiul modelelor Markov ascunse (HMM) (ex. [KROG'94], [BALD'94]). Conform [LIAO'03], dezvoltarea metodelor puternice pentru detectarea similarităţii proteinelor (şi a secvenţelor biologice in general) poate fi împărţită în patru generaţii.

Metodele timpurii sunt reprezentate de similarităţile perechilor (pairwise similarity). Algoritmul dinamic al lui Smith-Waterman (1981) a rămas referinţă datorită acurateţii rezultatelor, în timp ce algoritmi euristici cum ar fi cei dezvoltaţi pentru BLAST (Basic Local Alignment Search Tool)[18], FASTA (după FAST –All, sau comparaţie rapidă a proteinelor/nucleotidelor)[19] sau CLUSTAL[20] (pentru aliniamentul mai multor secvenţe) negociază o acurateţe redusă în schimbul îmbunătăţirii eficienţei.

A doua generaţie ar fi reprezentată de profile şi lanţuri Markov ascunse şi conţin metode bazate pe ideea de familie. Ei permit biologului să deducă/obţină apropape de trei ori mai multe omologii decât un simplu algoritm bazat pe compararea perechilor [LIAO'03].

În a treia generaţie, algoritmi ca PSI-BLAST [ALTS'97] şi SAM (Sequence Alignment and Modeling System)[21] folosesc informaţie stocată în baze de date mari şi îmbunătăţesc rezultatele peste metodele bazate pe profile, colectând secvenţe omoloage şi incorporând statisticile rezultate într-un model central .

În generaţia a patra creşterea acurateţii s-a câştigat modelând diferenţa dintre exemplele pozitive şi negative de alinament [LIAO'03].

Toate aceste metode sunt construite pe scoruri de similaritate dintre secvenţele aliniate, revelând evoluţia măsurilor de similaritate [KRAS'04], [LIAO'03], dar încă mai există interes în căutarea unor metode noi deoarece costurile calculelor interogărilor cresc liniar cu volumul bazelor de date de secvenţe biologice [CAMO'03]. În plus, cele mai multe dintre metodele existente implicit consideră că este importantă alegerea unui algoritm de căutare adecvat, matrice de scoruri sau funcţii care pot fi definite şi un set de parametri opţionali pentru care valorile optime corespund celor mai bune scoruri posibile de similaritate dintre două secvenţe.

3.3.1. Căutarea eficientă în baze de date

Mărimea secvenţei de acizi nucleici este într-o creştere continuă iar interersul în compararea secvenţelor biologice noi cu cele existente este de asemenea în creştere.

Problema de interes matematic este cum să se caute rapid în baze de date mari. Spre exemplu, pentru căutarea a 2000 secvenţe a 500 perechi de baze(bp)[22], fiecare comparată cu 500 noi secvenţe bp ar lua, cu tehnici de programare dinamică, timp proporţional cu $2000(500)^2 = 7.5 \times 10^8$. Acesta este un număr

[18] http://www.ncbi.nlm.nih.gov/BLAST/
[19] http://www.ebi.ac.uk/fasta33/
[20] http://www.ebi.ac.uk/clustalw/
[21] http://www.cse.ucsc.edu/research-compbio-sam.html
[22] În genetică, două nucleotide în straturi complementare opuse de ADN sau ARN care sunt conectate via legături de hidrogen sunt o pereche de baze (adesea prescurtat bp). Deoarece ADN este de obicei strat dublu , numărul de perechi de baze în "dsDNA" este egal cu numărul de nucleotide într-unul dintre straturi. În ADN, adenina (A) şi timina (T), precum şi guanina şi citozina, pot fi câte o pereche de baze. În ARN, timina este înlocuită de uracil wikipedia.org/wiki/Base_pairs.

inacceptabil de mare şi, în cele ce urmează, vor fi prezentate câteva abordări la problema căutării rapide.

O metodă deja cunoscută pentru aceste căutări este este aceea a lui Wilbur şi Lipman [WILB'83], [WILB'84]. Aceasta ţine cont de unele descoperiri biologice şi în plus metoda de programare dinamică a segmentelor maxime este folosită în acest scop. Waterman [WATE'84] sugerează conexiuni între metoda Wilbur-Lipman şi cea a lui Martinez.

Metoda Wilbur-Lipman

În două lucrări importante, Wilbur şi Lipman (1983, 1984) dezvoltă ceea ce ei numesc comparare dependentă de context. Metoda lor este conturată de urmatoarele aspecte (1) produce o listă L de regiuni de potrivire, toate de lungime fixă, (2) ordonează regiunile ca în secţiunea 2.3.7, (3) obţin un aliniament optimal prin procesarea listei L. De fapt ei dezvoltă o teorie pentru un context mult mai general al dependenţei şi în plus prezintă condiţii pentru măsurile lor de similaritate de a avea o distanţă asociată care este metrică. Pe mai departe discuţia va fi limitată la cazul regiunilor.

În secţiunea 2.3.7 s-a discutat cazul regiunilor. Aici, lista L a regiunilor potrivite exact poate fi restricţionată spre exemplu la acele regiuni de lungime exact 4. Wilbur şi Lipman [WILB'83] descriu un algoritm de hashing în timp liniar pentru a produce L cu regiuni de lungime fixă.

Se notează o regiune r prin (w; i, j) unde w este un cuvânt de lungime |w| care începe la poziţia *i* în secvenţa **a** şi poziţia *j* în secvenţa **b**. Ca şi în secţiunea 2.3.7, $r_1 < r_2$ dacă $i_1 + |w_1| - l < i_2$ şi $j_1 + |w_2| - l < i_2$. De asemenea fie regiunea $r_0 = (\phi; 0,0)$ un element minimal şi $r_* = (\phi; n, m)$ un element maximal. $\Gamma = (r_1, r_2, \ldots, r_l)$ este o cale dacă $p < q$ implică $r_p < r_q$. Scorul unei căi Γ este dat de următoarea relaţie

$$score(\Gamma) = \sum_{k=1}^{l} s(r_k) - \sum_{k=1}^{l-1} g(i_{k+1} - |w| - i_k - 1,\ j_{k+1} - |w_k| - j_k - 1), \qquad (3.3.1\text{-}1)$$

unde $s(\cdot)$ este un scor de similaritate pentru o regiune r_k, ca $|w_k|$, şi $g(\cdot, \cdot)$ este o penalitate pentru gap. Atunci,

$S(a, b) = \max\{score(\Gamma): \Gamma$ este o cale de la r_0 la $r_*\}$.

Algoritm Wilbur - Lipman.

Algoritmul este după cum urmează: se fac două liste a două regiuni L^-, ordonat de „<", şi L^+ ordonat după oridinea obişnuită „□" a celor mai bune scoruri din r_0 la regiunea listată în L^+.

(0) Fie $L^- = L$ şi $L^+ = \phi$

(1) $r_q = \{\min r: r \in L^-\}$
score(r_q) = S(r_q)

(2) Începe la cel mai mare element a lui L^+

(A) Mută în jos L^+ până când $r_u < r_q$ astfel încât

$$\gamma = score(r_u) - g(i_q - |w_{r_u}| - i_u - 1,\ j_q - |w_{r_u}| - j_u - 1) + s(r_q) > score(r_q).$$

Dacă nu există vreo regiune r_u în L^+ mai jos de r_q sub < cu un scor mai mare decât scorul (r_q) – s(r_q), sau dacă inegalitatea nu poate fi satisfăcută, se merge la (C)

(B) Se fixează score r_q = γ şi salt la (A)

(C) Se înlătură r_q din L^- şi se inserează în L^+ după regula impusă de „□".
Dacă $L^- \neq \phi$, salt la (A).

Este posibil să se modifice acest algoritm, pentru a obţine segmente de similaritate maximă în loc de aliniamente de similaritate maximă.

3.3.2. Un algoritm vectorizat pentru segmente maxime

O altă abordare de calcul rapid a fost propusă de T.F.Smith la Los Alamos national Laboratotry. Smith a modificat algoritmul Smith-Waterman astfel, utilizând arhitectura de vector este posibil să se realizeze comparaţii între un foarte mare număr de secvenţe de acizi nucleici în timp rezonabil. Spre exemplu, un studiu al lui Smith et al. [SMIT'84] raportează toate comparaţiile de perchi de secvenţe dintre 204 secvenţe de la vertebrate care au fost realizate în aproximativ 170 min, la o rată de peste 240 secvenţe per minut cu o medie a lungimii secvenţei de 800 nucleotide.

Vectorizarea algoritmilor este un subiect relativ nou. Ideea esenţială este că maşina execută un număr de operaţii simultan şi câştigă acest factor peste operaţiile obişnite liniare ale secvenţelor. Dacă atenţia este focalizată pe calcularea lui H, nici o calculare a lui H_{ij} executată nu poate depinde de rezultatele unor alte calcule executate simultan. Aceasta elimină linie cu linie sau coloană după coloană construirea matricei. Ce rămâne este calculare de blocuri ale lui H_{ij} pe diagonale negative, i.e. i + j = constant.

Collins şi Coulson [COLL'84] discută algoritmi de procesare paralelă pentru metoda lui Sellers. Abordarea lui Smith ar trebui să permită implementări mult mai eficiente pentru procesarea paralelă.

3.3.3. Metoda regiunilor

Metoda regiunilor lui Martinez (descrisă în secţiunea 2.3.7) şi metoda lui Wilbur-Lipman sunt evident asemănătoare, cu toate că Wilbur-Lipman vine cu o tehnică diferită de optimizare. Desigur că cele două tehnici au fost dezvoltate independent dar subiectul este evident. Pentru calcularea rapidă se limitează logic lista regiunilor şi aliniamentele vor fi produse mult mai rapid.

Martinez [MART'83] a sugerat abordarea acestor probleme via regiuni. Localizarea tuturor repetiţiilor a R secvenţe de lungime *n*, poate fi făcută în timp proporţional cu *n*R. „Repet" în acest context se referă la un cuvânt w care apare în toate secvenţele R. O asemenea repetare se numeşte regiune şi este plasată într-o listă L. O ordine parţială „<" în R dimensiuni (în loc de două dimensiuni) este plasată în L.

Exact aceeiaşi algoritmi din secţiunile 2.3.7 şi cei menţionaţi în această secţiune pot fi folosiţi pentru a produce un aliniament multiplu de secvenţe. Limitarea aici este că cerinţa de potriviri exacte în toate cele R secvenţe este imperios necesară. De asemenea, porţiunile aliniate ar fi foarte convingătoare şi un asemenea aliniament ar putea fi produs într-un timp rezonabil.
Karlin et al. [KARL'84] nu declară un algoritm pentru alinierea regiunilor lungi care se potrivesc, cu toate că din aceste potriviri rezultă aliniamente. Este evident că folosirea regiunilor ca intrări într-o metodă bazată pe regiuni ar produce aliniamente semnificative. În plus, dacă lista cu regiuni nu este total ordonată, o asemenea metodă de aliniament ar fi foarte utilă în rezolvarea posibilelor aliniamente.

3.3.4. Algoritmi de programare dinamică

Într-un articol elaborat de Sankoff şi Cedergren [SANK'83] sunt revizuite metodele care realizează aliniamente a R secvenţe fiind dat un arbore care le conectează. Fiecare nod interior al arborelui dat T are un grad de cel puţin trei şi cele R secvenţe sunt ataşate de cele R noduri terminale. Algoritmul construieşte secvenţe pentru fiecare nod interior şi dă un aliniament legând secvenţele originale şi cele reconstruite. Dacă arborele are N noduri interioare acesta este un aliniament de R+N secvenţe.
Waterman [WATE'84] încearcă să dea aspecte ale programării dinamice acestei probleme. Se presupune ca cele R secvenţe sunt a, b, ..., r. Costul unui aliniament general al lui R original plus N secvenţe construite este suma costului aliniamentului de perechi în arborele T. Ideea este de a gândi la alinierea a_1a_2, ... a_i, b_1b_2 ... b_j ..., r_1 r_2 ... r_s. Ultima coloană a aliniamentului poate fi înlăturată de lângă coloanele iniţiale. Pentru a, b, ...,r ultima coloană va apare ca

$$\varepsilon_1 a_i$$
$$\varepsilon_2 b_j$$
$$\ldots$$
$$\varepsilon_R r_s$$

unde $\varepsilon_1 \in \{0, 1\}$, $0 \cdot a_i = \Delta$, şi $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_R) \neq 0$. Această ultimă restricţie ţine doar ultima coloană de a nu fi toată Δ. Pasul de programare dinamică este

$$D_{i,j,\ldots,s} = \min_{\varepsilon \neq 0} \left\{ D_{i-\varepsilon_1, j-\varepsilon_2, \ldots, s-\varepsilon_R} + \min_{\varepsilon \neq 0} length \begin{pmatrix} \varepsilon_1 a_i \\ \varepsilon_2 b_i \\ . \\ . \\ \varepsilon_{R1} r_s \\ x_1 \\ . \\ . \\ x_N \end{pmatrix} \right\} .$$

Ultimul termen este pentru a indica faptul că literele $x_1x_2 \ldots x_N$ pentru N noduri interioare a fost determinat într-un mod optimal. Pentru acest scop este folosită metoda economică a lui Fitch [FITC'71], generalizată adecvat. Timpul de calcul al acestui algoritm pentru R secvenţe de lungime *n* este $O(2^R n^R N)$ unde 2^R provine din $\varepsilon_1, \ldots, \varepsilon_R$ la fiecare pas, n^R vine de la *i*, *j*, ..., *s*, şi N vine de la metoda economică a lui Fitch. Pentru secvenţe de lungime 100, aceasta este aproximativ $O(10^{2..3R} N)$ astfel R=3, este atit de larg cât este posibil.
Independent de munca lui Sankof [SANK'76], Waterman et al [WATE'76] prezintă un algoritm similar care nu presupune în mod explicit un arbore. În cadrul muncii lui Sankoff aceasta ar corespunde unui arbore, un nod interior şi R noduri terminale. A fost folosită o funcţie d(x,y,...,z) de R variabile pentru a generaliza metrica pe litere astfel, este posibil de a avea o ponderare diferită, dar ideea esenţială este în lucrarea lui Sankoff.

3.3.5. Un algoritm bazat pe conceptul de modele Markov ascunse

Secvenţele biologice funcţionale de obicei apar în familii şi multe dintre cele mai puternice metode de analiză sunt bazate pe identificarea relaţiilor dintre o secvenţă individuală cu o familie de secvenţe. Prin urmare, identificarea faptului că o secvenţă aparţine unei familii şi aliniind-o cu ceilalţi membri adesea permite inferenţe despre funcţia sa. Dacă spre exemplu avem un set de secvenţe care aparţin unei familii, se poate face o căutare într-o bază de date pentru mai mulţi membri folosind aliniament de perechi cu unul dintre membrii familiei cunoscute ca secvenţă de query. Pentru a fi mai aproape de realitate se poate face căutare după fiecare dintre membrii familiei unul câte unul. Oricum căutarea pe baza comparaţiei perechilor pentru oricare dintre secvenţe poate să nu găsească secvenţele apropiate ca distanţă cu cele avute deja. O abordare alternativă este de a folosi în căutare caracteristici statistice ale întregului set de secvenţe. Similaritatea, chiar şi atunci când apartenenţa la o familie este clară, datorită acurateţii aliniamentului, poate fi adesea îmbunătăţită prin concentrarea pe caracteristici care sunt conservate în intreaga familie.

În acest caz intervine aliniamentul de secvenţe multiple în care scopul este de a construi un model probabilistic. În particular, este dezvoltat un anume tip de HMM care să se potrivească bine modelării multiple de aliniamente. Acestea sunt denumite profile HMM după profilele standard, care sunt structuri apropiate nonprobabilistice introduse anterior de Gribskov, McLachlanm şi Eisenberg [GRIBS'87]. Profilele HMMs sunt probabil cea mai populară aplicaţie a HMM în biologia moleculară pâna în jurul anului 2000 [DURB'98].

Un model liniar Markov ascuns este o secvenţă de noduri fiecare corespunzând unei coloane într-un aliniament multiplu. În cazul HMM folosit în aplicaţia SAM[23] fiecare nod are, de asemenea, o stare de potrivire/matching, de inserare/insertion şi o stare de ştergere/deletion. Fiecare secvenţă foloseşte o serie a cestor stări pentru a traversa modelul de la început până la sfârşit. Folosirea unei stări de *potrivire*(*matching*) indică faptul că secvenţa are un caracter în acea coloană în timp ce folosirea unei stări de *ştergere* (*deletion*) semnalează contrariul. Stările de *inserare* permit secvenţelor de a avea caractere adiţionale între coloanele de aminoacizi. Stările de poziţii vecine sunt conectate prin linii unde, pentru fiecare dintre aceste linii, există asociată o probabilitate de tranziţie care este probabilitatea de trecere dintr-o stare în alta. Topologia de bază a modelului este ilustrată în Figura 3.3.5-1.

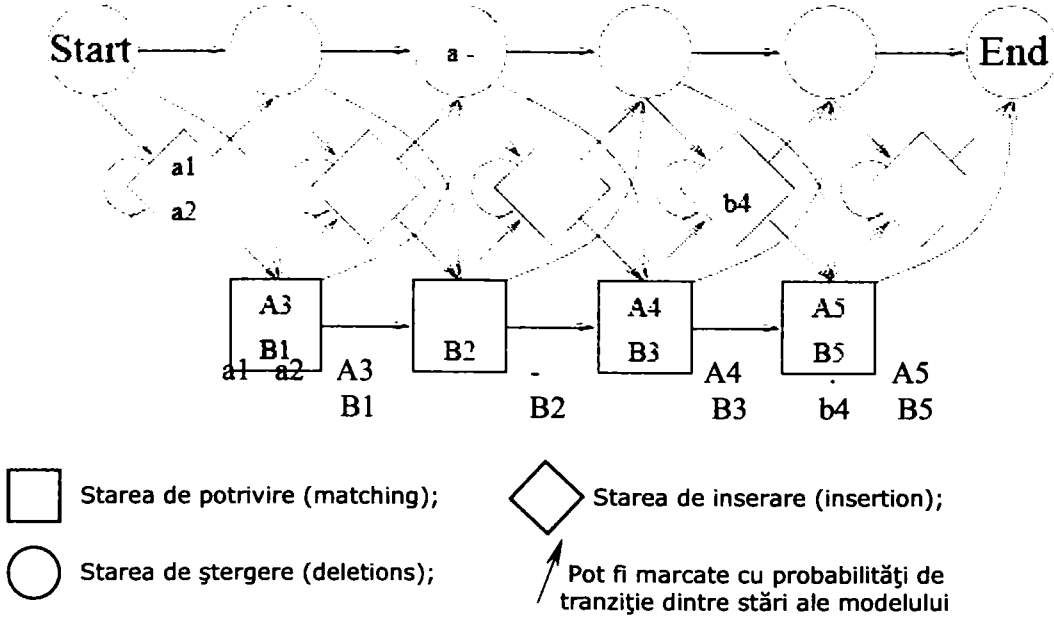


Figura 3.3.5-1. Un model Markov ascuns liniar şi exemplu de aliniament.

[23] Sequence Alignment and Modeling System,
http://www.cse.ucsc.edu/research/compbio/HMM-apps/HMM-applications.html

Topologia acestui model a fost aleasă pentru a caracteriza inserările şi ştergerile în mod similar cu tehnicile de aliniament biologic al secvenţelor pe baza penalităţilor caracterelor lipsă (gap penalties). Modelele sunt antrenate pe o familie de secvenţe de proteine utilizând un algoritm de maximizare al aşteptărilor şi o varietate de algoritmi euristici. Un model antrenat poate să fie folosit atât pentru a genera aliniament multiplu cât şi pentru căutare în baze de date a noi membri ai familiei.

Avantajul principal al acestor modele peste metodele standard de căutare îl reprezintă abilitatea lor de a caracteriza o întreagă familie de secvenţe. Astfel, fiecare poziţie are o distribuţie a bazelor, aşa cum o au tranziţiile între stări. Aceasta înseamnă că HMMs liniare au distribuţii ale caracterelor dependente de poziţie, precum şi inserţii şi penalităţi de ştergeri (gap penalties) dependente de poziţie. Aliniamentul fiecărei familii la un model antrenat va genera automat aliniamentul multiplu al secvenţelor componente. În multe feluri aceste modele corespund profilelor. Metoda implementată în SAM-T2K, de detectare a omologiilor distante, este o metodă HMM iterativă de căutare pentru crearea unui HMM dintr-o singură secvenţă de proteine sau aliniament sursă folosind căutarea iterativă a secvenţei într-o bază de date. Metoda este considerată de autori [HUGH'96] cel mai senzitiv algoritm bazat pe determinarea de la distanţă a omologiei unei secvenţe. Detalii care descriu pe larg strategia de funcţionare a SAM pot fi consultate la adresa:
http://www.cse.ucsc.edu/research/compbio/html_format_papers/hughkrogh96

În concluzie, această trecere în revistă a principalelor metode nu epuizează tipurile de probleme care prezintă interes. Algoritmii existenţi pot fi modificaţi pentru a folosi la soluţionarea unei mari varietăţi de probleme.

3.4. Identificarea regiunilor regulatorii şi tiparelor noi în date secvenţiale

Algoritmii şi tehnicile folosite în acest scop acoperă o mare varietate de abordări. O listă detaliată poate fi găsită în [BREJ'00]. În continuare vor fi identificate şi descrise pe scurt principalele categorii.

3.4.1. Căutarea exhaustivă

Cel mai bun algoritm pentru identificarea tiparelor semnificative în secvenţe biologioce lungi nu este usor de proiectat şi prin urmare, în multe aplicaţii, căutarea aleasă este cea exhaustivă. Una dintre cele mai simple metode în descoperirea tiparelor este de a enumera toate tiparele posibile care satisfac constrângerile impuse de utilizator. Pentru fiecare tipar este estimată mai întâi frecvenţa de apariţii în secvenţa de intrare şi pe baza acesteia este atribuit un scor sau o semnificaţie statistică. Apoi, sunt identificate tiparele cu cele mai mari scoruri sau tipare cu scoruri peste un anumit prag. Această metodă este adecvată doar când este vorba de tipare scurte deoarece timpul de execuţie creşte exponenţial cu lungimea tiparelor ce trebuie identificate. Pe de altă parte, această metodă poate întotdeauna să găsească cele mai interesante tipare [VANH'98], [TOMP'99], [SMIT'90].

Un mod mai eficient este de a aplica metoda de înlăturare a enumerării tiparelor. În această metodă este identificat un tipar lung şi fără goluri (gap) care este posibil să apară cu unele neportiviri în cel puţin K secvenţe. El poate începe de la tipare scurte care apar în cele mai multe K secvenţe şi să le extindă atâta timp cât suportul nu scade sub K. La fiecare pas este nevoie de extensia tiparului (identificat curent) în toate felurile posibile. Această extensie este urmată de căutarea apariţiilor tiparelor extinse încă în cel puţin K secvenţe. Odată ce obţinem un tipar care nu poate fi extins fără pierderea de suport, acest tipar este considerat cel maximal şi poate fi inregistrat ca final. Acest algoritm de căutare realizează de fapt o primă căutare în adâncime în arborele tuturor secvenţelor posibile iar ramurile care nu realizează suport pentru vreun tipar sunt înlăturate. Teoretic, această metodă poate funcţiona bine dar creştera exponenţială a timpului de căutare rămâne o problemă [RIGO'98].

Căutarea exhaustivă într-un graf ar fi o altă metodă din acest grup şi sunt multe posibilităţi de determinare a apariţiilor unui tipar. În [PEVZ'00], este descrisă problema căutării tiparului de lungime dată L care apare în toate secvenţele de intrare n cu cel mult d nepotriviri. Pentru tipare mai complicate este necesară folosirea abordărilor euristice care nu găsesc neparat cel mai bun tipar dar pot converge la o soluţie suboptimă. O descriere a unei asemenea tehnici, bazată pe partiţionarea Gibbs poate fi găsită în [BREJ'00] sau [LAWR'93].

3.4.2. Metode de învăţare automată

Când un tipar este exprimat sub forma unui model stochastic cum este modelul Markov ascuns sau matricea cu ponderi a poziţiilor (care este tot o versiune mai simplă a HMM), pot fi folosite tehnicile iterative de maximizare a aşteptărilor care nu converg în mod necesar la maximul global. În [LAWR'90], este folosit un algoritm de învăţare pe baza maximizării aşteptărilor (Expectation Maximization notat şi cu EM) al căror scop este de estimare a parametrilor unui model probabilistic/stochastic al tiparului, care apare o singură dată la o poziţie necunoscută în fiecare secvenţă din familia de secvenţe date. Această abordare poate fi extinsă la modele mult mai complicate ca cea cu goluri (gaps) flexibile sau modelul mixt finit [BAIL'94]. O descriere detaliată este facută în [BREJ'00].

Modelele Markov ascunse pot fi folosite ca modele pentru familii de secvenţe. Există trei chestiuni importante legate de utilizarea lor: topologia HMM, procesul de antrenare şi căutarea secvenţelor. După cum este descris în lucrări ca [KROG'94], [HUGH'96], modelele sunt formate din trei tipuri de stări: starea de potrivire (match state) care specifică probabilitatea de distribuţie a caracterelor (amino acid sau nucleotid) în fiecare poziţie conservată, stări de inserare care modelează goluri posibile dintre stări şi stări de ştergere (deletion states) care determină devierea unora din stările de potrivire. Odată ce au fost setaţi toţi parametrii modelului, poate fi calculată cea mai probabilă cale de-a lungul modelului pentru o secvenţă dată. În partea de antrenare, topologia HMM este dată în paralel şi în concordanţă cu o familie de secvenţe pe care dorim să o caracterizăm cu HMM. Parametrii modelului sunt estimaţi într-un mod în care el poate să genereze secvenţe similare cu cele din familia specificată. În [DURB'98] pot fi găsite explicaţii detaliate despre această abordare. Fiind dată o secvenţă, putem calcula (folosind algoritmul Viterbi) cel mai probabil aliniament al acestei secvenţe folosind tiparul reprezentat de HMM. Este menţionat în [HUGH'96] că în timp ce această metodă funcţionează, în practică trebuie aplicată căutarea în mod simultan în mai multe secvenţe datorită necestăţii unei cantităţi mari de date.

HMM au fost îmbunătăţite prin reducerea numărului de parametri implicaţi şi prin descoperirea de subfamilii. În primul caz înseamnă că topologia poate fi ajustată în timpul antrenării [HUGH'96] sau prin detectarea în secvenţe de tipare scurte care sunt imediat transformate în stări de potrivire în HMMs. Al doilea înseamnă că uneori familiile pot fi organizate în subfamilii aşa cum a fost făcut în [KROG'94] combinând câteva HMMs mici, cu topologie standard, într-unul mai mare. Un avantaj al acestui algoritm este că odată ce o parte a modelului este limitată la o subfamile numai secvenţele acestei subfamilii vor fi folosite pentru antrenarea acestei părţi de model.

3.4.3. Metode bazate pe alinierea secvenţelor

Găsirea tiparelor semnificative şi executarea aliniamentului local sunt funcţii strâns apropiate [BREJ'00]. Odată ce s-a executat aliniamentul a multiple secvenţe pot fi identificate mai uşor tiparele.

Propunerea autorilor din [GORO'97] este de a găsi cele mai semnificative secvenţe comune şi structuri dominante într-o mulţime de secvenţe ARN sau ADN. Idea este de a construi o schemă computaţională care aliniază local o colecţie de secvenţe ARN folosind constrngeri pentru secvenţe şi structuri. Ei identifică M≤N secvenţe conţinând cel mai semnificativ motiv comun care va apare în aliniamentul a M secvenţe. Subseturile slab aliniate sunt eliminate. În acest scop este folosit algoritmul lui Sankoff [SANK'85], pentru a alinia optim multiple secvenţe de ARN, dar el este simplificat datorită complexităţii de $O(L^{3N})$ a timpului, pentru N secvenţe de lungime L. Simplificarea este făcută în două moduri. Un mod este adaptarea algoritmului local Smith-Waterman pentru aliniamentul local al secvenţelor, maximizând un scor bazat pe o combinare a similarităţii secvenţelor şi structură. Această schimbare permite aflarea scorului cel mai ridicat al aliniamentului local al secvenţelor ARN. A doua simplificare este de a reduce complexitatea timpului la $O(L^4)$ prin nepermiterea structurilor ramificate. Autorii revendică faptul că nu se aşteaptă identificarea completă a motivelor structurale într-o singură trecere. De aceea, este aplicat un aliniament progresiv pentru perechi de secvenţe ca şi în tehnica folosită de Clustal 5.5 şi este urmată strategia algoritmului *„greedy"* în CONSENSUS [HERT'90] pentru a menţine soluţii intermediare cu scopul minimizării probabilităţii lipsei soluţiei optimale.

Datele folosite pentru investigare sunt din experimentele SELEX [TUER'90] (Sistematic Evolution of Ligands by Exponential Enrichment) pentre care s-a propus structura CONSENSUS. Constând în programare dinamica 4D şi un algoritm greedy pentru compararea perechilor de secvenţe, metoda a fost capabilă să recunoască complet aliniamentele publicate ale motivelor conservate. Nu întotdeauna a fost obţinută întreaga structură dar aliniamentul central este considerat a fi o dată de intrare utilă la metodele existente, în scopul completării identificării celor mai importante secvenţe şi motive structurale.

3.4.4. Metode care se bazează pe conceptul de graf

O presupunere despre aflarea tiparelor în secvenţele biologice este aceea că regiunile conservate în timpul evoluţiei sunt importante funcţional. Interesul constă în căutarea unui element regulator diferit de regiunile regulatorii de la multe gene ale aceleaşi specii. În acest sens, pot fi folosite regiunile regulatorii ale aceleiaşi gene luate de la mai multe specii. Aceasta se poate realiza folosind reprezentarea arborelui filogenetic al datelor biologice. În consecinţă, pot fi găsite tiparele scurte cel mai bine conservate în timpul evoluţiei [BREJ'00], [BLAN'00].

Folosirea conceptului de graf este găsită în [ESTE'04] unde a fost implementată o metodă top-down pentru minarea tiparelor celor mai frecvente în datele biologice. Fiind dată o bază de date de secvenţe D, un graf conceptual Γ şi o valoare suport minimă δ se poate descoperi mulţimea celor mai specifice tipare frecvente. Metoda descrisă este bazată pe următoarele observaţii cheie: tiparele frecvente trebuie să fie subsecvenţe ale secvenţelor de date; tiparele mai frecvente ar trebui generate înaintea celor mai puţin specifice pentru a evita generarea inutilă a tiparelor frecvente care nu sunt cele mai specifice. Pentru a expluata aceste observaţii, autorii din [ESTE'04] pornesc cu toate subsecvenţele identificate în datele de intrare (specificând o lungime maximă) şi continuă să le scurteze cu câte un element până când devin frecvente. Astfel, tiparele cu frecvenţă redusă sunt generalizate folosind conceptul graf până când ele devin frecvente.

3.4.5. Metode hibride

Există o varietate de alte metode care sunt realizate ca şi combinaţii ale metodelor deja prezentate. Pentru găsirea repetiţiilor tandem în secvenţele biologice, autorii [COWA'98] descriu o metodă iterativă pentru identificarea repetiţiilor pe baza propriului aliniament al unei secvenţe genomice date, în timp ce în [BENS'99] aceeaşi problemă este abordată folosind principii din teoria probabilităţii.

În această categorie sunt incluse metodele de minare bazate pe motoare web pentru căutarea de text care incorporează tehnici pentru interogări ale bazelor de date, potriviri şi prunning[24]. Lucrări în această direcţie sunt [WANG'00a] sau [REBH'98].

În [CHIA'04] sistemul de căutare în text este focalizat pe 4 tipuri de informaţii referitoare la gene: funcţii biologice, boli asociate, gene înrudite şi relaţii gene-gene. Sistemul de informaţii dezvoltat pentru gene (GIS) are două module: unul pentru filtrare (screening) şi unul pentru extragerea relaţiilor gene-gene. Primul furnizează informaţii despre funcţiile biologice, boli asociate şi gene înrudite pentru gena înterogată şi funcţia acestui modul este îndeplinită prin trei agenţi ai căutării de informaţie în documente (colectează documentele medicale din colecţia PubMed), selectarea propoziţiilor (selectează părţile importante ale abstractelor pentru o procesare ulterioară) şi analiza lexiconului (caută, numără şi indexează cuvintele cheie ale funcţiei biologice). Al doilea modul (activat după extragerea relaţilor genă-genă descrise în abstracte) estimează în ce măsură relaţia dintre o pereche de gene este pozitivă, cooperativă sau negativă.

3.5. Produse software dezvoltate pentru analiza secvenţelor biologice

Datorită cantităţii mari de date stocate automat este imperios necesară dezvoltarea de aplicaţii complexe în vederea explorării şi extragerii de informaţii cât mai utile. Prin urmare, există o multitudine de produse software destinate analizei secvenţelor biologice şi câteva dintre ele vor fi menţionate pe baza funcţiilor pe care le servesc.

Căutare în baze de date

Pentru simpla căutare în baze de date de secvenţe sunt menţionate câteva programe specializate. Majoritatea acestora folosesc tehnici bazate pe potrivire sau detalii de identificare existente în bazele de secvenţe. În general programele prezentate oferă mai mult decât simpla căutare sau identificare a unei secvenţe, aceasta putând fi doar o funcţie din ceea ce oferă toolul respectiv.

SRS-EMBL, **SRS6-EBI** (http://srs.embl-heidelberg.de:8000/srs5/, îşi derivă numele din Sequence Retrieval Systems, şi sunt programe utilizate pentru

[24] metodă care presupune înlăturarea ramificaţiilor inutile în pacurgerea unui algoritm

interogare generală a secvenţelor din baze de date aparţinând celor două instituţii EMBL şi EBI.http://srs6.ebi.ac.uk/srs6bin/cgi-bin/wgetz?-page+top)

Pfam –este atât o bază de secvenţe organizată pe grupuri de familii, ce conţine aliniamente multiple ale secvenţelor de ADN şi proteine cat şi un complex de programe care oferă posibiliteatea de căutare de informaţii în aceasta. Profilele modelelor Markov ascunse (profile HMM) construite din aliniamentele Pfam pot fi foarte utile pentru a recunoaşte dacă o nouă proteină aparţine sau nu unei familii existente chiar dacă omologia este slabă. Spre deosebire de metodele de aliniere a perechilor de secvenţe (e.g. BLAST, FASTA), HMMs Pfam lucrează sensibil cu proteinele care aparţin mai multor domenii. Ca detaliu, versiunea Pfam cea mai recentă, **18.0** (August 2005) conţine aliniamente şi modele pentru **7973** familii de proteine, furnizate de bazele de date de proteine **Swissprot 47.0** şi **SP-TrEMBL 30.0.** http://pfam.wustl.edu/. Din punct de vedere structural, Pfam este format în două moduri separate. Pfam-A sunt aliniamente multiple realizate de experţi umani în timp ce Pfam-B este o grupare generată automat a restului unei baze de date de proteine nonredundante derivată din baza de date PRODOM [BATE'04].

BLAST (**B**asic **L**ocal **A**lignment **S**earch **T**ool), este un program informatic complex care caută regiuni de similaritate locală între secvenţe. Programul compară secvenţele de nucleotide sau proteine cu secvenţe obţinute din diverse baze de date şi calculează semnificaţia statistică a potrivirilor. BLAST poate fi folosit şi pentru a deduce relaţii evoluţionare şi funcţionale între secvenţe precum şi pentru identificarea membrilor familiilor de gene. (http://www.ncbi.nlm.nih.gov/blast/Blast.cgi)

Prosite. Acest tool permite parcurgerea secvenţelor de proteine (fie din baza de cunoştinţe, Swiss-Prot/TrEMBL, fie furnizată de user) pentru detectarea aparţiilor de tipare, profile şi motive stocate în baza de date PROZITE sau caută alte baze de date de proteine pentru potriviri după motive specifice. (http://au.expasy.org/tools/scanprosite/)

PRINTS, reprezintă un compedium pentru ampretele proteinelor. O amprentă este un grup de motive conservate utilizate pentru a caracteriza o familie de proteine; puterea sa de diagnostic este rafinată de scanarea iterativă a amestecului de date oferite de SWISS-PROT/TrEMBL. De obicei motivele nu se suprapun, dar sunt separate de-a lungul unei secvenţe, deşi ele pot fi contigue (apropiate) în spaţiul 3D. Amprentele pot codifica plierile şi funcţionalităţile proteinelor mai flexibil şi mai puternic decât o pot face motivele singure, potenţialul de diagnostic complet derivând din contextul mutual oferit de vecinii motivelor. (http://umber.sbs.man.ac.uk/dbbrowser/PRINTS/)

InterPro este atât interfaţa web a unui motor de căutare cât şi o bază de date ce conţine familii de proteine, domenii şi locaţii funcţionale în care caracteristicile identificabile găsite în proteinele cunoscute pot fi aplicate secvenţelor de proteine necunoscute. (http://www.ebi.ac.uk/interpro/)

SMART (a Simple Modular Architecture Research Tool). Acest program permite identificarea şi adnotarea domeniilor genetice mobile şi analiza arhitecturii lor. Aceste domenii sunt extensiv adnotate ţinând cont de distribuţia pe linie de evoluţie, clasa funcţională, structuri terţiare şi reziduuri importante din punct de vedere funcţional. Fiecare domeniu găsit într-o bază de date nonredundante de proteine precum şi parametrii de căutare şi informaţiile taxonomice sunt stocate într-un sistem de baze de date relaţionale. Interfeţele utilizator la această bază de date permit căutări de proteine care conţin combinaţii specifice de domenii în gruparea definită. Toate detaliile sunt disponibile în publicaţiile asociate prezentării acestui tool. (http://smart.embl-heidelberg.de/)

Notă: În general, fiecare bază de date de secveţe biologice oferă pe lângă simpla stocare a datelor şi interfeţe cu mecanisme de căutare şi explorare a acestora.

Compararea de secvenţe biologice

Compararea perechilor de secvenţe biologice

Pentru compararea perechilor de secvenţe biologice există programe bazate pe diverse tehnici de similaritate care, conform afirmaţiilor existente pe site-ul oficial al Institutului European de Bioinformatică (EBI), recunosc omologiile apropiate cu o identitate mai mare de 30%.

În principal, algoritmii implicaţi în căutarea similarităţii şi aliniamentului de secvenţe folosesc două metode diferite de calculare (cantitativă) a similarităţii pentru o pereche de secvenţe. Ele sunt măsura distanţei şi măsura similarităţii (de obicei reprezentate sub forma unei matrice). În general, pentru orice pereche dată de secvenţe aceşti doi factori sunt invers proporţionali unul faţă de altul, astfel cu cât este mai mare distanţa dintre secvenţe cu atat este mai mică similaritatea şi vice versa.

În cele ce urmează atenţia va fi acordată la trei dintre cele mai utilizate moduri pentru parcurgerea bazelor de date de ADN şi de proteine în scopul determinării similarităţii unei secvenţe interogatoare cu o alta sau cu cele conţinute într-o bază de secvenţe. Primele două, FASTA şi BLAST sunt metode şi tooluri în acelaşi timp bazate pe algoritmi euristici care folosesc asumpţii şi aproximări. Ambele programme, cu toate că folosesc abordări diferite, mai întâi identifică potrivirile exacte şi foarte scurte dintre secvenţa de test şi cea/cele conţinute în baza de date. După care, cele mai bune potriviri scurte din prima etapă sunt extinse pentru a se face căutarea pentru secţiuni mai lungi de similaritate. În final, cele mai bune potriviri sunt optimizate cu ajutorul unor forme de programare dinamică.

BLAST (Basic Local Alignment Search Tool, http://www.ebi.ac.uk/blast/), amintit şi în categoria de programe cu ajutorul cărora se caută în baze de date de secvenţe biologice este folosit şi pentru a compara o secvenţă nouă cu cele conţinute în bazele de date de nucleotide sau proteine accesate prin alinierea noii secvenţe cu genele caracterizate anterior. Importanţa acestui tool este de a găsi regiuni de secvenţe similare care vor conţine chei funcţionale şi evoluţionare despre structura şi funcţia noii secvenţe. Regiunile de similaritate detectate via acest tip de

aliniament pot fi ori la nivel local unde regiunea de similaritate este bazată pe o singură locaţie sau la nivel global, unde regiunile de similaritate pot fi detectate de-a lungul codului genetic dealtfel neânrudit.

FASTA (pronunţat FAST-Aye, http://www.ebi.ac.uk/fasta33/) provine de la FAST-All, reflectând faptul că programul poate fi folosit pentru o comparare rapidă a proteinelor sau nucleotidelor. El atinge un nivel înalt de senzitivitate în căutarea similarităţii la o viteză ridicată. Aceasta performanţă este realizată prin executarea căutărilor optimizate pentru aliniamente locale folosind matricea de substituţie. Viteza ridicată a acestui program este obţinută folosind formele observate de potriviri de cuvinte (unde prin cuvânt se înţelege o secvenţă formată din 5-6 nucleotide sau amino acizi, în funcţie de dorinţa utilizatorului) pentru a identifica potenţialele potriviri înainte de începerea optimă a căutării, care este mare consumatoare de timp. Compromisul dintre viteză şi senzitivitate este controlat de un parametru care specifică mărimea cuvântului. Crescând valoarea acestui parametru descreşte numărul de potriviri de fond. Nu este investigată fiecare potrivire de cuvânt dar iniţial se caută segmente care să conţină ceva potriviri. Astfel, acest program demonstrează similaritatea secvenţelor şi căutarea omologiilor în baze de date de nucleotide şi proteine. El poate fi foarte specific când identifică regiuni lungi de similaritate scăzută în special pentru secvenţe puternic divergente. De asemenea se poate conduce căutarea similarităţii şi omologiei în baze de date complete, proteomice sau genomice.

Ssearch (http://ori.nibb.ac.jp/SIT/SSEARCH.html) este cea de-a treia abordare, numită Smith-Waterman şi care de asemenea este implementată sub forma unui tool. Este în totalitate bazată pe programare dinamică ce realizează efectiv toate comparaţiile posibile de perechi de secvenţe conţinute de baza de date. Prin urmare, este o tehnică mult mai senzitivă în comparaţie cu FASTA şi BLAST dar este mult mai costisitoare din punct de vedere al efortului computaţional şi al timpului. În general, senzitivitatea şi selectivitatea căutărilor făcute de FASTA şi BLAST sunt comparabile cu cele ale căutărilor în cazul Smith-Waterman datorită incorporării unor parametri statistici în cazul primelor două programe. Oricum, Smith-Waterman rămâne *"standardul de aur"* pentru aliniamentul perechilor de secvenţe proteină-proteină sau nucleotidă-nucleotidă, viteza sa putând fi îmbunătăţită prin utilizarea unui hardware specializat sau o platformă Linux extinsă [SSEA'05].

Software specializat pentru compararea de secvenţe multiple:

Compararea de secvenţe biologice multiple este strâns legată cu generarea de aliniamente multiple a secvenţelor. În acest scop, majoritatea programelor dezvoltate au la bază algoritmi destinaţi aliniamentului multiplu după care trecerea la utilizarea unei metode de determinare a similarităţii dintre secvenţele comparate. În general, toolurile existente sunt capabile să recunoască omologii distante cu o identitate mai mica decât 30%

PSI-BLAST. Denumirea provine de la Position Specific Iterative BLAST (PSI Blast) şi se referă la o caracteristică a lui BLAST 2.0 în care un profil (o tabelă care afişează frecvenţele[25] fiecărui amino acid în fiecare poziţie a secvenţei de proteine) este construit automat dintr-un aliniament multiplu al potrivirilor cu cele mai ridicate scoruri într-o căutare BLAST iniţială. Această strategie iterativă de căutare conduce la o creştere a senzitivităţii. (http://www.ncbi.nlm.nih.gov/BLAST/)

PDB BLAST. Acesta este o variantă îmbunătăţită a lui PSI BLAST pentru găsirea omologiilor distante ale secvenţei interogatoare într-o bază de date specifică. Metodele de căutare a profilelor pot găsi secvenţe omoloage mai distante decât în cazul căutării simple. PDB BLAST caută în baza de date PDB cu PSI BLAST folosind un profil generat din baza de date NR (Natural Resource, care este baza primară de date din NCBI). O asemenea îngustare a scopului căutării a fost ilustrat pentru a îmbunătăţi eficienţa căutării în baza de date NR. Prin urmare, căutarea PDB BLAST este compusă din două execuţii ale lui PSI BLAST: prima caută în baza de date NR sau alt stoc de secvenţe apoi, cu acest profil, execuţia finală caută în baza de date PDB (Protein Data Base). (http://bioinformatics.burnham-inst.org/pdb_blast/)

HMMer Server. Profilele modelelor Markov ascunse pot fi folosite pentru a face căutare senzitivă în bazele de date folosind descrieri statistice ale consensului[26] unei secvenţe de familii. În prezent, HMM-ul are o implementare software a profilelor pentru analiza secvenţelor de proteine distribuită gratuit. (http://bioweb.pasteur.fr/seqanal/motif/hmmer-uk.html)
O reprezentare sugestivă a relaţiilor dintre secvenţele biologice şi această metodă HMM este sugerată în figura următoare. Procedurile reprezentate şi care relaţionează cu bazele de secvenţe sunt implementate la adresa oficială indicată.

Sistemul software **SAM** (Sequence Alignment and Modelling System), http://www.soe.ucsc.edu/research/compbio/sam.html este o colecţie tooluri software flexibile pentru crearea, rafinarea şi utilizarea modelelor Markov ascunse liniare în vederea analizei secvenţelor biologice.

SAM Include programe şi scripturi pentru metoda SAM-T2K de detectare a omologiilor distante. Acesta este o metodă HMM iterativă de căutare pentru crearea unui HMM dintr-o singură secvenţă de proteine sau aliniament sursă folosind căutarea iterativă a secvenţei într-o bază de date. Metoda este considerată de autori cel mai senzitiv algoritm bazat pe determinarea de la distanţă a omologiei unei secvenţe.

---

[25] Frecvenţele sunt calculate din aliniamente multiple de secvenţe care conţin un domeniu de interes.

[26] În biologia moleculară şi bioinformatică, o secvenţă consensus este o cale de reprezentare a rezultatelor unui aliniament multiplu, unde sunt comparate una cu cealaltă secvenţe înrudite şi sunt găsite motive funcţionale similare. Secvenţa consensus arată ce reziduuri sunt conservate (sunt întotdeauna aceleaşi), şi care sunt variabile [WIKI'05h].

Detalii care descriu pe larg strategia de funcţionare a **SAM** pot fi consultate la adresa:http://www.cse.ucsc.edu/research/compbio/html_format_papers/hughkrogh96.

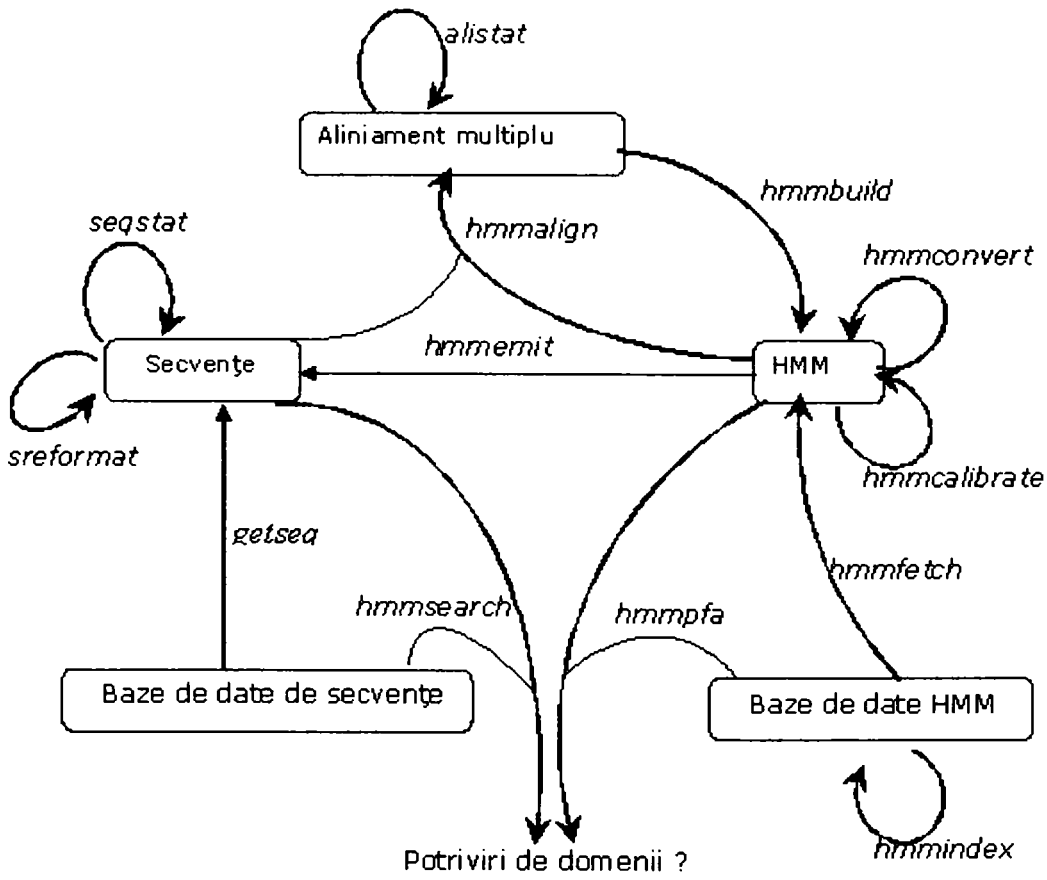


Figura 3.4.5-1. Reprezentarea relaţiilor dintre secvenţele biologice şi metoda HMM, figură preluată şi parţial tradusă din documentaţia detaliată, disponibilă la adresa de web a institutului Pasteur, Franţa (http://bioweb.pasteur.fr/seqanal/motif/hmmer-uk.html).

Tcoffee este un software care realizează aliniament multiplu de secvenţe, evaluarea aliniamentului multiplu, combinarea aliniamentelor multiple, alinierea de structuri (3DCoffee). După cum o descriu autorii metodei [NOTR'00], T-Coffee este o metodă pentru aliniamentul multiplu al secvenţelor care furnizează o îmbunătăţire

categorică a acurateţii în comparaţie cu cele mai comune alternative utilizate. Metoda este bazată în general pe abordarea uzuală a aliniamentului multiplu progresiv dar evită capcanele datorate naturii "greedy" a acestui algoritm. Cu T-coffee se preprocesează un set de date al tuturor perechilor de aliniamente dintre secvenţe. Acesta va furniza o bibliotecă de informaţie de aliniere care poate fi folosită pentru ghidarea alinamentului progresiv. Prin urmare, aliniamentele intermediare nu se bazează doar pe următoarea secvenţă care trebuie aliniată dar şi pe felul în care toate secvenţele sunt aliniate una cu alta. Aceste informaţii despre aliniament pot fi derivate din surse heterogene cum ar fi o mixtură de programme de aliniament şi/sau o superpoziţie a structurii. Biblioteca poate fi construită folosind o combinaţie de aliniament global sau local . Ca detalii, unele teste confirmă faptul că acest program complex are o acurateţe mai mare decât Clustal W (care urmează să fie descris) pentru secvenţe cu o identitate sub 30% dar, în schimb, este mai încet. Secvenţele de intrare sunt în format FASTA(Pearson) iar numărul maxim de secvenţe este 30 cu o lungime maximă de maximum 10 000 caractere.
http://igs-server.cnrs-mrs.fr/Tcoffee/tcoffee_cgi/index.cgi),

Clustal W este un program general pentru aliniament multiplu de secvenţe de proteine sau ADN. Aliniamentul se aplică pentru secvenţe cu diverse grade de divergenţă şi are o semnificaţie biologică, calculand cele mai bune potriviri pentru secvenţele selectate. El le aliniază astfel încât identităţile, similarităţile şi diferenţele pot fi văzute. Relaţiile evoluţionare pot fi văzute via cladograme[27] sau filograme[28] ce pot fi generate.
Metoda de bază a acestui aliniament constă din trei etape principale:

(i) toate perechile de secvenţe sunt aliniate separat cu scopul calculării matricei distanţă rezultînd divergenţa fiecărei perechi de secvenţe;
(ii) din matricea de distanţă este calculat un arbore ghid;
(iii) secvenţele sunt aliniate progresiv în conformitate cu ordinul de ramificare din arborele ghid.

Programul oferă posibilitatea alegerii modului de aliniament dintre o metodă rapidă de aproximare [BASH'87] ce permite să fie aliniate un număr foarte mare de secvenţe şi o alta mai înceată dar cu un grad mai mare de precizie. În cazul primei metode, scorurile sunt calculate folosind numărul de k-tupluri de potriviri (înseamnă reziduuri identice, de obicei de lungime 1 sau 2 pentru proteine sau de lungime 2-4 pentru secvenţe de nucleotide) în cel mai bun aliniament dintre două secvenţe minus o penalitate fixă pentru fiecare gap. A doua metodă, după cum o afirmă autorii, oferă o mai mare acurateţe a scorurilor, fiind complet bazată pe programarea dinamică şi foloseşte două penalităţi pentru gap-uri (pentru deschidere sau extindere) şi o matrice de ponderi a amino acizilor. Scorurile în acest caz sunt calculate ca număr de identităţi în cel mai bun aliniament împărţit la numărul de reziduuri comparate (poziţiile de gap sunt excluse).

Scorurile în ambele metode sunt calculate iniţial ca procent de scor identitate şi sunt convertite în distanţe prin impărţirea cu 100 şi apoi scăzând din

[27] Cladogramele sunt un arbori de diagrame folosiţi în clasifiarea biologica a membrilor din punct de vedere al evoluţiei pentru a ilustra relaţiile filogenetice [WIKI'05g]
[28] Filogramele sunt arbori filogenetici care indică relaţiile dintre diverse specii de plante sau animale grupate in funcţie de relatiile dintre ele şi de asemenea exprimă un sens al timpului sau ratei de evoluţie. Aspectul temporal al filogramei lipseşte dintr-o cladogramă [WIKI'05g]

1.0 pentru a furniza numărul de diferenţe per site. În aceste distanţe iniţiale nu se aplică nici o corecţie pentru substituţii multiple.
Pentru exemplificare, în Figura 3.4.5-2, preluată din materialul original [HIGG'94] sunt date 7 secvenţe biologice din baza de date SwissProt, al căror aliniament este calculat folosind metoda dinamică a programului Clustal W.

Arborele neorientat afişează toate ramurile cu lungimile proiectate pentru scalare. În arborele orientat, toate lungimile ramificaţiilor (semnifică numărul de diferenţe per reziduu de-a lungul fiecărei ramuri) sunt date la fel ca şi ponderile pentru fiecare secvenţă. În aliniamentul multiplu s-au folosit parametii impliciţi din Clustal W si seria PAM(3) de matrice de ponderi.

Spre deosebire de alte tooluri de aliniament multiplu, senzitivitatea metodelor de aliniament multiplu obţnuite a fost semnificativ îmbunătăţită pentru aliniamentul secvenţelor de proteine divergente. După cum afirma autorii lucrării [HIGG'94], Clustal W conţine unele caracteristici cum ar fi :

- Ponderile individuale sunt asociate fiecărei secvenţe într-un aliniament parţial cu scopul de a reduce ponderea secvenţelor aproape duplicate şi a o creşte pe aceea aparţinând celor mai divergente.
- Matricele de substituţie ale amino acizilor sunt variate la diferite stadii de aliniament în conformitate cu divergenţa secvenţelor care trebuie aliniate.
- Penalităţile specifice lipsei de reziduuri (residue-specific gap penalties) şi penalităţile reduse locale ale gap-urilor în regiunile hidrofilice încurajează noi gap-uri în potenţialele regiuni de nod (loop) mai degrabă decât structura secundară regulară.
- Poziţiile în primele aliniamente unde gap-urile au fost deschise primesc penalităţi locale reduse pentru a încuraja deschiderea de noi gap-uri la aceste poziţii. (http://www.ebi.ac.uk/clustalw/)

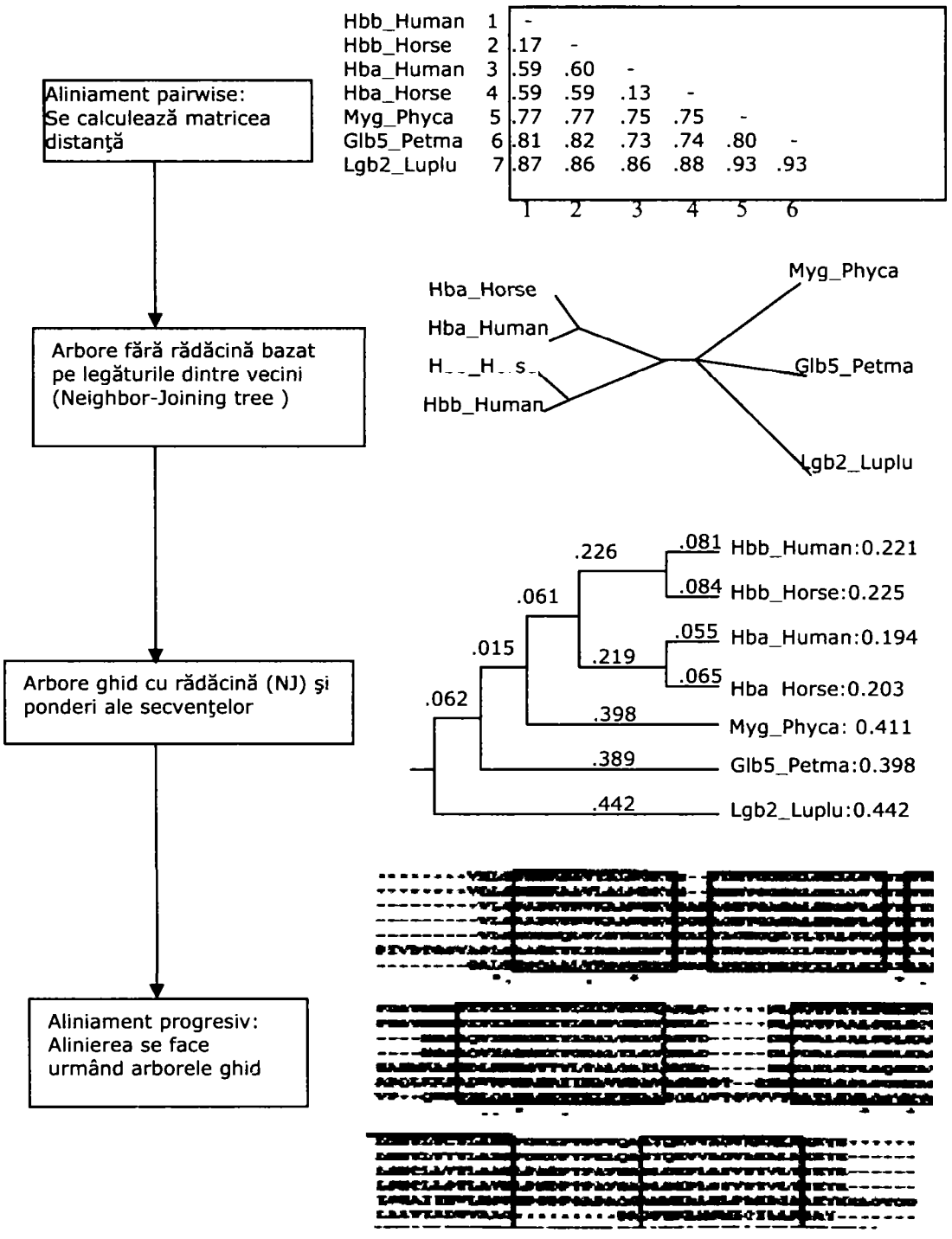


Figura 3.4.5-2: Rezltatul oferit de Clustal W în urma aplicării metodei dinamice pentru 7 secvenţe de proteine din baza de secvenţe SwissProt. Figura este reprodusa din lucrarea lui J.D.Thompson ©[THOM'94] care studiază îmbunătăţirea metodei de aliniament multiplu de secvenţe a lui Clustal W.

Software pentru recunoaşterea plierilor şi descompunerii secvenţelor biologice în structurile componente:

Proteinele pot fi clasificate fie după omologia secvenţelor (homology) fie după structura lor (folds=plieri). Ambele metode sunt pur informative. Schemele de clasificare în funcţie de plieri încearcă să "descompună" esenţa formelor structurale complicate ale protenei în chei, determinanţi comuni ai structurii lor (o codificare a structurii 3D). Aceste relaţii structurale abstracte pot revela în schimb relaţii evoluţionare şi/sau funcţionale între proteine cu un grad mare de divergenţă. Prin urmare, descompunerea secvenţelor (sequence threading) este o metodă de predicţie a structurii care este înrudită cu analiza profilului. Fiecare pliere cunoscută a unei proteine este reprezentată într-o matrice abstractă, bidimensională, de distanţe între reziduuri. Câteva dintre produsele software existente sunt amintite în cele ce urmează.

123D. 123D-este un program care combină profile de secvenţe, predicţia structurii secundare şi potenţialele capacităţi de contact pentru a descrie o secvenţă de proteine prin setul de structuri. Hidrofobicitatea este forţa majoră conducătoare pentru plierea proteinelor. Fără perechile de contact potenţiale se poate folosi un algoritm rapid de programare dinamică pentru alinierea unei secvenţe la o structură. Acest program apelează şi la serviciile alor subprograme specializate cum ar fi Fold library, Compare protein structures, Domain assignment.
(http://123d.ncifcrf.gov/123D+.html) Un exemplu este reprezentat în ANEXA 3.

FFAS (Fold & Function Assignment System) un program utilizat pentru analiza secvenţelor biologice din punct de vedere structural. Metoda de asociere a plierilor se bazează pe un algoritm de potrivire profil–la–profil. Profilul secvenţei interogatoare este generat cu PSI-BLAST folosind secvenţe din baza de date NR. El este potrivit cu profile ale secvenţelor din baza de date grupată PDB. Partea crucială a algoritmului este noua, bidimensionala schemă de ponderare care consideră topologia arborelui evoluţionar al tuturor proteinelor în familia omoloagă [RYCH'00]. (http://ffas.ljcrf.edu/ffas-cgi/cgi/ffas.pl). Detalii suplimentare ân ANEXA 3.

3D PSSM Fold Recognition Server, este o metodă web rapidă pentru recunoaşterea plierilor proteinelor folosind profilele secvenţelor reprezentate 1D sau 3D cuplate cu structura secundară şi Informaţia Potenţială de Solvatare (Solvation Potential Information). (http://www.sbg.bio.ic.ac.uk/~3dpssm/index2.html)
O exemplificare este prezentată în ANEXA 3.

Cn3D, este o aplicaţie ajutătoare pentru căutarea web care permite vizualizarea structurilor 3D din baze de secvenţe gestionate de NCBI Entrez. Această aplicaţie afişează simultan structura, secvenţa şi aliniamentul iar în prezent are o adnotare performantă şi caracteristici de editare ale aliniamentului.
(http://www.ncbi.nlm.nih.gov/Structure/CN3D/cn3d.shtml). Se poate vedea un exemplu în ANEXA 3.

Metodele şi exemplele de analiză şi comparare a secvenţelor biologice prezentate în acest ultim capitol nu şi-au propus să epuizeze toată cantitatea de produse şi metode existente la ora actuală ci să definească într-un mod cât mai sugestiv potenţialul oferit de metodele computaţionale aplicate în biologie. Astfel, toolurile prezentate fac parte din diverse categorii, în fucţie de aplicabilitatea lor şi tehnicile de prelucrare şi extragere a informaţiei din secvenţele biologice.

3.6. Concluzii

Deoarece tehnicile de recunoaştere au o largă aplicabilitate în domeniul bioinformaticii, iar obiectivul principal al tezei este de a veni cu o nouă soluţie la problemele de analiză şi căutare în baze de secvenţe biologice mari, pe parcursul acestui capitol s-a realizat o identificare a acestor tehnici în funcţie de scopul la care servesc, precum şi o investigare a produselor informatice existente. Astfel au fost identificate:

- metode de clasificare precum k-NN, clasificatori ce folosesc lanţuri sau modele Markov generalizate, clasificări pe baza identificării de caracteristici sau cu ajutorul reţelelor neuronale;
- algoritmi pentru căutare eficientă în baze de secvenţe având ca nuclee diverse metode de similaritate şi comparaţie a secvenţelor, care au fost deja prezentate în capitolul anterior;
- metode folosite în căutarea şi identificarea anumitor regiuni şi tipare folosind căutarea exhaustivă, metode de învăţare automată, metode bazate pe alinierea secvenţelor, metode care se bazează pe conceptul de graf şi metode hibride;
- produse informatice specializate care utilizează metodele de recunoaştere şi algoritmii de analiză şi comparaţie dezvoltaţi.

Până în momentul de faţă nu se cunoaşte vreo evaluare a performanţelor comparative a mai multor metode care să decidă asupra superiorităţii totale a uneia sau alteia. Aceasta se explică fie prin:

1. tipul de format al datelor prelucrate, deoarece nu este acelaşi lucru a se obţine secvenţe similare prin realizarea de aliniamente de secvenţe ca şi prin interogarea lingvistică a unor descrieri textuale despre aceste secvenţe;
2. scopul urmărit de utilizator. Scopul urmarit prin analiza şi/sau căutarea de secvenţe poate să fie de la unul simplu de a identifica obiectul de interes până la cel de a obţine detalii suplimentare despre respectivul „obiect".

Totuşi, dacă se presupune că se porneşte strict de la aceleaşi premise şi se urmăresc aceleaşi obiective se poate concluziona că până la urmă vor fi alese, ca şi în cazul metodelor de similaritate, metodele cu timpi computaţionali cât mai reduşi, consum minim de resurse şi gradul de acurateţe al rezultatelor ridicat. Astfel, metodele de comparare bazate pe profile şi lanţuri Markov ascunse şi care folosesc ideea de familie, permit biologului să deducă/obţină apropape de trei ori mai multe omologii decât un simplu algoritm bazat pe compararea perechilor [LIAO'03].

Multe dintre cele mai puternice metode de analiză a secvenţelor sunt acum bazate pe principiile modelării probabilistice. Exemple de asemenea metode includ:

- folosirea matricelor cu scoruri derivate probabilistic pentru a determina semnificaţia aliniamentelor secvenţelor,

- folosirea modelelor Markov ascunse ca baze pentru căutări de profile în scopul identificării membrilor la distanţă ai familiilor de secvenţe,
- şi inferenţa arborilor filogenetici folosind abordările probabilităţii maxime.

Caracteristic tuturor metodelor de comparaţie şi analiză este însă gradul ridicat de dependenţă de estimări de diverşi parametri, sau estimări biologice colectate din diverse surse. Astfel, dacă vorbim de metodele de aliniament este evidentă necesitatea implicării matricelor de substituţie şi a evaluărilor de penalităţi de gap sau inserţii şi ştergeri. În cazul în care explorăm folosind modelele Markov trebuie avută în vedere fie calcularea de diverse ponderi de tranziţie de la o stare la alta fie extragerea de „profile" din cadrul familiilor de secvenţe pentru a putea evalua gradul de aparteneţă al unei secvenţe la o familie sau alta.

Scopul principal al acestui capitol a fost de a include cele mai importante aspecte în practica actuală privind analiza secvenţelor biologice. Lista pubicaţiilor menţionate poate fi completată fară îndoială. Totuşi, cele mai specifice au fost întâmpinate. Datorită progresului continuu în cercetarea biologică, datele disponibile continuă să se extindă nu numai în termeni cantitativi dar de asemenea şi în ce priveşte formatele structurilor de stocare. Este de presupus astfel faptul că tehnicile de căutare şi analiză sunt adaptate acestui progres rapid rezultând în noi metodologii şi funcţii computaţionale.

Desigur că dezvoltarea aplicaţiilor software nu se limitează la utilizarea ADN/proteine sub formă de secvenţe biologice ci ele pot fi produse şi pentru o altă formă de reprezentare a genelor cum ar fi microvectorii (microarrays). Pentru acestea există de asemenea baze de date specializate cum ar fi ArrayExpress (locată la European Bioinformatics Institute http://www.ebi.ac.uk/arrayexpress/) şi analiza lor presupune tehnici specializate pentru procesarea imaginilor.

Atât acest capitol cât şi cel referitor la metodele de comparaţie a secvenţelor biologice pot fi considerate ca un cadru informaţional pentru a motiva şi a atrage atenţia asupra contribuţiei personale la dezvoltarea unei noi metode de comparaţie, analiză şi căutare în baze de secvenţe, ce va fi descrisă şi validată în capitolele urmatoare.

4. ANALIZA ŞI EVALUAREA CONTEXTULUI LINGVISTIC CU AJUTORUL MODELELOR STATISTICE PENTRU SISTEMELE DE RECUNOAŞTERE

Acest capitol îşi doreşte să prezinte la un nivel de percepţie cât mai facil, câteva fundamente matematice şi teoretice şi aplicarea lor în modelarea lingvistică din punct de vedere statistic, ce a devenit aşa după cum o arată şi multiple exemple, un instrument puternic şi eficient în procesarea şi extragerea informaţiei dintr-o cantitate mare de date .

O importanţă majoră în multe direcţii de cercetare o au lanţurile Markov astfel, ele îşi demonstrează utilitatea inclusiv în domeniul procesării lingvistice. În capitolul de faţă este exemplificată utilizarea lanţurilor Markov în modelarea limbajului natural, constituind şi elementul de bază al abordării experimentale conţinute de această teză. Evaluarea acestor modele se realizează cu măsuri ca entropia şi perplexitatea. Conform "Teoriei Informaţiei", există mai multe tipuri de entropie cunoscute şi utilizate la ora actuală dar pentru scopul propus, atenţia s-a limitat la formele de entropie şi cross-entropie (sau entropie încrucişată).

Valorificarea noţiunilor teoretice este regăsită în aplicaţia experimentală, motivând utilizarea lor precum şi deducerea de noi raţionamente.

Deoarece partea statistică şi a teoriei decizionale este cea care contribuie în mod esenţial la obţinerea de rezultate eficiente în rezolvarea problemelor atât în recunoaştere cât şi în procesarea şi explorarea informaţiei utilizând modelele lingvistice, lucrarea de faţă i-a acorat acest spaţiu.

Aplicaţia practică privind estimarea şi evaluarea modelelor lingvistice permite observarea şi analiza diverselor aspecte ale rezultatelor obţinute în vederea unor posibile direcţii de investigare concretizate ulterioar.

4.1. Fundament teoretic

4.1.1. Lanţuri Markov lingvistice

În matematică, un lanţ Markov (în timp discret), numit după matematicianul rus Andrei Andreievici Markov, este un proces stochastic în timp discret cu proprietatea Markov. Într-un asemenea proces trecutul este irelevant pentru previziunile viitorului cunoscând prezentul. Există de asemenea lanţuri Markov în timp continuu dar care nu fac parte din sfera de interes în acest moment.

Un lanţ Markov este o secvenţă X_1, X_2, X_3, ... de variabile aleatoare. Acoperirea acestor variabile, adică mulţimea valorilor posibile este numită spaţiul stărilor, valoarea lui X_n fiind starea procesului la momentul n. Dacă distribuţia probabilităţii condiţionate a lui X_{n+1} în starile anterioare depinde doar de X_n singur, atunci:

$$P\left(X_{n+1} = x \middle| X_0 X_1 X_2, \ldots, X_n\right) = P\left(X_{n+1} = x \middle| X_n\right) \qquad (4.1.1\text{-}1)$$

Unde x este una dintre stările procesului. Identitatea menţionată este proprietatea Markov [MARK'71]. Andrei Markov a venit cu primele rezultate pentru asemenea procese în 1906.

În lucrearea de importanţă majoră *"A Mathematical Theory of Communication"* [SHAN'48], Claude Shannon propune utilizarea lanţului Markov pentru a crea un model statistic din secvenţe de litere dintr-un fragment de text în limba engleză. Shanon a aproximat structura statistică a unui fragment de text folosind un model matematic Markov. Astfel, un model Markov de ordinul 0 prezice că fiecare literă din alfabet apare cu o probabilitatea fixă. Prin urmare, pentru o secvenţă "agggcagcgggcg", modelul Markov de ordin 0 prezice că litera "a" apare cu probabilitatea 2/13, "c" cu probabilitatea 3/13 şi "g" cu probabilitatea 8/13. Secvenţa următoare este un exemplu tipic pentru acest model:

a g g c g a g g g a g c g g c a g g g g . . .

Un model de ordinul 0 presupune că fiecare literă este aleasă independent. Aceasta nu coincide cu proprietăţile statistice ale limbajului (limbii engleze în acest caz) deoarece există o o mare corelaţie între literele successive dintr-un cuvânt sau propoziţie. Spre exemplu, litera "h" şi "r" cel mai adesea urmează literei "t" decât "c" sau "x". Se obţine un model mult mai rafinat dacă se permite exprimarea probabilităţii fiecărei litere successive care să depindă de cea anterioară sau cele anterioare. Prin urmare, un model Markov de ordinal k prezice că fiecare literă apare cu o probabilitate fixă dar acea probabilitate poate depinde de k litere consecutive anterioare (k-gram). De exemplu, dacă textul are 100 de apariţii ale lui "th", cu 60 de apariţii ale lui "the", 25 apariţii cu "thi", 10 apariţii cu "tha" şi 5 apariţii cu "tho", atunci modelul Markov de ordinal 2 prezice litera următoare urmând modelul 2-gram "th" şi este: "e" cu probabilitatea 3/5, "i" cu probabilitatea 1/4, "a"cu probabilitatea 1/10 şi "o" cu probabilitatea 1/20. Reprezentarea stărilor şi o simulare a generării de text pseudo-aleator poate fi consultată în ANEXA 4.

4.1.2. Construirea modelelor lingvistice statistice

În toate abordările modelelor lingvistice din diverse domenii de aplicabilitate definirea lor este aceeaşi. Prin urmare, un model lingvistic statistic este considerat o probabilitate de distribuţie peste toate propoziţiile sau alte unităţi lingvistice într-un limbaj [ROSE'00]. Acesta poate fi văzut de asemenea ca un model statistic pentru

generarea de text. În general, funcţia modelării lingvistice răspunde la întrebarea: cât de probabil al *i*-lea cuvânt dintr-o propoziţie s-ar produce cunoscând cele *i*-1 cuvinte precedente? În cele mai multe aplicaţii ale modelarii lingvistice, cum ar fi recunoaşterea vorbirii şi explorarea informaţiei, probabilitatea unei propoziţii este descompusă într-un produs de probabilităţi *n*-gram.

Dacă S ar fi o secvenţă compusă din *k* cuvinte,

$$S = w_1, w_2, \ldots, w_k .$$

Un model lingvistic *n*-gram consideră secvenţa de cuvinte S a fi un proces Markov cu probabilitatea

$$P_n(S) = \prod_{i=1}^{k} P\left(w_i \middle| w_{i-1}, w_{i-2}, w_{i-3}, \ldots, w_{i-n+1}\right), \qquad (4.1.2\text{-}1)$$

unde *n* referă ordinul procesului Markov. Când *n*= 2 spunem că avem un model lingvistic numit bigram ce este estimat folosind informaţii despre coexistenţa perechilor de cuvinte. În cazul în care *n*=1, spunem că avem unigram ce foloseşte numai estimări ale probabilităţilor cuvintelor individuale. Pentru aplicaţii ca recunoaşterea vorbirii sau traducătoare automate ordinea cuvintelor este importantă şi sunt folosite modelele de ordin mai înalt (de obicei trigram). În extragerea de informaţii (information retrieval) rolul ordinii cuvintelor este mai puţin clar şi modelele unigram au fost folosite extensiv.

Pentru stabilirea modelului lingvistic al cuvintelor, estimatorii probabilistici sunt de obicei derivaţi din frecvenţa modelelor *n*-gram din setul de date de antrenare. De obicei multe elemente de tipul *n*-gram este posibil să nu apară în setul de date curent utililizat pentru estimare, chiar dacă mărimea setului este imensă şi valoarea lui *n* este mică. Ca o consecinţă, pentru evenimentele rare sau neântâlnite, estimările probabilităţilor care sunt direct bazate pe numărarea frecvenţelor de apariţie devin o problemă. Această situaţie este adesea calificată drept „inconsistenţa setului de date". Ajustarea (smoothing) este folosită pentru rezolvarea problemei şi constituie o parte importantă în orice model lingvistic. Dintre cei mai cunoscuţi algoritmi sunt: „Add-one", „Witten-Bell Discounting", „Good-Turing Discounting", „Backoff", „Deleted Interpolation" majoritatea propuşi pentru recunoşterea vorbirii ce sunt descrişi în detaliu în [JURA'00]. Există însă şi alte metode de smoothing utilizate în regăsirea informaţiei (IR) şi care sunt: *ajustarea parametrilor* cunoscută şi ca „parameter smooting" sau *ajustare semantică* („semantic smoothing") descrişi în [LIU'04].

4.1.3. Măsuri de evaluare a modelelor lingvistice statistice

În practică, *entropia* şi *perplexitatea* sunt cele mai cunoscute metrici folosite în evaluarea sistemelor bazate pe lanturi Markov, cunoscute şi ca n-grame [JURA'00] şi pot fi exprimate în modele matematice într-o formă exactă.

În rezolvarea reală a problemelor este necesară însă estimarea probabilităţilor din observări ale proceselor aleatoare.

Vizând în mod direct domeniul lingvistic, din punct de vedere al eficienţei, un alfabet sursă întâlnit în practică ar trebui să aibă o distribuţie a probabilităţii mai mică decât cea optimă. Dacă alfabetul sursă este compus din *n* simboluri, atunci el poate fi comparat cu un "alfabet optimizat" cu *n* simboluri a căror distribuţie e uniformă. Rata entropiei alfabetului sursă cu entropia versiunii sale optimizate reprezintă eficienţa alfabetului sursă ce poate fi exprimată ca procent. Aceasta implică faptul că eficienţa unui alfabet sursă cu *n* simboluri poate fi simplu definită ca fiind egală cu entropia sa *n*-ară [WIKI'05a]. Uneori evaluarea modelelor lingvistice este realizată şi folosind o măsură numită „perplexitate".

4.1.3.1. Entropia

Originea conceptului de entopie este legată de Ludyig Boltzmamm (1877) şi i-a fost dată o interpretare probabilistică de Claude Shannon.

Adoptând descrierea făcută de Richard O. Duda în cartea sa *Pattern classification* [DUDA'01], din punct de vedere matematic, pentru o mulţime discretă de simboluri $\chi = \{v_1, v_2, \cdots, v_m\}$ cu probabilităţile asociate P_i, entropia distribuţiei discrete – o măsură a caracterului aleator şi impredictibil al unei secvenţe de simboluri obţinută din aceasta – este:

$$H(x) = -\sum_{i=1}^{m} P_i \log_2 P_i \quad , \qquad (4.1.3.1\text{-}1)$$

unde entropia este măsurată în *biţi* atunci când folosim logaritmul în baza 2. Pentru distribuţiile continue se foloseşte adesea baza *e* sau logaritmul natural notat *ln,* caz în care entropia este exprimată în "*nats*". În cazul în care o probabilitate dispare se apelează la faptul că $\lim_{P \to 0} P \log P = 0$ pentru a defini $0 \log 0$.

Făcând referire la aplicabilitatea entropiei la modul general, cartea lui Dorian Pyle, *"Data Preparation for Data Mining"* [PYLE'99] vine cu exemple de rapoarte ale analizei entropice alături de discuţii despre cum pot fi folosite aceste informaţii într-o prima fază de parcurgere a investigării datelor.

Utilizând calitatea de estimator al distanţei informaţiei, s-au dezvoltat algoritmi cu aplicabilitate în bioinformatică şi domeniul lingvistic [KALT'04]. Cele mai cunoscute şi aplicate forme ale entropiei sunt: entropia încrucişată, entropia de legatura, entropia condiţionată şi informaţia mutuală. În acest moment însă interesul va fi orientat către entropia încrucişată şi doar la nivel de definiţie spre perplexitate.

4.1.3.2. Entropia încrucişată

Entropia încrucişată ("*Cross-Entropy*") tradusă şi "*cross-entropie*", este utilă când nu cunoaştem probabilitatea de distribuţie care a generat datele actuale. Prin urmare, cross-entropia dintre o variabilă aleatoare X cu distribuţia reală a probabilităţii p(x) şi o altă funcţie model cu probabilitate q(x) (de obicei un model al lui p) este dată de expresia:

$$PH(X,q) = -\sum_{x} p(x)\log q(x)\text{ , unde } x \text{ parcurge X [MANN'00].} \qquad (4.1.3.2.\text{-}1)$$

Întâlnită în aplicaţiile ce vizează modelarea lingvistică, această entropie "încrucişată" presupune că termenul pentru care se aplică *log* este dedus dintr-un set diferit de cel folosit pentru estimarea ponderilor pentru modelul curent, numit şi set de antrenare şi, conform [VANC'03] se defineşte în modul următor:

$$C(x) = -\sum_{i} p_{test}(x_i)\log_2 p_{train}(x_i)$$
$$C(x) \geq H(x) \qquad (4.1.3.2\text{-}2)$$

$$\Downarrow$$

C(x) este considerată *o limită superioară* pentru estimarea unui proces aleator aşa după cum rezultă şi din [JURA'00].

În practică, această măsură a devenit larg utilizată. În sistemele de recunoaşterea vorbirii şi în modelarea lingvistică ea este aplicată pentru explorări ale conţinutului textual util în diverse domenii [JURA'00], [MANN'00], [YOUN'97], [DEMA'95].

Mai mult decât atât, după cum este descris în [YAO'04], entropia încrucişată este utilizată ca măsură a performanţei unei noi metode de identificare a sesiunilor[29] în explorarea bazelor de date. Alegerea acestei măsuri este motivată de relaţia dintre predicţie şi compresie (a se vedea în [YAO'04]) iar avantajul utilizării ei în cazul modelelor *n*-gram pentru detectarea sesiunilor se datorează faptului că pentru a calcula cross-entropia nu este necesară cunoaşterea limitelor reale ale sesiunii de căutare în datele de test. Această abordare are un impact semnificativ în domeniul bazelor de date şi data mining. Metoda de identificare îmbunătăţită a sesiunilor poate fi de asemenea aplicată în web logs[30] pentru a determina date mai bune şi de o mai mare acurateţe pentru *web log mining*.

---

[29] o sesiune db este o secvenţă de cereri prezentată sistemului db in scopul obtinerii unei anume operaţiuni

[30] Un weblog (în prezent cunoscut şi sub numele de blog) este o publicaţie web ce conţine în mod obişnuit articole periodice. Cu toate că majoritatea weblog-urilor erau manual actualizate, în prezent există aplicaţii care să automatizeze întreţinerea unor asemenea site-uri. Blog-urile pot fi de diverse forma, pornind de la jurnale individuale pînă la campanii politice, programe media şi corporaţii. Ele pot fi scrise de un autor ocazional sau ca urmare a colaborarilor unei comunităţi de autori. Multe weblog-uri permit vizitatorilor să lase comentarii publice. Totalitatea weblog-urilor sau blog- site-uri înrudite este adesea numită blogsferă. Când o mare cantitate de activităţi, informaţii, opinii erup în jurul unui subiect particular sau controversat in blogsferă se consideră că are loc o furtuna_blog sau roi_blog (conform dictionarului wikipedia)

Pentru a întregi viziunea supra aplicabilităţii cross-entropiei, în [KROE'04] ea este propusă şi ca alternativă la metodele clasice de grupare cum ar fi *k-medii*, *fuzzy k-medii* şi *cuantificarea vectorială liniară*. Autorii susţin că algoritmul propus este rapid şi sigur prezentând şi avantajele simplicităţii eliminând condiţiile de start şi erorile de eşantionare comparativ cu alte euristici care presupun simularea unor normalizatori sau căutări locale ghidate.

4.1.3.3. Perplexitatea

Perplexitatea este şi ea o metrică dedusă din valoarea entropiei şi este definită în modul următor:

$$PP(x) = 2^{H(x)}. \qquad (4.1.3.3\text{-}1)$$

Pentru o variabilă aleatoare uniform distribuită cu N valori posibile, entropia şi perplexitatea în acest caz sunt *N*.

Aşa cum există mai multe moduri de exprimare a entropiei, perplexitatea este şi ea denumită în funcţie de entropia care o determină. Se poate întâlni astfel perplexitate condiţionată [SHAN'51] care pentru un corpus suficient de mare este de obicei un indicator al cantităţii de informaţie furnizate de model. Cu cât este mai mică perplexitatea condiţionată cu atât modelul conţine mai multă informaţie; prin urmare un model mai bun. Aceasta se datorează faptului că modelul captează cât mai multă informaţie şi orice incertitudine rămasă este reflectată în perplexitatea condiţionată [ROSE'94]. Cea mai mare aplicabilitate o are în modelarea lingvistică pentru sistemele de recunoaştere a vorbirii, în măsurarea confuzabilităţii modelelor lingvistice. Ea depinde de marimea vocabularului (care este indicator dacă nu există un model lingvistic relevant) şi factorul de ramificare (în medie, numarul de cuvinte care pot urma după un cuvânt dat).

4.2. Implementarea unei aplicaţii pentru analiza şi evaluarea modelelor lingvistice[31].

Urmând etapele standard necesare în estimarea şi evaluarea modelelor lingvistice, a fost dezvoltat un set de programe utilizate pentru explorarea potenţialului lingvistic al modelelor statistice şi evaluarea lor. Ca urmare, în acest capitol vor fi prezentate şi analizate rezultatele obţinute în fiecare stadiu experimental. Este necesar de menţionat faptul că aceste rezultate exprimă gradul de predictibilitate al unui limbaj (limba engleză în acest caz) la scară redusă, pe motivul limitării corpusului experimental dar şi vine cu observaţii noi privind posibilitatea investigării mult mai profunde a unor aspecte particulare cu aplicabilitate în explorarea informaţiei (Information Retrieval).

[31] Implementarea programelor şi rezultatele experimentale au fost realizate în timpul unei burse de studiu şi cercetare oferită de Universitatea Katolică din Leuven, Belgia, departamentul ESAT-PSI, Speech group.

4.2.1. Descrierea aplicaţiei

Implementarea acestei aplicaţiei presupune un set de programe independente din punct de vedere al implementării dar dependente din punct de vedere funcţional. Ele sunt scrise în limbalul Perl[32] şi oferă posiblitatea de a explora într-un mod flexibil comportamentul măsurilor destinate informaţiei aflate sub formă de text. În continuare va fi dată o descriere a fiecăruia dintre aceste programe alături de detaliile necesare privind funcţionalitatea lor. Ordinea de execuţie este: procesarea textului, generarea modelului lingvistic, evaluarea entropiei simple şi evaluarea entropiei după aplicarea unei metode de ajustare. Programele pot să fie executate atat pe platforma oferita de sistemele de operare Unix cât şi Windows fiind rulate din linia de comandă.

4.2.2. Procesarea textului

Textele de obicei conţin nu numai cuvinte ci şi semne de punctuaţie sau semne adiţionale pe care dorim sau nu să le considerăm în timpul procesului de evaluare, în funcţie de scopul dorit. În termeni tehnici, aceasta este considerată curăţarea textului ("text cleaning").

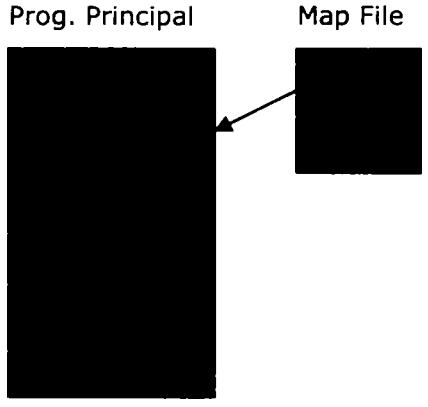
Normalizarea este un aspect semnificativ în dezvolarea multor aplicaţii de procesare a textelor. Una din activităţile importante implică rezolvarea problemelor de ambiguitate privind literele scrise cu majuscule. Spre exemplu, în cazurile textelor mixte, cuvintele scrise cu majuscule denotă de obicei nume proprii dar există poziţii speciale în text unde acestea sunt aşteptate. Rezolvarea ambiguităţii acestor cuvinte conduce în general la identificarea numelor proprii cu toate că în multe cazuri numele proprii pot coincide cu simple cuvinte existente în vocabularul normal.Ex: "Pasare", "Negru"....În general, această problemă presupune o analiză mai amănunţită. O altă funcţie a normalizării textului este determinarea limitelor propoziţiilor care de asemenea poate conţine situaţii ambigue presupunând la rândul lor o analiză mai aprofundată. În cele mai multe cazuri, împărţirea unui text în propoziţii este o chestiune simplă: un punct, un semn al exclamării sau întrebării semnalează sfârşitul de propoziţie. Mai sunt însă şi cazuri în care un punct semnifică un punct zecimal sau este o parte a unui abreviaţii. Prin urmare nu va semnala sfarsitul de propoziţie dar pot exista situaţii în care o abreviere însăşi poate fi ultima componentă a unei propoziţii. Această problemă este de asemenea exacerbată de faptul că abrevierile nu formează un set închis, adică nu pot fi listate toate abrevierile posibile. Poate fi chiar problematic faptul că unele abrevieri coincid cu cuvinte obisnuite: "in", poate fi abreviere pentru "inch", "no" poate fi abreviere pentru "number", "bus" pentru "business" etc. Prin urmare, în timpul procesării textului utilizatorul trebuie să se gândească foarte bine ce fel de modificări generale doreşte să execute. În acest sens, aplicaţia dezvoltată ajută utilizatorul să "cureţe textul" după propriile criterii pe care le menţionează într-un fisier standard. Programul foloseşte un fişier adiţional care conţine expresii regulare definite de utilizator sau asupra cărora acesta poate să intervină. Pentru experimentele curente, userul are posibilitatea de a manipula următorele funcţii:

[32] Practical Extraction and Reporting Language, este un limbaj de programare folosit adesea pentru crearea de programe CGI (***C**ommon **G**ateway **I**nterface*)

- Marcarea limitelor propoziţiilor considerând că fiecare propoziţie începe la un rând nou şi este finalizată la întâlnirea unuia sau mai multor semne de punctuaţie urmate de caracterul ce desemnează linie nouă;
- Înlăturarea semnelor de punctuaţie;
- Substituirea unor caractere sau şiruri de caractere;
- Înlăturarea oricărei expresii numerice sau substituirea ei cu caracterul dorit.

În cazul absenţei acestui fişier nu va fi făcută nici o modificare asupra textului. Ieşirea generată de execuţia acestui prim program o constituie un nou fişier text care se va constitui fişier de intrare pentru următorul program. Pentru detalii despre modul în care acest program funcţionează poate fi utilizată opţiunea de help.

Reprezentarea procesului 1:



Map file – conţine expresii regulare definite de user urmând reguli predefinite.
User-ul poate manipula task-urile:
- Marcarea sfârşitului de propoziţii;
- Înlăturarea punctuaţiei;
- Substituirea caracterelor şi a sirurilor de caractere;
- Înlăturarea oricărei expresii numerice sau substituirea ei cu un caracter de marcaj.

Programul principal – execută modificările stabilite de user în map file => un nou fişier text.
Dacă *Map file* lipseşte – nu se execută vreo modificare asupra textului de intrare.

Experimente:

În această primă etapă este utilizată o colecţie de 536 fişiere text din Wall Street Journal (WSJ) corpus, conţinând 1527 linii, 304694 cuvinte şi 1845026 caractere. Acest corpus este deja considerat normalizat prin urmare este corectă presupunerea conform căreia fiecare propoziţie începe la o linie nouă.
Cantitatea totală de text a fost împărţită în două. Un set de 183 fişiere (5167 linii, 103970 cuvinte, 629966 caractere) şi un set de antrenare de 352 fişiere(10109 linii, 200724 cuvinte, 1215060 caractere). Asupra fiecărui set a fost aplicată aceeaşi procesare:

- Punctuaţia considerată nesemnificativă în estimarea entropiei la nivel de literă a fost înlăturată. Prin urmare, caractere ca "- ( ) [] : ; ' | " au fost înlăturate păstrând în schimb punctele din interiorul propoziţiilor deoarece există ambiguităţi în decizia funcţiei lor sintactice. De asemenea, sunt păstrate marcajele apostrof întâlnite în forma condensată a cuvintelor cu toate că există situaţii în care sunt utilizate cu rol de marcaj de citat;
- Sunt păstrate toate cifrele însoţite sau nu de £ ,$, # ce pot simboliza tipuri de monede;

- Spaţiile multiple din interiorul propoziţiilor sunt substituite cu spaţii simple şi pentru uşurinţa interpretării au fost înlocuite cu caracterul ^ ;
- Pentru a urmări limitele propoziţiilor, ele sunt marcate.

4.2.3. Generatorul de modele lingvistice

Funcţia clasică a modelării lingvistice este de a prezice cuvântul următor cunoscându-le pe cele anterioare. Urmând experimentul lui Shannon din 1951 de a ghici următorul caracter într-un text au fost dezvoltate multe aplicaţii. În mod asemănător cu explicaţiile pentru modelele *n*-gram la nivel de cuvânt [MANN'00] şi la nivel de litere avem aceeaşi abordare. Funcţia de predicţie a următorului caracter/literă poate fi considerată ca tentativă de estimare a funcţiei de probabilitate P:

$$P(x_n | x_1, \ldots, x_n). \qquad (4.2.3\text{-}1)$$

Într-o asemenea abordare probabilistică este folosită o clasificare pe baza caracterelor anterioare, *istoria*, pentru prezicerea caracterului următor. Având la bază examinarea unei mari cantităţi de text, putem prezice care literă tinde să urmeze alte litere. Prin urmare, este necesară o metodă de a grupa *istoriile* care sunt oarecum similare, cu scopul de a furniza predicţii rezonabile referitoare la litera aşteptată să urmeze. O cale posibilă de a le grupa o constituie presupunerea Markov conform căreia numai contextul local anterior –ultimele puţine litere- afectează urmatoarea literă. Prin urmare, dacă este construit un model în care toate istoriile care au aceleaşi (*n*-1) litere plasate în aceeaşi clasă de echivalenţă atunci avem un model Markov de ordinul (*n*-1) sau un *n*-gram model de litere (ultima literă a *n*-gram-aticii fiind litera pe care o prezicem).

În aplicaţia dezvoltată, utilizatorul poate genera modele lingvistice pentru orice valoare dorită a lui *n*. Valoarea implicită este 1. Astfel, se poate obţine mărimea curentă a vocabularului, valoare necesară în evaluarea entropiei. O funcţie suplimentară o constituie şi construirea unei histograme pentru a urmari frecvenţa apariţiilor unor anumite *n*-gram-atici.

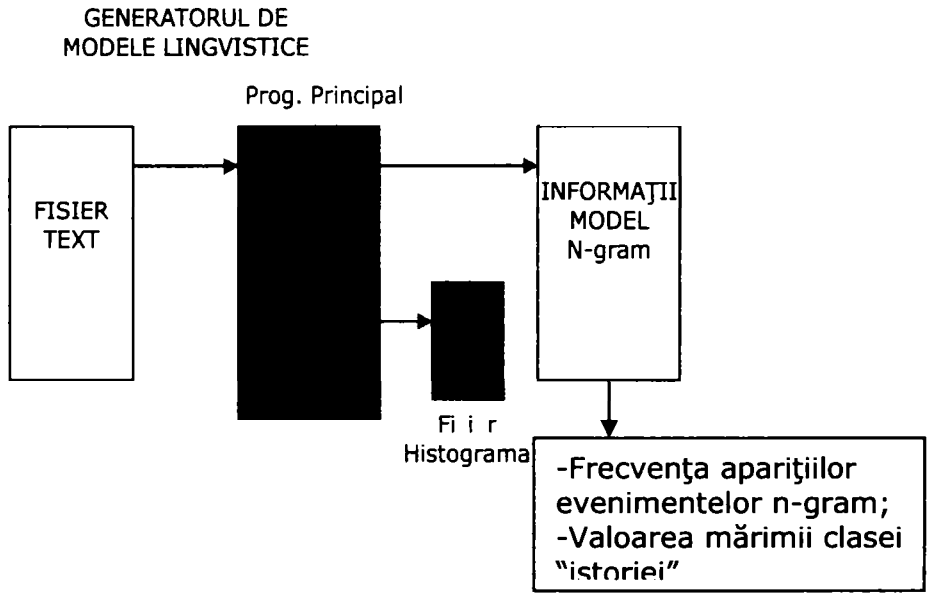
Exemplu:
Numărul evenimentelor *n*-gram ce au apărut de 5 ori = 14 (40.94 %)

Aceasta înseamnă că există 14 evenimente *n*-gram distincte care se repetă de 5 ori fiecare şi ele reprezintă în jur de 41% din numărul total de unităţi *n*-gram identificate. Acest tip de informaţie poate fi utilizat pentru o analiză detaliată a conţinutului textual.

Experimente:

Utilizând fişierele rezultate în etapele de procesare a textului au fost generate modele pentru 2,3,4,...20 –gram-e. Dimensiunea vocabularului utilizat este obţinută din generarea 1-gram şi are valoarea de 74 (poate fi văzut în tabelul din ANEXA 4).

Reprezentarea procesului 2:



4.2.4. Estimarea entropiei şi analiza rezultatelor

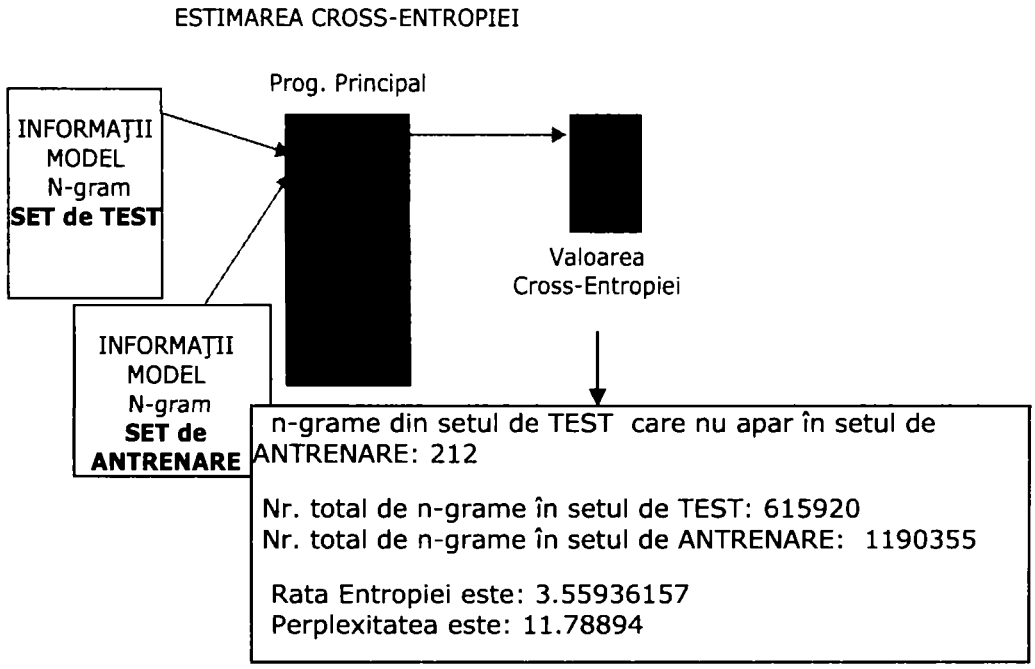
Pentru estimarea entropiei în cazul fiecărui model sunt utilizate fişiere rezultate din etapa anterioară astfel: pentru fiecare unic eveniment *n*-gram din setul de testare este evaluată ecuaţia 4.1.3-1, echivalentă în acest caz cu entropia încrucişată exprimată în (4.1.3.2-1). Prima probabilitate este obţinută din frecvenţa relativă a evenimentelor *n*-gram din setul de testare şi estimarea logaritmică este log_2 din probabilitatea condiţionată a aceluiaşi eveniment dar estimată din setul de antrenare. Şi această etapă furnizează informaţii suplimentare privind cantitatea de evenimente implicate în estimarea entropiei, informaţie ce poate fi de asemenea utilizată pentru o analiză suplimentară şi aprofundată a textului utilizat.

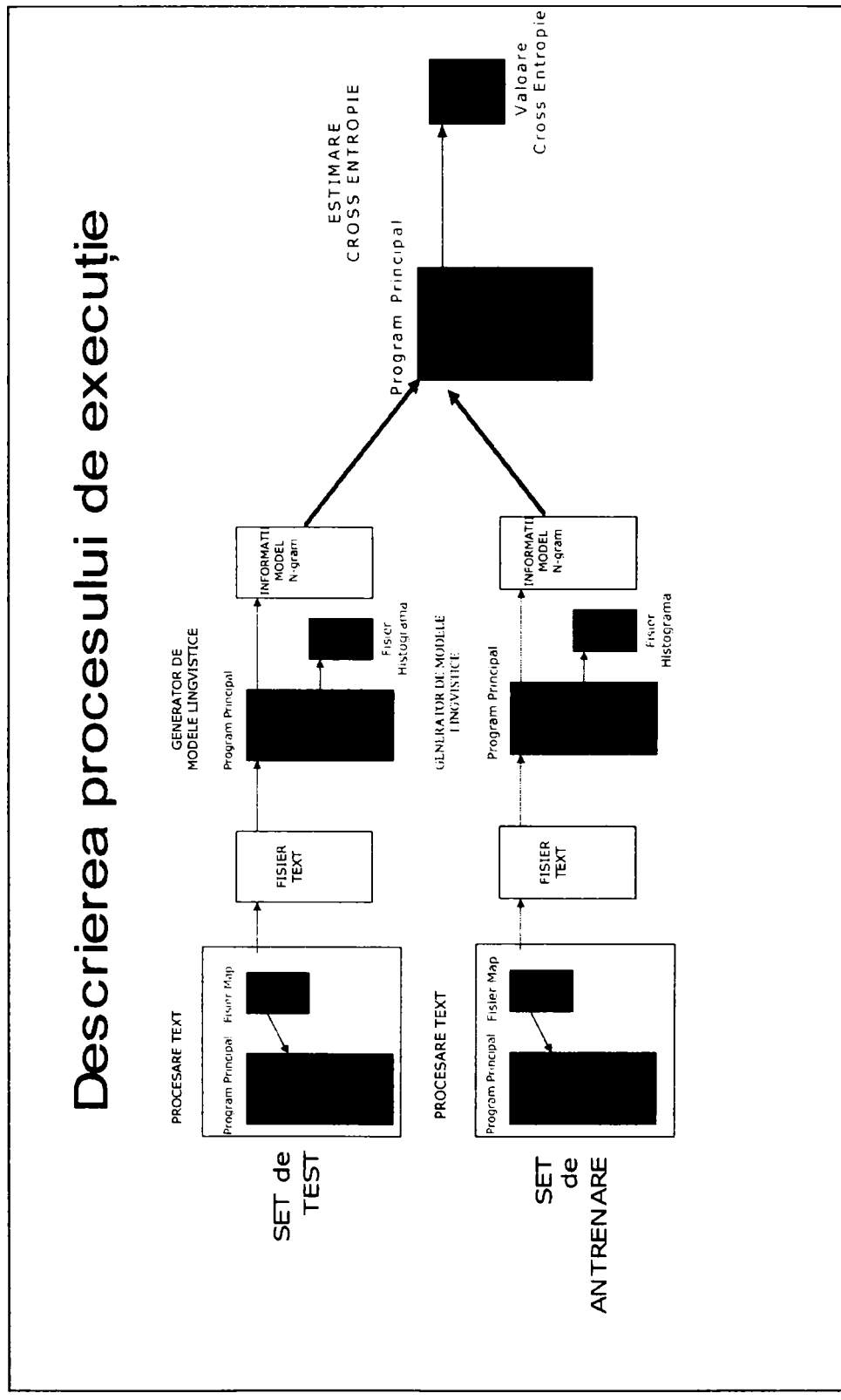
Experimente:

Pentru evaluarea simplă a entropiei ale cărei valori sunt reprezentate în Figura 4.2.4-1 este evident faptul: cu cât este mai mare numărul de caractere consecutive cunoscute cu atât este mai uşor de prezis următorul caracter. În cazul de faţă, simpla entropie atinge valoarea 0 după aproximativ 11 caractere consecutive. Întrebarea care se ridică acum este în ce măsură această estimare este bună sau nu. Aceasta implică o altă întrebare şi anume: "când entropia atinge valori mici?". Răspunsul imediat ar fi: atunci când corpusul de antrenare nu acoperă sau acoperă doar parţial setul de testare şi acest lucru poate fi observat analizând numărul de evenimente care nu se găsesc în setul de antrenare. Valorile pot fi observate în coloana a 4-a a

Tabel 4.2.4-1. Aceste evenimente *n*-gram participă la entropia finală cu valoarea 0. Un alt răspuns ar veni din situaţii când textele au subiecte tematice comune -setul de testare şi de antrenare aparţin unui domeniu foarte specific- caz în care sunt mult mai puţine variaţii în propoziţii decat în propoziţi la modul general (Ex: rapoartele radiologice).

Reprezentarea procesului 3




Descrierea procesului de execuţie
SET de TEST
PROCESARE TEXT
Program Principal
Fisier Map
FISIER TEXT
GENERATOR DE MODELE LINGVISTICE
Program Principal
Fisier Histograma
INFORMATII MODEL N-gram
SET de ANTRENARE
PROCESARE TEXT
Program Principal
Fisier Map
FISIER TEXT
GENERATOR DE MODELE LINGVISTICE
Program Principal
Fisier Histograma
INFORMATII MODEL N-gram
ESTIMARE CROSS ENTROPIE
Program Principal
Valoare Cross Entropie

Căutând raspuns la prima întrebare şi anume: în ce măsură şi dacă estimarea este bună sau nu, sunt realizate cateva investigaţii. Un indicator poate fi valoarea lui *n* din modelul *n*-gram. Acesta ar însemna că dacă *n* este un număr mare şi valoara entropiei tinde să fie foarte mică este posibil ca evaluarea sa fie acceptată. Această afirmaţie este susţinută de interpretarea generală a valorii entropiei/perplexităţii. Dar ca şi măsură de comparaţie se propune valoarea entropiei obţinută când setul de testare este acoperit în mod strict. Cu alte cuvinte, aceasta înseamnă valoarea entropiei când setul de testare şi cel de antrenare coincid. Valorile obţinute sunt prezentate în Tabel 4.2.4-2 iar reprezentarea grafică a acestor valori este realizată în Figura 4.2.4-2. Intenţia este de a urmări cum se comportă valorile entropiei obţinute utilizând un set de antrenare care să coincidă cu cel de testare în comparaţie cu situaţia în care cele două seturi sunt diferite. Ceea ce se aşteptă este faptul că valorile entropiei obţinute în primul caz ar trebui să fie un reper pentru evaluarea entropiei unui model. După cum o arată graficul din Figura 4.2.4-2, simpla entropie se comportă mai bine decât în cazul strictei acoperiri. Diferenţa dintre cele două evoluţii provine de la conţinutul setului de antrenare şi evenimentele *n*-gram purtătoare de valoare 0 în evaluarea entropiei. Până la nivelul 6 al modelului, cele două estimări evoluează aproape similar. După acesta însă diferenţa este vizibilă. O presupunere pentru secţiunile cu evoulţie similară este faptul că evenimentele non-participante nu influenţează într-un mod semnificativ evaluarea finală sau că distribuţia evenimentelor *n*-gram în setul de antrenare este foarte similară cu acelora din setul de testare. Pentru valorile de la 6 la 15 pentru model, diferenţa provine din valoarea raţiei diferite a probabilitaţilor condiţionate din cele două seturi. După nivelul 15 însă diferenţa este din nou nesemnificativă ceea ce sugerează faptul că majoritatea evenimentelor *n*-gram reprezintă expresii regulare care au aceeaşi structură în ambele seturi.

O altă întrebare ar fi în ce măsură influenţează valoarea finală a entropiei evenimentele *n*-gram care nu participă la evaluarea ei. Prin urmare, atenţia este acum direcţionată spre frecvenţa apariţiei acestor evenimente. Spre exemplu, în Figura 4.2.4-3 sunt reprezentate evenimentele *n*-gram cu frecvenţa minimă în setul de testare.

După procesare, setul de testare conţine 626254 unităţi (tokens) şi setul de antrenare 1210573 (aprope dublu). Analizând reprezentarea grafică a frecvenţelor minime se poate observa spre exemplu, că în setul de testare sunt 283 evenimente 2-gram care au loc o singură dată şi reprezintă 0.05% din numărul total de evenimente 2-gram (menţiune: total nu înseamnă evenimente 2-gram unice.) În, pe coloana a patra, sunt prezentate numărul de evenimente care sunt în setul de testare şi nu sunt găsite în cel de antrenare. Pentru 2-gram numărul de evenimente este de 212. Cei doi indicatori, frecvenţa minimă şi evenimentele absente, sunt reprezentate în

Figura 4.2.4-4 şi se poate observa evoluţia lor similară. Oferă acestastă evoluţie a indicatorilor o măsură a termenilor particulari sau expresiilor unice utilizate în text?

Într-adevar, ar putea oferi o referinţă asupra conţinutului setului de testare atâta timp cât indică numărul de evenimente *n*-gram mai putin frecvente în setul de testare alături de cel al evenimentelor *n*-gram inexistente în setul de antrenare. Să fie o simplă coincidenţă este prea puţin posibil dar experimente viitoare vor ajuta la clarificarea acestei observaţii.

n-grams	Nr.of n-grams in TEST set	Nr.of n-grams in TRAIN set	Nr.of events which do not occur in TRAIN set	Entropy	Perplexity
2	615920	1190355	212	3.55936	11.78894
3	610753	1180246	2153	2.91602	7.54766
4	605586	1170138	10397	2.21442	4.64096
5	600419	1160031	31379	1.63220	3.09987
6	595252	1149933	69916	1.19669	2.29214
7	590086	1139854	123185	0.85135	1.80419
8	584921	1129788	184062	0.58090	1.49579
9	579763	1119730	245694	0.38028	1.30160
12	564326	1089635	388032	0.09469	1.06784
15	548928	1059705	454837	0.02116	1.01478
18	533583	1029945	475142	0.00574	1.00399
20	523397	1010234	476943	0.00306	1.00212

Tabel 4.2.4-1. Rezultatele obţinute aplicând procedura de obţinere a entropiei simple

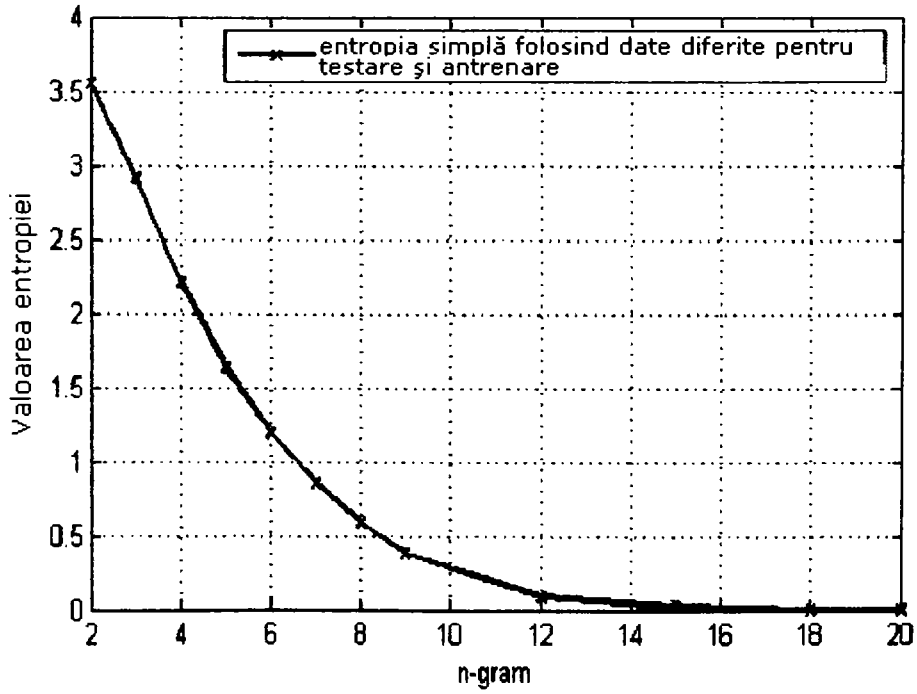


Figura 4.2.4-1. Reprezentarea valorilor pentru estimarea entropiei simple

n-grams	Entropy	Perplexity
2	3.54644	11.68382
3	2.87977	7.36034
4	2.16972	4.49936
5	1.61828	3.07010
6	1.22700	2.34081
7	0.92361	1.89686
8	0.67934	1.60142
9	0.48557	1.40014
12	0.16516	1.12129
15	0.05409	1.03821
18	0.02068	1.01444
20	0.01175	1.00818

Tabel 4.2.4-2. Evaluarea entropiei pentru cazul în care setul de testare şi cel de antrenare coincid (overlapping data)

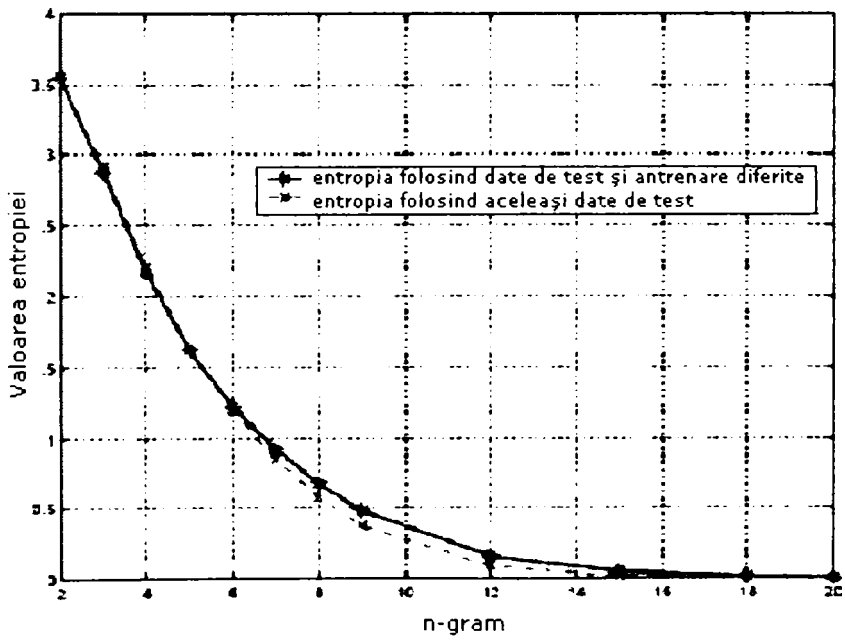


Figura 4.2.4-2. Reprezentarea evoluţiei valorilor entropiei când setul de testare coincide cu setul de antrenare.

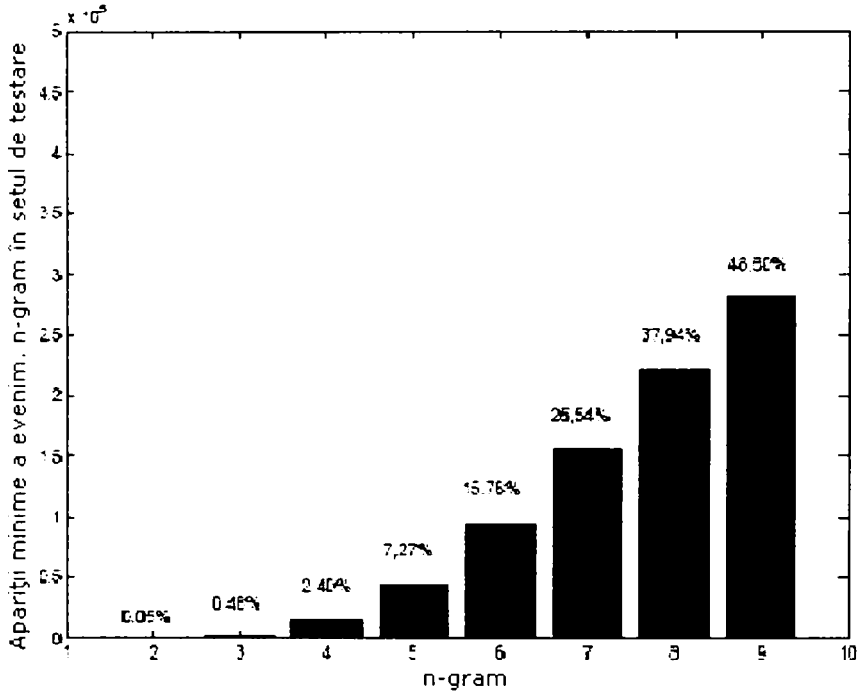


Figura 4.2.4-3. Reprezentarea evenimentelor n-gram cu frecvenţă minimă în setul de testare

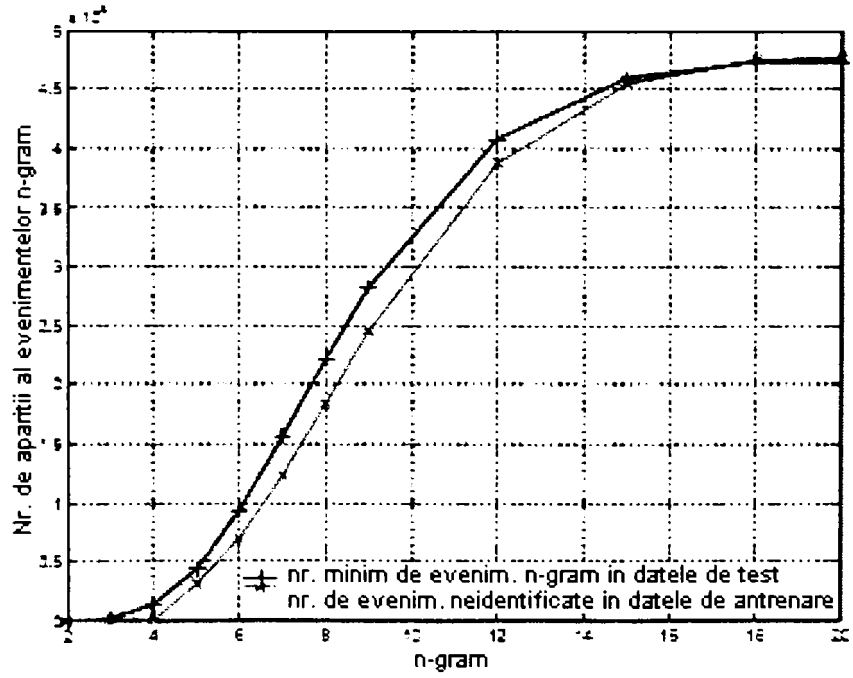


Figura 4.2.4-4. Reprezentarea evoluţiei frecvenţei minime şi a evenimentelor n-gram unice identificate în setul de testare

Experimentele de investigare au continuat şi cu aplicarea unei metode de smoothing în estimarea entropiei iar rezultatele au fost prezentate în [BOGA'03] dar pentru scopul propus în această teză ele sunt mai puţin relevante. Totuşi, efectul aplicării metodei va fi menţionat în secţiunea de concluzii a acestui capitol.

4.3. Concluzii

La finalul acestui capitol se poate concluziona cu faptul că modelele lingvistice statistice se pot identifica drept un domeniu de importanţă majoră în analiza şi expluatarea unei cantităţi mari de informaţie fiind totodată uşor de manipulat şi cu rezultate eficiente în aplicarea lor [BOGA'03c], [BOGA'04a].

Aplicaţia descrisă a fost dezvoltată cu scopul de a oferi posibilitatea investigării într-un mod flexibil a măsurilor aplicate în evaluarea textelor pentru modelarea lingvistică. În ceea ce priveşte rezultatele obţinute vor fi punctate aspectele cele mai importante:

- După cum se menţionează şi în [GIVE'96], acoperirea este un aspect foarte important în recunoaşterea bazată pe modele lingvistice.

- Există întotdeauna o negociere între fliabilitate/robusteţe şi detalii[BOGA'04b]. Aceasta este total dependentă de mărimea setului de antrenare. Cu cât este mai mare setul de antrenare, cu atât se vor obţine rezultate acceptabile pentru valorile mai mari ale lui *n*. Un corpus de antrenare redus va impune o alegere a valorii lui *n* mai mică pentru a fi siguri că un asemenea model poate să existe. În cazul în care gradul de fiabilitate devine o problemă se poate folosi o tehnică de smoothing.

- Metoda de smoothing care nu a mai fost descrisă în acest capitol dar a fost testată în [BOGA'03b] şi a dus la concluzia că "add-one" nu este o alegere bună în evaluarea entropiei modelelor lingvistice statistice pentru sisteme de recunoaştere. Această afirmaţie este susţinută şi în [JURA'00], [GOOD'98]. Alegerea metodei de smoothing optime este dificilă întrucât există o mare diversitate de metode al căror studiu presupune o analiză amplă. Aşa după cum o afirmă şi autorii în [CHEN'98], există o lipsă evidentă a studiilor care să compare în mod sistematic performanţa a mai mult de câteva dintre aceste tehnici pe seturi multiple de date. Prin urmare, din literatura de specialitate, este imposibil de a măsura performanţa relativă a algoritmilor existenţi decât în puţine cazuri particulare. Se poate spune că fiecare algoritm poate funcţiona bine în anumite situaţii. În acest sens este relevantă lucrarea [CHEN'98].

Evaluarea este în mod argumentat cea mai importantă parte al oricărui proiect de cercetare [GIVE'96]. Fără metode adecvate şi măsuri/metrici general acceptate este dificil de evidenţiat vre-un progres. O măsură a acurateţii, care nu este una directă, este perplexitatea. În această evaluare însă au fost utilizate valorile entropiei ca măsură a indeciziei şi predictibilităţii deoarece perplexitatea este

derivată automat din aceasta (a se vedea explicaţiile din definiţia dată în (4.1.3.3-1). Pentru evaluarea performanţei unei model lingvistic ea trebuie realizată asupra aceloraşi date de antrenare deoarece sunt mulţi factori care o pot influenţa. Cu alte cuvinte, entropia/perplexitatea luate în afara contextului nu mai au nici o semnificaţie [GIVE'96]. Investigaţiile făcute asupra acestui set de antrenare au avut ca scop observarea modului diferit în care se manifestă modelele lingvistice şi de a căuta estimatori care să poată oferi repere în evaluarea modelelor lingvistice.
Ca şi subiect principal pentru o viitoare investigaţie este considerat căutarea modului în care poate fi acoperit un text particular, ce fel de parametri sunt implicaţi în evaluarea textelor particulare, cum se comportă modelele lingvistice în textele particulare şi care este acurateţea măsurilor estimate?

5. CONTRIBUŢII LA UTILIZAREA TEHNICILOR DE RECUNOAŞTERE ÎN ANALIZA SECVENŢELOR BIOLOGICE

5.1. Descrierea instanţei de rezolvat

Proteomica se referă la studiul unei colecţii celulare de proteine (în acelşi mod în care genomica se referă la studiul genelor) şi găseşte o largă aplicaţie în bioinformatica actuală. Întrebările tipice care se aplică aproape tuturor genelor sunt următoarele: ce proteină produce fiecare genă, când această proteină este produsă, şi care este rolul său funcţional? În timp ce secvenţa genomică ne poate informa despre ce proteine are potenţial celula de a le produce, şi analiza expresiilor de microvectori poate oferi un răspuns aproximativ despre ce proteine sunt produse, doar abordarea proteomică poate oferi o imagine concretă a biochimiei fundamentale a unei celule. Una dintre abordările proteomice cele mai importante o reprezintă comparaţiile dintre secvenţele de proteine. Necesitatea pentru asemena comparaţii provine din interesul în detectarea omologiilor dintre proteine care pot, la rândul lor, să implice similarităţi structurale şi funcţionale. Aşa după cum a mai fost menţionat în capitolul introductiv, proteinele sunt molecule largi, complexe, compuse din amino acizi şi compararea şi gruparea lor în conformitate cu gradul de similaritate presupune algoritmi specializaţi.

Cele mai frecvent folosite metode au fost deja descrise în capitolele 2 şi 3 şi în general sunt bazate pe proceduri algoritmice complexe pentru aliniament de secvenţe. Desigur că dezvoltarea metodelor de similaritate a cunoscut o dinamică remarcabilă dar, în ciuda maturităţii metodologiilor dezvolatate în această direcţie, derivarea de noi măsuri de similaritate a proteinelor este încă un domeniu de cercetare activ. Interesul este reînoit, datorită creşterii continue a volumului bazelor de secvenţe disponibile care reclamă proceduri algoritmice alternative eficiente din punct de vedere al costurilor şi care pot cuantifica în mod corect similaritatea secvenţelor fără a depinde de un anume tip de aliniament. Separat de eficienţă, o a doua specificare de importanţă egală pentru stabilirea măsurilor de similaritate este evitarea parametrilor care trebuie setaţi de utilizator (o caracteristică inerentă în majoritatea metodologiilor descrise până acum). Acesta este de obicei cazul abordărilor similarităţii în mod clasic, în care utilizatorul întâmpină o mulţime de dificultăţi în alegerea unui algoritm de căutare adecvat, matrice de calcul sau funcţi, precum şi setul de de parametri opţionali ai cărori valori optime să corespundă celei mai bune similarităţi. O varietate de noi metode alternative pentru exprimarea similarităţii dintre secvenţele biologice au devenit disponibile în diverse aplicaţii aşa după cum au fost evidenţiate în secţiunea 2.2.

În capitolul de faţă este introdusă o nouă abordare pentru măsura similarităţii dintre două secvenţe de proteine. Ea a fost inspirată de aplicarea cu succes a conceptului de entropie pentru analiza informaţiei în domeniul modelării statistice a limbajului (Young and Bloothoof [YOUN'97], Manning and Schütze

[MANN'00], Jurafsky and Martin[JURA'00]). Mai concret, modelarea *n*-gram este aplicată fiecărei secvenţe de proteine iar o nouă măsură derivată din conceptul de cross-entropie este apoi folosită pentru compararea perechilor de secvenţe. Cu toate că acest concept de *n*-gram a fost regăsit în lucrări anterioare, e.g. [GANA'04], [ERHA'80], [KARL'91], [KARL'96], munca actuală este de fapt rezultatul primelor încercări de a adopta acest pas dual pentru compararea secvenţelor biologice.

Pornind de la fundamentele teoretice necesare evaluării textului, descrise şi experimentate în capitolul anterior, care au relevat potenţialul măsurilor probabilistice de evaluare a contextului, se va trece spre o nouă propunere de abordare a similarităţii secvenţelor biologice. Metoda se aplică pentru secvenţe de proteine descrise de structura lor primară.

5.2. Propunerea unei noi metode de comparare a proteinelor pe baza evaluării modelelor lingvistice Markov

5.2.1. Noţiuni teoretice utilizate

Există diverse tipuri de modele lingvistice care pot fi folosite pentru a captura diferite aspecte ale regularităţilor limbajului natural [WANG'03]. Lanţurile Markov sunt în general considerate printre cele mai fundamentale concepte pentru construirea modelelor lingvistice. Folosind explicaţiile din capitolul 4 cu privire la definirea şi evaluarea modelelor lingvistice Markov aici se va face o particularizare a lor pentru a susţine modul în care sunt folosite.
După cum este descris în [MANN'00] şi [BROW'92], entropia unei variabile aleatoare *X* care

ia valori într-un domeniu $\aleph$, şi are o probabilitate a funcţiei densitate, *P(X)* este definită ca:

$$H(X) = -\sum_{X\in\aleph} P(X)\log P(X\ ). \qquad (5.2.1\text{-}1)$$

Relativ recent, în lucrarea lui Van Uytsel şi Comparnolle [VANU'98], a fost adoptată ideea generală a entropiei în cazul particular în care o secvenţă scrisă $W=\{w_1,w_2,\ldots,\ w_{k-1},w_k,\ w_{k+1},\ldots\}$ este tratată ca bazată pe compoziţia modelului lingvistic *L*, rezultând următoarea formulă de estimare:

$$\hat{H}_L(X) = -\frac{1}{N}\sum_{W^*} \mathrm{Count}\left(w_i^n\right)\log_2 p_L\left(w_{i+n}\middle|w_i^{n-1}\right), \qquad (5.2.1\text{-}2)$$

unde variabila *X* este de forma unei *n*-gram $X = w_i^n \Leftrightarrow \{w_i,w_{i+1},\ldots,w_{i+n-1}\}$ şi $\mathrm{Count}(w_i^n)$ este numărul apariţiilor lui w_i^n. Suma se execută asupra tuturor combinaţiilor consecutive posibile w_i de lungime *n* (i.e

$W^*=\{\{w_1,w_2,\ldots,w_n\},\{w_2,w_3,\ldots,w_{n+1}\},\ldots.\}$) şi N este numărul total de n-grame din secvenţa investigată. Al doilea termen din suma euaţiei 5.2-2, adică $\log_2 p_L\left(w_{i+n}\middle|w_i^{n-1}\right)$, este *log* din probabilitatea condiţionată care conectează al n-lea element dintr-o n-gram cu cele n-1 elemente precedente. Urmând principiul estimării probabilităţii maxime (MLE), el poate fi estimat folosind frecvenţele relative:

$$\hat{p}\left(w_{i+n}\middle|w_i^{n-1}\right)=\frac{\text{Count}\left(w_{i+n}\right)}{\text{Count}\left(w_i^{n-1}\right)} \quad . \tag{5.2.1-3}$$

Echivalent cu ecuaţia 5.2-2 se poate folosi expresia

$$\hat{H}_L(X)=-\sum_{W^*} p\left(w_{i+n}\right)\log_2 p_L\left(w_{i+n}\middle|w_i^{n-1}\right) \tag{5.2.1-4}$$

unde

$$\hat{p}\left(w_{i+n}\right)=\frac{\text{Count}\left(w_{i+n}\right)}{N} \tag{5.2.1-5}$$

considerată ca frecvenţa relativă a evenimentului n-gram w_i^n.

Cross-entropia dintre probabilitatea de distribuţie actuală *P(X)* (care parcurge o variabilă aleatoare *X*) şi probabilitatea distribuţiei *Q(X)* estimată dintr-un model este definită astfel:

$$H(X,\ Q)\ =\ -\sum_{X\in\aleph} P(X)\log Q\ (X). \tag{5.2.1-5}$$

Aici trebuie menţionate două remarci importante. Prima, cross-entropia unui proces stocastic, măsurată prin folosirea unui model, este o limită superioară pentru entropia procesului (i.e. $H(X)\le H(X,Q)$) [MANN'00], [BROW'92]). A doua, aşa după cum este menţionată în [JURA'00], între două modele, cel mai bun este cel cu valoarea cross-entropiei minimă.

Estimarea entropică de mai sus (luată împreună cu forma generală din ecuaţia 5.2.1-1 şi 5.2.1-2) sugerând o cale directă de trecere de la entropie la formularea cross-entopiei) a fost baza în construirea noii măsuri de similaritate a proteinelor, ce va fi descrisă în continuarie.

5.2.2. Descrierea metodei

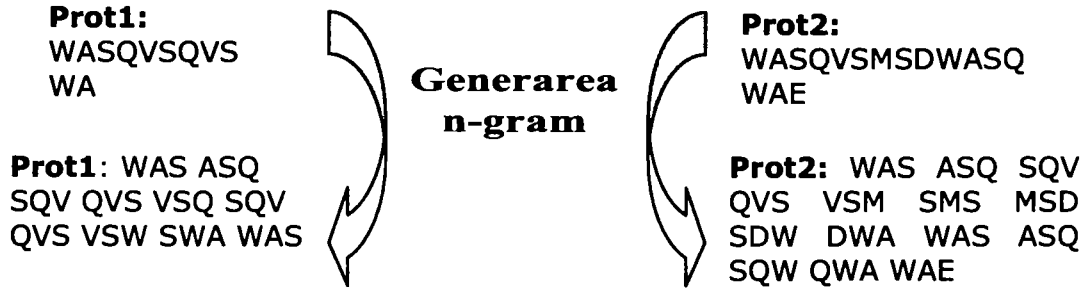
Aşa după cum s-a mai menţionat la începutul acestui capitol, secvenţele de proteine provenind de la diverse organisme pot fi abordate ca text scris într-un limbaj universal în care alfabetul constă din 20 simboluri distincte, amino acizii. Asocierea unei secvenţe de proteine structurii sale, dinamica funcţională şi rolul său biologic devin astfel analoge cu asocierea dintre cuvinte şi înţelesul lor semantic din limbajul natural. S-a sugerat că această analogie poate fi expluatată prin aplicarea modelării lingvistice statistice şi a tehnicilor de clasificare a textelor pentru progresul « înţelegerii » secvenţelor biologice. Oamenii de ştiinţă din acest domeniu hibrid de

cercetare consideră că identificarea regulilor de gramatică/sintaxă pot releva sistematici de o importanţă ridicată pentru ştiinţele biologice şi medicale. Pentru exemplificarea noii abordări alegem o secvenţă de proteine ipotetică WASQVSENR. În modelarea *2*-gram rezultată avem urmatoarele elemente (tokens/cuvinte) {WA AS SQ QV VS SE EN NR}, în timp ce în reprezentarea *3*-gram token-ii identificaţi sunt {WAS ASQ SQV QVS VSE SEN ENR}. Pe baza frecvenţei de apariţie a acestor « cuvinte » (determinată prin numărare) şi prin formarea unor rapoarte de frecvenţă corespunzătoare, poate fi estimată cu uşurinţă entropia unui model *n*-gram folosind (5.2-2). Această măsură este indicativă pentru a determina cât de bine o secvenţă specifică de proteine este reprezentată de către modelul *n*-gram corespunzător. Dacă această măsură ar putea fi aplicată pe două proteine distincte (şi ne ajută să decidem care proteină este cel mai bine reprezentată de model), ieşirile nu pot fi folosite pentru compararea directă a celor două proteine. Acest neajuns a direcţionat spre derivarea cross-entropiei corespunzătoare, în care modelul *n*-gram este mai întâi construit pe baza calculării « cuvintelor » unei secvenţe de proteine *Y* şi apoi folosit pentru secvenţa *X*, contrastând cele două secvenţe. Astfel, conţinutul de informaţie comună dintre două proteine *X*, *Y* este exprimat via formula :

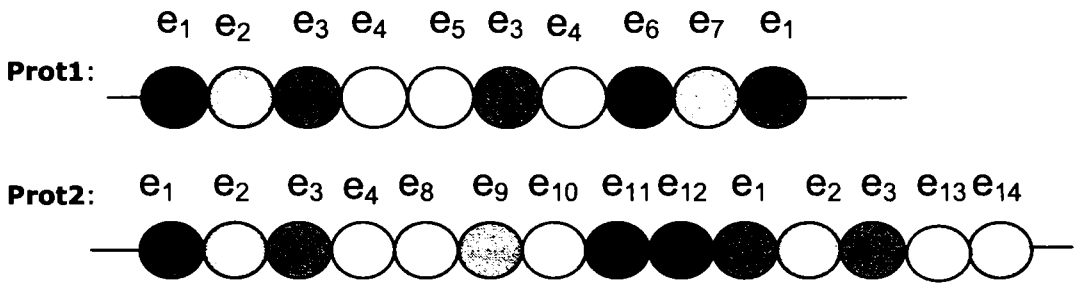
$$E(X,Y) = -\sum_{\text{all } w_i^n} P_X(w_i^n)\log P_Y\left(w_{i+n} \mid w_i^{n-1}\right). \qquad (5.2.2\text{-}1)$$

Primul termen $P_X\left(w_i^n\right)$ din (5.3-1) corespunde secvenţei de proteine *X* (şi rezultă din calcularea frecvenţei cuvintelor acestei secvenţe specifice). Al doilea termen se referă la secvenţa *Y* pe baza căreia modelul trebuie estimat (de asemenea rezultatele sale vor veni din calcularea frecvenţei de apariţie acuvintelor componente). Variabila w_i^n va parcurge toate cuvintele (reprezentate de *n*-grame) ale secveţei de proteine considerată referinţă. Pentru o înţelegere mai facilă se va exemplifica folosind două secvenţe aleatoare de proteine.

Procesarea secvenţelor de proteine.



Pentru fiecare secvenţă descompusă în "cuvinte" *n*-gram, numite şi evenimente, se reprezintă figurativ succesiunea acestora. Astfel, fiecărui eveniment unic îi corespnde o culoare.



Folosind informaţiile organizate sub formă de evenimente, pentru fiecare secvenţă este păstrat un tablou de date care conţine informaţiile privitoare la frecvenţa de apariţie a fiecărui eveniment sub următoarea formă:

Prot1:	Eveniment	Frecvenţe relative din (1)	Probabilităţi condiţionate din (2)
WAS	e1	2/10	2/2
ASQ	e2	1/10	1/1
SQV	e3	2/10	2/2
QVS	e4	2/10	2/2
VSQ	e5	1/10	1/2
VSW	e6	1/10	1/2
SWA	e7	1/10	1/1

Prot2:	Eveniment	Frecvenţe relative din (1)	Probabilităţi condiţionate din (2)
WAS	e1	2/14	2/3
ASQ	e2	2/14	2/2
SQV	e3	2/14	2/2
QVS	e4	1/14	2/2
VSM	e8	1/14	1/1
SMS	e9	1/14	1/1
MSD	e10	1/14	1/1
SDW	e11	1/14	1/1
DWA	e12	1/14	1/1
QWA	e13	1/14	1/2
WAE	e14	1/14	1/3

Odată introdusă noua măsură de similaritate, se va trece la descrierea modului său de folosire pentru a efectua căutări în baza de date de proteine. Punctul esenţial al acestei abordări constă în faptul că atât proteina necunoscută (ex. o nouă proteină descoperită) aflată în postura de query cât şi fiecare proteină din baza de secvenţe sunt reprezentate via codificare *n*-gram şi similaritatea introdusă anterior este folosită pentru a le compara sub această formă de reprezentare (descompuse în evenimente *n*-gram).

5.2.3. Descrierea modul de aplicare al noii măsuri de similaritate

Au fost identificate două moduri în care similaritatea bazată pe *n*-grame este angajată în căutarea eficientă într-o bază de date de secvenţe [BOGA'05a]. Cea mai directă implementare este denumită în mod justificat: "***metoda directă***". Un al doilea algoritm, "***metoda alternantă***", a fost identificat în scopul medierii cazurilor în care proteinele care se compară pot fi de lungimi foarte diferite. Este uşor de observat implicarea acestui aspect în valoarea raportului dintre numărul de cuvinte din secvenţa de referinţă implicată în calcularea valorii similarităţii şi numărul total de cuvinte din respectiva secvenţă (implicat în prima probabilitate din ecuaţia 5.3-1). Modul de experimentare cu ambele metode şi de a compara performanţa lor oferă posibilitatea de a verifica senzitivitatea măsurii propuse privind lungimea secvenţelor.

Metoda directă: Fie S_q secvenţa de proteină query şi $\{S\}=\{S_1, S_2, \ldots S_N\}$ baza de date de proteine. Primul pas este calcularea scorului 'perfect' (PS) sau scor 'referinţă' pentru proteina query. Aceasta este realizată prin calcularea $E(S_q,S_q)$ folosind secvenţa de query atât ca referinţă cât şi ca model în ecuaţia 5.2.2-1. În al doilea pas, fiecare proteină S_i, i=1...N, din baza de date serveşte ca secvenţă model în calcularea unui scor de similaritate $E(S_q,S_i)$, folosind aceeaşi ecuaţie 5.3-1, cu proteina query servind ca secvenţă de referinţă. În acest mod, sunt calculate N similarităţi $E(S_q,S_i)$, i=1,..,N. În final, aceste similarităţi sunt comparate cu scorul perfect PS. Calculând diferenţele absolute $D(S_q,S_i)=|E(S_q,S_i)-PS|$, sunt exprimate 'discrepanţele' în sensul conţinutului informaţiei dintre proteina query şi proteinele din baza de date. Ordonând aceste N diferenţe putem uşor identifica cele mai similare proteine cu proteina query ca fiind acelea cu cea mai mica valoare a diferenţei $D(S_q,S_i)$.

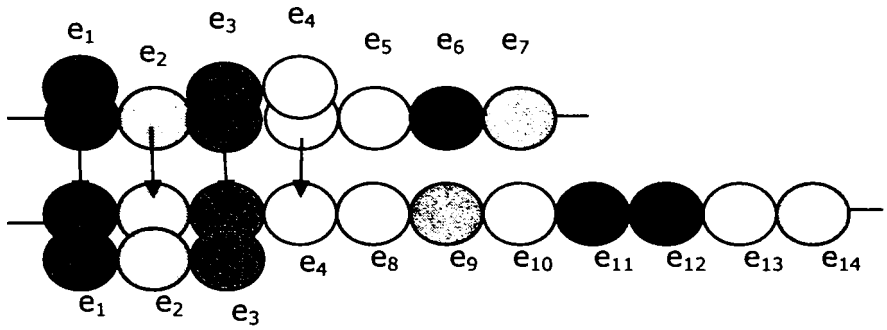
Reprezentarea metodologiei pentru *metoda directă*:

Pas1. Calcularea scorului de referinţă: Calculează pentru fiecare secvenţă scorul său "perfect" folosind (5.2-1) unde P este exprimat din prima şi a două coloană a aceluiaşi tabel

$$S_{Prot1/Prdt}=-(\frac{2}{10}\log_2\frac{2}{2}+\frac{1}{10}\log_2\frac{1}{1}+\frac{2}{10}\log_2\frac{2}{2}+\frac{2}{10}\log_2\frac{2}{2}+\frac{1}{10}\log_2\frac{1}{2}+\frac{1}{10}\log_2\frac{1}{2}+\frac{1}{10}\log_2\frac{1}{1})=$$
$$=-(0+0+0+0+0-0.1-0.1+0)=-0.2$$

Pas2. Calculează scorul Prot1/Prot2 folosind (5.2.2-1) cu P_X exprimat în prima coloană a Prot1 şi P_Y din a doua coloană a Prot2.

$$S_{Prot1/Prot2}=-(\frac{2}{10}\log_2\frac{2}{3}+\frac{1}{10}\log_2\frac{2}{2}+\frac{2}{10}\log_2\frac{2}{2}+\frac{2}{10}\log_2\frac{1}{1})=-[0.2(1-1.58)]=0.116$$



Pas3. Calcularea similarităţii pentru *metoda directă* :

$$Dist(Prot1, Prot2)=|S_{Prot1/Prot1}- S_{Prot1/Prot2}|=|0.2-0.116|=0.084$$

Metoda alternantă: Singura diferenţă faţă de metoda directă este la pasul doi, atunci când se compară secvenţa query cu fiecare secvenţă din baza de date. Rolul de secvenţă referinţă şi model se poate interschimba în funcţie de care are lungimea cea mai mică (secvenţa cea mai scurtă joacă rolul de referinţă în (5.3-1)). Restul paşilor, estimarea scorului perfect, ordonarea şi selectarea urmează aceeşi procedură ca şi în cazul metodei directe.

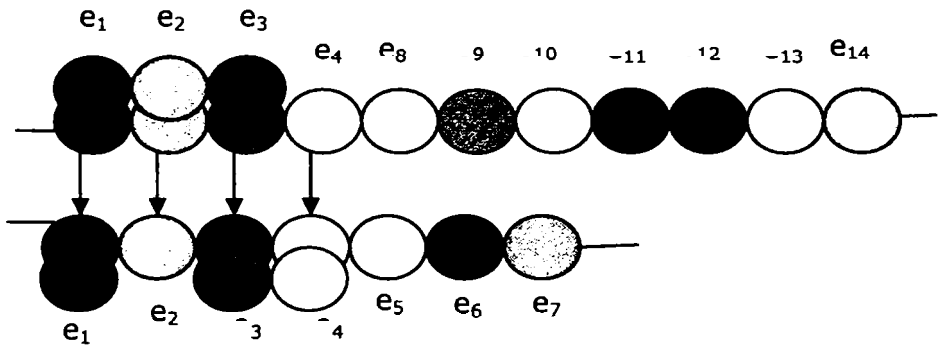
Reprezentarea metodologiei pentru *metoda alternantă*:

Pas1. Calcularea scorului de referinţă: Calculează pentru fiecare secvenţă scorul său "perfect" folosind (5.2-1) unde P este exprimat din prima şi a două coloană a aceluiaşi tabel

$$S_{Prot1/Prdt}=-(\frac{2}{10}\log_2\frac{2}{2}+\frac{1}{10}\log_2\frac{1}{1}+\frac{2}{10}\log_2\frac{2}{2}+\frac{2}{10}\log_2\frac{2}{2}+\frac{1}{10}\log_2\frac{1}{2}+\frac{1}{10}\log_2\frac{1}{2}+\frac{1}{10}\log_2\frac{1}{1})=$$
$$=-(0+0+0+0+0-0.1-0.1+0)=-0.2$$

Pas2. Calculează scorul Prot1/Prot2 folosind (5.3-1) cu P_X exprimat în prima coloană a Prot2 şi P_Y din a doua coloană a Prot1.

$$S_{Prot2/Prot1}=-(\frac{2}{14}\log_2\frac{2}{2}+\frac{2}{14}\log_2\frac{1}{1}+\frac{2}{14}\log_2\frac{2}{2}+\frac{1}{14}\log_2\frac{2}{2})=-(0+0+0+0)=0$$



Pas3. Calcularea similarităţii pentru *metoda alternantă* :

$$Dist(Prot1, Prot2)=|S_{Prot1/Prot1}- S_{Prot2/Prot1}|=|0.2-0|=0.2 .$$

Descrierea algoritmică a celor două metode

***Metoda directă*:**

Fie setul ***S*** de proteine, P_X este întotdeauna estimat ca frecvenţa relativă urmând ecuaţia 5.2.1-5 şi P_Y este estimat folosind ecuaţia 5.2.1-3.

Repetă

*Pentru fiecare proteină din set, obţine scorul referinţă(perfect) aplicand ecuaţia 5.2.2-1 în modul următor:

X = secvenţa de proteină curenta (query);

w_i^n parcurge toate evenimentele n-gram din X;

P_X este estimat din X;

P_M este estimat de asemenea din acelaşi X ;

RS = rezultat;

Repetă

Pentru fiecare proteină din mulţimea S-X, aplică ecuaţia 5.3-1 unde:

X = secvenţa de proteină curenta (query);

Y= secvenţa de proteină curentă din mulţimea S-X;

X_n parcurge toate elementele n-gram din X;

P_X este estimat din X;

P_Y este estimat din Y;

Res = rezultat;

Score = abs(RS-Res);

colectează toate scorurile obţinute pentru fiecare secvenţă X considerată query şi restul din setul de date;

până când mulţimea S-X este vidă;

până când mulţimea S este vidă.

Metoda alternantă:

Fie setul S de proteine. P_X este întotdeauna estimat urmând ecuaţia 5.2.1-5 şi P_Y este estimat folosind ecuaţia 5.2.1-3.

Repetă

*Pentru fiecare proteină din set, obţine scorul referinţă(perfect) aplicand ecuaţia 5.2.2-1 în modul următor:

X secvenţa de proteină curenta (query);

w_i^n parcurge toate n-gram din X;

P_X este estimat din X;

P_Y este estimat de asemenea din acelaşi X ;

RS :=abs(rezultat);

Repetă

**Pentru* fiecare proteină din mulţimea S-X, aplică ecuaţia 5.2.2-1 unde:

X secvenţa de proteină curenta (query);

Y secvenţa de proteină curentă din mulţimea S-X;

**Dacă* lungimea proteinei X este mai mare decât lungimea lui Y

atunci

X_n parcurge toate elementele n-gram din Y;

P_X este estimat din Y;

P_Y este estimat din X;

Res := abs(rezultat);

Score := abs(RS-Res);

**colectează* toate scorurile obţinute pentru fiecare secvenţă X considerată query şi restul din setul de date;

până când mulţimea S-X este vidă;

până când mulţimea S este vidă.

În ambele cazuri, variabla *Score* reprezintă distanţa dintre două numere reale cunoscută ca dist(a, b)=|a-b|, unde a, b∈ **R** şi |x| este valoarea absolută a lui x.

5.2.4. Implementare şi detalii de performanţă ale noii metode

Urmând metodologia deja descrisă, implementarea propriu-zisă respectă parcurgerea acestor etape. Astfel, utilizând limbajul de programare Perl s-au construit programe specializate care să execute operaţiile de: construire a modelelor *n*-gram, stocare a informaţiei conţinute de fiecare secvenţă precum şi de evaluare şi estimare a disimilarităţii/similarităţii dintre secvenţele comparate. Ca şi în cazul programelor care constituie setul destinat pentru analiza lingvistică descris în capitolul anterior, şi aceste programe sunt independente din punct de vedere al implementării dar dependente funcţional. Ele se constituie ca module de scripturi

lansate în execuţie din linia de comandă, care pot fi rulate atat pe platforma oferită de sistemul de operare Windows cât şi Unix/Linux, beneficiind de un grad înalt de portabilitate.

Conform studiilor din ingineria proceselor software, în general produsele soft pot fi evaluate fie în mod direct fie indirect. Prin evaluarea directă se înţelege determinarea costurilor şi a eforturilor asociate. Ea presupune calculul numărului liniilor de cod (LOC - lines of code) scrise, determinarea vitezei de execuţie, a dimensiunii memoriei, precum şi a numărului de defecte raportat într-un anumit interval de timp [FLOR'96]. Evaluarea indirectă a produsului reprezintă în fapt o analiză a funcţionalităţii, calităţii, complexităţii, eficienţei, fiabilităţii, întreţinerii şi multor altor caracteristici. Pentru implementarea algoritmilor introduşi în acest capitol însă, atenţia nu a fost orientată în mod special pe aspectele de implementare cât pe testarea performanţei în ce priveşte acurateţea rezultatelor generate. Aceasta, deoarece este evidentă simplitatea şi avantajul utilizării noii metode şi a resurselor informaţionale folosite. În plus, implementarea metodei s-a dorit a fi realizată cât mai modular şi execuţia este secvenţială, pentru a putea accesa şi modifica uşor orice porţiune care ar necesita intervenţii pe parcursul execuţiei sau în vederea unei alternative la ideea de bază. Toate acestea fac din setul de programe implementat un instrument experimental uşor de manevrat.

Prin urmare, o analiză a complexităţii în această situaţie se poate realiza folosind noţiunea de complexitate algoritmică. Este foarte convenabilă clasificarea algoritmilor pe baza cantităţii relative de spaţiu necesar şi specificarea creşterii cerinţelor de timp/spaţiu ca o funcţie dependentă de mărimea datelor de intrare.
Astfel există noţiunea de complexitate a timpului, care reprezintă timpul de execuţie al programului ca funcţie dependentă de mărimea datelor de intrare şi noţiunea de complexitate a spaţiului, semnificând cantitatea de memorie necesară în timpul execuţiei programului, tot în funcţie de mărimea datelor de intrare.

Cea mai comună metrică pentru calcularea complexităţii timpului şi spaţiului este notaţia „O" (o cu majusculă). Aceasta înlătură toţii factorii constanţi astfel încât timpul de execuţie poate fi estimat în relaţie cu N, pentru N reprezentând o valore care poate tinde spre infinit. Deoarece O(*expresie*) reprezintă toate funcţiile care cresc mai încet sau egal cu *expresie* , în general el este este folosit pentru a exprima limitele superioare.

În cazul de faţă, s-a adoptat ca măsură a complexităţii numarul de paşi pe care îi execută algoritmul astfel: dacă expresia numărului de paşi este polinomială, se va considera termenul dominant neglijându-se coeficientul numeric. Analog şi la expresie exponentială. Adică: dacă expresia este numarul de paşi $N=2*n^3*e^n+7*n+1$ (unde *n* este numarul de elemente şi *e* este baza logaritmilor), atunci complexitatea $C=n^3*e^n$ sau $O(n^3*e^n)$ se doreşte a fi o complexitate cât mai mică [AHO'85].

Având în vedere faptul că pentru obţinerea similarităţii este necesară o procesare prealabilă a secvenţelor şi stocarea de informaţie relevantă, vor fi aproximate complexităţile separat pentru fiecare etapă. Astfel, procesarea unei secvenţe presupune o complexitate de timp şi spaţiu de O(N), unde N este lungimea secvenţei de intrare minus (*n*-1), *n* fiind ordinul modelului. Pentru evaluarea propriu-zisă a similarităţii însă se aproximează o complexitate de timp O(N×M), pentru ambele abordări (metoda directă şi alternantă), unde N şi M sunt lungimile datelor de intrare (organizate ca succesiuni de evenimente *n*-gram). În ce priveşte

complexitatea spaţiului stocat pentru *metoda directă* se estimează o complexitate O(N), cu N lungimea secvenţei procesate cu rol de query, iar pentru *metoda alternantă* complexitatea spaţiului poate fi aproximată de O(min{N,M}). Aceste expresii, care exprimă o complexitate relativ redusă se datorează facilităţii de lucru cu vectori de asociere, oferită de limbajul Perl (Practical Extraction and Report Language).

5.3. Identificarea secvenţelor mutante ale unei proteine. Experimente.

Progresul rapid în cercetare legat de bazele de date biologice care conţin secvenţe genomice ale diferitelor organisme deschid provocarea de a decifera "misterul vieţii". Cu scopul deserviri acestui obiectiv unele dintre cele mai importante interese se pot folosi de tehnicile de data mining (DM). *Data Mining* este definită în general ca un ansamblu de diferite tehnici utilizate pentru a extrage informaţie ascunsă, predictivă din baze de date multiple şi de dimensiuni mari stocate sub diverse forme [BOGA'06a]. Aceste tehnici sunt orientate spre descoperirea de cunoştinţe şi merg dincolo de simpla caracterizare statistică a datelor disponibile. Secvenţele biologice pot fi un domeniu foarte important al DM, fiind capabile să deservească unele dintre cele mai importante arii de interes ale analizei secvenţelor biologice. Noua strategie propusă pentru determinarea similarităţii dintre secvenţele biologice este testată ăn aceste experimente ca metodă de DM în scopul identificării secvenţelor mutante ale secvenţei de căutare [BOGA'06a].

5.3.1. Baza de secvenţe

Strategia propusă pentru măsurarea similarităţii proteinelor a fost demonstrată şi validată folosind o bază de date ce conţine în total 100 de secvenţe de proteine [BOGA'05b]. Au fost formate două grupuri distincte de proteine după cum urmează. Primele 50 de intrări din baza de date corespund proteinelor selectate aleator din baza de proteine publică NCBI [NCBI'05]. Ultimele 50 de intrări corespund proteinelor rezultate ca urmare a unor mutaţii diverse ale genei responsabile pentru producerea de cancer, **p53**. Mutaţile au fost selectate aleator din baza de secvenţe creată utilizând descrierea oferită de International Agency for Research on Cancer (IARC) Lyon, Franţa[33]. Baza de date IARC folosită este sub formă de fişier în format Excel şi conţine 18 585 întrări organizate în 42 coloane. Fiecare linie (coloană) reprezintă o singură mutaţie căreia îi este asignat un singur număr unic de identificare. Un număr unic de identificare este de asemenea atribuit mostrei de tumoră şi pacientului. Din fiecare înregistrare se folosesc câmpurile care definesc acea mutaţie:

- numărul de identificare unic al mutaţiei;
- locaţia mutaţiei în intron sau exon[34] în gena p53 ;

[33] http://www.iarc.fr/p53/Somatic.html

[34] **Exonii** sunt regiuni ale AND-ului dintr-o genă care nu sunt înlăturate prin transcrierea ARN şi sunt păstrate în molecula finală mesager de ARN(mRNA)[WIKI'05g].

- pentru mutaţii în exon, numărul codon-ului la care este locată mutaţia (1-393);
- poziţia nucleotidelor mutaţiei pe baza intrării în bănca genetică, X54156;
- valorile de Yes (True) sau No (False) pentru a indica dacă poziţia mutaţiei cade într-o secvenţă CpG (Cytosine-phosphor-Guanine),
- valorile Yes (True) sau No (False) pentru a indica dacă poziţia mutaţiei cade într-o locaţie de conexiune ("splice site");
- natura mutaţiei;
- pentru mutaţii în exoni, secvenţa de bază a codon-ului în care are loc mutaţia;
- secvenţa bazei mutate;
- tipul de amino acid (wild-type amino acid) codificat în codon-ul în care are loc mutaţia (abreviere din trei litere a amino acidului);
- amino acidul mutat codificat în codon-ul în care a avut loc mutaţia (abreviaţia de trei litere a amino acidului) şi
- efectul mutaţiei şi al numărului de codon la care poate apare un codon de STOP.

Din baza de date creată au fost luate aleator 50 de mutaţii care produc *missense mutation* (când modificarea nucleotidelor produce modificare în codon, atunci amino acidul se modifică şi în final şi proteina), *nonsense mutations* (când modificarea în nucleotide determină modificarea codonului în TAA, TAG sau TGA – care sunt codon de STOP şi prin urmare nu se va produce nici un amino acid, rezultând într-o proteină mai scurtă) şi *silent mutations* (când schimbarea în nucleotide determină modificarea codon-ului dar nu se modifică nici amino acizii şi prin urmare nici proteina).

Acest set de 50 de proteine, numit în acest context grupul p53, se aşteaptă să formeze un grup compact de forme textuale în spaţiul semnificaţiilor biologice. În mod contrar, restul de 50 de proteine trebuie să apară ca forme textuale în acelşi spaţiu care nu numai să difere de un altul dar cu precădere de grupul p53. Setul complet de secvenţe poate fi consultat în ANEXA 4.

5.3.2. Descrierea experimentelor. Rezultate.

Cu scopul de a ilustra cele două variante ale strategiei propuse, în primul rând au fost urmaţi nişte paşi clasici ai analizei exploratorii a datelor (*Exploratory Data Analysis*). Matricea conţinând toate măsurile de *disimilaritate* posibile obţinute $D(S_i,S_j)$, $i,j=1,2,...N$ pentru secvenţele date este ilustrată în Figura 5.3.2-1, ca imagine la scara gri, pentru ambele variante algoritmice ale noii metode şi trei modele *n*-gram diferite. În schema de vizualizare adoptată toate matricele prezentate (după o normalizare adecvată), partajează o scală comună în care 1(alb) corespunde distanţei maxime în fiecare matrice. Merită menţionat faptul că reprezentarea spaţială 'ideală' în acest caz este o matrice albă cu un singur segment negru în colţul din dreapta jos. Este prin urmare evident din Figura 5.3.2-1 faptul că modelarea *4*-gram urmată de versiunea 'alternantă' a noului algoritm atinge o performanţă aproape excelentă în căutarea în baza de secvenţe date.

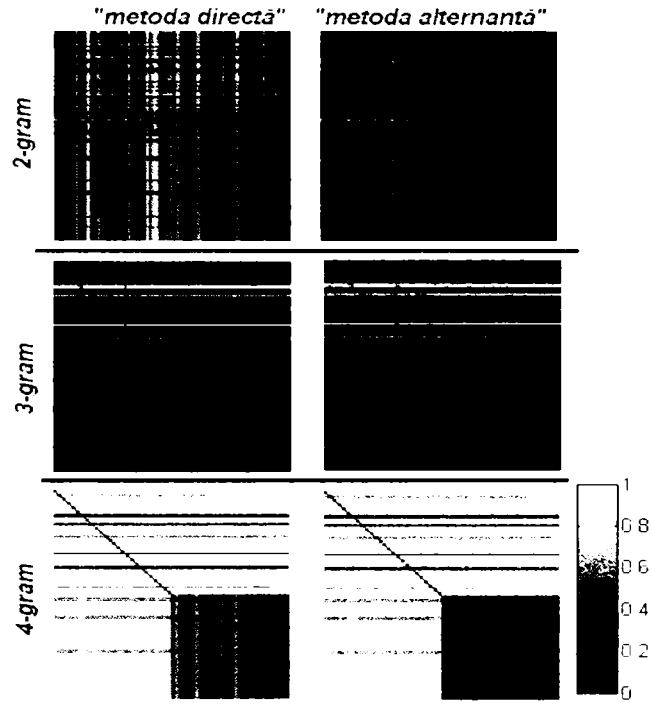


Figura 5.3.2-1. Vizualizarea matricelor conţinând disimilartăţile tuturor perechilor determinate de cele 100 de proteine din baza de date utilizată.

Datorită separării evidente a secvenţelor în baza de date s-a trecut la determinarea apartenenţei secvenţelor vizualizate ca strâns grupate în colţul negru dreapta-jos al Figura 5.3.2-1. Astfel, cu scopul inverstigării lor, rezultatele obţinute în urma aplicării metodei alternante pentru modelul 4-gram sunt proiectate într-un spaţiu dimensional redus, reprezentat în Figura 5.3.2-2. Proteinele necunoscute sunt marcate cu albastru(formele definite de contur) iar cele provenind din mutaţiile genei p53 în culoarea roşu(formele reprezentate cu continut hasurat). Este evident faptul că în acest caz s-a obţinut o foarte bună soluţie la problema identificării/separării a două clase.

Menţiune: Metoda de reprezentare vizuală a matricelor de disimilaritate îi aparţine lui Dr. Nikos Laskaris, Universitatea Aristotel din Thessaloniki, Grecia, cu al cărui consimţământ şi îndrumare a fost folosită.

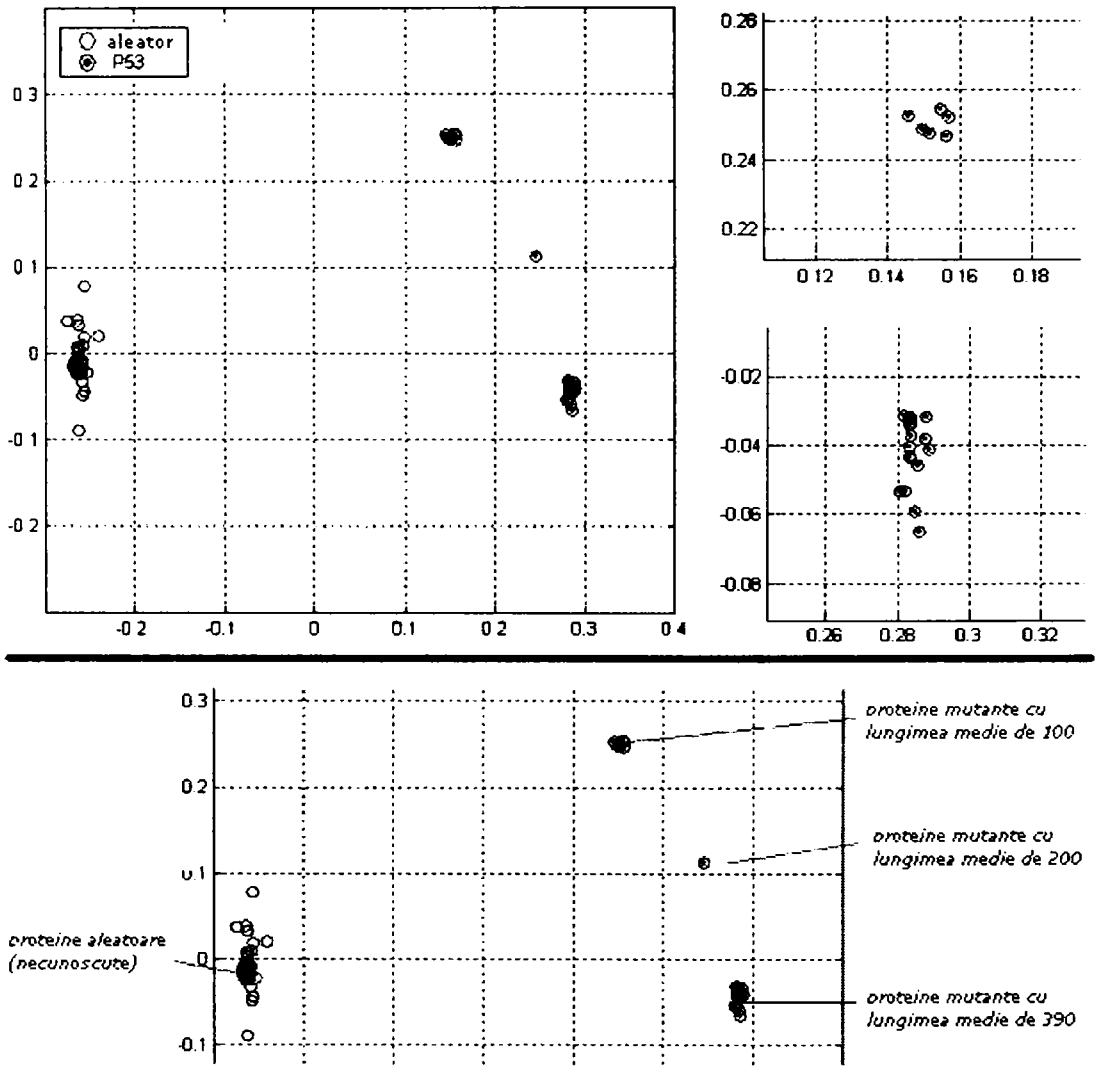


Figura 5.3.2-2. Reprezentarea în spaţiu dimensional redus a secvenţelor de proteine folosind măsura disimilarităţii pentru setul experimental de 100 secvenţe

În al doilea rând, cu scopul furnizării unei măsuri cantitative a performanţei celor două variante, s-a adoptat un index al acurateţii care este derivat din metoda ROC (Receiver Operating Characteristic) şi care a câştigat popularitate în validarea rezultatelor obţinute din căutarea în baze de date (Liao and Noble [LIAO'03], Schäffer et al [SCHÄ'01]). Acest index, de obicei referit ca scor ROC trunchiat, este raportul dintre suprafaţa aflată sub curba ROC (în proiectarea valorilor adevarate şi pozitive versus false şi pozitive, pentru diferite praguri de disimilaritate). Mai explicit, după cum este menţionat şi în [SCHÄ'01]), pentru un număr T de cazuri adevărate şi pozitive posibil de a fi găsite şi un număr fix de cazuri false şi pozitive n, acest index este proporţia dată de dreptunghiul [0, T] [0, n] care se află sub curba de senzitivitate. El ia valori în intervalul [0-1], cu 1 corespunzând celei mai

ridicate performanţe. Acest scor ROC a fost introdus în Tabel 5.3.2-1 pentru diferite modele *n*-gram şi ambele versiuni ale metodei iar în Figura 5.3.2-3 este o reprezentare grafică a traiectoriei urmate de fiecare metodă în circumstanţele impuse de modelul *n*-gram folosit şi diverse valori prag pentru evaluare.

Tabel 5.3.2-1 Scorul ROC obţinut pentru ambele metode pentru diferite valori prag.

model *n*-gram	**Aria normalizata aflata sub curba ROC**	
	Metoda Directă	***Metoda Alternantă***
2-gram	0.589	0.680
3-gram	0.723	0.817
4-gram	0.900	0.978

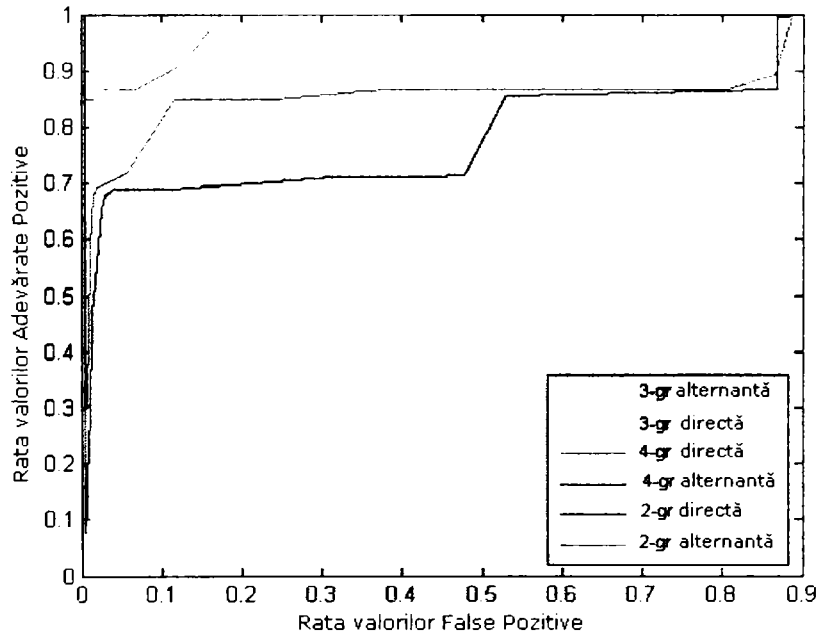


Figura 5.3.2-3. Reprezentarea evoluţiei valorilor raporturilor adevărate şi pozitive versus false şi pozitive

5.3.3. Analiza şi interpretarea rezultatelor

Metoda prezentată constituie un prim pas în investigarea implicării modelării lingvistice în caracterizarea, manevrarea şi înţelegerea datelor biologice sub formă de secvenţe. În mod special, s-a studiat folosirea conceptului de cross-entropie aplicat modelelor *n*-gram cu scopul căutării eficiente în baze de date. Rezultatele experimentale au indicat soliditatea noii strategii algoritmice în exprimarea similarităţii dintre proteine [BOGA'05].

Fiind dată simplitatea conceptuală a abordării acestei metode noi, ea apare ca o alternativă de dorit tehnicilor bine consolidate de până acum.

Considerând problema generală de separare dintre măsurile de similaritate "globală" şi "locală" dintre proteine, trebuie menţionat faptul că noua abordare aparţine primei categorii.

În timpul evaluării acestei noi metode s-a observat că dintre cele două variante introduse cea mai bună performanţă este asociată cu a doua (metoda alternantă). Această remarcă semnifică nevoia de a veni cu îmbunătăţiri care să depăşească posibilele identificări eronate ale secvenţelor similare datorate diferenţei semnificative în lungime. În cazul excepţional cînd toate secvenţele comparate ar avea aceeşi lungime, *metoda directă* este echivalentă cu *metoda alternantă* şi se comportă foarte bine.

În ce priveşte ordinul modelului *n*-gram folosit, după testarea ordinelor 2,3,4,5 s-a observat că performanţa metodei creşte odată cu ordinul modelului pâna la 4. După ordinul 5, datorită lipsei datelor, estimatorii probabilităţii maxime devin nerezonabil de uniformi şi mici. Aceasta stabileşte o limita superioară pentru modelul ales pentru această bază de date (probabil că un model de ordin ceva mai mare ar putea funcţiona în alte baze de date).

Înainte de a se trece la munca de îmbunătăţire a acestei metode trebuie remarcat faptul că este o tehnică statistică la bază. Ea ar putea deveni mai performantă prin incorporarea de cunoştinţe biologice cum ar fi lucrul cu grupe funcţionale de amino acizi

În final, un alt aspect care merită a fi luat în considerare este de a testa comportamentul acestei metode vis-a-vis de dimensiunea bazelor de date de secvenţe.

5.4. Determinarea similarităţii secvenţelor dintr-o bază structurată de proteine (SCOP). Experimente.

5.4.1. Baza de secvenţe

Aceeaşi strategie propusă pentru măsurarea similarităţii proteinelor a fost demonstrată şi validată de data aceasta folosind un set de 1460 proteine extrase din baza de resurse de secvenţe Astral SCOP, versiunea 1.67. În conformitate cu organizarea structurală a secvenţelor biologice în baza de secvenţe SCOP [SCOP], şi strategia sa de căutare, toate proteinele au fost selectate din clasa *a* (care include toate alfa proteinele caracterizate prin aceeaşi structură α-helix) astfel încât să prezinte grade diferite de similaritate. Din corpusul original disponibil au fost selectate şi incluse în baza de date experimentală numai acele familii de secveţe care conţin cel puţin 10 secvenţe (această restricţie va fi apreciată mai târziu deoarece a fost dictată de măsura de ***precizie*** adoptată pentru evaluarea metodei).

În acest fel, au fost selectate 31 de familii diferite, populate inegal (a se vedea în Tabel 5.4.1-1). Reamintesc aici că adnotatrea bazei de date experimentale păstrează adnotarea originală care se bazează pe semnificaţia biologică a conceptului de similaritate şi ca urmare poate fi considerată ca « valoare de

adevăr » pentru clasificarea proteinelor şi masurile de similaritate testate. În consecinţă, se aşteaptă ca toate proteinele aparţinând aceleiaşi familii să apară ca un grup compact de forme textuale şi fiind dată o măsură adecvată de similaritate am putea diferenţia diverse familii.

Baza de date (de 1460 proteine) a fost organizată în trei seturi diferite, deoarece rezultatele experimentale obţinute prin noua metodă se doresc a fi comparare cu cele furnizate de metoda adoptată de un alt program care poate avea limitări privind numărul de secvenţe procesate (poate accepta la intrare doar pana la 500 secvenţe). În Tabel 5.4.1-2 sunt incluse toate informaţiile necesare înţelegerii organizării adoptate. Fiecare set conţine un număr diferit de familii (depinde de numărul de secveţe conţinut de fiecare famile) astfel încât numărul total de secvenţe să nu depăşescă 500 per set. Baza de date completă (organizată în 3 subseturi) este disponibilă la cerere şi va putea fi accesată public de pe pagina web a laboratorului AIIA(Artificial Intelligence and Information Analisys, Universitatea Aristotel, Salonic, Grecia).

Tabel 5.4.1-1. Numărul de superfamilii şi familii folosite în experimente.

	Nr. Superfamii	Nr. Familii
Set1	5	(1,5), (2,3),(3,3), (4,2), (5,2) Total =15
Set2	4	(1,2), (2,4), (3,2), (4,2) Total=10
Set3	5	(1,1), (2,2), (3,1), (4,1), (5,1) Total=6

Pentru fiecare set s-a menţionat numărul superfamiliilor (în prima coloană). În a doua coloană este prezentat în mod explicit sub formă de perechi de forma (x,y), numărul de familii (y) care aprţin superfamiliei corespunzătoare (x).

Tabel 5.4.1-2. Familiile SCOP incluse în experimente[a]

Set1		Set2		Set3	
ID-ul familiei de proteine	Nr. Secv.	ID-ul familiei de proteine	Nr. secv.	ID-ul familiei de proteine	Nr. Secv.
a.39.1.1	10	a.1.1.2	28	a.123.1.1	109
a.39.1.2	30	a.1.1.3	213	a.22.1.1	103
a.39.1.4	14	a.4.1.1	21	a.22.1.3	15
a.39.1.5	119	a.4.1.2	19	a.45.1.1	85
a.39.1.8	16	a.4.1.12	67	a.93.1.1	78
a.26.1.1	32	a.4.1.3	11	a.133.1.2	76
a.26.1.2	45	a.25.1.1	52		
a.26.1.3	30	a.25.1.2	37		466
a.35.1.1	10	a.138.1.1	22		
a.35.1.2	26	a.138.1.3	27		
a.35.1.5	13				
a.3.1.1	85		497		
a.3.1.2	21				
a.118.1.1	34				
a.118.1.14	12				
	497				

[a]Pentru fiecare familie numele complet al fiecărei proteine este disponibil la: http://astral.berkeley.edu/scopseq-1.67.html.
Simbolul fiecărei proteine menţionate conţine patru domenii reprezentând complet clasificarea structurală biologică a proteinelor. Spre exemplu a.26.1.2 înseamnă clasa 'a', plierea '26', superfamilia '1', şi familia '2'. 'Nr.seq' indică numărul de secvenţe de proteine conţinute de fiecare familie.

5.4.2. Descrierea experimentelor.

Cele două variante ale strategiei propuse în analiza similarităţii sunt testate folosind noua bază de secvenţe extrasă din SCOP[BOGA'06d]. Pentru aceasta au fost urmaţi aceiaşi paşi clasici ai analizei exploratorii a datelor care au fost menţionaţi şi în experimentele anterioare. Matricea conţinând toate măsurile de disimilaritate posibile obţinute $D(S_i,S_j)$, i,j=1,2,...N pentru seturile 1-3 este ilustrată în Figura 5.4.2- respectiv, ca imagine la scara gri, pentru ambele variante algoritmice ale noii metode şi trei modele *n*-gram diferite. În schema de vizualizare adoptată toate matricele obţinute (după o normalizare adecvată), partajează o scală comună în care 1(alb) corespunde distanţei maxime în fiecare matrice. Se reaminteşte faptul că reprezentarea spaţială 'ideală' este o matrice albă cu nişte segmente negre în jurul liniei diagonale. Din aceste trei figuri este evident faptul că modelarea bazată pe modelul *4*-gram urmtă de ambele versiuni ale noului algoritm are o foarte bună performanţă în compararea secvenţelor din baza de proteine dată. Cu scopul de a furniza măsuri de performanţă cantitative pentru cele două variante, s-a adoptat şi un index al acurateţii căutării care este derivat din măsura ***precizie*** [BAEZ'99]. Acest index este raportul calculat prin împărţirea numarului de proteine corect clasificate (identificate de algoritm ca primele 10 cele mai similare) cu 10 (numărul minim de secvenţe din fiecare famile). Şi anume, fiecare proteină a fost la un moment dat considerată query şi s-a măsurat acurateţea primelor 10 secvenţe identificate în cadrul setului ca fiind cele mai similare cu secvenţa de query. Cu alte cuvinte, luând în considerare eticheta de clasă/familie a fiecărei proteine s-au identificat proteinele care au aceeaşi etichetă cu secvenţa de query (i.e. a number from 1 to 10). Procedura a fost repetată pentru toate proteinele în seturi individuale iar la final s-a făcut media părţilor estimate cu scopul de a furniza un scor de precizie totală pentru fiecare set în parte. Pentru completitudine, s-au repetat măsurările preciziei pentru cazul în care algoritmul nou s-ar fi aplicat întregului număr de 1460 de secvenţe din baza de proteine. Valorile obţinute astfel nu au fost semnificativ diferite faţă de valorile corespunzătoare celor trei seturi, dovedind o evidenţă în plus a robusteţii noii metode şi indică faptul că performanţa sa este bună raportat la mărimea bazei de date.

În bioinformatică, algoritmii de clasificare încearcă să grupeze secvenţele care sunt cumva înrudite. La modul general, algoritmii de clasificare sunt bazaţi pe legături singulare construind o închidere tranzitivă a secvenţelor cu o similaritate care depăşeşte un anumit prag. Scorul de similaritate este adesea bazat pe aliniamentul de secvenţe. De obicei clasificarea secvenţelor este folosită pentru a produce un set non-redundant de secvenţe reprezentative şi clasele de secvenţe sunt adesea sinonime (dar nu identice) cu familiile de proteine. Determinarea unei structuri reprezentative pentru fiecare grupare este scopul multor iniţiative genomice. Obiectivul general de a grupa proteinele în familii conduce la detectarea mai senzitivă de noi membri şi o discriminare îmbunătăţită împotriva potrivirilor false, pe baza caracteristicilor esenţiale conservate într-o familie.

Cea mai evidentă măsură a similarităţii (sau disimilarităţii) dintre două secvenţe este distanţa dintre ele. O cale de a începe investigarea modului de grupare este de a defini o metrică adecvată şi de a calcula matricea distanţelor dintre toate perechile de secvenţe. Dacă distanţa este o măsură bună a disimilarităţii, atunci se aşteaptă ca distanţele dintre secvenţele aflate în aceeaşi grupare să fie semnificativ mai mici decât distanţele faţă de secvenţele aflate în alte gupări [BOGA'06b].

Ca urmare a observeţiei din experimentele realizate asupra setului de 100 de secvenţe, unde s-a observat potenţialul noii metode de a clasifica secvenţele în două categorii evidente (secvenţe mutante ale proteinei generate de gena p53 şi restul) s-a propus o analiză mai aprofundată a acestui caz considerând rezultatelele obţinute din baza structurată de proteine. Astfel, folosind statisticile Hubert [LASK'02] în determinarea factorilor de corelaţie dintre grupările de secvenţe evidenţiate în reprezentarea vizuală şi valoarea lor de adevăr furnizată de codificarea din baza de date SCOP, s-au obţinut valorile tabulate în Tabel 5.4.2-2. Statisticile lui Hubert reprezintă o tehnică de corelare şi măsoară gradul de corespondenţă liniară dintre matricea distanţelor unor modele şi matricea etichetelor categoriilor corespunzătoare. Descrieri detaliate ale acestei metode pot fi gasite în [JAIN'88].

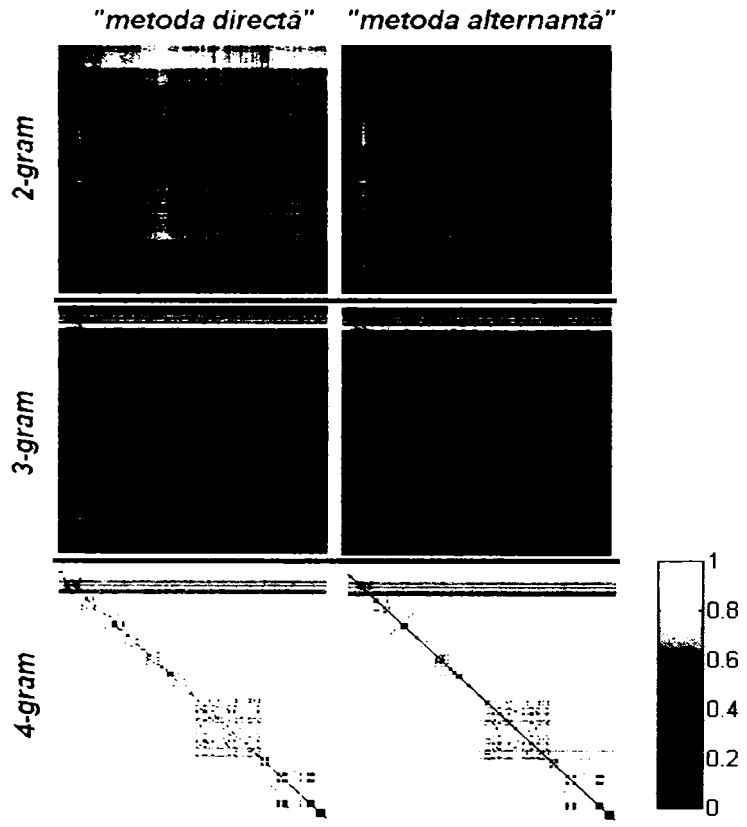


Figura 5.4.2-1.

Figura 5.4.2-1. Vizualizarea matricilor care conţin toate disimilarităţile posibile pentru perechile de proteine ale Set1 format din 497 secvenţe pentru modele 2,3,4-gram.

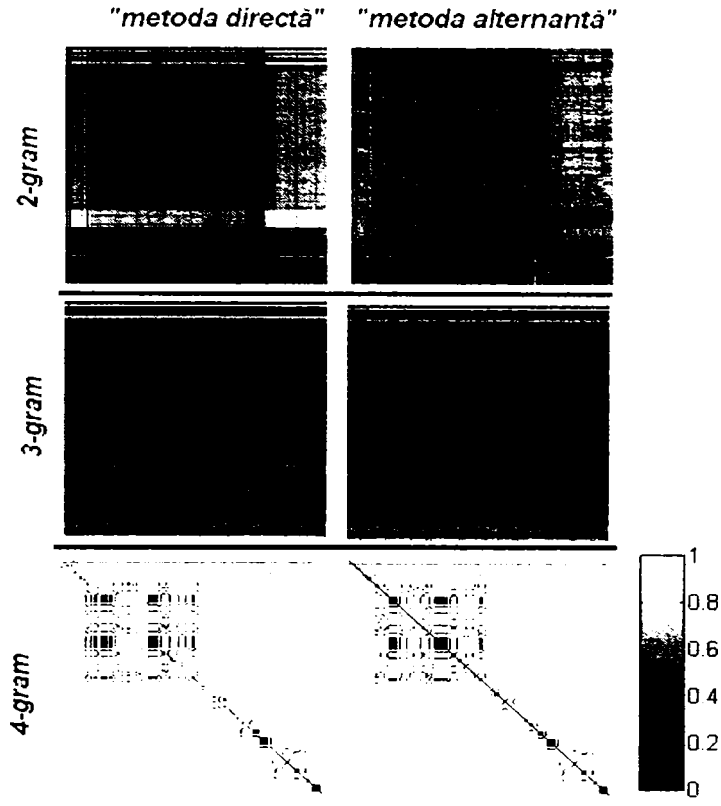


Figura 5.4.2-2.

Figura 5.4.2-2. Vizualizarea matricilor care conţin toate disimilarităţile posibile pentru perechile de proteine ale Set2 format din 497 secvenţe pentru modele 2,3,4-gram.

În Tabel 5.4.2-1 sunt incluse valorile de precizie furnizate de cele două abordări ale noului algoritm pentru diferite modele *n*-gram.

Tabel 5.4.2-1. Valorile preciziei

Set	Metoda Directă			Metoda Alternantă		
	2-gram	3-gram	4-gram	2-gram	3-gram	4-gram
1	0.439	0.662	0.830	0.471	0.646	0.823
2	0.446	0.650	0.874	0.439	0.605	0.860
3	0.534	0.865	**0.931**	0.574	0.828	**0.919**

Sunt prezentate valorile **preciziei** obţinute din rezultatele oferite de noua măsură de similaritate pentru ambele abordări testând modele 2,3,4-gram pentru setul de date structurate obţinut din baza de proteine SCOP.

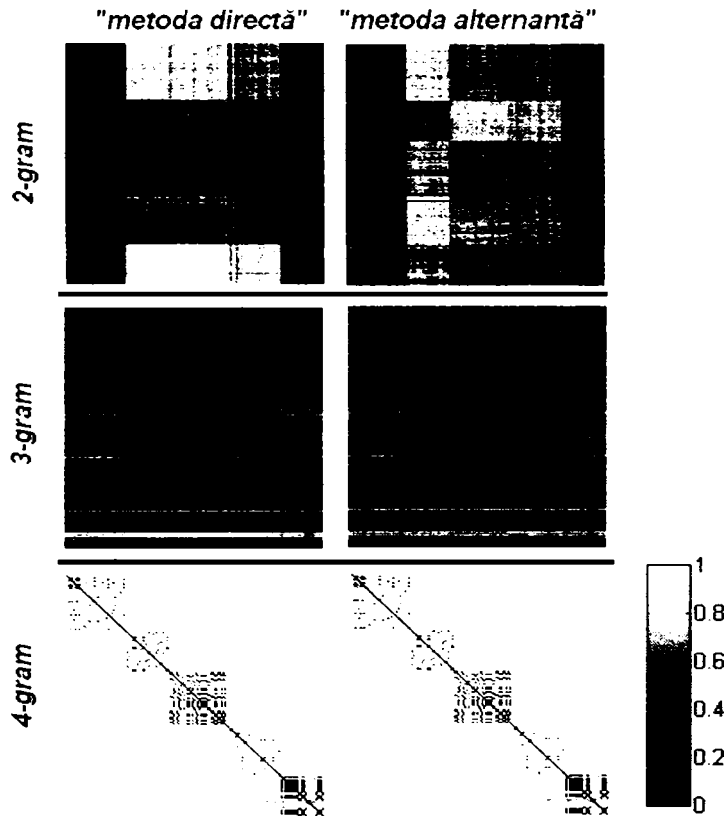


Figura 5.4.2-3. Vizualizarea matricelor care conţin toate disimilarităţile posibile pentru perechile de proteine ale Set3 format din 466 secvenţe pentru modele 2,3,4-gram.

Tabel 5.4.2-2. Valorile coeficienţilor de corelaţie pentru Set1, 2 şi 3 folosind statisticile Hubert.

Set	Metoda Directă	Metoda Alternantă
1	0.1230	0.1110
2	0.0347	0.0354
3	0.3339	0.3622

În interpretarea valorilor coeficienţilor de corelaţie sunt considerate bune valorile mari şi nu este cazul atins cu presupunerea cu care s-a pornit (posibila identificare a clasificarii biologice structurale a secvenţelor în familii/superfamilii). În aceste circumstanţe, intervenţia raţionamentului biologului ajută la elucidarea semnificaţiei grupurilor obţinute. Explicaţia vine din faptul că de multe ori, secvenţele de lungimi diferite şi cu un conţinut parţial identic pot aparţine unei aceleiaşi familii biologice. Astfel, grupurile identificate în acest caz sunt secvenţele similare în reprezentare textuală. Această concluzie este deja justificată de

rezultatele obţinute în identificarea grupului proteinelor mutante din baza de date anterioară.

5.4.3. Analiza şi interpretarea rezultatelor

În timpul evaluării noii metode folosind baza de date structurată, s-a observat performanţa apropiată a celor două variante algoritmice indicând lungimea apropiată a secvenţelor care se aşteaptă să aparţină aceleiaşi familii. Această remarcă induce observaţia conform căreia acele secvenţe (de lungime apropiată şi care par a fi asemanatoare) au un grad ridicat de similaritate. Dacă a doua metodă (*alternantă*) ar fi dat rezultate mai bune, ar fi trebuit să ne aşteptăm să avem diferenţe semnificative de lungime între secvenţele clasificate drept similare şi prin urmare aparţinând aceleiaşi familii. Aceasta presupune că este important să se vină cu îmbunătăţiri care să depăşească posibilele identificări eronate ale secvenţelor similare datorate diferenţei semnificative în lungime. Aşa după cum s-a observat şi în cazul primelor experimente pentru ordinul modelului *n*-gram folosit, după testarea ordinelor 2,3,4 s-a observat în Tabel 5.4.2-1. Valorile preciziei ce conţine valorile preciziei şi Figurile 5.4.2-1, 2, 3 că performanţa metodei creşte odată cu ordinul modelului pâna la 4. După ordinul 5, datorită lipsei datelor, estimatorii probabilităţii maxime devin nerezonabil de uniformi şi mici.

Dacă acordăm mai multă atenţie reprezentării vizuale din rezultatele obţinute (formele rezultate de-a lungul diagonalei principale în prezentarea 2D corespund unor grupuri bine definite de proteine, în special în cazul modelării cu 4-gram), putem considera că structura relevată prin folosirea noii metode de similaritate poartă o semnificaţie bilogică. Mai ales, presupunerea că fiecare grup definit este indicativ pentru existenţa unei familii/superfamilii de proteine. Cea mai clară evidenţă este oferită de Fig. 3, unde al treilea set de date conţine un număr de 6 familii aparţinând la 5 superfamilii diferite (conform definirii biologice) iar această informaţie coincide cu numarul de 5 grupuri identificate, care constituie o informaţie indusă de date (data-driven information).

Această prezumţie este demontată de calcularea coeficienţilor de corelaţie dintre etichetele corespunzătoare fiecărei secvenţe din setul original de date şi gradul de similaritate indicat de grupările rezultate. Chiar dacă precizia indică un grad ridicat de performanţă a metodei nu se poate spune acelaşi lucru despre clasificarea structurală. Totuşi, este de apreciat faptul că noua metodă de similaritate poate oferi o măsură de grupare pentru secvenţe similare din punct de vedere al compoziţiei lor textuale.

5.5. Concluzii

Metoda prezentată în această lucrare constituie un prim pas în investigarea implicării modelării lingvistice în caracterizrea, manevrarea şi înţelegerea datelor biologice sub formă de secvenţe. În mod special s-a studiat folosirea conceptului de cross-entropie aplicat modelelor *n*-gram cu scopul căutării eficiente în baze de date. Rezultatele experimentale au indicat soliditatea noii strategii algoritmice în exprimarea similarităţii dintre proteine. Fiind dată simplitatea conceptuală a acestei noi abordări, ea apare ca o alternativă de dorit tehnicilor bine consolidate de până acum. Aşa după cum au fost remarcate în secţiunile de analiză a rezultatelor

experimentale, principalele concluzii ale acestui capitol sunt rezumate în cele ce urmează.

- Considerând problema generală de separare dintre măsurile de similaritate "globală" şi "locală" dintre proteine, trebuie menţionat faptul că noua abordare aparţine primei categorii;
- Conceptul de *n*-gram chiar dacă a fost folosit în lucrări anterioare în domeniul bioinformaticii, cum ar fi [GANA'04b], [ERHA'80], [KARL'91], [KARL'96], pentru abordarea actuală el are meritul de a fi prima încercare de a adopta acest pas dual în compararea secvenţelor biologice. Prin urmare, au fost necesare unele experimente pentru a descoperi cel mai bun mod în care aceste idei pot fi valorificate în domeniul specific de aplicabilitate;
- Rezultatele obţinute în urma aplicării celor două variante ale noi metode în ambele experimente releva un grad ridicat de senzitivitate la lungimea secvenţelor. Dacă în primul caz metoda alternantă furnizează cele mai bune rezultate, în cazul bazei de date strucurate performanţa lor este oarecum apropiată. Această remarcă subliniază nevoia de a veni cu îmbunătăţiri care să depăşească posibilele identificări eronate ale secvenţelor similare datorate diferenţei semnificative în lungime.
- În cazul excepţional cînd toate secvenţele comparate ar avea aceeşi lungime, *metoda directă* este echivalentă cu *metoda alternantă* şi se comportă foarte bine.
- Pentru ordinul modelului *n*-gram folosit, după testarea ordinelor 2,3,4,5 în ambele experimente, s-a observat că performanţa metodei creşte odată cu ordinul modelului pâna la 4. După ordinul 5, datorită lipsei unei cantităţi mai mari de date, estimatorii probabilităţii maxime devin nerezonabil de uniformi şi mici. Aceasta stabileşte o limita superioară pentru modelul ales pentru această bază de date. Este foarte probabilă existenţa unor grupări de 4 amino acizi cu importanţă biologică ce participă în mod decisiv la caracterizarea secvenţelor de proteine.
- Folosirea rezultatelor furnizate de detectarea similarităţii secvenţelor în formarea de clusteri este evidentă doar că trebuie facută precizarea că această grupare se bazează pe similaritatea compoziţiei textuale. Desigur că intervenind cu raţionamente biologice se poate obţine mult mai mult decât o clasificare textuală.
- Înainte de a se trece la munca de îmbunătăţire a acestei metode trebuie remarcat faptul că aceasta este o tehnică pur statistică. Ea ar putea fi îmbunătăţită prin incorporara de cunoştinţe biologice cum ar fi lucrul cu grupe funcţionale de amino acizi. De asemenea, se are în vedere căutarea unei metode de ajustare a valorilor pentru menţinerea evenimentelor care nu participă la evaluarea secvenţelor şi care ar putea contribui la o mai mare consistenţă a evaluarii.

Ca o constatare finală, această metodă oferă un mod eficient de a captura caracteristicile comune ale secvenţelor comparate şi în acelaşi timp de a evita misiunea mai puţin plăcută de a alege parametrii, funcţii suplimentare sau metode de evaluare. Performanţa ridicată şi caracterul facil de implementare, luate împreună cu eficienţa computaţională fac din această metodă o alternativă promiţătoare la măsurile sofisticate, bine cunoscute, utilizate în comparearea secvenţelor de proteine.

MENŢIUNI:
Aceste experimente au fost realizate în cadrul proiectului EU Biopattern: Computational Intelligence for biopattern analysis in Support of eHealthcare, Network of Excellence Project No. 508803. Adresez pe această cale mulţumiri d-lui dr.ing. Nikos Laskaris pentru sprijinul acordat în realizarea şi interpretarea rezultatelor experimentale obţinute.

6. STUDIU COMPARATIV AL ALGORITMULUI DE SIMILARITATE PROPUS

Pentru a evalua performanţa noii metode comparativ cu a altor algoritmi existenţi în domeniul bioinformaticii este necesară o abordare comună atât a datelor de testare cât şi a funcţiilor executate.

În cazul de faţă s-a dorit obţinerea valorilor de similaritate pe baza comparaţiilor multiple de perechi de secvenţe. Aceasta este realizată folosind unul dintre cele mai populare programe de aliniament multiplu, Clustal W accesibil la http://www.ebi.ac.uk/clustalw/, site-ul oficial al Institutului European de Bioinformatică. Mai există şi alte programe care atestă o performanţă ridicată sau apropiată cu cea furnizată de Clustal W, în special din punct de vedere computaţional dar în cazul de faţă ele nu furnizează valori concrete de similaritate decât realizează aliniamentul multiplu al secvenţelor identificând regiunile comune. Acesta este însă un aspect mai puţin folositor în analiza rezultatelor obţinute cu ajutorul noii metode propuse, deoarece în prezenta fază a dezvoltarii aplicaţiei se urmăreşte mai puţin alinierea secvenţelor cât identificarea gradului de similaritate şi eventuala grupare corectă a secvenţelor similare.

Un alt aspect care se constituie într-un impediment în testarea noii metode în comparaţie cu alte programe este faptul că majoritatea acestora determină similaritatea unei secvenţe doar raportat la una din bazele de date standard prevăzute de aplicaţie. Aceasta limitează o analiză mai detaliată a rezultatelor obţinute şi a posibilelor observaţii suplimentare. Prin urmare, Clustal W permite o analiză comparativă pentru o bază de date comună, propusă de utilizator.

6.1. Algoritmul CLUSTAL W

Pentru alinierea a două secvenţe practica standard este de a folosi programarea dinamică [Need/Wunsch]. Aceasta ar garanta un aliniament optimal din punct de vedere matematic, fiind dată o tabelă de valori pentru potriviri şi nepotriviri între amino acizi sau nucleotide(ex. Matricele PAM250 sau BLOSUM62) şi penalităţi pentru inserţii sau ştergeri de diferite lungimi [THOM'94]. Încercările de generalizare a programării dinamice la aliniamente multiple sunt limitate la un număr redus de secvenţe scurte [THOM'94]. Pentru mai mult de 8 proteine de lungime medie problema este dificil de calculat, depinzand mult de putrerea computaţionala disponibilă. Prin urmare, toate metodele capabile să lucreze cu date mai mari, în timpi computaţionali rezonabili, folosesc euristici.

Clustal W este considerat unul dintre cele mai populare programe folosite pentru aliniamentul multiplu de secvenţe biologice şi a beneficiat de multe îmbunătăţiri faţă de versiunea iniţială Clustal V. El realizează un aliniament multiplu progresiv iar metoda de aliniament constă din trei mari etape: (i) toate perechile de

secvenţe sunt aliniate separat cu scopul de a calcula o matrice a distanţelor ce furnizează divergenţa fiecărei perechi de secvenţe; (ii) un arbore ghid este calculat pe baza matricei de distanţe; (iii) secvenţele sunt aliniate progresiv în conformitate cu gradul de ramificare din arborele ghid. Un exemplu ce foloseşte 7 secvenţe de globină cu structura terţiară cunoscută este ilustrat în Figura 3.4.5-2 din capitoulu 3 alături de alte observaţii generale referitoare la metoda de aliniament pe care o foloseşte.

Aliniamentul se aplică pentru secvenţe cu diverse grade de divergenţă şi are o semnificaţie biologică, calculand cele mai bune potriviri pentru secvenţele selectate. Ele sunt aliniate astfel încat identităţile, similarităţile şi diferenţele pot fi văzute. Relaţiile evoluţionare pot fi văzute via cladograme[35] sau filograme[36] ce pot fi opţional generate.

Programul oferă posibilitatea alegerii modului de aliniament dintre o metodă rapidă de aproximare [BASH'87] ce permite să fie aliniate un număr foarte mare de secvenţe şi o alta mai înceată dar cu un grad mai mare de precizie. În cazul primei metode, scorurile sunt calculate folosind numărul de k-tupluri de potriviri (înseamnă reziduuri identice, de obicei de lungime 1 sau 2 pentru proteine sau de lungime 2-4 pentru secvenţe de nucleotide) în cel mai bun aliniament dintre două secvenţe minus o penalitate fixă pentru fiecare gap. A doua metodă, după cum o afirmă autorii, oferă o mai mare acurateţe a scorurilor, fiind complet bazată pe programarea dinamică şi foloseşte două penalităţi pentru gap-uri (pentru deschidere sau extindere) şi o matrice de ponderi a amino acizilor. Scorurile în acest caz sunt calculate ca număr de identităţi în cel mai bun aliniament împărţit la numărul de reziduuri comparate (poziţiile de gap sunt excluse).

Scorurile în ambele metode sunt calculate iniţial ca procentaj de scor identitate şi sunt convertite în distanţe prin împărţirea cu 100 şi apoi scăzând din 1.0. În aceste distanţe iniţiale nu se aplică nici o corecţie pentru substituţii multiple.

În ce priveşte complexitatea acestui algoritm interesul este orientat spre complexitatea algoritmului de estimare a similarităţii dintre secvenţe. Astfel, conform unui studiu realizat în BMC Bioinformatics [EDGA'04] măsura de similaritate este identitatea fracţională calculată dintr-un aliniament global. Aliniamentul global al unei perechi de secvenţe sau profile este calculat folosind algoritmul spaţiului linear al lui Myers-Miller [37 din Edgar, Muscle] care este spaţiul O(L) şi timpul $O(L^2)$ în secvenţa tipică de lungime L. Fiind date N secvenţe şi astfel $N(N-1)/2= O(N^2)$ perechi, este nevoie, prin urmare, de timpul $O(N^2L^2)$ şi spaţiul $O(N^2+L)$ pentru constrirea matricei distanţă. Mai departe, operaţiile realizate cresc gradul de complexitate in timp şi spaţiu.

Interfaţa preluată din programul Clustal W accesibil online pe site-ul oficial al Institutului European de Bioinformatică(http://www.ebi.ac.uk/clustalw/), cu toate opţiunile de prelucrare posibile, este prezentată în Figura 6.1-1.

[35] Cladogramele sunt un arbori de diagrame folosiţi în clasifiarea biologica a membrilor din punct de vedere al evoluţiei pentru a ilustra relaţiile filogenetice[WIKI'06a]

[36] Filogramele sunt arbori filogenetici care indică relaţiile dintre diverse specii de plante sau animale grupate in funcţie de relatiile dintre ele şi de asemenea exprimă un sens al timpului sau ratei de evoluţie. Aspectul temporal al filogramei lipseşte dintr-o cladogramă[WIKI'06a]

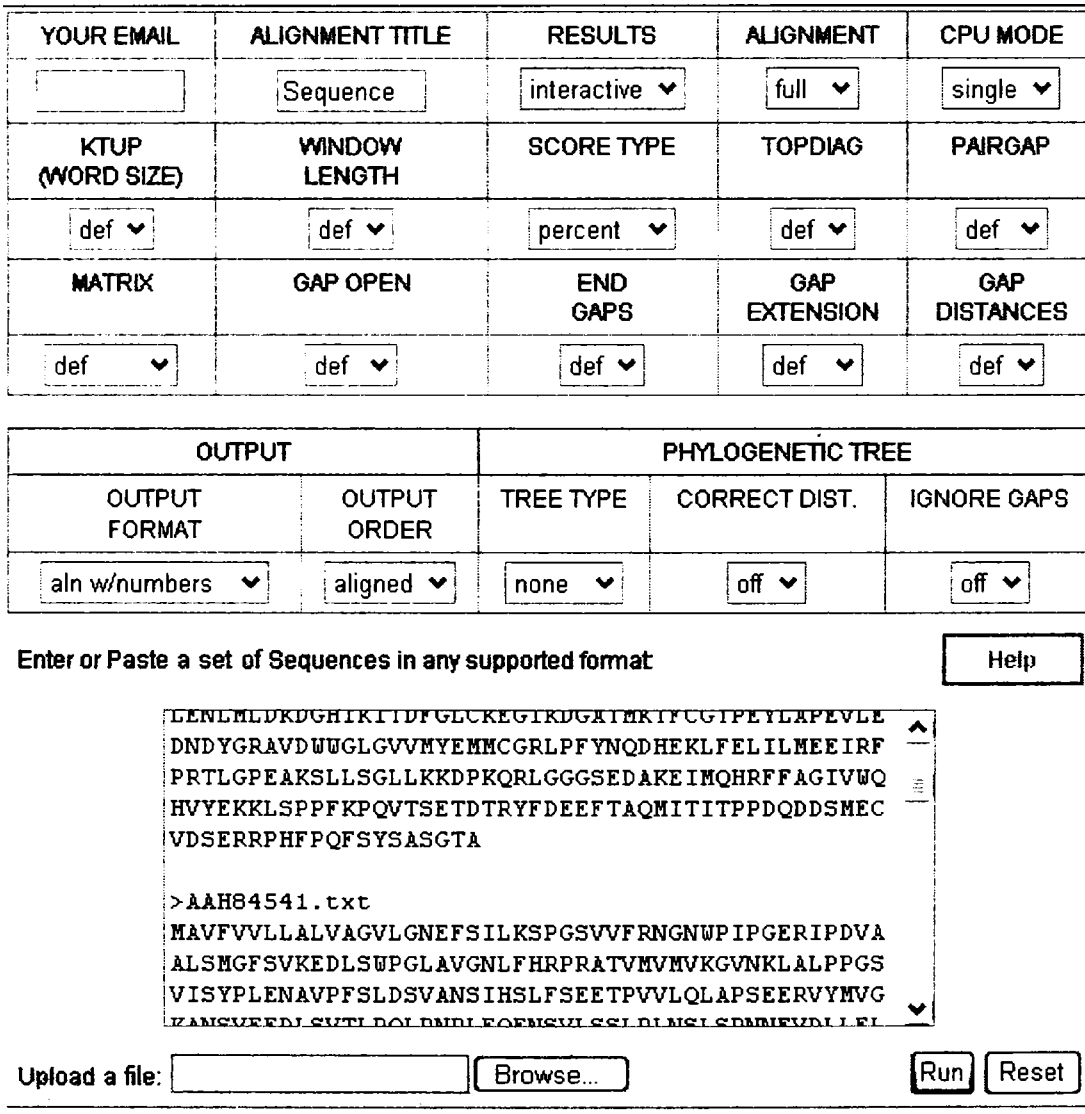


Figura 5.4.3-1 Interfaţa oferită de programul Clustal W

6.2. Experimente

Cu scopul realizării comparaţiei rezultatelor furnizate de Clustal W şi cele ale noii strategii propuse s-a folosit baza de proteine prezentată în subcapitolul 5.4.1. Aceasta răspunde necesităţii programului Clustal W de a considera un număr mai mic sau egal cu 500 de secvenţe la o execuţie. Acest tool presupune un set de parametri de intrare şi în acest scop s-a decis utilizarea valorilor parametrilor impliciţi cum ar fi : Protein Gap Open Penalty = 10.0, Protein Gap Extension Penalty = 0.2, Protein matrix = Gonnet, oferiţi de European Molecular Biology Laboratory şi European Bioinformatics Institute (EMBL-EBI) la adresa oficială de web http://www.ebi.ac.uk/

Pentru validare s-a folosit aceeaşi măsură dată de *precizie*[32] după cum a fost deja prezentată în 5.4.2. Astfel, pentru cele trei seturi de secvenţe şi metoda de similaritate aplicată de Clustal W au fost obţinute valorile prezentate în Tabel 6.2-1, alături de cele obţinute prin noua metodă propusă.

Merită menţionat faptul că noua strategie algoritmică atinge (în cazul modelului 4-gram al setului 3) performanţa ridicată a metodei CLUSTAL W.

Tabel 6.2-1 Valorile **preciziei** obţinute pe baza rezultatelor de similaritate furnizate de programul CLUSTAL W sunt în coloana 'CLUST.W' urmate de coloanele corespunzătoare celor două metode noi de similaritate folosind modele 2,3,4-gram pentru celei trei seturi de date.

Set	Clust.W	Metoda Directă			Metoda Alternantă		
		2-gram	3-gram	4-gram	2-gram	3-gram	4-gram
1	0.872	0.439	0.662	0.830	0.471	0.646	0.823
2	0.921	0.446	0.650	0.874	0.439	0.605	0.860
3	**0.932**	0.534	0.865	**0.931**	0.574	0.828	**0.919**

De asemenea, s-a aplicat aceeaşi metodă de verificare a gradului de corelare dintre grupurile obţinute utilizând similaritatea furnizată cu metoda din Clustal W şi gruparea pe baza căreia este structurată baza de secvenţe SCOP [BOGA'06e]. Astfel, folosind statisticile lui Hubert, rezultatele finale sunt prezentate în Tabel 6.2-2 alături de cele obţinute cu noua metodă.

Tabel 6.2-2. Valorile de corelaţie pentru set1, 2 şi 3 ale bazei de secvenţe obţinute din Astral SCOP folosind statisticile Huberts

Set	Clustal W	Metoda Directă	Metoda Alternantă
1	0.202	0.123	0.111
2	0.127	0.034	0.035
3	**0.297**	**0.334**	**0.362**

6.3. Interpretarea rezultatelor

Compararea rezultetelor obţinute cu Clustal W şi noua metodă arată că în termeni de performanţă a determinării similarităţii, noua metodă atinge performaţa obţinută de Clustal W pentru modelul 4-gram. În ce priveşte rezultatele aplicării tehnicii de corelare, după cum se observă, cele mai bune rezultate sunt obţinute pentru setul 3 de secvenţe. Astfel, recurgând la o investigare vizuală a secvenţelor implicate s-a observat că în setul 1 şi 2 multe secvenţe sunt de lungime relativ mică şi deoarece noua metoda funcţioneaza bine cu precădere pe secvenţe lungi, aceasta poate fi o explicaţie a diferenţelor de valori obţinute pentru aceste seturi de secvenţe. S-a observat că valoarea de adevăr a gruparilor biologice oferite de baza de secvenţe SCOP se potriveşte strict pentru funcţiile biologice ale proteinelor

acordând mai puţină importanţă aspectelor reziduale şi lungimii secvenţelor de proteine. Cu alte cuvinte, uneori interesul biologic nu este strans legat de toate aspectele care caracterizează secvenţele biologice.

6.5. Concluzii

Cele mai evidente concluzii la care s-a ajuns în urma investigaţiilor din acest capitol sunt cele ce urmează.

- În compararea dintre noua metodă de similaritate şi cea realizată de programul Clustal W, un prim aspect îl reprezintă complexitatea algoritmică. Astfel, dacă în cazul noii metode vorbim de o complexitate maximă de timp situată în jur de O(N×M) pentru ambele abordări, cu N şi M lungimile a două secvenţe comparate), în ce priveşte complexitatea spaţiului stocat pentru *metoda directă* se estimează o complexitate O(N), cu N lungimea secvenţei procesate cu rol de query, iar pentru *metoda alternantă* complexitatea spaţiului poate fi aproximată de O(min{N,M}). Aşa după cum a fost menţionat în cazul algoritmului Clustal W pentru o pereche de secvenţe de lungime L s-a estimat o complexitate a timpului de $O(L^2)$ şi a spaţiului de O(L). Astfel dacă e vorba de S secvenţe, complexitatea este calculată după formula $S(S-1)/2 = O(S^2)$ perechi, prin urmare se estimează timpul $O(S^2L^2)$ şi spaţiul $O(S^2+L)$ pentru constrirea matricei distanţă. În aceleaşi circumstanţe noua metodă ar presupune o complexitate a timpului $O(S^2NM)$ iar a spaţiului de $O(S^2+N)$ pentru *metoda directă* şi respectiv $O(S^2+\min\{N,M\})$ pentru *metoda alternantă*. Din acest punct de vedere, este evidentă o uşoară superioritate a noii metode propuse.
- Un alt atuu al noii metode este pus în valoare de faptul că în termeni de performanţă rezultatele obţinute pentru modelul 4-gram sunt foarte apropiate de cele ale Clustal W.
- Considerând simplicitatea algoritmică şi eficienţa computaţională a noii metode este justificată sugerarea ei ca primă alegere atunci când se realizează o căutare în baze mari de secvenţe.
- Există o motivare întemeiată de a perfecţiona această nouă metodă dar în acest moment trebuie remarcat faptul că în ce priveşte determinarea similarităţii noua tehnică folosită atinge performanţele metodei CLUSTAL W.

Menţiuni
Această parte de cercetare a fost susţinută de proectul European Biopattern: Computational Intelligence for biopattern analysis in Support of eHealthcare, Network of Excellence Project No. 508803.

7. CONCLUZII FINALE

Progresul rapid în cercetare legat de secvenţe genomice/proteomice motivează nevoia de conceptualizare şi analiză a acestora care îmbină cunoştinţele biologiei moleculare cu cele matematice şi ale sistemelor automate de calcul pentru eficientizarea determinării de răspunsuri la diverse întrebări biologice. Avantajele cooperării între aceste domenii ale ştiinţei valorificate în acestă teză sunt:

- Păstrarea informaţiei biologice sub forma organizată a bazelor de secvenţe;
- Posibiliteatea de explorare a unei cantităţi imense de informaţii în timp scurt;
- Extragerea automată de caracteristici pe baza comparaţiei secvenţelor, operaţiune de altfel dificilă şi mare consumatoare de timp prin analiza cu metode biologice sau manuale.

Pentru analiza şi compararea secvenţelor biologice este necesară folosirea de metode şi tehnici algoritmice adecvate. Dintre metodele de analiză abordate cu preponderenţă în această teză se conturează două tendinţe:

- Metode de aliniament bazate pe programarea dinamică,
- Metode orientate spre folosirea modelelor Markov ascunse.

Datorită complexităţii problemei de comparare a secvenţelor cu scopul determinării similarităţii sau al identificării de segmente cu rol biologic important au fost puse în valoare următoarele aspecte:

- Pentru determinarea similarităţii sunt necesare funcţii pentru evaluarea aliniamentelor în vederea stabilirii distanţei dintre secvenţe, care implică folosirea matricelor de substituţie sau tehnici algoritmice optimizate independente de aliniamentul automat al secvenţelor;
- Aliniamentele multiple folosesc de obicei tehnicile pentru compararea de perechi de secvenţe pe care le incorporează în algoritmi de parcurgere şi evaluare a tuturor secvenţelor implicate în procesul de comparare;
- Diversitatea de metode implică identificarea unei mulţimi distincte de determinare a similarităţii, stabilirea celui mai performant bazându-se pe minimizarea timpului şi spaţiului computaţional;
- Metodele de aliniament şi determinare a similarităţii propuse urmăresc îmbunătăţirea continuă a performanţelor sistemului de analiză urmărind performanţe ridicate la costuri cât mai mici;
- Teza subliniază implicarea estimărilor statistice în evaluarea rapoartelor de asociere pentru ponderea elementelor în matricele de substituţie şi în

- funcţiile de evaluare ale aliniamentelor, cu precădere pentru modelele Markov ascunse.

Metodele identificate se consituie în nuclee care stau la baza programelor comerciale destinate comparării, analizei şi extragerii automate de informaţii relevante din baze mari de secvenţe.

Tehnicile de recunoaştere au o largă aplicabilitate în domeniul bioinformaticii, iar obiectivul principal al tezei este de a veni cu o nouă soluţie la problemele de analiză şi căutare în baze de secvenţe biologice mari. S-a realizat o identificare a acestor tehnici în funcţie de scopul la care servesc şi s-a constatat, în concordanţă cu alte aprecieri ale specialiştilor, că pâna în momentul de faţă nu se cunoaşte vreo evaluare a performanţelor comparative a mai multor metode care să decidă asupra superiorităţii totale a uneia sau alteia. Explicaţia vine din aspecte ca:

- Tipul de format al datelor prelucrate. Nu este acelaşi lucru a se obţine secvenţe similare prin realizarea de aliniamente de secvenţe ca şi prin interogarea lingvistică a unor descrieri textuale despre aceste secvenţe;
- Scopul urmărit de utilizator. Scopul urmărit prin analiza şi/sau căutarea de secvenţe poate să fie de la unul simplu de a identifica obiectul de interes până la cel de a obţine detalii suplimentare despre respectivul „obiect".

Pornind strict de la aceleaşi premise şi urmărind aceleaşi obiective se poate concluziona că vor fi alese metodele cu timpi computaţionali cât mai reduşi, consum minim de resurse şi gradul ridicat de acurateţe al rezultatelor.

Un aspect caracteristic tuturor metodelor de comparaţie şi analiză investigate în această teză este gradul ridicat de dependenţă de estimări de diverşi parametri, sau estimări biologice colectate din diverse surse.

- La metodele de aliniament este evidentă necesitatea implicării matricei de substituţie şi a evaluărilor de penalităţi de gap (pentru marcarea absenţelor de elemente) sau inserţii şi ştergeri de amino acizi;
- Folosind modelele Markov ascunse trebuie avută în vedere calcularea de diverse ponderi de tranziţie de la o stare la alta sau extragerea de „profile" din cadrul familiilor de secvenţe pentru a putea evalua gradul de apartenenţă al unei secvenţe la o familie, folosind algoritmi bazati pe principiile implementării automatelor finite deterministe (AFD).

Prin analizarea metodelor existente în compararea secvenţelor biologice şi a experimentelor folosind principiile modelării lingvistice aplicate pentru sistemele de recunoaştere a vorbirii s-a dorit realizarea unui cadru informaţional pentru a motiva şi a atrage atenţia asupra contribuţiei personale a autoarei la dezvoltarea unei noi metode eficiente de comparaţie, analiză şi căutare în baze de secvenţe biologice.

Abordări ale conceptelor folosite în sistemele de recunoaştere a vorbirii sunt întâlnite în literatura biologică de specialitate orientate cu precădere către analiza secvenţelor, construirea de profile pentru gruparea lor sau generarea de aliniamente de secvenţe.

Din experimentele realizate de autoare, se poate concluziona că modelele lingvistice statistice se pot identifica drept un domeniu de importanţă majoră în

analiza şi expluatarea unei cantităţi mari de informaţie fiind totodată uşor de manipulat şi cu rezultate eficiente în aplicarea lor.

Metoda nou propusă în această teză constituie un prim pas în investigarea implicării modelării lingvistice în caracterizarea, manevrarea şi înţelegerea datelor biologice sub formă de secvenţe.

În mod special s-a studiat folosirea conceptului de entropie încrucişată (cross entropie) aplicat modelelor *n*-gram cu scopul căutării eficiente în baze de date. Rezultatele experimentale au indicat soliditatea noii strategii algoritmice în exprimarea similarităţii dintre proteine.

Fiind dată simplitatea conceptuală a acestei noi abordări, ea apare ca o alternativă de dorit tehnicilor studiate şi utilizate până în prezent. Pe baza analizei rezultatelor experimentale referitoare la eficienţa metodei propuse se evidenţiază aspectele principale:

+ Considerând problema generală de separare dintre măsurile de similaritate "globală" şi "locală" dintre proteine, noua abordare aparţine primei categorii;
+ Conceptul de *n*-gram, folosit în lucrări anterioare în domeniul bioinformaticii, pentru abordarea actuală are meritul de a fi prima încercare de a adopta acest pas dual în compararea secvenţelor biologice.
+ Rezultatele obţinute în urma aplicării celor două variante ale noi metode (metoda directă şi metoda alternantă) în ambele experimente relevă un grad ridicat de senzitivitate la lungimea secvenţelor.
    - Dacă în cazul bazei de secvenţe cu mutaţii ale genei P53 metoda alternantă furnizează cele mai bune rezultate, în cazul bazei de date strucurate derivate din SCOP performanţa lor este oarecum apropiată.
    - În cazul excepţional cînd toate secvenţele comparate ar avea aceeşi lungime, *metoda directă* este echivalentă cu *metoda alternantă* şi se comportă foarte bine.
+ Pentru ordinul modelului *n*-gram folosit, după testarea ordinelor 2,3,4,5 în ambele experimente, s-a observat că performanţa metodei creşte odată cu ordinul modelului pâna la 4. După ordinul 5, datorită lipsei datelor, estimatorii probabilităţii maxime devin nerezonabil de uniformi şi mici.
    - Aceasta stabileşte o limita superioară pentru modelul ales în această bază de secvenţe (este sugerată observarea existenţei unor grupări de 4 amino acizi cu importanţă biologică ce pot participa în mod decisiv la caracterizarea secvenţelor de proteine).
+ Rezultatele furnizate de reprezentările grafice ale experimentelor pentru detectarea similarităţii secvenţelor în formarea de clusteri este evidentă dar

cu precizarea că această grupare se bazează pe similaritatea compoziţiei textuale.

- Intervenind cu raţionamente biologice se poate obţine mult mai mult decât o clasificare textuală.

+ Metoda propusă foloseşte tehnici statistice de evaluare dar ea poate fi dezvoltată şi îmbunătăţită prin incorporara de cunoştinţe biologice cum ar fi lucrul cu grupe funcţionale de amino acizi şi impunerea unor ajustări în cazul evenimentelor singulare, purtătoare de valoarea zero în momentul evaluării.

Ca o remarcă finală, această metodă oferă un mod eficient de a captura caracteristicile comune ale secvenţelor comparate şi în acelaşi timp de a evita misiunea de a alege parametri, funcţii suplimentare sau metode de evaluare. Performanţa ridicată şi caracterul facil de implementare, împreună cu eficienţa algoritmică fac din această metodă o alternativă promiţătoare la măsurile sofisticate, bine cunoscute, utilizate în comparearea secvenţelor de proteine.

Compararea performanţelor noii metode de determinare a similarităţii propuse şi Clustal W, unul dintre cele mai populare programe de aliniament multiplu de secvenţe, relevă:

+ Din punct de vedere al complexităţii algoritmice vorbim de o complexitate maximă de timp situată în jur de $O(N \times M)$ pentru ambele abordări, cu N şi M lungimile a două secvenţe comparate.
+ În ce priveşte complexitatea spaţiului stocat:
  - pentru *metoda directă* se estimează o complexitate O(N), cu N lungimea secvenţei procesate cu rol de query (referinţă în căutare),
  - iar pentru *metoda alternantă* complexitatea spaţiului poate fi aproximată de $O(\min\{N,M\})$.
+ Aşa după cum a fost menţionat în cazul algoritmului Clustal W, pentru o singură secvenţă de lungime L s-a estimat o complexitate a timpului de $O(L^2)$ şi a spaţiului de O(L). Dacă e vorba de N secvenţe, se observă o superioritate a noii metode propuse.

Un alt avantaj al noii metode este pus în valoare de faptul că în termeni de performanţă rezultatele obţinute pentru modelul 4-gram sunt foarte apropiate de cele ale Clustal W.

Datorită simplităţii algoritmice şi eficienţei sistemului automat de determinare a similarităţii este justificată sugerarea ei ca primă alegere atunci când se realizează o căutare în baze mari de secvenţe.

Există o motivare întemeiată de a perfecţiona această nouă metodă propusă, ea fiind în acest format o primă variantă de identificare a similarităţii folosind într-o mare măsură tehnici de evaluare statistică.

Contribuţii personale

Partea de contribuţii personale ale autoarei este structurată pe capitole pentru a facilita identificarea lor rapidă în conţinutul tezei.

În capitolul 3 s-a realizat o investigare a tehnicilor de determinare automată a similarităţii secvenţelor biologice propuse relativ recent. Ele vin ca alternativă la metodele de aliniament fundamentale studiate fi folosite în prezent şi oferă o imagine de ansamblu asupra noilor alternative propuse/experimentate pentru determinarea similarităţii secvenţelor (reprezentate ca structuri primare în cazul proteinelor).

În capitolul 4 aportul de contribuţii personale se evidenţiază în cadrul experimentelor care pun în valoare potenţialul modelelor lingvistice statistice în analiza textuală. Se porneşte de la implementarea programelor de analiză folosite, continuă cu perspectiva din care este abordată analiza modelelor lingvistice statistice folosite în sistemele automate de recunoaştere a vorbirii şi se finalizează în interpretările şi concluziile ce vor deveni bază de pornire în propunera noii măsuri de evaluare a contextului secvenţelor biologice.

O prezentare sintetică a acestor aspecte este următoarea:

- Dezvoltarea aplicaţiei in limbajul Perl 5.0 care permite o analiză flexibilă a modelelor lingvistice statistice;
- Experimente cu modele lingvistice de dimensiuni 2, 3, ..., 9, 12, 15, 18, 20 care depăşesc evaluările uzuale limitate de obicei la gradul de 4 şi foarte rar 5.
- Se vine cu propunerea unui indicator al evaluarii gradului de acurateţe al estimării şi anume, valoarea cross entropiei obţinute când setul de testare este acoperit în mod strict.
- Este realizat un studiu asupra influenţei evenimentelor *n*-gram care nu participă la evaluarea entropică. Astfel, atenţia a fost direcţionată spre frecvenţa apariţiei acestor evenimente constatând că evoluţia indicatorilor de frecveţă poate oferi o referinţă asupra conţinutului setului de testare.

În capitolul 5, atât noua metodologie propusă pentru determinarea similarităţii dintre secvenţele biologice, discutată pe larg în capitolele corespunzătoare, cât şi implementarea ei întregesc partea de contribuţii personale aduse la elaborarea acestei teze de doctorat. În cele ce urmează sunt remarcate aspectele principale ale acestor contribuţii:

- Un fapt evident este transferul de cunoştinţe din domeniul sitemelor automate de recunoaştere în cel al biologiei moleculare.
- Tehnica aplicată în modelarea lingvistică pentru sistemele automate de recunoaştere a vorbirii a suferit modificări în sensul restrângerii numărului de evenimente considerat. Mai precis de la cel posibil, oferit de toate combinaţiile de lungimea *n*-gram ce ar putea fi întâlnite, la cel al evenimentelor existente în secvenţele curente analizate.

- Realizarea profilului de referinţă al secvenţei de căutare, adică a *modelului perfect* pe baza căruia se estimează distanţa dintre secvenţele comparate este un concept nou care fixează un nivel al aşteptărilor distribuţiei datelor (conceptualizat ca scor perfect) pentru secvenţa cu care se face comparaţia.
- Determinarea unei formule de calcul al similarităţii dintre două secvenţe pe baza evaluării statistice a conţinutului secvenţelor. Distanţa dintre scorul perfect al secvenţei de query şi cel obţinut prin proiectia acesteia asupra celei cu care se compară furnizează gradul de disimilaritate dintre cele două secvenţe.
- Abordarea *metodei alternante* care vine să evidenţieze nevoia de simetrie în aplicarea metodei, fapt expluatat însă în evidenţierea diferenţei de lungime a secvenţelor. Din această perspectivă se poate considera că noua metodă de similaritate realizează o "proiecţie" a secvenţei de query asupra secvenţei cu care este comparată.
- Rezultatele obţinute în cazul comparaţiei unei secvenţe de căutare cu mai multe secvenţe existente într-o bază de date pot să furnizeze şi un criteriu de ordonare al acestora în funcţie de valoarea scorului de similaritate/disimilaritate. Prin urmare, aceasta poate fi considerată şi o metodă de data mining.

O recunoaştere a caracterului inovativ propus prin aceasta nouă metodă este susţinută de publicaţiile naţionale şi internaţionale în care este prezentată şi analizată sub diverse aspecte (i.e. metodă de similaritate pentru secvenţe biologice, metodă de identificarea rapidă a secvenţelor mutante, folosirea metodei în procesul de grupare de secvenţe, pentru data mining, în compararea de secvenţe text în general).

În capitolul 6 se realizează o analiză şi evaluare comparativă a performanţelor metodei propuse cu cele ale programului Clustal W. Aceasta este realizată din punct de vedere al preciziei valorilor experimentale şi a complexităţii algoritmului care stă la baza generării valorilor de similaritate.

Destinată domeniului medical şi în special biologiei meleculare, teza are un character esenţialmente interdisciplinar combinând cunoştinţe de: biologie, automatizări, matematică, informatică, calculatoare. Încadrarea ei în domeniul automaticii este susţinută de scopul pe care îl deserveşte îndependent de asistenţa omului („automatos" în limba greacă înseamnă „functionare de la sine") precum şi de suportul teoretic al acestei specializări. Deoarece automatica este preocupată de automatizarea celor mai diferite (prin natura lor) procese (tehnice, energetice, militare etc.), tot astfel şi preocuparile cuprinse în această teză – analiza automată a similarităţii secvenţelor biologice – urmareşte realizarea automată a scopului propus, ceea ce o poate încadra în domeniul preocupărilor specializării de "Automatică".

Din punct de vedere al perspectivelor în dezvoltarea noii metode propuse şi analizate în această teză autoarea menţionează:

- Se poate considera că această nouă abordare a problematicii comparaţiei secvenţelor biologice este în esenţă valabilă pentru orice alt tip de secvenţe textuale. Desigur că există posibilitatea de a-i creşte gradul de performanţă intervenind în abordarea statistică folosită.
- Uşor de remarcat este senzitivitatea la lungimea secvenţelor comparate, aspect care poate fi folosit ca indicator al gradului de divergenţă în reprezentarea secvenţelor.
- Gruparea fiind o acţiune realizată tot pe baza determinării de relaţii de similaritate/disimilaritate dintre obiecte poate beneficia de o nouă strategie de determinare a similarităţii pentru cazul obiectelor reprezentate textual.
- Elaborarea unui produs IT performant folosind Oracle RDBMS (Relational DataBase Management System), bazat pe crearea unei structuri relaţionale poate concretiza utilitatea noii metode în practică. Acesta va pune în valoare atât relaţiile de similaritate dintre secvenţele biologice cât şi alte caracteristici ale secvenţelor implicate, venind semnificativ în sprijinul cercetarii bio-medicale.

Astfel, se conturează un potenţial semnificativ de folosire şi dezvoltare a acestei noi metode propuse în cazuri greu accesibile strategiilor existente.

ANEXA 1

Codurile acceptate de acizi nucleici sunt:

Nucleic Acid Code	Meaning
A	**A**denosine
C	**C**ytidine
G	**G**uanine
T	**T**hymidine
U	**U**racil
R	G A (pu**R**ine)
Y	T C (p**Y**rimidine)
K	G T (**K**etone)
M	A C (a**M**ino group)
S	G C (**S**trong interaction)
W	A T (**W**eak interaction)
B	G T C (not A) (**B** comes after A)
D	G A T (not C) (**D** comes after C)
H	A C T (not G) (**H** comes after G)
V	G C A (not T, not U) (**V** comes after U)
N	A G C T (a**N**y)
-	gap of indeterminate length

Codurile acceptate de amino acizi sunt:

Amino Acid Code	Meaning
A	Alanine
B	Aspartic acid or Asparagine
C	Cysteine
D	Aspartate
E	Glutamate
F	Phenylalanine
G	Glycine
H	Histidine
I	Isoleucine
K	Lysine
L	Leucine

M	Methionine
N	Asparagine
P	Proline
Q	Glutamine
R	Arginine
S	Serine
T	Threonine
U	Selenocysteine
V	Valine
W	Tryptophan
Y	Tyrosine
Z	Glutamate or Glutamine
X	Any
*	translation stop
-	gap de of indeterminate length

- **ALN/ClustalW format**

```
CLUSTAL W (1.82) multiple sequence alignment

FOSB_MOUSE
MFQAFPGDYDSGSRCSSSPSAESQYLSSVDSFGSPPTAAASQECAGLGEMPGSFVPTVTA 60
FOSB_HUMAN
MFQAFPGDYDSGSRCSSSPSAESQYLSSVDSFGSPPTAAASQECAGLGEMPGSFVPTVTA 60

************************************************************

FOSB_MOUSE
ITTSQDLQWLVQPTLISSMAQSQGQPLASQPPAVDPYDMPGTSYSTPGLSAYSTGGASGS
120
FOSB_HUMAN
ITTSQDLQWLVQPTLISSMAQSQGQPLASQPPVVDPYDMPGTSYSTPGMSGYSSGGASGS
120

********************************.***************:*.**:******
```

- **AMPS Block file format**

```
>Seq_1
>A0231
>HAHU
>Four_Alpha
>Globin
>GLobin_C
*
```

```
ARNDLQ
AAAAAA
PPPPPP
PP PPP
WW WWW
LLLLLL
IIVVLL
*
```

- **Codata**

```
ENTRY        IXI_234
SEQUENCE
           5      10      15      20      25      30
     1 T S P A S I R P P A G P S S R P A M V S S R R T R P S P P G
    31 P R R P T G R P C C S A A P R R P Q A T G G W K T C S G T C
    61 T T S T S T R H R G R S G W S A R T T T A A C L R A S R K S
    91 M R A A C S R S A G S R P N R F A P T L M S S C I T S T T G
   121 P P A W A G D R S H E
///
ENTRY        IXI_235
SEQUENCE
           5      10      15      20      25      30
     1 T S P A S I R P P A G P S S R - - - - - - - - - R P S P P G
    31 P R R P T G R P C C S A A P R R P Q A T G G W K T C S G T C
    61 T T S T S T R H R G R S G W - - - - - - - - - - R A S R K S
    91 M R A A C S R S A G S R P N R F A P T L M S S C I T S T T G
   121 P P A W A G D R S H E
///
```

- **EMBL**

```
ID   MMFOSB     standard; RNA; MUS; 4145 BP.
XX
AC   X14897;
XX
SV   X14897.1
XX
DT   23-NOV-1989 (Rel. 21, Created)
DT   12-SEP-1993 (Rel. 36, Last updated, Version 2)
XX
DE   Mouse fosB mRNA
XX
KW   fos cellular oncogene; fosB oncogene; oncogene.
XX
OS   Mus musculus (house mouse)
OC   Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
Mammalia;
OC   Eutheria; Rodentia; Sciurognathi; Muridae; Murinae; Mus.
XX
```

```
RN   [1]
RP   1-4145
RX   MEDLINE; 89251612.
RA   Zerial M., Toschi L., Ryseck R.P., Schuermann M., Mueller R., Bravo R.;
RT   "The product of a novel growth factor activated gene, fos B, interacts with
RT   JUN proteins enhancing their DNA binding activity";
RL   EMBO J. 8:805-813(1989).
XX
DR   MGD; MGI:95575; Fosb.
DR   SWISS-PROT; P13346; FOSB_MOUSE.
DR   TRANSFAC; T00291; T00291.
XX
CC   clone=AC113-1; cell line=NIH3T3;
XX
FH   Key             Location/Qualifiers
FH
FT   source          1..4145
FT                   /db_xref="taxon:10090"
FT                   /organism="Mus musculus"
FT   CDS             1202..2218
FT                   /db_xref="SWISS-PROT:P13346"
FT                   /note="fosB protein (AA 1-338)"
FT                   /protein_id="CAA33026.1"
FT
/translation="MFQAFPGDYDSGSRCSSSPSAESQYLSSVDSFGSPPTAAASQECA
FT
GLGEMPGSFVPTVTAITTSQDLQWLVQPTLISSMAQSQGQPLASQPPAVDPYDMPGTSY
FT
STPGLSAYSTGGASGSGGPSTSTTTSGPVSARPARARPRRPREETLTPEEEEKRRVRRE
FT
RNKLAAAKCRNRRRELTDRLQAETDQLEEEKAELESEIAELQKEKERLEFVLVAHKPGC
FT
KIPYEEGPGPGPLAEVRDLPGSTSAKEDGFGWLLPPPPPPPLPFQSSRDAPPNLTASLF
FT
THSEVQVLGDPFPVVSPSYTSSFVLTCPEVSAFAGAQRTSGSEQPSDPLNSPSLLAL"
XX
SQ   Sequence 4145 BP; 960 A; 1186 C; 1007 G; 991 T; 1 other;

     ataaattctt attttgacac tcaccaaaat agtcacctgg aaaacccgct ttttgtgaca  60
     aagtacagaa ggcttggtca catttaaatc actgagaact agagagaaat actatcgcaa  120
     actgtaatag acattacatc cataaaagtt tccccagtcc ttattgtaat attgcacagt  180
     gcaattgcta catggcaaac tagtgtagca tagaagtcaa agcaaaaaca aaccaaagaa  240
     aggagccaca agagtaaaac tgttcaacag ttaatagttc aaactaagcc attgaatcta  300
     tcattgggat cgttaaaatg aatcttccta caccttgcag tgtatgattt aacttttaca  360
     gaacacaagc caagtttaaa atcagcagta gagatattaa aatgaaaagg tttgctaata  420
     gagtaacatt aaataccctg aaggaaaaaa aacctaaata tcaaaataac tgattaaaat  480
     tcacttgcaa attagcacac gaatatgcaa cttggaaatc atgcagtgtt ttatttaaga  540
             .
             .
```

```
    ctctcaatga ctctgatctc cggtntgtct gttaattctg gatttgtcgg ggacatgcaa     4080
    ttttacttct gtaagtaagt gtgactgggt ggtagatttt ttacaatcta tatcgttgag     4140
    aattc                                                                4145
//
```

- **GCG/MSF**

```
DNA_MULTIPLE_ALIGNMENT 1.0
Four anthropoidea
MSF: 50  Type: N  Check: 2666 ..
//

Name: Homo_sapiens     Len: 50   Check: 8318   Weight: 1.00
Name: Pan_paniscus     Len: 50   Check: 7854   Weight: 1.00
Name: Gorilla_gorilla  Len: 50   Check: 7778   Weight: 1.00
Name: Pongo_pigmaeus   Len: 50   Check: 8716   Weight: 1.00
//

Homo_sapiens        AGUCGAGUC...GCAGAAAC
Pan_paniscus        AGUCGCGUCG..GCAGAAAC
Gorilla_gorilla     AGUCGCGUCG..GCAGAUAC
Pongo_pigmaeus      AGUCGCGUCGAAGCAGA..C
Homo_sapiens        GCAUGAC.GACCACAUUUU.
Pan_paniscus        GCAUGACGGACCACAUCAU.
Gorilla_gorilla     GCAUCACGGAC.ACAUCAUC
Pongo_pigmaeus      GCAUGACGGACCACAUCAUC
Homo_sapiens        CCUUGCAAAG
Pan_paniscus        CCUUGCAAAG
Gorilla_gorilla     CCUCGCAGAG
Pongo_pigmaeus      CCUUGCAGAG
```

- **GDE**

```
{
name "Short name for sequence"
longname "Long (more descriptive) name for sequence"
sequence-ID "Unique ID number"
creation-date "mm/dd/yy hh:mm:ss"
direction [-1|1]
strandedness [1|2]
type [DNA|RNA||PROTEIN|TEXT|MASK]
offset (-999999,999999)
group-ID (0,999)
creator "Author's name"
descrip "Verbose description"
comments "Lines of comments that can be fairly arbitrary text about a
sequence. Return characters are allowed, but no internal double quotes
or brace characters. Remember to close with a double quote"
sequence "gctagctagctagctagctcttagctgtagtcgtagctgatgctagct
gatgctagctagctagctgatcgatgctagctgatcgtagctgacg
```

gactgatgctagctagctagctagctgtctagtgtcgtagtgcttattgc"
}

- **Genebank**

```
LOCUS       MMFOSB                  4145 bp    mRNA    linear   ROD 12-SEP-1993
DEFINITION  Mouse fosB mRNA.
ACCESSION   X14897
VERSION     X14897.1  GI:50991
KEYWORDS    fos cellular oncogene; fosB oncogene; oncogene.
SOURCE      Mus musculus.
  ORGANISM  Mus musculus
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Rodentia; Sciurognathi; Muridae; Murinae; Mus.
REFERENCE   1  (bases 1 to 4145)
  AUTHORS   Zerial,M., Toschi,L., Ryseck,R.P., Schuermann,M., Muller,R. and
            Bravo,R.
  TITLE     The product of a novel growth factor activated gene, fos B,
            interacts with JUN proteins enhancing their DNA binding activity
  JOURNAL   EMBO J. 8 (3), 805-813 (1989)
  MEDLINE   89251612
   PUBMED   2498083
COMMENT     clone=AC113-1; cell line=NIH3T3.
FEATURES             Location/Qualifiers
     source          1..4145
                     /organism="Mus musculus"
                     /db_xref="taxon:10090"
     CDS             1202..2218
                     /note="fosB protein (AA 1-338)"
                     /codon_start=1
                     /protein_id="CAA33026.1"
                     /db_xref="GI:50992"
                     /db_xref="MGD:95575"
                     /db_xref="SWISS-PROT:P13346"

/translation="MFQAFPGDYDSGSRCSSSPSAESQYLSSVDSFGSPPTAAASQEC

AGLGEMPGSFVPTVTAITTSQDLQWLVQPTLISSMAQSQGQPLASQPPAVDPYDMPGT

SYSTPGLSAYSTGGASGSGGPSTSTTTSGPVSARPARARPRRPREETLTPEEEEKRRV

RRERNKLAAAKCRNRRRELTDRLQAETDQLEEEKAELESEIAELQKEKERLEFVLVAH

KPGCKIPYEEGPGPGPLAEVRDLPGSTSAKEDGFGWLLPPPPPPPLPFQSSRDAPPNL

TASLFTHSEVQVLGDPFPVVSPSYTSSFVLTCPEVSAFAGAQRTSGSEQPSDPLNSPS
LLAL"
BASE COUNT     960 a   1186 c   1007 g    991 t      1 others
ORIGIN
1 ataaattctt attttgacac tcaccaaaat agtcacctgg aaaacccgct ttttgtgaca
```

```
61 aagtacagaa ggcttggtca catttaaatc actgagaact agagagaaat actatcgcaa
121 actgtaatag acattacatc cataaaagtt tccccagtcc ttattgtaat attgcacagt
181 gcaattgcta catggcaaac tagtgtagca tagaagtcaa agcaaaaaca aaccaaagaa
241 aggagccaca agagtaaaac tgttcaacag ttaatagttc aaactaagcc attgaatcta
.
.
.
.
1021 gcagagggaa cttgcatcga aacttgggca gttctccgaa ccggagacta agcttccccg
1081 agcagcgcac tttggagacg tgtccggtct actccggact cgcatctcat tccactcggc
1141 catagccttg gcttcccggc gacctcagcg tggtcacagg ggccccccctg tgcccaggga
1201 aatgtttcaa gcttttcccg gagactacga ctccggctcc cggtgtagct catcaccctc
1261 cgccgagtct cagtacctgt
```

- **NBRF/PIR**

```
>P1;FOSB_MOUSEFOSB_MOUSE  338 bases
MFQAFPGDYD SGSRCSSSPS AESQYLSSVD SFGSPPTAAA SQECAGLGEM
PGSFVPTVTA ITTSQDLQWL VQPTLISSMA QSQGQPLASQ PPAVDPYDMP
GTSYSTPGLS AYSTGGASGS GGPSTSTTTS GPVSARPARA RPRRPREETL
TPEEEEKRRV RRERNKLAAA KCRNRRRELT DRLQAETDQL EEEKAELESE
IAELQKEKER LEFVLVAHKP GCKIPYEEGP GPGPLAEVRD LPGSTSAKED
GFGWLLPPPP PPPLPFQSSR DAPPNLTASL FTHSEVQVLG DPFPVVSPSY
TSSFVLTCPE VSAFAGAQRT SGSEQPSDPL NSPSLLAL*
```

- **PDB format**

```
COMPND    MOL_ID: 1;
COMPND   2 MOLECULE: S-ADENOSYLMETHIONINE SYNTHETASE;
COMPND   3 CHAIN: A, B;
COMPND   4 SYNONYM: MAT, ATP\:L-METHIONINE S-ADENOSYLTRANSFERASE;
COMPND   5 EC: 2.5.1.6;
COMPND   6 ENGINEERED: YES;
COMPND   7 BIOLOGICAL_UNIT: TETRAMER;
COMPND   8 OTHER_DETAILS: TETRAGONAL MODIFICATION
```

- **Pfam/Stockholm format**

```
# STOCKHOLM 1.0
#=GF ID CBS
#=GF AC PF00571
#=GF DE CBS domain
#=GF AU Bateman A
#=GF CC CBS domains are small intracellular modules mostly found
#=GF CC in 2 or four copies within a protein.
#=GF SQ 67
#=GS O31698/18-71 AC O31698
```

```
#=GS O83071/192-246 AC O83071
#=GS O83071/259-312 AC O83071
#=GS O31698/88-139 AC O31698
#=GS O31698/88-139 OS Bacillus subtilis
O83071/192-246          MTCRAQLIAVPRASSLAE..AIACAQKM....RVSRVPVYERS
#=GR O83071/192-246 SA  999887756453524252..55152525....36463774777
O83071/259-312          MQHVSAPVFVFECTRLAY..VQHKLRAH....SRAVAIVLDEY
#=GR O83071/259-312 SS  CCCCCHHHHHHHHHHHHH..EEEEEEEE....EEEEEEEEEE
O31698/18-71           MIEADKVAHVQVGNNLEH..ALLVLTKT....GYTAIPVLDPS
#=GR O31698/18-71 SS    CCCHHHHHHHHHHHHHHH..EEEEEEEE....EEEEEEEEHHH
O31698/88-139           EVMLTDIPRLHINDPIMK..GFGMVINN......GFVCVENDE
#=GR O31698/88-139 SS   CCCCCCCHHHHHHHHHHH..HEEEEEEE....EEEEEEEEEH
#=GC SS_cons            CCCCCHHHHHHHHHHHHH..EEEEEEEE....EEEEEEEEEH
O31699/88-139           EVMLTDIPRLHINDPIMK..GFGMVINN......GFVCVENDE
#=GR O31699/88-139 AS   ____________________*_____________________________
#=GR_O31699/88-139_IN   _____________1_____________2__________0____
//
```

Phylip

```
1 338 I
 FOSB_MOUSE   MFQAFPGDYD SGSRCSSSPS AESQYLSSVD SFGSPPTAAA
SQECAGLGEM
          PGSFVPTVTA ITTSQDLQWL VQPTLISSMA QSQGQPLASQ PPAVDPYDMP
          GTSYSTPGLS AYSTGGASGS GGPSTSTTTS GPVSARPARA RPRRPREETL
          TPEEEEKRRV RRERNKLAAA KCRNRRRELT DRLQAETDQL EEEKAELESE
          IAELQKEKER LEFVLVAHKP GCKIPYEEGP GPGPLAEVRD LPGSTSAKED
          GFGWLLPPPP PPPLPFQSSR DAPPNLTASL FTHSEVQVLG DPFPVVSPSY
               TSSFVLTCPE VSAFAGAQRT SGSEQPSDPL NSPSLLAL
```

Raw

```
ataaattcttattttgacactcaccaaaatagtcacctggaaaacccgctttttgtgaca
aagtacagaaggcttggtcacatttaaatcactgagaactagagagaaatactatcgcaa
actgtaatagacattacatccataaaagtttccccagtccttattgtaatattgcacagt
gcaattgctacatggcaaactagtgtagcatagaagtcaaagcaaaaacaaaccaaagaa
aggagccacaagagtaaaactgttcaacagttaatagttcaaactaagccattgaatcta
tcattgggatcgttaaaatgaatcttcctacaccttgcagtgtatgatttaacttttaca
```

RSF(Rich Sequence Format)

```
!!RICH_SEQUENCE 1.0
..
{
name  chkhba
type    DNA
longname  chkhba
checksum    980
creation-date  4/15/98 16:42:47
```

```
strand  1
sequence
  ACACAGAGGTGCAACCATGGTGCTGTCCGCTGCTGACAAGAACAACGTCAAGGGCATCTT

CACCAAAATCGCCGGCCATGCTGAGGAGTATGGCGCCGAGACCTTGGAAAGGATGTTCAC
  CACCTACCCCCCAACCAAGACCTACTTCCCCCACTTCGATCTGTCACACGGCTCCGCTCA
  ...
}
{
name  davagl
type   DNA
longname  davagl
checksum   7399
creation-date  4/15/98 16:42:47
strand  1
sequence

GTGCTCTCGGATGCTGACAAGACTCACGTGAAAGCCATCTGGGGTAAGGTGGGAGGCCAC
  GCCGGTGCCTACGCAGCTGAAGCTCTTGCCAGAACCTTCCTCTCCTTCCCCACTACCAAA
  ...
}
```

- **UniProtKB/Swiss-Prot**

```
ID   FOSB_MOUSE     STANDARD;      PRT;   338 AA.
AC   P13346;
DT   01-JAN-1990 (Rel. 13, Created)
DT   01-JAN-1990 (Rel. 13, Last sequence update)
DT   15-JUN-2002 (Rel. 41, Last annotation update)
DE   Protein fosB.
GN   FOSB.
OS   Mus musculus (Mouse).
OC   Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
OC   Mammalia; Eutheria; Rodentia; Sciurognathi; Muridae; Murinae; Mus.
OX   NCBI_TaxID=10090;
RN   [1]
RP   SEQUENCE FROM N.A.
RX   MEDLINE=89251612; PubMed=2498083;
RA   Zerial M., Toschi L., Ryseck R.-P., Schuermann M., Mueller R.,
RA   Bravo R.;
RT   "The product of a novel growth factor activated gene, fos B, interacts
RT   with JUN proteins enhancing their DNA binding activity.";
RL   EMBO J. 8:805-813(1989).
RN   [2]
RP   SEQUENCE FROM N.A.
RX   MEDLINE=92158623; PubMed=1741260;
RA   Lazo P.S., Dorfman K., Noguchi T., Mattei M.-G., Bravo R.;
RT   "Structure and mapping of the fosB gene. FosB downregulates the
```

RT activity of the fosB promoter.";
RL Nucleic Acids Res. 20:343-350(1992).
CC -!- FUNCTION: FOSB INTERACTS WITH JUN PROTEINS ENHANCING THEIR DNA
CC BINDING ACTIVITY.
CC -!- SUBUNIT: HETERODIMER (BY SIMILARITY).
CC -!- SUBCELLULAR LOCATION: NUCLEAR.
CC -!- INDUCTION: BY GROWTH FACTORS.
CC -!- SIMILARITY: BELONGS TO THE BZIP FAMILY. FOS SUBFAMILY.
CC ---

CC ---
DR EMBL; X14897; CAA33026.1; -.
DR EMBL; AF093624; AAD13196.1; -.
DR PIR; S04108; TVMSFB.
DR PIR; S35477; S35477.
DR HSSP; P01100; 1FOS.
DR TRANSFAC; T00291; -.
DR MGD; MGI:95575; Fosb.
DR InterPro; IPR000837; Leuzip_Fos.
DR InterPro; IPR004827; TF_bZIP.
DR Pfam; PF00170; bZIP; 1.
DR PRINTS; PR00042; LEUZIPPRFOS.
DR SMART; SM00338; BRLZ; 1.
DR PROSITE; PS00036; BZIP_BASIC; 1.
KW Nuclear protein; DNA-binding.
FT DNA_BIND 161 179 BASIC MOTIF.
FT DOMAIN 183 211 LEUCINE-ZIPPER.
SQ SEQUENCE 338 AA; 35976 MW; E9D031A4BEAE48EC CRC64;
MFQAFPGDYD SGSRCSSSPS AESQYLSSVD SFGSPPTAAA SQECAGLGEM PGSFVPTVTA
ITTSQDLQWL VQPTLISSMA QSQGQPLASQ PPAVDPYDMP GTSYSTPGLS AYSTGGASGS
GGPSTSTTTS GPVSARPARA RPRRPREETL TPEEEEKRRV RRERNKLAAA KCRNRRRELT
DRLQAETDQL EEEKAELESE IAELQKEKER LEFVLVAHKP GCKIPYEEGP GPGPLAEVRD
LPGSTSAKED GFGWLLPPPP PPPLPFQSSR DAPPNLTASL FTHSEVQVLG DPFPVVSPSY
TSSFVLTCPE VSAFAGAQRT SGSEQPSDPL NSPSLLAL
//

ANEXA 2

Aliniament de secvenţe[37]

Algoritmul de aliniament al secvenţelor folosind o abordare a programării dinamice încearcă întotdeauna să urmeze cel mai bun rezultat până la un moment dat.

Se încearcă alinierea a două secvenţe prin inserarea unor gap-uri la diferite locaţii necunoscute (i.e. obţinerea a două noi secvenţe cu goluri astfel încât să fie maximizat scorul acestui aliniament). Măsurarea scorului poate fi autodefinit. El este determinat prin **"premiul de potrivire", "penalitatea de nepotrivire" şi "penalitatea de gap"**. Cu cât scorul este mai mare cu atât este mai bun aliniamentul. Dacă ambele penalităţi sunt setate la 0 scopul este de a găsi întotdeauna un aliniament cu **potriviri maxime** până la un moment. **Potrivire maximă** = cel mai mare număr de potriviri pentru o secvenţă care se poate obţine prin permiterea tuturor ştergerilor posibile unei alte secvenţe.

Se obişnuieşte compararea a două secvenţe de ADN sau Proteine pentru predicţia de similaritate a funcţionalităţii lor.

Există trei algoritmi fundamentali larg răspândiţi: **Needleman-Wunsch (1970), Sellers (1974), Smith-Waterman (1981)**

Aliniamentul global: primii doi algoritmi sunt folosiţi pentru aliniamentul global.

Se presupune că se aliniază şi se potrivesc cele două secvenţe de la stânga la dreapta. Fiecare celulă din matricea de scoruri denotă un sub-aliniament între cele două subsecvenţe care formează o sub-matrice. În acest moment există trei moduri de a înainta, *"inserează un gap secvenţei X"*-- mergâng în jos în matrice, *"inserarea unui gap în secvenţa Y"*-- mergând la dreapta în matrice, *"încearcă o potrivire"* -- ambele secvenţe X şi Y merg mai departe în mod diagonal. Pentru acelaşi motiv, pentru fiecare celulă în matrice, există trei sub-matrice, (i.e. trei perechi de subsecvenţe conduc la acest sub-aliniament). Alegând cel mai bun scor dintre aceste trei căi, se obţine scorul optimal pentru sub-matricea curentă.

---

[37] Conform (http://www2.cs.uh.edu/~zhenzhao/Review/alignment.htm) University of Houston, Texas

ABC&AJ

	A	B	C	N
A	2	1	0	-1
J	1	1	0	-1
C	0	0	3	2

ABCN&AJ

	A	B	C	N
A				
J				
C	0	0	3	2

ABC&AJC

	A	B	C	N
A				
J				
C				

Needleman Wunsch algorithm with gap

	A	B	C	N
A	2	1	0	-1
J	1	1	0	-1
C	0	0	3	2

Target

ABCN&AJC
Target Cell Score

current match (one alignment)

ABC
AJ _

ABCN
AJ_ _

ABC
AJC

ABC
A_ _

after match

ABCN
AJ _C

ABCN_
AJ_ _C

ABCN
AJC _

ABC_N
A_ _ C

So: M[i,j]=M[i-1,j-1]+S[i,j]
M[i,j]: for ABCN & AJ_C
M[i-1,j-1]: for ABC & AJ_
M[i,j]=M[i-1,j-1]+Compare(N&C)
while Compare(N&C)=S[i,j]

So: M[i,j]=M[left/upper]-Gap,
instead of M[i,j]=M[upper]-Gap+S[i,j]
Because: M[i,j] is the score for ABCN_ & AJ_ _C
M[upper] is for ABCN & AJ_ _,
M[i,j]=M[upper]+Compare(_ &C),
while Compare(_ &C)=–Gap, or W(r)--a gap function

So: from 'ABC & A' to 'ABCN & A_ _C'
essentially, it follows 'ABC & A_ _' and
'ABC_ & A_ _J' and 'ABC_N & A_ _JC'
i.e., always go down to M[i-1,j-1], then step
forward together. So as to the other jumps.
Gap penalty will be applied in the above way.
i.e.,
r in [1,j-1), s in [1,i-1), W(r) and W(s) are penalty functions.

make more sense: A & AJC have score 2-1-1=0, instead of -1

Penalitatea de gap:

O funcţie generală de penalitate gap are forma $\mathbf{W(k) = c_{open} + c_{length} * k}$, unde **k** este lungimea gap-ului, $\mathbf{c_{open}}$ este constanta de penalitate de deschidere a gap-ului (gap-open penalty constant), şi $\mathbf{c_{length}}$ constanta de penalitate a lungimii de gap (gap-length penalty constant), de asemenea, uneori $\mathbf{c_{end}}$ va fi aplicată.

Pot să existe diferite formule pentru calculul scorului.

Problema:

Pentru alinierea a două secvenţe: ABCNJRQCLCRPM, AJCJNRCKCRBP

Algoritmul Needleman Wunsch : M[i,j]=Max{S[i,j]+M[i-1,j-1],M[i-1,j-k], M[i-r,j-1]}, nu poate face faţă cazului cu gap.

Condiţie:

Fiind dat: Premiul de potrivire: **Pa**=1 puncte, nici o penalitate pentru nepotrivire şi gap.

Proces:

Pas 1: pentru completarea matricei **S** cu 1 dacă e potrivire, altfel 0.

	A	B	C	N	J	R	Q	C	L	C	R	P	M
A	1	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	1	0	0	0	0	0	0	0	0
C	0	0	1	0	0	0	0	1	0	1	0	0	0
J	0	0	0	0	1	0	0	0	0	0	0	0	0
N	0	0	0	1	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	1	0	0	0	0	1	0	0
C	0	0	1	0	0	0	0	1	0	1	0	0	0
K	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	1	0	0	0	0	1	0	1	0	0	0
R	0	0	0	0	0	1	0	0	0	0	1	0	0
B	0	1	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

Pas 2: calculează scorul pe baza formulei **M[i,j]=Max{S[i,j]+M[i-1,j-1],M[i-1,j-k], M[i-r,j-1]}**. Următorul caz porneşte invers, din dreapta jos, astfel formula anterioară ar trebui modificată relativ. De asemenea, există o presupunere: linia (-1 prima linie) şi coloana (-1 prima coloană) care nu sunt afişate în matrice, dar vor fi folosite pentru calcularea primei linii (linia 0) şi coloane (coloana 0), sunt toate 0, ceea ce înseamnă un şir gol potrivit unei secvenţe în cazul absenţei penalităţii de gap.

	A	B	C	N	J	R	Q	C	L	C	R	P	M
A	1	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	1	0	0	0	0	0	0	0	0
C	0	0	1	0	0	0	0	1	0	1	0	0	0
J	0	0	0	0	1	0	0	0	0	0	0	0	0
N	0	0	0	1	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	1	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

Pas 3: se parcurge înapoi din colţul stânga sus, şi întotdeauna se selectează valoarea maximă din cea mai îndepărtată coloană şi linie, după care se sare la următorul maximum în următoarea linie sau coloană.

	A	B	C	N	J	R	Q	C	L	C	R	P	M
A	8	7	6	6	5	4	4	3	3	2	1	0	0
J	7	7	6	6	6	4	4	3	3	2	1	0	0
C	6	6	7	6	5	4	4	4	3	3	1	0	0
J	6	6	6	5	6	4	4	3	3	2	1	0	0
N	5	5	5	6	5	4	4	3	3	2	1	0	0
R	4	4	4	4	4	5	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

Pas 4: prezintă aliniamentul ca algoritm al lui Sellers arătat mai jos.

Algoritmul lui Sellers:

M[i,j]=Max{S[i,j]+M[i-1,j-1],M[i,j-1]+W(1),M[i-1,j]+W(1)}. //1 este numărul unu şi nu litera 'l'.

Condiţie:
Fiind dat: Premiul de Potrivire: **Pa**=2 puncte, Penalitatea de Nepotrivire: **Pm**=-1 puncte, Penalitatea de Gap: **Pg**=-1 point.

Proces:

Primul pas: se crează matricea iniţială, (N+1)X(M+1), M este pentru axa X , N pentru Y, şi această matrice este numită **S**.
Aceasta este, se ia partea albă a următoarei matrice completând Pa şi Pm relativ. Dacă nu există potrivire, nici nepotrivire şi nici gap, valoarea celulei este implicit 0.

Prima linie (linia -1) şi prima coloană (coloana -1) care sunt marcate cu gri înseamnă potrivirea unei secvenţe cu una goală/un gap. Strict vorbind, ele aparţin celui de-al doilea pas. În timp ce în acest pas ele toate ar trebui să fie **Pg** (ca şi -1 sau 0), exceptând intersecţia dintre linia -1 şi coloana -1 care este 0.

Oricum, este de sens comun de a le avea disponibile în mod direct.

		A	B	C	N	J	R	Q	C	L	C	R	P	M
A		2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
J		-1	-1	-1	-1	2	-1	-1	-1	-1	-1	-1	-1	-1
C		-1	-1	2	-1	-1	-1	-1	2	-1	2	-1	-1	-1
J		-1	-1	-1	-1	2	-1	-1	-1	-1	-1	-1	-1	-1
N		-1	-1	-1	2	-1	-1	-1	-1	-1	-1	-1	-1	-1
R		-1	-1	-1	-1	-1	2	-1	-1	-1	-1	2	-1	-1
C		-1	-1	2	-1	-1	-1	-1	2	-1	2	-1	-1	-1
K		-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
C		-1	-1	2	-1	-1	-1	-1	2	-1	2	-1	-1	-1
R		-1	-1	-1	-1	-1	2	-1	-1	-1	-1	2	-1	-1
B		-1	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
P		-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	2	-1

Pasul secund: se crează matricea de scoruri numită **M**, din stânga-sus spre direcţia dreapta-jos.

D: înseamnă din diagonala apropiată, L: vecinătatea stângă, U: vecinul superior pe baza formulei:

M[i,j]=Max{S[i,j]+M[i-1,j-1],M[i-1,j]-Pg,M[i,j-1]-Pg},
1<=i<=M+1,1<=j<=N+1, -Pg ar putea fi mai general ca: şi aminteşte **săgeata** dinspre celula/celulele care este/sunt *Max{S[i,j]+M[i-1,j-1],M[i-1,j]-Pg,M[i,j-1]-Pg}*, la *M[i,j]*.

		A	B	C	N	J	R	Q	C	L	C	R	P	M
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13
A	-1	D2	L1	L0	L-1	L-2	L-3	L-4	L-5	L-6	L-7	L-8	L-9	L-10
J	-2	U1	D1	DL0	DL-1	D1	L0	L-1	L-2	L-3	L-4	L-5	L-6	L-7
C	-3	U0	DU0	D3	L2	L1	DL0	DL-1	D1	L0	DL-1	L-2	L-3	L-4
J	-4	U-1	DU-1	U2	D2	D4	L3	L2	L1	DL0	DL-1	DL-2	DL-3	DL-4
N	-5	U-2	DU-2	U1	D4	LU3	D3	DL2	DL1	DL0	DL-1	DL-2	DL-3	DL-4
R	-6	U-3	DU-3	U0	U3	D3	D5	L4	L3	L2	L1	D1	L0	L-1
C	-7	U-4	DU-4	DU-1	U2	DU2	U4	D4	D6	L5	DL4	L3	L2	L1
K	-8	U-5	DU-5	U-2	U1	DU1	U3	DU3	U5	D5	DL4	DL3	DL2	DL1
C	-9	U-6	DU-6	DU-3	U0	DU0	U2	DU2	D5	DLU4	D7	L6	L5	L4
R	-10	U-7	DU-7	U-4	U-1	DU-1	D2	DLU1	U4	D4	U6	D9	L8	L7
B	-11	U-8	D-5	U-5	U-2	DU-2	U1	D1	U3	DU3	U5	U8	D8	DL7
P	-12	U-9	U-6	DU-6	U-3	DU-3	U0	DU0	U2	DU2	U4	U7	D10	L9

Al treilea pas: parcurgerea înapoi din colţul dreapta-jos în modul următor:

Parcurgerea simplă: fără reamintirea săgeţilor, adecvate potrivirilor maxime, altfel se pot introduce erori.

//întotdeauna se porneşte de la cea mai mare linie şi ce mai îndepărtată şi cea mai îndepărtată coloană.

```
for(i=M+1,j=N+1;i>0&&j>0;)
 {
    highlight the cell P(x,y) with biggest number in column P.x in 0~j and
row P.y in 0~i;
    if(ties happen) trace every P(x,y);
    i=P.x; j=P.y;
    i--; j--;
 }
```

	i	A	B	C	M	J	R	Q	C	L	C	R	P	M
j														
A			-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12
J		1		0	-1	1	0	-1	-2	-3	-4	-5	-6	-7
C		0	0		2	1	0	-1	1	0	-1	-2	-3	-4
J		-1	-1	2	2		3	2	1	0	-1	-2	-3	-4
M		-2	-2	1		3	3	2	1	0	-1	-2	-3	-4
R		-3	-3	0	3	3		4	3	2	1	1	0	-1
C		-4	-4	-1	2	2	4	4		5	4	3	2	1
K		-5	-5	-2	1	1	3	3	5		4	3	2	1
C		-6	-6	-3	0	0	2	2	5	4		6	5	4
R		-7	-7	-4	-1	-1	2	1	4	4	6		8	7
B		-8	-8	-5	-2	-2	1	0	3	3	5	8	8	7
P		-9	-9	-6	-3	-3	0	-1	2	2	4	7		9 ← start

Parcurgerea standard:
se porneşte din **start**, se urmează săgeţile înapoi la **end.**

end

	i	A	B	C	M	J	R	Q	C	L	C	R	P	M
j														
A		2	1	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
J		1	1	0	-1	1	0	-1	-2	-3	-4	-5	-6	-7
C		0	0	3	2	1	0	-1	1	0	-1	-2	-3	-4
J		-1	-1	2	2	4	3	2	1	0	-1	-2	-3	-4
M		-2	-2	1	4	3	3	2	1	0	-1	-2	-3	-4
R		-3	-3	0	3	3	5	4	3	2	1	1	0	-1
C		-4	-4	-1	2	2	4	4	6	5	4	3	2	1
K		-5	-5	-2	1	1	3	3	5	5	4	3	2	1
C		-6	-6	-3	0	0	2	2	5	4	7	6	5	4
R		-7	-7	-4	-1	-1	2	1	4	4	6	9	8	7
B		-8	-8	-5	-2	-2	1	0	3	3	5	8	8	7
P		-9	-9	-6	-3	-3	0	-1	2	2	4	7	10 ← 9	← start

Al patrulea pas: se inserează gap-uri şi se calculează scorul.

Pornind din colţul dreapta-sus (end), se citesc secvenţele X şi respectiv Y. Dacă există o săgeată în jos, se inserează un gap pentru secvenţa X , o săgeată la dreapta, inserează un gap pentru secvenţa Y.
Rezultate:

2-1+2-1+2-1+2-1+2-1+2+2-1+2-1=9

A	B	C	N	J		R	Q	C	L	C	R		P	M
A	J	C		J	N	R		C	K	C	R	B	P	
A	B	C		N	J	R	Q	C	L	C	R		P	M
A	J	C	J	N		R		C	K	C	R	B	P	

2-1+2-1+2-1+2-1+2-1+2+2-1+2-1=9

Aliniament Local: Smith-Waterman,

M[i,j]=Max(M[i-1,j-1]+S[i,j], Max(M[i,r]+W(r)), Max(M[s,j]+W(s)), 0)

Idea vine din aliniamentul global şi este similară în mod special cu algoritmul lui Sellers (1974). A se vedea urmatoarele:

	H	E	A	G	A	W	G	H	E	E
P	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
A	0.00	0.00	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
W	0.00	0.00	0.00	0.00	0.00	2.00	0.67	0.33	0.00	0.00
H	0.00	0.00	0.00	0.00	0.00	0.67	1.67	?		
E										
A										
E										

Difertenţa dintre aliniamentul local şi global constă în următoarea observaţie: aliniamentul local setează o celulă la 0 când ea devine negativă, ceea ce înseamnă că **o potrivire va fi repornită din această celulă** (deoarece valoarea unei celule vine din trei direcţii, M[i,j] atingând 0 denotă faptul că scorurile din toate cele trei direcţii sunt balansate a fi 'fără valoare'), în timp ce aliniamentul global va considera rezultatul de la început care este cumulativ la potrivirea curentă. De asemenea, când se parcurge înapoi, aliniamentul local va porni din valoarea maximă, deşi aliniamentul global va parcurge întreaga lungime a celor două secvenţe. Un

aliniament local ar trebui să fie metoda aleasă atunci când omologia a două secvenţe este distantă sau secvenţele sunt prea lungi

Proces:

Primul pas: se iniţializează matricea **S** pentru a fi completată cu 0.

	H	E	A	G	A	W	G	H	E	E
P	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
A	0.00									
W	0.00									
H	0.00									
E	0.00									
A	0.00									
E	0.00									

Pasul doi: se continuă iniţializarea matricei **S** prin adăugarea de puncte de Potrivire/Nepotrivire la matricea obţinută anterior. A se nota, linia (-1 prima) şi coloana (-1 prima) nu sunt afişate, deoarece ele nu vor avea efect pe linia 0 şi coloana 0 după aplicarea la formulă.

Al treilea pas: se calculează matricea de scoruri şi se parcurge înapoi de la cea mai ridicată valare dintr-o celulă din matrice şi urmând săgeţile până la celula care conţine valoarea 0 (care nu se include).

	H	E	A	G	A	W	G	H	E	E
P	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
A	0.00	0.00	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
W	0.00	0.00	0.00	0.67	0.00	2.00	0.67	0.33	0.00	0.00
H	0.00	0.00	0.00	0.00	0.33	0.67	1.67	1.67	0.33	0.00
E	0.00	1.00	0.00	0.00	0.00	0.33	0.33	1.33	2.67	1.33
A	0.00	0.00	2.00	0.67	1.00	0.00	0.00	0.00	1.33	2.33
E	0.00	1.00	0.67	1.67	0.33	0.67	0.00	0.00	1.00	2.33

Al patrulea pas: se prezintă rezultatul ca în algoritmul Sellers de mai sus.

ANEXA 3

123D,(http://123d.ncifcrf.gov/123D+.html). 123D-este un program care combină profile de secvenţe, predicţia structurii secundare şi potenţialele capacităţi de contact pentru a descrie o secvenţă de proteine prin setul de structuri.
Pentru o secvenţă oarecare de proteină selectată din baza de date a NCBI rezultatele furnizate de acest program sunt de următoarea formă :

Database of **4381** domains
Matrix: **gonnet**
Secondary structure prediction
Contact Capacity Potentials
Contact Capacity Potentials
Penalty for Gap opening: **-10.000000**
Penalty for Gap extension: **-1.000000**

Q_Seq: **musculus** length: **437**a.a.
20 best hits:

PDB ID	raw score	Z-score	a.a. aligned	% identities	SCOP family
1vhh__ 162 a.a.	6499	33.62	162	100	d.65.1.2
1at0__ 145 a.a.	4911	25.00	145	35	b.86.1.1
1cyda_ 244 a.a.	1036	3.98	236	14	c.2.1.2
1dq3a1 168 a.a.	995	3.76	135	15	b.86.1.2
1eqja2 274 a.a.	952	3.53	232	13	c.76.1.3
2ae2a_ 260 a.a.	948	3.51	250	14	c.2.1.2
1qg6a_ 257 a.a.	919	3.35	246	13	c.2.1.2
1e6wa_ 260 a.a.	916	3.33	251	11	c.2.1.2

....
Alignments:

```
seq 1vhh_ 6499 437 0.00 -100000.00 162 0
s_m = 6191 s_ss = 124 s_cc = 184 go = 0 ge = 0
HHHHHHHHHHHEH  HE                H   HH   HHHHH
MLLLLARCFLVILASSLLVCPGLACGPGRGFGKRRHPKKLTPLAYKQFIPNVAEKTLGAS
                        ***************************
.............................RRHPKKLTPLAYKQFIPNVAEKTLGAS
                        UUUUU    TTTEEETT TTTTTTTT

 EEE EEE      H       EEEE        HHHHHHHHHHHHHHEEHEEH
GRYEGKITRNSERFKELTPNYNPDIIFKDEENTGADRLMTQRCKDKLNALAISVMNQWPG
************************************************************
```

```
GRYEGKITRNSERFKELTPNYNPDIIFKDEENTGADRLMTQRCKDKLNALAISVMNQWPG
T     TTTTTTGGG EE  TTTEEE TTTT  GGGEE HHHHHHHHHHHHHHHHHTTT

 EEEE E         HHH     EEEEEE      HHHHHHHHHHH    EEEE
VKLRVTEGWDEDGHHSEESLHYEGRAVDITTSDRDRSKYGMLARLAVEAGFDWVYYESKA
************************************************************
VKLRVTEGWDEDGHHSEESLHYEGRAVDITTSDRDRSKYGMLARLAVEAGFDWVYYESKA
T EEEEET TTTT  TTTTGGGG  EEEEEETTT GGGHHHHHHHHHHH TTEEEEEETT
...
```

FFAS (Fold & Function Assignment System) un program utilizat pentru analiza secvenţelor biologice din punct de vedere structural.

Aliniamentele profil-la-profil au avantajul informaţiei prezente în secvenţele omoloage de proteine pentru a amplifica modelul secvenţei care defineşte familia. Ca un rezultat, ele sunt în stare să detecteze omologiile îndepărtate dincolo de utilizarea altor metode de comparare a secvenţelor.

O variantă îmbunătăţită a acestui tool este serverul FFAS03 care acceptă secvenţe de proteine furnizate de utilizator şi generează automat un profil care este comparat după aceea cu câteva biblioteci de profile ale secvenţelor incluzând PDB proteins, COG, PFAM, SCOP şi o bibliotecă de obiective genomice structurale. Serverul permite, de asemenea, accesul la analiza aliniamentului, aliniament multiplu şi tooluri de modelare comparativă.

Descrierea metodei se poate face pe scurt menţionând paşii principali. În primul pas este pregatit aliniamentul multiplu al secvenţelor folosind PSI-BLAST. Sunt rulate cinci execuţii pe baza de secvenţe de proteine NR85S. În al doilea pas, toate secvenţele găsite de PSI-BLAST sunt utilizate pentru a construi profilul. Ponderile sunt asignate secvenţelor pe baza similarităţii lor cu toate celelalte secvenţe incluse în acelaşi rezultat al PSI-BLAST. În plus, FFAS aliniază două profile folosind un algoritm local de programare dinamică. Valoarea scorului de comparaţie dintre poziţiile *n* şi *m* din cele două profile este calculată ca şi produs (dot product) al coloanei a *n*-a din primul profil şi a *m*-a coloană din al doilea profil. După asocierea de valori tuturor poziţiilor matricea obţinută este normalizată. Aliniamentul optimal este calculat de un algoritm de programare dinamică. În ultimul pas, scorul elementar astfel obţinut, este translatat în scorul final FFAS comparându-l cu distribuţia scorurilor primare obţinute prin compararea secvenţelor neânrudite.

Datele de intrare pentru serverul FFAS sunt sub forma secvenţelor de amino acizi în format FASTA. Serverul acceptă secvenţe de lungime cuprinsă între 25 şi 2000 reziduuri dar algoritmul a fost optimizat pentru secvenţe de lungime cuprinsă între 50 şi 500 reziduuri.

Exemplu :
Prezintă cel mai bun model din fiecare bază de date şi toate mostrele cu scoruri semnificative acoperind diverse fragmente căutate. O parte demonstrativă din rezultatul execuţiei pentru o secvenţă de proteine idicată (mus

musculus selectată aleator din baza de secvenţe locată pe serverul NCBI) este ilustrată de tabelul următor.

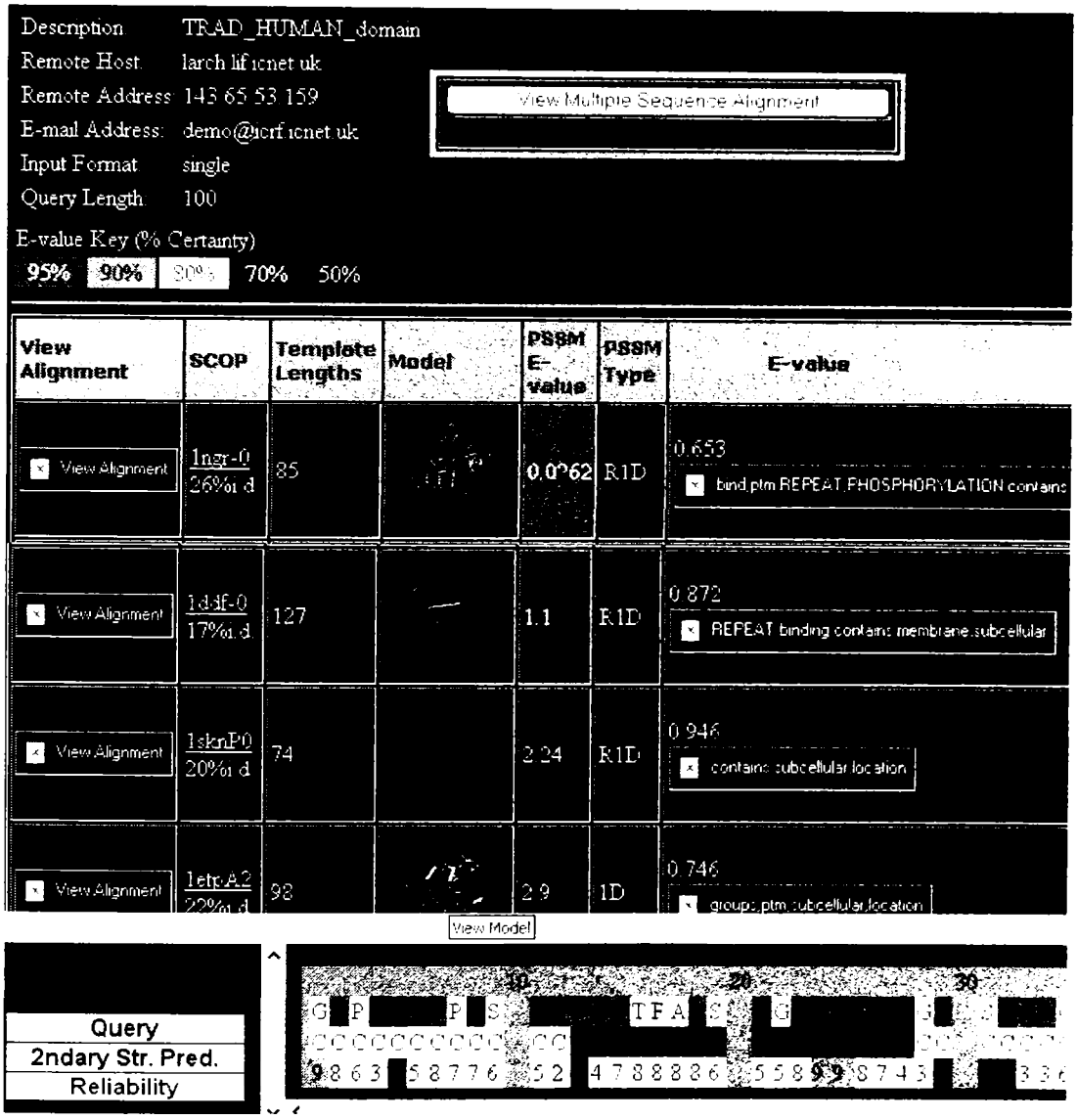
#	Query	Length	Result vs.	Range	Score	%id	Covered by template(s)
1	sonic hedgehog [Mus musculus] (11/16/05)	430	PDB0205	34-191	**-85.700**	94	1vhh mol:protein length:162 Sonic Hedgehog
				193-344	**-26.900**	34	1at0 mol:protein-het length:145 17-Hedgehog
2	gi|18203507|sp|Q9ULZ3| ASC_HUMAN Apoptosis-associated speck-like protein containing a CARD (hASC) (PYD and CARD domain containing protein) (Target of methylation-induced silencing 1) (Caspase recruitment domain protein 5) (11/16/05)	195	PDB0205	1-91	**-49.300**	100	1ucp A mol:protein length:91 Apoptosis-Associated Speck-Like Protein Conta
				103-195	**-28.600**	19	1cww A mol:protein length:102 Apoptotic Protease Activating Factor 1

3 (11/16/05)	195	PfamA160	5-91	**-38.900**	21	PF02758.5; AIM2_HUMAN/3-87; PAAD/DAPIN/Pyrin domain
			112-195	**-28.400**	11	PF00619.9; CED3_CAEVU/7-91; Caspase recruitment domain
4 gi|18203507|sp|Q9ULZ3| ASC_HUMAN Apoptosis-associated speck-like protein containing a CARD (hASC) (PYD and CARD domain containing protein) (Target of methylation-induced silencing 1) (Caspase recruitment domain protein 5) (11/16/05)	195	PfamA160	5-91	**-38.900**	21	PF02758.5; AIM2_HUMAN/3-87; PAAD/DAPIN/Pyrin domain
			112-195	**-28.400**	11	PF00619.9; CED3_CAEVU/7-91; Caspase recruitment domain

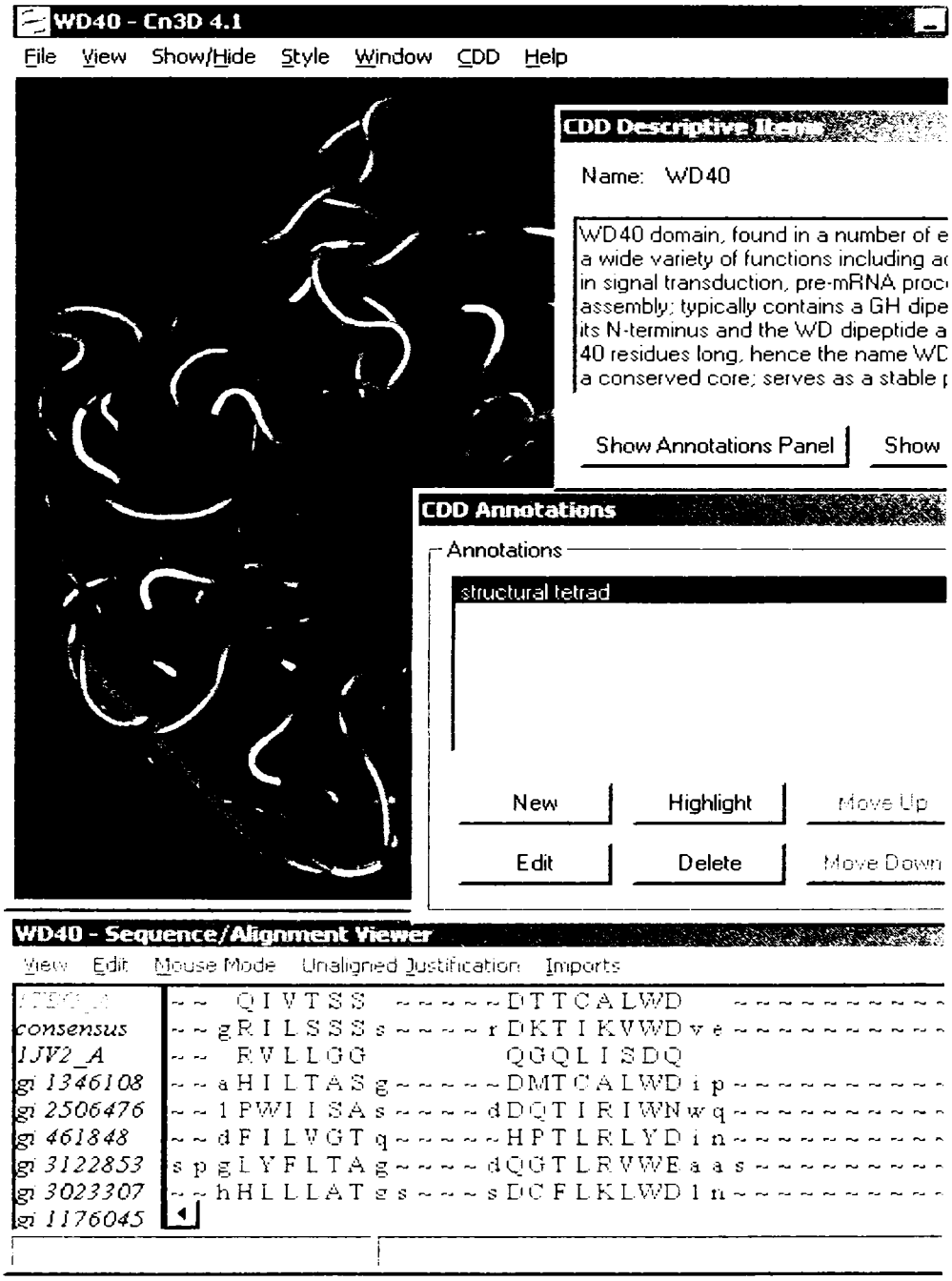
3D PSSM Fold Recognition Server , este o metodă web rapidă pentru recunoaşterea plierilor proteinelor folosind profilele secvenţelor reprezentate 1D sau 3D cuplate cu structura secundară şi Informaţia Potenţială de Solvatare (Solvation Potential Information).

Serverul 3D-PSSM este proiectat să preia secvenţa de proteină supusă investigării şi să încerce să prezică structura tridimensională şi funcţia sa probabilă. Se utilizează o bibliotecă de structuri de proteine deja cunoscute faţă de care este descompusă şi evaluată pentru compatibilitate proteina investigată. Sunt utilizate o varietate de componente de evaluare : 1D-PSSMs (profile de secvenţe construite din omologii apropiate) ; 3D-PSSMs (profile mai generale conţinând mai multe omologii îndepărtate), potrivirea elementelor structurii secundare şi tendinţele reziduurilor în secvenţe de query pentru a ocupa diverse nivele ale accesibilităţii solventului.

Exemplu de rezultate oferite în urma analizei pentru o secvenţă de proteine oarecare, TRAD_HUMAN_domain sunt ilustrate în tabelul următor.



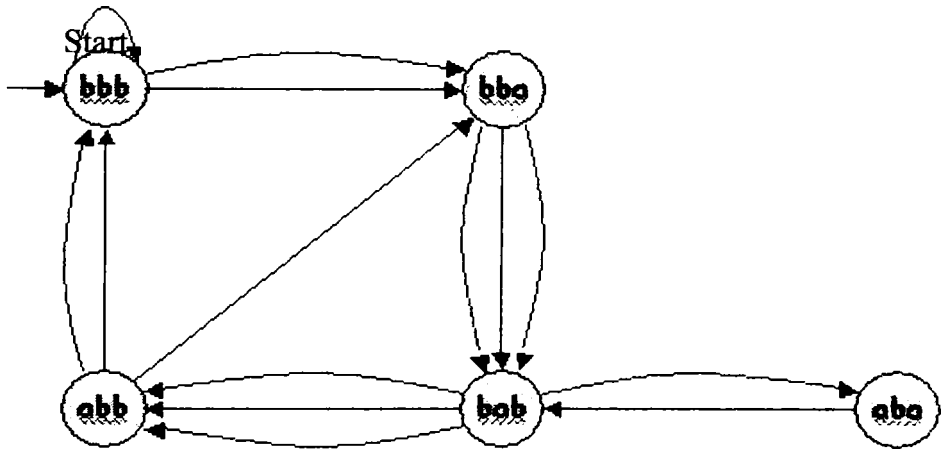
Cn3D, este o aplicaţie ajutătoare pentru căutarea web care permite vizualizarea structurilor 3D din baze de secvenţe gestionate de NCBI Entrez.
În cele ce urmează este făcută o exemplificare simplă a caracteristicilor afişate pentru domeniul WD40. Cn3D prezintă o structură de proteine reprezentativă, aliniamentul familiei şi tablouri de adnotare cu informaţii despre caracteristicile adnotate ale acestei familii de proteine. Reziduurile conservate într-o formă caracteristică acestui domeniu sunt marcate luminos.


WD40 - Cn3D 4.1
File View Show/Hide Style Window CDD Help
CDD Descriptive Item
Name: WD40
WD40 domain, found in a number of e
a wide variety of functions including ac
in signal transduction, pre-mRNA proc
assembly; typically contains a GH dipe
its N-terminus and the WD dipeptide a
40 residues long, hence the name WD
a conserved core; serves as a stable p
Show Annotations Panel
Show
CDD Annotations
Annotations
structural tetrad
New
Highlight
Move Up
Edit
Delete
Move Down
WD40 - Sequence/Alignment Viewer
View Edit Mouse Mode Unaligned Justification Imports
consensus
1JV2_A
gi 1346108
gi 2506476
gi 461848
gi 3122853
gi 3023307
gi 1176045

ANEXA 4

Simularea unui lanţ Markov şi generarea pseudo-aleatoare de text

Reprezentat grafic, lanţul Markov este compus dintr-o mulţime de stări cu o stare iniţială numită stare de start şi un set de tranziţii de la o stare la alta. Figura de mai jos ilustrează un lanţ Markov cu 5 stări şi 14 tranziţii.



Simularea unui lanţ Markov. Pentru a simula traiectoria prin lanţul Markov se incepe la starea de Start. La fiecare pas se selectează unul din arcele de plecare la alegere şi se trece la starea vecină. Spre exemplu, dacă lanţul Markov este în starea "bab", atunci va trazita în starea "abb" cu probabilitatea ¾ şi în starea "aba" cu probabilitatea ¼ . Mai jos sunt descrişi primii 10 paşi ai unei triectorii posibile pornind din "bbb":

bbb -> bbb -> bba -> bab -> abb -> bbb -> bbb -> bba -> bab -> abb

Generarea de text. Pentru generarea unui text pseudo-aleator în conformitate cu un model Markov de ordinul k construind un lanţ Markov dint-o secvenţă de text se pot parcurge următoarele etape: Se include o stare pentru fiecare *k*-gram distinct care apare în text; de asemenea, se includ N tranziţii, una pentru fiecare *k*-gram în text, care trece la următoarea *k*-gram (suprapusă); se tratează textul ca un şir ciclic pentru a atinge marginile. Spre exemplu, dacă *k* =3 şi secvenţa de text este

"bbbabbabbbbaba", atunci primele două tranziţii sunt de la "bbb" la "bba" şi de la "bba" la "bab" iar ultima (ciclic) este de la "abb" la "bbb". Lanţul Markov complet este ilustrat în figura anterioară.

vocabulary components total=74	occurrence frequency total frequency=621087	vocabulary components total=74	occurrence frequency total frequency=621087
!	18	a	41264
#	8	b	6490
$	180	c	13827
%	27	d	19810
&	29	e	60113
'	1779	f	9749
.	6370	g	8830
/	6	h	22864
0	1556	i	34080
?	34	j	664
A	1569	k	3728
B	1275	l	19678
C	1254	m	11381
D	622	n	34709
E	652	o	36601
F	532	p	9306
G	821	q	552
H	940	r	32847
I	1368	s	33433
J	500	t	41192
K	302	u	13006
L	696	v	4783
M	1785	w	8393
N	777	x	890
O	475	y	7999
P	831	z	459
Q	26	ξ	5167
R	765	Π	5167
S	1950	1	1220
T	2081	2	640
U	336	3	358
V	177	4	387
W	909	5	534
X	9	6	332
Y	170	7	357
Z	33	8	400
`	98407	9	608

Vocabularul activ utilizat în experimente

Detalii despre modul în care a fost creată baza de secvenţe mutante

Un set de date format din 50 de secvenţe de proteine aparţinând genei p53 [X54156 din NCBI GenBank] şi 50 secvenţe aleatoare au fost folosite pentru a evalua algoritmul propus pentru similaritatea proteinelor.
Mutaţile au fost selectate aleator din baza de secvenţe creată utilizând descrierea oferită de International Agency for Research on Cancer (IARC) Lyon, Franţa[38]. Baza de date IARC folosită este sub formă de fişier în format Excel şi conţine 18 585 întrări organizate în 42 coloane. Fiecare linie (coloană) reprezintă o singură mutaţie căreia îi este asignat un singur număr unic de identificare. Un număr unic de identificare este de asemenea atribuit mostrei de tumoră şi pacientului. Din fiecare înregistrare se folosesc câmpurile care definesc acea mutaţie:

- numărul de identificare unic al mutaţiei;
- locaţia mutaţiei în intron sau exon[39] în gena p53 ;
- pentru mutaţii în exon, numărul codon-ului la care este locată mutaţia (1-393);
- poziţia nucleotidelor mutaţiei pe baza intrării în bănca genetică, X54156;
- valorile de Yes (True) sau No (False) pentru a indica dacă poziţia mutaţiei cade într-o secvenţă CpG (Cytosine-phosphor-Guanine),
- valorile Yes (True) sau No (False) pentru a indica dacă poziţia mutaţiei cade într-o locaţie de conexiune ("splice site");
- natura mutaţiei;
- pentru mutaţii în exoni, secvenţa de bază a codon-ului în care are loc mutaţia;
- secvenţa bazei mutate;
- tipul de amino acid (wild-type amino acid) codificat în codon-ul în care are loc mutaţia (abreviere din trei litere a amino acidului);
- amino acidul mutat codificat în codon-ul în care a avut loc mutaţia (abreviaţia de trei litere a amino acidului) şi
- efectul mutaţiei şi al numărului de codon la care poate apare un codon de STOP.

Din baza de date creată au fost luate aleator 50 de mutaţii care produc *missense mutation* (când modificarea nucleotidelor produce modificare în codon, atunci amino acidul se modifică şi în final şi proteina), *nonsense mutations* (când modificarea în nucleotide determină modificarea codonului în TAA, TAG sau TGA – care sunt codon de STOP şi prin urmare nu se va produce nici un amino acid, rezultând într-o proteină mai scurtă) şi *silent mutations* (când schimbarea în nucleotide determină modificarea codon-ului dar nu se modifică nici amino acizii şi prin urmare nici proteina). Setul de secvenţe este urmatorul :

>AAH84538.txt
MSDVAIVKEGWLHKRGEYIKTWRPRYFLLKNDGTFIGYKERPQDVDQREAPLNNFSVAQCQL
MKTERPRPNTFIIRCLQWTTVIERTFHVETPEEREEWTTAIQTVADGLKKQEEEEMDFRSGSPS

[38] http://www.iarc.fr/p53/Somatic.html
[39] **Exonii** sunt regiuni ale AND-ului dintr-o genă care nu sunt înlăturate prin transcrierea ARN şi sunt păstrate în molecula finală mesager de ARN(mRNA)[WIKI'05g].

DNSGAEEMEVSLAKPKHRVTMNEFEYLKLLGKGTFGKVILVKEKATGRYYAMKILKKEVIVAKD
EVAHTLTENRVLQNSRHPFLTALKYSFQTHDRLCFVMEYANGGELFFHLSRERVFSEDRARFYG
AEIVSALDYLHSEKNVVYRDLKLENLMLDKDGHIKITDFGLCKEGIKDGATMKTFCGTPEYLAP
EVLEDNDYGRAVDWWGLGVVMYEMMCGRLPFYNQDHEKLFELILMEEIRFPRTLGPEAKSLLS
GLLKKDPKQRLGGGSEDAKEIMQHRFFAGIVWQHVYEKKLSPPFKPQVTSETDTRYFDEEFTA
QMITITPPDQDDSMECVDSERRPHFPQFSYSASGTA

>AAH84541.txt
MAVFVVLLALVAGVLGNEFSILKSPGSVVFRNGNWPIPGERIPDVAALSMGFSVKEDLSWPGL
AVGNLFHRPRATVMVMVKGVNKLALPPGSVISYPLENAVPFSLDSVANSIHSLFSEETPVVLQL
APSEERVYMVGKANSVFEDLSVTLRQLRNRLFQENSVLSSLPLNSLSRNNEVDLLFLSELQVLH
DISSLLSRHKHLAKDHSPDLYSLELAGLDEIGKRYGEDSEQFRDASKILVDALQKFADDMYSLY
GGNAVVELVTVKSFDTSLIRKTRTILEAKQAKNPASPYNLAYKYNFEYSVVFNMVLWIMIALALA
VIITSYNIWNMDPGYDSIIYRMTNQKIRMD

>AAH84544.txt
GPNARRVGSPGPRGVPGGKSAVGAPRPRCPGAMAMVTGGWGGPGGDTNGVDKAGGYPRAA
EDDSASPPGAASDAEPGDEERPGLQVDCVVCGDKSSGKHYGVFTCEGCKSFFKRSIRRNLSY
TCRSNRDCQIDQHHRNQCQYCRLKKCFRVGMRKEAVQRGRIPHSLPGAVAASSGSPPGSALA
AVASGGDLFPGQPVSELIAQLLRAEPYPAAAGRFGAGGGAAGAVLGIDNVCELAARLLFSTVE
WARHAPFFPELPVADQVALLRLSWSELFVLNAAQAALPLHTAPLLAAAGLHAAPMAAERAVAFM
DQVRAFQEQVDKLGRLQVDSAEYGCLKAIALFTPDACGLSDPAHVESLQEKAQVALTEYVRAQ
YPSQPQRFGRLLLRLPALRAVPASLISQLFFMRLVGKTPIETLIRDMLLSGSTFNWPYGSGQ

>AAH84554.txt
MASGSAGKPTGEAASPAPASAIGGASSQPRKRLVSVCDHCKGKMQLVADLLLLSSEARPVLFE
GPASSGAGAESFEQCRDTIIARTKGLSILTHDVQSQLNMGRFGEAGDSLVELGDLVVSLTECS
AHAAYLAAVATPGAQPAQPGLVDRYRVTRCRHEVEQGCAVLRATPLADMTPQLLLEVSQGLSR
NLKFLTDACALASDKSRDRFSREQFKLGVKCMSTSASALLACVREVKVAPSELARSRCALFSGP
LVQAVSALVGFATEPQFLGRAAAVSAEGKAVQTAILGGAMSVVSACVLLTQCLRDLAQHPDGG
AKMSDHRERLRNSACAVSEGCTLLSQALRERSSPRTLPPVNSNSVN

>AAH84612.txt
LLLLVAILLGSAVQTKTFDRIIGGEECVPHSQPWQVALYYFSDYLCGGILIDEWWVLTAAHCNQ
SNLQVLLGAHNRTKPTDHKQYTYAVKICPRCDFDPVTYNNDIMLLKLASKANMNCHVKTIQLA
SDLVEDNTECLASGWGTITSPEENYPDKLQCVNLSTVSNSECQACYPEDDITDNMLCAGNMA
GGKDTCKGDSGGPLVCNGELHGITSWGHYICGLPNKPGVFTKVFNYIDWISDIMQNENPCCY
ETT

>AAH84617.txt
MAVPKKSTKERLESLLDDLEVLSREVIETLALSRSQKLSQPGEENQILELLIQKDGEFQELMKVA
FSQGKTHQEMQVLEKEVEKRDSDIQQLQKQLKEAEHILATAVYQAKEKLKSIDKARKGVISSE
ELIKYAHRISASNAVCAPLTWVPGDPRRPYPTDLEMRSGLLGQMSNLPTNGVNGHLPGDALAA
GRLPDVLAPQYPWQSSDMSMNMLPPNHSNEFLMESLGPNKENEEDVEVMSTDSSSSSSDSD

>AAH84660.txt
MVHTARLSRRLCSAGGSMELGQLGHSLGLNGQQLERLVEVEEGRSLVLRVQEGDQKVAVYRS
DLRLCQTPKCAGDCGQLHLCRFYILGSCNRSPCKFNHCIRNAHNLSVLQKFHLDSMPIDELRQ
LLLQNDPNLLPDICSHYNRGDGPHGSCTYKNKCNKLHVCQHFLQGDCKFGEKCKRCHDLSEE
ETLKKLTKWQLSDSLLPGLLETYTNAHTLKSSCDRPPRNVVKSAEKKTKPGPSKSGTSQEIEEI
CLYFIRNSCSFKEKCVREHFHLPYRWQVYTNGTWKDLDNMEMIEKSYCDPNSRAVLVNLDFEA

MTYQSNKVKRLSTPSSASKPPHFVFTTDWKWYWMDEYNKWVEYGTESDRHANSTICSSDLE
NFYQSNQTADVKFKAGKHKYSLSFKNMVQRNLQYETKRRVCRRPVFVSGKDVKKKRSSTSEP
SKEDKNTPEHWDKGQTPDLGYKLVLLSPLSEEYSKIEAMFCRTLRTIRIHSIERIQNLALWEVYQ
W

>AAH84676.txt
MTFEPDPADLALSSIPGHETFDPRRHRFSEEELKPQPIMKKARKVQVPEEQKDEKYWSRRYKN
NEAAKRSRDARRLKENQISVRAAFLEKENALLRQEVVAVRQELSHYRAVLSRYQAQHGTL

>AAH84693.txt
MEGTQEALSGKMRLLFTPAARTSLLMLRLNEAALRALQECQQQQVRPVIAFQGQRGYLRLPGP
GWSCLFSFIVSQCGQEGGGLDLVYQRLGRSGPNCLHCLGSLRERLTIWAAMDTIPAPLLAQEH
LTEGTRESESWQDSEDEPEGHPQMALQEVSDPLASNHEQSLPGSSSEPMAQWEVRNHTYLS
NREPDQPLPSSASQKRLDKKRSAPITTEEPEEKRPRALPLASSPLQGLSNQDSPEEQDWGQDA
DGDSRLEQSLSVQSASESPSPEEVPDYLLQYSTIHSAEQQQAYEQDFETDYAEYRILHARVGA
ASQRFTELGAEIKRLQRGTPEHKVLEDKIVQEYKKFRKRYPSYSEEKRRCEYLHEKLSHIKGLIL
EFEEKNRGS

>AAH84726.txt
MTTSQKHRDFVAEPMGEKPVGSLAGIGDALGKRLEERGFDKAYVVLGQFLVLKKDEDLFREWL
KDTCGANAKQSRDCFGCLREWCDAFL

>BC008979.txt
MSDSWVPNSASGQDPGGRRRAWAELLAGRVKREKYNPERAQKLKESAVRLLRSHQDLNALLL
EVEGPLCKKLSLSKVIDCDSSEAYANHSSSFIGSALQDQASRLGVPVGILSAGMVASSVGQIC
TAPAETSHPVLLTVEQRKKLSSLLEFAQYLLAHSMFSRLSFCQELWKIQSSLLLEAVWHLHVQG
IVSLQELLESHPDMHAVGSWLFRNLCCLCEQMEASCQHADVARAMLSDFVQMFVLRGFQKNS
DLRRTVEPEKMPQVAVDVLQRMLIFALDALAAGVQEESSTHKIVRC

>BC036531.txt
MFSFVDLRLLLLLAATALLTHGQEEGQVEGQDEDIPPITCVQNGLRYHDRDVWKPEPCRICVC
DNGKVLCDDVICDETKNCPGAEVPEGECCPVCPDGSESPTDQETTGVEGPKGDTGPRGPRGP
AGPPGRDGIPGQPGLPGPPGPPGPPGPPGLGGNFAPQLSYGYDEKSTGGISVPGPMGPSGPRG
LPGPPGAPGPQGFQGPPGEPGEPGASGPMGPRGPPGPPGKNGDDGEAGKPGRPGERGPPGP
QGARGLPGTAGLPGMKGHRGFSGLDGAKGDAGPAGPKGEPGSPGENGAPGQMGPRGLPGE
RGRPGAPGPAGARGNDGATGAAGPPGPTGPAGPPGFPGAVGAKGEAGPQGPRGSEGPQGVR
GEPGPPGPAGAAGPAGNPGADGQPGAKGANGAPGIAGAPGFPGARGPSGPQGPGGPPGPKG
NSGEPGAPGSKGDTGAKGEPGPVGVQGPPGPAGEEGKRGARGEPGPTGLPGPPGERGGPGS
RGFPGADGVAGPKGPAGERGSPGPAGPKGSPGEAGRPGEAGLPGAKGLTGSPGSPGPDGKT
GPPGPAGQDGRPGPPGPPGARGQAGVMGFPGPKGAAGEPGKAGERGVPGPPGAVGPAGKDG
EAGAQGPPGPAGPAGERGEQGPAGSPGFQGLPGPAGPPGEAGKPGEQGVPGDLGAPGPSGA
RGERGFPGERGVQGPPGPAGPRGANGAPGNDGAKGDAGAPGAPGSQGAPGLQGMPGERGA
AGLPGPKGDRGDAGPKGADGSPGKDGVRGLTGPIGPPGPAGAPGDKGESGPSGPAGPTGAR
GAPGDRGEPGPPGPAGFAGPPGADGQPGAKGEPGDAGAKGDAGPPGPAGPAGPPGPIGNVG
APGAKGARGSAGPPGATGFPGAAGRVGPPGPSGNAGPPGPPGPAGKEGGKGPRGETGPAGR
PGEVGPPGPPGPAGEKGSPGADGPAGAPGTPGPQGIAGQRGVVGLPGQRGERGFPGLPGPSG
EPGKQGPSGASGERGPPGPMGPPGLAGPPGESGREGAPGAEGSPGRDGSPGAKGDRGETGP
AGPPGAPGAPGAPGPVGPAGKSGDRGETGPAGPAGPVGPVGARGPAGPQGPRGDKGETGEQ
GDRGIKGHRGFSGLQGPPGPPGSPGEQGPSGASGPAGPRGPPGSAGAPGKDGLNGLPGPIGP
PGPRGRTGDAGPVGPPGPPGPPGPPGPPSAGFDFSFLPQPPQEKAHDGGRYYRADDANVVRD
RDLEVDTTLKSLSQQIENIRSPEGSRKNPARTCRDLKMCHSDWKSGEYWIDPNQGCNLDAIK

VFCNMETGETCVYPTQPSVAQKNWYISKNPKDKRHVWFGESMTDGFQFEYGGQGSDPADVA
IQLTFLRLMSTEASQNITYHCKNSVAYMDQQTGNLKKALLLQGSNEIEIRAEGNSRFTYSVTVD
GCTSHTGAWGKTVIEYKTTKTSRLRIIDVAPLDVGAPDQEFGFDVGHVCFL

>M17398.txt
MEPFVVLVLCLSFMLLFSLWRQSCRRRKLPPGPTPLPIIGNMLQIDVKDICKSFTNFSKVYGPVF
TVYFGMNPIVVFHGYEAVKEALIDNGEEFSGRGNSPISQRITKGLGIISSNGKRWKEIRRFSLT
NLRNFGMGKRSIEDRVQEEAHCLVEELRKTKASPCDPTFILGCAPCNVICSVVFQKRFDYKDQ
NFLTLMKRFNENFRILNSPWIQVCNNFPLLIDCFPGTHNKVLKNVALTRSYIREKVKEHQASLD
VNNPRDFMDCFLIKMEQEKDNQKSEFNIENLVGTVADLFVAGTETTSTTLRYGLLLLLKHPEVT
AKVQEEIDHVIGRHRSPCMQDRSHMPYTDAVVHEIQRYSDLVPTGVPHAVTTDTKFRNYLIPK
SFDNKIMLA

>NM0003972.txt
MAVSRLDRLFILLDTGTTPVTRKAAAQQLGEVVKLHPHELNNLLSKVLIYLRSANWDTRIAAGQ
AVEAIVKNVPEWNPVPRTRQEPTSESSMEDSPTTERLNFDRFDICRLLQHGASLLGSAGAEFEV
QDEKSGEVDPKERIARQRKLLQKKLGLNMGEAIGMSTEELFNDEDLDYTPTSASFVNKQPTLQ
AAELIDSEFRAGMSNRQKNKAKRMAKLFAKQRSRDAVETNEKSNDSTDGEPEEKRRKIANVVI
NQSANDSKVLIDNIPDSSSLIEETNEWPLESFCEELCNDLFNPSWEVRHGAGTGLREILKAHG
KSGGKMGDSTLEEMIQQHQEWLEDLVIRLLCVFALDRFGDFVSDEVVAPVRETCAQTLGVVLK
HMNETGVHKTVDVLLKLLTQEQWEVRHGGLLGIKYALAVRQDVINTLLPKVLTRIIEGLQDLDD
DVRAVAAASLVPVVESLVYLQTQKVPFIINTLWDALLELDDLTASTNSIMTLLSSLLTYPQVQQC
SIQQSLTVLVPRVWPFLHHTISSVRRAALETLFTLLSTQDQNSSSWLIPILPDMLRHIFQFCVLE
SSQEILDLIHKVWMELLSKASVQYVVAAACPWMGAWLCLMMQPSHLPIDLNMLLEVKARAKE
KTGGKVRQGQSQNKEVLQEYIAGADTIMEDPATRDFVVMRARMMAAKLLGALCCCICDPGVN
VVTQEIKPAESLGQLLLFHLNSKSALQRISVALVICEWAALQKECKAVTLAVQPRLLDILSEHLY
YDEIAVPFTRMQNECKQLISSLADVHIEVGNRVNNNVLTIDQASDLVTTVFNEATSSFDLNPQV
LQQLDSKRQQVQMTVTETNQEWQVLQLRVHTFAACAVVSLQQLPEKLNPIIKPLMETIKKEEN
TLVQNYAAQCIAKLLQQCTTRTPCPNSKIIKNLCSSLCVDPYLTPCVTCPVPTQSGQENSKGST
SEKDGMHHTVTKHRGIITLYRHQKAAFAITSRRGPTPKAVKAQIADLPAGSSGNILVELDEAQK
PYLVQRRGAEFALTTIVKHFGGEMAVKLPHLWDAMVGPLRNTIDINNFDGKSLLDKGDSPAQE
LVNSLQVFETAAASMDSELHPLLVQHLPHLYMCLQYPSTAVRHMAARCVGVMSKIATMETMNI
FLEKVLPWLGAIDDSVKQEGAIEALACVMEQLDVGIVPYIVLLVVPVLGRMSDQTDSVRFMAT
QCFATLIRLMPLEAGIPDPPNMSAELIQLKAKERHFLEQLLDGKKLENYKIPVPINAELRKYQQD
GVNWLAFLNKYKLHGILCDDMGLGKTLQSICILAGDHCHRAQEYARSKLAECMPLPSLVVCPP
TLTGHWVDEVGKFCSREYLNPLHYTGPPTERIRLQHQVKRHNLIVASYDVVRNDIDFFRNIKFN
YCILDEGHVIKNGKTKLSKAVKQLTANYRIILSGTPIQNNVLELWSLFDFLMPGFLGTERQFAAR
YGKPILASRDARSSSREQEAGVLAMDALHRQVLPFLLRRMKEDVLQDLPPKIIQDYYCTLSPLQ
VQLYEDFAKSRAKCDVDETVSSATLSEETEKPKLKATGHVFQALQYLRKLCNHPALVLTPQHPE
FKTTAEKLAVQNSSLHDIQHAPKLSALKQLLLDCGLGNGSTSESGTESVVAQHRILIFCQLKSM
LDIVEHDLLKPHLPSVTYLRLDGSIPPGQRHSIVSRFNNDPSIDVLLLTTHVGGLGLNLTGADTV
VFVEHDWNPMRDLQAMDRAHRIGQKRVVNVYRLITRGTLEEKIMGLQKFKMNIANTVISQEN
SSLQSMGTDQLLDLFTLDKDGKAEKADTSTSGKASMKSILENLSDLWDQEQYDSEYSLENFM
HSLK

>NM000729.txt
MNSGVCLCVLMAVLAAGALTQPVPPADPAGSGLQRAEEAPRRQLRVSQRTDGESRAHLGALL
ARYIQQARKAPSGRMSIVKNLQNLDPSHRISDRDYMGWMDFGRRSAEEYEYPS

>NM001895.txt

MSGPVPSRARVYTDVNTHRPREYWDYESHVVEWGNQDDYQLVRKLGRGKYSEVFEAINITNN
EKVVVKILKPVKKKKIKREIKILENLRGGPNIITLADIVKDPVSRTPALVFEHVNNTDFKQLYQTL
TDYDIRFYMYEILKALDYCHSMGIMHRDVKPHNVMIDHEHRKLRLIDWGLAEFYHPGQEYNVR
VASRYFKGPELLVDYQMYDYSLDMWSLGCMLASMIFRKEPFFHGHDNYDQLVRIAKVLGTEDL
YDYIDKYNIELDPRFNDILGRHSRKRWERFVHSENQHLVSPEALDFLDKLLRYDHQSRLTAREA
MEHPYFYTVVKDQARMGSSSMPGGSTPVSSANMMSGISSVPTPSPLGPLAGSPVIAAANPLG
MPVPAAAGAQQ

>NM004028.txt
MVAFKGVWTQAFWKAVTAEFLAMLIFVLLSLGSTINWGGTEKPLPVDMVLISLCFGLSIATMVQ
CFGHISGGHINPAVTVAMVCTRKISIAKSVFYIAAQCLGAIIGAGILYLVTPPSVVGGLGVTMVH
GNLTAGHGLLVELIITFQLVFTIFASCDSKRTDVTGSIALAIGFSVAIGHLFAINYTGASMNPARS
FGPAVIMGNWENHWIYWVGPIIGAVLAGGLYEYVFCPDVEFKRRFKEAFSKAAQQTKGSYMEV
EDNRSQVETDDLILKPGVVHVIDVDRGEEKKGKDQSGEVLSSV

>NM005143.txt
MSALGAVIALLLWGQLFAVDSGNDVTDIADDGCPKPPEIAHGYVEHSVRYQCKNYYKLRTEGD
GVYTLNDKKQWINKAVGDKLPECEADDGCPKPPEIAHGYVEHSVRYQCKNYYKLRTEGDGVY
TLNNEKQWINKAVGDKLPECEAVCGKPKNPANPVQRILGGHLDAKGSFPWQAKMVSHHNLTT
GATLINEQWLLTTAKNLFLNHSENATAKDIAPTLTLYVGKKQLVEIEKVVLHPNYSQVDIGLIKL
KQKVSVNERVMPICLPSKDYAEVGRVGYVSGWGRNANFKFTDHLKYVMLPVADQDQCIRHYE
GSTVPEKKTPKSPVGVQPILNEHTFCAGMSKYQEDTCYGDAGSAFAVHDLEEDTWYATGILSF
DKSCAVAEYGVYVKVTSIQDWVQKTIAEN

>NM018684.txt
MADEQEIMCKLESIKEIRNKTLQMEKIKARLKAEFEALESEERHLKEYKQEMDLLLQEKMAHVE
ELRLIHADINVMENTIKQSENDLNKLLESTRRLHDEYKPLKEHVDALRMTLGLQRLPDLCEEEEK
LSLDYFEKQKAEWQTEPQEPPIPESLAAAAAAAQQLQVARKQDTRQTATFRQQPPPMKACLSC
HQQIHRNAPICPLCKAKSRSRNPKKPKRKQDE

>NM024596.txt
MNEHLSSLIKKKRKCMQPKDFNFKTPENDKRFQKKFEKMAKELQRQKTNLDDDVPILLFESNG
SLIYTPTIEINSSHHSAMEKRLQEMKEKRENLSPTSSQMIQQSHDNPSNSLCEAPLNISRDTLC
SDEYFAGGLHSSFDDLCGNSGCGNQERKLEGSINDIKSDVCISSLVLKANNIHSSPSFTHLDK
SSPQKFLSNLSKEEINLQRNIAGKVVTPHQKQAAGMSQETFEEKYRLSPTLSSTKGHLLIHSRP
RSSSVKRKRVSHGSHSPPKEKCKRKRSTRRSIMPRLQLCRSEGRLQHVAGPALEALSCGESSY
DDYFSPDNLKERYSENLPPESQLPSSPAQLSCRSLSKKERTSIFEMSDFSCVGKKTRTVDITNF
TAKTISSPRKTGNGEGRATSSCVTSAPEEALRCCRQAGKEDACPEGNGFSYTIEDPALPKGHD
DDLTPLEGSLEEMKEAVGLKSTQNKGTTSKISNSSEGEAQSEHEPCFIVDCNMETSTEEKENLP
GGYSGSVKNRPTRHDVLDDSCDGFKDLIKPHEELKKSGRGKKPTRTLVMTSMPSEKQNVVIQ
VVDKLKGFSIAPDVCETTTHVLSGKPLRTLNVLLGIARGCWVLSYDWVLWSLELGHWISEEPF
ELSHHFPAAPLCRSECHLSAGPYRGTLFADQPVMFVSPASSPPVAKLCELVHLCGGRVSQVPR
QASIVIGPYSGKKKATVKYLSEKWVLDSITQHKVCAPENYLLSQ

>O76273 .txt
MKGISKILSASIMVMKLGNVYSAVPLCSNTYDPSQQQPSYVLIPSTPEAITNCAYSPKNAYVPS
SPTTSSSTPGTNNDNETSPTTEDVGTCKISVVKHCDTPGASSTPCEPEQTIPAQPVTMATVTPA
IIASVQTPSVVSVIPVTQKVIQPATMIVPPSSIIPGYYPNGTPAAPGQQGQILSGSVLAPGASSC
QLVPGNTPGQMLPGMTPGVSPCLPTQGGDGSNQTIPGIVYPCQPGQGGSGSNQTIPGVISPC
QPGQGGSGSNQTIPGIVYPCQPGQGGSGSNQTIPGVISPCQPGQGGSGSNQTIPGIVYPCQP
GQNGDGSNQTIPGVISPCQPGQGGNGNGTTGQPGQCVSVPQTPNPIAMPPISGISGNGYPTS

TTYTQSLGQLGPCIDVQKPTSSCESQTNEKSTMQYAMEACAAPTPTVVIGNSEYLVGPGMYSS
LTSPCNSCCQC
>P62242.txt
MGISRDNWHKRRKTGGKRKPYHKKRKYELGRPAANTKIGPRRIHTVRVRGGNKKYRALRLDV
GNFSWGSECCTRKTRIIDVVYNASNNELVRTKTLVKNCIVLIDSTPYRQWYESHYALPLGRKKG
AKLTPEEEEILNKKRSKKIQKKYDERKKNAKISSLLEEQFQQGKLLACIASRPGQCGRADGYVL
EGKELEFYLRKIKARKGK

>P63210 .txt
MPVINIEDLTEKDKLKMEVDQLKKEVTLERMLVSKCCEEVRDYVEERSGEDPLVKGIPEDKNPF
KELKGGCVIS

>P63234.txt
MSLLTEVETYVLSIVPSGPLKAEIAQRLEDVFAGKNTDLEALMEWLKTRPILSPLTKGILGFVFTL
TVPSERGLQRRRFVQNALNGNGDPNNMDRAVKLYRKLKREITFHGAKEIALSYSAGALASCMG
LIYNRMGAVTTEVAFGLVCATCEQIADSQHRSHRQMVATTNPLIRHENRMVLASTTAKAMEQM
AGSSEQAAEAMEVASQARQMVQAMRAIGTHPSSSAGLKDDLLENLQAYQKRMGVQMQRFK

>P63274.txt
MGRVRTKTVKKAARVIIEKYYTRLGNDFHTNKRVCEEIAIIPSKKLRNKIAGYVTHLMKRIQRGP
VRGISIKLQEEERERRDNYVPEVSALDQEIIEVDPDTKEMLKLLDFGSLSNLQVTQPTVGMNFK
TPRGAV

>P63289 .txt
HSDAVFTDNYTRLRKQMAVKKYLNSILN

>P63302.txt
MALAVRVVYCGACGYKSKYLQLKKKLEDEFPGRLDICGEGTPQATGFFEVMVAGKLIHSKKKG
DGYVDTESKFLKLVAAIKAALAQG

>P67778.txt
MAAKVFESIGKFGLALAVAGGVVNSALYNVDAGHRAVIFDRFRGVQDIVVGEGTHFLIPWVQK
PIIFDCRSRPRNVPVITGSKDLQNVNITLRILFRPVASQLPRIYTSIGEDYDERVLPSITTEILKSV
VARFDAGELITQRELVSRQVSDDLTERAATFGLILDDVSLTHLTFGKEFTEAVEAKQVAQQEAE
RARFVVEKAEQQKKAAIISAEGDSKAAELIANSLATAGDGLIELRKLEAAEDIAYQLSRSRNITY
LPAGQSVLLQLPQ

>P67780.txt
MAYPFQLGLQDATSPIMEELLHFHDHTLMIVFLISSLVLYIISLMLTTKLTHTSTMDAQEVETVW
TILPAIILILIALPSLRILYMMDEINNPSLTVKTMGHQWYWSYEYTDYEDLNFDSYMIPTQELKPG
ELRLLEVDNRVVLPMEMTIRMLISSEDVLHSWAVPSLGLKTDAIPGRLNQTTLMAMRPGLYYGQ
CSEICGSNHSFMPIVLEMVPLSYFETWSALMV

>P67809.txt
MSSEAETQQPPAAPPAAPALSAADTKPGTTGSGAGSGGPGGLTSAAPAGGDKKVIATKVLGTV
KWFNVRNGYGFINRNDTKEDVFVHQTAIKKNNPRKYLRSVGDGETVEFDVVEGEKGAEAANV
TGPGGVPVQGSKYAADRNHYRRYPRRRGPPRNYQQNYQNSESGEKNEGSESAPEGQAQQRR
PYRRRRFPPYYMRRPYGRRPQYSNPPVQGEVMEGADNQGAGEQGRPVRQNMYRGYRPRFRR
GPPRQRQPREDGNEEDKENQGDETQGQQPPQRRYRRNFNYRRRRPENPKPQDGKETKAADP
PAENSSAPEAEQGGAE

>P67811.txt
MLSLDFLDDVRRMNKRQLYYQVLNFGMIVSSALMIWKGLMVITGSESPIVVVLSGSMEPAFHR
GDLLFLTNRVEDPIRVGEIVVFRIEGREIPIVHRVLKIHEKQNGHIKFLTKGDNNAVDDRGLYKQ
GQHWLEKKDVVGRARGFVPYIGIVTILMNDYPKFKYAVLFLLGLFVLVHRE

>P67997.txt
MANLGCWMLVLFVATWSDLGLCKKRPKPGGWNTGGSRYPGQGSPGGNRYPPQGGGGWGQ
PHGGGWGQPHGGGWGQPHGGGWGQPHGGGWGQGGGTHNQWHKPSKPKTSMKHMAGA
AAAGAVVGGLGGYMLGSAMSRPLIHFGNDYEDRYYRENMYRYPNQVYYRPVDQYSNQNNFV
HDCVNITIKQHTVTTTTKGENFTETDVKMMERVVEQMCITQYEKESQAYYQRGSSMVLFSSPP
VILLISFLIFLIVG

>P68037.txt
MAASRRLMKELEEIRKCGMKNFRNIQVDEANLLTWQGLIVPDNPPYDKGAFRIEINFPAEYPFK
PPKITFKTKIYHPNIDEKGQVCLPVISAENWKPATKTDQVIQSLIALVNDPQPEHPLRADLAEEY
SKDRKKFCKNAEEFTKKYGEKRPVD

>Q6A548.txt
MPRAPRCRAVRALLRGRYREVLPLATFLRRLGPPGRLLVRRGDPAAFRALVAQCLVCVPWGARP
PPAAPCFRQVSCLKELVARVVQRLCERGARNVLAFGFALLDGARGGPPVAFTTSVRSYLPNTVT
ETLRGSGAWGLLLRRVGDDVLTHLLARCALYLLVAPSCAYQVCGPPLYDLCAPASLPLPAPGLP
GLPGLPGLGAGAGASADLRPTRQAQNSGARRRRGSPGSGVPLAKRPRRSVASEPERGAHRSF
PRAQQPPVSEAPAVTPAVAASPAASWEGGPPGTRPTTPAWHPYPGPQGVPHDPAHPETKRFLY
CSGGRERLRPSFLLSALPPTLSGARKLVETIFLGSAPQKPGAARRMRRLPARYWRMRPLFQELL
GNHARCPYRALLRTHCPLRAMAAKEGSGNQAHRGVGICPLERPVAAPQEQTDSTRLVQLLRQ
HSSPWQVYAFLRACLCWLVPTGLWGSRHNQRRFLRNVKKFISLGKHAKLSLQELTWKMKVRD
CTWLHGNPGACCVPAAEHRRREEILARFLVLVDGHIYVVKLLRSFFYVTETTFQKNRLFFYRKSV
WSQLQSIGIRQLFNSVHLRELSEAEVRRHREARPALLTSRLRFLPKPSGLRPIVNMDYIMGART
FHRDKKVQHLTSQLKTLFSVLNYERARRPSLLGASMLGMDDIHRAWRTFVLRIRAQNPAPQLY
FVKVDVTGAYDALPQDRLVEVIANVIRPQESTYCVRHYAVVQRTARGHVRKAFKRHVSTFADL
QPYMRQFVERLQETSLLRDAVVIEQSSSLNEAGSSLFHLFLRLVHNHVVRIGGKSYIQCQGVPQ
GSILSTLLCSLCYGDMERRLFPGIEQDGVLLRLVDDFLLVTPHLTQAQAFLRTLVKGVPEYGCRA
NLQKTAVNFPVEDGALGSAAPLQLPAHCLFPWCGLLLDTRTLEVSCDYSSYAHTSIRASLTFSQ
GAKPGRNMRRKLLAVLRLKCCALFLDLQVNGIHTVYMNVYKIFLLQAYRFHACVLQLPFNQPVR
KNPSFFLRVIADTASCCYSLLKARNAGLSLGAKGASGLFPSEAARWLCLHAFLLKLAHHSGTYR
CLLGALQAAKAHLSRQLPRGTLAALEAAADPSLTADFKTILD

>Q8EUC5 .txt
MAIKSVLYDKYKKDVTKSLMKEFNYKSTMEIPRLEKIVINSGLGDATSDSKIVEIGLKELHLITG
QKPIATKSKKSIATFKLREGQAIGAKVTLRRENMWNFLTKLISIAIPRIRDFRGLSTKSFDGNGN
YTFGIKEQIIFPEIVYDDVKKLRGFDITIVTSAKTDKEAMFLLKELGMPFVKTKEAK

>Q8EUK2 .txt
MAVPQRKVTHSRKAKRGSHLHLSIPTLVACKRCGKKITPHRVCNSCGYYKNKKVPQIEA

>Q8EUM5.txt
MENKNQKHNNEFHEKNQQSQKDNNNVKKENLHEDQSDLNDANFDDGGKKNKLKNDIKKLN
HLVDSLKNENNNKQAKIESLERQINLLNENFKSEVIKKASEAQTKLDEKIKEFQAKYETELKHA
KKYALKSSAIELIDIVSNFELAVNSKVTNPEIANYLKGFQMFANMFKNYFQQNGITEIPVNLNDD
FNAEVMQAFETQKAPNTQPNKVIKIIKKGYKLHDIVLVPATVIVSE

>Q8EW28.txt
MEFNFEPSNEKDILKKYSRNLNEEVSANKLNKIIGREQEIRRVIEILSRKEKNNPVLIGEPGVGK
TAIVEGFVQKIISKEVPENLVHCVVYEVNLSSLIAGTFLQGEFEKRLNALIKEAKQNNGAVILFID
EIHQLMGMGKAGNNSGMDAANILKPIMARGEIKIIGATTSNEYRQYIEKDGALERRFQKILVEE
PTPEEALTIMRGLKEKWEIYHKVRIQDNALVASVKLSERYISDKYLPDKAIDLIDEAAAKIKTEA
HTSPSEPINKKIFYLETEKIALSKEEGTNQKERINEIEIELEKLKQERDLVEKEWKEQKEQQVAL
NKIKKEIEKNNWDVERYQNQGEYTEASKILYSVLPELKKKLEAIEKSISENKKVLIKDYISEIDVA
EIISRITKIPLNKIFEKEQDKLLNLFNNLKKRVKGQDEALKLVSDTVLKNRVGINNPNRPIGSFLF
VGPTGVGKTEVAKSLAENLFNTEKAIVRINMSEYMEKHSISRLIGAPPGYIGYEQAGELSEQIRR
KPYSVVLLDEIEKAHPDILNVLLQVLDEGTLKDNQGRNINFKNTIIIMTSNVGALYLMENKEDMF
ERELKTSFKPEFLNRIDEIIKFNSITMEFAKEIAQKMLDDLEKRLKDNNYQITFDKSVVEYVAKN
GYSKEYGARPINRFIQKTIENFITESILKNELIKDRSTIINFANNKLAILKSN

>Q8EWR9.txt
MKIIKITPRGFCKGVVDAYATCKKIAKLYPNHEKYLIGWLVHNKEIIKELEELGIQTKDDKNHSR
SEIIDSIEIKDKNNPPIVIFSAHGTDQKTIDKAREKGLVVFDTTCIYVTKTHDLIKEKIEQGYQIFY
IGVNNHPETISTLSIDKSIILIETVNDIENIRTESEKPIFVTNQTTISIYEFEEITETLKSKYKNIEFK
NDICNAAKDRQDAVINMPSEVDLLLVVGDIKSNNSKKLVEIGIKKQIESHLIWNTKNIKDEWFI
NKKCLAITSGCSTPTWLANYVIIFLEKKLGTAND

>Q8EX21 .txt
MATIAQLIRHDRKDKFKKSKSPALMYTYNSLKKKRTYNPSPYKRGVCTRVGTMTPKKPNSALR
KYAKVKLTNGYEVLAYIPGEGHNLQEHSVVLIEGGRVKDLPGVRYHIVRGTLDTSGVEKRRQQ
RSGYGAKRPKEKKE

>Q8EX25.txt
MLLVAFKDPRITRIAKVNLIGGQAKPGPQLASIGINMGEFTKQFNDQTKDKNGEVIPCIITAFKD
KSFTFEIKTTPVTMLLKKAANIKVGAKNSKTETVATISREKALEIAKTKLVDTNANDEEAVLRMV
AGSAKQMGIKIEGVDPVVHKDGKKK

>Q8MKI5.txt
MENTENSVDAKSFKNAETKILHGSKSMDSGMSFDNSYKMDYPEMGLCIIINNKNFHKSTGMA
PRSGTDVDAANLRETFTNLKYEVRNKNDLTCEEILELMNSVSKEDHSKRSSFVCVLLSHGDEGI
IFGTNGPVDLRKVTGFFRGDYCRSLTGKPKLFIIQACRGTELDCGIETDSGIEDDMACQKIPVE
ADFLYAYSTAPGYYSWRNSKDGSWFIQSLCAMLKLYAHKLEFMHILTRVNRKVATEFESFSLD
SAFHGKKQIPCIVSMLTKELYLYH

>Q8NB59.txt
MAIEGGERTCGVHELICIRKVSPEAVGFLSAVGVFIILMLLLFLYINKKFCFENVGGFPDLGSEYS
TRKNSQDKIYNSYMDKDEHGSSSESEDEALGKYHEALSRTHNSRLPLADSRQRNYAWETRQK
YSPLSAEYDGYSSEASIDEGNCIQRMRRTPPLDELQPPPYQDDSGSPHLSCTPSEIGDSKCEFS
HCSNSPRCSYNKCPSEGSTGHEIESFHNKGYEEDVPSDSTAVLSPEDMSAQGSSSQLPKPFDP
EPEAKYGTLDVTFDYDSQEQKLLVTVTAVTDIPTYNRTGGNSWQVHLVLLPIKKQRAKTSIQRG
PCPVFTETFKFNHVESEMIGNYAVRFRLYGVHRMKKEKIVGEKIFYLTKLNLQGKMSLPVILEPS
YNHSGCDSQMSVSEMSCSESTSSCQSLEHGSVPEILIGLLYNATTGRLSAEVIKGSHFKNLAA
NRPPNTYVKLTLLNSMGQEMSKCKTSIRRGQPNPVYKETFVFQVALFQLSDVTLILSVYNKRSM
KREEMIGWISLGLNSSGEEELNHWTEMKESKGQQVCRWHALLES

>Q99J21.txt
MATPAGRRASETERLLTPNPGYGTQVGTSPAPTTPTEEEDLRRRLKYFFMSPCDKFRAKGRKPC
KLMLQVVKILVVTVQLILFGLSNQLVVTFREENTIAFRHLFLLGYSDGSDDTFAAYTQEQLYQAI

FYAVDQYLILPEISLGRYAYVRGGGGPWANGSALALCQRYYHRGHVDPANDTFDIDPRVVTDC
IQVDPPDRPPDIPSEDLDFLDGSASYKNLTLKFHKLINVTIHFQLKTINLQSLINNEIPDCYTFSIL
ITFDNKAHSGRIPIRLETKTHIQECKHPSVSRHGDNSFRLLFDVVVILTCSLSFLLCARSLLRGFL
LQNEFVVFMWRRRGREISLWERLEFVNGWYILLVTSDVLTISGTVMKIGIEAKNLASYDVCSIL
LGTSTLLVWVGVIRYLTFFHKYNILIATLRVALPSVMRFCCCVAVIYLGYCFCGWIVLGPYHVKF
RSLSMVSECLFSLINGDDMFVTFAAMQAQQGHSSLVWLFSQLYLYSFISLFIYMVLSLFIALITG
AYDTIKHPGGTGTEKSELQAYIEQCQDSPTSGKFRRGSGSACSLFCCCGRDSPEDHSLLVN//

>Q9NUX5.txt
MSLVPATNYIYTPLNQLKGGTIVNVYGVVKFFKPPYLSKGTDYCSVVTIVDQTNVKLTCLLFSGN
YEALPIIYKNGDIVRFHRLKIQVYKKETQGITSSGFASLTFEGTLGAPIIPRTSSKYFNFTTEDHK
MVEALRVWASTHMSPSWTLLKLCDVQPMQYFDLTCQLLGKAEVDGASFLLKVWDGTRTPFPS
WRVLIQDLVLEGDLSHIHRLQNLTIDILVYDNHVHVARSLKVGSFLRIYSLHTKLQSMNSENQT
MLSLEFHLHGGTSYGRGIRVLPESNSDVDQLKKDLESANLTANQHSDVICQSEPDDSFPSSGS
VSLYEVERCQQLSATILTDHQYLERTPLCAILKQKAPQQYRIRAKLRSYKPRRLFQSVKLHCPKC
HLLQEVPHEGDLDIIFQDGATKTPDVKLQNTSLYDSKIWTTKNQKGRKVAVHFVKNNGILPLS
NECLLLIEGGTLSEICKLSNKFNSVIPVRSGHEDLELLDLSAPFLIQGTIHHYGCKQCSSLRSIQN
LNSLVDKTSWIPSSVAEALGIVPLQYVFVMTFTLDDGTGVLEAYLMDSDKFFQIPASEVLMDDD
LQKSVDMIMDMFCPPGIKIDAYPWLECFIKSYNVTNGTDNQICYQIFDTTVAEDVI

>Q9NWW6.txt
MKTFIIGISGVTNSGKTTLAKNLQKHLPNCSVISQDDFFKPESEIETDKNGFLQYDVLEALNME
KMMSAISCWMESARHSVVSTDQESAEEIPILIIEGFLLFNYKPLDTIWNRSYFLTIPYEECKRRR
STRVYQPPDSPGYFDGHVWPMYLKYRQEMQDITWEVVYLDGTKSEEDLFLQVYEDLIQELAKQ
KCLQVTA

>Q9NXK6.txt
MLSLKLPRLFSIDQIPQVFHEQGILFGYRHPQSSATACILSLFQMTNETLNIWTHLLPFWFFAWR
FVTALYMTDIKNDSYSWPMLVYMCTSCVYPLVSSCAHTFSSMSKNARHICYFLDYGAVNLFSL
GSAIAYSAYTFPDALMCTTFHDYYVALAVLNTILSTGLSCYSRFLEVQKPRLCKVIRVLAFAYPYT
WDSLPIFYRLFLFPGESAQNEATSYHQKHMIMTLLASFLYSAHLPERLAPGRFDYIGHSHQLFH
VCVILATHMQMEAILLDKTLRKEWLLATSKPFSFSQIAGAILLCIIFSLSNIIYFSAALYRIPKPEL
HKKET

>Q9UKH8.txt
MGQTKSKIKSKYASYLSFIKILLKRGGVKVSTKNLIKLFQIIEQFCPWFPEQGTLDLKDWKRIGK
ELKQAGRKGNIIPLTVWNDWAIIKAALEPFQTEEDSVSVSDAPGSCLIDCNEKTRKKSQKEME
GLHCEYVAEPVMAQSTQNVDYNQLQEVIYPETLKLEGKGPELVGPLESKP

>Q9Y3S2.txt
MPKKKTGARKKAENRREREKQLRASRSTIDLAKHPCNASMECDKCQRRQKNRAFCYFCNSVQ
KLPICAQCGKTKCMMKSSDCVIKHAGVYSTGLAMVGAICDFCEAWVCHGRKCLSTHACACPL
TDAECVECERGVWDHGGRIFSCSFCHNFLCEDDQFEHQASCQVLEAETFKCVSCNRLGQHSC
LRCKACFCDDHTRSKVFKQEKGKQPPCPKCGHETQETKDLSMSTRSLKFGRQTGGEEGDGAS
GYDAYWKNLSSDKYGDTSYHDEEEDEYEAEDDEEEEDEGRKDSDTESSDLFTNLNLGRTYAS
GYAHYEEQEN

>Q9Y6I0.txt
WASQVSENRPVCKAIIQGKQFEGLVDTGADVSIIALNQWPKNWPKQKAVTGLVGIGTASEVY
QSTEILHCLGPDNQESTVQPMITSIPLNLWGRDLLQQWGAEITMPAPSYSPTSQKIMTKMGYIP
GKGLGKNEDGIKIPVEAKINQEREGIGNPC

>protmutt1.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt10.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKT

>protmutt11.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTX

>protmutt1176.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEGSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt1179.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEESSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt1181.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDTSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt1184.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDYSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP

QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt1187.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt1192.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSIGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt12.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt13.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKT

>protmutt14.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTY

>protmutt15.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYLGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt1501.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA

PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt1505.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPSGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt1510.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPRSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt1540.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt16.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYHGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt17.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYHGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt18.txt

MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQRSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt19.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQCSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt2.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPS

>protmutt20.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQVSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt21.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt22.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGGYGFRLGFLHSGTAKSVTCTY
SPALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGL
APPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPI
LTIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSS
PQPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTS
RHKKLMFKTEGPDSD

>protmutt23.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGRYGFRLGFLHSGTAKSVTCTYS

PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt24.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGS

>protmutt25.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGS

>protmutt26.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt27.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYSFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt28.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYDFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt29.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt3.txt

MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSRKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt30.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGLRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt31.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGSRLGFLHSGTAKSVTCTY
SPALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGL
APPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPI
LTIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSS
PQPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTS
RHKKLMFKTEGPDSD

>protmutt32.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGCRLGFLHSGTAKSVTCTY
SPALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGL
APPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPI
LTIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSS
PQPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTS
RHKKLMFKTEGPDSD

>protmutt33.txt
MEEPQSDPSVKPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt4.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP

QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt5.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQ

>protmutt6.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQRTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt7.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQNTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt740.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDVRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt742.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDERNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPILT
IITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSPQ
PKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSRH
KKLMFKTEGPDSD

>protmutt745.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDD

>protmutt748.txt

MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDINTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPILT
IITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSPQ
PKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSRH
KKLMFKTEGPDSD

>protmutt750.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRHTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt753.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRITFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPILT
IITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSPQ
PKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSRH
KKLMFKTEGPDSD

>protmutt755.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNAFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt8.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKIYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP
QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

>protmutt9.txt
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPRVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTYS
PALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSSP

QPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSR
HKKLMFKTEGPDSD

BIBLIOGRAFIE

[AHO'74] **Aho, V.A., Hopcroft, J.E., Ullman, J.D.,:** „ *The design and Analysis of Computer Algorithms*", Addison-Wesley, Menlo Park, California, 1974;

[ALTS'90] **Altschul, S., Gish, W., Miller, W., Myers, E., Lipman, D.J.,:** "*Basic local alignment search tool*", J. Mol. Biol., vol. 215, no.3, pp:403-410, 1990.

[ALTS'97] **Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Y., Miller, W., Lipman, D.,:** "*Gapped Blast and PSI-Blast: a new generation of protein database search programs*", Nucleic Acids Research, vol.25, no.17, pp: 3389-3402 1997.

[ANAS'03] **Anastasiadis, A.D., Magoulas, G.D., Liu, X.,:***"Classification of Protein Localisation Patterns via Supervised Neural Network Learning"*, Proceedings of the Fifth Symposium on Intelligent Data Analysis (IDA-03), Berlin, Germany, 2003.

[ARRA'84a] **Arratia, R.A.,and Watterman, M.S.,:** "An Erdos-Renyi Low with Shifts", Adv. Appll. Math.(in press).

[ARRA'84b] **Arratia, R.A., Gordon, L., and Watterman, M.S.,:** "An Extreme Value Distribution for Sequence Matching", manuscript, 1984.

[ARSL'01] **Arslan, N.A.,** Egecioglu Ö., Pevzner, P.A.: ***"A new approach to sequence comparison: normalized sequence alignment"***, Bioinformatics, vol.17 no.4 , pp: 327-337, 2001.

[BACO'86] **Bacon, D.J., and Anderson, W.F.:** "*Multiple sequence alignment*", J. Mol.Biol. vol. 191, pg: 153-161, 1986.

[BAEZ'99] Baeza-Yates, R., Ribeiro-Neto, B., in "Retrieval Evaluation", *Modern Information Retrieval*, Ed. Addison Wesley, 1999, pp:75-81.

[BAIL'94] **Bailey, T.L., Elkan C.,:** "*Fitting a mixture model by expectation maximization to discover motifs in biopolymers*", Proceedings of International Conference on Intelligent Systems for Molecular Biology, PubMed, 2:28-36, 1994.

[BALD'94] **Baldi, P., Chauvin, Y., Hunkapiller, T., McClure, M.A.,:** *"Hidden Markov models of biological primary sequence information"*, Proc. Natl.Acad.Sci.USA 91(3):1059-1036, 1994.

[BART'96] **Barton, G.J.:** *"Protein Sequence Alignment and Database Scanning"* in M.J. E. Sternberg, Protein Structure Prediction – a practical approach, IRL Press at Oxford University Press, 1996.

[BASH'87] **Bashford, D., Chothia, C. and Lesk, A.M.:** *"Determinants of a protein fold: Unique features of the globin amino acid sequences",* J.Mol.Biol. 196:199-216, 1987.

[BATE'04] **Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E.L.,, Studholme, D.J., Yeats, C., Eddy, S.R.,:** *"The Pfam protein families database",*Nucleic Acids Research, 32, 2004.

[BELL'00] **Bellagarda, J.,:** *"Exploitiong latent semantic information in statistical language modeling,"* Proceedings IEEE, vol. 88, no.8, pp:1279-1296, 2000.

[BENS'99] **Benson, G.,:** "Tandem repeats finder: a program to analyze DNA sequences", *Nucleic Acids Res.*, 27(2):573-80, 1999.

[BERG'85] **Berger, J.,:** *"Statistical Decision Theory and Bayesian Analysis"*, Springer-Verlag, New York, 1985.

[BIG'05] **BioInformatics Glossary**, http://big.mcw.edu/display.php/2241.html

[BLAN'00] **Blanchette, M., Schwikowski, B., Tompa, M.,:** *"An exact algorithm to identify motifs in orthologous seqencesfrom multiple species"*, Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology(ISMB),: 37-45, 2000.

[BOGA'03a] **Bogan-Marta, A., Wambacq, P.:** *"From Latent Semantic Indexing to Text Categorization?"*, published in the Journal "Analele Universitatii Bucuresti", No. LII, sera Matematica- Informatica", vol. 1, 2003, ISSN 1224-7170, pg. 87-96.

[BOGA'03b] **Bogan-Marta,A., Robu, N.,:** *"Investigations Over ``add one'' Smoothing Method for Speech Recognition",* Proceedings of EMES'03, May 24-26, Oradea, Romania, pp.93-97.

[BOGA'03c] **Bogan-Marta,A., Robu,N., :** *"Indicators for Text Relevance Using Techniques from Information Theory Field"*, Proceedings of SINTES 11th Edition, Section Computer Science & Engineering, Craiova, Romania, 23-24 October 2003, ISBN 973–8043–415–5, ISBN 973–8043–415–5 pg. 286-290.

[BOGA'04a] **Bogan-Marta, A., Robu, N.,:** *"The Use of Statistical Models For Large Text Data Analysis"*, Analele Universitatii din Craiova, Seria Inginerie Electrica, no. 27, vol.1, ISSN 1223-530X, pp.87-92, 2004.

[BOGA'04b] **Bogan-Marta,A., Győrödi, R., Győrödi,C.,:** *"Implications of syntactic and semantic categories of grammars on a speech recognition tool"*, in Proceedings of The Sixth International Scientific Conference, ECI 2004, Kosice-Herľany, Slovakia, pp. 86-90, 2004.

[BOGA'05a] **Bogan-Marta,A., Gavrielides,M.A., Pitas, I., Lyroudia,K.,:** *"A New Statistical Measure of Protein Similarity based on Language Modeling"*, GENSIPS 05, IEEE International Workshop on Genomic Signal Processing and Statistics , Newport, Rhode Island, ISBN 9-812-38846-X, 2005.

[BOGA'05b] **Bogan-Marta,A., Laskaris,N., Gavrielides,M.A., Pitas, I., Lyroudia,K.,:** *"A novel efficient protein similarity measure based on n-gram modeling"*, CIMED 2005, IEEE, IEE, Second International Conference on Intelligence in Medicine and Healthcare, Costa da Caparica, Lisbon, Portugal, ISBN:0-86341-520-2, pp. 122-127,2005.

[BOGA'06c] **Bogan-Marta, A., Robu, N., Pater, M.,:** *"String comparison in terms of statistical evaluation applied on biological sequences"*, IEEE, Proceedings of ICCCC'06, Baile Felix Spa-Oradea, pp.86-91, 2006.

[BOGA'06d] **Bogan-Marta,A., Pitas, I., Lyroudia,K.,:** *"Statistical Method of Context Evaluation for Biological Sequence Similarity"*, IFIP 2006, Santiago de Chile, published in 'Artificial Intelligence in Theory and Practice', Springer , 2006.

[BOGA'06e] **Bogan-Marta, Robu,N.,:** *"A study of Sequence Clustering on Protein's Primary Structure using a Statistical Method"*, Acta Polytechnica Hungarica, Journal of Applied Sciences at Budapest Tech, Hungary, vol3, issue3, ISSN 1785-8860, pp.17-27, 2006.

[BOGA'06a] **Bogan-Marta,A., Robu,N.,:** *"Markow chains for a New Datamining Techique applied on Biological Sequences"*, CONTI'2006, The 7th International Conference on Technical Informatics, 8-9 June 2006, Timisoara, Romania, pp.227-230.

[BOGA'06b] **Bogan-Marta,A., Robu,N.,:** *"Clustering Sequences with a Statistical Content Evaluation Method"*, SACI2006, 3rd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence, Timişoara, Romania, May 25-26, 2006, pp.59-69.

[BORD'91] **Bordo, D., and Argos, P.,:** *"Suggestions for "safe" residue substitutions in site-directed mutagenesis."*, J Mol Biol., Feb 20; 217(4), pp:721-9, 1991.

[BOSW'84] **Boswell, D.R., and MacLachlan, A.D.,:** *"Sequence Comparison by Exponentially-damped Allignment"*, Nucleic Acid Res., 12, 457-467, 1984.

[BREJ'00] **Brejová, B.,** DiMarco, C., Vinař, T., Hidalgo, S.R., Holguin, G., Patten, C.,: *"Finding Patterns in Biological Sequences"*, Project report for CS798g, Fall 2000.

[BROW'92] **Brown, P.F., Della Pietra, A. S., Della Pietra, V.J., Mercer Robert, L.R., and Jennifer, C.L.,:** "An estimation of an upper bound for the entropy of English", in *Association for Computational Linguistics,* Yorktown Heights, NY 10598, P.O. Box 704, 1992.

[BYER'84] **Byers, T.H., and Waterman, M.S.,:** *"Determining All Optimal and Near-optimal Solutions When Solving Shortest Path Problems by Dynamic Programming"*, Operat. Res. (in press),1984.

[CAMO'03] **Camoglu, O., Kahveci, T., Singh, A.K.:** *"PSI: indexing protein structures for fast similarity search"*. PubMed, Bioinformatics, vol. 19 Suppl. 1, pp: i81-i83, 2003.

[CHIA'04] **Chiang, J-H., Yu, H-C., Hsu, H-J.,:** *"Gis: a biomedical text-mining system for gene information discovery"*, Bioinformatics, 20(1): 120-121, 2004.

[CLAV'93] **Claverie, J-M.,:** *"Detecting frame shifts by amino acid sequence comparison"*, J. Molecular Biology 234: 1140-1157, 1993.

[COHE'75] **Cohen, D.N., Reichert, T.A., and Wong, A.K.C.,:** *"Matching Code Sequences Utilizing Context Free Quality Measures"*, Math. Biosci., 24, pp.25-30, 1975.

[COLL'84] **Collins, J.F., and Coulson, A.F.W.,:** *"Applications of Paralell Processing Algorithms for DNA Sequence Analysis"*, Nucleic Acids Research., 12, 181-192,

[COWA'98] **Coward E., Drablos, F.,:** *"Detecting periodic patterns in biological sequences"*, Bioinformatics, 14: 498-507, 1998.

[DAVI'92] **Jones, D.T, Taylor, W.R., and Thornton, J.M.,:** *"The rapid generation of mutation data matrices from protein sequences"*, CABIOS 8: 275-282, 1992.

[DAYH'78] **Dayhoff, M.O.**: *"Atlas of Protein Sequence and Structure"*, Natl. Biomed. Res. Found., Washington, Vol. 5, Suppl. 3, pp. 345-352, 1978.

[DELC'75] **Delcoigne, A., and Hansen, P.,:** *"Sequence Comparison by Dynamic Programming"*, Biometrika, 62, 661-664, 1975.

[DEMA'95] **De Marcken, K., :** *"On the unsupervised induction of phrase-structure grammars"*, SIGDAT, 1995, http://www.demarcken.org/carl/papers/sigdat.pdf

[DESP'02] **M. Deshpande, M., Karypis, G.,:** *"Evaluation of techniques for Classifying Biological Sequences"*, Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, : 417 – 431, 2002.

[DINK'67] **Dinkelbach, W.,:** *"On nonlinear fractional programming,"* Management Science, vol. 18, no.7, pp: 492-498, 1967.

[DUDA'01] **Duda, R.O., Hart, P. E., Stork., D. G.,:** *" Pattern classification"*, Information Theory, John Wiley&Sons, Inc., 2nd edition, 2001.

[DUMA'82] **Dumas, J.P., and Nino, J.,:** *"Efficient Algorithms for Folding and Comparing Nucleic Acid Sequences"*, Nucleic Acids Res., 80, 197-206, 1082.

[DUME'56] **Dumey, A.I.,:** *" Indexing for Rapid Random-acces Memory"*, Comput. Automat., 5, 6-8, 1956.

[DURB'98] **Durbin, R., Eddy, S., Krogh, A., Mitchison, G.,:** *"Biological Sequence Analysis. Probabilistic models of proteins and nucleic acid,"* Cambridge University Press, 1998.

[EDGA'04] **Edgar, R.C.,:** *"MUSCLE: multiple sequence alignment with high accuracy and high throughput"*, Nucleic Acids Research 32(5), 1792-97, 2004.

[ERHA'80] **Erhan,S., Marzolf,T., Cohen,L., :** *"Amino-acid neighborhood relationships in proteins: breakdown of amino-acid sequences into overlapping doublets, triplets and quadruplets."* Int. J. Biomed Comput, vol. 11(1), pp:67-75, 1980.

[ERIC'83] **Erickson, B.W., and Sellers, P.H.,:** *"In Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison"*, Eds. D. Sankoff and J.B. Kruskal, Addison-Wesley, London, pp. 55-90, 1983.

[ESKI'01] **Eskin, E., Grundy, W.N., and Singer, Y.,:** *"Using mixtures of common ancestors for estimating the probabilities of discrete events in biological sequences,"* Bioinformatics, vol.17 Suppl 1, pp: 65-73, 2001.

[ESTE'04] **Ester, M., Zhang, X.,:** *"A Top-Down Method for Mining Most-Specific Frequent Patterns in Biological Sequences"*, Proceedings of the Fourth SIAM International Conference on Data Mining, Lake Buena Vista, Florida, USA, April 22-24, 2004.

[FICK'84] **Fickett, J.W.,:** *"Fast Optimal Alignment"*, Nucleic Acids Res., 12: 175-180, 1984.

[FITC'67] **Fitch, W.M., and Margoliash, E.,:** "Construction of Polygenetic Trees", Science, 155, 279-284, 1967.

[FITC'69] **Fitch, W.M.,:** *"Locating Gaps in Amino Acid Sequences to Optimize the Homology Between Two Proteins"*, Biochem. Genet., 3, 99, 1969.

[FITC'71] **Fitch, W.M.,:** *"Towards Defining the Course of Evolution: Minimum Change for a Specific Tree Topology"*, Syst.Zool.20, 406-416, 1971.

[FITC'83] **Fitch, W.M., and Smith, T.F.,:** *"Optimal Sequence Alignments"* , Proc., natn., Acad. Sci. U.S.A., 80, 1382-1386, 1983.

[FLOR'96] **Floroiu, C.,:** *"Evaluarea produselor software"*, arhiva revista Byte Romania, disponibil online la: http://www.byte.ro/byte96-05/eva.html, (accesibil la 1 Septembrie 2006).

[GANA'04] **Ganapathiraju, M.K., Klein-Seetharaman, J., Balakrishnan, N., and Reddy, R.,:** *"Characterization of protein secondary structure-application of latent semantic analysis using different vocabulary,"* IEEE Signal Processing Magazine, vol. 21, no.3, pp: 78-87, 2004.

[GANA'04b] **Ganapathiraju,M., Manoharan,V., and Klein-Seetharaman,J.,:** *"Statistical sequence analysis using n-grams"*, J. Appl. Bioinformatics, vol.3 (2), pp:193-200 , 2004

[GOAD'82] **Goad, W.B.,Kanehisa, M.I.,:** *"Pattern Recognition in Nucleic Acid Sequences I. A General Method for Finding Local Homologies and Symmetries"*, Nucleic Acid Res., 10: 247-263, 1982.

[GONN'92] **Gonnet, G.H., Cohen, M.A., Benner, S.A.,:** *"Exhaustive Matching of the Entire Protein Sequence Database"*, Science 256: 1443-1445, 1992.

[GORD'73] **Gordon, A.D.,:** *"A Sequence Comparison Statistic and Algorithm"*, Biometrika, 60, 197-200, 1973.

[GORO'97] **Gorodkin, J., Heyer, L. J., Stormo, G. D.,:** *"Finding the most significant common sequence and structure motifs in a set of RNA sequences"*, Nucleic Acids Research, 25(18): 3724-3732, 1997.

[GOTO'82] **Gotoh, O.,:** *"An inproved algorithm for matching biological sequences."* J. Mol. Biol. vol.162, pp:705-708, 1982.

[GRIBS'87] **Gribskov, M., McLachlan, A.D., and Eisenberg, D.,:** *„Profile analysis: detection of distantly related proteins."* Proceedings of the National Academy of Sciences of the USA, 84:4355-4358, 1987.

[HENI'92] **Henikoff, S., Henikoff, J.G.**: *"Amino acid substitution matrices from protein blocks"*, în Proceedings of Natural Academic Sciences, 89:10915--10919, 1992.

[HERT'90] **Hertz, G.Z., Hartzell, G.W.,III, Stormo, G.D.,:***"Identification of consensus patterns in unaligned DNA sequences known to be functionally related"*, Comput. Appl. Biosci.6,81-92.

[HIGG'94] **Higgins, D., Thompson J., Gibson, T., Thompson, J.D., Higgins, D.G., Gibson, T.J.,:** *"CLUSTAL W: improving the sensitivity of progressivemultiple sequence alignment through sequence weighting,position-specific gap penalties and weight matrix choice"*, Nucleic Acids Res. 22:4673-4680, 1994.

[HUGH'96] **Hughey, R.,and Krogh, A.,:** *"Hidden Markov models for sequence analysis:extension and analysis of the basic method."* Computer Applications in the Biosciences, 12: 95-107.

[JAIN'88] **Jain, A.K., Dubes, R.C.,:** *"Algorithms for Clustering Data"*, Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[JAKK'00] **Jaakkola, T., Diekhans, M., and Haussler, D.,:** *"A discriminative framework for detecting remote protein homologies, "* J. Computational Biology, vol. 7, pp: 95-114, 2000.

[JOHN'86] **Johnson, M.S., and Doolittle, R.F.,:** *"A method for the simultaneous alignment of three or more amino acid sequences,"* J. Mol. Evol., vol. 23, pp: 267-278, 1986.

[JURA'00] **Jurafsky, D., Martin, J. H.,**:*"Speech and Language Processing:An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition."* Prentice Hall, 1st edition, January 26, 2000.

[KALT'04] **Kaltchenko, A.,:** *"Algorithms for Estimating Information Distance with Application to Bioinformatics and Linguistics"*, The Computing Research Repository disponibilă online la : http://arxiv.org/abs/cs.CC/0404039 (la data de 8 Septembrie 2006), 2004.

[KANE'82] **Kanehisa, M.I., and Goad, W.B.,:** „Pattern Recognition in Nucleic Acid Sequences II. An Efficient Method for Finding Locally Stable Secondary Structures", Nucleic Acids Research, 10, 265-277, 1982.

[KARC'98] **Karchin R., Hughey, R.,:** *"Weighting hidden Markov models for maximum discrimination"*, Bioinformatics, 14(9): 772-782, 1998.

[KARL'83] **Karlin, S., Ghandour, G., Ost, F., Tavare, S., and Korn, L.J.:** *„New Approaches for Computer Analysis of Nucleic Acid Sequences"*, Proc. Natn. Acad. Sci. U.S.A., 80, 5660-5664, 1983.

[KARL'84] **Karlin, S., Ghandour, G., and Foulser, D.E.,:** „Comparative Analysis of Human and Bovine Papallimaviruses", Mol.Biol.Evol., 1, 357-370, 1984.

[KARL'91] **Karlin,S.,Bucher,P.,Brendel,V., and Altschul,S.F.,:** *"Statistical methods and insights for protein and DNA sequences"*, Annu Rev Biophis Biophys Chem., vol 20, 175-203, 1991.

[KARL'96] **Karlin, S.,Bruge,C.,:** *"Trinucleotide repeats and long homopeptides in genes and proteins associated with nervous system disease and development"*, Proc Natl Acad Sci USA, vol. 93(4), pp:1560-1565, 1996.

[KASI'99] **Kasif, S.,:** *"Datascope: Mining Biological Sequences"*, IEEE Intelligent Systems, : 38-43, Nov/Dec. 1999.

[KATT'00] **Katti, M.V., Sami-Subbu, R., Ranjekar, P.K., and Gupta, V.S.:** *"Amino acid repeat patterns in protein sequences: Their diversity and structural-functional implications". Protein Sci.,* vol. 9, pp: 1203-1209, 2000.

[KORN'77] **Korn, L.J., Queen, C.L., and Wegman M.N.,:** „Computer Analysis of Nucleic Acid Regulatory Sequences", Proc. Natn.Acad. sci. U.S.A., 74, 4401-4405, 1977.

[KRAS'04] **Krasnogor, N., Pelta, D.A.,:** *"Measuring the similarity of protein structures by means of the universal similarity metric,"* Bioinformatics Advance Access, vol. 20, pp: 1015-1021, 2004.

[KROE'04] **Kroese, D. P., Rubinstein, R.Y., Taimre, T.,:** *"Application of the Cross-Entropy Method to Clustering and Vector Quantization"*, Journal of Machine Learning Research, 2004.

[KROG'94] **Krogh, A., Brown, M., Mian, I., Sjolander, K., Haussler, D.,:** *"Hidden Markov models in computational biology: Application to protein modeling"*, J. Molecular Biology, 235:1501-1531, 1994.

[KRUS'83] **Kruskal, J.B.,and Sankoff, D.,:** *"In Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison"*, Eds. D. Sankoff and J.B. Kruskal, Addison-Wesley, London, 1983.

[LAND'98] **Landauer, T., Foltx, P., and Laham, D.,:** *"Introduction to latent semantic analysis,"* Discourse Processes, vol. 25, pp:259-284, 1998.

[LASK'02] **Laskaris, N.A., Ioannides, A.A.,:** *"Semantic geodesic maps: a unifying geometrical approach for studying the structure and dynamics of Single trials evoked responses. Electroenceph."* Clinical Neurophysiology, 113:1209-1226, 2002.

[LAQU'81] **Laquer, H.T.,:** *„Asimptotic Limits for a Two-dimensional Recursion"*, Stud.appl.Math. 46, 271-277, 1981.

[LAWR'93] **Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.F., Wootton, J.C.,:** *"Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment"*, Science, 262(5131):208-214, 1993.

[LEE'94] **Kowlakowski, L.F., and Rice, K.A.,:** *"Accepted-Mutation Parsimony Functionally Classifies G-Protein-Coupled Receptors"*, Science (*in press*), 1994.

[LESK'86] **Lesk, A., Levitt, M., Chothia, C.,:***"Alignment of the amino acid sequences of distantly"*, Protein Engineering, 1(1):77-78, 1986.

[LEVE'66] **Levenstein, V.I., :** *"Binary Codes Capable of Correcting Deletions, Insertions, and Reversals"*, Cybernet, Control Theory,10, pp. 707-701.

[LI'01] **Li, M., Badger, J.H., Chen, X., Kwon, S., Kearney, P., and Zangh, H.:** *"An information based sequence distance and its application to whole mitochondrial genome phylogeny,"* Bioinformatics, vol.17, pg: 149-154, 2001.

[LI'03] **Li, M., Chen, X., Li, X., Ma, B., and Vitanyi, P., :** *"The similarity metric,"* Proceedings of the 14th ACM-SIAM Symposium Discrete Algorithms, 2003.

[LI'97] **Li, M., and Vitany, P.,:** *"An introduction to Kolmogorov Complexity and Its Applications,"* Springer Verlag, 1997.

[LIAO'03] **Liao, L., Noble, W.S.,:** *"Combining pairwise sequence similarity and support vector machines for remote protein homology detection,"* J. Comp. Biol., vol.10, no.6, pp.857-868, 2003.

[LIU'04] **Liu, X., Croft, W.B., :** *"Statistical Language Modeling For Information Retrieval"*, Center for Intelligent Information Retrieval, Univ. of Massachusetts,http://ciir.cs.umass.edu/pubfiles/ir-318.pdf

[LORD'03] **Lord, P.W., Stevens, R.D., Brass, A., Goble, C.A.,:** *"Semantic similarity measures as tools for exploring the gene ontology,"* PubMed, Pac. Symp. Biocomput., pp: 601-612, 2003.

[MADE'95] **Madej, T., Gibrat, J.F., and Bryant, S.H.,:** *"Threading a database of protein cores,"* Proteins: Structure, Function and Genetics, vol. 23, pp: 356-369, 1995.

[MANN'00] **Manning, C.D., Schütze, H.,:** *"Foundations of statistical natural language processing"*, Massachusetts Institute of Technology Press, Cambridge, Massachusetts London, England, pag:60-60;554 – 556;557 – 588, 2000.

[MARK'03] **Markowetz, F., Edler, L., Vingron, M.,:** *"Support Vector Machines for Protein Fold Class Prediction"*, Biometrical Journal, 45(3): 377 – 389, 2003.

[MARK'71] **Markov, A.A.:** *"Extension of the limit theorems of probability theory to a sum of variables connected in a chain.",* reprinted in Appendix B of: R. Howard. *Dynamic Probabilistic Systems, volume 1: Markov Chains*. John Wiley and Sons, 1971.

[MART'80] **Martinez, H.M.,:** " A New Algorithm for Calculating RNA Seconday Structure", Manuscript, .

[MART'83] **Martinez, H.M.,:** "An Efficient Method for Finding Repeats in Molecular Sequences", Nucleic Acids Res., 11, 4629-4634, 1983.

[MITC'97] **Mitchell, T.,:***"Machine learning, WCB McGraw-Hill"*, Boston, 1997.

[MURZ'95] **Murzin, A.G.,Brenner, S.E., Hubbard, T., Chithia, C.,:** *"SCOP: A structural classification of protein database for the investigation of sequences and structures"*, J.Mol.Biol., 247, 1995.

[NCBI'05] **N**ational **C**enter for **B**iotechnology **I**nformation, accesibil online la adresa: (http://www.ncbi.nlm.nih.gov/), (accesat la 5 Iunie 2005).

[NEED'70] **Needelman, S.B., Wunsch, C.D.,:** *"A General Method Applicable to the Search for Similarityes in Amino Acids Sequences of Two Proteins"*, J. Mol. Biol, 48, pp.444-453, 1970.

[NOTR'00] **Notredame, C., Higgins, and D., Heringa J.,:** *"T-Coffee: A novel method for multiple sequence alignments"* Journal of Molecular Biology, 302: 205-217, 2000.

[PARK'98] **Park, J., Karplus, K., Barrett, C., Hughey, R., Haussler, D., Hubbard, T., and Chotia, C.,:** *"Sequence comparisons using multiple sequences detect three times asa many remote homologues as pairwise methods"*, J. Mol. Biol., vol. 284 no.4, pp: 1201-1210, 1998.

[PEAR'88] **Pearson, W.R., and Lipman, D.G.,:** *"Improved tools for biological sequence analysis"*, Proc. Natl Acad. Sci. USA, vol. 85, pp: 2444-2448, 1988.

[PEAR'96] **Pearson, W.R.,:** *"Effective protein sequence comparison"*, Methods Enzymol., vol. 266, pp: 227-259, 1996.

[PEAR'97] **Pearson, W.R.,:** *"Identifying distantly related protein sequences"*, Comput. Applic. Biosci., vol.13, pp: 325-332, 1997.

[PEAR'98] **Pearson, W.R.,:** *"Empirical statistical estimates for sequence similarity searches,"* J. Mol. Biol., vol. 276, pp: 71-84, 1998.

[PEVZ'00] **Pevzner, P.A., Sze, S.H.,:** *"Combinatorial approaches to finding subtle signals in DNA sequences"*, Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology(ISMB), pp: 269 – 278, 2000.

[PYLE'99] **Pyle, D.,:** *"Data Preparation for Data Mining"*, Morgan Kaufmann, Bk&CD Rom edition, March 15, 1999.

[QUEE'82] **Queen, C.M., Wegman, N., and Korn, L.J.,:** *"Improvements to a Program for DNA Analysis: a procedure to find Homologies Among Many Sequences"*, Nucleic Acids Res., 10, 449-456, 1982.

[REBH'98] **Rebhan, M., Chalifa-Caspi, V., Priluski, V., Lancet, D.:** *" GeneCards: a novel functional genomics compendium with automated data mining and query reformulation support"*, Bioinformatics, 14, no.8, pp: 656-664, 1998.

[REIC'73] **Reichert, T.A., Cohen, D.N., and Wong, A.K.C.,:** *"An Application of Information Theory to Genetic Mutations and Matching of Polypeptide Sequences"*, J. Theor. Biol., 42, 245-261, 1973.

[RIGO'98] **Rigoutsos, I., Floratos, A.,:** *"Motif discovery without alignment or enumeration"*, Proceedings of the second annual international conference on Computational Molecular Biology (RECOMB), New York, pp:221-227, 1998.

[ROSE'00] **Rosenfeld, R., :** *"Two decades of statistical language modeling: where do we go from here?"* Proceedings of the IEEE, 88(8), 2000.

[ROSE'94] **Rosenfeld, R.:** *"Adaptive Statistical Language Modeling: A Maximum Entropy Approach"*. Ph.D. Thesis. Carneige Mellon University, April 1994.

[RYCH'00] **Rychlewski, L., Jaroszewski, L., Li, W., Godzik**, A. *Protein Science* 9:232-241, 2000.

[SAIG'04] **Saigo, H., Vert, J-P., Ueda, N., Akutsu, T.:** *"Protein homology detection using string alignment kernels,"* J. Bioinformatics, vol.20 no.11, pp:1682-1689, 2004.

[SANK'72] **Sankoff, D.,:** *"Matching Sequences Under Deletion-Insertion Constraints"*, Proc. Natn. Acad. Sci. U.S.A., 68, 4-6, 1972 .

[SANK'73] **Sankoff, D., and Sellers, P.H.,:** "*Shortcuts, Diversions and Maximal Chainsin Partiallz Ordered Sets*", Discrete Math., 4, 287-293, 1973.

[SANK'76] **Sankoff, D., Cedergren, R.J., and Lapalme, G.,:** "*Frequency of Insertion-Deletion, Transversion, and Transition in the Evolution of 5S Ribosomal RNA*," J. Mol.Evol. vol.7, ppg: 133-149, 1976.

[SANK'83] **Sankoff, D., and Kruskal, J.B.,:** *"Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison"*, Addison-Wesley, London, 1983.

[SANK'85] **Sankoff, D.,:** *"Simultaneous Solution of the RNA Folding; Alignment and Protosequence Problem"*, SIAM, J. Applied Mathematics, 45, pp: 810-825, 1985.

[SANT'89] **T.G. Santner and D.E. Duffy,:** "*The Statistical Analysis of Discrete Data*," Springer-Verlag, New York, 1989.

[SARK'02] **Sarkar, I.N., Phil, M., and Rindflesch, T.C.,:** "*Discovering Protein Similarity using Natural Language Processing*," Cognitive Science Branch of Lister Hill National Center for Biomedical Communications, National Library of medicine, 2002.

[SCHÄ'01] **Schäffer, A., , Aravind, L., Madden, L., Shavirin, S., Spouge, J., Wolf, Y., Koonin, E., Altschul, S.,** Nucleic Acids Research, 29, (14), pp: 2994-3005, 2001.

[SCOP] **S**tructural **C**lassification **O**f **P**roteins, site-ul oficial disponibil online la: http://scop.mrc-lmb.cam.ac.uk/scop/, (accesat la 2 Mai 2005).

[SEAR'95] **Searls, D.B., and Murphy, K.P.,:** *"Automata-theoretic models of mutation and alignment"*, Proceedings of the third International Conference on Intelligent Systems for Molecular Biology, pp: 341-349, 1995.

[SELL'74a] **Sellers, P.,:** *"An Algorithm for the Distance Between Two Finite Sequences"*, Comb. Theorz, 16, 253-258, 1974.

[SELL'74b] **Sellers, P.,:** *"On the Theory and Computation of Evolutionary Distances"*, SIAM J. Appl. Math., 26, pp.787-793, 1974.

[SELL'79] **Sellers, P.,:** *"Pattern Recognition in Genetic Sequences"*, Proc. Natn. Acad. Sci. U.S.A., 76, pp.3041, 1979.

[SELL'80] **Sellers, P.,:** "The *theory and Computation of Evolutionary Distances: Pattern Recognition*",J. Algorithms, 1, pp.359-373, 1980.

[SHAN'51] **Shanon, C.E.,:** *"Prediction and Entropy of Printed English"*, Bell Systems Technical Journal, 30: 50-64, 1951.

[SHEP'80] **Shepard, R.N.,:** *"Multidimensional Scaling, Tree-Fitting, and Clustering"*, Science, 210, pp.390-398, 1980.

[SJOL'96] **Sjolander, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., Mian, I.S. and Haussler, D.:** *"Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology*," J. Bioinformatics, Vol 12, pp: 327-345, 1996.

[SMIT'81] **Smith, T.F., Waterman, M.S., Fitch, W.M.,:** *"Comparative Biosequence Metrics"*, J. Mol. Evol, 18, 38-46, 1981.

[SMIT'81a] **Smith, T.F., and Waterman, M.S.,:** *"Identification of Common Molecular Subsequences"*, J. Mol. Biol., 147, 195-197, 1981.

[SMIT'81b] **Smith, T.F., Waterman, M.S., Fitch, W.M.,:** *"Comparison of Biosequences"*, Adv. Appl. Math., 2, 482-489, 1981.

[SMIT'84] **Smith, T.F., Waterman, and Burks, C.,:** *"The Statistical Distribution of Nucleic Acids Similarities"*, In prep.

[SMIT'90] **Smith, H.O., Annau, T.M., Chandrasegaran, S.,:** *"Finding sequence motifs in groups of functional related proteins"*, Proceedings of the National Academy of Sciences of the United States of America, 87(2): 826-830, 1990.

[SSEA'05] **SSEARCH**, descrierea aplicaţiei disponibilă online la data de 9 iunie 2005 la adresa: http://ori.nibb.ac.jp/SIT/SSEARCH.html.

[STAN'70] **Stanton, R.G., and Cowan, D.D.,:** *"Note on a 'Square Functional' Equation"*, SIAM Rev.12,277-297, 1970.

[STUD'78] **Studnicka, G., rahn, G., Cummings, I., and salser, W.,:** *„Computer Method for Predicting the Seconday Structure of Single Stranded RNA"*, Nucleic Acids Res., 5, 3365-3387, 1978.

[TAYL'84] **Taylor, P.,:** *„A fast Homology Program for Aligning Biological Sequences"*, Nucleic Acid Res., 12, 447-455, 1984.

[THOM'94] **Thompson, J.D., Higgins, D.G., and Gibson, T.J.:** *" CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice."* Nucleic Acids Research, 22:4673-4680, 1994.

[TOMP'99] **Tompa, M.,:** *"An exact method for finding short motifs in sequences, with application to the ribosome binding site problem"*, in Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB), pp: 262-271, 1999.

[TUER'90] **Tuerk, C.,Gold, L.,:** *"Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase"*, Science, 249: 505-510, 1990.

[UKKO'83] **Ukkonen, E.,:** *"On Approximate String Matching"*, Proc. Int. Conf. Found. Comp.Theor.Lectures Notes in Comp. Sci., 158, 487-496, 1984.

[UKKO'84] **Ukkonen, E.,:** *"Algorithms for Approximate String Matching"*, Informat. Control (in press), 1984.

[ULAM'72] **Ulam, S.M.,:** *"In Applications of Number Theory to Numerical Analysis"*, Ed. S.K. Zaremba, Academin Press New York, pp.1-3, 1972.

[VANC'03] **Van Compernolle, D.,:** *"Spoken Language Science and Technology"*,2003, http://www.esat.kuleuven.ac.be/compi/pub/spoken_language/TOC.htm

[VANH'98] **van Helden, J., Andre, B., Collado-Vides, J.,:** *"Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotides frequencies"*, J. of Molecular Biology, 281(5), pp:827-832, 1998.

[WAGN'83] **Wagner, R.H.,:** *"In Time Warps, Sring Edits, and Macromolecules: the Theory and Practice of Sequence Comparison"*, Eds. D. Sankoff and J.B. Kruskal, Addison-Wesley, London, pp. 215-235, 1983.

[WANG'00a] **Wang, J.T.L., Ma,Q., Shasha, D., Wu, C.H.,:** *"Application of Neural Networks to Biological Data Mining: A Case Study in Protein Sequence Classification"*, Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, Massachusetts, United States, 2000.

[WATE'76] **Waterman *et al.*, 1976 Waterman, M. S., Smith, T. F. , Beyer, W. A.**: *"Some biological sequence metrics."*, Advances in Mathematics, 20, 367-387, 1976.

[WATE'83] **Waterman, M.S.:** *"Sequence Alignment in the Neighborhood of the Optimum with General Applications to Dynamic Programming"*, Proc. Natn. Acad. Sci. U.S.A., 80, 3123-3124, 1983.

[WATE'84] **Waterman, M.S.:** *"General methods of sequence comparison"*, Bulletin of Methematical Biology, Vol. 46, No. 4, pp. 473-500, printed in Great Britain, 1984.
http://www.cmb.usc.edu/papers/msw_papers/msw-054.pdf

[WATE'86] **Waterman, M.S.:** *"Multiple sequence alignment by consensus,"* Nucleic Acids Res.,vol. 14, no.22, pp: 9095–9102, 1986.

[WHEE'96] **Wheeler, D.:** *"Weight Matrices for Sequence Similarity Scoring"*, sinteză, http://www.icp.ucl.ac.be/~opperd/private/matrices.html;

[WIKI'05a] **Wikipedia,** The free encyclopedia.Nucleotide.
http://en.wikipedia.org/wiki/Nucleotide

[WIKI'05b] **Wikipedia**, Sequencing,
http://en.wikipedia.org/wiki/Biological_sequence;

[WIKI'05c] **Wikipedia**, The free enciclopedia. Protein folding.
http://en.wikipedia.org/wiki/Protein_folding;

[WIKI'05d] **Wikipedia**, The free enciclopedia. Peptide.
http://en.wikipedia.org/wiki/Peptide;

[WIKI'05e] **Wikipedia**, Biological database
http://en.wikipedia.org/wiki/Biological_database;

[WIKI'05f] **Wikipedia**, The free enciclopedia. Multiple alignment.
http://en.wikipedia.org/wiki/Sequence_alignment

[WIKI'05g] **Wikipedia,**The free enciclopedia. Exon.
http://en.wikipedia.org/wiki/Exon

[WIKI'06a] **Wikipedia,**The free enciclopedia. Cladistics. http://en.wikipedia.org/wiki/Cladistics

[WILB'83] **Wilbur, W.J., and Lipman, D.J,:** *"Rapid Similarity Searches of Nucleic Acid and Protein Data Bank"*, Proc. Natn.Acad.Sci.U.S.A., 80, 726-730, 1983.

[WILB'84] **Wilbur, W.J., and Lipman, D.J,:** *"The Context Dependent Comparison of Biological Sequences"*, SIAM, J. appl. Math., in press, 1984.

[WONG'74] **Wong, A.K.C., Reichert, T.A., Cohen, D.N., and Aygun, B.O.,:** *"A Generalized Method for Matching Informational Macromolecular Code Sequences"*, Comput., Biol., Med., 4, pp.43-57, 1974.

[WU'00] **Wu, C.H., McLarty, J.,:** *"Neural Networks and Genome Informatics", Elsevier Science,* 2000.

[WU'03] **Wu, K-P., Lin, H-N., Sung T-Y., and Su, W-L.,:** *"A new Similarity Measure among Protein Sequences,"* IEEE Computer Society Bioinformatics Conference (CSB'03), 2003.

[YAO'04] **Yao, Q., Huang, X., An, A.:** *"Applying Language Modeling to Session Identification from Database Trace Logs"* work in progress, http://www.cs.yorku.ca/~qingsong/research/ngram.pdf

[YOUN'97] **Young, S., Bloothooft, G.:** *"Corpus-Based Methods in Language and Speech Processing"*, Kluwer Academic Publishers, 1997.

[ZHAN'99] **Zhang, Z., Berman, P., Wiehe, T., and Miller, W.,:** *"Post-processing long pairwise alignments,"* Bioinformatics, vol.15, no.12, pp: 1012-1019, 1999.

INDEX

N

O

P

R

S

V