CONTRIBUȚII LA ANALIZA ȘI PRELUCRAREA DATELOR ÎN ANALIZA GENETICĂ

Teză destinată obţinerii
titlului ştiinţific de doctor inginer
la
Universitatea Politehnica Timişoara
în domeniul CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
de către

Dr. Med. Ing. Nicolae Teodor Meliță

Conducător ştiinţific:	Prof.univ.dr.ing. Ștefan Holban
Referenţi ştiinţifici:	Prof.univ.dr.ing. Robert Gyorodi
	Prof.univ.dr.mat. Alexandru Cicortaș
	Prof.univ.dr.ing. Vasile Stoicu-Tivadar

Ziua susţinerii tezei: 31.10.2016

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
12. Ingineria sistemelor
13. Inginerie energetică
14. Calculatoare şi tehnologia informaţiei
15. Ingineria materialelor
16. Inginerie şi Management
17. Arhitectură
18. Inginerie civilă și instalații

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

Evoluțiile recente în tehnologia informației au revoluționat numeroase domenii de cercetare. Medicina, este poate unul dintre cei mai importanți beneficiari ai acestor progrese. Volumul datelor care descriu aspecte cu impact direct sau indirect în practica medicală a crescut exponențial în ultima decadă. Aceste noi perspective, au solicitat metodele specifice inteligenței artificiale pentru a oferi soluții la imperativele de prelucrare și analiză a vastului influx de date disponibile pentru cercetare. Specialiștii în inteligență artificială au răspuns acestei provocări prin elaborarea unor metode din ce în ce mai complexe pentru a adresa cerințele noilor exigențe.

Algoritmii evoluționiști reprezintă unelte extrem de utile în acest deziderat. Algoritmii genetici (AG) au fost aplicați cu succes pentru a rezolva diverse probleme de optimizare în domenii variate.

Teza de doctorat propune o privire proaspătă asupra algoritmilor genetici, cu abordări originale, inspirate din evoluția naturală, pentru a adresa dificultățile cu care se confruntă implementarea clasică. Am studiat utilitatea și îmbunătățirea aplicabilității algoritmilor genetici în analiza genetică, cu precădere pentru cercetarea datelor oferite de tehnologia ADN microarray. Metodele propuse modelează fenomene biologice care stau la baza evoluției naturale. Am testat performanța algoritmului propus în analiza de date reale ADN microarray, iar rezultatele obținute demonstrează o îmbunătățire semnificativă față de abordarea clasică și relevă utilitatea metodei ca alternativă la maniera actuală de a efectua un astfel de experiment. Generalitatea principiilor din natură modelate, ne îndreptățesc să prevedem utilitatea abordării noastre în tratarea altor provocări, adresate de diferite domenii, inteligenței artificiale.

Timişoara, August 2016 Nicolae Teodor Meliță

Această teză de doctorat nu ar fi fost realizată fără suportul necondiționat al domnului Prof. Dr. Ing. Ștefan Holban. Îi mulțumesc domnului profesor pentru suportul profesionist, acordat în calitate de conducător științific și pentru deschiderea cu care mi-a împărtășit cunoștințele, experiența și timpul domniei sale. Sunt recunoscător pentru răbdarea și suportul moral pe care domnul profesor mi l-a oferit ori de câte ori a fost necesar.

Mulțumesc familiei pentru susținere, înțelegere și fiului meu Tomic pentru buna dispoziție oferită zi de zi.

Meliță, Nicolae Teodor

Contribuții la analiza și prelucrarea datelor în analiza genetică

Teze de doctorat ale UPT, Seria 14, Nr. 34, Editura Politehnica, 2016, 152 pagini, 90 figuri, 28 tabele.

ISSN: 2069-8216
ISSN-L:2069-8216
ISBN: 978-606-35-0101-2

Cuvinte cheie: algoritmi genetici, dominanță incompletă, atribuire aleatoare a cromozomilor, algoritmi evoluționiști, DNA micoarray

Rezumat,
Teza de doctorat propune alternative la abordarea clasică în implementarea algoritmilor genetici. Soluțiile noastre modelează principii ale evoluției naturale, cercetate detaliat în biologie. Algoritmul genetic diploid propus, utilizează un model inspirat din principiul dominanței incomplete ca alternativă la definirea unei scheme de mapare a genotipului la fenotip. Dominanța incompletă oferă avantaje în privința exploatării și în flexibilitatea algoritmului genetic diploid. Atribuirea aleatorie a cromozomilor, operator original, modelat după un proces din desfășurarea naturală a meiozei la eucariote, îmbunătățește explorarea comparativ cu operatorii clasici. Operatorii introduși pentru mutații, de asemenea modelați după procese confirmate în biologie, oferă avantaje suplimentare în acest sens. Testele efectuate cu operatorii propuși demonstrează utilitatea și aplicabilitatea practică a algoritmului în analiza ADN microarray.

Cuprins

1. INTRODUCERE

Evoluția fulminantă a tehnologiei din deceniile recente are un impact major asupra activității și cercetării în toate domeniile. Modalități noi, avansate de achiziție a informației au fost imaginate și implementate pentru cele mai diverse sfere de interes teoretic si practic, iar capacitatea de stocare a datelor e crescut exponențial. Medicina este unul dintre domeniile care a avansat enorm și este într-un continuu proces de evoluție. De la noi metode imagistice de diagnostic sau teste de laborator avansate pentru detectarea precoce a diverse patologii, până la procedee moderne de intervenție chirurgicală asistate de roboți, pentru tratament, toate disciplinele medicale au progresat. Ramurile medicale preclinice au evoluat corespunzător și permit înțelegerea mai intimă a diferitelor procese fiziologice sau patologice de interes.

O consecință a acestor desfășurări este influxul major de date, datorat achiziției de semnale diverse prin noile metode disponibile. Informații indisponibile deunăzi sunt astăzi înregistrabile și interpretabile pentru a descrie procese încă necunoscute. Dacă tehnica de calcul a răspuns pe măsură la imperativul de stocare și organizare a acestei afluențe de informații, metode noi și eficiente de analiză, interpretare și integrare a acestor date sunt necesare pentru a valorifica noile oportunități ale realității actuale.

Capitolul introductiv conturează cadrul în care cercetarea ce stă la baza acestei teze de doctorat s-a desfășurat. Motivăm alegerea temei de doctorat și trasăm obiectivele propuse. De asemenea, descriem structura tezei în contextul finalității urmărite.

1.1. Motivație

Metodele inteligenței artificiale (IA) au fost incremental solicitate pentru a analiza și înțelege procese modelate în cele mai diferite domenii de studiu. Medicina a beneficiat semnificativ de progresele din domeniul IA și este de previzibil că această legătură este doar începutul unei asocieri reciproc avantajoase. O disciplină cu impact major în medicină care a înflorit în mod spectaculos în perioada recentă și cu suportul IA este bioinformatica. Mai mult, imperativele noi din bioinformatică au apelat adesea la metodele specifice IA și au stimulat evoluția lor în consecință. Analiza genelor diferențial exprimate constituie una dintre direcțiile fundamentale în bioinformatică, iar în acest domeniu, utilitatea metodelor IA s-a atestat prin rezultate spectaculoase, cu impact în activitatea medicală clinică.

Algoritmii evoluționiști (AE) sunt o parte importantă a disciplinei inteligenței artificiale și au fost adesea utilizați pentru a interpreta date achiziționate pentru a descrie modele din cele mai diverse. Impactul abordărilor evoluționiste în bioinformatică este important, iar un domeniu care a fost adresat cu metode evoluționiste este analiza datelor achiziționate cu tehnologia ADN microarray. În această materie, AE au fost utilizați cu succes limitat și considerăm că aceasta direcție trebuie explorată și dezvoltată.

Algoritmii evoluționiști sunt clădiți pe principii testate timp de miliarde de ani, în care procese evolutive au răspuns cu succes și în mod divers, la schimbările și provocările mediului înconjurător. Odată cu o mai bună înțelegere a proceselor care susțin și determină evoluția în biologie, sporește posibilitatea de-a modela mai intim aceste principii în inteligența artificială, cu îmbunătățirea algoritmilor evoluționiști actuali și șansa de-ai utiliza cu succes pentru a aborda provocări din ce în ce mai diverse.

Tehnologia ADN microarray este o metodă larg utilizată și bine fundamentată în bioinformatică. Accesul facil la seturi de date reale și rezultatele corespunzătoare, obținute prin metode diverse, de numeroși cercetători, oferă o oportunitate majoră de a dezvolta metodele IA într-un cadru consolidat și extensiv explorat. Posibilitatea de-a evalua performanța unor metode noi de AE în analiza unor date reale, șansa de-a compara comportamentul metodelor propuse cu abordări intens testate și potențialul de-a îmbunătăți metodele de analiză a datelor ADN microarray concomitent, este foarte atractivă.

Deoarece tehnologia ADN microarray a beneficiat de o atenție și dezvoltare majoră în ultima decadă, unelte software diverse și foarte bine testate sunt disponibile. Am dezvoltat aplicațiile solidare tezei de doctorat în mediul R[1], perfect integrabile în Bioconductor[2]. Bioconductor oferă o platformă flexibilă pentru analiza datelor în bioinformatică, acces facil la seturi de date reale și metode implementate de cercetători din domenii diverse. În plus, este foarte popular printre cercetătorii din mediul academic din toată lumea, constituiți într-o comunitate foarte prietenoasă. De asemenea, contribuția utilizatorilor este încurajată atât în R cât și în Bioconductor.

Această teză de doctorat introduce principii modelate din evoluția biologică naturală cu scopul de-a îmbunătăți performanța și aplicabilitatea algoritmilor genetici (AG). Principiul dominanței incomplete din biologie este modelat și implementat în contextul unui AG diploid. De asemenea, explorăm oportunitatea tratării unui set haploid de cromozomi în comparație cu implementarea clasică în care este utilizat un singur cromozom pentru a codifica configurația atributelor. Consecutiv, cercetăm oportunitățile oferite de implementarea diploidă și numărul variabil de cromozomi pentru introducerea unui nou operator de recombinare, inspirat din atribuirea aleatorie a cromozomilor în timpul meiozei și formularea a cinci operatori noi pentru a modela tipuri de mutații descrise în genetica modernă.

1.2. Obiective

În realizarea tezei de doctorat de față, am urmărit modelarea și implementarea unor principii care fundamentează evoluția naturală în biologie, cu scopul îmbunătățirii performanței și aplicabilității algoritmilor genetici. Punctual, ne-am propus:

1) Modelarea dominanței incomplete,

2) Estimarea oportunității de-a reprezenta genomul printr-un variabil de cromozomi,

3) Modelarea principiului atribuirii aleatorii a cromozomilor din timpul meiozei și introducerea unui nou operator de recombinare corespunzător,

4) Testarea noilor modele și operatori în contextul selectării atributelor în analiza datelor obținute cu tehnologia ADN microarray,

5) Realizarea unui pachet software integrabil în R și Bioconductor, accesibil pentru testare și utilizare de către cercetătorii în domeniul analizei datelor ADN microarray.

Obiectivele enumerate mai sus au fost atinse și sunt dezbătute pe parcursul tezei de doctorat. Pe parcursul dezvoltării modelelor și evaluării rezultatelor, alte obiective au fost adăugate, explorate și testate:

6) Modelarea mutației fără sens,
7) Modelarea mutației cu deplasare,
8) Modelarea mutației cu ștergerea unui segment de cromozom,
9) Modelarea mutației ștergerea unui întreg cromozom,
10) Modelarea transpozonilor.

1.3. Publicații

Progrese incremental obținute și principii modelate pe parcursul cercetărilor și implementărilor din teza de doctorat au fost comunicare și prezentate în conferințe cu participare internațională și jurnale respectate în domeniile abordate.

1) Nicolae Teodor MELIȚĂ și Ștefan HOLBAN. An Incomplete Dominance Genetic Algorithm Approach to Microarray Data Analysis. In Proceedings of the 12th IEEE International Conference on Intelligent Computer Communication and Pocessing (ICCP), Cluj, Romania, Septembrie 2016, IEEE.

Prezentarea în cadrul conferinței introduce principiul dominanței incomplete și evaluează rezultatele obținute la utilizarea implementării cu un algoritm genetic diploid, pentru selectarea atributelor într-un context de date reale ADN microarray.

2) Nicolae Teodor MELIȚĂ, Irinel POPESCU și Stefan HOLBAN. A Genetic Algorithm Approach to DNA Microarrays Analysis of Pancreatic Cancer. Advances in Electrical and Computer Engineering, no. 2/2008, vol. 8, 2008, pp. 43-48.

Articolul publică rezultatele obținute în selectarea atributelor cu un algoritm genetic haploid, într-un experiment real ADN microarray. Reprezintă rezultatul unei colaborări cu Centrul de Chirurgie Generală şi Transplant Hepatic, Institutul Clinic Fundeni în proiectul "Gene Expression Profile and Biomarkers Study Correlated with Clinicopathological Parameters in Pancreatic Cancer" (GENOPACT) - grant CEEX 56/2005.

3) Nicolae Teodor MELIȚĂ și Ștefan HOLBAN. A Genetic Algorithm - Support Vector Machine Approach to DNA Microarrays Supervised Learning. Conferința Development And Application Systems, Ediţia a IX-a, Suceava, România, Mai 22-24, 2008.

Prezentarea în cadrul conferinței cuprinde evaluarea oportunității utilizării algoritmilor genetici pentru selectarea atributelor în experimente de tip microarray.

4) Nicolae Teodor MELIȚĂ și Ștefan HOLBAN. dGAselID: An R Package for Selecting a Variable Number of Features in High Dimensional Data. The R Journal, în publicare.

Articolul prezintă pachetul software rezultat în urma implementării cercetărilor cuprinse în lucrarea de doctorat. Pachetul R dGAselID este complet integrabil în Bioconductor.

5) Nicolae Teodor MELIȚĂ. Evaluating statistical packages for genome-wide association studies. Research Poster, mentori: Dr. Wacholder și Kai Yu, Ph.D, Division of Cancer Epidemiology and Genetics, National Institutes of Health, Bethesda, SUA, 2007.

Poster-ul prezintă metodele statistice implementate în pachete software disponibile pentru analiza datelor în studiile genome-wide association. National Institutes of Health este una dintre cele mai prestigioase autorități de cercetare în medicină din lume, iar proiectul respectiv a cuprins dezvoltarea unui pachet software pentru abordarea acestui tip de experiment, foarte nou la acea dată, în mediul R.

1.4. Structura tezei de doctorat

Capitolul 2 al tezei de doctorat conturează cadrul teoretic pe care cercetările de față au fost fundamentate. Este prezentată tehnologia ADN microarray și sunt discutate aspecte din realizarea practică a unui experiment utilizând această tehnologie. În continuare, sunt descrise principii și metode ale inteligenței artificiale utilizate pe scară largă în analiza genetică, în particular cu tehnologia ADN microarray.

Capitolul al treilea descrie metodele propuse pentru îmbunătățirea performanței în selectarea atributelor cu ADN microarray. Ulterior prezentării algoritmilor genetici în general, este introdus modelul dominanței incomplete. Modelul este discutat ca alternativă de mapare a genotipului la fenotip în algoritmii genetici și sunt ilustrate principiile din biologie care susțin această propunere. De asemenea, este introdus principiul sortării aleatorii al cromozomilor în biologie și operatorul modelat pentru AG. În continuare, sunt abordate modalitățile de apariție a mutației fără sens, mutației cu deplasare, mutației cu ștergerea unui segment de cromozom, mutației ștergerea unui întreg cromozom și a transpozonilor în genetică. Pentru fiecare dintre aceste fenomene este propus un model și în consecință, un operator pentru utilizare în AG.

Capitolul 4 introduce pachetul R dGAselID dezvoltat pentru a implementa contribuțiile din teza de doctorat și a le integra în Bioconductor. Sunt discutate formatele de date utilizate, funcțiile disponibile și parametrii corespunzători, metodele incluse pentru vizualizarea evoluției și a rezultatelor obținute.

Capitolul 5 prezintă experimente cu date reale ADN microarray cu discutarea rezultatelor obținute în evaluarea principiilor și operatorilor introduși în capitolele anterioare.

Capitolul 6 concluzionează teza de doctorat, cu emfază pe contribuțiile autorului și abordează punctual experiența în evaluarea metodelor și fiecărui operator propus.

Bibliografia include resursele care au fost consultate pe parcursul realizării tezei de doctorat.

2. CADRU TEORETIC

Tehnologia ADN microarray reprezintă un progres major în analiza genetică. Metodologia a fost utilizată în numeroase studii din genetică și biologie moleculară, iar impactul a depășit granițele cercetării fundamentale. Teste diagnostice dezvoltate pe fundamentul tehnologiei ADN microarray sunt utilizate în activitatea clinică modernă.

Capitolul de față introduce tehnologia ADN microarray și revizuiește principiile biologice care justifică utilizarea metodei și semnificația rezultatelor unor astfel de studii. Descriem, de asemenea, etapele în desfășurarea unui astfel de experiment și provocările cu care metoda se confruntă. Contextul în care tehnologia este utilă, modul în care rezultatele pot fi interpretate și impactul în activitatea clinică sunt deopotrivă abordate.

Statistica este cea care oferă metodele utilizate într-o analiză standard cu date de ADN microarray. Totuși, concluziile și semnificația biologică a rezultatelor se realizează de către biologi și geneticieni supra-specializați în această direcție, depinzând semnificativ de experiența și cunoștințele fiecărui cercetător. Un interes deosebit am acordat unor metode ale disciplinei inteligenței artificiale care pot fi utilizate pentru analiza genetică în contextul acestei tehnologii și pot susține investigatorii în activitatea de interpretare și validare biologică a rezultatelor. Am analizat rolul unor metode de învățare supervizată și nesupervizată în deslușirea datelor vaste rezultate dintr-un astfel de studiu și am revizuit câteva metode de inteligență artificială, adesea utilizate în abordarea acestei problematici.

2.1. Tehnologia DNA microarray

Tehnologia microarray a fost utilizată pentru diferite aplicații și cadre de cercetare. Dacă utilizarea primordială a fost pentru analiza genelor diferențial exprimate, analiza matisării și fuziunii în maturarea pre-ARN-ului la ARN mesager, a polimorfismelor uninucleotidice pentru decelarea alelelor cu diferențe de până la o nucleotidă sau cercetarea interacțiunilor acizilor nucleici cu proteinele cad în spectrul de aplicabilitate al metodei.

Analiza genelor diferențial exprimate cu ajutorul ADN microarray, probabil, aplicația cel mai extensiv tratată în literatura de specialitate, a cunoscut o evoluție și o influență remarcabile în analiza genetică, de când tehnologia a fost introdusă [3].
Dogma centrală în biologie enunțată încă din 1956 [4] explică modul în care informația genetică codificată în ADN devine exprimată la nivelul fenotipului prin intermediul ARN și al proteinelor sintetizate în urma transcripției și respectiv a translației. Consecința dogmei centrale în biologie este că există o relație de cauzalitate între expresia genetică și proprietățile relevate în fenotip, prin intermediul proteinelor sintetizate, pe filiera ARN. Fiecare genă produce ARN mesager specific. Expresia genetică determină tipul de ARN mesager produs la un moment dat, într-o anumită mostră biologică, și în consecință, tipul proteinelor sintetizate. Evaluarea expresiei genetice între mostre aflate în condiții diferite, descrie condițiile respective. Expresia genetică poate fi evaluată prin cuantificarea

ARN-ului mesager sau al proteinelor sintetizate. În cazul proteinelor, tehnologii ca Western Blot, spectrometrie de masă sau cromatografie oferă soluții în acest sens. ARN-ul mesager poate fi cuantificat cu metode ca Northern Blot, RT-qPCR, secvențializare sau microaray.

ADN microarray oferă o imagine a condițiilor care circumscriu o instanță la un moment particular și oportunitatea descrierii proceselor biologice complexe plecând de la impactul asupra fenotipului al expresiei genetice. Tehnologia permite măsurarea concentrației ARN-ului mesager produs prin transcripție într-o instanță, la un anumit moment. În plus, permite evaluarea expresiei genetice a mii de gene imobilizate pe un singur chip. Probele sunt reprezentate de oligonucleotide în cantități infime, de ordinul picomolilor, fixate pe un suport solid. Probe multiple pentru o singură genă sunt reprezentate pe biochip. Aceasta abordare aduce informații valoroase pentru:

1) descoperirea unor marcheri cu valoare în diagnosticarea unor boli,
2) dezvoltarea de medicamente cu eficiență sporită pentru o anumită patologie,
3) descrierea diferitelor stadii într-o anumită patologie,
4) schimbările produse de contactul cu anumiți patogeni.

Principiul pe care este dezvoltată tehnologia ADN microarray, este că molecula monocatenară de ADN are tendința de-a se alinia cu molecula complementară pentru a forma legături de hidrogen între nucleotidele complementare și în consecință, structura dublu helix. În cazul ADN microarray, mii până la 25000 de oligonucleotide, fragmente scurte monocatenare de ADN (25-100 nucleotide), sunt imobilizate pe un suport solid (Fig. 2.1). Suportul solid variază în funcție de producătorul chip-ului și este de obicei realizat din plastic, sticlă sau silicon. În condiții speciale, oligonucleotide complementare, etichetate a priori cu un primer specific, care ajung în contact cu cele imobilizate pe chip, hibridizează pentru a forma fragmente cu structură dublu helix. După ce hibridizarea a avut loc, sau în anumite variante anterior hibridizării, eticheta este înlocuită cu o culoare fluorescentă. Culorile utilizate în mod clasic sunt Cy3 și Cy5. Cy3 emite cu o lungime de undă de 570 nm. Cy5 emite cu o lungime de undă de 670 nm.

Experimentele cu ADN microarray sunt complexe, necesită o pregătire și execuție atentă. Unelte speciale sunt disponibile în diferite variante comerciale, în funcție de producător. În general, câteva etape sunt obligatorii în realizarea cu succes a unui experiment de acest tip:

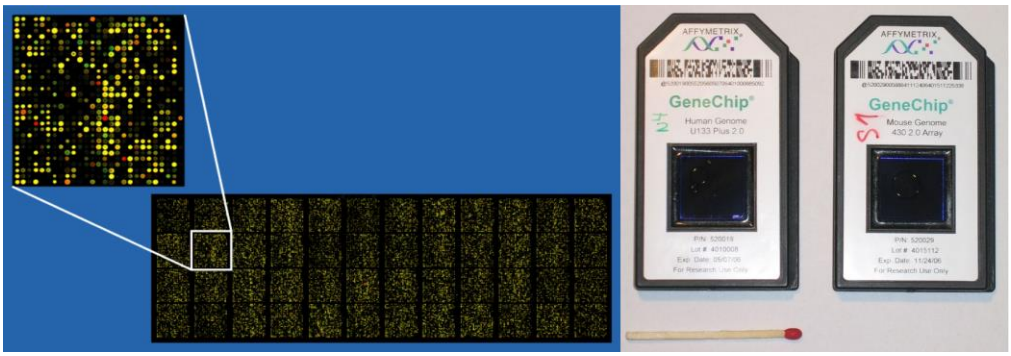
1) Inițial, ARN-ul mesager este izolat din proba biologică de interes,
2) ARN-ul mesager este purificat,
3) ARN-ul este transformat în ADN complementar (cADN) prin transcripție inversă,
4) cADN-ul astfel obținut este amplificat,
5) cADN-ul este etichetat cu primeri speciali,
6) cADN-ul astfel etichetat este pus în contact cu probe imobilizate pe chip-uri special preparate. Diferite variante și tehnologii de fabricație pentru chipuri sunt oferite de firme specializate în biotehnologii. Hibridizarea între fragmentele complementare de probe fixate pe biochip și cADN-ul țintă are loc în condiții bine controlate,
7) Culori fluorescente sunt adăugate etichetelor existente. Această etapă poate avea loc și înainte de hibridizare în unele variante comerciale,
8) Chip-ul este ulterior spălat pentru a îndepărta fragmentele de cADN care nu au format legături de hidrogen suficient de solide, sau au hibridizat nespecific,

9) Scannere speciale sunt utilizate pentru a achiziționa o imagine a cADN-ului etichetat care a hibridizat cu probele de pe chip. Măsurările cuantifică numărul fotonilor emiși de probe, după stimularea prealabilă cu laser, pe întregul chip,

10) Analiza și prelucrarea rezultatelor se efectuează cu pachete software oferite de producători pentru platformele comerciale sau pachete special concepute pentru astfel de experimente. R si Bioconductor sunt opțiuni populare. În prelucrarea datelor microarray, corectarea fundalului, normalizarea și sumarea sunt pași indispensabili pentru obținerea valorilor corecte care cuantifică nivelul expresiei genetice.

Chip-urile ADN sunt implementate în două variațiuni, cu una (Fig. 2.1 b)) sau două culori (Fig. 2.1 a)). Chip-urile cu două culori permit hibridizarea a două mostre pe un singur chip, în timp ce tehnologia cu o singură culoare necesită chip-uri diferite pentru fiecare mostră. Limitările tehnologiei au înclinat balanța înspre varianta cu o singură culoare, datorită consistenței sporite a măsurătorilor.

O limitare a tehnologiei microarray este reprezentată de rata redusă de raport semnal/zgomot [5] cu consecințe pentru sensibilitatea metodei. Acest neajuns se datorează densității probelor care pot fi imobilizate într-o anumită suprafață. În plus, numărul imens de gene evaluate în raport cu numărul mic de exemple, amplifică această dificultate, în etapa analizei datelor. Consecința constă în dificultatea de a izola zgomotul de procesul biologic exprimat prin măsurători. Implementarea chip-urilor cu structură tridimensională a adresat parțial acest neajuns. O evoluție remarcabilă este introducerea recentă a unor variante comerciale de biochipuri care pot fi adaptate de cercetător în privința oligonucleotidelor imobilizate, astfel încât investigatorul își poate modela cercetarea scopului urmărit.



a) b)

Fig. 2.1 - Exemplu de chip ADN microarray. a) Tehnologie Stanford; b) Tehnologie Affymetrix. (sursă – Academic Dictionaries and Encyclopedias, www.enacademic.com, domeniu public.)

2.2. Metode actuale de recunoaştere a formelor în analiza genetică

Evoluţia diacronică a umanităţii, de la primele unelte create de homo erectus în urmă cu 1,5 milioane de ani, la recentul accelerator de particule de la CERN a fost posibilă numai datorită muncii şi inteligenţei a generaţii de oameni iscusiţi. Etimologia cuvântului inteligenţă - verbul din limba latină „intellegere", cu semnificaţia „a înţelege" – trădează substanţa conceptului, dar sensul său este în continuă înnoire. Alfred Binet definea inteligenţa ca „judecată, altfel spus, înţelepciune, simţ practic, iniţiativă, facultatea de adaptare la circumstanţe" [6]. Cu toate că termenul de inteligenţă suferă o continuă redefinire, varianta lui Alfred Binet conturează satisfăcător semnificaţia conceptului. În esenţă, inteligenţa este capacitatea de adaptare la mediul înconjurător.

Psihologia modernă tratează inteligenţa drept o combinaţie selectivă a unor procese cognitive precum percepţie, învăţare, memorie, judecată şi rezolvarea de probleme, cu scopul adaptării eficiente la mediul înconjurător.

Inteligenţa umană este îngrădită de limitele organice ale sistemului nervos central şi periferic, precum şi a organelor senzoriale umane. Spre exemplu, percepţia umană, conştientizarea sau înţelegerea informaţiilor senzoriale, este condiţionată de restricţiile biologice ale organelor de simţ, ceea ce impune limite inteligenţei. Sistemul auditiv uman, permite perceperea vibraţiilor între 20 şi 20.000 Hz. Se presupune că, abilitatea oamenilor de a percepe auditiv infrasunetele şi ultrasunetele, ar fi însemnat o percepţie diferită a mediului înconjurător şi consecutiv, o capacitate diferită de adaptare la mediu.

O posibilă şansă de victorie în lupta cu graniţele organice ale inteligenţei umane o reprezintă inteligenţa artificială. În anul 1956, John McCarthy introducea termenul **inteligenţă artificială** pentru a denumi ştiinţa şi ingineria creării de maşini inteligente [7]. În accepţiunea modernă, inteligenţa artificială reprezintă ramura ştiinţei calculatoarelor care are ca obiect crearea de agenţi inteligenţi. Stuart Russel preferă denumirea „agent raţional" şi îl defineşte drept orice entitate care poate fi văzută ca percepându-şi mediul înconjurător şi acţionând asupra mediului cu efectori pentru a-şi asigura succesul maxim [8]. Aşadar un agent inteligent execută două procese de bază: percepţia mediului înconjurător şi acţiunea cu succes asupra acestuia. Revenind la exemplul anterior, tehnologia actuală permite realizarea unor senzori cu lărgime de bandă mult superioară organelor de simţ umane. Calculul reprezintă studiul schimbării, iar umanitatea dispune la ora actuală de sisteme capabile să execute operaţiuni matematice sau calcule cu viteză net superioară creierului uman. Beneficiem, de asemenea, de efectori foarte eficienţi, capabili să execute cu precizie operaţiuni foarte numeroase şi precise; roboţii industriali din orice tip de bandă de producţie fiind cel mai simplu exemplu. Suntem astfel îndreptăţiţi să sperăm că am putea realiza agenţi raţionali extrem de eficienţi. Problema este dacă am putea realiza agenţi raţionali superiori inteligenţei umane în general sau doar mai eficienţi pe o problemă specifică.

Pentru a considera un agent ca fiind în general inteligent, trebuie să ne raportăm la ceea ce psihologia actuală consideră despre inteligenţa umană. Aşadar, un astfel de agent ar trebui să fie capabil de judecată, cunoaştere, învăţare, planificare, comunicare, percepţie şi de a manipula obiecte. Un astfel de agent nu a fost încă realizat şi reprezintă finalitatea pe termen lung a cercetătorilor IA care se concentrează pe „General intelligence" sau „strong AI".

Cu toate că domeniul inteligenţei artificiale este încă în fază incipientă, progresele recente în domeniul inteligenţei artificiale deschid orizonturi noi în numeroase domenii şi produc rezultate remarcabile. O parte semnificativă a cercetătorilor IA consideră că „strong AI" este imposibil de realizat şi se concentrează pe „IA aplicat", au ca finalitate crearea de sisteme inteligente specializate pe o problemă particulară. Această ramură a IA cunoaşte un real succes în ultimul deceniu, iar aplicaţiile existente deja în medicină, economie, armată sau sisteme de securitate aduc beneficii importante. O altă ramură importantă a IA este „simularea cognitivă", cu aplicaţii foarte valoroase pentru cercetătorii din neuroştiinţă şi psihologie cognitivă.

Metode specifice inteligenței artificiale au fost utilizate în mod tradițional în bioinformatică [9] și în special pentru a adresa imperativele de descoperire a unor forme sau clasificare în datele achiziționate cu tehnologia microarray. Teza de față adresează problema analizei datelor rezultate din studiile ADN microarray în analiza genetică. Ne interesează aşadar problema punctuală a descoperirii formelor în seturile imense de date numerice rezultate din studiile ADN microarray şi explorăm posibilităţile pe care „IA aplicat" le oferă pentru un astfel de deziderat. Anterior analizei cu metode de inteligență artificială, datele microarray trebuie prelucrate în mod corespunzător. Aplicații clasice [10] ale metodelor IA în analiza microarray sunt: determinarea genelor diferențial exprimate, clasificarea și determinarea unei rețele genetice.

Testele de inteligenţă sunt foarte uzuale în zilele noastre şi probabil, oricine şi-a testat vreodată inteligenţa a trebuit să răspundă la întrebări de tipul „Completaţi cu următorul număr în secvenţa: 1, 2, 4, 8, 16". Răspunsul pe care orice om cu inteligenţa medie l-ar da imediat, reprezintă un proces simplu de recunoaştere a formelor, efectuat foarte rapid, urmat de o predicţie banală, pe baza trendului observat. Practic, subiectul distinge o forma „şir de puteri ale lui 2" în acest set de date şi prezice că următoare valoare este 32. Este evident că pe un set de date ca cele rezultate dintr-un experiment DNA microarray, de unde rezultă matrice conţinând milioane numere reale, găsirea unei forme este extrem de complicată pentru orice om cu inteligenţă superioară. Aşadar urmărim crearea unor agenţi inteligenţi foarte performanţi pentru această cerinţă.

Revoluţia tehnologică a ultimilor 20 de ani a dus la creşterea capacităţii de stocare a datelor şi ieftinirea semnificativă a logisticii necesare pentru stocare. Numai Large Hadron Collider generează aproximativ 15 pentaocteţi (15000 de teraocteţi) de date în fiecare an. Doar Interferometrul European compus din 16 telescoape, produce 16 Gb/secundă de date astronomice într-o sesiune de 25- zile de observaţie, sistemul de sateliţi NASA pentru meteorologie ajung la 46MB/s, iar Google dispune de peste 4 miliarde de pagini, mai multe sute de teraocteţi. La ora actuală, se apreciază că la fiecare 20 de luni, cantitatea de date stocate se dublează. Această situaţie a dus la utilitatea găsirii unor metode, automate sau semiautomate de analiză. În anii 1989 şi 1990 apar termenii de Knowledge Discovery in Databases (KDD), respectiv Data Mining, iar în 1995 a avut loc prima conferinţă internaţională cu tema KDD şi DM. Deşi iniţial termenii s-au suprapus, actualmente DM este considerat un subdomeniu al KDD.

Disciplina care se ocupă cu studiul extragerii de informaţii din seturile voluminoase de date este minarea de date (Data mining). Minarea datelor abordează problematica descrierii datelor de dimensiuni foarte mari cu proprietăți necunoscute [11]. Prin definiţie, **Data Mining** (DM) este procesul extragerii automate sau semiautomate de informaţii implicite, necunoscute anterior şi potenţial folositoare din date. Finalitatea DM este descoperirea unor forme bine

definite, care pot fi generalizate pentru a face predicţii pe date noi şi care oferă informaţii explicite cu privire la structura datelor studiate. Sunt deopotrivă importante, capacitatea de a generaliza formele descoperite şi posibilitatea de a oferi o semnificaţie explicită formelor descoperite. Pe lângă DM, KDD are ca obiecte şi tehnici de selectare, preprocesare a datelor, precum şi analiză a rezultatelor. DM este un domeniu multidisciplinar, la intersecţia a trei mari discipline: inteligenţă artificială, statistică şi baze de date. Metode specifice DM sprijină cercetarea în domenii variate, care beneficiază de noile capabilități de achiziție și stocare a datelor. Bioinfomaticienii sunt adesea sprijiniți [12] de uneltele DM.

DM se sprijină pe fundamentul constituit de bazele de date. Cu toate acestea abordarea DM prezintă un caracter inductiv, în timp ce bazele de date au caracter deductiv, şi implicit, capacitate de generalizare. Statistica oferă uneltele necesare validării rezultatelor obţinute prin data mining, în timp ce metodele tehnice ale DM sunt oferite de **Machine Learning** (ML), domeniu al inteligenţei artificiale. Relaţia dintre statistică şi DM este mai intimă decât la prima vedere. În afară de posibilitatea validării rezultatelor DM, statistica este strâns legată de Machine Learning, metodele celor două discipline evoluând în paralel şi interdependent. Introduceri [13-16] foarte utile în principiile și metodele ML sunt accesibile cercetătorilor interesați.

Menţionam anterior că o condiţie necesară pentru ca o entitate să poată fi considerată drept agent inteligent este capacitatea de-a acţiona cu succes asupra mediului înconjurător. Pentru a stabili însă dacă un agent a acţionat cu succes sau nu, trebuie comparată o acţiune recentă cu una mai veche într-o situaţie identică. Dacă performanţa se îmbunătăţeşte pe seama repetiţiei unei situaţii putem consideră că agentul respectiv a învăţat din experienţă.

Este importantă observaţia că ML presupune învăţare în mod practic, şi nu teoretic. În principiu, machine learning este preocupat cu algoritmii de învățare în date cu structură parțial cunoscută [17]. Rezultatele învăţării sunt legate de performanţă şi nu de acumularea unei cantităţi de informaţie sau de cunoaştere. Totuşi, procesul de învăţare, presupune în accepţiunea actuală gândire, intenţie şi scop din partea celui care învaţă. Învăţarea fără gândire, intenţie sau scop din partea celui care învaţă este mai degrabă antrenament (eng. training). Spre exemplu, folosim cuvântul dresaj pentru antrenarea sau învăţarea unui câine, deoarece scopul şi intenţia învăţării aparţine dresorului sau profesorului. Este diferit procesul de învăţare în cazul unui medic chirurg care urmează un curs de perfecţionare într-o nouă tehnică operatorie. Acesta va învăţa cu intenţie şi scop. În general, putem sintetiza că în antrenament scopul este al antrenorului în timp ce învăţarea presupune scop din partea studentului. Aplicată agenţilor inteligenţi, învăţarea presupune obţinerea unor descrieri structurale din exemple, unde descrierile structurale obţinute sunt ulterior utile pentru înţelegerea, descrierea datelor şi predicţii [18]. Spre exemplificare, una dintre primele aplicaţii practice ale ML a fost studiul comportamentului cumpărătorului în supermarket. Astfel, un anumit trend al traiectoriilor cumpărătorilor a fost observat, iar descrierea acestuia reprezintă o descriere structurală a comportamentului clienţilor magazinului. Această descriere este foarte utilă, deoarece putem face predicţii cu privire la mărfurile care vor avea succes la cumpărători, iar magazinul poate dezvolta o politică eficientă a stocurilor şi achiziţiilor. Totuşi, acest rezultat poate fi obţinut prin modelare statistică. Avantajul net oferit de ML este posibilitatea de înţelegere a descrierii obţinute. Astfel, după înţelegerea structurii, vânzătorul va putea dezvolta o politică de marketing eficientă, promova anumite mărfuri, expunându-le în zonele vizitate de cei mai mulţi clienţi predispuşi a cumpăra un anumit produs. De o atenție

specială s-au bucurat şi aplicațiile ML în bioinformatică [19, 20], subiect remarcabil în literatură datorită importanței practice a metodelor dezvoltate.

O parte importantă a inteligenţei umane şi consecutiv, oricărui proces de învăţare este recunoaşterea formelor. Este un proces pe care fiecare om îl desfăşoară de nenumărate ori în fiecare zi. De la recunoaşterea fizionomiei mamei, la recunoaşterea persoanelor cunoscute mai târziu, sau chiar empatia cu persoane absolut străine, oamenii execută procese de recunoaştere a formelor. Diferenţierea între fructele pe care oamenii le cumpără când merg la piaţă, pe baza aspectului lor, este un alt exemplu de recunoaştere a formelor. Decizia de-a fugi de un câine sau a îl mângâia are la bază un alt proces de recunoaştere a formelor, iar exemplele pot continua. Procesul de recunoaştere a formelor se bazează pe experienţa anterioară, deoarece mintea umană clasifică ceea ce este perceput asemănător celei mai apropiate reprezentări deja cunoscute. Ştim că fructul din faţa noastră este măr deoarece l-am mai văzut, îi cunoaştem gustul. Putem să-l diferenţiem de o pară deoarece avem experienţă în privinţa perelor. Probabil, un om înfometat, care are în jurul său mere şi guava, sau orice alte fructe care nu-i sunt familiare, va mânca merele, iar fructele necunoscute, i-ar scăpa chiar observaţiei.

Dacă inteligenţa umană este foarte utilă pentru astfel de acţiuni din fiecare moment, nu avem posibilitatea de-a discerne cu uşurinţă între un om sănătos şi un pacient bolnav de cancer şi stabili cu certitudine tipul şi gradul maladiei de care suferă pentru a-i administra tratamentul potrivit. Desigur, semiologia medicală oferă unelte utile pentru a bănui un anumit diagnostic, aceste bănuieli pot fi ulterior verificate prin teste imagistice şi nenumărate examene de laborator, dar totuşi, pentru foarte multe boli, este încă foarte dificil a pune un diagnostic acurat în timp util, cu şansa unui tratament prompt şi eficient. Multe dintre studiile de ADN microarray au finalitatea de-a oferi unelte de diagnostic precoce pentru maladii greu de diagnosticat clasic. Ideea de bază este compararea expresiilor genetice ale ţesuturilor sănătoase şi bolnave, în speranţa de a determina o structură a genelor exprimate diferenţial cu obiectivul de-a înţelege procesul fiziopatologic şi a obţine unelte de diagnostic precoce şi tratament eficient. Datorită vastităţii datelor rezultate din studiile de ADN microarray şi finalităţii cercetătorilor, ML a fost folosit cu succes pentru determinarea unei descrieri structurale a datelor. Subdomeniul ML care oferă uneltele necesare descoperirii formelor în studiile de ADN microarray este **Pattern Recognition** (PR). Metodele recunoașterii formelor sunt utilizate pe larg în analiza datelor ADN microarray, spectrometria de masă și experimentele de secvențializare din noua generație. PR poate fi definit ca procesul de achiziționare a datelor brute și acționare pe baza categoriilor de date. Recunoașterea formelor [21–23] oferă soluții pentru învățarea unui fenomen pe baza unor exemple care îl descriu cu ajutorul unor proprietăți specifice, oferă răspunsuri aplicate unei probleme individuale de detectare a formelor într-un anumit context.

PR are ca finalitate, obţinerea unei descrieri structurale, concretizată într-o clasificare pe baza informaţiilor statistice care pot fi obţinute şi a cunoştinţelor a priori despre datele studiate. În teza de faţă ne vom concentra asupra metodelor PR aplicabile în studiile de ADN microarray. Evaluarea și confirmarea utilității metodelor de recunoaștere a formelor în bioinformatică a fost tratată pe larg [24, 25] în literatură. Duda, Hart şi Stork [26], propun o formă generală a sistemelor de recunoaştere a formelor. În viziunea lor, un astfel de sistem este format dintr-un senzor, o metodă de extragere a atributelor importante ale obiectelor studiate şi un clasificator pentru sortarea datelor în funcţie de atributele considerate ca fiind caracteristice.

2.2.1. Similaritate şi di-similaritate

Studiul de faţă are ca finalitate îmbunătăţirea metodelor de discriminare între pacienţi suferinzi de diferite boli sau între subiecţi bolnavi şi sănătoşi, prin tehnicile oferite de analiza genetică. Mai exact, ne propunem să perfecţionăm metodele de PR pentru detectarea formelor expresiilor genetice specifice anumitor afecţiuni. Ulterior, ne dorim să folosim formele descoperite pentru a putea clasifica un nou subiect. Practic, ne propunem să detectăm asemănări specifice numai pacienţilor cu boala studiată sau, echivalent, să detectăm diferenţe constante între pacienţii cu o anumită afecţiune şi indivizii sănătoşi, concretizate prin genele diferenţial exprimate.

Aşadar, un prim aspect foarte important pentru demersul nostru este stabilirea unor metode de a exprima cantitativ, măsurabil, asemănarea sau deosebirea dintre doi indivizi din punct de vedere al profilurilor expresiilor genetice. Datele rezultate în urma unei analize de ADN microarray se prezintă ca matrice de numere reale cu dimensiuni foarte mari. În aceste matrice, în general, pe linii sunt reprezentate exemplele (fiecare chip), iar pe coloane sunt reprezentate atributele (expresiile probelor). Fiecare exemplu poate fi reprezentat ca un vector de coordonate cu valori numerice reale, fiecare coordonată corespunzând unui atribut în spaţiul vectorial abstract. Ne interesează aşadar, măsurile cantitative de similaritate sau di-similaritate aplicabile acestor obiecte, reprezentate de vectori de coordonate cu valori reale.

Similaritatea este o măsură cantitativă a asemănării sau apropierii dintre două instanţe, iar di-similaritatea reprezintă măsura cantitativă a diferenţei sau distanţei dintre două exemple. Cele două măsuri, aplicate aceleiaşi comparaţii trebuie să fie complementare spre aceeaşi concluzie. Altfel spus, dacă două obiecte de comparat sunt foarte similare, măsura di-similarităţii lor trebuie să fie foarte mică. Dacă însă, două obiecte de comparat sunt foarte diferite, măsura di-similarităţii lor trebuie să fie mare, iar măsura similarităţii lor trebuie să fie foarte redusă. Este aşadar natural ca relaţia dintre două obiecte să poată fi exprimată echivalent prin măsuri ale similarităţii sau di-similarităţii.

Diferenţa sau distanţa dintre două obiecte este, în general, mai uşor de calculat numeric, aşadar vom porni expunerea noastră cu prezentarea măsurilor di-similarităţii. În matematică, măsura di-similarităţii elementelor unui set este dată de metrică. Prin definiţie, metrica [27, 28], distanţa sau funcţia distanţă este o funcţie $d : X \times X \to R$, care pentru $\forall x, y, z$ în X are proprietăţile:

1. $d(x, y) \geq 0$ - distanţa este întotdeauna pozitivă,
2. $d(x, y) = 0$ dacă şi numai dacă $x = y$ - distanţa este zero dacă şi numai dacă este măsurată între un obiect şi el însuşi,
3. $d(x, y) = d(y, x)$ - simetrie,
4. $d(x, z) \leq d(x, y) + d(y, z)$ - subaditivitate sau satisface inegalitatea triunghiului. (2.1)

Dacă atât măsura similarităţii (S_{PQ}) cât şi a disimilaritatăţii (D_{PQ}) dintre două obiecte *P* şi *Q* sunt exprimate prin măsuri normalizate în intervalul $[0,1]$, între cele două cantităţi există relaţia $D_{PQ} = 1 - S_{PQ}$.

2.2.1.1. Distanţa euclidiană

Prin definiţie, distanţa euclidiană dintre două puncte A cu coordonatele $(a_1, a_2, ..., a_n)$ şi B cu coordonatele $(b_1, b_2, ..., b_n)$ într-un spaţiu euclidian n-dimensional este dată de formula:

$$D_{Euclidian\breve{a}AB} = \left(\sum_{i=1}^{n} | a_i - b_i |^2 \right)^{1/2} \quad (2.2)$$

Intuitiv, această distanţă este rezultatul măsurării cu liniarul între două puncte şi reprezintă o generalizare a teoremei lui Pitagora.

Pentru a obţine o reprezentare grafică plastică, vom calcula distanţa euclidiană dintre două puncte bidimensionale, A(2,2) şi B(10,10). Distanţa se calculează conform formulei (2.2). Figura (2.2) oferă reprezentarea grafică a distanţei euclidiene dintre punctele A şi B.

2.2.1.2. Distanţa Manhattan

Distanţa Manhattan mai poartă şi numele de City Block Distance sau Taxicab Distance, deoarece reprezintă distanţa parcursă de un taxi între două puncte ale unui oraş cu o parte din străzi paralele şi perpendiculare pe restul. Deoarece Manhattan-ul are o astfel de configuraţie, distanţa este cunoscută cu numele acestei insule din New York.

Prin definiţie, distanţa Manhattan dintre două puncte A cu coordonatele $(a_1, a_2, ..., a_n)$ şi B cu coordonatele $(b_1, b_2, ..., b_n)$ într-un spaţiu euclidian n-dimensional este dată de formula:

$$D_{ManhattanAB} = \sum_{i=1}^{n} | a_i - b_i | \quad (2.3)$$

Vom ilustra distanţa Manhattan dintre aceleaşi puncte bidimensionale, A(2,2) şi B(10,10), pentru a obţine o reprezentare grafică, în plan, a di-similarităţii. Figura (2.3) oferă vizualizarea grafică a distanţei Manhattan dintre punctele A şi B. Se observă că cele două variante desenate cu roşu au, în valoare absolută, aceeaşi valoare, egală cu distanţa Manhattan dintre punctele A şi B.

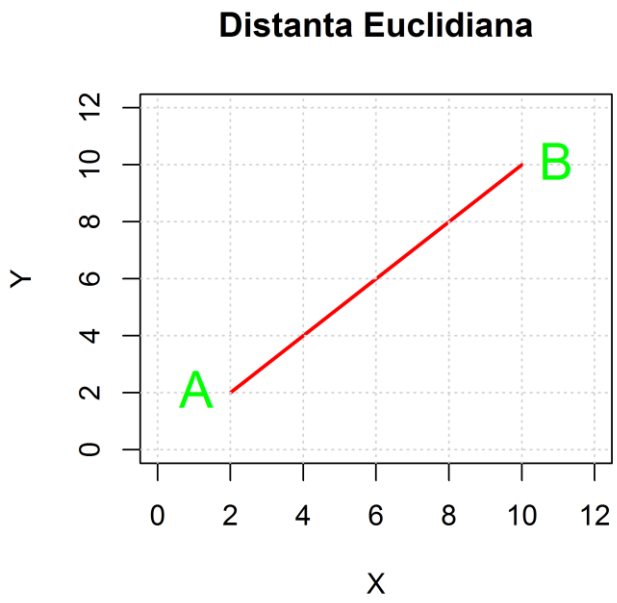


Fig. 2.2 - Distanţa euclidiană dintre două puncte bidimensionale A(2,2) şi B(10,10).

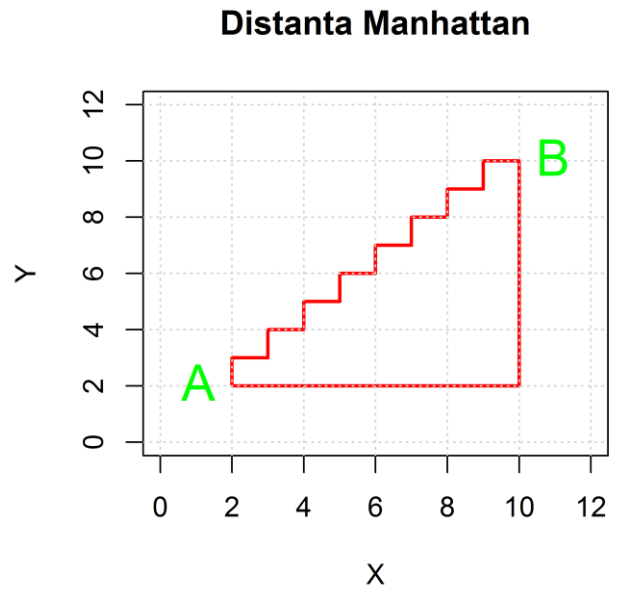


Fig. 2.3 – Distanţe Manhattan dintre două puncte bidimensionale P(2,2) şi Q(10,10). Cele două trasee reprezentate cu roşu sunt echivalente şi egale cu distanţa Manhattan dintre cele două puncte.

2.2.1.3. Distanţa Minkowski

Prin definiţie, distanţa Minkowski de ordinul *k* dintre două puncte A cu coordonatele $(a_1, a_2, ..., a_n)$ şi B cu coordonatele $(b_1, b_2, ..., b_n)$ într-un spațiu euclidian n-dimensional este dată de formula:

$$D_{Minkovski^k AB} = \left(\sum_{i=1}^{n} | a_i - b_i |^k \right)^{\frac{1}{k}} \quad (2.4)$$

Distanţa Minkowski reprezintă metrica generalizată în spaţiul euclidian. Dacă înlocuim k=1 sau k=2, obţinem distanţa Manhattan, respectiv euclidiană. Un mare avantaj al distanţei Minkowski este că ea poate fi utilizată atât pentru calculul distanţelor între exemple cu atribute numerice, cât şi pentru cele cu atribute nominale.

2.2.1.4. Distanţa Chebyshev

Poartă numele matematicianului rus Pafnuty Lvovich Chebyshev. Mai este cunoscută şi ca distanţa tablei de şah, deoarece reprezintă numărul minim de mutări pe care un rege trebuie să-l execute pentru a ajunge într-un anumit pătrat, de pe tabla de şah.

Prin definiţie, distanţa Chebyshev dintre două puncte A cu coordonatele $(a_1, a_2, ..., a_n)$ şi B cu coordonatele $(b_1, b_2, ..., b_n)$ într-un spaţiu euclidian n-dimensional este dată de formula:

$$D_{ChebyshevAB} = \lim_{k \to \infty} \left(\sum_{i=1}^{n} | a_i - b_i |^k \right)^{1/k} = \max_{i=0}^{n} | a_i - b_i | \quad (2.5)$$

Formula distanţei Chebyshev poate fi de asemenea obţinută din formula distanţei Minkowski, pentru cazul particula când $k = \infty$.

2.2.1.5. Distanţa Canberra

Prin definiţie, distanţa Canberra dintre două puncte A cu coordonatele $(a_1, a_2, ..., a_n)$ şi B cu coordonatele $(b_1, b_2, ..., b_n)$ într-un spaţiu euclidian n-dimensional este dată de formula:

$$D_{CanberraAB} = \sum_{i=1}^{n} \frac{| a_i - b_i |}{| a_i | + | b_i |} \quad (2.6)$$

Funcţia *dist* în pachetul stats din R oferă posibilitatea calculării distanţelor menţionate.

2.2.1.6. Coeficientul de corelaţie

Coeficientul de corelaţie reprezintă o măsură a similarităţii dintre obiecte. Prin definiţie, coeficientul de corelaţie a două puncte A cu coordonatele $(a_1, a_2, ..., a_n)$ şi B cu coordonatele $(b_1, b_2, ..., b_n)$ este dată de formula:

$$r = \frac{\sum_{i=1}^{n}\left(a_i - \bar{a}\right)\left(b_i - \bar{b}\right)}{\left[\sum_{i=1}^{n}\left(a_i - \bar{a}\right)^2 \sum_{i=1}^{n}\left(b_i - \bar{b}\right)^2\right]^{\frac{1}{2}}}, \text{ unde } \bar{a} = \frac{1}{n}\sum_{i}^{n} a_i \text{ , iar } \bar{b} = \frac{1}{n}\sum_{i}^{n} b_i \qquad (2.7)$$

Coeficientul de corelaţie poate fi văzut drept măsură a separaţiei angulare dintre doi vectori, ale căror coordonate au fost standardizate la medie. Asemenea cosinusului, coeficientul de corelaţie poate lua valori în intervalul [-1,1]. Valoarea 1 a coeficientului de corelaţie indică situaţia când vectorii P şi Q au aceeaşi direcţie şi acelaşi sens. Valoarea -1 a coeficientului de corelaţie indică situaţia când vectorii P şi Q au aceeaşi direcţie şi dar sens opus.

Funcţia *cor*, din pachetul stats, oferă posibilitatea calculării coeficientului de corelaţie. R pune la dispoziţia utilizatorului metode simple şi foarte sugestive de a vizualiza asemănările şi deosebirile dintre instanţe, bazate doar pe măsurile expuse anterior.

2.2.1.7. Impactul măsurilor similarității și di-similarității

Distanţele euclidiană, Manhattan şi Chebyshev sunt cazuri particulare ale distanţei Minkowski (tabelul 2.1) pentru diferite ordine (k). În plus, distanţa Minkowski poate fi utilizată şi pentru atribute nominale, deşi această necesitate apare relativ rar în studiile de ADN microarray. Distanţa Canberra este utilă în special când toate valorile atributelor sunt foarte apropiate de zero, deoarece este mai sensibilă la fluctuaţiile mici. În plus, distanţele Minkowski, euclidiană, Manhattan, Chebyshev şi Canberra reprezintă măsuri ale di-similarităţii, în timp ce coeficientul de corelaţie cuantifică similaritatea.

Rezultate comparative ale distanţelor dintre punctele bidimensionale A(2,2) şi B(10,10), calculate cu diferitele distanţe discutate anterior, sunt prezentate în tabelul 2.2.

În studiile de ADN microarray urmărim descoperirea unor forme relevante în colecţii de chipuri conţinând un număr imens de oligonucleotide. Ne aşteptăm aşadar ca, într-o astfel de colecţie, să existe similitudini între chipurile reprezentând ţesuturi sănătoase. Ne aşteptăm de asemenea, să existe asemănări între cipurile reprezentând ţesuturi specifice bolii studiate şi ar fi normal să existe un grad de di-similaritate ridicat între chip-urile aparţinând celor două grupuri.

Tabel 2.1 – Măsuri ale di-similarităţii

Distanţă	Formulă	Ordin Minkowski
Minkowski	$D_{Minkovski^k AB} = \left(\sum_{i=1}^{n} \mid a_i - b_i \mid^k\right)^{\frac{1}{k}}$	k
Euclidiană	$D_{EuclidianăAB} = \left(\sum_{i=1}^{n} \mid a_i - b_i \mid^2\right)^{1/2}$	k=2
Manhattan	$D_{ManhattanAB} = \sum_{i=1}^{n} \mid a_i - b_i \mid$	k=1
Chebyshev	$D_{ChebyshevAB} = \lim_{k\to\infty} \left(\sum_{i=1}^{n} \mid a_i - b_i \mid^k\right)^{1/k} = \max_{i=0}^{n} \mid a_i - b_i \mid$	k= ∞
Canberra	$D_{CanberraAB} = \sum_{i=1}^{n} \frac{\mid a_i - b_i \mid}{\mid a_i \mid + \mid b_i \mid}$	-

În general, măsurarea similarităţii sau di-similarităţii dintre instanţe nu va fi suficientă pentru a descoperi forme în seturile de date ADN microarray, însă ele pot oferi informaţii foarte valoroase în faza de preprocesare a datelor şi pot evidenţia exemplele cu valori extreme care pot ascunde erori şi necesită analiză suplimentară.

O parte dintre algoritmii de grupare sau clasificare sunt foarte sensibili la măsura utilizată pentru determinarea similarității sau di-similarității. În cazul utilizării unor astfel de metode, este foarte importantă această alegere.

Într-un studiu de ADN microarray, cel mai uzual, ne interesează asemănarea sau deosebirea dintre două condiții, în termeni de gene diferenţial exprimate pe chipurile corespunzătoare. Spre exemplu, să presupunem că ne interesează să stabilim asemănările sau deosebirile dintre 4 pacienţi, pe baza a 5 atribute din setul de date Golub [29]. Datele Golub sunt accesibile pe situl proiectului Bioconductor și reprezintă măsurători ale expresiei genetice cu 7,129 de probe, pentru 72 de pacienți suferinzi de leucemie acută. Cazurile discriminează două clase, în funcție de diagnostic: leucemie acută limfoblastică (ALL) și leucemie acută mieloidă (AML).

Tabel 2.2 – Distanțe între punctele A(2,2) și B(10,10)

Distanţă	Valoare numerică
Minkowski	10.07937
Euclidiană	11.31371
Manhattan	16
Chebyshev	8
Canberra	1.33

Tabelul 2.3 prezintă 5 atribute ale exemplelor cu numerele de identificare 56, 59, 52 şi 53 din setul de date. Utilizăm aceste date pentru a explora diferitele modalităţi de a calcula măsurile cantitative ale similarităţii sau di-similarităţii. Tratăm cele patru exemple (chipuri) ca puncte într-un spaţiu cu 5 dimensiuni corespunzătoare celor cinci atribute. Astfel, chipul corespunzător exemplului cu numărul de identificare 56 este reprezentat de punctul cu coordonatele (257, 1377, 198, 59, 509) în spaţiul 5-dimensional asociat atributelor luate în considerare. Figura 2.4, ilustrează măsurile similarităţii şi di-similarităţii pentru datele din tabelul 2.3.

O vizualizare foarte sugestivă a similarităţii/di-similarităţii dintre cele 72 exemple (chipuri) din setul de date Golub, pe baza celor 7129 de atribute, calculate cu măsurile prezentate anterior, este oferită în Fig. 2.5.

Diferitele măsuri ale similarităţii sau di-similarităţii prezentate anterior reprezintă opţiuni în mâna cercetătorului. Nu există un standard unanim acceptat în privinţa unei anumite măsuri de similaritate, potrivită tuturor studiilor de ADN microarray. Datorită diversităţii datelor rezultate în urma unui astfel de experiment, este foarte importantă explorarea datelor şi stabilirea măsurilor similarităţii care descriu convenabil fiecare situaţie şi sunt potrivite metodelor de învăţare ce vor fi utilizate.

Este evident că măsurile disimilarităţilor sunt foarte diferite, dar aceste rezultate au valoare pur ilustrativă şi nu sunt foarte sugestive. În explorarea unui set de date format din instanţe şi atribute mult mai numeroase, este esenţial ca o măsură a di-similarităţii să ilustreze cât mai bine diferenţele dintre exemplele aparţinând diferitelor clase. Este aşadar recomandabilă vizualizarea rezultatelor obţinute cu diferite distanţe ca în fig. 2.5 şi stabilirea unei măsuri a similarităţii sau di-similarităţii care descrie cel mai bine datele studiate şi sunt potrivite metodelor de învăţare alese.

Tabel 2.3 – subset din setul de date Golub.

ID	AB000449_at	AB002559_at	AC000064_cds1_at	AF005043_at	AF006084_at
56	257	1377	198	59	509
59	195	1357	183	25	963
52	-41	857	141	17	3084
53	90	1872	682	7	1230

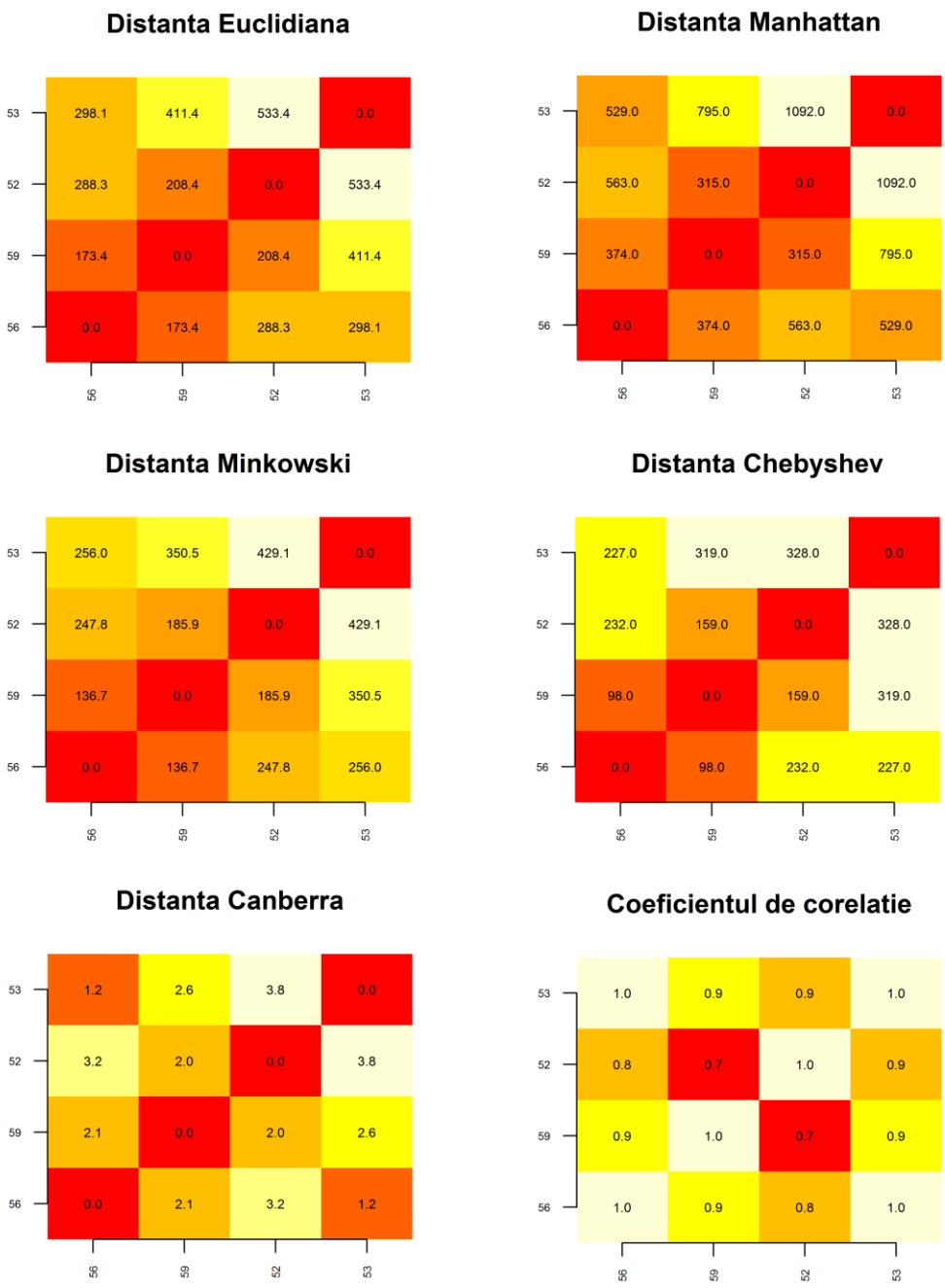


Fig. 2.4 – Măsuri ale similarităţii/di-similarităţii între instanţele setului Golub.

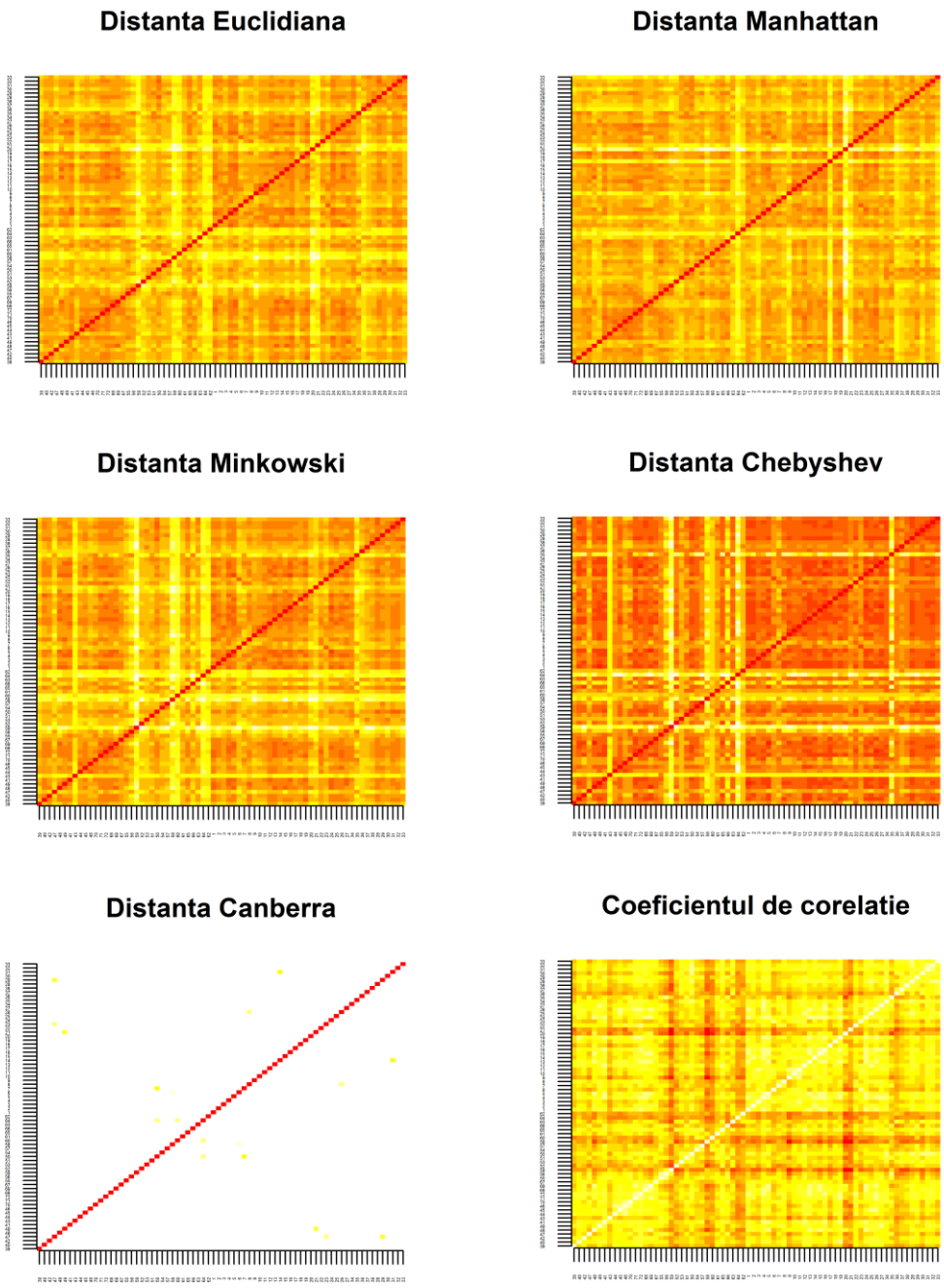


Fig. 2.5 – Măsuri ale similarităţii/di-similarităţii între instanţele setului Golub.

2.2.2. Tehnici de grupare

În general, metodele inteligenţei artificiale au corespondent în aspecte ale raţionamentului uman. Fiecare dintre noi suntem puşi, de mai multe ori pe zi, în situaţia de a efectua grupări ale obiectelor din jurul nostru. Oamenii ordonează obiectele din mediul înconjurător, pe baza similarităţii dintre ele, a utilităţii lor sau a frecvenţei de utilizare. Obiectele care ţin de vestimentaţie sunt păstrate în dulap, iar acolo ele sunt aranjate după interesul posesorului. Probabil că hainele utilizate mai des vor avea un loc uşor accesibil, hainele de vară vor fi grupate laolaltă, separat de cele de iarnă sau costumele vor fi aşezate în apropierea cămăşilor.

Gruparea reprezintă trierea datelor pe baza unor similarităţi naturale între obiecte. Tehnicile de grupare reprezintă metode de învăţare în context nesupervizat, omiţând clasele din care instanţele fac parte. Finalitatea grupării este descoperirea unor tendinţe naturale, similitudini sau deosebiri între obiectele studiate şi trierea lor în grupuri corespunzătoare. Rezultatul învăţării este separarea instanţelor studiate în grupuri ce poartă numele de clustere.

În cadrul unui studiu de ADN microarray poate fi interesantă gruparea instanţelor cât şi gruparea atributelor. Este deopotrivă interesantă descoperirea similitudinilor între cipuri cât şi decelarea unor trenduri în nivelele de expresie ale diferitelor gene. Dacă prima abordare oferă informaţii cu privire la similitudinea dintre pacienţii cu diferite diagnostice, a doua metodă poate oferi informaţii foarte valoroase în privinţa tendinţelor nivelelor de expresie a unor gene în diferite etape ale evoluţiei unei afecţiuni.

2.2.2.1. Gruparea ierarhică

Tehnicile de grupare ierarhică au ca finalitate descrierea unei ierarhii de gupe (clustere) cu elementele unei mulţimi, pe seama similitudinilor sau diferenţelor dintre obiecte. Rezultatul unei astfel de tehnici este reprezentat de o ierarhie de clustere şi nu o grupare rigidă a obiectelor într-un număr prestabilit de grupe. Avantajul acestei abordări constă în valoarea exploratorie a metodelor, fără a fi necesară nici o presupunere asupra numărului de clustere existente în datele studiate.

Există două strategii de grupare ierarhică: aglomerativă şi divizivă. *Strategia aglomerativă* presupune iniţializarea grupării cu un număr de clustere egal cu numărul obiectelor studiate, fiecărui obiect corespunzându-i propriul cluster şi unirea succesivă a clusterelor. *Strategia divizivă* presupune iniţializarea grupării cu singur cluster care conţine toate obiectelor studiate şi divizarea succesivă a obiectelor pe clustere corespunzătoare.

Modul de unire sau divizare a clusterelor corespunzătoare metodelor aglomerativă şi respectiv divizivă, presupun o măsură cantitativă a similarităţii sau di-similarităţii. Oricare dintre măsurile similarităţii prezentate în capitolul 2 pot servi acestui scop. Evaluăm distanţe între clustere conţinând mai multe obiecte, este aşadar necesară stabilirea unui criteriu de evaluare a di-similarităţii dintre clustere pe seama distanţei dintre obiectele componente. Cele mai comune criterii de evaluare sunt:

- *Single linkage* – evaluează distanţa dintre două clustere ca fiind egală cu cea mai mică distanţă dintre două obiecte aparţinând grupurilor diferite. Similaritatea dintre două clustere este egală cu cea mai mare măsură cantitativă a similarităţii dintre un obiect aparţinând primului cluster şi unul care face parte din al doilea (Fig. 2.6 b)),
- *Complete linkage* – evaluează distanţa dintre două clustere drept egală cu cea mai mare distanţă dintre două obiecte aparţinând grupurilor diferite (Fig. 2.6 c)),
- *Average Linking* – evaluează distanţa dintre două clustere ca fiind egală cu media distanţelor dintre toate perechile de obiecte aparţinând grupelor diferite,
- *Centroid Linkage* – evaluează distanţa dintre două clustere drept egală cu distanţa dintre centroizii fiecărui cluster (Fig. 2.6 d)),
- *Ward's Criterion* [30] – evaluează creşterea varianţei în clusterul creat,
- *V-Linkage* – evaluează probabilitatea ca cele două clustere să provină din aceeaşi distribuţie.

Algoritmul nu oferă posibilitatea de-a corecta o eroare dintr-un pas anterior. Odată ce o uniune sau divizare a fost efectuată, chiar dacă operaţiunea nu este conformă cu realitatea, nu se mai poate reveni asupra erorii. În plus, obiectul atribuit eronat unui anumit cluster, va influenţa paşii următori.

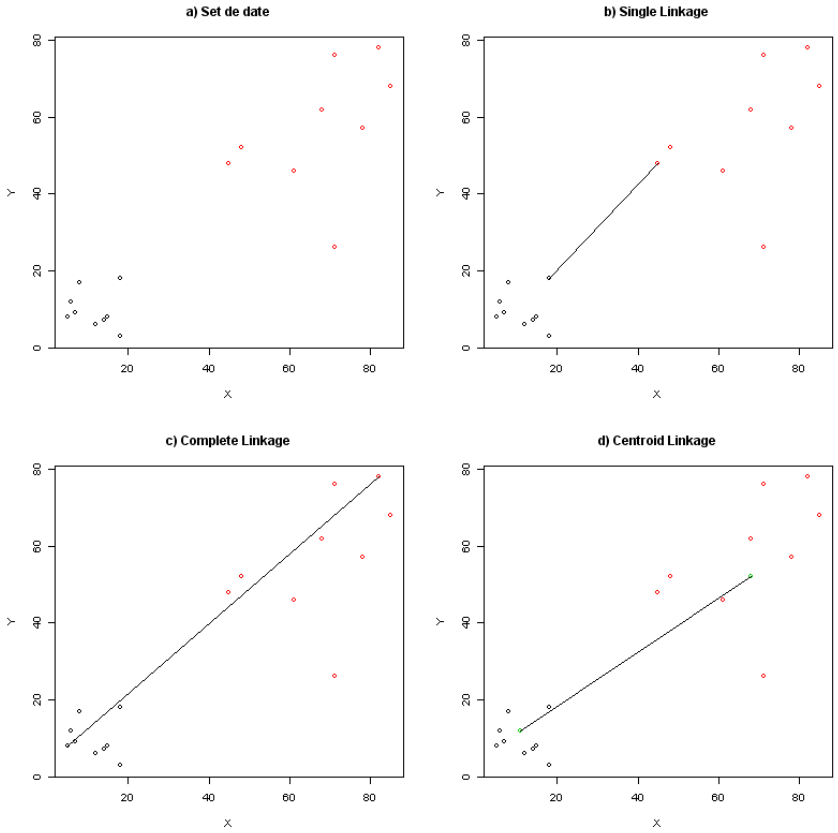


Fig. 2.6 – Criterii de evaluare uzuale

1. fiecare obiect din setul de date este considerat drept cluster, aşadar, numărul de clustere este egal cu numărul obiectelor din set.
2. cele mai similare două clustere sunt unite într-unul singur, numărul clusterelor rămase scade cu 1
3. calculează similarităţile dintre clusterele rămase,
4. repetă paşii 2-3 până când rămâne un singur cluster (sau numărul dorit de clustere).

Criteriul de evaluare ales este foarte important şi criterii diverse vor duce la rezultate complet diferite. Obiectele foarte apropiate aparţinând unor grupe diferite, structurile convexe, concave sau încapsulate ale datelor reprezintă provocări serioase pentru gruparea ierarhică.

Rezultatul unei grupări ierarhice este reprezentat grafic sub forma unei diagrame cu structură de arbore, numită *dendogramă*. Într-o dendogramă rădăcina reprezintă clusterul conţinând toate obiectele studiate, iar frunzele reprezintă obiectele studiate. Fiecare nod reprezintă un pas în procesul de grupare ierarhică, uniune sau diviziune. Dendogramele sunt, în general, însoţite de o scară a similarităţii. Folosind această scară, arborele poate fi tăiat pentru a obţine numărul de clustere dorit sau gradul de similaritate considerat acceptabil.

Proiectul R oferă diferite implementări ale grupării ierarhice. Pachetele *stats* şi *MLInterfaces* oferă funcţii foarte flexibile pentru aplicarea grupării ierarhice, cu diferitele variaţii, în analiza genetică. Vom exemplifica în continuare metodele grupării ierarhice aplicate setului de date Golub (Fig. 2.7).

Un aspect foarte important în gruparea cu oricare dintre metodele prezentate îl reprezintă alegerea măsurii similarității utilizate. În general, se știu prea puține informații despre datele rezultate în urma unui experiment microarray, pentru a se putea decide care măsură pentru similaritate este cea mai potrivită. Obținerea aspectului grupării cu diferite metode și variate măsuri ale similarității este dezirabilă pentru abordarea nesupervizată a unui astfel de set de date. Ilustrăm în Fig. 2.8 impactul asupra grupării cu metoda complete, al diferitelor măsuri ale similarității. Rezultatul grupării utilizând distanța euclidiană este prezentat în Fig. 2.7.

Se poate observa cu uşurinţă că metodele grupării ierarhice, deşi foarte intuitive, pot fi eficiente. Deşi toate dendogramele de mai jos sunt rezultate cu distanţe euclidiene, diferenţele de performanţă sunt foarte mari şi se datorează criteriilor diferite de evaluare. Este evident că, pentru această problemă, criteriile Complete Linkage şi Ward oferă cele mai satisfăcătoare rezultate.

Cunoaştem a priori că avem de-a face cu două grupe de pacienţi, aşadar, putem tăia dendograma corespunzător. O metodă de a verifica grafic validitatea unei metode de grupare este oferită de siluetă (Silhouette). Silueta ilustrează clusterele rezultate, oferind o imagine asupra similarităţii obiectelor atribuite aceluiaşi cluster şi a di-similarităţii lor faţă de celelalte clustere. Utilizând această metodă pentru a aprecia rezultatul grupării ierarhice cu criteriul Complete Linkage (Fig. 2.9).

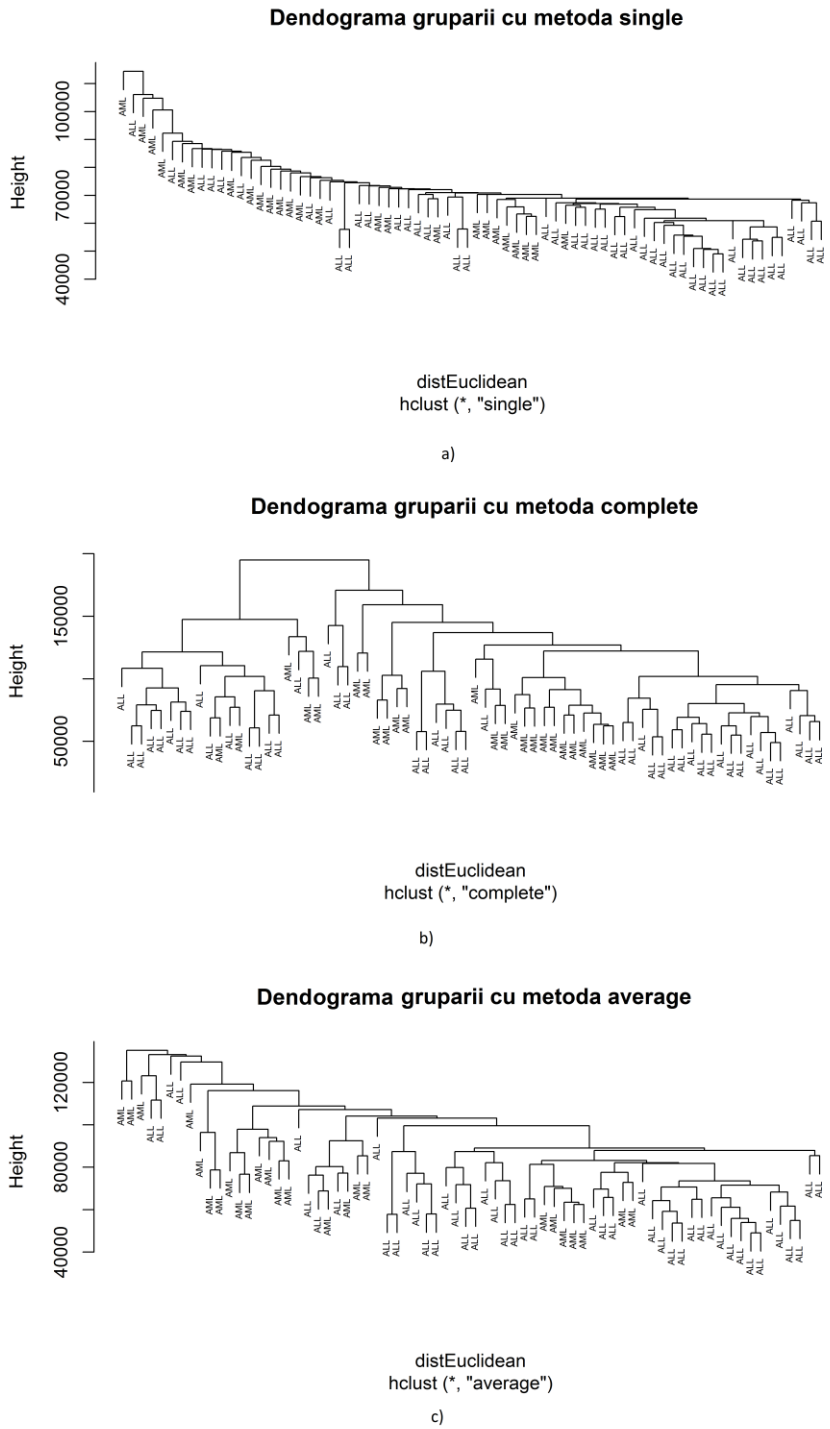
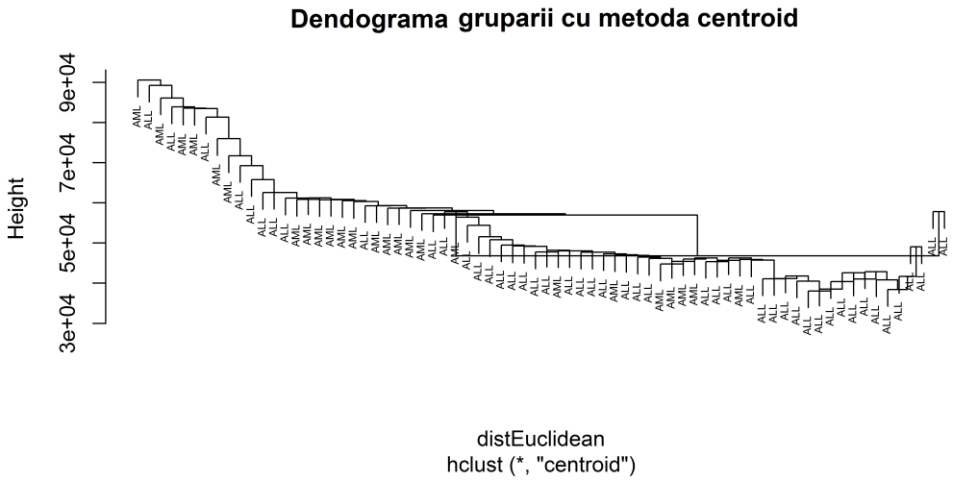
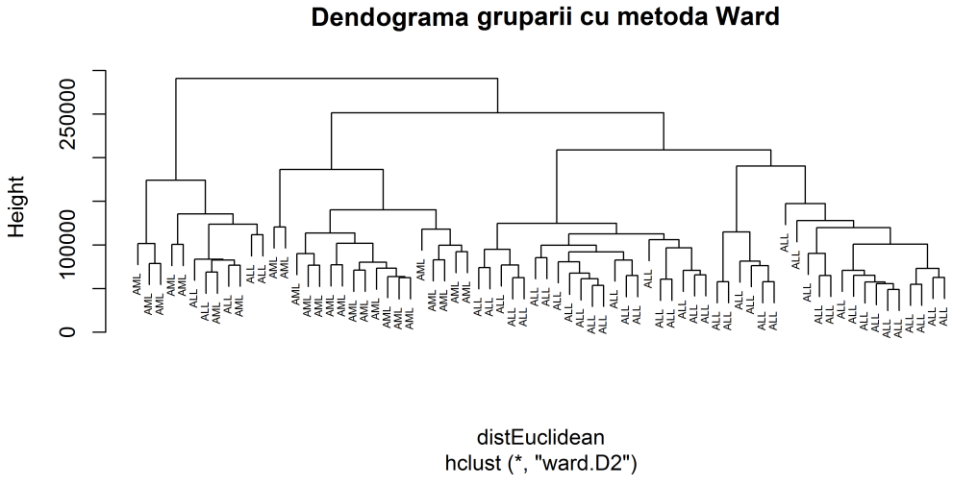


Fig. 2.7 - Dendograme rezultate în urma grupărilor ierarhice cu diferite criterii cu datele Golub.



d)



e)

Fig. 2.7 (continuare) – Dendograme rezultate în urma grupărilor ierarhice cu diferite criterii cu datele Golub.

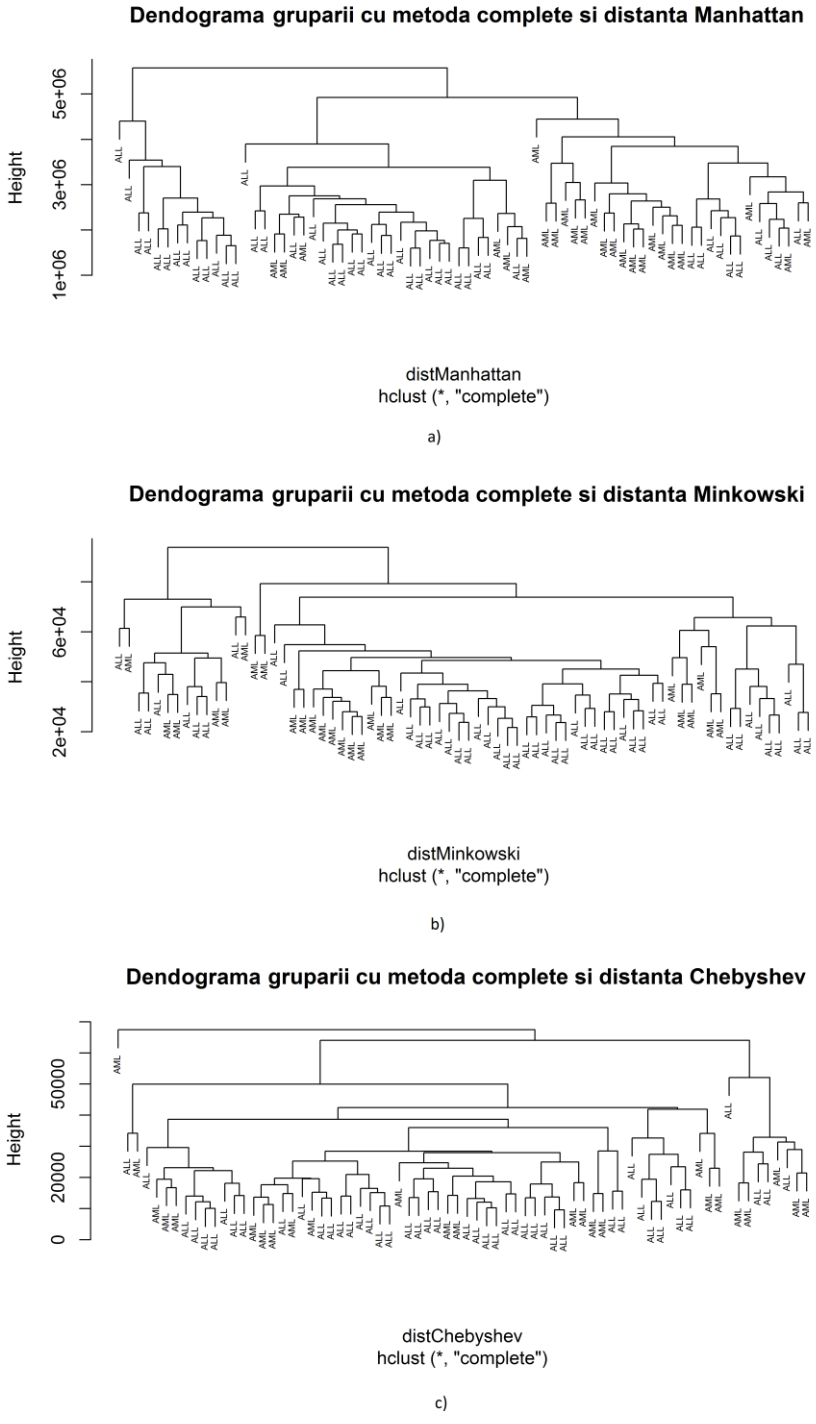


Fig. 2.8 - Dendograme rezultate în urma grupărilor ierarhice cu diferite criterii cu datele Golub.

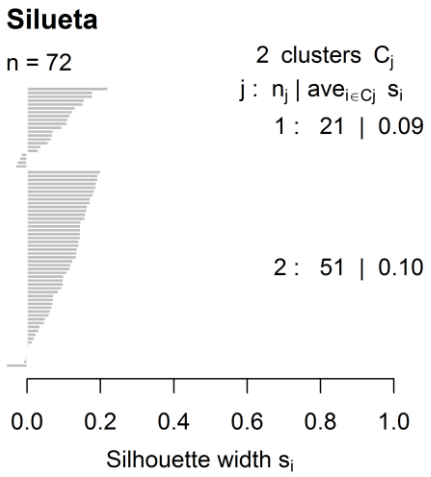


Fig. 2.9 – Silueta grupării ierarhice pe datele Golub, cu criteriul Complete-Linkage

O metodă puternică de vizualizare (Fig. 2.10) a datelor de ADN microarray, concomitent cu gruparea lor prin clusterizare ierarhică este oferită de funcţia `heatmap()` din pachetul *stats*.

Variantele agnes şi diana oferite în pachetul cluster, pot fi utilizate pentru clusterizarea ierarhică. Diferenţa majoră între cele două abordări este construcţia arborelui de tip aglomerativ (bottom-up), în cazul agnes sau diviziv (top-down), pentru diana. Rezultate exemplificative, obţinute cu aceste funcţii pe setul de date Golub, sunt ilustrate în Fig. 2.11 şi 2.12.

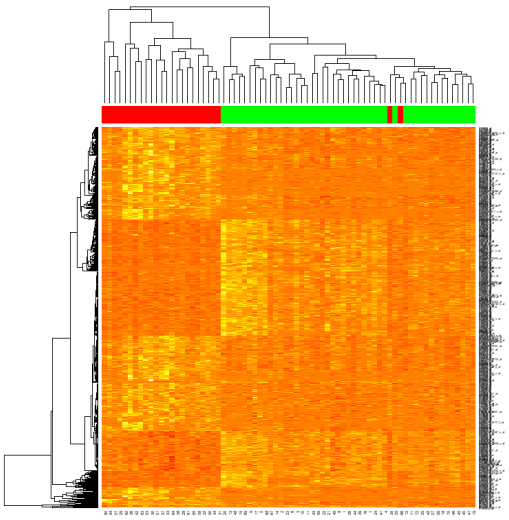
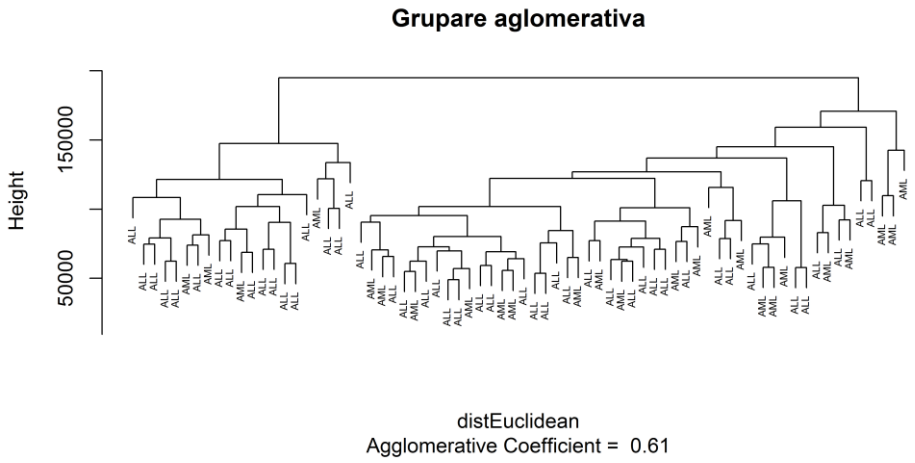
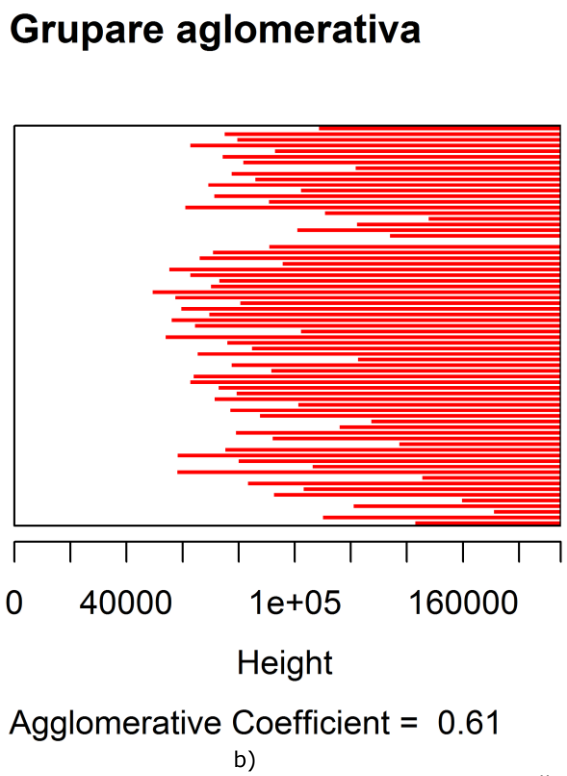


Fig. 2.10 – Heatmap pentru datele Golub.

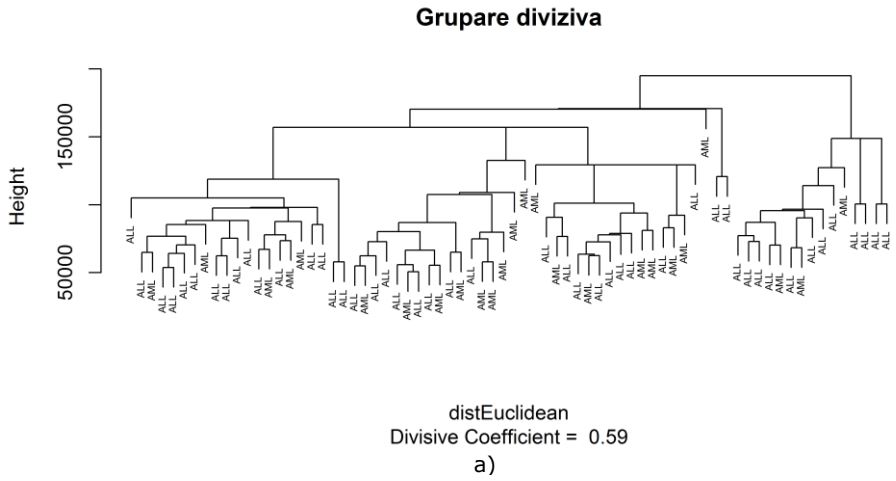


a)

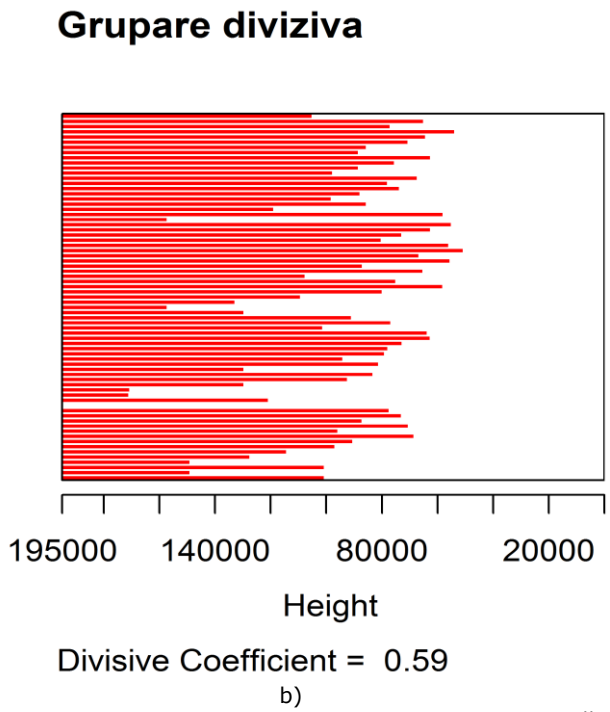


b)

Fig. 2.11 – Gruparea AGNES pe setul de date Golub, a) Dendogramă, b) Siluetă.



a)



b)

Fig. 2.12 – Gruparea DIANA pe setul de date Golub, a) Dendogramă, b) Siluetă.

2.2.2.2. Gruparea k-means

Un mare neajuns al grupării ierarhice îl reprezintă imposibilitatea de-a corecta eventualele erori din paşii anteriori. O soluţie la această problemă este abordarea propusă de tehnicile de grupare prin partiţionare. Dacă în cazul grupării ierarhice rezultatul este reprezentat de *ierarhia de clustere*, care poate fi ulterior analizată, iar utilizatorul poate decide asupra numărului de clustere existente în datele studiate, rezultatul metodelor de grupare prin partiţionare este reprezentat de *un număr prestabilit de partiţii* în datele analizate.

K-means este un algoritm iterativ, propus în 1957 de către Stuart Lloyd [31]. Iniţial, obiectele sunt grupate într-un număr de k clustere, în mod aleatoriu sau euristic. Se calculează apoi, media fiecărui cluster. Media fiecărui cluster este reprezentată de un obiect virtual, numit şi cetroid sau centru de greutate al clusterului. Iteraţiile următoare constau, fiecare, din două etape distincte. Într-o primă fază, fiecare obiect este redistribuit clusterului cu centrul de greutate cel mai apropiat. Celulele Voroni [32] în raport cu fiecare cetroid oferă o imagine plastică a modului de redistribuire a obiectelor pe clustere. Într-un al doilea pas, se calculează centrele de greutate ale noilor grupuri. Procesul continuă până când fazele de redistribuire a obiectelor în clustere nu mai produc schimbări în configuraţia grupurilor. Rezultatul este o grupare a obiectelor într-un număr prestabilit de clustere, astfel încât obiectele aparţinând unui cluster sunt similare şi diferite de obiectele din celelalte clustere.

1. partiţionează aleatoriu obiectele în k clustere,
2. asociază fiecare obiect din setul de date centroidului cel mai similar, indicat de diagramele Voroni ale centroizilor,
3. calculează centroidul fiecărui cluster rezultat,
4. repetă paşii 2-3 până când nu mai apar modificări de configuraţie.

Există dezavantaje majore ale metodei k-means. În primul rând, uneori, este foarte greu de prevăzut a priori, câte clustere cu importanţă practică există în datele studiate. Deoarece metoda impune stabilirea prealabilă a numărului de grupuri ţintă, sunt foarte importante cunoştinţele a priori despre datele analizate. În al doilea rând, deşi metoda poate corecta eventualele erori de distribuire a obiectelor pe clustere, rezultatul depinde foarte mult de repartizarea a priori, aleatorie sau euristică a obiectelor în grupele iniţiale. Astfel, există riscul de convergenţă într-un optim local. Pentru a neutraliza acest neajuns, este recomandabilă rularea repetată a algoritmului pe setul de date analizat sau folosirea unor metode evoluţioniste.

Pachetele *stats* şi *MLInterfaces* oferă implementări ale tehnicii k-means cu diferite variaţiuni. În cele ce urmează vom ilustra (fig. 2.13) utilizarea k-means (k=2) pentru analiza setului de date Golub.

```
> kmeanscl$cluster
39 40 42 47 48 49 41 43 44 45 46 70 71 72 68 69 67 55 56 59 52 53 51 50 54 57
 1  2  2  1  2  2  1  1  1  2  1  1  2  2  2  2  2  1  1  1  2  2  1  2  1  1
58 60 61 65 66 63 64 62  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
 1  1  1  2  1  2  1  1  2  2  2  2  2  1  1  2  2  2  2  2  2  2  2  2  2  2
19 20 21 22 23 24 25 26 27 34 35 36 37 38 28 29 30 31 32 33
 2  2  2  2  1  2  2  2  1  2  1  2  2  2  2  2  2  2  2  2
> tr <- table(kmeanscl$cluster, labels)
> tr
   labels
    ALL AML
  1  14  10
  2  33  15
```

Fig. 2.13 – k-means (k=2) pe setul de date Golub.

2.2.2.3. Gruparea Fuzzy k-means

O soluţie la problema esenţială a algoritmului k-means, convergenţa într-un optim local, este oferită de o variantă a algoritmului original, numită Fuzzy k-means. Faţă de abordarea k-means, această variantă atribuie fiecare obiect din setul de date, fiecărui cluster, dar cu probabilităţi diferite. Pentru setul de date Golub, algoritmul fuzzy k-means implementat în pachetul R *e1071*[33] realizează gruparea din fig. 2.14.

2.2.2.4. Gruparea k-medoid

Media unui set de date depinde de toate instanţele din colecţia de date. Aşadar, dacă în setul de date există instanţe cu valori aberante, media setului de date va fi serios afectată. Diferite tipuri de erori pot afecta datele ADN microarray, în consecinţă, valorile aberante sunt comune în seturile de date vaste. Un parametru ce poate caracteriza un set de date şi nu este foarte afectat de valorile aberante este mediana. Spre exemplu, pentru set de valori S=(1,2,3,4,5,6,7,8,105), media are valoarea 15.66667. Mediana setului S, pe de altă parte, are valoarea 5. Valoarea 105 poate fi prezentă datorită unei erori în colectarea datelor, iar o analiză pe S ar putea fi puternic afectată de valoarea extremă. Folosirea medianei în locul mediei, este aşadar o soluţie pentru a adresa problema valorilor extreme. În seturile vaste de date, această idee este concretizată prin alegerea medoidului drept caracteristică a unui cluster.

Medoidul reprezintă obiectul din cluster al cărui di-similaritate medie, faţă de toate celelalte obiecte din cluster este minimă. Pornind de la algoritmul k-means, ideea utilizării medoidului în loc de centroid, a dus la apariţia algoritmului k-medoid. Este de remarcat că, în cazul k-means, gruparea se efectua în jurul centroidului, un obiect virtual, care nu exista în realitate în setul de date analizat. În abordarea k-medoid, medoidul este reprezentat de un obiect real, prezent în setul de date, considerat caracteristic pentru un anumit cluster deoarece prezintă cea mai mică di-similaritate medie faţă de restul obiectelor din acelaşi grup.

```
> cmeansRes$cluster
39 40 42 47 48 49 41 43 44 45 46 70 71 72 68 69 67 55 56 59 52 53 51 50 54 57
 1  1  2  1  2  1  1  1  1  1  1  1  2  1  2  2  1  1  1  1  2  2  1  2  1  1
58 60 61 65 66 63 64 62  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
 1  1  1  2  1  2  1  1  2  2  2  2  2  1  1  1  2  2  2  1  2  2  2  2  2  2
19 20 21 22 23 24 25 26 27 34 35 36 37 38 28 29 30 31 32 33
 2  2  2  1  1  2  2  2  1  2  1  2  2  2  2  1  2  2  2  2
> tr <- table(cmeansRes$cluster, labels)
> tr
   labels
    ALL AML
  1  22  11
  2  25  14
```

Fig. 2.14 – Fuzzy k-means (k=2) pe setul de date Golub.

O versiune larg folosită de k-medoid este PAM (Partitioning Around Medoids) [34]. Algoritmul se iniţializează cu cele k centre alese aleator, unde k este specificat în prealabil. Fiecare obiect din setul de date este atribuit apoi unui cluster, pe criteriul distanţei minime faţă de medoizii aleşi anterior. În următorii paşi se selectează fortuit un alt medoid, dintre obiectele din setul de date. Pentru fiecare candidat de medoid, se calculează costul înlocuirii vechiului medoid cu noul candidat. Dacă acest cost este negativ, se trece la înlocuirea vechiului medoid cu noul candidat. Dacă rezultatul costului este pozitiv, vechiul medoid este păstrat şi se trece la selectarea aleatorie a unui nou candidat. Acest proces este repetat până când nu mai apar schimbări de configuraţie în setul de date.

1. selectează aleator k obiecte drept medoizi,
2. asociază fiecare obiect din setul de date medoidului cel mai similar,
3. selectează aleator un alt obiect drept medoid,
4. calculează costul reamplasării acestui medoid,
5. dacă acest cost < 0 păstrează candidatul drept medoid,
   altfel, dacă acest cost > 0 păstrează vechiul obiect drept medoid,
6. repetă paşii 2-5 până când nu mai apar modificări de configuraţie.

Pentru analiza setului de date Golub, am utilizat funcţia `pam()` din pachetul *cluster*. Am grupat pe 2 clustere, iar metrica folosită a fost distanţa euclidiană. Figura 2.15 ilustrează rezultatele obţinute prin gruparea datelor cu PAM.

2.2.2.5. Gruparea CLARA

PAM suferă de performanţe limitate când se lucrează pe seturi de date foarte vaste, cum este cazul în studiile de ADN microarray. O soluţie posibilă este oferită de implementarea CLARA. Principiul care stă la baza acestui algoritm este că un eşantion colectat aleatoriu din datele complete, este reprezentativ pentru întregul set. Este de aşteptat ca medoizii obţinuţi cu PAM din acest eşantion să fie reprezentativi pentru întregul set de date. Silueta grupării prin metoda CLARA a datelor Golub filtrate sunt prezentate în Fig. 2.16.

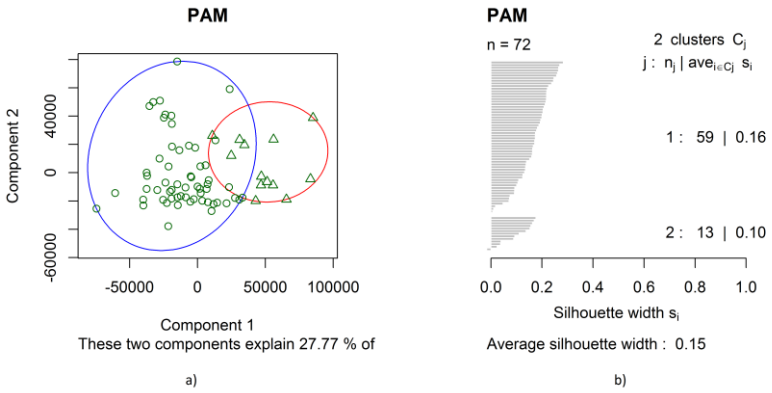


Fig. 2.15 – a) Gruparea şi b) silueta PAM pe setul de date Golub.

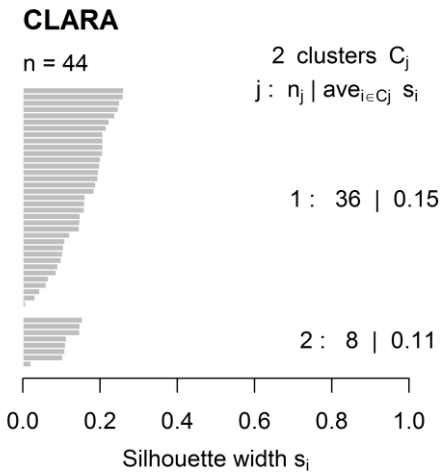


Fig. 2.16 - Silueta CLARA pe setul de date Golub.

2.2.3. Tehnici de clasificare

Studiul nostru are ca finalitate discriminarea între grupe de pacienţi, bolnavi şi sănătoşi sau bolnavi cu diferite diagnostice, pe baza unor atribute, în speţă, genele diferenţial exprimate. În cadrul învăţării supervizate urmărim să discriminăm între pacienţii aparţinând diferitelor grupuri pe baza unui număr restrâns de instanţe disponibile (set de antrenament) şi a utiliza această informaţie pentru a face preziceri pentru instanţe noi, în acest caz, viitori pacienţi. Aceste noi instanţe, care nu au fost utilizate la învăţarea din setul de antrenament şi pentru care ne dorim să prezicem apartenenţa la diferitele clase (diagnosticarea), reprezintă setul de testare. Această abordare utilizează informaţia, cunoscută a priori, a apartenenţei instanţelor

la diferitele clase. Învățarea supervizată are ca scop studierea felului în care atributele dintr-un set de date cu clase cunoscute descriu acele clase și utilizarea acestor informații pentru a prezice clasa din care fac parte obiecte noi, care nu au făcut parte din setul de date de antrenament. Încadrarea unei instanțe într-o clasă cunoscută este un cadru comun în care învățarea supervizată este o tehnică extrem de utilă.

Deosebirea esenţială între metodele de învăţare nesupervizată, studiate anterior, şi metodele de clasificare, constă în atitudinea faţă de clasele reale ale datelor din setul de antrenament. Dacă metodele de învăţare nesupervizate ignorau această informaţie, metodele de clasificare au ca finalitate descoperirea unui mod de a separa datele studiate, astfel încât instanţele aparţinând fiecărei clase studiate să fie clar evidenţiate şi separate de celelalte.

Într-un experiment microarray, se pot învăța elemente foarte importante din clasificarea probelor în funcție de diferite condiții studiate, diverse stadii în cadrul aceleiași patologii sau influența unor factori asupra unei boli de interes, patogeni sau tratamente. Aplicabilitatea diferiților algoritmi de învățare supervizată în analiza datelor microarray a fost evaluată [36, 37] pe date reale.

2.2.3.1. Clasificatorul Naïve Bayes

Cea mai simplă abordare în rezolvarea unei astfel de probleme este să ghicim clasa din care nou pacient face parte. Spre exemplu, în cazul în care avem două clase posibile, bolnav sau sănătos, o metodă de-a ghici clasa din care face parte pacientul este „datul cu banul”. Dacă moneda este perfect echilibrată, există şanse de 50% ca un nou pacient să fie clasificat drept sănătos şi 50% ca pacientul să fie etichetat drept bolnav. Am creat practic cel mai simplu clasificator pentru a adresa problema noastră. Clasificăm astfel noii pacienţi pe baza probabilităţii de a obţine una dintre variantele cap sau pajură, asociate în prealabil uneia dintre clasele posibile, bolnav sau sănătos.

Să presupunem că boala pe care urmărim să o diagnosticăm apare la un procent cunoscut, de exemplu, 1 din 6 indivizi în populaţia din care provin exemplele din studiul nostru. Prin această metodă, dacă moneda este perfect echilibrată, după un număr suficient de mare de clasificări ne aşteptăm să obţinem un număr egal de pacienţi clasificaţi drept bolnavi şi sănătoşi, ceea ce evident, nu corespunde realităţii. Aşadar, un mod foarte simplu de a evalua clasificatorul nostru este să comparăm rezultatele obţinute cu starea naturală, cunoscută a priori. În cazul de mai sus, un prim semn de performanţă satisfăcătoare a clasificatorului creat ar fi etichetarea a 1/6 dintre noile exemple drept bolnav şi a 5/6 drept sănătos.

Prin urmare, un mod foarte intuitiv de a îmbunătăţi metoda noastră de clasificare ar fi să folosim un zar în locul monedei. Am îmbunătăţit astfel metoda de clasificare ţinând cont de o informaţie cunoscută a priori, şi anume că starea naturală presupune 1/6 persoane bolnave iar 5/6 sănătoase. Un alt mod de a îmbunătăţi metoda datului cu banul este utilizarea unei reguli de decizie foarte simplă. Din moment ce cunoaştem că doar 1/6 dintre indivizi sunt bolnavi, putem decide pentru fiecare exemplu că este sănătos, aşadar 5/6 dintre indivizi vor fi clasificaţi corect drept sănătoşi. Astfel, avem deja la dispoziţie două metode diferite care clasifică pacienţii corespunzător stării naturale, 1/6 bolnavi şi 5/6 sănătoşi. Din abordarea anterioară rezidă cel puţin două aspecte foarte interesante. În primul rând, se evidenţiază aspectul că este de dorit să includem în construcţia

clasificatorului informaţiile cunoscute a priori. În al doilea rând, ar fi important să avem o măsură cantitativă a performanţei unui clasificator pentru a alege dintre toate metodele posibile de a discerne între exemple, clasificatorul cel mai performant pe problema noastră.

Considerăm că pentru boala studiată, nu există un test de laborator care discriminează perfect pacienţii sănătoşi de cei bolnavi. Dacă un astfel de test ar exista, nu am mai avea obiectul studiului de PR asupra acestei boli. Aşadar, să presupunem că avem la dispoziţie, în plus faţă de starea naturală, patru analize de laborator ale căror rezultate sunt importante pentru boala studiată. De exemplu, considerăm că avem patru analize de laborator ale căror rezultate sunt disponibile pentru un număr de 15 pacienţi concomitent cu diagnosticele fiecăruia. Rezultatele analizelor de laborator pentru cei 15 pacienţi, sunt prezentate alături de diagnosticul fiecăruia în tabelul 2.4. Aceste informaţii, asociate stării naturale cunoscute, deschid calea spre un clasificator mai bun, pentru un nou individ suspectat de boala studiată. Să presupunem că un nou individ se prezintă cu datele prezentate în tabelul 2.5. Astfel, problema noastră, diagnosticarea noului individ, poate fi formulată în termeni de probabilităţi.

Tabel 2.4 – Set de antrenament pentru clasificatorul naïve Bayes

ID Pacient	Test 1	Test 2	Test 3	Test 4	Diagnostic
1	normal	scăzută	normală	normal	sănătos
2	normal	scăzută	scăzută	normal	sănătos
3	normal	scăzută	crescuta	normal	bolnav
4	normal	crescuta	normală	normal	bolnav
5	normal	crescuta	scăzută	normal	bolnav
6	normal	crescuta	crescuta	pozitiv	bolnav
7	normal	crescuta	crescuta	normal	bolnav
8	normal	normală	normală	normal	bolnav
9	normal	normală	scăzută	normal	sănătos
10	normal	normală	crescuta	normal	bolnav
11	pozitiv	scăzută	normală	pozitiv	sănătos
12	pozitiv	scăzută	scăzută	normal	sănătos
13	pozitiv	crescuta	crescuta	pozitiv	sănătos
14	pozitiv	normală	scăzută	pozitiv	bolnav
15	pozitiv	normală	scăzută	normal	sănătos

Tabel 2.5 – Instanţă de testare pentru clasificatorul naïve Bayes

ID Pacient	Test 1	Test 2	Test 3	Test 4	Diagnostic
16	pozitiv	normală	normală	pozitiv	?

Preotul Thomas Bayes este autorul teoremei cu titlul omonim care oferă soluţia la problema combinării acestor informaţii cu scopul de a obţine o regulă de decizie. Teorema lui Bayes [38] este:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.8}$$

unde:

$P(A)$ - probabilitatea de apariţie a evenimentului A, poartă numele de *probabilitate a priori*, deoarece nu depinde de evenimentul B.

$P(B)$ - probabilitatea de apariţie a evenimentului B, poartă numele de *evidenţă*.

$P(A|B)$ - probabilitatea de apariţie a evenimentului A, condiţionată de evenimentul B, poartă numele de *probabilitate a posteriori*, deoarece depinde de evenimentul B.

$P(B|A)$ - probabilitatea de apariţie a evenimentului B, condiţionată de evenimentul A, poartă numele de *plauzibilitate*, deoarece depinde de evenimentul B.

Aşadar, putem rescrie Teorema lui Bayes mai plastic, în cuvinte, de forma:

$$\text{Probabilitate a posteriori} = \frac{\text{Probabilitate a priori} \times \text{Plauzibilitate}}{\text{Evidenta}} \tag{2.9}$$

În cazul nostru, probabilitatea a priori este dată de şansa ca un individ să sufere de boala studiată. Din datele pe care le avem la dispoziţie, probabilităţile a priori ca un individ să fie bolnav sau sănătos sunt P(bolnav) = 8/15 şi respectiv P(sănătos) = 7/15. Deoarece există doar două variante posibile, suma celor două probabilităţi a priori trebuie să fie 1. Cele patru teste pe care le avem la dispoziţie sunt de fapt evenimente considerate drept independente. Independenţa evenimentelor stă la baza teoremei lui Bayes. Prezumţia de independenţă este cea care oferă posibilitatea multiplicării probabilităţilor, însă este evident că, de cele mai multe ori evenimentele nu sunt de fapt independente, iar presupunerea poate fi considerată a fi naivă. Acesta este şi motivul asocierii lui „naive” la numele clasificatorului corespunzător.

Pentru a aplica teorema lui Bayes, este foarte util să sumarizăm datele pacienţilor din setul de antrenament (tabelul 2.6) şi să calculăm frecvenţele de apariţie a fiecărui eveniment (tabelul 2.7):

Tabel 2.6 – Revizuirea datelor despre pacienţi

Atribut	Valoare	Bolnav	Sănătos	Total
Test 1	normal	7	3	**10**
	pozitiv	1	4	**5**
Test 2	scăzută	1	4	**5**
	crescută	4	1	**5**
	normală	3	2	**5**
Test 3	normală	2	2	**4**
	scăzută	2	4	**6**
	crescută	4	1	**5**
Test 4	normal	6	5	**11**
	pozitiv	2	2	**4**
Total		**8**	**7**	

Tabel 2.7 – Frecvenţe observate

Atribut	Valoare	Bolnav	Sănătos
Test 1	normal	7/8	3/7
	pozitiv	1/8	4/7
Test 2	scăzută	1/8	4/7
	crescută	4/8	1/7
	normală	3/8	2/7
Test 3	normală	2/8	2/7
	scăzută	2/8	4/7
	crescută	4/8	1/7
Test 4	normal	6/8	5/7
	pozitiv	2/8	2/7

Pentru a trata un eveniment nou E=(Test 1 = pozitiv, Test 2 = normală, Test 3 = normală, Test 4 = pozitiv), în speţă a prezice diagnosticul noului pacient, formulăm ipoteza nulă: „Pacientul este sănătos”.

$$P(sanatos \mid E) = \frac{P(E \mid sanatos)P(sanatos)}{P(E)}$$

Automat, ipoteza alternativă devine „pacientul este bolnav”.

$$P(bolnav \mid E) = \frac{P(E \mid bolnav)P(bolnav)}{P(E)}$$

Cunoaştem probabilităţile a priori $P(sanatos)$ şi $P(bolnav)$. Datorită prezumţiei de independenţă, putem să calculăm $P(E \mid sanatos)$ şi $P(E \mid bolnav)$ după cum urmează:

$$P(E \mid sanatos) = P(pozitiv \mid sanatos) \cdot P(normala \mid sanatos) \cdot P(normala \mid sanatos) \cdot P(pozitiv \mid sanatos)$$

$$P(E \mid bolnav) = P(pozitiv \mid bolnav) \cdot P(normala \mid bolnav) \cdot P(normala \mid bolnav) \cdot P(pozitiv \mid bolnav)$$

Consecutiv expresiile teoremei lui Bayes pentru situaţia studiată devin:

$$P(sanatos \mid E) = \frac{P(pozitiv \mid sanatos) \cdot P(normala \mid sanatos) \cdot P(normala \mid sanatos) \cdot P(pozitiv \mid sanatos) \cdot P(sanatos)}{P(E)}$$

$$P(bolnav \mid E) = \frac{P(pozitiv \mid bolnav) \cdot P(normala \mid bolnav) \cdot P(normala \mid bolnav) \cdot P(pozitiv \mid bolnav) \cdot P(bolnav)}{P(E)}$$

Înlocuind frecvenţele corespunzătoare din tabelul 2.7 în formulele de mai sus, obţinem probabilităţile ca noul pacient să fie sănătos, şi respectiv bolnav.

$$P(sanatos \mid E) = \frac{4/7 \cdot 2/7 \cdot 2/7 \cdot 2/7 \cdot 7/15}{P(E)} = \frac{0.00622}{P(E)}$$
$$P(bolnav \mid E) = \frac{1/8 \cdot 3/8 \cdot 2/8 \cdot 2/8 \cdot 8/15}{P(E)} = \frac{0.0015625}{P(E)}$$

Cu toate că probabilitatea $P(E)$ de apariţie a evenimentului E=(Test 1 = pozitiv, Test 2 = normală, Test 3 = normală, Test 4 = pozitiv) nu este cunoscută, acest termen dispare la normalizare şi obţinem probabilităţile:

$$P(sanatos \mid E) = \frac{4/7 \cdot 2/7 \cdot 2/7 \cdot 2/7 \cdot 7/15}{P(E)} = \frac{0.00622}{0.00622 + 0015625} \cdot 100 = 79.92\%$$
$$P(bolnav \mid E) = \frac{1/8 \cdot 3/8 \cdot 2/8 \cdot 2/8 \cdot 8/15}{P(E)} = \frac{0.0015625}{0.00622 + 0015625} \cdot 100 = 20.08\%$$

În concluzie, probabilitatea ca individul analizat să fie sănătos este mai mare decât ca el să fie bolnav. Regula de decizie potrivită este alegerea clasificării cu probabilitatea mai mare. Analog, aceeaşi regulă poate fi reformulată ca alegerea erorii cu probabilitatea cea mai mică. În acest caz, decizia ar fi că pacientul este sănătos.

Principii izvorâte din teoria lui Bayes au fost adesea utilizate şi dezvoltate special pentru analiza datelor [39] microarray. Proiectul R pune la dispoziţia utilizatorului un clasificator naive Bayes în pachetul *e1071*. Pentru studiile ADN microarray există posibilitatea utilizării pachetului MLInterfaces. Toate implementările oferă posibilitatea utilizării estimatorului Laplace pentru evitarea situaţiilor cu probabilităţi egale cu 0.

Atributele au valori reale continue în datele de ADN microarray. Astfel, plauzibilitatea din teorema lui Bayes nu mai poare fi exprimată prin probabilităţi, ci prin densităţi de probabilitate. Considerând că valorile numerice ale atributelor provin dintr-un anumit tip de distribuţie. Formula 2.8 a teoremei lui Bayes va suferi modificări în sensul că atât evidenţa, cât şi plauzibilitatea vor fi exprimate prin densităţi de probabilităţi. Vom adopta convenţia notării probabilităţilor cu litera P majusculă, iar a densităţilor de probabilitate cu litera p minusculă. Astfel, formula teoremei lui Bayes devine:

$$P(A \mid B) = \frac{p(B \mid A)P(A)}{p(B)} \tag{2.10}$$

Pentru a identifica în practică modificările suferite de clasificatorul Naive Bayes pentru lucrul cu valori numerice şi a obţine o impresie comparativă cu cazul valorilor nominale, vom modifica uşor setul de date anterior, considerând că al patrulea test de laborator oferă rezultate în valori reale. Astfel, atât setul de antrenament din tabelul 2.4, cât şi instanţa de testare prezentată în tabelul 2.5 vor fi schimbate. Noul set de antrenament şi forma noii instanţe de testare sunt prezentate în tabelele 2.8, respectiv 2.9.

Tabel 2.8 – Set de antrenament pentru clasificatorul naïve Bayes

ID Pacient	Test 1	Test 2	Test 3	Test 4	Diagnostic
1	normal	scăzută	normală	150	sănătos
2	normal	scăzută	scăzută	160	sănătos
3	normal	scăzută	crescuta	170	bolnav
4	normal	crescuta	normală	190	bolnav
5	normal	crescuta	scăzută	180	bolnav
6	normal	crescuta	crescuta	850	bolnav
7	normal	crescuta	crescuta	210	bolnav
8	normal	normală	normală	180	bolnav
9	normal	normală	scăzută	220	sănătos
10	normal	normală	crescuta	240	bolnav
11	pozitiv	scăzută	normală	800	sănătos
12	pozitiv	scăzută	scăzută	165	sănătos
13	pozitiv	crescuta	crescuta	900	sănătos
14	pozitiv	normală	scăzută	1200	bolnav
15	pozitiv	normală	scăzută	175	sănătos

Tabel 2.9 – Instanţă de testare pentru clasificatorul naïve Bayes

ID Pacient	Test 1	Test 2	Test 3	Test 4	Diagnostic
16	pozitiv	normală	normală	2000	?

Presupunem că valorile numerice ale atributului Test 4 provin din distribuţii normale cu mediile $\mu_{sanatos}$, μ_{bolnav} iar deviaţiile standard $\sigma_{sanatos}$şi σ_{bolnav}. Decizia distribuţiei alese aparţine cercetătorului. Distribuţia Gaussiană este foarte frecvent folosită însă, în funcţie de configuraţia datelor, orice altă distribuţie poate fi utilizată. Funcţia densităţii de probabilitate a distribuţiei normale cu media μ şi deviaţia standard σ este dată de formula:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.11}$$

Astfel, densităţile de probabilitate corespunzătoare se calculează utilizând formula 2.11. Pentru exemplul nostru, datele vor fi sumarizate uşor diferit faţă de cazul discutat anterior, cu valori nominale (tabelul 2.6). Datele sumarizate pentru cazul cu valori reale sunt prezentate în tabelul 2.10.

Tabel 2.10 – Revizuirea datelor despre pacienţi

Atribut	Valoare	Bolnav	Sănătos	Total
Test 1	normal	7	3	**10**
	pozitiv	1	4	**5**
Test 2	scăzută	1	4	**5**
	crescută	4	1	**5**
	normală	3	2	**5**
Test 3	normală	2	2	**4**
	scăzută	2	4	**6**
	crescută	4	1	**5**
Test 4		Media = 402.5 Deviaţia standard = 396.0429	Media = 367.1429 Deviaţia standard= 331.8616	**15**
Total		**8**	**7**	

Pentru a putea aplica teorema lui Bayes, trebuie doar să înlocuim valorile x=2000, media $\mu_{sanatos} = 367.1429$ şi deviaţia standard $\sigma_{sanatos} = 331{,}8618$, media $\mu_{bolnav} = 402{,}5$ şi deviaţia standard $\sigma_{bolnav} = 396{,}0429$.

Aşadar, şi în cazul setului de date cu valori numerice, noua instanţă reprezintă un pacient sănătos cu probabilitatea de 74,48%.

Deşi clasificatorul Bayes prezintă o abordare foarte simplă a problemei clasificării, metoda oferă rezultate remarcabile în foarte multe situaţii, atât pe date cu valori numerice cât şi pe date cu valori nominale şi chiar în situaţiile când datele pentru antrenament sunt restrânse. Clasificatorul Bayes reprezintă o abordare probabilistică [40] a problemei învăţării supervizate, care, deşi este satisfăcătoare în multe situaţii, prezintă două dezavantaje majore. În primul rând, presupunerea că atributele sunt independente este naivă de cele mai multe ori. În al doilea rând faptul că pentru atributele cu valori numerice presupunem că provin dintr-o anumită distribuţie de probabilitate, care este cu siguranţă doar o aproximaţie a situaţiei reale, denaturează rezultatele.

Testăm în cele ce urmează comportamentul clasificatorului Naive Bayes pe setul de date Golub, format din 72 de instanţe şi 7129 de atribute. Utilizăm implementarea clasificatorului Naive Bayes oferită în pachetul MLInterfaces şi evaluăm performanţa sa utilizând 5-fold Cross Validation (Fig. 2.17).

Naive Bayes a reuşit să clasifice instanţele setului de date Golub cu o acurateţe de aproximativ 97,2%, ceea ce este încurajator. Experimentul de mai sus confirmă validitatea metodei şi oferă o imagine sugestivă a puterii metodelor probabilistice de clasificare. Acest rezultat demonstrează că în pofida neajunsurilor discutate mai sus şi a simplităţii metodei, Naive Bayes poate da rezultate satisfăcătoare în anumite probleme. În capitolele următoare vom avea posibilitatea de-a compara performanţa acestei metode pe setul de date Gloub, cu alţi clasificatori, cu complexitate crescută, care adresează problemele cu care se confruntă Naive Bayes.

```
>   naiveBayesCross5   =   MLearn(ALL.AML~.,   data=Golub_Merge,   naiveBayesI,+
+xvalSpec("LOG", 5, balKfold.xvspec(5)))
> confuMat(naiveBayesCross5)
     predicted
given ALL AML
  ALL  47   0
  AML   2  23
> accuracy<-(confuMat(naiveBayesCross5)[1]  +  confuMat(naiveBayesCross5)[4])  /+
+(confuMat(naiveBayesCross5)[1]      +      confuMat(naiveBayesCross5)[2]      +
+confuMat(naiveBayesCross5)[3] + confuMat(naiveBayesCross5)[4])
> accuracy
[1] 0.9722222
```

Fig. 2.17 – Clasificare cu Naive Bayes și validare încrucișată 5-fold.

2.2.3.2. Clasificatorul k-Nearest Neighbor (kNN)

Foarte multe dintre modelele inteligenţei artificiale încearcă să simuleze un proces specific inteligenţei umane. Inteligenţa umană recunoaşte, percepe, formele pe baza experienţei proprii individului. Oamenii recunosc obiecte pe care le ştiu în prealabil, în momentul când se reîntâlnesc cu ele [41]. Astfel, un copil care vede pentru prima oară un creion, nu va şti ce reprezintă sau ce utilitate are acest obiect. După ce va fi învăţat ce este un creion şi la ce foloseşte, în momentul când se va întâlni pentru prima oară cu un pix, el va considera că noul obiect este un creion. Practic, copilul a executat un proces de clasificare, al cărui rezultat este etichetarea noului obiect, pixul, drept cel mai apropiat exemplu pe care îl cunoştea deja, creionul. După ce părintele i-a explicat că noul obiect este în realitate un pix şi care sunt diferenţele dintre un creion şi un pix, copilul va asimila noile cunoştinţe şi, cel mai probabil, în momentul când se va întâlni ulterior cu un alt pix, va şti să-l clasifice corect. Este foarte probabil ca, în momentul când acelaşi copil va vedea pentru prima oară un stilou, să-l eticheteze drept pix, datorită similitudinilor dintre aspectul pixurilor şi stilourilor, în general, mai accentuate decât cu creioanele. Dacă pe măsură ce înaintează în vârsta, individul respectiv va începe să aprecieze estetic stilourile, va fi capabil să diferenţieze chiar între diferitele mărci care manufacturează stilouri şi variatele linii de modele. Practic, la baza acestui proces stau experienţa anterioară şi capacitatea de-a asimila, învăţa, din fiecare nou exemplu sau instanţă. Metodele *instance based learning* au ca finalitate învăţarea printr-un proces asemănător.

Pentru a obţine o imagine comparativă a modalităţilor de abordare, vom relua exemplul datelor de laborator pentru diagnosticul bolii din capitolul anterior. Vom considera că dispunem de doar două teste de laborator cu date numerice, astfel încât să beneficiem de avantajul reprezentării grafice în două dimensiuni. Să presupunem aşadar că avem la dispoziţie datele din tabelul 2.11 şi ne propunem să clasificăm pacientul cu datele din tabelul 2.12.

Reprezentarea grafică a datelor oferă o imagine mai intuitivă a problemei. În figura de mai jos (Fig. 2.18 a.) sunt prezentate datele de laborator ale celor 15 pacienţi cu diagnosticul cunoscut. Pacienţii suferinzi de boala studiată sunt reprezentaţi cu culoarea neagră, iar indivizii sănătoşi sunt coloraţi cu verde. Individul nou, al cărui diagnostic ne propunem să-l prezicem este figurat cu roşu.

Tabel 2.11 – Set de antrenament pentru clasificatorul kNN

ID Pacient	Test 1	Test 2	Diagnostic
1	235	150	sănătos
2	340	360	sănătos
3	245	170	bolnav
4	423	490	bolnav
5	229	580	bolnav
6	565	850	bolnav
7	235	210	bolnav
8	623	180	bolnav
9	220	220	sănătos
10	419	240	bolnav
11	325	800	sănătos
12	334	665	sănătos
13	380	900	sănătos
14	480	1200	bolnav
15	640	175	sănătos

Tabel 2.12 – Instanţă de testare pentru clasificatorul kNN

ID Pacient	Test 1	Test 2	Diagnostic
16	460	1000	?

O metodă de a eticheta noul caz este să-i atribuim individului clasa cu rezultatele cele mai asemănătoare, similare, la testele de laborator, deoarece acestea sunt singurele informaţii de care dispunem. Cazul cel mai similar este reprezentat de cel cu distanţa euclidiană cea mai mică faţă de noua instanţă. Intuitiv, putem determina grafic care este acest exemplu. Trasăm un cerc cu originea în noua instanţă şi augmentăm raza acestui cerc până când intersectează unul din exemplele din setul de date (Fig. 2.18 b.). Exemplul care intersectează primul cercul nostru (Fig. 2.19 a.) va fi cel mai similar cazului studiat, aşadar, atribuim noului individ eticheta instanţei cu care este cel similar. Din figura de mai sus, este evident că, prin această metodă, noul individ ar fi clasificat drept sănătos. O altă abordare posibilă este să ţinem cont de mai mult decât cel mai apropiat vecin, spre exemplu, să evaluăm cei mai apropiaţi 2 vecini (Fig. 2.19). În acest caz, decizia poate fi luată prin votul majoritar al vecinilor evaluaţi. Metoda noastră ar duce la un rezultat indecis, deoarece dintre cele două exemple cele mai similare unul este pacient suferind de boala studiată, iar al doilea reprezintă un individ sănătos. Este aşadar de dorit să evaluăm un număr impar de vecini pentru a evita această situaţie. Dacă evaluăm k=3 sau k=5 vecini (Fig. 2.19 c., d.) rezultatele sunt contradictorii. Pentru 3 vecini evaluaţi obţinem rezultatul că pacientul este bolnav. Dacă însă evaluăm 5 vecini, rezultatul va fi identic cu ceea ce am obţinut prin evaluarea unui singur vecin, prezicerea diagnosticului noului pacient este, sănătos.

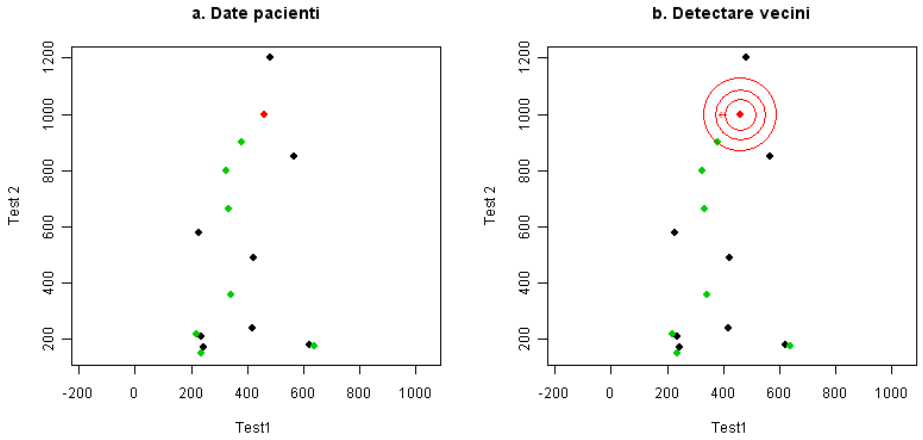


Fig. 2.18 – Ilustrarea principiului clasificării kNN.

Sorgintea numelui clasificatorului kNN este tocmai în această abordare, iar k reprezintă posibilitatea de-a alege numărul celor mai apropiaţi vecini evaluaţi. Exemplul de mai sus relevă câteva caracteristici esenţiale ale clasificatorului kNN. O caracteristică foarte importantă este că nu am presupus absolut nimic despre distribuţia din care provin datele noastre. Clasificatorul kNN face parte din metodele non-parametrice de învăţare supervizată, iar acest aspect reprezintă un avantaj în report cu clasificatorul naive Bayes. În al doilea rând, nu există un proces de antrenare bine definit. Faza de antrenament poate fi considerată doar memorarea exemplelor din setul de date, împreună cu etichetele lor. Algoritmul kNN poate fi sintetizat astfel:

1. calculează distanţele dintre instanţa de testare şi instanţele din setul de antrenament
2. ordonează distanţele în ordine crescătoare şi selectează cei k vecini evaluaţi
3. etichetează instanţa de testare potrivit votului majoritar al celor k vecini selectaţi

Distanţele calculate de kNN pot fi oricare dintre măsurile di-similarităţii, în funcţie de structura datelor. Se observă că, în faza de antrenament, algoritmul doar memorează datele şi etichetele lor. Doar în momentul testării unei noi instanţe se efectuează calculele, distanţele noii instanţe faţă de toate celelalte instanţe din setul de antrenament, asemenea copilului care învaţă tipurile de instrumente pentru scris. Acest tip „pasiv” de învăţare bazată pe instanţe, poarte numele de *lazy learning*. Alegerea distanței este foarte însemnată pentru performanța discriminărilor în general, dar anumiți clasificatori cum este kNN sunt foarte sensibili la alegerea modului de calculare a similarității.

Proiectul R dispune de variate implementări ale kNN în diferite pachete. Implementările din pachetele *class* şi *MLInterfaces* sunt cele mai uzitate pentru versiunea clasică a kNN în analiza genetică. Spre exemplu, pentru rezolvarea exemplului din tabelul 5.2, poate fi utilizat codul R din fig. 2.20.-2.23.

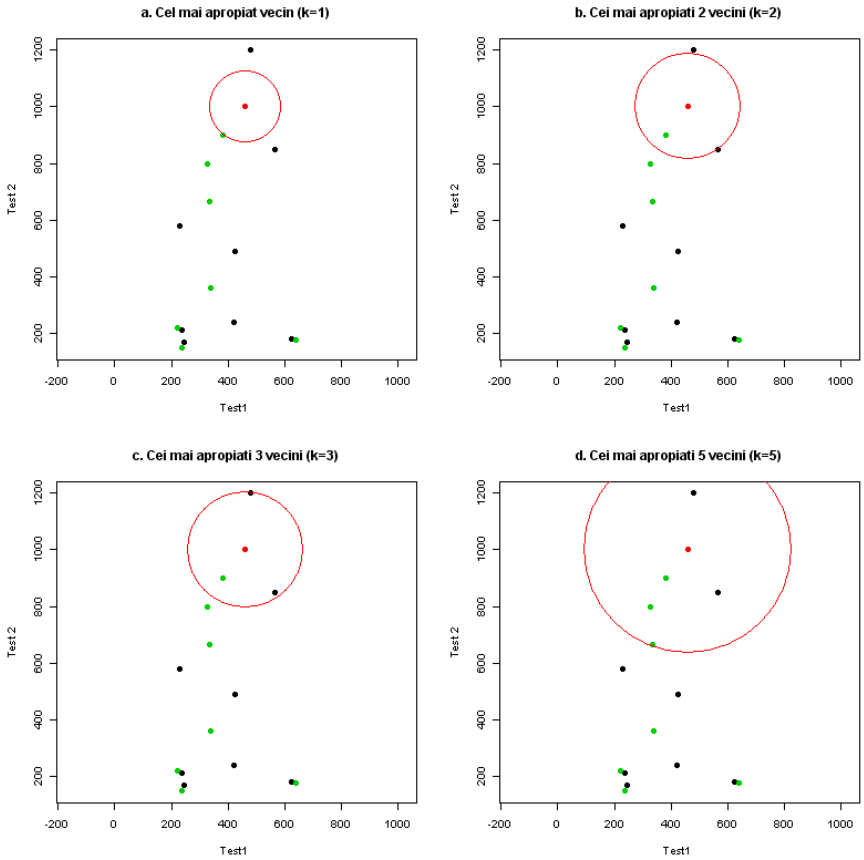


Fig. 2.19 – kNN pentru situaţiile k=1, k=2, k=3, k=5.

```
> library(MLInterfaces)
> rm(list=ls())
> DataPacienti<-read.table(file="C:/Data/knn.txt",header=TRUE)
> DataPacienti<-DataPacienti[,2:4]

########## kNN k=1
> knn1 = MLearn(Diagnostic ~ ., DataPacienti, knnI(k=1), 1:15)
> knn1
MLInterfaces classification output container
The call was:
MLearn(formula = Diagnostic ~ ., data = DataPacienti, .method = knnI(k
= 1),
    trainInd = 1:15)
Predicted outcome distribution for test set:
sanatos
      1
Summary of scores on test set (use testScores() method for details):
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      1       1       1       1       1       1
```

Fig. 2.20 – Clasificare cu kNN pentru k=1.

```
> ############# kNN k=2
> knn2 = MLearn(Diagnostic ~ ., DataPacienti, knnI(k=2), 1:15)
> knn2
MLInterfaces classification output container
The call was:
MLearn(formula = Diagnostic ~ ., data = DataPacienti, .method = knnI(k
= 2),
    trainInd = 1:15)
Predicted outcome distribution for test set:
bolnav
     1
Summary of scores on test set (use testScores() method for details):
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.5     0.5     0.5     0.5     0.5     0.5
```

```
> ############# kNN k=2
> knn2 = MLearn(Diagnostic ~ ., DataPacienti, knnI(k=2), 1:15)
> knn2
MLInterfaces classification output container
The call was:
MLearn(formula = Diagnostic ~ ., data = DataPacienti, .method = knnI(k
= 2),
    trainInd = 1:15)
Predicted outcome distribution for test set:
sanatos
      1
Summary of scores on test set (use testScores() method for details):
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.5     0.5     0.5     0.5     0.5     0.5
```

Fig. 2.21 – Clasificare cu kNN pentru k=2.

```
> ############# kNN k=3
> knn3 = MLearn(Diagnostic ~ ., DataPacienti, knnI(k=3), 1:15)
> knn3
MLInterfaces classification output container
The call was:
MLearn(formula = Diagnostic ~ ., data = DataPacienti, .method = knnI(k
= 3),
    trainInd = 1:15)
Predicted outcome distribution for test set:
bolnav
     1
Summary of scores on test set (use testScores() method for details):
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.6667  0.6667  0.6667  0.6667  0.6667  0.6667
```

Fig. 2.22 – Clasificare cu kNN pentru k=3.

```
> ############## kNN k=5
> knn5 = MLearn(Diagnostic ~ ., DataPacienti, knnI(k=5), 1:15)
> knn5
MLInterfaces classification output container
The call was:
MLearn(formula = Diagnostic ~ ., data = DataPacienti, .method = knnI(k
= 5),
    trainInd = 1:15)
Predicted outcome distribution for test set:
sanatos
      1
Summary of scores on test set (use testScores() method for details):
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.6     0.6     0.6     0.6     0.6     0.6
```

Fig. 2.23 – Clasificare cu kNN pentru k=5.

Am rulat de două ori algoritmul pentru situaţia k=2 deoarece, aşa cum ne aşteptam din distribuţia voturilor, rezultatul a fost indecis, iar alegerea s-a făcut fortuit. În două rulări succesive, algoritmul a decis diferit.

Este aşadar evident că, modul în care este ales k, poate influenţa major rezultatul clasificării. Acest aspect se datorează sensibilităţii kNN faţă de configuraţia locală a datelor. Datorită modului de etichetare, votul majorităţii, dacă nu există un echilibru în setul de antrenament între exemplele cu diferite etichete, exemplele mai frecvente tind să domine în rezultatul votului. De asemenea, distanţa aleasă ca măsură a di-similarităţii este deopotrivă foarte importantă. Nu există un standard pentru alegerea acestor parametrii, ei depind foarte mult de configuraţia datelor studiate. Aşadar, se încearcă uzual, stabilirea parametrilor optimi pentru clasificarea cu kNN pe problemă dată. Metoda cea mai folosită este căutarea acestor parametrii cu *validare încrucișată* pe setul de antrenament.

Testăm în continuare utilitatea abordării kNN pentru clasificarea exemplelor din setul de date Golub cu 72 de instanţe și 7129 de atribute, cu k=8. Rezultatele sunt ilustrate în fig. 2.24.

```
> #k=8
> knnGolub = MLearn(ALL.AML ~ ., data=Golub_Merge, knnI(k=8, +      +
l=5),xvalSpec("LOG", 5, balKfold.xvspec(5)))
> confuMat(knnGolub)
     predicted
given ALL AML
  ALL  47   0
  AML   7  14
> accuracy
[1] 0.8970588
```

Fig. 2.24 – Clasificarea datelor Golub cu kNN pentru k=8.

Deşi rezultatele kNN pe setul nostru de date nu sunt satisfăcătoare, în general, algoritmul funcţionează foarte bine pe multe probleme. În conformitate cu teorema No Free Lunch, nici un algoritm nu va avea rezultate optime pe orice problemă de învăţare. Comparativ cu naive Bayes, pe setul de date Golub, kNN discriminează cu o acurateţe inferioară. Totuşi acest aspect se poate interpreta doar în sensul că pe această problemă, învăţarea setului de date Golub cu 72 de instanţe şi 7129 de atribute, naive Bayes are performanţă superioară, dar nicidecum nu se poate extrapola această concluzie în general.

Aplicaţiile lui kNN depăşesc aria clasificării. kNN a fost folosit cu succes în probleme de interpolare, chiar regresie. kNN a fost de asemenea, utilizat cu succes [42] în analiza datelor microarray.

2.2.3.3. Clasificatorul liniar

În foarte multe situaţii de zi cu zi, oamenii sunt puşi în ipostaza de-a lua decizii pe baza unei clasificări. Uneori clasificarea impune o regiune de decizie foarte clară, intuitivă, alteori nu. Spre exemplu, în situaţia copilului care învaţă instrumentele de scris, chiar dacă regiuni de decizie există, ele sunt complexe, greu de descris şi, cu siguranţă nu sunt conştientizabile de către subiect. Totuşi, există situaţii când subiectul îşi propune regiuni de decizie foarte clare pentru o anumită clasificare. Un exemplu poate fi linia de demarcaţie din mijlocul şoselei. Şoferul urmăreşte această linie pentru a se asigura că se află în regiunea destinată lui, sigură, a drumului. Depăşirea acestei linii de demarcaţie poate însemna un eveniment rutier, aşadar, de fiecare dată când un şofer priveşte această linie, el îşi clasifică poziţia curentă drept sigură sau nesigură. Un individ care îşi doreşte să mănânce un măr parţial alterat poate îmbrăţişa o atitudine asemănătoare. După ce ar decela partea degradată a mărului, ar putea decide să tăie cu un cuţit, un plan prin măr, astfel încât să păstreze doar zona comestibilă a fructului. Teoretic, el ar aprecia zonele vizibile ale mărului în comestibilă şi alterată în funcţie de anumite atribute (culoare, senzaţie la palpare). Putem considera inspecţia mărului drept un antrenament pentru luarea deciziei. Dacă, pentru simplitate, el ar decide să separe cele 2 părţi cu o singură tăietură de cuţit, ar imagina prin măr, o regiune de decizie plană. Astfel acest plan ar reprezenta în esenţă, o combinaţie liniară a atributelor după care copilul a luat decizia (culoare, senzaţie la palpare), iar inspecţia mărului ar fi utilă doar pentru a stabili poziţia planului de tăiere. Este posibil ca la inspecţia după tăiere a zonei comestibile a mărului, în interior, să existe o parte degradată sau nu. Dacă partea alterată a dispărut se poate considera că decizia cu privire la poziţia planului a fost corectă. Există însă şi posibilitatea ca interiorul mărului să fie în mare parte alterat chiar şi în zona care părea comestibilă, şi în acest caz, alegerea suprafeţei plane pentru separarea celor două fragmente, deşi simplă, nu a fost o decizie fericită. Acest exemplu este desigur exagerat, însă introduce ideea din spatele multor metode de clasificare extrem de eficiente.

Clasificatorul liniar are ca finalitate discriminarea după o regiune de decizie prestabilită, funcţie de combinaţia liniară a atributelor. Funcţia $d(x)$ este discriminant liniar pentru două clase de obiecte dacă:

$$d(x) = w^t x + w_0 \tag{2.12}$$

unde:
x - vectorul atributelor instanţei,
w – vector al ponderilor,
w_0 - valoare de tăiere.

Astfel, dacă $d(x) > 0$, instanţa curentă este etichetată aparţinând unei clase, iar dacă $d(x) < 0$, instanţa este etichetată aparţinând celei de-a doua clase. Situaţia $d(x) = 0$ reprezintă o situaţie nedeterminată.

Forma suprafeţei de decizie este dată de ecuaţia $d(x) = 0$, şi reprezintă un hiperplan care divide spaţiul atributelor în 2 semispaţii. Instanţele celor două clase sunt dispuse de o parte şi de alta a hiperplanului astfel definit. Valoarea funcţiei $d(x)$ oferă distanţa dintre x şi hiperplan.

Pentru a adresa cazul mai multor clase de obiecte nu este suficient un singur hiperplan de separare. Aşadar, nu este suficient un singur clasificator liniar. Există mai multe variante pentru combinarea mai multor clasificatori liniari astfel încât să poată rezolva probleme multi-clase. Varianta cea mai comună poartă numele de maşină liniară şi constă în definirea unui număr de clasificatori liniari egal cu numărul claselor. Decizia de clasificare va fi luată pe valoarea maximă $d_i(x)$, unde i ia valori între 1 şi numărul de clase [26].

Finalitatea în rezolvarea unui clasificator liniar o reprezintă determinarea vectorului ponderilor. În procesul de antrenament, vectorii instanţelor sunt cunoscuţi, aşadar parametrul este vectorul ponderilor. În consecinţă, pot exista mai multe soluţii, respectiv mai multe hiperplane de separe la pentru o problemă dată. Clasificatorul liniar va găsi unul dintre aceste hiperplane, dar nu obligatoriu, hiperplanul optim de separare. O altă consecinţă a acestei parametrizări este că putem generaliza clasificatorul liniar introducând termenii produs între componentele lui x. Aceste expresii vor continua să fie liniare în raport cu parametrii, iar suprafeţele de separare pot câştiga astfel în complexitate.

Există mai multe metode de a găsi o soluţie, a calcula ponderile. Tehnicile au la bază ideea definirii unui criteriu $J(s)$, astfel încât pentru o soluţie a setului de ecuaţii $s^t x + s_0 > 0$, criteriul $J(s)$ este minim. Un criteriu poate fi structurat pe numărul instanţelor clasificate greşit de către clasificatorul liniar. Spre exemplu:

$$J(s) = \sum_{x \in E} -s^t x \tag{2.13}$$

unde *E* reprezintă mulţimea instanţelor clasificate greşit cu ponderile soluţiei *s*, poartă numele de criteriu perceptron şi este, poate cea mai simplă alegere în acest sens. Există numeroase alte variante de alegere a criteriului, cu avantaje şi dezavantaje specifice.

O modalitate de-a aborda problema minimizării criteriului $J(s)$ are la bază metoda pantei abrupte (*gradient descent)*. Metoda presupune căutarea din aproape în aproape, în spaţiul soluţiilor, a minimului local, prin coborârea pe panta cea mai abruptă. Algoritmul se iniţializează cu valori arbitrare pentru vectorul ponderilor şi actualizează valorile lui *s*, pas cu pas, după formula:

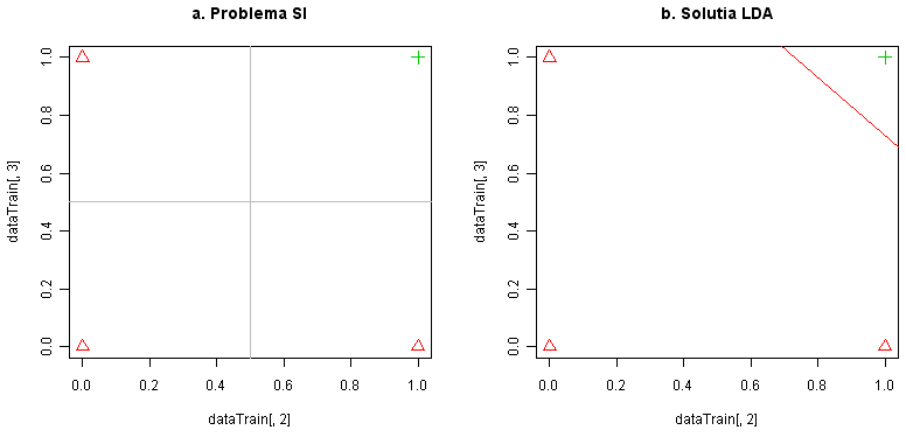
$$s_{n+1} = s_n - \lambda_n \nabla J(s_n) \tag{2.14}$$

unde λ_n este o valoare pozitivă care cuantifică magnitudinea pasului la iteraţia n. Algoritmul continuă până când $\lambda_n \nabla J(s_n)$ atinge o valoare ţintă prestabilită. Practic algoritmul converge când valorile ponderilor găsite, asigură o rată de eroare la clasificare, satisfăcătoare. Soluţia găsită (*s*) reprezintă vectorul ponderilor care oferă orientarea hiperplanului de separare între cele două clase. Este astfel evident că, în urma acestei abordări pe o problemă dată, pot rezulta o multitudine de soluţii satisfăcătoare, respectiv hiperplane care separă acceptabil clasele studiate.

Soluţia clasificatorului liniar trebuie să excludă posibilitatea situaţiilor indecise şi să ofere generalitate rezultatului. Acest aspect este rezolvat prin introducerea unei valori *m*, astfel încât $w^t x + w_0 \geq m > 0$. Această valoare impune o *margine* clasificatorului liniar.

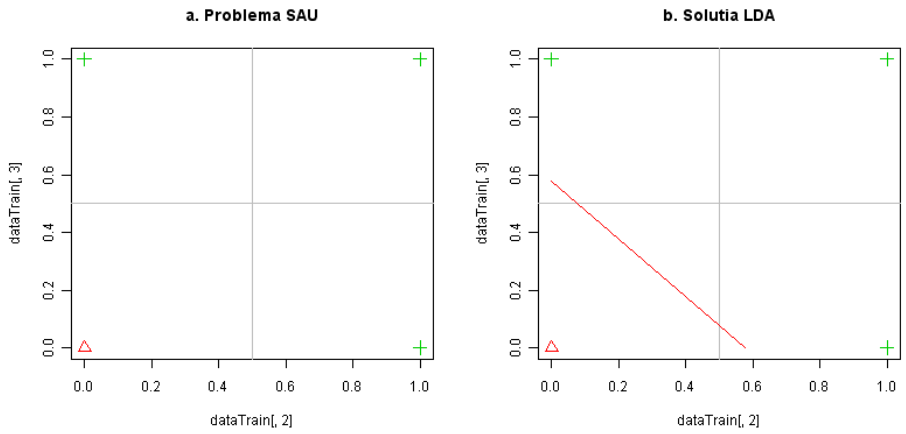
Problema majoră a clasificatorului liniar constă lipsa de flexibilitate datorită suprafeţei de separare. Rareori, în problemele de clasificare, obiectele aparţinând claselor diferite sunt separabile printr-un hiperplan. În aceste puţine situaţii, clasificatorul liniar este o alegere perfectă. Pentru clasele care nu pot fi însă satisfăcător separabile printr-un hiperplan, clasificatorul nu oferă rezultatele dorite. Vom studia în continuare soluţiile oferite de clasificatorul liniar la problemele ŞI (Fig. 2.25 a)), SAU (Fig. 2.25 b) şi XOR (Fig. 2.25 c)).

Clasificatorii liniari nu au nici o problemă în separarea instanţelor din problemele ŞI ori SAU. Multiple variante de calcul al ponderilor duc la variante de separare. Când vine însă vorba de problema XOR, clasificatorul liniar returnează o eroare. Obiecte care pot fi separate cu un hiperplan se numesc liniar separabile. Cu alte cuvinte, dacă pentru un set de obiecte aparţinând la două clase diferite, se poate găsi un vector soluţie s, care separă satisfăcător reprezentanţii celor două grupe, obiectele se numesc *separabile liniar*. În primele două situaţii (ŞI, SAU), obiectele sunt separabile liniar, aşadar, clasificatorul a reuşit să ofere o soluţie problemei separării instanţelor. În cea de-a treia situaţie, XOR, clasificatorul liniar întâmpină dificultăţi deoarece obiectele nu au proprietatea de-a fi liniar separabile.

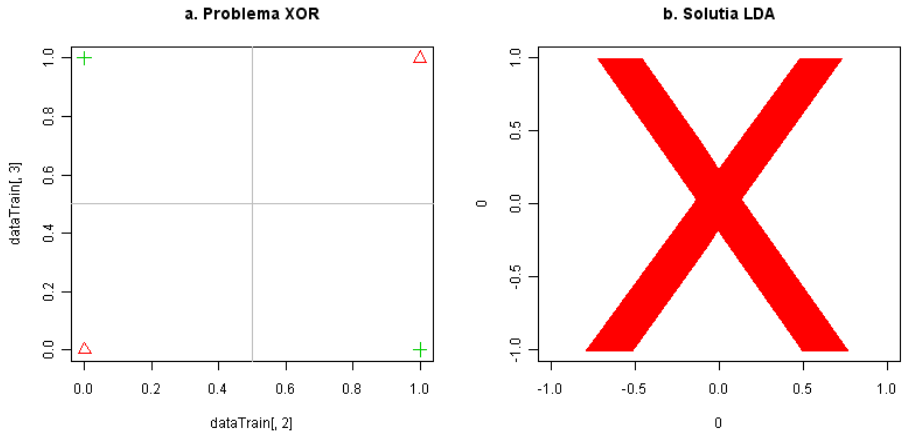


a)

Fig. 2.25 – Soluţia LDA la problema a) ŞI, b) SAU, c)XOR.



b)



c)

Fig. 2.25 continuare – Soluţia LDA la problema a) ŞI, b) SAU, c)XOR.

Modelele liniare au fost adesea utilizate pentru analizarea genelor diferențial exprimate. Succesul limitat se datorează și faptului că procese biologice foarte complexe, modelate astfel, sunt departe de a fi satisfăcător descrise cu modele liniare. Așadar, pentru studiile de analiză genetică, aplicaţiile clasificatorului liniar sunt limitate. Totuşi, aplicaţiile altor clasificatori, înrudiţi cu clasificatorul liniar, pot fi foarte eficiente. Pachetul MLInterfaces dispune de o implementare a clasificatorului liniar uşor aplicabilă în analiza ADN microarray. Vom ilustra în cele ce urmează (Fig. 2.26) performanţa şi aplicabilitatea pe setul de date Golub.

Cu toate că am obţinut o acurateţe ridicată, clasificatorul suferă datorită numărului ridicat de variabile şi a coliniarităţii lor. O utilitate sporită a clasificatorului liniar poate fi obţinută în combinaţie cu un algoritm evoluţionist, pentru selecţia atributelor.

2.2.3.4. Maşini de suport vectorial (SVM)

Dezavantajele majore ale clasificatorului liniar au fost potenţialitatea unor soluţii variate la aceeaşi problemă şi limitarea aplicabilităţii la obiectele liniar separabile. SVM reprezintă o metodă de învăţare supervizată avansată, care adresează aceste probleme. SVM au fost introduşi în 1963 de matematicianul rus Vladimir Vapnik [43].

Asemenea clasificatorului liniar, finalitatea SVM este clasificarea datelor prin găsirea unui hiperplan de separare între instanţele celor două grupe. Spre deosebire de clasificatorul liniar care returnează unul dintre variantele posibile de hiperplan capabil să separare obiectele satisfăcător, SVM caută hiperplanul optim de separare între instanţe. Acest hiperplan poartă numele de hiperplan de margine maximă (maximum-margin hyperplane). Acest rezultat reprezintă teoretic soluţia care oferă cel mai înalt grad de generalitate.

Abordarea SVM impune maximizarea distanţei dintre două hiperplane paralele, cu proprietatea de-a separa perfect obiectele aparţinând a două clase (-1, respectiv +1). Aşadar, urmărim determinarea a două hiperplane cu ecuaţiile:

$$w^t x_i + w_0 \geq +1, \text{ dacă instanţa i aparţine clasei +1}$$
$$w^t x_i + w_0 \leq -1, \text{ dacă instanţa i aparţine clasei -1} \quad (2.15)$$

Maximizarea distanţei dintre cele două hiperplane înseamnă, de fapt, minimizarea normei euclidiene a vectorului greutăţilor w, iar această problemă poate fi transcrisă în termeni de optimizare cvadratică şi rezolvată în consecinţă.

Câteva instanţe din setul de date vor satisface ecuaţiile:

$$w^t x_{SVj} + w_0 = +1, \text{ unde instanţa j aparţine clasei +1}$$
$$w^t x_{SVj} + w_0 = -1, \text{ unde instanţa j aparţine clasei -1} \quad (2.16)$$

Aceste instanţe poartă numele de vectori suport (support vectors). Spre exemplificare, Fig. 2.27 ilustrează caracteristicile SVM pentru problema SAU. Pentru rezolvarea problemei, SVM va considera instanţele (0,0), (0,1) şi (1,0) drept vectori suport. Hiperplanul de margine maximă, în cazul acesta o dreaptă, va fi situat echidistant între cele două drepte paralele corespunzătoare vectorilor suport.

```
> LDA1 = MLearn(ALL.AML ~ ., data=Golub_Merge, ldaI, + +xvalSpec("LOG",
5, balKfold.xvspec(5)))
> confuMat(LDA1)
     predicted
given ALL AML
  ALL  46   1
  AML   6  19
> accuracy
[1] 0.9027778
```

Fig. 2.26 – Clasificarea datelor Golub cu LDA.

Versiunea de SVM discutată anterior reprezintă doar o extensie a clasificatorului liniar care beneficiază de o generalitate îmbunătăţită şi nu poate clasifica decât obiecte separabile liniar. O versiune ulterioară a SVM [44], apărută în 1992, propune o abordare diferită, ce îi conferă posibilitatea de a clasifica date neseparabile liniar. Ideea care stă la baza acestei variante este că, există întotdeauna un spaţiu cu un număr de dimensiuni, în general mai mare decât spaţiul original al atributelor, în care instanţele aparţinând celor două clase devin separabile liniar. Plecând de la această idee, prin transformarea neliniară a instanţelor din spaţiul original al atributelor într-un astfel de spaţiu, poate fi folosit un SVM pentru a separa instanţele în noua configuraţie. În esenţă această transformare se rezumă la aplicarea unei funcţii neliniare datelor din spaţiul original şi separarea lor prin hiperplanul optim în spaţiul multidimensional rezultat. Funcţia neliniară care transformă datele originale poartă numele de funcţie esenţială (kernel function).

Hiperplanul optim astfel determinat, reprezintă o suprafaţă neliniară în spaţiul original al atributelor astfel încât SVM dobândeşte capacitatea de-a clasifica date neseparabile liniar.

Aşadar, clasificarea instanţelor neseparabile liniar cu SVM presupune doi paşi foarte importanţi:

1. transformarea datelor originale cu ajutorul unei funcţii esenţiale:

$$x_i \rightarrow \varphi(x_i) \tag{2.17}$$

2. determinarea hiperplanului optim în spaţiul multidimensional rezultat. Acest hiperplan va satisface ecuaţia:

$$w^t \varphi(x_i) = 0 \tag{2.18}$$

Nu există garanţia că, prin alegerea unei anumite funcţii kernel, instanţele din spaţiul original al atributelor devin liniar separabile. Totuşi, există o mare varietate de posibile funcţii kernel şi posibilitatea ca, aplicând funcţia potrivită, să obţinem situaţia de obiecte liniar separabile. Implementarea SVM din pachetul e1071 beneficiază de patru opţiuni de funcţii kernel: liniară, polinomială, sigmoidă şi radială.

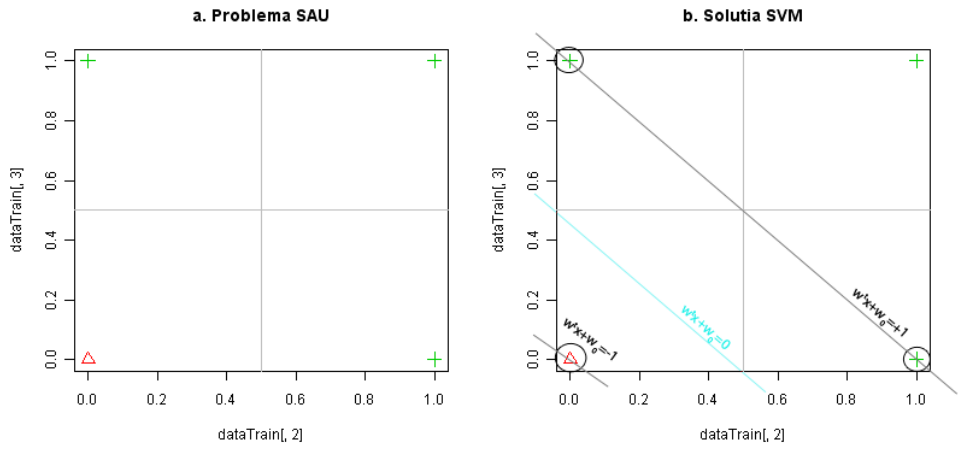


Fig. 2.27 – Soluţia SVM la problema SAU.

Revenind la exemplul XOR, verificăm performanţa SVM în învăţarea instanţelor. Vom folosi funcţia kernel liniară (Fig. 2.28), echivalentă primei versiuni de SVM şi comparativ, funcţia radială (Fig. 2.29). Este evident că, folosind funcţia kernel liniară, prima versiune de SVM, clasificatorul nu reuşeşte să înveţe datele liniar neseparabile.

```
> x1<-c(1,1,0,0)
> x2<-c(1,0,1,0)
> clasa<-c(0,1,1,0)
> dataTrain<-as.data.frame(t(rbind(clasa,x1,x2)))
> svmLinear <- svm(clasa ~., data = dataTrain, kernel="linear",
type="C+ +-classification")
> print(svmLinear)

Call:
svm(formula = clasa ~ ., data = dataTrain, kernel = "linear", type =
"C-+ +classification")

Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  1
      gamma:  0.5

Number of Support Vectors:  4

> svmLinear$fitted
1 2 3 4
1 1 1 1
Levels: 0 1
```

Fig. 2.28 – Soluția SVM cu kernel liniar la problema XOR.

```
> svmRadial <- svm(clasa ~., data = dataTrain, kernel="radial",
type="C-+ +classification")
> print(svmRadial)

Call:
svm(formula = clasa ~ ., data = dataTrain, kernel = "radial", type =
"C-+ +classification")

Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.5

Number of Support Vectors:  4

> svmRadial$fitted
1 2 3 4
0 1 1 0
Levels: 0 1
```

Fig. 2.29 – Soluția SVM cu kernel radial la problema XOR.

Funcţia radială însă reuşeşte transformarea neliniară a datelor XOR, astfel încât ele devin liniar separabile după transformare.

Abordarea SVM, datorită gradului ridicat de generalitate a soluţiei, posibilităţii de a separa instanţe liniar neseparabile şi a opera pe date de dimensiuni foarte mari, îi recomandă pentru studiile de analiză genetică. De asemenea, soluții pentru selectarea atributelor esențiale clădite pe SVM [45] au fost propuse și utilizate cu succes. Testăm în continuare comportamentul a SVM pe datele Golub cu 72 de instanţe și 7129 de atribute. Evaluăm performanţele SVM cu cele patru funcţii kernel (Fig. 2.30-2.33) implementate în pachetul MLInterfaces.

```
> # SVM linear
> ############################################################
> svmLinear = MLearn(ALL.AML ~ ., data=Golub_Merge, svmI, +
+kernel="linear", xvalSpec("LOG", 5, balKfold.xvspec(5)))
> confuMat(svmLinear)
     predicted
given ALL AML
  ALL  47   0
  AML   2  23
> accuracy
[1] 0.9722222
```

Fig. 2.30 – Clasificarea datelor Golub prin metoda SVM cu kernel liniar.

```
> # SVM radial
> ############################################################
>  svmRadial  =  MLearn(ALL.AML  ~  .,  data=Golub_Merge,  svmI,  +
+kernel="radial", xvalSpec("LOG", 5, balKfold.xvspec(5)))
> confuMat(svmRadial)
     predicted
given ALL AML
  ALL  42   5
  AML   3  22
> accuracy
[1] 0.8888889
```

Fig. 2.31 – Clasificarea datelor Golub prin metoda SVM cu kernel radial.

```
> # SVM sigmoid
> ############################################################
>  svmSigmoid  =  MLearn(ALL.AML  ~  .,  data=Golub_Merge,  svmI,  +
+kernel="sigmoid", xvalSpec("LOG", 5, balKfold.xvspec(5)))
> confuMat(svmSigmoid)
     predicted
given ALL AML
  ALL  46   1
  AML   2  23
> accuracy
[1] 0.9583333
```

Fig. 2.32 – Clasificarea datelor Golub prin metoda SVM cu kernel sigmoid.

```
> # SVM polynomial
> ############################################################
>  svmPolynomial  =  MLearn(ALL.AML  ~  .,  data=Golub_Merge,  svmI,  +
+kernel="polynomial", xvalSpec("LOG", 5, balKfold.xvspec(5)))
> confuMat(svmPolynomial)
     predicted
given ALL AML
  ALL  45   2
  AML  19   6
> accuracy
[1] 0.7083333
```

Fig. 2.33 – Clasificarea datelor Golub prin metoda SVM cu kernel polynomial.

Pentru acest set de date, cei patru clasificatori SVM, evaluaţi utilizând 5-fold cross validation, oferă rezultate încurajatoare dar nespectaculoase. Funcţiile kernel liniară, radială, sigmoidă și polinomială, produc rezultate diferite, dar datele studiate nu beneficiază de utilizarea funcţiilor de transformare testate. Funcţia kernel polinomială produce rezultate semnificativ mai slabe decât în spaţiul original al atributelor. Totuşi acest aspect nu exclude posibilitatea utilizării unei alte transformări, mai eficientă.

2.2.3.5. Selectarea atributelor

Selectarea atributelor are ca scop determinarea unui subset din totalul atributelor evaluate, care se dorește a fi subgrupul optim, ideal pentru învățarea datelor cu un anumit algoritm supervizat. Selectarea atributelor poate facilita descoperirea de marcheri cu valoare diagnostică pentru diferite patologii și înțelegerea intimă a proceselor biologice care descriu acea condiție la nivel molecular. Reducerea dimensionalității are ca scop selectarea atributelor esențiale și determinarea unei combinații optime dintre aceste atribute pentru a descrie problema studiată. Există două abordări diferite în acest sens: selectarea și extragerea atributelor. Distincția majoră între ele este că în selectarea atributelor se urmărește selectarea sub-setului optimal de atribute care descrie problema studiată. În extragerea atributelor, combinarea unor caracteristici în noi variabile care pot descrie fenomenul studiat este permisă.

Principiile care stau la baza selectării atributelor importante izvorăsc din statistică, în particular, analiza regresională. În cazul regresiei, testarea ipotezei statistice $\beta_k = 0$, este utilizată pentru a evalua importanța unei anumite variabile predictor în model. Pe baza acelor idei și principii, au fost dezvoltate metode statistice complexe și algoritmi de inteligență artificială special concepuți pentru acest deziderat. Impactul major al tehnologiei microarray în bioinformatică a determinat dezvoltarea de metode statistice și de IA, special destinate selectării atributelor în acest tip de experiment. Selectarea atributelor poate fi implicită sau explicită. În abordarea explicită, genele sunt tratate separat și ordonate în funcție de rezultatul unui test statistic, adesea t-test. Selectarea implicită a atributelor este parte indisolubilă din metodologia de clasificare.

Trei direcții majore în selectarea atributelor s-au bucurat de popularitate în diferite aplicații. O abordare uzuală constă în utilizarea metodelor de tip filtrare. În

această abordare, atributele importante sunt selectate pe baza proprietăților lor așa cum sunt ele reprezentate în setul de date, anterior utilizării lor pentru clasificare. În filtrare, fiecare genă poate fi considerată individual. Se aplică unul sau mai multe filtre și doar genele corespunzătoare sunt păstrate în setul de date pentru analize suplimentare. Ulterior, valoarea subsetului filtrat este evaluată utilizând o metodă supervizată de învățare. Relațiile dintre atribute nu pot fi surprinse în acest context. Două categorii de metode sunt utilizate adesea pentru filtrarea datelor în experimentele cu date ADN microarray: nespecifice și specifice.

Filtrarea nespecifică urmărește detectarea atributelor caracterizate de o variabilitate redusă în setul de date și eliminarea lor din cercetare. În acest context, nu se ține cont de diferitele clase prezente în setul de date. Se consideră că atributele care variază foarte puțin sau deloc în setul de date nu pot fi valoroase în descrierea claselor existente. Așadar eliminarea lor din analizele viitoare este utilă și reduce zgomotul prezent în date. În general, acest pas în analiza datelor abordează și problema valorilor absente (NA sau not available). Există diferite abordări pentru tratarea valorilor NA în analiza microarray, iar eliminarea acestora din analizele consecutive poate fi parte din filtrarea nespecifică.

Filtrarea specifică se efectuează considerând diferitele clase prezente în setul de date analizat. În acest tip de filtrare, scopul este de a detecta atribute care ar putea explica apartenența unui exemplu la o anumită clasă, cunoscută a fi reprezentată în setul de date. Diferite teste statistice sunt uzual utilizate pentru filtrare specifică (t-test, ANOVA, modelul Cox și ROC sunt câteva opțiuni). Setul de date trebuie, în general, explorat anterior filtrării specifice cu teste statistice. Oportunitatea utilizării unui anumit test statistic în defavoarea altora trebuie să corespundă configurației datelor studiate. Rezultatul filtrării specifice este reducerea atributelor la un număr care poate fi analizat și interpretat de către un biolog experimentat. În multe situații, în special în cazul clasificării binare, abordările de tip t-test oferă informații utile despre variabilele importante, dar rezultatele sunt departe de un subset optimal de atribute [46] pentru problema studiată. Extensii ale t-test special concepute pentru analiza microarray [47, 48] au fost propuse. Metode de ierarhizare a genelor diferențial exprimate au utilizat testul Wilcoxon sau t-test cu permutații. Aceste teste nu sunt constrânse de presupunerea că valorile atributelor provin dintr-o distribuție normală [49].

Pachetul genefilter [50] în Bioconductor este specializat în filtrarea genelor pentru experimentele cu ADN microarray. Cercetătorul are flexibilitatea de-a își defini propriile filtre în funcție de condițiile experimentului desfășurat. În analiza datelor microarray, filtrarea a fost adesea utilizată în preambulul clasificării [51, 52].

Abordarea wrapper evaluează subgrupuri de atribute pe baza performanței unui clasificator supervizat angajat să învețe datele utilizând acel subset. Căutarea în spațiul atributelor are o origine și un sens. Căutarea se poate desfășura fie adăugând, fie eliminând atribute din setul analizat. De asemenea, numărul atributelor prezente la inițializarea algoritmului caracterizează originea căutării și trebuie considerate în conceperea unui experiment de acest tip. Rezultatele unei astfel de căutări depind de clasificatorul supervizat ales și nu garantează universalitatea selecției. Algoritmii clădiți pe principiile hill-climbing pot fi aplicați în acest deziderat. Neajunsul acestor abordări constă în tendința acestor algoritmi de-a converge într-o soluție sub-optimală. Un răspuns la abordările deterministice îl reprezintă algoritmii stocastici care pot depăși optime locale în căutarea unei soluții optime globale în spațiul atributelor. Natura aleatorie a căutării cu metode stocastice impune necesitatea unui număr de repetiții pentru fiecare căutare pentru rezultate consistente. Rezultatele fiecărei căutări sunt sensibil diferite și o evaluare a

rezultatelor obținute la toate căutările efectuate se impune pentru a putea trage concluzii valide.

Avantajul acestei metode este că permite capturarea unor relații necunoscute între atribute, proprietate foarte dezirabilă în cazul analizei genelor diferențial exprimate. Neajunsul metodei în cazul microarray este că numărul mic de exemple dintr-un astfel de set de date expune analiza la pericolul de overfitting. Metode wrapper au fost de asemenea utilizate cu succes pentru identificarea grupurilor de gene diferențial exprimate [53] care pot exprima o relație de cauzalitate cu o anumită condiție. Algoritmii genetici au fost utilizați [54, 55] pentru atingerea acestei finalități. Abordări hibride au fost de asemenea propuse în trecut [56, 57].

Abordarea intrinsecă (embedded) a fost de asemenea utilizată mai recent pentru selectarea atributelor. În acest caz un selectarea atributelor este intrinsecă clasificatorului supervizat și are loc în timpul clasificării.

2.2.4. Evaluarea performanţei clasificatorilor

2.2.4.1. Teorema No Free Lunch

Teorema No Free Lunch abordează problematica performanţei generale a clasificatorilor. Teorema stabileşte că nu putem găsi un algoritm de clasificare, în general, superior „datului cu banul”. Dacă un clasificator este superior „datului cu banul” pe o anumită problemă este doar datorită potrivirii algoritmului la problema studiată. Deşi poate suna descurajator, teorema susţine că nu există un clasificator care are performanţe superioare celorlalţi pe toate problemele studiate. Consecinţa teoremei este că, pentru o problemă dată, trebuie căutat clasificatorul care oferă performanţa satisfăcătoare.

Aşa cum a fost enunţată de autorii ei, Wolpert şi Macready, teorema No Free Lunch [58] este „oricare doi algoritmi sunt echivalenţi când performanţa fiecăruia este considerată media performanţelor lui pe toate problemele posibile” [59].

Numele teoremei No Free Lunch își are sorgintea într-o metaforă adesea folosită pentru a simboliza semnificaţia teoremei. Considerăm că trei colegi de serviciu iau zilnic masa împreună la restaurant, iar unul este omnivor, altul este vegetarian şi altul este carnivor, dar este pe deasupra, foarte econom. Restaurantele au, în linii mari, acelaşi meniu. Preţurile diferă (un restaurant specializat în carne de porc, va avea preţuri bune pentru specialităţi şi preţuri mari pentru mâncarea vegetariană). Astfel, preţul plătit de carnivor este în general, mai mic, deoarece el fiind zgârcit alege restaurantele specializate în carne. Omnivorul va plăti în general aproximativ la fel, preţul plătit de el nu depinde de alegerea restaurantului. Vegetarianul este cel care plăteşte în general cel mai mult, deoarece mănâncă în restaurantele specializate în carne. Cel mai câştigat este carnivorul deoarece el foloseşte informaţii suplimentare pentru a optimiza preţul, şi anume, alege restaurantele despre care ştie a priori că sunt specializate în prepararea cărnii.

Este aşadar evidentă necesitatea de-a cuantifica performanţa clasificatorilor pe un set anumit de date şi a compara performanţe ale diferiţilor algoritmi, pe problema studiată, pentru a alege metoda cea mai potrivită cazului considerat.

2.2.4.2. Măsuri cantitative ale performanţei clasificatorilor

Clasificarea reprezintă procesul de învăţare dintr-un set de date format din instanţe cu valorile atributelor şi etichetele cunoscute. În esenţă, finalitatea clasificării constă în găsirea unei funcţii care asociază o etichetă oricărei secvenţe posibile de valori ale atributelor. Aşadar, este important ca funcţia rezultată în urma unui astfel proces de învăţare supervizat, să stabilească o corespondenţă cât mai exactă între instanţe din setul de date şi etichetele corespunzătoare, pe baza valorilor atributelor. Este de asemenea foarte important ca această funcţie să poată prezice corect etichetele unor date noi, care nu au fost prezente în setul de date ce a stat la baza învăţării. Clasificarea poate fi privită ca un raţionament inductiv.

În general, clasificarea constă în două procese diferite. Într-un prim pas, numit antrenament (training), se urmăreşte aproximarea cât mai acurată pe un set de date, a unei funcţii care returnează etichetele corecte ale instanţelor pe baza valorilor atributelor. Setul de date folosit pentru acest prim pas poartă numele de set de antrenament. Un al doilea pas presupune testarea funcţiei rezultate din procesul anterior pe instanţe noi. Finalitatea acestei testări constă în prezicerea etichetelor noilor exemple, aplicând funcţia rezultată la pasul anterior, pe valorile atributelor noilor instanţe.

Procesul de testare al noilor instanţe constă în prezicerea etichetelor exemplelor aplicând funcţia găsită în urma pasului de antrenament pe valorile corespunzătoare ale atributelor. Acest pas este echivalent cu o testare de ipoteze statistice. Pentru fiecare instanţă se poate defini o *ipoteză nulă*, presupunere asupra stării de fapt, de forma: instanţa x aparţine clasei A. Automat, pentru fiecare instanţă este definită şi *ipoteza alternativă*: instanţa x nu aparţine clasei A. Dacă ne confruntăm cu două clase de instanţe, de exemplu pacienţi bolnavi şi sănătoşi, ipoteza nulă ar putea fi: pacientul este bolnav. Ipoteza alternativă, ar fi în acest caz: pacientul nu este bolnav. Pentru ipoteza alternativă este semantic echivalent: pacientul este sănătos. Ulterior, aceste ipoteze statistice sunt testate şi confirmate sau infirmate. Acceptarea ipotezei nule atrage respingerea ipotezei alternative şi invers. Funcţia rezultată în urma antrenamentului reprezintă tocmai un test statistic asupra acestor ipoteze.

Rezultatele obţinute în urma testării ipotezelor statistice oferă informaţia cantitativă asupra performanţei clasificării. Astfel, ne aşteptăm ca o parte dintre instanţele testate să fie clasificate corect, iar un număr de exemple să fie etichetate eronat. Evident, ne-am dori ca numărul erorilor de etichetare să fie minim pentru a putea considera că am obţinut o clasificare performantă a datelor de testare. Erorile care pot apărea în urma unui proces de testare de ipoteze statistice sunt de două tipuri:

1. **Eroare de tipul I** sau **„eroare α ”** = eroarea de a respinge ipoteza nulă, când, în realitate, ea era adevărată. Rezultatul eronat survenit poartă numele de ***fals negativ***.
2. **Eroare de tipul II** sau **„eroare β ”** = eroarea de a accepta ipoteza nulă, când, în realitate, era falsă, iar ipoteza alternativă era adevărată. Rezultatul eronat obţinut poartă numele de ***fals pozitiv***.

Aceste rezultate, pentru toate instanţele testate, pot fi rezumate într-un tabel care poartă numele de *matrice de confuzie* (tabelul 2.14). Această matrice cuprinde numărul tuturor instanţelor testate, după rezultatul obţinut.

Tabel 2.13 - Paşii procesului de clasificare

Nr.	Proces	Date utilizate	Finalitate
1	Antrenament	Set de antrenament	găsirea unei funcţii, astfel încât, f(valoriatribute)= etichetăcunoscută pentru fiecare instanţă a setului de date de antrenament
2	Testare	Instanţe de testare	Utilizarea funcţiei cu valori discrete, rezultate în pasul anterior, pentru prezicerea f(valoriatribute)= etichetaprezisa asupra instanţelor de testare

Tabel 2.14 – Matricea de confuzie

		Starea reală	
		Pozitiv	**Negativ**
Rezultatul Modelului	**Pozitiv**	Adevărat pozitive (True Positive)	Fals pozitive (False Positive) (Eroare de tip I)
	Negativ	Fals negative (False Negative) (Eroare de tip II)	Adevărat negative (True Negative)

Să revenim la exemplele în care studiem un număr de pacienţi, o parte bolnavi, cu un anumit diagnostic şi o altă parte sănătoşi, iar ipoteza nulă a este: pacientul este bolnav. În cadranul din stânga-sus a matricei de confuzie apare numărul tuturor instanţelor clasificate bolnav, care reprezintă pacienţi în realitate suferinzi. În cardanul din dreapta-jos, apare numărul indivizilor etichetaţi drept sănătoşi, care sunt în realitate sănătoşi. Dacă această diagonală a matricei de confuzie reprezintă instanţele clasificate corect, separate în funcţie de etichetele reale, cealaltă diagonală va însuma erorile de clasificare. Cadranul din dreapta-sus prezintă fals pozitivele, reprezentând indivizi sănătoşi, clasificaţi greşit de modelul studiat drept bolnavi. Cadranul din stânga-jos, prezintă fals negativele adică instanţele clasificate drept indivizi sănătoşi, în realitate fiind oameni bolnavi.

Rezultatele prezentate în matricea de confuzie stau la baza măsurilor cantitative ale performanţei clasificării. Cea mai utilizată dintre aceste măsuri este probabil acurateţea clasificării.

Acurateţea reprezintă proporţia pacienţilor clasificaţi corect, în conformitate cu situaţia lor reală, fie că ei aparţin unei clase sau alteia. Formula de calcul a acurateţei (2.19) raportează numărul instanţelor clasificate corect, numărului total de instanţe testate. În cazul testării indivizilor bolnavi şi sănătoşi, acurateţea este reprezentată de raportul tuturor indivizilor etichetaţi corect, fie că ei sunt sănătoşi sau bolnavi, cu numărul total al instanţelor clasificate.

$$Acuratete = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.19}$$

Este foarte interesant de analizat în ce măsură clasificatorul poate eticheta corect o anumită clasă. În cazul testării pacienţilor este foarte interesant de cuantificat în ce măsură un anumit test, în acest caz clasificatorul, poate detecta pacienţii cu un anumit diagnostic. Această măsură este oferită de raportul dintre exemplele corect detectate ca fiind oameni bolnavi cu totalul exemplelor de indivizi bolnavi prezenţi în setul de testare (2.20) şi poartă numele de *sensibilitate* a clasificării.

$$Senzitivitate = \frac{TP}{TP + FN} \tag{2.20}$$

O altă măsură foarte utilizată a rezultatului clasificării pe o singură clasă de instanţe este *specificitatea*. Ea reprezintă proporţia de instanţe clasificate ca negative în numărul real de negative din setul de antrenament (2.21).

$$Specificit\ ate = \frac{TN}{TN + FP} \tag{2.21}$$

Deşi utilităţile sensibilităţii şi specificităţii sunt oarecum limitate, deoarece, niciuna nu oferă separat imaginea de ansamblu a performanţei clasificării, împreună sunt foarte valoroase. De exemplu, pentru un test cu aplicabilitate medicală, este de aşteptat să nu ofere rezultate ideale, sensibilitate 100% şi specificitate 100%. Această situaţie ideală ar însemna că toţi indivizii testaţi sunt etichetaţi corect, dacă suferă de o anumită boală sunt diagnosticaţi prin metoda utilizată, iar dacă sunt sănătoşi sunt perfect clasificaţi ca atare prin testul folosit. În general, metodele de testare din medicină au o anumită marjă de eroare acceptată. Aşadar este foarte important uneori, de stabilit performanţa unei clasificări pe fiecare clasă. În funcţie de situaţie, este uneori de preferat ca un individ bolnav să nu fie sub nici o formă ratat, chiar cu riscul de a trata în exces, chiar şi indivizi sănătoşi. Într-o astfel de situaţie, este de dorit ca sensibilitatea unui test să fie cât mai mare, chiar în dauna unei specificităţi reduse. Alteori este acceptabilă eroarea pe anumiţi indivizi, cu intenţia de-a nu trata inutil oameni sănătoşi.

Spre exemplu, un test la modă este PSA (Prostate-Specific Antigen) pentru detectarea cancerului de prostată. Deşi un indicator bun al cancerului de prostată, testul PSA oferă numeroase rezultate fals pozitive, deoarece reprezintă un marker specific activităţii prostatei şi nu carcinomului. Aşadar, rezultatele PSA pot fi crescute în afecţiuni ale prostatei şi altele decât carcinomul. Deoarece carcinomul de prostată reprezintă o afecţiune foarte agresivă, iar prognosticul bolii depinde de faza în care este diagnosticată, la bărbaţii de 50-75 de ani este preferat o atitudine mai agresivă în diagnosticare şi tratament. Pentru această grupă de pacienţi este foarte importantă specificitatea testului, iar un test pozitiv duce la teste diagnostice invazive suplimentare. Este de preferat în această situaţie, ca pacientul să fie supus metodelor invazive de diagnosticare chiar cu riscul de-a fi executate unui individ sănătos, pentru a exclude diagnosticul de carcinom. Pentru pacienţii de peste 75 de ani, rezultatul pozitiv al PSA nu mai este la fel de alarmant, iar pentru aceste situaţii, deoarece evoluţia bolii se poate întinde pe 10 ani, iar manevrele invazive de diagnostic şi tratament sunt caracterizate de un grad crescut de risc, este de preferat o atitudine mai defensivă a terapeutului. Aşadar, medicul este mai interesat în astfel de situaţii de aspectul specificităţii testului şi acţionează în consecinţă.

2.2.4.3. Bias şi Varianţă

Asemenea studiilor statistice, analizele de Machine Learning sunt afectate de două largi clase de erori. *Erorile statistice* reprezintă diferenţa dintre o valoare estimată sau măsurată şi valoarea reală cauzată de variaţii aleatorii şi sunt în general independente. *Erorile de sistem* sunt diferenţe între valoarea estimată sau măsurată şi valoarea reală cauzată de variaţii constante şi sunt dependente.

Faza de antrenament a procesului de învăţare supervizată, reprezintă de fapt, determinarea unei funcţii care are ca argumente valorile atributelor instanţelor şi returnează etichetele lor. O sursă a erorilor care pot apărea atât la antrenament cât şi la testare se datorează probabil faptului că funcţia găsită prin antrenament aproximează doar funcţia reală, cea care a guvernat apariţia setului de date de antrenament.

Vom nota cu $f(x)$ funcţia care a generat datele de antrenament, iar cu $\hat{f}(x)$ funcţia estimată de un clasificator prin antrenament.

Deviaţia pătratică medie (MSE, din englezescul Mean Squarred Root Error) oferă măsura cantitativă a modului în care $\hat{f}(x)$ reuşeşte să aproximeze funcţia care a generat în realitate datele de antrenament $f(x)$ şi se defineşte ca:

$$MSE = E\{[\hat{f}(x) - f(x)]^2\} = \{E[\hat{f}(x) - f(x)]\}^2 + E\{[\hat{f}(x)]^2\} - \{E[\hat{f}(x)]\}^2 \qquad (2.22)$$

Primul termen al ecuaţiei (5.3.1) reprezintă pătratul unei măsuri ce poartă numele de *bias* şi reprezintă diferenţa dintre valoarea estimată şi valoarea reală (2.23). Bias-ul reprezintă o măsură a acurateţei cu care $\hat{f}(x)$ îl putem aproximează pe $f(x)$.

$$Bias = E[\hat{f}(x) - f(x)] \qquad (2.23)$$

A doua parte a ecuaţiei (5.3.1) poartă numele de *varianţă* şi oferă măsura variaţiei valorii estimate pentru $f(x)$, când setul de training variază (2.24). Varianţa cuantifică specificitatea.

$$Varianţa = E\{[\hat{f}(x)]^2\} - \{E[\hat{f}(x)]\}^2 \qquad (2.24)$$

Există o strânsă relaţie între bias şi varianţă, complexitatea clasificatorului şi dimensiunea setului de antrenament [26].

O modalitate larg utilizată pentru reducerea varianţei poartă numele de *Bagging*. Abordarea Bagging constă în antrenarea mai multor clasificatori pe seturi de antrenament diferite, formate din subseturi de instanţe alese aleatoriu din setul de antrenament original, cu înlocuire. Rezultatul final al clasificării este dat de votul majoritar al tuturor clasificatorilor antrenaţi în acest fel.

O altă abordare se adresează reducerii bias-ului şi poartă numele de *boosting*. Metoda boosting presupune antrenarea unui clasificator pe tot setul de date de antrenament. Instanţele clasificate greşit de către acest clasificator sunt identificate şi se trece la crearea unui nou set de antrenament, în care se pune accent pe instanţele clasificate anterior greşit. Un nou clasificator este antrenat pe acest set de date şi se identifică din nou instanţele etichetate greşit. Rezultatul final al clasificării după un număr de repetiţii ai paşilor de mai sus este oferit de votul majoritar al tuturor clasificatorilor antrenaţi în acest mod. Există diferite moduri de a atribui ponderi sporite instanţelor clasificate greşit sau a descreşte ponderile exemplelor etichetate corect. La baza acestei abordări stă presupunerea că, se poate obţine un clasificator foarte puternic prin combinarea unui număr arbitrar de clasificatori slabi. Există diferite implementări care au la bază ideea de boosting; poate cel mai cunoscut algoritm fiind AdaBoost.

2.2.4.4. Evaluarea şi compararea performanţei clasificatorilor

Finalitatea unui clasificator este prezicerea etichetelor unor instanţe noi, care nu sunt disponibile în momentul antrenării. Este aşadar, foarte importantă testarea posibilităţilor de generalizare a unui clasificator nou creat, pe anumite date. O modalitate de-a testa clasificatorul pe date noi constă în separarea unor instanţe din setul de antrenament şi tratarea lor ca date de testare. Deoarece rezultatul antrenamentului depinde indisolubil de volumul şi configuraţia setului de date utilizat, sunt utilizate diferite abordări în separarea setului de date original în instanţe de antrenament şi exemple de testare.

2.2.4.4.1. Metoda Hold-out

Abordarea Hold-out constă din eliminarea unui subset de instanţe din setul de antrenament şi tratarea lor ca date de testare. Subsetul de date, format din instanţele eliminate din setul de antrenament original, poartă numele de *set de validare*. Astfel, clasificatorul este antrenat pe instanţele rămase în setul de antrenament după eliminarea subsetului de validare, iar performanţa clasificatorului este testată pe setul de validare. Instanţele din setul de validare nu sunt utilizate pentru antrenament.

Modul de selectare al instanţelor de validare din setul de antrenament, cât şi volumul acestui subset, sunt flexibile. Totuşi, este important ca setul de date de validare să nu reprezinte o proporţie prea mare din setul de date original,mai ales când numărul instanţelor de antrenament disponibile este redus. De asemenea, este de dorit ca instanţele setului de validare să fie reprezentative pentru clasele implicate, proporţii aproximativ egale de exemple aparţinând tuturor claselor studiate sunt recomandabile.

2.2.4.4.2. Metoda "leave-one-out" de validare încrucişată

Abordarea leave-one-out adresează mult mai bine decât hold-out atât problema dimensiunii setului de antrenament cât şi a configuraţiei particulare a datelor. Practic, dacă setul de antrenament este format din *n* instanţe, sunt create *n* seturi de antrenament, fiecare set fiind reprezentat de setul original din care a fost extras, câte un exemplu. Consecutiv, este antrenat clasificatorul studiat de *n* ori, pe cele *n* seturi de antrenament rezultate, şi testat de fiecare dată pe instanţa care nu a făcut parte din setul de antrenament pe care a fost creat. Acest mod de a antrena şi testa încrucişat clasificatorul stă la baza numelui acestei tehnici de validare.

Performanţa clasificatorului este considerată a fi media performanţelor pe fiecare din cele *n* instanţe de testare. În general, metodele încrucişate de validare oferă o mai bună imagine asupra performanţei clasificatorului pe date noi de testare.

2.2.4.4.3. Metoda k-fold de validare încrucişată

În spiritul ideii folosite şi în abordarea leave-one-out, se poate împărţi setul de antrenament original în *k* subseturi de dimensiuni aproximativ egale. Antrenăm ulterior clasificatorul de *k* ori. La fiecare pas de antrenare eliminăm din setul de original unul dintre cele *k* subseturi. Practic, la fiecare antrenare, folosim ca set de date un subset format din cele *k-1* subseturi rămase din setul original şi evaluăm performanţa clasificatorului astfel antrenat, pe setul subsetul ignorat în faza de antrenament. Performanţa clasificatorului este media performanţelor obţinute pe cele *k* seturi de testare.

Valoarea lui *k*, dimensiunea setului de date de testare poate fi aleasă de către utilizator, dar trebuie să respecte condiţiile impuse de setul de antrenament original. O valoare prea mare a lui *k* pe un set de date de antrenament restrâns, anulează avantajele metodei. Pentru valoarea *k=1*, algoritmul devine echivalent cu metoda leave-one-out. Totuşi, pentru clasificatorii care necesită calcule complexe pe seturi de date foarte vaste, alegerea unui *k* mai mare poate însemna o reducere majoră a duratei de execuţie a algoritmului.

2.3. Concluzii

Studiile ADN microarray au ca finalitate descoperirea unui grup restrâns de gene care sunt legate cauzal de evoluţia unei anumite patologii. În general, se urmăreşte posibilitatea diagnosticării precoce a patologiei studiate prin analiza expresiei genetice. Succesul acestei abordări este extrem de important pentru pacienţii care suferă de boli cu evoluţii foarte rapide, tratabile eficient în stadiile incipiente, dar netratabile în stadiile avansate. În plus, succesul în determinarea genelor legate cauzal de o anumită boală oferă premisele înţelegerii corecte a etiologiei şi pune bazele elaborării de tratamente eficiente împotriva respectivei patologii.

Într-un prim pas al analizei PR este recomandabilă vizualizarea datelor cu diferite metode şi stabilirea unor măsuri potrivite pentru descrierea corectă a similarităţii sau di-similarităţii între instanţele studiate. Măsurile utilizate pentru cuantificarea similarităţii sau di-similarităţii între obiecte au un impact important în analiza datelor cu metodele inteligenței artificiale. În general, iar pentru cercetarea datelor ADN microarray în particular, alegerea metodei potrivite pentru un anumit set de date, a priori învățării nesupervizate sau supervizate, este indicată pentru rezultate consistente.

Ulterior, utilizarea metodelor de grupare poate evidenţia forme în structura datelor, fără a utiliza informaţia apartenenţei instanţelor la diferitele clase. Acest aspect reprezintă particularitatea fundamentală a metodelor de învăţare nesupervizate. Am distins două abordări diferite în atingerea acestui deziderat.

1) Metodele de grupare ierarhică returnează o structură a datelor, care exprimă diferitele grade de similaritate sau di-similaritate ale instanţelor. Această abordare oferă cercetătorului posibilitatea de-a descoperi particularităţi ale patologiei studiate care nu au fost prevăzute.
2) Metodele de grupare care returnează un număr prestabilit de grupe din instanţele prezente în date sunt foarte utile când cercetătorul urmăreşte studierea similitudinilor sau deosebirilor dintr-un set de instanţe aparţinând unor diagnostice diferite sau unor stadii diferite în evoluţia unei patologii.

În cazul învățării nesupervizate, nu se cunosc a priori informații despre clasele prezente în setul de date, iar învățarea are ca finalitate determinarea acestor clase. O aplicație comună a învățării nesupervizate în analiza expresiei genetice este tentativa de a descoperi noi clase de tumori. Caracteristica esenţială comună metodelor de grupare prezentate este finalitatea de-a descoperi forme în datele studiate, fără a utiliza informaţia apartenenţei instanţelor la diferitele clase. Acest aspect reprezintă particularitatea fundamentală a metodelor de învăţare nesupervizate. Metodele de grupare urmăresc descoperirea unor forme în structura datelor de ADN microarray. Finalitatea unui studiu de ADN microarray este adesea descoperirea unui grup restrâns de gene care sunt legate cauzal şi pot explica evoluţia unei anumite patologii. În acest context, deşi metodele de grupare nu pot atinge acest deziderat, ele oferă posibilitatea înţelegerii setului de date studiat şi determină premise pentru alegerea metodelor de studiu suplimentare. În plus, metode complexe destinate descoperirii unui subgrup de gene esențial în explicarea unei patologii, au fost dezvoltate pe principii ale tehnicilor de grupare [35].

Utilitatea şi performanţa foarte bună a unor metode relativ simple de grupare este confirmată de analiza datelor Golub din exemplul ilustrat. Deşi nici o metodă nu descoperă forme satisfăcătoare în datele studiate, ele sunt încurajatoare şi oferă speranţa unor performanţe foarte ridicate în pasul învăţării supervizate. Este evident, în rezultatele obţinute, că există asemănări între pacienţii diagnosticaţi cu ALL şi că pacienţii diagnosticaţi AML prezintă expresii genetice similare. Este, de asemenea, vădit că există deosebiri între expresiile genetice ale pacienţilor diagnosticaţi cu ALL şi cei care suferă de AML. Este necesară analiza suplimentară a datelor, cu metode mai complexe, pentru a înţelege mai bine aceste tendinţe şi a stabili cauzalitatea între un grup restrâns de gene şi patologia studiată.

Metodele de clasificare prezentate în acest capitol oferă soluţii complexe la problema discriminării claselor în studiile de ADN microarray. Fie că învăţarea abordează instanţele separat (kNN) sau laolaltă, fie că urmăresc determinarea unei suprafeţe predefinite pentru a separa datele sau determină probabilitatea apartenenţei unei instanţe la o clasă, fiecare dintre metodele discutate prezintă avantaje şi dezavantaje.

Performanţa perfectă la antrenament, situaţia când un clasificator reuşeşte separarea perfectă a datelor de antrenament, nu este neapărat soluţia ideală. Finalitatea învăţării supervizate este utilizarea clasificatorului construit pe instanţe noi. Astfel, este greu de crezut că datele de antrenament surprind perfect toate aspectele şi particularităţile instanţelor ce aparţin fiecărei clase studiate. În practică, datorită costurilor crescute şi a posibilităţii de apariţie a unor erori în colectarea datelor, este de aşteptat ca setul de antrenament să nu reprezinte perfect problema studiată. Este foarte posibil ca învăţarea pe un set de antrenament imperfect să nu determine o metodă perfectă de prezicere a etichetelor unor noi instanţe. Situaţia unui rezultat perfect în urma clasificării pe setul de antrenament, dar cu posibilităţi restrânse de generalizare pe date de testare poartă numele de *overfitting* şi se datorează antrenării excesive a clasificatorului pe particularităţile setului de antrenament.

Se impune, aşadar, determinarea unor metode de a alege abordarea ideală pentru o problemă dată. În acest sens, este de dorit stabilirea unor măsuri cantitative care cuantifică performanţa fiecărui clasificator şi permit compararea lor. Aceste măsuri trebuie să ofere atât imaginea performanţei clasificatorului la antrenament cât şi posibilitatea de-a îl utiliza pe date noi, indisponibile la momentul învăţării. Măsurile de evaluare a performanţei şi validare expuse, oferă premise solide pentru găsirea clasificatorului potrivit într-un anumit context, estimarea dependenţei rezultatelor clasificării de particularităţile setului de date şi prevenirea antrenamentului excesiv.

În conformitate cu teorema No Free Lunch, nu există un clasificator ideal pentru orice problemă de învăţare supervizată şi consecutiv, pentru orice studiu de tip ADN microarray. Structura datelor şi rezultatele obţinute în urma aplicării metodelor de învăţare nesupervizată pot ghida cercetătorul în efortul de a găsi clasificatorul ideal pentru o anumită problemă. Totuşi, se impune necesitatea evaluării şi comparării performanţelor mai multor clasificatori pentru decelarea unei soluţii convenabile pentru fiecare studiu de PR.

Utilizarea acestor metode foarte eficiente este facilitată de mediul R prin implementările orientate pe studiile de bioinformatică, foarte flexibile, cu posibilităţi de extensie foarte facile şi foarte bine documentate.

Pentru atingerea finalităţii studiilor de microarray, determinarea unui grup restrâns de gene care pot fi legate cauzal de o anumită patologie, sunt de importanţă capitală metodele de selecţie a atributelor (oligonucleotidelor) semnificative. Aceste metode reprezintă direcţia de dezvoltare a lucrării de faţă şi se concretizează în propunerea metodei evoluate de selectare a atributelor semnificative prezentată în capitolul următor.

3. METODĂ PROPUSĂ PENTRU SELECTAREA UNUI NUMĂR RESTRÂNS DE ATRIBUTE, INTERPRETABILE DIN PUNCT DE VEDERE BIOLOGIC

Algoritmii evoluționiști (AE) utilizează principii din evoluția naturală pentru a oferi răspunsuri la probleme de optimizare. Principiile evoluției naturale sunt testate pe parcursul a miliarde de ani de continuă adaptare la mediul înconjurător și optimizare. Deoarece inteligența stă la baza capacității de adaptare la condiții noi din mediul înconjurător, pare natural ca domeniul inteligenței artificiale să exploreze posibilitatea utilizării acestor principii, cu eficiență atât de solid confirmată diacronic. Bioinformatica este una dintre disciplinele care au beneficiat semnificativ de dezvoltarea în domeniul AE [60].

Algoritmii genetici (AG) fac parte din domeniul AE și au fost aplicați cu succes în diferite probleme de optimizare. De asemenea, algoritmii genetici au devenit o metodă stocastică de elecție pentru a selecta atribute în diferite domenii.

Obiectivul nostru este selectarea unui număr restrâns de atribute, interpretabile biologic, din date achiziționate cu tehnologia ADN microarray. Propunem o metodă îmbunătățită, fundamentată pe AG și concepută pentru selectarea atributelor în general, dar optimizată pentru aplicațiile cu date microarray. Modelăm fenomene care fundamentează evoluția naturală cu scopul ameliorării performanței AG în acest context.

Metoda descrisă în continuare, gravitează în jurul unui algoritm genetic diploid, dar beneficiază de dezvoltări originale modelate din evoluția naturală:

1. o abordare inspirată de **dominanța incompletă** pentru maparea genotipului la fenotip,
2. un operator fasonat după fenomenul **atribuirii aleatorii a cromozomilor** în timpul meiozei,
3. alternative de **operatori pentru mutație**, inspirați din genetica umană, concepuți pentru particularitățile studiilor microarray.

3.1. Algoritmii genetici

Algoritmii genetici au fost teoretizați de către Holland în urmă cu cinci zeci de ani. Tot Holland, a introdus noțiunea de schemă și teorema schemelor [61] pentru a formaliza procesul de evoluție în AG. Propunerea clasică, adesea menționată ca AG simplu în literatură, deși limitată ca aplicabilitate de codificarea binară și reprezentarea haploidă s-a bucurat de un imens succes și a reprezentat o inflexiune în inteligența artificială. De atunci, versiunea inițială a fost îmbunătățită și adaptată pentru a răspunde unei palete variate de aplicații. Abordarea AG simplă a fost extinsă în moduri variate cu reprezentarea diploidă a cromozomilor [62] și diferiți noi operatori, o parte importantă modelați pe observații din evoluția naturală [63].

Finalitatea AG este evoluția înspre soluția care optimizează un criteriu prestabilit. Nomenclatura AG este împrumutată din genetică pentru a sublinia sorgintea principiilor în evoluția naturală. Populația este formată din indivizi, reprezentând soluții. Indivizii sunt codificați de un genotip care se exprimă prin fenotip. Genotipul codifică atribute în forma unui șir de gene. În reprezentarea binară clasică, genele codifică atribute, respectă poziții fixe în genotip, numite loci și au alele cu valorile 0 și 1. Testarea adaptabilității unui fenotip la mediul înconjurător, problema propusă, se realizează prin evaluarea unei funcții fitness. Rezultatul acestei evaluări reprezintă adaptabilitatea sau fitness-ul individului. Adaptarea exploatează structura mediului înconjurător, depinde atât de mediul înconjurător cât și de funcția fitness aleasă. Pseudocodul prezentat în Fig. 3.1, ilustrează abordarea generală în implementarea algoritmilor genetici.

Algoritmii genetici sunt eficienți în a găsi direcțiile optime de căutare în spațiul soluțiilor. Pentru determinarea soluțiilor optime, abordarea hill-climbing este mai potrivită. Un context în care algoritmii genetici sunt de preferat altor metode de căutare [64] este conturat de câteva criterii:

1) spațiul soluțiilor este vast și nu se cunosc informații despre configurația lui,
2) funcția fitness este afectată de zgomot,
3) scopul căutării este satisfăcut și prin găsirea unui optim local.

Aceste condiții descriu excelent condițiile din activitatea de selectare a atributelor în experimentele ADN microarray. De asemenea o consecință a teoremei schemelor este că un AG favorizează perpetuarea schemelor scurte, cu fitness mediu, superior mediei populației. În contextul datelor microarray, ne propunem selectarea unui grup restrâns de gene care poate descrie fenotipul studiat, dintr-un număr imens de candidați. Cadrul general de aplicabilitate al AG constituie un argument pentru testarea utilității lor în acest context. În mod tradițional, rezultatele unei analize microarray sunt evaluate cu metode statistice. Importanța genelor individuale pentru o condiție analizată a fost adesea exprimată în funcție de p-value și fold change. Modele ANOVA au fost, de asemenea, utilizate cu succes în scopul identificării genelor diferențial exprimate cu tehnologia microarray [65]. Aceste proprietăți nu descriu importanța biologică a genelor sau contribuția lor într-un anumit proces. Validarea biologică a datelor se realizează manual și depinde foarte mult de experiența cercetătorului. Analiza genelor diferențial exprimate urmărește identificare unor grupuri de gene cu semnificație patologică, mai degrabă decât gene individuale. Este de așteptat ca gene co-regulate sau gene care sunt implicate în același pathway să fie descoperite într-un experiment pentru determinarea genelor diferențial exprimate cu tehnologia microarray. O parte dintre

```
Start
generație=0
GENEREAZĂ populația inițială
EVALUEAZĂ indivizii din populația inițială
REPETĂ
       ÎMPERECHEAZĂ părinții
       RECOMBINĂ părinții și generează moștenitori
       APLICĂ MUTAȚII cu o probabilitate predefinită
       EVALUEAZĂ indivizii
       SELECTEAZĂ indivizii pentru generația următoare
       generație = generație + 1
PÂNĂ CÂND condiția de terminare este satisfăcută
Stop
```

Fig. 3.1 – Pseudocod pentru implementarea generală a AG

clasificatorii supervizați permit, concomitent cu atingerea obiectivului principal, clasificarea și decelarea unor caracteristici utilizabile în selectarea atributelor importante. În general, o analiză separată a atributelor importante este însă necesară pentru a determina sub-grupul de atribute care poate descrie o relație cauzală cu configurația claselor reprezentate într-un set de date.

Un AG persevereză în două activități: explorare și exploatare [64]. Algoritmul explorează soluții noi pentru a se adapta mai bine la mediul înconjurător. Concomitent, exploatează adaptările deja dobândite pe parcursul căutării. Pentru ca o căutare cu AG să fie eficientă, trebuie să existe un echilibru între aceste două activități. Dacă exploatarea este favorizată excesiv, există riscul de overfitting. Dacă explorarea este favorizată evoluția poate fi împiedicată.

3.1.1. Inițializarea AG

Populația inițială de indivizi este, în general, generată fortuit dintr-o distribuție uniformă discretă. Datorită acestui aspect, replicații ale unei căutări cu un algoritm genetic converg adesea înspre soluții diferite. Acest aspect a fost adresat prin replicări multiple ale căutării sau alternative deterministice la generarea aleatorie a populației inițiale.

3.1.2. Recombinarea

Recombinarea modelează un principiu din biologia celulară care stă la baza diversității genetice naturale. În timpul meiozei, are loc un schimb de informație genetică, crossing-over, între cromozomi omologi. Analog, în algoritmii genetici, operatori specializați susțin evoluția prin recombinarea a segmente din genotipul părinților în momentul generării moștenitorilor. Ipoteza Building block [66] subliniază importanța recombinărilor în procesul de evoluție.

Petru ca evoluția să aibă loc, nu este suficientă doar o recombinare a atributelor separat codificate în genotip. Este de dorit ca recombinările să provoace structuri funcționale noi, cu impact în fenotip [67]. Câștigul în adaptabilitate astfel obținut, determină reprezentarea superioară a codificării lor genetice în generațiile ulterioare. În algoritmii genetici, de obicei, recombinările au loc cu o probabilitate prestabilită la inițializarea algoritmului. În timp, au fost propuși diferiți operatori pentru a susține evoluția în algoritmii genetici. Operatorii clasici, care s-au bucurat de succes major, sunt prezentați în continuare.

3.1.2.1. Recombinarea într-un punct

În această variantă un locus este ales aleatoriu. Moștenitorul va fi codificat de o parte din genotip identică unui părinte, până la locus-ul respectiv, combinată cu un segment de genotip de la alt părinte, după același locus. Recombinarea într-un punct este destul de limitată în termeni de combinații care pot fi generate și este foarte afectată de influența poziției, locus-ul genei în genotip (positional bias) [68].

Genele cu poziții foarte apropiate sunt moștenite preferențial. Fragmentele de cromozom recombinate conțin întotdeauna capete ale lanțului, ceea ce imprimă acestor zone din cromozomi o tendință specială de recombinare. În consecință, ambele capete ale unui genotip nu vor fi niciodată conservate în moștenitor. Ilustrarea recombinării într-un punct este prezentată în Fig. 3.2.

3.1.2.2. Recombinarea în două puncte

Recombinarea în două puncte ameliorează (Fig. 3.3) unele dintre dificultățile cu care se confruntă versiunea anterioară. Deși este afectată de influența poziției, această variantă nu mai conferă statut special capetelor lanțului și poate genera scheme inaccesibile recombinării într-un punct. În acest caz, două locus-uri sunt alese fortuit, iar segmentul dintre ele va fi obiectul schimbului de gene.

3.1.2.3. Recombinarea uniformă

Influența poziției asupra recombinării a fost adresată în recombinarea uniformă (Fig. 3.4). În această abordare fiecare locus este independent afectat de recombinare cu o probabilitate prestabilită. Utilitatea acestei variante este limitată în datele ADN microarray. Finalitatea de-a determina grupuri de gene care explică un fenotip nu este servită în acest context.

```
> Parinte1
          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
Parinte1     1    1    1    1    1    1    1    1    1     1
> Parinte2
          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
Parinte2     0    0    0    0    0    0    0    0    0     0
> Crossover1Punct(Parinte1, Parinte2)
          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
Copil1       0    0    0    1    1    1    1    1    1     1
Copil2       1    1    1    0    0    0    0    0    0     0
```

Fig. 3.2 – Recombinarea într-un punct

```
> Parinte1
          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
Parinte1     1    1    1    1    1    1    1    1    1     1
> Parinte2
          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
Parinte2     0    0    0    0    0    0    0    0    0     0
> Crossover2Puncte(Parinte1, Parinte2)
          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
Copil1       1    1    0    0    1    1    1    1    1     1
Copil2       0    0    1    1    0    0    0    0    0     0
```

Fig. 3.3 – Recombinarea în două puncte

```
> Parinte1
          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
Parinte1     1    1    1    1    1    1    1    1    1     1
> Parinte2
          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
Parinte2     0    0    0    0    0    0    0    0    0     0
> CrossoverUniform(Parinte1, Parinte2)
          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
Copil1       1    0    1    0    1    1    0    1    0     1
Copil2       0    1    0    1    0    0    1    0    1     0
```

Fig. 3.4 – Recombinarea uniformă

3.1.3. Mutația

Introducerea operațiunii de mutație în algoritmii genetici de asemenea, modelează un proces din evoluția naturală. Genotipul poate fi alterat prin multiple mecanisme, toate rezultând într-o schimbare la nivelul informației genetice care își găsește sau nu, exprimarea în fenotip. De asemenea, schimbările apărute în genotip pot fi transmisibile generației următoare sau nu. Pentru ca o schimbare în genotip să poată fi considerată mutație, ea trebuie să fie transmisibilă moștenitorilor. Mutația poate transfera generației viitoare caracteristici dezirabile, care suportă adaptarea la mediu, sau proprietăți indezirabile, care dimpotrivă, alterează fitness-ul purtătorului. Deoarece pentru a putea discuta despre mutație, ea trebuie să fie cel puțin o dată transmisă, efectele induse pot fi evaluate numai la generația succesoare apariției. Datorită acestor incertitudini cu privire la impactul asupra evoluției, probabilitatea ca o mutație să apară este, în general, mult mai mică decât în cazul recombinărilor în algoritmii genetici.

Tendința AG de-a converge într-un optim local în defavoarea unuia global a fost adresată prin metode menite să mențină diversitatea în generațiile înaintate. Alegerea operatorilor de recombinare și mutație au un rol semnificativ în atingerea acestui scop foarte dezirabil pentru obținerea de rezultate solide.

Operatorul pentru mutație cel mai frecvent implementat în algoritmii genetice este mutația punctuală. În această abordare, un locus este ales în mod aleatoriu. La acel locus, alela prezentă este înlocuită cu alternativa. Probabilitatea ca o mutație să apară este specificată la inițializarea algoritmului.

3.1.4. Selecția

Selecția poate avea loc la nivelul părinților și determină indivizii care vor fi utilizați pentru împerecheri. La nivelul generației, selecția este utilizată pentru a determina configurația populației din următoarea iterație.

Principiul evoluției prin selecție își are, de asemenea, sorgintea într-o lege naturală [69]. Indivizii care se adaptează mai bine la mediul înconjurător, au șanse sporite de supraviețuire și în consecință, au șanse sporite de-a își transmite informația genetică. Astfel, genele lor vor fi mai bine reprezentate în generațiile ulterioare. Diacronic această operațiune a fost abordată în diferite modalități.

3.1.4.1. Metoda turnirului

O metodă populară de selecție este metoda turnirului. În această abordare, un număr de n indivizi sunt aleși aleatoriu din populația curentă și supuși unui turnir. Dintre câștigători turnirului sunt selectați indivizii cei mai adaptați, cu fitness superior. Alegerea unui număr de k câștigători se poate face stocastic, alocând indivizilor ordonați după performanță, probabilități diferite de-a fi selecționați după formula:

$$p(1-p)^{i-1} \tag{3.1}$$

unde $i \in \{1,2,...,n\}$ este poziția în clasament obținută de un individ. Câștigătorii turnirului pot fi selecționați și deterministic, alegând cei mai performanți reprezentați în ordine descrescătoare a performanței lor.

3.1.4.1. Metoda ruletei

O altă alterativă de selecție adesea utilizată este metoda ruletei. În această abordare, fiecărui individ din populație îi este alocată o probabilitate de-a fi ales, în funcție de performanța sa în generația curentă, după formula:

$$p_i = \frac{f_i}{\sum_{t=1}^{n} f_t} \tag{3.2}$$

unde f_i este fitness-ul individului respectiv.

Metoda ruletei oferă indivizilor mai puțin adaptați șansa de-a fi selectați, proprietate dezirabilă pentru conservarea diversității.

3.1.4.1. Elitism

Elitismul [70] reprezintă o altă alternativă populară în implementarea selecției. Un număr prestabilit dintre indivizii cel mai bine adaptați sunt copiați intact în generația următoare. Elitismul protejează cei mai adaptați indivizi dintr-o populație de efectele recombinărilor și mutațiilor.

3.2. Metodă propusă pentru selectarea unui număr restrâns de atribute

Finalitatea algoritmului genetic implementat o reprezintă selecția atributelor din datele de ADN microarray. Datele într-un astfel de experiment constau într-un număr limitat de exemple și disproporționat de multe atribute. Din acest motiv, studiile de microarray impun, în general, utilizarea validării încrucișate. Studii anterioare au evaluat utilitatea algoritmilor genetici haploizi în selectarea atributelor pentru cercetările cu ADN microarray [71]. Numeroși autori [72-76] au concluzionat că AG diploizi se comportă mai bine în contexte dinamice în comparație cu implementările haploide. Am decis să abordăm finalitatea noastră cu o implementare diploidă și pentru a implementa și beneficia de oportunitățile pe care un cadru diploid le oferă în termeni de posibilități de recombinare pentru susținerea explorării în timpul căutării. Ne-am aplecat asupra tehnicilor de recombinare și mutație, datorită tendinței observate în utilizarea AG haploid de-a converge înspre un optim local și incapacitatea de-a părăsi o astfel de inflexiune. În plus, edificiul diploid oferă posibilitatea abordării superioare a selecției pentru reproducerea în generația următoare.

Scopul unui astfel de experiment nu este găsirea unui clasificator supervizat care poate discrimina perfect între două clase de exemple. Ne propunem să determinăm, dintr-un mare număr de gene diferențial exprimate, un subgrup care poate caracteriza și determina cauzal cele două clase de exemple. Relația cauzală între un subgrup de gene determinat prin metodele inteligenței artificiale nu poate fi stabilită în mod direct. Validarea biologică ulterioară a rezultatelor obținute prin metodele IA este obligatorie și depășește scopul acestei lucrări de doctorat. Studiile de microarray sunt realizate în echipe multidisciplinare tocmai datorită acestor exigențe.

Determinarea unui marcher genetic a fost diacronic una dintre finalitățile prestabilite într-un studiu microarray. Utilizarea unor marcheri specifici pentru detecția precoce a unor patologii sau prezicerea riscului la care un individ este supus în a se confrunta cu o anumită condiție este foarte dezirabilă. Cu toate acestea, relațiile complexe dintre genele diferențial exprimate în cadrul unei patologii este un domeniu foarte important, care fundamentează înțelegerea intimă a unei patologii și poate determina decisiv strategia terapeutică. Cunoștințele actuale în genetică permit evaluarea unor grupuri de gene implicate în diferite căi genetice cu implicații în patologie, iar aceste informații trebuie evaluate și pentru determinarea unor marcheri specifici. Prin urmare, selectarea unui număr restrâns de atribute, interpretabile din punct de vedere biologic este finalitatea noastră.

Arhitectura AG propus este fundamentată pe reprezentarea atributelor în genotip. Fiecare probă din setul de date microarray este reprezentată de o genă în genotip. Prin urmare, numărul atributelor din setul de date este egal cu numărul genelor din genotip. Fiecare locus poate fi ocupat de o alelă *1* sau *0*. *Alelele 1* codifică pentru prezența atributului de la acel locus în clasificarea supervizată. *Alelele 0* semnifică ignorarea acelui atribut la discriminarea dintre cele două clase de exemple. Un genotip are aspectul unui șir de valori *0* și *1*, iar atributele din setul de date corespunzătoare fiecărui locus codificat *1* în genotip, sunt utilizate în clasificare.

Am ales să utilizăm clasificatori supervizați pentru a evalua adaptabilitatea la mediu a genotipurilor testate. Acuratețea acestor clasificatori în discriminarea între clasele prezente în date a fost utilizată pentru evaluarea numerică a adaptabilității.

Plecând de la principiul No Free Lunch am încercat să lăsăm deschisă posibilitatea utilizării cât mai multor variante de clasificatori. Pachetul MLInterfaces [77] în Bioconductor acordă această șansă prin abordarea unitară a acestei probleme. În implementarea noastră, perfect compatibilă cu MLInterfaces, acuratețea obținută cu orice algoritm de clasificare supervizată implementat în acest pachet R, poate fi utilizată în algoritmul genetic propus.

Prin urmare, un genotip, șir de valori *0* și *1* cu lungimea egală cu numărul atributelor din setul de date codifică pentru un clasificator supervizat, a priori stabilit, angajat în a învăța exemplele pe seama sub-grupului de atribute specificat prin alelele cu valoarea=*1*. Fiecare individ dispune de două seturi haploide de cromozomi, în consecință, de doi clasificatori care utilizează aceeași tehnică de învățare, dar considerând subgrupuri diferite de atribute pentru discriminarea între exemple.

Populația inițială este constituită dintr-un număr prestabilit de indivizi, cu reprezentare diploidă, așadar de un număr dublu de genotipuri. Generarea genotipurilor este realizată aleatoriu, după o distribuție uniformă discretă. Indivizii din populația inițială sunt generați prin împerecherea fortuită a genotipurilor. Numărul indivizilor din populația inițială și al genelor active în fiecare genotip sunt stabilite prin parametrii precizați de utilizator la inițializarea algoritmului.

Un aspect extrem de important în proiectarea unui algoritm genetic diploid îl reprezintă maparea genotipului la fenotip, semnificativ diferită față de implementările haploide. Prezența a două alele corespunzător fiecărui locus, în fiecare dintre cele două seturi de cromozomi, necesită o abordare specială. În general, această problemă a fost adresată prin definirea unor scheme de dominare, individualizate pentru un cadru specific de optimizare. În această teză de doctorat propunem o abordare originală, inspirată din evoluția naturală, pentru maparea genotipului la fenotip în algoritmii diploizi. Modelată după principiul dominanței incomplete în genetică, propunerea noastră nu necesită definirea unei scheme de dominare și avantajează explorarea în AG. Metoda noastră este discutată pe larg în subcapitolele 3.2.1 și 3.2.2.

Următoarea etapă în AG propus o reprezintă condensarea genotipurilor în cromozomi. Utilizatorul poate specifica la inițializarea căutării numărul de cromozomi din genotip. Orice valoare cuprinsă în intervalul *[1, 22]* este acceptată. Valoarea *1* pentru acest parametru determină tratarea genotipului ca un singur cromozom, în timp ce valoarea *22* provoacă repartizarea atributelor din setul de date pe *22* de cromozomi de dimensiuni inegale. Distribuirea genelor după numărul de cromozomi solicitat nu se realizează echilibrat. Am ales să utilizăm repartizarea genelor umane pe cei *22* autozomi, prezentată în tabelul 3.1, ca model în acest scop.

În situația în care o altă valoare în intervalul *[1, 22]* este aleasă, repartizarea atributelor se realizează inegal, respectând dimensiunile relative ale autozomilor. Distribuirea atributelor într-un număr de cromozomi s-a făcut cu două scopuri. Pe de o parte, această abordare permite conceperea și implementarea unor noi operatori pentru recombinări și mutații, care modelează fidel fenomenele corespunzătoare în biologie și sunt potențial utili în ameliorarea explorării AG. Pe de altă parte, evoluția tehnologiei ADN microarray înspre variante adaptabile de către utilizator în termeni de probe imobilizate pe chip, permite o abordare superioară în selectarea atributelor. În general, într-un studiu ADN microarray, numărul mare al genelor fixate pe un chip comercial, conceput pentru o gamă largă de aplicații, face imposibilă utilizarea ordinii genelor pe chip în decizia configurației cromozomilor din algoritmul genetic. Noile variante de biochip-uri adaptabile, deschid această oportunitate. Gruparea unor gene cu roluri similare, cunoscute, în diferite căi deja

descrise pe aceiași cromozomi ar reprezenta o abordare foarte dezirabilă, cu efecte potențial remarcabile. Din păcate, nu am avut acces la această tehnologie pentru a studia implicațiile unui astfel de cadru. Testarea algoritmului propus în teza de doctorat în acest context reprezintă o direcție de cercetare pentru viitor.

În etapa următoare, populația inițială este evaluată prin prisma performanței clasificatorilor supervizați aleși, în discriminarea claselor de exemple din setul de date. Media celor două valori pentru acuratețe, corespunzător subgrupurilor considerate în fiecare dintre cele două seturi haploide de cromozomi caracterizează adaptabilitatea la mediu a individului respectiv. Indivizii astfel stratificați, după performanță, sunt ulterior supuși operațiunilor de recombinare, aplicate între cele două seturi haploide de cromozomi ai fiecăruia. Recombinările asigură evoluția spre generații cu indivizi mai bine adaptați, în acord ipoteza Building Block [66].

Recombinările sunt realizate cu un operator original, care modelează fenomene din meioză ce stau la baza evoluției în natură. Operatorul de recombinare, implementează atribuirea aleatorie a cromozomilor, a priori afectați de recombinări în două puncte, genotipurilor generate în această etapă. O prezentare mai detaliată a operatorului de recombinare propus în această teză de doctorat este cuprinsă în subcapitolele 3.4.1. și 3.4.2.

Tabel 3.1 - Repartizarea atributelor pe cromozomi

Cromozom Nr.	% dintre atribute
1	9.17%
2	7.64%
3	5.81%
4	4.89%
5	5.19%
6	5.81%
7	5.50%
8	4.28%
9	4.28%
10	4.28%
11	6.11%
12	4.89%
13	2.44%
14	3.66%
15	3.66%
16	3.97%
17	4.89%
18	1.83%
19	5.19%
20	2.75%
21	1.22%
22	2.44%

Metoda de selecție a seturilor de cromozomi din generației următoare exploatează elitismul, adesea implementat în algoritmii genetici. Pe de o parte seturile de cromozomi care au codificat clasificatorul mai puțin performant în fiecare individ sunt eliminate. O elită din seturile haploide de cromozomi care au fost mai adaptate în fiecare individ la evaluarea generației curente, este păstrată pentru iterația consecutivă. Proporția genotipurilor elitiste, conservate în generația următoare, este aleasă de utilizator, la inițializarea căutării. Din genotipurile obținute prin recombinări sunt eliminate fortuit un număr de instanțe egal cu valoarea stabilită pentru elitism. Acest pas este implementat pentru a perpetua o populație de dimensiune constantă pe parcursul căutării.

Seturile haploide de cromozomi selectate pentru a face parte din generația viitoare sunt supuse alterării prin mutație, cu o incidență specificată la inițializare. Pe parcursul cercetărilor noastre, am constatat că mutația clasică, este insuficientă pentru a susține capacitatea AG de-a părăsi un optim local. Incidențe scăzute ale mutației clasice nu avantajează acest comportat foarte dezirabil, iar incidențe sporite afectează semnificativ exploatarea. Așadar, am explorat posibilitatea altor operatori pentru mutații. Evoluția naturală a constituit sursa de inspirație pentru alte implementări, discutate separat în subcapitole dedicate în continuare.

Generația următoare este creată consecutiv, prin asamblarea indivizilor din seturile haploide de cromozomi împerecheate fortuit. Această nouă generație este ulterior evaluată și analizată pe parcursul unei noi iterații. Aceste etape se execută repetat pe parcursul unui număr de iterații specificat la inițializarea căutării, condiția de terminare a algoritmului.

Este de așteptat ca frecvența genelor utile în procesul de învățare să crească în populațiile din generațiile avansate. Această ipoteză își are sorgintea într-un principiu enunțat de J. Baldwin [78] și justifică abordarea selecției atributelor cu ajutorul algoritmilor genetici. Un număr de replicații al selectării atributelor cu algoritmul genetic este necesar pentru a adresa componenta stocastică a căutării și a obține rezultate semnificative. Interpretarea atributelor selectate cel mai frecvent, trebuie realizată cumulativ din rezultatele obținute în fiecare din repetițiile experimentului.

3.3. Dominanța incompletă

3.3.1. Dominanța incompletă în biologie

Fiecare celulă din organismul uman, cu excepția gameților, are la dispoziție, în nucleu, două copii ale fiecărui autozom. Autozomi sunt toți cromozomii, mai puțin X și Y. O copie a fiecărui autozom provine de la mamă, iar cealaltă este moștenită de la tată. Celulele somatice, au la dispoziție două copii ale fiecărui autozom și sunt prin urmare, diploide. Gameții conțin o singură copie a fiecărui autozom și sunt indicați drept haploizi. Cele două copii ale fiecărui cromozom, poartă numele de omologi. Fiecare dintre cromozomii omologi este moștenit de la unul dintre părinți și are, la aceiași loci, gene pentru aceleași tratamente. În consecință, celulele somatice au, în nucleu, două copii pentru fiecare genă, la același locus, în cromozomi omologi. Genele prezente la același locus în cromozomi omologi poartă

numele de alele. Alele identice sunt prezente în cromozomi omologi homozigoți pentru respectivul locus. Cromozomii omologi care au alele diferite sunt numiți heterozigoți pentru locus-ul specificat.

Fiecare genă codifică pentru un ARN specific. O bună parte dintre gene codifică pentru un ARN mesager specific și prin urmare, determină producția unei anumite proteine. Proteinele sunt responsabile de funcții variate, cu efecte detectabile în fenotip.

În cazul organismelor diploide, se pune problema modului în care alele diferite, prezente în cromozomi omologi heterozigoți, își găsesc exprimarea în fenotip. Mai multe modele au fost descrise în biologie pentru a explica acest fenomen. Primul și cel mai celebru model este fundamentat [79] de Gregor Johann Mendel. Încă din 1865, Mendel a descris un model în care una dintre cele două alele se exprimă în fenotip, iar cealaltă nu. El a descris relația dintre cele două alele în termenii: caracter dominant, pentru alela care se exprimă în fenotip și caracter recesiv pentru alela a cărei prezență nu produce efecte vizibile în fenotip. Respectând nomenclatura introdusă de Mendel, această relație poartă numele de dominanță.

Progresele din genetica modernă permit astăzi înțelegerea mai detaliată a relațiilor dintre alele. În prezent sunt descrise mai multe modele pentru diverse tipuri interacțiuni între alele. Un mod sugestiv de-a prezenta astfel de modele, foarte popular în genetică, este tabelul Punnet. Îl vom utiliza și noi pentru a ilustra principiile considerate. În tabelul de mai jos, prezentăm un caz fictiv, în care un organism moștenește gene care-i determină culoarea, de la generația anterioară. Există două alele posibile **R** și **a** pentru gena care determină culoarea organismului. **R** este alela dominantă și se exprimă în fenotip prin culoarea roșie. În notația din genetică alela dominantă este reprezentată de o literă majusculă, iar cea recesivă este notată cu litere minuscule. Alela **a** determină culoarea albastră în fenotip. Tabelul 3.2 prezintă combinațiile posibile și efectele în fenotip. Dacă organismul moștenește atât pe linie paternă cât și maternă alele **R**, el va avea culoarea roșie. Dacă ambele alele moștenite sunt **a**, el va fi albastru. Iar în cazul în care va moșteni o alelă **R** și una **a**, culoarea lui va fi roșie, deoarece alela **a** este dominată de **R** și nu se exprimă în fenotip.

O alternativă la acest model, îl reprezintă dominanța incompletă. În acest tip de interacțiune între alele, adesea întâlnită în natură, nici o alelă nu domină asupra celeilalte și expresia în fenotip a nici uneia nu este suprimată. Fenotipul heterozigot va fi intermediar între variantele de homozigote. Principiul dominanței incomplete este ilustrat în Tabelul 3.3, pentru același exemplu de organism fictiv de mai sus. În acest caz, ambele alele au fost notate cu litere majuscule, deoarece nici una nu este dominată. Se observă în tabel că fenotipul heterozigot RA va avea culoarea violet, o combinație a efectelor celor două alele.

Tabel 3.2 – Dominanță completă

		Moștenire de la TATĂ	
		R	**a**
Moștenire de la MAMĂ	**R**	**RR**	**Ra**
	a	**Ra**	**aa**

Tabel 3.3 – Dominanță incompletă

		Moștenire de la TATĂ	
		R	**A**
Moștenire de la MAMĂ	**R**	**RR**	**RA**
	A	**RA**	**AA**

Înțelegerea modelului dominanței incomplete a adus beneficii majore în genetică și are implicații clinice în medicina modernă. G. Tortora [80] ilustrează principiul dominanței incomplete cu exemplul siclemiei. Siclemia este un tip de anemie în care capacitatea eritrocitelor de-a transporta oxigen este afectată de calitatea hemoglobinei. Mai mult, forma eritrocitelor este afectată și ia aspect de seceră, motiv pentru care boala poartă și numele de anemie cu celule în seceră. Alela responsabilă pentru sintetizarea hemoglobinei defecte este **HbS** și este cauza siclemiei. Alela care se exprimă în hemoglobina pură este **HbA**. Indivizii sănătoși sunt homozigoți **HbAHbA**. Cazurile homozigot **HbSHbS** sunt pacienți suferinzi de siclemie. Ei prezintă un tablou clinic sugestiv pentru genotipul moștenit. Situația heterozigot HbAHbS reprezintă purtătorul afecțiunii. Acest pacient se prezintă cu anemie ușoară, deoarece doar jumătate din hemoglobina lor este funcțională, dar tabloul clinic nu este sugestiv pentru siclemie, ci intermediar între homozigot **HbAHbA** și homozigot **HbSHbS**. Totuși, acești pacienți, deși nu suferă foarte mult datorită prezenței **HbS** o pot transmite urmașilor.

Alte modele alternative de interacțiuni între alele cum ar fi co-dominanța sau moștenirea complexă, care consideră și influența mediului asupra fenotipului, sunt descrise, dar nu fac obiectul acestei teze de doctorat.

3.3.2. Dominanța incompletă în algoritmii genetici

Implementarea unui algoritm genetic diploid (AGD), deși fundamentată pe aceleași principii ca și varianta haploidă, este semnificativ mai complexă și se confruntă cu imperative suplimentare. În proiectarea unui AGD maparea genotipului la fenotip este complicată, comparativ cu algoritmii genetici haploizi. Un algoritm genetic evaluează adaptabilitatea la mediul înconjurător pe seama fenotipului, iar codificarea acestuia în genotip este mai elaborată în implementările diploide. Dubla prezență a alelelor la fiecare locus, în fiecare dintre cele două seturi de cromozomi prezente într-un individ, necesită tratament suplimentar.

Diacronic, au fost propuse diferite soluții pentru maparea genotipului la fenotip în contextul algoritmilor genetici diploizi. Diferiți autori, au propus și experimentat o varietate largă de scheme de dominare, special concepute pentru a răspunde cerințelor unor experimente specifice. O schemă de dominanță, imaginată de Ng și Wong [73] utilizează un model cu patru alele, două dominante și două recesive. Propunerea lor este ilustrată în tabelul 3.4. Alelele dominante (notate **0** și **1**) sunt notate cu majuscule, iar cele recesive (notate **i** și **o**) cu minuscule, respectând tradiția din genetică. Alelele recesive sunt mascate în fenotip de prezența celor dominante, iar alelele dominante au șanse egale de-a se exprimă în fenotip.

Tabel 3.4 – Schema de dominanță Ng-Wong

	0	**o**	**1**	**i**
0	0	0	0/1	0
o	0	0	1	0/1
1	0/1	1	1	1
i	0	0/1	1	1

O serie de propuneri au abordat acest aspect prin operații numerice. Ryan [74] a propus un model în care fiecare alelă are atribuită o valoare numerică. Exprimarea în fenotip depinde, în acest caz, de rezultatul însumării valorilor respective în comparație cu o valoare de tăiere prestabilită. O abordare probabilistică a fost propusă de Yang [81]. El a utilizat un vector de probabilități pentru a determina alelele exprimate în fenotip. Uyar și Harmanaci [82] au utilizat conceptul de penetranță din genetică pentru a dezvolta schema lor de dominare. De asemenea, variante de scheme de dominare capabile a se adapta pe parcursul căutării au fost analizate în [83] și au fost determinate proprietăți dezirabile ale acestora în contexte dinamice.

Dominanța incompletă promovează ideea unui fenotip intermediar între variantele homozigote. Într-un mod foarte simplificat, să presupunem că o genă este responsabilă de producerea unui ARN mesager specific și în consecință de sinteza unei anumite proteine. Alela defectă determină sinteza unei proteine nefuncționale. Prezența alelei imperfecte într-unul dintre cele două seturi de cromozomi ale unui organism diploid, poate avea drept consecință sinteza a jumătate din cantitatea necesară din acea proteină și apariția unui fenotip intermediar între normal și situația în care acea proteină ar lipsi în totalitate. O situație interesantă ar fi dacă proteina defectă produce efecte dezavantajoase, sau nesemnificative, pentru adaptarea la mediu într-un anumit context. Totuși, această proteină s-ar putea dovedi avantajoasă într-un context diferit și un individ astfel echipat ar fi mai flexibil la condițiile externe.

Într-un algoritm genetic diploid, putem aplica acest concept după cum urmează. Fiecare set de cromozomi prezent în individ, poate fi considerat cu efectele sale particulare. Un individ reprezentat prin două seturi de cromozomi, are la dispoziție doi clasificatori supervizați, diferiți în privința atributelor considerate, pentru a se adapta în același context. Adaptabilitatea unui astfel de individ este apreciată prin prisma mediei performanțelor în acomodarea la mediul înconjurător al celor doi clasificatori. Într-un scenariu cu validare încrucișată, foarte dezirabil în analiza datelor microarray, unde numărul exemplelor este net inferior sumedeniei atributelor, această abordare poate fi avantajoasă. În această situație, procesul de selecție al indivizilor pentru generația următoare avantajează explorarea față de situația în care căutarea s-ar efectua cu un algoritm genetic haploid. În plus, aspectul dinamic al mediului înconjurător este abordat mai flexibil în implementarea diploidă, cu beneficii în privința adresării riscului de adaptare excesivă.

În algoritmul propus în teza de doctorat, pentru analiza datelor de ADN microarray, performanța adaptării la mediul înconjurător este evaluată pe seama acurateței în discriminarea între două clase de exemple cu ajutorului unui tip de clasificator supervizat, la alegere. Fiecare clasificator utilizează un sub-grup dintre atributele prezente în setul de date. Prin urmare, un individ va fi evaluat prin prisma valorii medii a acurateților celor doi clasificatori supervizați codificați în cele două seturi intrinseci de cromozomi.

3.4. Atribuirea aleatorie a cromozomilor

3.4.1. Atribuirea aleatorie a cromozomilor în meioză

Procesul de diviziune celulară fundamentează finalitatea supraviețuirii individului și în consecință al speciei. Celulele dintr-un organism trebuie să se dividă pentru atingerea unor obiective diferite. Celule somatice trebuie să se dividă pentru a regenera diferite țesuturi și a susține funcții variate în organism. Celule speciale, numite germinative, se divid pentru a crea gameți, proces care stă la baza reproducerii organismului și, în consecință, a propășirii speciei.

Diviziunea celulelor somatice persistă pe parcursul vieții fiecărui organism, cu caracteristici deosebite pentru diferitele tipuri de țesuturi. În general, celule somatice diferențiate pentru realizarea optimă a funcții bine stabilite, în țesuturi specifice, se divid cu finalitatea de-a produce celule noi, identice, capabile să susțină aceleași funcții. În acest context, bagajul genetic al celulei care urmează a se divide, trebuie păstrat în integralitatea lui, iar modificările la nivelul ADN nu sunt dezirabile. Acest tip de diviziune celulară poartă numele de mitoză. Multiple mecanisme de control sunt destinate supervizării modului în care informația genetică este transmisă, cu scopul păstrării integrității acesteia în generația următoare. Aceste mecanisme de control pot împiedica diviziunea celulelor cu ADN alterat. Eșecul acestor mecanisme în a preveni diviziunea celulelor cu ADN degradat poate rezulta în patologii care amenință supraviețuirea subiectului.

Pe de altă parte, celulele germinative se divid într-o manieră fundamental diferită. Acest proces, numit meioză, are ca scop reproducerea individului, iar modul de desfășurare servește finalitatea. Rezultatul meiozei, celule numite gameți, conțin în nucleu un singur set de cromozomi. Reducerea numărului cromozomilor din celulele germinative diploide, cu formarea de gameți haploizi este fundamentală și obligatorie în pregătirea fertilizării pentru reproducerea organismului. În timp ce mecanisme de control protejează transmiterea informației genetice prin meioză, un grad de diversitate genetică este permis. Scopul toleranței pentru un anumit grad de variabilitate este fără îndoială, evoluția generației următoare în sensul adaptării superioare la mediul înconjurător. Fără a intra în toată complexitatea modului în care meioza se desfășoară, este important de subliniat maniera în care diversitatea genetică este promovată în timpul diviziunii celulare. Meioza se desfășoară în două etape cu caracteristici diferite, numite meioză I și meioză II. Pe parcursul meiozei I, consecutiv replicării ADN-ului, cromozomii omologi, cu structură dublă, formează sinapse. Cromozomii omologi se asamblează în structuri numite tetrade. În această configurație, are loc schimbul de informație genetică între cromozomii omologi. Această comunicare, prin intermediul unor fragmente de ADN, este permisă la nivelul unor poziții specifice de contact, numite chaisme. Fenomenele care au loc în timpul diviziunilor celulare sunt discutate detaliat în [84]. Acest proces a fost pe larg utilizat în algoritmii genetici și s-a concretizat în diferiți operatori de recombinare (crossover în acord cu nomenclatura din genetică).

În biologie, meioza I este etapizată în patru faze:

1) Profază I
- materialul genetic se organizează și condensează,
- se produc sinapse între cromozomii omologi, cu apariția tetradelor,
- în interiorul tetradelor, la nivelul chiasmelor, are loc schimbul de informație genetică (crossing-over)
- membrana nucleară dispare,
- începe constituirea fusului meiotic,
- centrozomii migrează la poluri opuse ale fusului
- microtubule se atașează la nivelul kinetochor-ilor.

2) Metafază I
- tetradele se aliniază echidistant față de cele două poluri ale fusului meiotic prin acțiunea microtubulelor,
- se formează placa metafazică,
- orientarea tetradelor fața de cei doi poli este fortuită,

3) Anafază I
- cromozomii omologi sunt separați și atrași spre poli opuși prin acțiunea microtubulelor,

4) Telofază I
- o nouă membrană nucleară se configurează la fiecare pol pentru a circumscrie fiecare set de cromozomi.

Meioza I este urmată de meioza II, un proces similar mitozei, cu finalitatea producerii a câte două celule haploide din fiecare celulă rezultată după prima etapă. Așadar, patru celule haploide, cu material genetic recombinat, rezultă în urma fiecărei diviziuni meiotice a unei celule diploide. Ilustrare sugestivă a meiozei este prezentată în Fig. 3.5.

Fenomenul de recombinare are loc în profaza I și este o sursă importantă de diversitate genetică, modelat în forme variate în diferite implementări de algoritmi genetici. O altă sursă de variabilitate genetică o reprezintă atribuirea aleatorie a cromozomilor în meioză. Alinierea, fortuită în privința sensului, a tetradelor în raport cu cei doi poli are loc în timpul metafazei I și se concretizează consecutiv, în atragerea independentă și întâmplătoare a cromozomilor cu structură dublă spre unul dintre poli, prin acțiunea microtubulelor, pe parcursul anafazei I. Această sursă de diversitate genetică este extrem de importantă, iar cuantumul informației genetice comunicate în timpul acestui proces merită exploatat.

Importanța recombinărilor și atribuirii aleatorii a cromozomilor pentru evoluția naturală este subliniată prin antiteză cu absența lor din mitoză. Meioza I cu reducerea numărului de cromozomi și procesele care asigură diversitatea genetică au ca finalitate evoluția generației viitoare. De asemenea, mecanismele de control al integrității informației genetice permit evoluția, în contrast cu rigiditatea specifică mitozei.

Un alt aspect care trebuie menționat este condensarea materialului genetic în timpul profazei special pentru a asigura conservarea informației genetice în timpul diviziunilor celulare. Această desfășurare, coroborată cu fenomenul producerii chiasmelor, subliniază importanța configurației spațiale atât pentru garantarea integrității moștenirii ADN, cât și pentru asigurarea evoluției naturale.

3.4.2. Atribuirea aleatorie a cromozomilor în AG

Algoritmii genetici utilizează în mod tradițional operatori de recombinare pentru a asigura explorarea și susține evoluția. Operatorii de recombinare s-au bucurat de o atenție sporită și diacronic, au fost propuse numeroase variante și versiuni, unele cu utilitate foarte specifică, pentru anumite contexte de căutări sau optimizări, alte cu aplicabilitate generală. Cele mai populare propuneri au fost recombinările într-unul sau două puncte. Cei doi operatori sunt ilustrați în Fig. 3.6 și Fig. 3.7. Al doilea fenomen care asigură diversitatea genetică, atribuirea aleatorie a cromozomilor în timpul meiozei nu a fost exploatat corespunzător în AG.

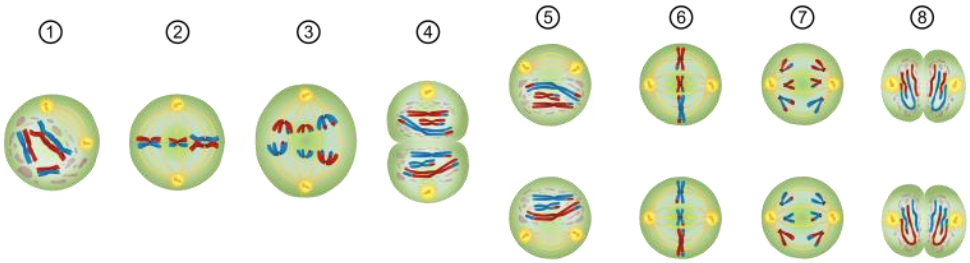


Fig. 3.5 – Etapele meiozei (sursa imaginii: wikipedia.com, cu drepturi libere de utilizare și modificare.)

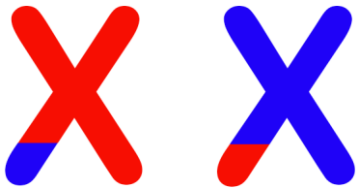


Fig. 3.6 – Recombinarea într-un punct.

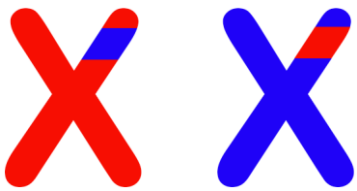


Fig. 3.7 – Recombinarea în două puncte.

Teza de doctorat de față propune un algoritm care beneficiază de modelarea atribuirii aleatorii a cromozomilor în timpul meiozei. De asemenea, testarea și evaluarea efectelor acestui model în analiza datelor cu ADN microarray este dezvoltată în ultimele capitole ale tezei de doctorat.

Am pornit de la premisa că atribuirea fortuită a cromozomilor în meioză contribuie foarte semnificativ la diversitatea genetică și în consecință, la evoluție. În consecință, ne așteptăm ca modelarea acestui fenomen natural să avantajeze explorarea în AG.

În abordarea noastră, genotipul este a priori configurat într-un număr variabil de cromozomi. Tratarea genotipului ca seturi de cromozomi este o condiție obligatorie în vederea modelării acestui fenomen. În abordarea cu un singur cromozom, efectul ar fi identic cu cel al recombinărilor în două puncte.

Impactul utilizării modelului atribuirii aleatorii a cromozomilor în AG cu un număr variabil de cromozomi este ilustrat în Fig. 3.8. Tratăm un genotip cu informația genetică distribuită pe trei cromozomi pentru claritate. Odată cu utilizarea unui număr sporit de cromozomi, cresc și efectele operatorului propus de noi. Figura reprezintă și recompensele în comparație cu operatorii clasici de recombinare într-unul sau două puncte. Primul nivel surprinde o celulă diploidă cu materialul genetic organizat pe trei cromozomi, a priori replicați. Cromozomii moșteniți pe linie maternă și paternă sunt ilustrați cu culori diferite, roșu și respectiv albastru. Este ilustrat fenomenul de formare al sinapselor. Efectele recombinărilor într-unul sau două puncte sunt reprezentare pe nivelul al doilea, iar rezultatul obținut prin atribuirea aleatorie a cromozomilor este evident în nivelul inferior al figurii. Ultimul nivel al figurii prezintă impactul asupra materialului genetic pregătit pentru a fi transmis generației viitoare, corespunzător setului haploid de cromozomi rezultat în urma meiozei II din natură. Consecutiv meiozei II, patru astfel de seturi ar trebui figurate, dar ilustrația a fost simplificată pentru claritate. Este evidentă recombinarea superioară din punct de vedere al diversității genetice în abordarea cu atribuirea aleatorie a cromozomilor, figurată la nivelul inferior al figurii.

Recombinarea într-un punct are o valoare limitată în contextul selectării atributelor în datele de ADN microarray. Numărul imens de atribute și necesitatea selectării unui subset redus, într-un astfel de studiu, obligă la genotipuri foarte lungi, cu puține gene activate în fiecare set haploid de cromozomi. În consecință, efectele recombinării într-un punct, puternic afectate de locii genelor activate, vor fi indezirabile. Pe de o parte, recombinările într-un punct vor favoriza cele două capete ale șirului de gene. Pe de altă parte, schemele foarte lungi datorită configurației genotipului, sunt distruse și explorarea nu este servită corespunzător. Din acest motiv, nu am utilizat recombinări într-un punct în experimentele noastre.

Recombinările în două puncte au fost preferate deoarece adresează limitările variantei într-un punct. Pe de o parte, capetele cromozomilor nu mai sunt favorizate, iar schemele lungi sunt mai bine conservate. Riscul recombinărilor într-un punct îl reprezintă șansa ridicată ca o recombinare să nu producă efecte în cromozom. Acest risc apare datorită numărului mic de gene active în genotip, comparativ cu dimensiunea acestuia. Separarea pe un număr de cromozomi adresează această problemă, iar atribuirea aleatorie a cromozomilor servește în plus explorarea. Aceste realități motivează și decizia de-a nu implementa un parametru pentru stabilirea șansei ca o recombinare să se producă. Fiecare cromozom în algoritmul nostru, suferă o recombinare în pregătirea generației viitoare.

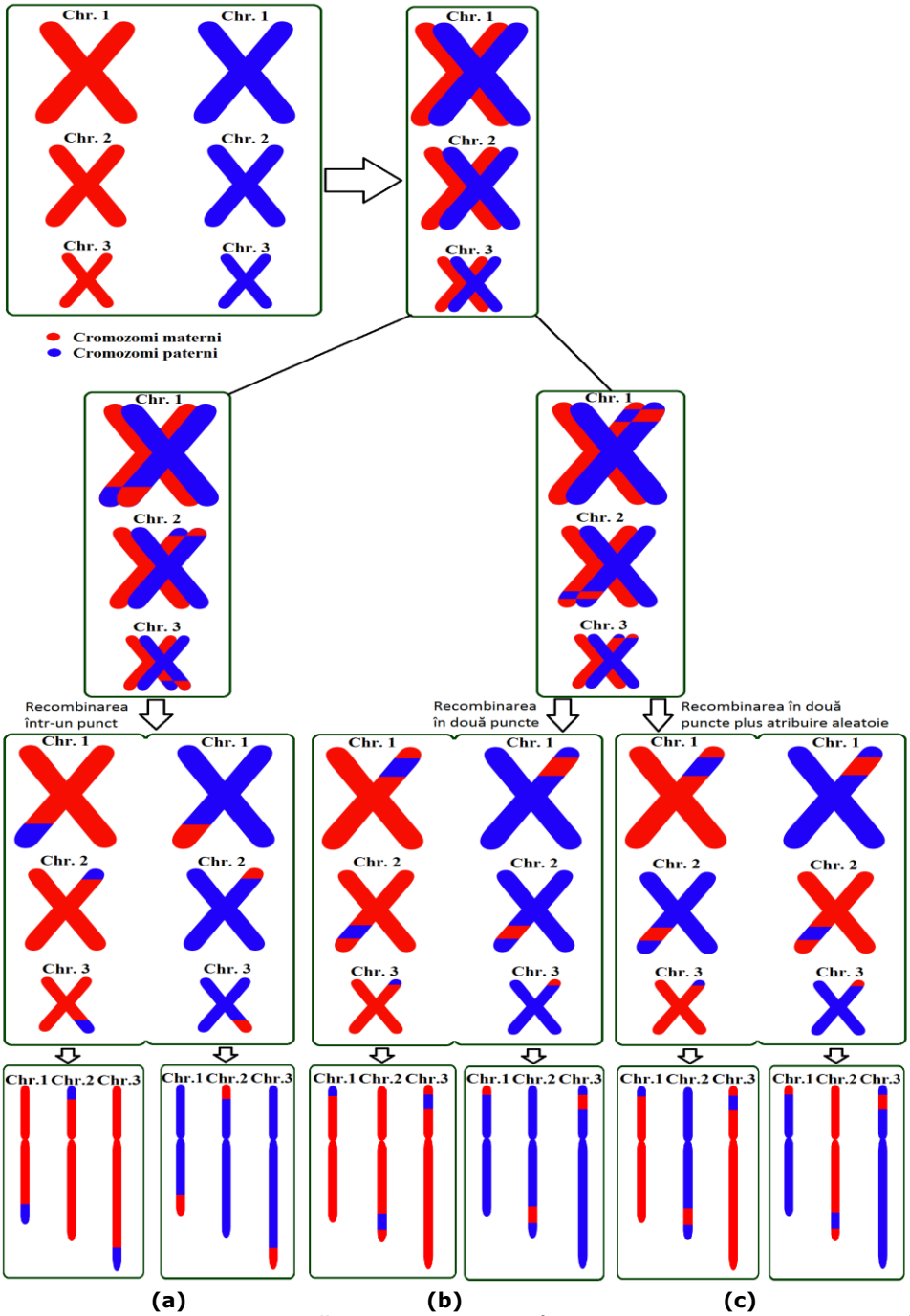


Fig. 3.8 - Diversitatea genetică (a)recombinarea într-un punct; (b)recombinarea în două puncte; (c)recombinarea în două puncte urmată de atribuirea aleatorie a cromozomilor.

Propunerile de operatori pentru recombinare care nu sunt afectați de ceea ce se numește în IA positional bias, cum ar fi recombinarea uniformă, nu sunt dezirabile în contextul aplicației noastre. Urmărim să beneficiem de efectele unui bias pozițional când selectăm sub-grupuri de atribute în datele microarray. Există influențe cunoscute care favorizează prezența unor grupuri de gene selectate din date microarray, iar bias-ul pozițional servește acest deziderat. În plus, fenomenele care asigură diversitatea genetică în natură sunt puternic influențate de bias pozițional, caracteristică păstrată într-o anumită măsură în modelul nostru.

O ilustrare a comparației între abordarea noastră și metode clasice, cu efectele în promovarea diversității genetice, așa cum este ea modelată în AG propus este ilustrată în Fig. 3.9. Prezentăm un genotip fictiv cu doar 10 gene pentru claritate. Evident, în cazul unor genotipuri lungi și unui număr sporit de cromozomi, aceste efecte sunt cu atât mai importante.

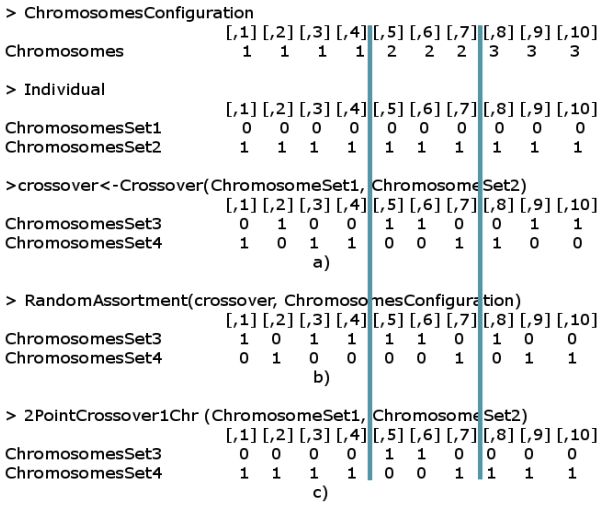


Fig. 3.9 - Diversitatea genetică (a) recombinarea în două puncte; (b) recombinarea în două puncte urmată de atribuirea aleatorie a cromozomilor; (c) recombinarea clasică în două puncte cu un singur cromozom.

3.5. Operatori pentru mutații

În algoritmii genetici, operatorii pentru mutații sunt concepuți pentru a susține explorarea și diversitatea genetică pe parcursul evoluției. Efectul mutațiilor asupra exploatării trebuie considerat în contextul fiecărui experiment și în general, șansa ca o mutație să aibă loc este net inferioară probabilității apariției unei recombinări. În timp ce recombinările urmăresc asocierea fragmentelor din genotip, cu scopul adaptării superioare la mediul înconjurător, mutațiile sunt mai importante în susținerea tendinței de-a părăsi un optim local în favoarea evoluției.

Pentru ca o schimbare în genotip să poată fi considerată mutație, ea trebuie să fie permanentă și transmisibilă generației viitoare. Din acest motiv, operatorii pentru mutații în algoritmii genetici sunt aplicați indivizilor care vor forma generația succesivă, ulterior recombinărilor.

Operatorii pentru mutație își au sorgintea în biologie. Mutații pot apărea la nivelul ADN-ului diferitor tipuri de celule în organismul uman, cu efecte foarte diferite. O mutație la nivelul materialului genetic într-o celulă somatică, poate determina o neoplazie. Într-o celulă a embrionului în dezvoltare, o mutație poate rezulta în malformații congenitale. Când materialul genetic al unei celule germinative este tulburat de o mutație, efectele ei se transmit generațiilor viitoare de indivizi, care moștenesc ADN-ul defect. Există numeroase modalități în care o mutație poate apărea și afecta evoluția naturală. Progresele în biologia moleculară și genetică permit înțelegerea aprofundată a unora dintre aceste mecanisme. În subcapitolele următoare modelăm tipurile de mutații care sunt interesante din punct de vedere al susținerii explorării în algoritmii genetici.

De departe, cel mai frecvent implementat operator pentru mutații în AG este mutația punctuală. În această variantă, o alelă din genotip este înlocuită cu alternativa, la un locus ales aleatoriu. Șansa ca o mutație să apară este stabilită la inițializarea algoritmului. În Fig. 3.9 sunt ilustrate efectele mutației punctuale. Pentru transparență, utilizăm o populație fictivă formată din patru indivizi și genotipul format din 8 gene, distribuite pe doi cromozomi, cu patru gene fiecare. În acest exemplu, șansa cu o mutație să apară este de 5%.

În selectarea atributelor din datele de ADN microarray, se lucrează cu mii de atribute și se urmărește selectarea unui sub-grup restrâns dintre acestea. Utilitatea mutației într-un punct în acest context este limitată. Cu toate că mutația într-un punct tulbură eficient seturile haploide de cromozomi din populație, fenotipul indivizilor nu este afectat așa cum am dori. Numărul genelor active într-un individ, sporește progresiv în generațiile evoluate. Capacitatea unui AG de-a părăsi un optim local și a continua evoluția este limitată. Utilizarea unei șanse prea mari de a apărea o mutație, are efect negativ asupra explorării.

Ne-am îndreptat așadar atenția spre biologie pentru a determina modele alternative de-a realiza mutațiile în AG. Există deosebiri fundamentale între codul genetic în natură și principiile utilizate în algoritmii genetici. Mecanismele de apariție ale mutațiilor în genetică nu pot fi modelate fidel în operatori pentru algoritmii genetici. Cu toate acestea, principii învățate din genetică pot fi utilizate pentru îmbunătățirea unor astfel de operatori. Foarte multe variante de operatori pentru mutații au fost propuși diacronic de diferiți autori, iar genetica a fost adesea izvorul de inspirație pentru acele metode. Nu considerăm că propunerile descrise în continuare sunt originale în totalitate sau că principiile utilizate nu au fost abordate anterior. Este însă foarte interesantă utilizarea acestor abordări în contextul algoritmului propus de noi pentru selectarea atributelor în datele de ADN microarray.

3.5.1. Mutația fără sens în biologie

În genetică, mutația fără sens (eng. nonsense mutation) este un caz special de perturbare a ADN. Tulburarea apare la nivelul unei singure nucleotide în ADN. Modificarea respectivă, prin transcripție, devine un codon stop în ARN. Consecutiv, translația este terminată prematur, iar proteina care ar fi codificată nu mai este sintetizată complet.

```
> individuals<-Mutations(individuals, 5)
> individualsOriginal
  Id 1005_at 1007_s_at 1008_f_at 1009_at 1020_s_at 1030_s_at 1038_s_at 1039_s_at
1  1       0         1         0       1         1         0         1         1
2  1       0         1         1       1         1         0         1         0
3  2       0         1         0       1         1         1         0         1
4  2       0         1         0       1         1         0         1         1
5  3       1         1         1       0         1         0         1         0
6  3       0         1         0       1         1         1         1         0
7  4       1         0         1       1         0         1         1         0
8  4       1         0         1       0         0         1         1         1
> individuals
  Id 1005_at 1007_s_at 1008_f_at 1009_at 1020_s_at 1030_s_at 1038_s_at 1039_s_at
1  1       0         1         0       1         1         0         1         1
2  1       0         1         1       1         1         0         1         0
3  2       0         0         0       1         1         1         0         1
4  2       0         1         0       1         1         0         1         1
5  3       1         1         1       0         1         0         1         0
6  3       0         1         0       1         1         1         0         0
7  4       1         0         1       1         0         1         1         0
8  4       1         0         1       1         0         1         1         1
```

Fig. 3.9 – Efectele mutației într-un punct.

3.5.2. Mutația fără sens în algoritmii genetici

În algoritmii genetici, mutația fără sens nu poate fi modelată fidel situației din biologie. Fenomenul rezultat în urma acestui tip de mutație poate fi utilizat însă în conceperea unui operator de mutație valoros în AG.

Operatorul pentru mutația fără sens anulează toate alelele prezente într-un cromozom, consecutiv unui locus selectat aleator. Această abordare utilizează distribuția genelor pe cromozomi în algoritmul propus de noi, doar genelor consecutive locus-ului selectat fortuit, pe un cromozom selectat întâmplător, le este atribuită valoarea 0. Genele prezente pe alți cromozomi ale aceluiași fenotip nu sunt afectate de mutație. Efectele operatorului propus sunt ilustrate în Fig. 3.10. Genele sunt distribuite pe doi cromozomi, așa cum este sugerat pe nivelul superior al figurii. Genele care suferă modificări în urma mutației sunt evidențiate. Frecvența de apariție a mutației este la alegerea utilizatorului.

```
> chrConf
[1] 1 1 1 1 2 2 2 2
> individualsOriginal
  Id 1005_at 1007_s_at 1008_f_at 1009_at 1020_s_at 1030_s_at 1038_s_at 1039_s_at
1  1       1         0         1       1         0         1         1         0
2  1       1         0         0       1         1         1         0         1
3  2       1         1         1       0         1         1         0         0
4  2       0         0         1       0         1         1         1         1
5  3       1         1         0       1         1         1         0         0
6  3       1         0         0       0         1         1         1         1
7  4       1         0         0       1         1         0         1         1
8  4       1         1         0       1         1         1         0         0
> individuals
  Id 1005_at 1007_s_at 1008_f_at 1009_at 1020_s_at 1030_s_at 1038_s_at 1039_s_at
1  1       1         0         1       1         0         1         1         0
2  1       1         0         0       1         0         0         0         0
3  2       1         1         1       0         1         1         0         0
4  2       0         0         1       0         1         1         1         1
5  3       1         1         0       1         1         1         0         0
6  3       1         0         0       0         1         1         1         1
7  4       1         0         0       0         1         0         1         1
8  4       1         1         0       1         1         0         0         0
```

Fig. 3.10 – Efectele mutației fără sens.

3.5.3. Mutația cu deplasare în biologie

Codul genetic este descris în biologie de proprietăți care determină modul în care mutațiile se întâmplă în natură. Pe de o parte, codul genetic nu se suprapune, pe de altă parte este continuu. Prin urmare, când o nucleotidă din șir este întâmplător eliminată sau adăugată, tot subșirul consecutiv este decodificat în mod eronat. Așadar, o schimbare apărută la nivelul unei singure nucleotide, poate produce efecte mai semnificative decât o variație a unui singur aminoacid într-o proteină. Aceste tipuri de perturbări la nivelul ADN sunt numite mutații cu deplasare (eng. frame shift mutation).

3.5.4. Mutația cu deplasare în algoritmii genetici

Operatorul propus pentru mutația cu deplasare utilizează principiul din genetică, dar nu modelează întocmai fenomenul biologic. Mutația alterează cromozomul în sensul deplasării la stânga a șirului începând cu o poziție generată aleatoriu. Ultima poziție de pe cromozom este completată ulterior cu alela 0. Un singur cromozom dintr-un set haploid este afectat de mutație. Șansa ca o mutație să se producă este specificată la inițializarea algoritmului. Cromozomii afectați și locus-urile interesate sunt alese la întâmplare. Efectele acestei mutații sunt ilustrate în Fig. 3.11.

3.5.5. Ștergerea unui segment în biologie

În timpul meiozei, au loc recombinări, schimburi de informație genetică între cromozomii omologi organizați în tetrade. Este posibil să apară erori în această etapă. Segmente din cromozomi pot fi șterse complet dintr-un cromozom și adăugate excesiv în omolog. Astfel, ambii omologi sunt anormali, iar fenotipul este afectat consecutiv.

3.5.6. Ștergerea unui segment în algoritmii genetici

În implementarea noastră pentru mutația cu ștergerea unui segment de cromozom, cu o probabilitate specificată la inițializare, cromozomii care suferă mutația sunt selectați întâmplător. Ulterior, pentru fiecare cromozom astfel ales, sunt generate aleatoriu marginile unui interval și toate alelele din acel interval sunt anulate. Un exemplu de mutație cu ștergerea unui segment este prezentat în Fig. 3.12.

3.5.7. Ștergerea unui cromozom în biologie

Erori pot apărea și la separarea și atribuirea cromozomilor recombinați în timpul meiozei. Pot rezulta astfel celule cu un număr incorect de cromozomi, mai mare sau mai mic. Rezultatul acestui tip de eroare are întotdeauna efecte dramatice asupra fenotipului și șanselor de supraviețuire ale individului.

3.5.8. Ștergerea unui cromozom în algoritmii genetici

Operatorul pentru mutație prin ștergerea unui cromozom (Fig. 3.13), anulează toate alelele de pe cromozomi aleși fortuit. Șansa ca o mutație cu ștergerea întregului cromozom să se producă este stabilită la lansarea algoritmului.

```
> chrConf
[1] 1 1 1 1 2 2 2 2
> individualsOriginal
  Id 1005_at 1007_s_at 1008_f_at 1009_at 1020_s_at 1030_s_at 1038_s_at 1039_s_at
1  1       0         1         1       0         1         1         0         1
2  1       1         1         0       1         0         1         1         0
3  2       1         1         1       1         0         0         0         1
4  2       1         1         1       0         1         0         1         0
5  3       1         1         1       0         0         0         1         1
6  3       1         1         0       1         1         0         1         0
7  4       1         0         0       1         1         0         1         1
8  4       0         1         1       1         1         0         0         1
> individuals
  Id 1005_at 1007_s_at 1008_f_at 1009_at 1020_s_at 1030_s_at 1038_s_at 1039_s_at
1  1       1         1         0       0         1         1         0         1
2  1       1         1         0       1         0         1         1         0
3  2       1         1         1       0         0         0         0         1
4  2       1         1         1       0         1         0         1         0
5  3       1         1         1       0         0         0         1         1
6  3       1         1         0       1         1         0         1         0
7  4       1         0         0       1         1         0         1         1
8  4       0         1         1       1         1         0         0         1
```

Fig. 3.11 – Efectele mutației cu deplasare.

```
> chrConf
[1] 1 1 1 1 2 2 2 2
> individualsOriginal
  Id 1005_at 1007_s_at 1008_f_at 1009_at 1020_s_at 1030_s_at 1038_s_at 1039_s_at
1  1       0         0         1       1         0         1         1         1
2  1       1         1         0       1         0         1         0         1
3  2       1         0         1       1         1         1         0         0
4  2       0         1         1       1         1         0         1         0
5  3       1         1         0       0         1         1         1         0
6  3       1         0         0       1         1         0         1         1
7  4       0         0         1       1         1         1         0         1
8  4       1         1         1       0         1         0         0         1
> individuals
  Id 1005_at 1007_s_at 1008_f_at 1009_at 1020_s_at 1030_s_at 1038_s_at 1039_s_at
1  1       0         0         1       1         0         1         1         1
2  1       1         1         0       1         0         1         0         1
3  2       1         0         1       1         0         0         0         0
4  2       0         1         0       0         1         0         1         0
5  3       1         1         0       0         1         1         1         0
6  3       1         0         0       1         1         0         0         1
7  4       0         0         1       1         1         1         0         1
8  4       1         1         1       0         1         0         0         1
```

Fig. 3.12 – Efectele ștergerii unui segment de cromozom.

```
> chrConf
[1] 1 1 1 1 2 2 2 2
> individualsOriginal
  Id 1005_at 1007_s_at 1008_f_at 1009_at 1020_s_at 1030_s_at 1038_s_at 1039_s_at
1  1       0         1         0       1         0         1         1         1
2  1       0         0         1       1         1         0         1         1
3  2       0         1         0       1         1         1         1         0
4  2       1         1         1       0         0         1         0         1
5  3       1         0         1       1         1         0         1         0
6  3       1         1         1       0         1         0         0         1
7  4       0         0         1       1         0         1         1         1
8  4       1         0         1       1         1         0         1         0
> individuals
  Id 1005_at 1007_s_at 1008_f_at 1009_at 1020_s_at 1030_s_at 1038_s_at 1039_s_at
1  1       0         1         0       1         0         1         1         1
2  1       0         0         1       1         0         0         0         0
3  2       0         1         0       1         1         1         1         0
4  2       1         1         1       0         0         1         0         1
5  3       1         0         1       1         1         0         1         0
6  3       1         1         1       0         1         0         0         1
7  4       0         0         1       1         0         1         1         1
8  4       1         0         1       1         1         0         1         0
```

Fig. 3.13 – Efectele ștergerii unui cromozom.

3.5.9. Transpozonii în biologie

Transpozonii sunt secvențe care își pot schimba poziția în lanțul ADN. Se consideră că existența transpozonilor este datorată unor fragmente de ADN viral care s-au inserat în ADN-ul uman. Deoarece ADN-ul este alcătuit din zone largi care nu codifică pentru ARN specific, transpozonii nu afectează adesea fenotipul. Uneori, transpozonii se inserează în exoni și influențează codificarea ARN-ului. În acest caz, prezența lor poate influența fenotipul.

3.5.10. Transpozoni în algoritmii genetici

Operatorul pentru mutație inspirat din caracteristicile transpozonilor, selectează aleatoriu, cu o șansă prestabilită, cromozomi care vor suferi mutația. Ulterior, sunt generate fortuit un locus cu alelă 1 și o valoare pentru distanța deplasării. Distanța poate rezulta în valori negative, specificând o migrație la stânga, sau pozitive pentru deplasarea la dreapta, cu numărul de poziții specificat. Funcționarea operatorului de mutație inspirat de transpozoni este ilustrat în Fig. 3.14.

3.6. Concluzii

Propunem o abordare nouă, o alternativă la definirea unei scheme rigide de dominare pentru maparea genotipului la fenotip, care modelează dominanța incompletă din biologie. Abordarea noastră este simplă, flexibilă unor cadre experimentale variate și ne așteptăm să susțină evoluția AG, în special în contexte dinamice. Implementările de algoritmi genetici diploizi din literatură utilizează, în

general, scheme de dominare special concepute pentru a adresa o problemă particulară. Propunerea noastră este originală, nu impune definirea unei scheme de dominare, iar sorgintea conceptului într-un fenomen care fundamentează evoluția naturală, testat pe parcursul a miliarde de ani, îi conferă generalitate.

Am conceput un operator original, diferit de alternativele tratate în literatura de specialitate, pentru recombinări, inspirat din atribuirea aleatorie a cromozomilor în timpul meiozei. Juxtapus cu distribuirea genelor pe un număr variabil de cromozomi de lungimi diferite, noul operator susține explorarea pe parcursul evoluției AG.

Am implementat variante de operatori pentru mutații, sugerați de fenomene din genetica umană. Teoretic, aceste propuneri ar trebui să adreseze tendința mutației punctuale de-a amplifica numărul atributelor considerate pe măsura evoluției generațiilor. Urmărim, de asemenea, cu ajutorul operatorilor concepuți, să stimulăm o tendință a AG de-a părăsi un optim local pe parcursul evoluției.

În analiza datelor achiziționate cu tehnologia ADN microarray, finalitatea este adesea descoperirea unui subgrup de gene semnificative pentru o anumită patologie. Este dezirabil ca genele din subgrupul selectat, nu doar să poată fi legate cauzal de condiția studiată, consecutiv validării biologice, ci să existe relații interpretabile între genele alese. Din aceste motive, selectarea atributelor din datele ADN microarray trebuie să răspundă unor particularități bine conturate. Atât modelul dominanței incomplete, cât și noul operator pentru recombinări cu atribuirea aleatorie a unui număr variabil de cromozomi, au fost concepuți pentru a răspunde finalității și contextului experimental impus de selectarea atributelor din acest tip de date.

Capitolul 5 este dedicat testării validității acestor propuneri și evaluarea impactului fiecăreia în selectarea unui număr restrâns de atribute, interpretabile biologic, din datele ADN microarray.

```
> chrConf
[1] 1 1 1 1 2 2 2 2
> individualsOriginal
  Id 1005_at 1007_s_at 1008_f_at 1009_at 1020_s_at 1030_s_at 1038_s_at 1039_s_at
1  1       1         1         1       0         0         0         0         1
2  1       1         1         1       0         1         0         0         0
3  2       1         1         1       0         1         0         0         0
4  2       1         0         0       1         1         0         0         1
5  3       0         0         1       1         0         1         0         1
6  3       1         0         0       0         1         0         1         1
7  4       0         1         0       0         0         1         1         1
8  4       0         1         1       1         0         0         0         1
> individuals
  Id 1005_at 1007_s_at 1008_f_at 1009_at 1020_s_at 1030_s_at 1038_s_at 1039_s_at
1  1       1         1         1       0         0         0         0         1
2  1       1         1         1       0         1         0         0         0
3  2       1         1         1       0         0         1         0         0
4  2       1         0         0       1         1         0         0         1
5  3       0         0         1       1         1         0         0         1
6  3       1         0         0       0         1         0         1         1
7  4       0         1         0       0         0         1         1         1
8  4       0         1         1       0         0         0         0         1
```

Fig. 3.14 – Efectele transpozonilor.

4. PACHETUL R dGAselID

Evoluţia remarcabilă a tehnicii de calcul vine în întâmpinarea necesităţii de-a analiza volumele mari de date rezultate în urma cercetărilor actuale. Algoritmi care necesitau accesul la sisteme de calcul foarte puternice în urmă cu numai zece ani pot fi astăzi implementaţi pe calculatoare personale.

Abordarea propusă pentru selectarea atributelor în datele ADN microarray este implementată în pachetul software dGAselID. Deși este conceput pentru a selecta atribute în datele de ADN microarray, algoritmul poate fi aplicat unei game largi de probleme care impun selectarea unui număr variabil de atribute în date cu un număr mare de dimensiuni. Toate metodele propuse și testările efectuate pe parcursul tezei de doctorat sunt desfășurate utilizând dGAselID.

Pentru a fi ușor accesibil cercetătorilor interesați și a integra sau compara metode propuse de alți investigatori, pachetul dGAselID a fost dezvoltat în compatibilitate cu un proiect software utilizat pe larg în mediul academic pentru analiza datelor rezultate din studii de tip ADN microarray.

Capitolul de fața descrie pachetul software dGAselID, dezvoltat pentru a implementa algoritmul propus în teza de doctorat. Revizuim detalii legate de formatul datelor de intrare, funcțiile disponibile și semnificația argumentelor acceptate. De asemenea, rezumăm aspecte legate de utilizarea practică a soft-ului și modului de vizualizare și sintetizare al rezultatelor.

4.1. R și Bioconductor

Odată cu evoluţia tehnicii de calcul, au fost dezvoltate pachete software complexe, care pun la dispoziţia cercetătorilor o gamă largă de algoritmi utilizaţi în PR şi oferă posibilitatea dezvoltării de metode noi. Între multiplele pachete software orientate pe IA şi analiză statistică, un avânt remarcabil l-a cunoscut limbajul de programare R. R a fost dezvoltat din limbajul de programare S, de către Robert Gentleman şi Ross Ihaka la Auckland University din Noua Zeelandă. La ora actuală, R a fost îmbrăţişat de un număr mare de statisticieni, ingineri şi cercetători din toate domeniile. Accesul liber la surse şi documentaţia variată au atras cercetători din domenii foarte variate şi, implicit, a determinat dezvoltarea unui număr impresionant de pachete software, implementări de algoritmi aplicabili în domenii foarte variate. Atât R cât şi diferite pachete cu documentaţiile aferente pot fi descărcate gratuit de pe pagina oficială a proiectului.

Interesul progresiv faţă de tehnologia microarray a impulsionat dezvoltarea de pachete R care vin în întâmpinarea cercetătorilor bioinformaticieni ce lucrează cu diferitele tehnologii ADN microarray. Un efort conjugat care sprijină cercetătorii în bioinformatică este Bioconductor. Asemenea limbajului R, **Bioconductor** este un proiect open source şi open development, în continuă expansiune şi modernizare, ce oferă implementări foarte bine documentate pentru metode necesare cercetătorilor, în special celor care se concentrează pe analiza genetică. Bioconductor a fost demarat în 2001 la Fred Hutchinson Cancer Research Center şi actualmente este dezvoltat de echipa Bioconductor core team formată din cercetători de la multiple institute şi universităţi din întreaga lume. Pachete, documentaţii şi o varietate de seturi de date, pot fi descărcate gratuit de pe pagina oficială a proiectului.

Calitatea şi diversitatea implementărilor, portabilitatea, continua evoluţie, flexibilitatea pentru dezvoltarea de metode proprii şi accesul facil la documentaţie recomandă R şi Bioconductor pentru cercetarea noastră. Pachetul **MLInterfaces** [77] din Bioconductor oferă o abordare foarte convenabilă pentru multe dintre metodele PR.

Am ales implementarea algoritmului propus de noi în mediul R, integrat cu Bioconductor. Popularitatea lor în mediul academic este binemeritată și justificată de paleta largă de aplicații disponibile pentru analiza datelor multi-dimensionale. Proiectul R înglobează o gamă variată de metode de analiză statistică, iar Bioconductor integrează instrumente foarte valoroase în analiza genetică. În plus ambele proiecte sunt deschise și beneficiază de contribuțiile numeroșilor cercetători implicați activ în domeniile respective. Un argument semnificativ pentru alegerea noastră a fost accesul liber la implementări, surse, documentație și comunitatea pasionată și sociabilă de cercetători care utilizează și contribuie la cele două proiecte. Bioconductor oferă acces liber la numeroase seturi de date ADN microarray și facilitează posibilitatea de-a dezvolta și compara metode pentru analiza genetică. R și Bioconductor oferă o multitudine de pachete special implementate pentru a facilita fiecare pas în analiza microarray, de la achiziție și preprocesare, până la inferențe și interpretarea semnificației biologice a rezultatelor obținute. Prezentarea și evaluarea diferitelor metode disponibile în acest sens a fost abordată pe larg în jurnale [85]. Metodologia de cercetare cu ADN microarray cu emfază pe uneltele oferite în R și Bioconductor [86] a fost tratată extins în literatura de specialitate.

Bioconductor utilizează structuri de date special concepute pentru datele de ADN microarray, dar flexibile la utilizare în cadre variate de analiză. Am decis să beneficiem de clasa ExpressionSet din Bioconductor, atât datorită accesului facil la seturi de date în acest format pe pagina proiectului, dar și datorită maleabilității acestui format la tipuri de date variate. Algoritmul implementat de noi utilizează date formatate după specificațiile clasei ExpressionSet. Din acest motiv, pachetul este perfect integrabil în Bioconductor.

Numeroși contribuitori au dezvoltat pachete R orientate pe algoritmi genetici. Diferite propuneri de algoritmi genetici pentru contexte de optimizare **GA** [87], **gaoptim** [88]**, genalg** [89]**, nsga2R**[90] sau pentru cadre de analiză speciale **kofnGA**[91], **STPGA**[92] sunt disponibile pe sit-ul proiectului R. Alți autori au implementat algoritmi genetici [93, 94] pentru selectarea atributelor în diferite contexte.

4.2. Pachetul software dGAselID

Pachetul software **dGAselID** a fost creat pentru a implementa abordarea propusă în teza de doctorat. Pachetul Bioconductor MLInterfaces oferă implementări flexibile pentru diverse metode supervizate și nesupervizate specifice ML. De asemenea, metode de validare încrucișată sunt disponibile, iar pentru algoritmi specifici există variante optimizate pentru utilizarea în acest context, care aduc beneficii majore în privința timpului de execuție. Acest aspect este foarte dezirabil în utilizarea acestor clasificatori pentru evaluarea adaptabilității în algoritmul genetic diploid propus, unde numărul evaluărilor este imens, iar avantajul în durata necesară pentru execuția a sute de iterații devine evident. Orice algoritm de învățare implementat în MLInterfaces este accesibil de către AG în dGAselID, pentru evaluarea adaptabilității. Valoarea numerică utilizată pentru caracterizarea

adaptabilității este acuratețea în discriminarea dintre cele două clase de exemple a oricărui astfel de clasificator.

Algoritmul genetic diploid propus în pachetul **dGAselID** atribuie fiecare probă din setul de date de intrare unei gene în genotip, cu locus-ul corespunzător poziției din datele originale. Datele de intrare trebuie să fie în formatul impus de specificațiile clasei ExpressionSet din Bioconductor. Această clasă a fost special gândită pentru a stoca și integra datele necesare pentru descrierea, execuția și replicarea unui experiment cu ADN microarray. Cu toate acestea, datele voluminoase pentru experimente care necesită analiză de recunoașterea formelor din domenii variate, pot fi cu ușurință formatate corespunzător. Considerăm așadar că adoptarea acestui format pentru datele de intrare nu reprezintă o limitare a pachetului software **dGAselID**, ci mai degrabă o flexibilizare în vederea integrării cu alte metode implementate în Bioconductor, pentru analiză genetică și nu numai.

Clasa ExpressionSet înmagazinează informații variate despre un experiment ADN microarray. Pe lângă măsurătorile expresiilor genetice, structura înregistrează date despre exemple, tehnologia utilizată și cadrul experimental în care au fost colectate datele respective. Pentru a descrie sugestiv structura datelor de intrare în algoritmul propus, ilustrăm clasa ExpressionSet cu ajutorul setului de date ALL [95], utilizat în partea experimentală a lucrării de doctorat. Aspectul setului de date este prezentat în Fig. 4.1.

Măsurătorile pentru expresia genetică a probelor imobilizate pe chip sunt stocate sub forma unei matrice, cu probele pe linii și exemplele pe coloane. Formatul de stocare al măsurătorilor este prezentat în Fig. 4.2. Pentru fiecare dintre exemplele setului de date sunt memorate numeroase alte caracteristici, în concordanță cu specificațiile clasei AnnotatedDataFrame din Bioconductor. Datele de fenotip, diferă de la un experiment la altul, iar structura este flexibilă în a acomoda orice date de fenotip disponibile. Datele de fenotip înregistrate în pachetul ALL sunt prezentate în Fig. 4.3. Metode de accesare al acestor informații sunt, de asemenea, implementate.

Datele în formatul acesta sunt preluate din algoritmul genetic diploid și tratate după metoda discutată în capitolul 3. Genotipurile sunt generate aleatoriu, cu nu număr de gene active solicitat de utilizator printr-un parametru la inițializare. Distribuția genelor pe cromozomi se realizează, de asemenea, în funcție de opțiunile specificate de cercetător. În etapa următoare, este generată populația inițială și sunt evaluați indivizii cu funcția de evaluare apelată, respectând cadrul impus de implementarea dominanței incomplete. Funcția de evaluare aleasă este specificată la inițializare, și poate fi reprezentată de oricare dintre metodele disponibile în pachetul MLInterfaces din Bioconductor. Pachetul include implementări pentru clasificatorii uzual utilizați în analiza datelor de microarray, dar, de asemenea, aplicabili pentru diferite alte scenarii de căutare a formelor în date vaste. Implementările svmI, ldaI, rdaI, knnI, randomForrestI, dldaI, nnetI, qdaI, naiveBayesI, sunt doar o parte dintre opțiunile oferite de pachetul MLInterfaces. O parte dintre metodele oferite de pachetul MLInterfaces, utilizează la rândul lor implementări cuprinse în alte [33, 96] pachete R sau Bioconductor.

Abordarea integrată din punct de vedere al formatului de stocare al datelor, metodelor disponibile pentru clasificare supervizată sau nesupervizată, sau opțiuni pentru vizualizare recomandă MLInterfaces și Bioconductor pentru căutarea formelor în analiza genetică, dar și pentru diferite alte aplicații. Validarea încrucișată, dacă este dorită de utilizator, sau modul de separare al datelor în subseturi pentru testare și antrenament, trebuie specificată în concordanță cu notațiile din pachetul MLInterfaces.

```
> library(ALL)
> data(ALL)
> ALL
ExpressionSet (storageMode: lockedEnvironment)
assayData: 12625 features, 128 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: 01005 01010 ... LAL4 (128 total)
  varLabels: cod diagnosis ... date last seen (21 total)
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
  pubMedIds: 14684422 16243790
Annotation: hgu95av2
```

Fig. 4.1 – Setul de date ALL.

```
> exprs(ALL)[1:10, 1:8]
            01005     01010    03002    04006    04007     04008     04010     04016
1000_at   7.597323  7.479445 7.567593 7.384684 7.905312  7.065914  7.474537  7.536119
1001_at   5.046194  4.932537 4.799294 4.922627 4.844565  5.147762  5.122518  5.016132
1002_f_at 3.900466  4.208155 3.886169 4.206798 3.416923  3.945869  4.150506  3.576360
1003_s_at 5.903856  6.169024 5.860459 6.116890 5.687997  6.208061  6.292713  5.665991
1004_at   5.925260  5.912780 5.893209 6.170245 5.615210  5.923487  6.046607  5.738218
1005_at   8.570990 10.428299 9.616713 9.937155 9.983809 10.063484 10.662059 11.269115
1006_at   3.656143  3.853979 3.646808 3.874289 3.547361  3.771648  3.747963  3.318315
1007_s_at 7.623562  7.543604 7.916954 6.816397 7.516981  7.726716  7.288960  7.724153
1008_f_at 8.903547  9.903953 8.494499 9.533983 8.871669  9.424092  8.999938  8.006848
1009_at   9.371888  9.322177 9.304982 9.135370 9.627175  9.189420  9.312164  9.946430
```

Fig. 4.2 – Stocarea măsurătorilor expresiei genetice în setul de date ALL. Doar primele 8 exemple și primele 10 atribute sunt reprezentate pentru claritate.

Populația este ulterior afectată cu operatorii de recombinare și mutație. Recombinările în două puncte au loc la nivelul fiecărui cromozom. În analiza datelor ADN microarray, unde se caută un subset redus de gene dint-un număr imens de probe, seturile haploide de cromozomi au consecutiv, un număr restrâns de gene active. Prin urmare, un număr important de recombinări în două puncte nu vor afecta cromozomii respectivi. În consecință, nu am considerat că este necesară implementarea unui parametru suplimentar pentru a specifica șansa ca o recombinare să aibă loc. Utilizatorul are însă la dispoziție modalități eficiente de-a influența rata recombinărilor. Deoarece fiecare cromozom suferă o recombinare, ajustarea numărului cromozomilor pe care sunt distribuite genele, afectează rata recombinărilor. În plus, utilizarea opțională a operatorului de atribuire aleatorie a cromozomilor contribuie la explorarea algoritmului genetic. De asemenea, mai multe variante de operatori pentru mutație pot fi selectați sau combinați, în funcție de imperativele cercetării.

```
> print(summary(pData(ALL)))
     cod             diagnosis          sex          age              BT     remission
 Length:128         Length:128         F   :42   Min.   : 5.00   B2     :36   CR  :99
 Class :character   Class :character   M   :83   1st Qu.:19.00   B3     :23   REF :15
 Mode  :character   Mode  :character   NA's: 3   Median :29.00   B1     :19   NA's:14
                                                 Mean   :32.37   T2     :15
                                                 3rd Qu.:45.50   B4     :12
                                                 Max.   :58.00   T3     :10
                                                 NA's   :5       (Other):13

      CR              date.cr           t(4;11)         t(9;22)        cyto.normal
 Length:128         Length:128         Mode :logical   Mode :logical   Mode :logical
 Class :character   Class :character   FALSE:86        FALSE:67        FALSE:69
 Mode  :character   Mode  :character   TRUE :7         TRUE :26        TRUE :24
                                       NA's :35        NA's :35        NA's :35

    citog             mol.biol    fusion protein   mdr          kinet       ccr
 Length:128         ALL1/AF4:10   p190     :17    NEG :101   dyploid:94   Mode :logical
 Class :character   BCR/ABL :37   p190/p210: 8    POS : 24   hyperd.:27   FALSE:74
 Mode  :character   E2A/PBX1: 5   p210     : 8    NA's:  3   NA's   : 7   TRUE :26
                    NEG     :74   NA's     :95                            NA's :28
                    NUP-98  : 1
                    p15/p16 : 1

  relapse        transplant          f.u            date last seen
 Mode :logical   Mode :logical   Length:128         Length:128
 FALSE:35        FALSE:91        Class :character   Class :character
 TRUE :65        TRUE :9         Mode  :character   Mode  :character
 NA's :28        NA's :28
> dim(pData(ALL))#Dimensiunea datelor de fenotip
[1] 128  21
> names(pData(ALL))       #Variabile prezente în fenotip
 [1] "cod"           "diagnosis"     "sex"           "age"           "BT"
 [6] "remission"     "CR"            "date.cr"       "t(4;11)"       "t(9;22)"
[11] "cyto.normal"   "citog"         "mol.biol"      "fusion protein" "mdr"
[16] "kinet"         "ccr"           "relapse"       "transplant"    "f.u"
[21] "date last seen"
```

Fig. 4.3 – Datele despre fenotip în setul de date ALL.

Seturile haploide de cromozomi care se vor regăsi în generația viitoare sunt selectate și reunite cu seturile obținute prin recombinare. O nouă populație de indivizi este generată prin împerecherea întâmplătoare a acestor genotipuri. Consecutiv, este inițiată o nouă iterație a algoritmului genetic.

Pachetul dGAselID este construit în jurul funcției `DGAselID()`. Această funcție inițializează algoritmul și stabilește cadrul selectării atributelor. Argumentele acceptate de funcția `DGAselID()` permit ajustarea căutării în raport cu imperativele cercetării și cunoștințele disponibile despre configurația datelor. Tabelul 4.1 sintetizează aceste argumente. Un exemplu de inițializare a algoritmului genetic cu funcția `DGAselID()` este prezentat în Fig. 4.4.

`DGAselID()` apelează la nevoie alte funcții, în raport cu cadrul experimental stabilit la inițializare. Alte funcții execută etape individuale în algoritmul genetic propus sau acțiunea operatorilor de recombinare ori mutație, selectați în contextul specificat. Tabelul 4.2 prezintă funcțiile implementate în pachetul **dGAselID** și schițează acțiunea fiecăreia.

```
> res<-DGAselID(smallALL, "mol.biol", method=knn.cvI(k=8, l=5), +
+ trainTest=1:79, startGenes=12, populationSize=200, +
+ iterations=500, noChr=5, randomAssortment=TRUE, +
+ mutationChance=0.005, elitism=5)
```

Fig. 4.4 – Variabile accesibile în rezultatul returnat de algoritmul genetic propus.

Tabel 4.1 - Argumente acceptate de funcția `DGAselID()`

Argument	Descriere
`x`	Setul de date în format `ExpressionSet`
`response`	Variabilă răspuns
`startGenes`	Numărul alelelor=1 în genotipurile inițiale
`populationSize`	Dimensiunea populație inițiale
`iterations`	Generații țintă (condiția de terminare)
`elitism`	Elitism în procente
`mutationRate`	Șansa ca o mutație punctuală să apară
`nonSenseRate`	Șansa ca o mutație fără sens să apară
`frameShiftRate`	Șansa ca o mutație cu deplasare să apară
`chrSegDelRate`	Șansa ca o mutație cu ștergerea unui segment de cromozom să apară
`chrDelRate`	Șansa ca o mutație cu ștergerea unui cromozom să apară
`transposonRate`	Șansa ca transpozoni să producă o mutație
`method`	Clasificator utilizat în evaluarea adaptabilității
`trainTest`	Separarea datelor în subseturi de antrenament/testare sau validare încrucișată
`noChr`	Numărul cromozomilor în genotip
`randomAssortment`	Valoarea TRUE, se aplică operatorul de atribuire aleatorie a cromozomilor
`size`	Exclusiv pentru nnetI
`decay`	Exclusiv pentru nnetI
`alpha`	Exclusiv pentru rdaI
`delta`	Exclusiv pentru rdaI

Tabel 4.2 - Funcții implementate în pachetul **dGAselID**

Funcție	Descriere
`DGAselID()`	Funcția principală
`InitialPopulation()`	Generează seturile de cromozomi inițiale
`Individuals()`	Generează indivizii
`splitChromosomes()`	Distribuie genele pe numărul dorit de cromozomi
`RandomizePop()`	Generează populația pentru iterația următoare
`EvaluationFunction()`	Evaluează adaptabilitatea indivizilor
`AnalyzeResults()`	Ordonează indivizii după performanță și înregistrează rezultatele la fiecare generație
`Crossover()`	Execută recombinarea în două puncte
`RandomAssortment()`	Execută atribuirea aleatorie a cromozomilor
`Mutations()`	Execută mutația punctuală
`NonSenseMutations()`	Execută mutația fără sens
`FrameShiftMutation()`	Execută mutația cu deplasare
`LargeSegmentDeletion()`	Execută mutația cu ștergerea unui segment de cromozom
`WholeChromosomeDeletion()`	Execută ștergerea unui cromozom
`Transposons()`	Execută mutația datorată transpozonilor
`Elitism()`	Execută selecția elitistă
`PlotGenAlg()`	Reprezintă grafic evoluția după fiecare generație

Unul dintre dezavantajele selectării atributelor cu algoritmul genetic diploid propus este timpul de execuție. Pe parcursul realizării tezei de doctorat funcțiile descrise mai sus au suferit modificări și optimizări pentru a adresa acest neajuns.

Îmbunătățirile s-au dovedit substanțiale, dar timpul de execuție a rămas un inconvenient. Selecția atributelor într-un set de date cu 1000 de probe și 80 de exemple se realizează pe parcursul a aproximativ 10 minute, pe un calculator personal cu specificații medii. Un număr de replicații ale experimentului sunt necesare pentru a concluziona asupra atributelor importante în fiecare investigare, ceea ce multiplică durata necesară unui experiment.

În timpul rulării algoritmului, informații despre evoluție sunt disponibile utilizatorului. Date cu privire la acuratețea minimă, medie și maximă în populația curentă sunt afișate pe ecran după fiecare evaluare. De asemenea, cercetătorul este informat în legătură cu numărul mutațiilor efectuate la fiecare iterație și etapa curentă în desfășurarea algoritmului.

Algoritmul afișează, de asemenea, o reprezentare grafică a evoluției după fiecare iterație. Evoluțiile acurateței maxime și medii acompaniază o histogramă a genelor cel mai frecvent selectate, în reprezentarea grafică prezentată după evaluarea fiecărei generații. Capturi de ecran din timpul desfășurării algoritmului sunt prezentate în Fig. 4.5 și Fig. 4.6.

Pe parcursul execuției, algoritmul înregistrează caracteristicile evoluției. Informații cu privire la frecvența de selecție a fiecărei gene și a prezenței ei în setul haploid de cromozomi mai adaptat din individ sunt memorate. De asemenea individul cel mai performat în fiecare iterație este consemnat în datele de ieșire. După ce condiția de terminare, numărul de generații specificat, este îndeplinită, aceste informații sunt disponibile pentru analiză suplimentară. Variabilele înregistrate în rezultatul final sunt afișate în Fig. 4.7 și descrise în tabelul 4.3.

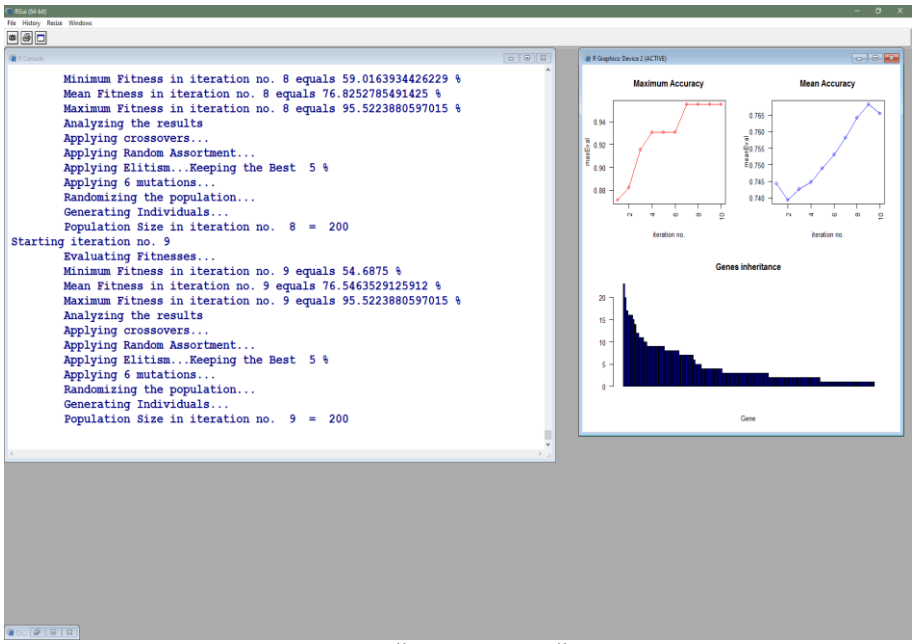


Fig. 4.5 - Captură de ecran după 9 generații.

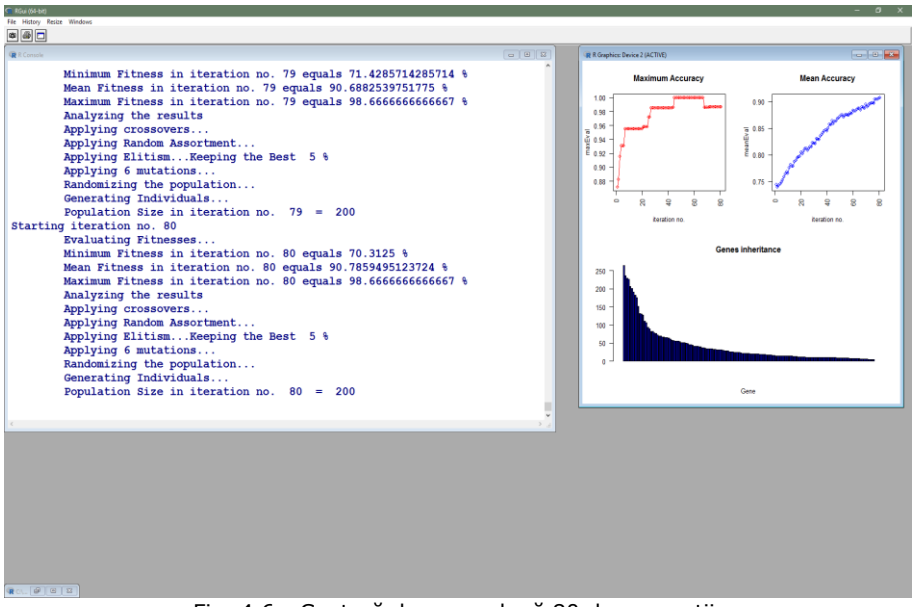


Fig. 4.6 - Captură de ecran după 80 de generații.

```
> names(result)
[1] "DGenes"           "dGenes"           "MaximumAccuracy" "MeanAccuracy"
[5] "MinAccuracy"      "BestIndividuals"
```
Fig. 4.7 – Variabile accesibile în rezultatul returnat de algoritmul genetic propus.

Rezultatele obținute în urma selectării atributelor cu algoritmul propus pot fi vizualizate cu metode disponibile în pachet. O altă perspectivă asupra căutării poate fi obținută prin reprezentarea grafică a evoluției celui mai adaptat individ în fiecare generație (Fig. 4.8). De asemenea ilustrări ale evoluției acurateței maxime (Fig. 4.9) și medii (Fig. 4.10) pot fi generate și interpretate în completare.

Tabel 4.3 - Rezultatul returnat de algoritm

Variabilă	**Descriere**
`DGenes`	Apariții în genele selectate pe parcursul tuturor generațiilor
`dGenes`	Apariții în genele eliminate pe parcursul tuturor generațiilor
`MaximumAccuracy`	Acuratețea maximă în fiecare generație
`MeanAccuracy`	Acuratețea medie în fiecare generație
`MinAccuracy`	Acuratețea minimă în fiecare generație
`BestIndividuals`	Cel mai adaptat individ în fiecare generație

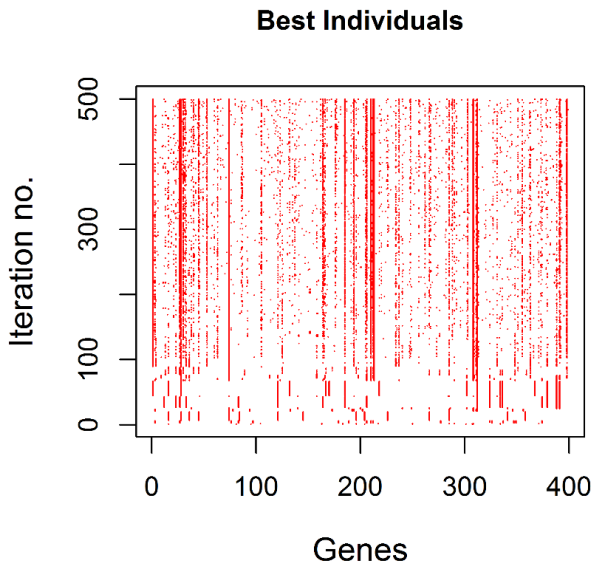


Fig. 4.8 – Evoluția celui mai adaptat set haploid de cromozomi.

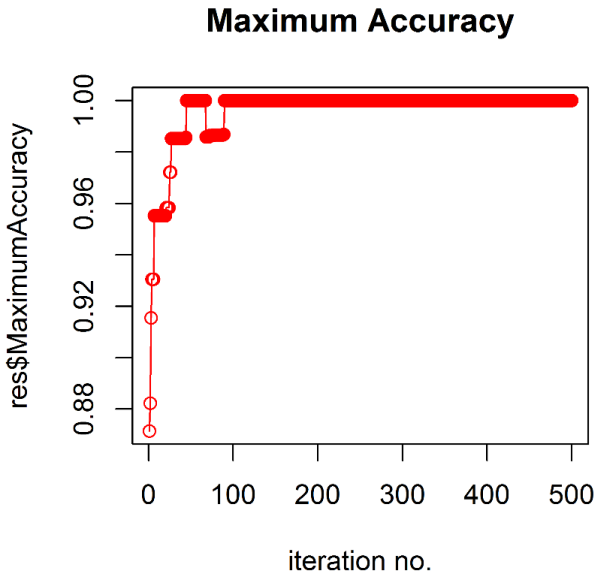


Fig. 4.9 – Evoluția acurateței maxime după 500 de generații.

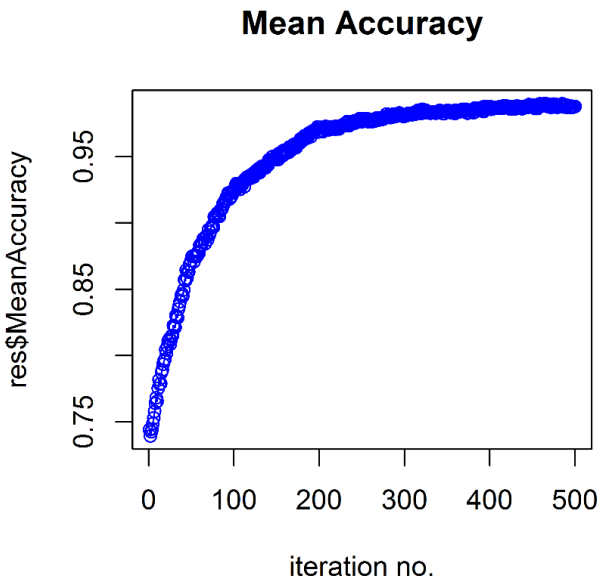


Fig. 4.10 – Evoluția acurateței medii după 500 de generații.

Pe de o parte, genele selectate cel mai frecvent pe parcursul întregii evoluții pot fi vizualizate (Fig. 4.11), iar un număr de gene considerate semnificative pentru analize ulterioare poate fi apoi obținut cu numele și o descriere sumară (Fig. 4.12). Bioconductor oferă o varietate de metode de vizualizare, analiză și interpretare a rezultatelor de analiză genetică. Toate aceste metode sunt disponibile și accesibile facil cu formatul datelor de ieșire returnat de algoritmul nostru.

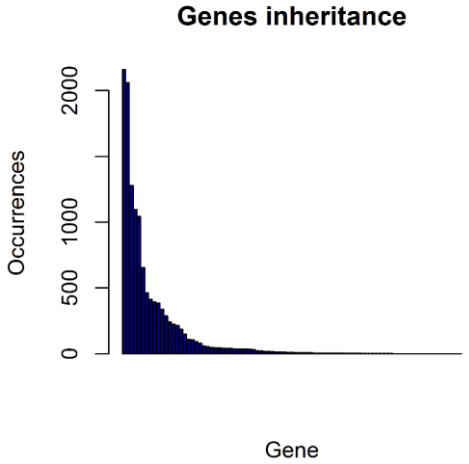


Fig. 4.12 – Genele selectate cel mai frecvent după 500 de generații.

```
$`39730_at`
[1] "ABL proto-oncogene 1, non-receptor tyrosine kinase"

$`32562_at`
[1] "endoglin"

$`37283_at`
[1] "meningioma (disrupted in balanced translocation) 1"

$`37027_at`
[1] "AHNAK nucleoprotein"

$`39837_s_at`
[1] "zinc finger protein 467"

$`1636_g_at`
[1] "ABL proto-oncogene 1, non-receptor tyrosine kinase"

$`1635_at`
[1] "ABL proto-oncogene 1, non-receptor tyrosine kinase"

$`37099_at`
[1] "arachidonate 5-lipoxygenase activating protein"

$`AFFX-HSAC07/X00351_3_st`
[1] "actin, beta"

$`36630_at`
[1] "TSC22 domain family member 3"
```

Fig. 4.12 – Descrierea primelor 10 cel mai selectate gene.

4.3. Concluzii

Pachetul software **dGAselID** propune o soluție originală și eficientă la problema selectării atributelor în datele ADN microarray. Pachetul este perfect integrabil și beneficiază de toate metodele de analiză și vizualizare din Bioconductor și deopotrivă, completează metodele de analiză a datelor implementate în acel mediu cu o alternativă pentru selectarea atributelor în acest tip de experiment. Metoda propusă este flexibilă și aplicabilă unei game largi de probleme care impun selectarea atributelor în date vaste. Operatorul original pentru atribuirea aleatorie a cromozomilor și modelul dominanței incomplete pentru maparea genotipului la fenotip favorizează explorarea și produc rezultate superioare metodelor implementate în MLInterfaces pentru selectarea atributelor. Diferitele opțiuni de operatori pentru mutații oferă flexibilitate în tratarea a diverse seturi de date sau aplicații de selectare a atributelor. Elasticitatea în utilizarea clasificatorilor supervizați și a multiplelor variante de validare încrucișată sunt foarte dezirabile în contextul selectării atributelor în date microarray sau alte contexte. Pentru rezultate foarte consistente, atât cunoștințele a priori despre datele analizate, cât și testarea unora dintre opțiunile disponibile în pachet pot determina cadrul optim pentru algoritmul diploid propus.

5. EXPERIMENTE

Pe parcursul realizării tezei de doctorat, am studiat metodele domeniului inteligenței artificiale care pot susține obiectivul stabilirii unei relații de cauzalitate între un grup restrâns de gene și o anumită patologie cu ajutorul tehnologiei ADN microarray. De asemenea, am analizat fenomenele responsabile de evoluția naturală, deslușite și explicate detaliat de genetica modernă cu scopul modelării lor pentru ameliorarea performanței algoritmilor genetici. Aceste activități s-au concretizat în metodele propuse în capitolul 3. Algoritmul genetic diploid, conceput pentru selectarea unui număr restrâns de atribute din date vaste, beneficiind de contribuțiile noastre a fost implementat în pachetul software dGAselID, prezentat în capitolul precedent.

În această etapă, ne propunem să testăm performanța metodei concepute de noi, în implementarea din pachetul dGAselID, pentru a evalua impactul propunerilor noastre asupra selectării atributelor din date ADN microarray reale. Am sugerat metode de-a adresa aspecte diferite ale evoluției algoritmilor genetici. În consecință vom testa propunerile noastre separat, pentru a aprecia impactul fiecăreia asupra comportamentului AG. Efectul cumulativ al metodelor descoperite a susține semnificativ selectarea atributelor cu algoritmul evoluționist, sunt testate consecutiv.

Am utilizat un set de date reale ADN microarray, disponibil grație proiectului Bioconductor, pentru a evalua metoda noastră într-un context concret. Popularitatea de care tehnologia microarray s-a bucurat în deceniul recent, permite accesul la date și rezultate reale și fundamentează o relație mutual avantajoasă între IA și bioinformatică. IA oferă metode îmbunătățite pentru abordarea problematicilor din bioinformatică. Pe de altă parte, bogăția datelor și rezultatelor disponibile în bioinformatică asigură un cadru propice progresului metodelor IA, aplicabile ulterior în domenii variate.

Metodele propuse de noi care vor fi testate și evaluate în capitolul de față sunt:

1. **dominanța incompletă** pentru maparea genotipului la fenotip,
2. **operatorul atribuirii aleatorii a cromozomilor** în contextul unui număr variabil de cromozomi cu dimensiuni diferite,
3. **operatorii pentru mutații**.

5.1. Setul de date Acute Lymphoblastic Leukemia

Experimentele din acest capitol analizează setul de date Acute Lymphoblastic Leukemia (ALL) [95]. Am ales să utilizăm acest set de date ADN microarray în experimentul nostru deoarece rezidă în date reale, care au fost în prealabil prelucrate. Procesul de asamblare, prelucrare și normalizare al datelor rezultate în urma unui astfel de experiment este în sine o provocare, iar fiecare etapă poate fi afectată de diferite tipuri de erori sau metode de abordare. Setul ALL constă în 128 de exemple, pacienți suferinzi de leucemie și pentru fiecare dintre ele, 12625 de atribute reprezentând probe de pe chipuri Human Genome U95 Set produse de compania Affymetrix. Informațiile au fost colectate la Dana Faber Cancer

Institute din Boston și sunt oferite gratuit, în formatul Expression Set, pe situl Bioconductor. Pachetul conținând setul ALL este publicat de Li X. În plus, pachetul cu datele respective este public de o perioadă suficient de îndelungată pentru a fi accesat și analizat cu o paletă largă de metode. Rezultatele analizelor setului de date ALL au fost publicate și criticate in extenso [97]. Figura 5.1 ilustrează caracteristicile setului de date ALL, în implementarea din Bioconductor.

Formatul Expression Set utilizat în Bioconductor și adoptat ca standard pentru datele de intrare în pachetul software dezvoltat, rezultat în urma cercetărilor din cursul doctoratului de față, înmagazinează numeroase informații importante despre datele respective. Spre exemplu, descrierea cadrului experimental care a generat datele utilizate de noi sunt prezentate în Fig. 5.2.

Datele care descriu fenotipul cazurilor înregistrate în setul de date poate fi examinat facil (Fig. 5.3). O serie de variabile considerate de către cercetătorii care au colectat datele sunt disponibile pentru a descrie fiecare caz în parte. Investigatorii au stocat variabile care descriu sexul (`sex`) și vârsta (`age`) fiecărui pacient. Informații cu privire la diagnosticul și evoluția fiecărui pacient sunt, de asemenea, disponibile în variabilele `BT`, `remission` și `CR`. Fenotipul setului de date ALL cuprinde și o altă categorie de date, cu importanță deosebită, despre pacienții monitorizați.

```
> ALL
ExpressionSet (storageMode: lockedEnvironment)
assayData: 12625 features, 128 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: 01005 01010 ... LAL4 (128 total)
  varLa a: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
  pubMedIds: 14684422 16243790
Annotation: hgu95av2
bels: cod diagnosis ... date last seen (21 total)
  varMetadat
```

Fig. 5.1 - Caracteristicile setului de date ALL

```
> experimentData(ALL)
Experiment data
  Experimenter name: Chiaretti et al.
  Laboratory: Department of Medical Oncology, Dana-Farber Cancer Institute,
Department of Medicine, Brigham and Women's Hospital, Harvard Medical School,
Boston, MA 02115, USA.
  Contact information:
  Title: Gene expression profile of adult T-cell acute lymphocytic leukemia
identifies distinct subsets of patients with different response to therapy and
survival.
  URL:
  PMIDs: 14684422 16243790
```

Fig. 5.2 - Cadrul experimental care a generat datele ALL

Anormalități cunoscute a fi implicate în apariția leucemiei sunt testate și înmagazinate pentru fiecare caz. Spre exemplu, variabilele `t(4;11)` și `t(9;22)` exprimă prezența sau absența translocațiilor sinonime. Variabila `mol.biol` înregistrează clasificarea din punct de vedere al biologie moleculare al cazurilor considerate. Prezența acestor aspecte în datele de fenotip oferă o oportunitate deosebită pentru studiul nostru.

În general, studiile de microarray sunt dezvoltate de echipe multidisciplinare cu componența complexă, deoarece diferite tipuri de activități și etape trebuie executate de cercetători specializați. Colectarea datelor, executarea experimentelor, validarea și preprocesarea măsurătorilor obținute sunt urmate de analiză statistică, detectarea formelor, iar rezultatele obținute trebuie validate biologic. Fiecare dintre aceste etape ridică probleme specifice și sunt realizate de cercetători cu pregătire și experiență diferite.

Pentru a adresa acest aspect și a testa validitatea propunerii noastre, o abordare alternativă poate fi adoptată. Ar fi foarte dificil să evaluăm validitatea metodei propuse utilizând date noi, în scopul de-a descoperi marcheri necunoscuți sau a stabili legătura cauzală dintre un grup de gene și o anumită patologie. Ar fi, de asemenea, problematic să validăm un grup nou, necunoscut, de gene, responsabil pentru o anumită boală. Analiza semnificației biologice ale unor astfel de descoperiri ar pune probleme suplimentare față de finalitatea noastră.

```
> print(summary(pData(ALL)))
     cod             diagnosis          sex          age              BT
 Length:128         Length:128         F   :42   Min.   : 5.00   B2     :36
 Class :character   Class :character   M   :83   1st Qu.:19.00   B3     :23
 Mode  :character   Mode  :character   NA's: 3   Median :29.00   B1     :19
                                                 Mean   :32.37   T2     :15
                                                 3rd Qu.:45.50   B4     :12
                                                 Max.   :58.00   T3     :10
                                                 NA's   :5       (Other):13

remission      CR              date.cr           t(4;11)
 CR  :99   Length:128         Length:128         Mode :logical
 REF :15   Class :character   Class :character   FALSE:86
 NA's:14   Mode  :character   Mode  :character   TRUE :7
                                                 NA's :35

  t(9;22)        cyto.normal        citog              mol.biol
 Mode :logical   Mode :logical   Length:128         ALL1/AF4:10
 FALSE:67        FALSE:69        Class :character   BCR/ABL :37
 TRUE :26        TRUE :24        Mode  :character   E2A/PBX1: 5
 NA's :35        NA's :35                           NEG     :74
                                                    NUP-98  : 1
                                                    p15/p16 : 1

   fusion protein    mdr          kinet       ccr           relapse
 p190     :17     NEG :101   dyploid:94   Mode :logical   Mode :logical
 p190/p210: 8     POS : 24   hyperd.:27   FALSE:74        FALSE:35
 p210     : 8     NA's:  3   NA's   : 7   TRUE :26        TRUE :65
 NA's     :95                             NA's :28        NA's :28

 transplant          f.u            date last seen
 Mode :logical   Length:128         Length:128
 FALSE:91        Class :character   Class :character
 TRUE :9         Mode  :character   Mode  :character
 NA's :28
```

Fig. 5.3 – Fenotipul asociat datelor ALL

Problema validării metodei propuse poate fi abordă în mod diferit. Ne propunem să descoperim un grup de gene care poate fi utilizat pentru a discrimina optim între exemple ale unor categorii bine cunoscute și descrise. Datele de fenotip despre biologia moleculară a pacienților incluși în studiul ALL oferă această oportunitate. O parte din genele cauzal responsabile de clasificarea din punct de vedere al biologiei moleculare în datele ALL sunt bine cunoscute. Așadar, ne așteptăm ca o metodă validă de selectare a atributelor să descopere aceste probe pe chip-urile considerate, ca fiind atribute esențiale în discriminarea între cazurile cu clasificări diferite în privința biologiei moleculare. De asemenea, ne așteptăm ca un număr de gene care nu sunt legate cauzal de aceste categorii să apară în subseturile de gene descoperite. Interpretarea semnificației lor reprezintă un aspect interesant pentru cercetarea noastră. Deși validarea biologică a acestor gene nu este accesibilă în studiul de față, date despre genele respective pot fi interpretate prin prisma cunoștințelor familiare despre ele.

O altă oportunitate oferită de setul de date selectat este compararea rezultatelor obținute cu concluziile obținute prin alte metode, accesibile în Bioconductor. O comparație cu metodele de analiză adesea utilizate poate revela aspecte valoroase în evaluarea metodei propuse în teza de doctorat.

La inspecția datelor de fenotip din Fig. 5.3 devine evident că cel mai bine reprezentate cazuri din punct de vedere al clasificării după biologia moleculară sunt exemplele cu `BCR/ABL pozitiv` și `negativ` (37 și 74 de cazuri dintre pacienți). După eliminarea din setul de date al pacienților cu leucemie acută limfoblastică cu celule T și a celorlalte clasificări de biologie moleculară, 79 exemple (Fig. 5.4) sunt păstrate pentru analiză suplimentară. Dintre acestea, 42 reprezintă exemple negative, iar 37 pozitive de `BCR/ABL`, într-o proporție dezirabilă pentru analiza de recunoașterea a formelor.

Deoarece dintre cele 12625 de probe măsurate pentru fiecare pacient, majoritatea valorilor nu variază cu patologia studiată și nu sunt valoroase în descrierea ei, datele pot fi consecutiv filtrate. Astfel, se obține o reducere a zgomotului din date și se câștigă semnificativ în privința timpului de execuție al algoritmului. Un număr cât mai mare de replicări pentru fiecare experiment, este dezirabil pentru a adresa componenta aleatorie a metodei și a obține rezultate solide. Așadar, timpul de execuție pentru fiecare dintre aceste repetiții este multiplicat semnificativ în desfășurarea unui experiment credibil. Utilizarea algoritmilor genetici impune un set de experimente prealabile pentru determinarea parametrilor de inițializare a căutării. Selectarea valorilor dezirabile pentru șansa ca o mutație să apară, pentru fiecare tip de mutație propus, sau numărul de indivizi păstrați în generația viitoare prin elitism, necesită rulări multiple ale algoritmului. Prin urmare, am decis să investigăm trei seturi de date pe parcursul testărilor. Am lucrat cu setul de date complet, format din 79 exemple și 12625 de probe. De asemenea, am analizat date obținute prin filtrarea nespecifică a setului complet după condițiile IQR(x) > 0.5 și cel puțin 25% dintre valori > log2(100).

Rezultatul a fost un subset cu 79 exemple și 2391 dintre cele 12625 de probe în setul original. În plus, am aplicat o metodă de filtrare specifică după testul t, cu valoarea de tăiere p=0.1 obținând o submulțime formată din 628 de atribute față de cele 2391 în setul filtrat. Aceste operațiuni sunt accesibile în Bioconductor cu pachetul **genefilter** [50], special dezvoltat pentru acest tip de activitate.

```
> print(summary(pData(smallALL)))
     cod             diagnosis          sex          age          BT
 Length:79          Length:79          F   :28   Min.   :15.00   B : 4
 Class :character   Class :character   M   :50   1st Qu.:19.00   B1: 9
 Mode  :character   Mode  :character   NA's: 1   Median :27.50   B2:35
                                                 Mean   :32.92   B3:22
                                                 3rd Qu.:48.25   B4: 9
                                                 Max.   :58.00
                                                 NA's   :3

 remission      CR             date.cr           t(4;11)
 CR  :60   Length:79          Length:79          Mode :logical
 REF : 9   Class :character   Class :character   FALSE:62
 NA's:10   Mode  :character   Mode  :character   NA's :17

  t(9;22)        cyto.normal        citog             mol.biol
 Mode :logical   Mode :logical   Length:79          BCR/ABL:37
 FALSE:36        FALSE:48        Class :character   NEG    :42
 TRUE :26        TRUE :14        Mode  :character
 NA's :17        NA's :17

   fusion protein   mdr         kinet       ccr           relapse
 p190     :17     NEG :60   dyploid:56   Mode :logical   Mode :logical
 p190/p210: 8     POS :18   hyperd.:19   FALSE:45        FALSE:24
 p210     : 8     NA's: 1   NA's   : 4   TRUE :16        TRUE :37
 NA's     :46                           NA's :18        NA's :18

 transplant          f.u            date last seen
 Mode :logical   Length:79          Length:79
 FALSE:53        Class :character   Class :character
 TRUE :8         Mode  :character   Mode  :character
 NA's :18
```

Fig. 5.4 – Fenotipul asociat datelor ALL de tip B, cu BCR/ABL pozitiv și negativ

Experimentele prezentate în acest capitol, au ca finalitate evaluarea metodei propuse pentru selectarea atributelor în datele ADN microarray. Setul de date ALL, este analizat cu finalitatea selectării unui subgrup restrâns de gene diferențial exprimate, între pacienți cu clasificare BCR/ABL pozitivă sau negativă, care pot discrimina între cele două categorii.

5.2. Evaluarea dominanței incomplete

Pentru evaluarea schemei dominanței incomplete (DI) descrise în capitolul 3.3 în contextul unui algoritm genetic diploid descris în capitolul 3.2 am analizat cele trei seturi de date pentru selectarea atributelor principale în discriminarea dintre clasele `BCR/ABL` negativ și pozitiv.

Clasificatorul supervizat utilizat pentru evaluarea adaptabilității indivizilor este *kNN*. Am ales să utilizăm *kNN* deoarece reprezintă o metodă rapidă și transparentă de clasificare, care permite evaluarea facilă a performanței algoritmului genetic în acest context. Clasificarea cu *kNN* este ușor interpretabilă în raport cu importanța atributelor. Funcția utilizată în algoritmul genetic a fost implementarea `knn.cvI` din pachetul **MLInterfaces**. Această funcție este optimizată pentru validarea încrucișată și testele noastre au demonstrat o îmbunătățire semnificativă în evaluarea populației cu această implementare, din punct de vedere al timpului de execuție. Am utilizat valoarea `k=8` pentru kNN, iar decizia s-a luat cu votul majoritar a `l=5` dintre vecinii apropiați. Aceste valori pentru *k* și *l* au fost determinate empiric

pentru setul de date ALL. Am utilizat această metodă, `knn.cvI(k=8, l=5)`, în toate experimentele din acest capitol pentru a sprijini concluziile în privința performanței operatorilor genetici prin uniformitatea evaluării adaptabilității.

Algoritmii genetici au fost inițializați în mod diferit pe fiecare dintre cele trei seturi de date considerate. Particularitățile la inițializare pentru fiecare dintre cele trei seturi de date analizate sunt ilustrate în tabelul 5.1. Populații mai numeroase și gene mai multe în fiecare genom au fost utilizate în funcție de particularitățile fiecărui subset. Condiția de terminare a fost aceeași în toate situațiile, 500 de generații de evoluție. De asemenea, valori uniforme peste toate seturile de date pentru elitism și rata mutațiilor punctuale au fost stabilite empiric la valorile 5% și respectiv 0.005.

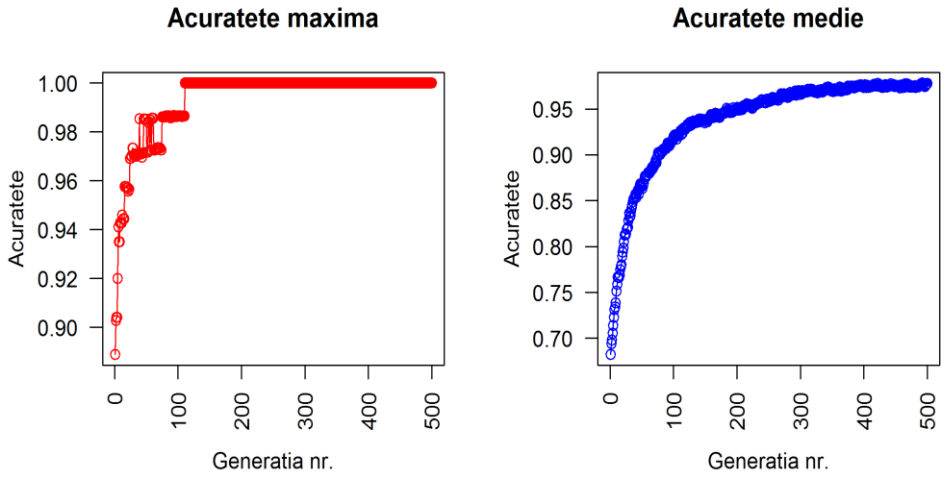
Am realizat 20 de replicații în acest cadrul experimental cu fiecare set de date analizat și am interpretat rezultatele obținute. O primă observație la inspecția reprezentărilor grafice ale evoluțiilor a fost că dominanța incompletă reprezintă o abordare care favorizează evoluția. Fiecare dintre experimentele efectuate s-a concretizat în evoluții spre sub-seturi de atribute foarte eficiente în discriminarea dintre cele două clase cu kNN. Evoluții spre clasificatori cu acurateţe de >95% și populații de clasificatori cu media >92% au fost obținute în toate replicările. Aspectul evoluției într-unul dintre experimente este prezentat în Fig. 5.5. Selectarea atributelor cu algoritmul genetic propus a dus la rezultate net superioare în acuratețea discriminării între cele două clase de exemple. Rezultatele obținute cu clasificatorii supervizați pe datele cu toate atributele au fost adesea nesatisfăcătoare. Acurateţe de 82% a fost obținută cu clasificatorul SVM, dar valori de 100%, cum au fost adesea atinse cu algoritmul nostru sunt marcat ameliorate.

Analiza atributelor cel mai frecvent selectate în fiecare scenariu analizat evidențiază în plus eficiența metodei propuse. Cel mai frecvent selecționate cinci probe din fiecare set de date analizat sunt prezentate în tabelul 5.2.

Între atributele cel mai frecvent selectate de algoritmul genetic drept semnificative în discriminarea dintre clasele BCR/ABL pozitiv și negativ apar adesea probe reprezentând "ABL proto-oncogene 1, non-receptor tyrosine kinase". Acest rezultat sugerează o legătură puternică între cele două clase studiate și nivelul expresiei pentru această genă. Chip-urile comerciale utilizează copii multiple ale aceleiași gene tocmai pentru a oferi o măsură a calității măsurătorilor și rezultatelor obținute. Selectarea a mai multe copii ale aceleiași probe subliniază eficiența AG în selectarea atributelor din datele de ADN microarray. Anumite anomalii ale genei ABL au fost implicate în etiologia diferitelor tipuri de leucemie [98, 99] în ultimii 20 de ani și reprezintă motivul utilizării testării BCR/ABL în evaluarea pacienților de leucemie.

Tabel 5.1 - Inițializare AG pe diferitele seturi de date.

ID Set	Lungimea Genomului (nr. atribute)	Gene active	Indivizi în populație	Clasificatori kNN
1	12625	30	500	1000
2	2391	20	200	400
3	628	12	100	200



a) b)

Fig. 5.5 – Evoluția AG a) Evoluția acurateței maxime în 500 generații; b) Evoluția acurateței medii în 500 generații.

Tabel 5.2 - Cel mai frecvent selecționate cinci probe din fiecare set

Nr.	Setul de date cu **12625** de probe		Setul de date cu **2391** de probe		Setul de date cu **628** de probe	
	Affymetrix ID	**Gene ID**	**Affymetrix ID**	**Gene ID**	**Affymetrix ID**	**Gene ID**
1	$`39730_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"	$`39730_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"	$`39730_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"
2	$`1635_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"	$`37027_at`	"AHNAK nucleoprotein"	$`38052_at`	"coagulation factor XIII A chain"
3	$`1636_g_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"	$`38052_at`	"coagulation factor XIII A chain"	$`39338_at`	"S100 calcium binding protein A10"
4	$`38052_at`	"coagulation factor XIII A chain"	$`33440_at`	"zinc finger E-box binding homeobox 1"	$`40480_s_at`	"FYN proto-oncogene, Src family tyrosine kinase"
5	$`38968_at`	"SH3-domain binding protein 5"	$`1635_at `	"ABL proto-oncogene 1, non-receptor tyrosine kinase"	$`1636_g_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"

De asemenea, observăm că setul de date cu 12625 de probe surprinde trei copii ale aceleiași probe, în comparație cu sub-seturile filtrate. În ansamblu, rezultatele obținute cu setul de date complet au fost mai consistente pe parcursul diferitelor replicări. Suprapunerea rezultatelor a fost foarte puternică la analiza celor mai selectate 5 atribute. Când sunt considerate cele mai semnificative 30 de probe descoperite de algoritmul genetic în fiecare dintre cele trei seturi de date, intersecția soluțiilor pierde din consistență. Acest aspect este absolut normal și nu afectează utilitatea metodei, din moment ce ne propunem selectarea unui subset restrâns dintre probe, pentru analiză și validare biologică succesivă.

Prezența probei `38968_at`, reprezentând "SH3-domain binding protein 5", poate fi de asemenea, semnificativă. Alterarea domeniului SH3 al genei "ABL proto-oncogene 1, non-receptor tyrosine kinase" este asociată cu efecte oncogene, dar interpretarea acestui rezultat în situația dată, depășește scopul și posibilitățile studiului de față.

Proba cu identificarea `38052_at`, reprezentând "coagulation factor XIII A chain" a fost deopotrivă selectată de algoritmul genetic propus. Această genă a fost implicată în etiologia unor tipuri de leucemie acută, dar nu a fost asociată cu clasificarea biologică BCR/ABL, iar apariția ei în rezultatele noastre necesită interpretare biologică.

În general, pe parcursul discutării experimentelor efectuate pentru a testa eficiența algoritmului propus, ne limităm la a raporta rezultate și a observa asociații. Interpretarea biologică a acestor rezultate necesită specializări și experiențe în domenii diferite și nu reprezintă obiectivul studiului nostru. Este motivul pentru care, întotdeauna, studiile de tip ADN microarray sunt efectuate de echipe multi-disciplinare, în care sunt implicați cercetători cu pregătire diferită. Fără o astfel de abordare multi-disciplinară, o relație cauzală între probe descoperite prin analiza datelor microarray și o anumită patologie nu pot fi stabilite.

Experimentele prezentate susțin ideea oportunității modelării dominanței incomplete în implementarea algoritmilor genetici diploizi. Algoritmul genetic utilizând acest model pentru maparea genotipului la fenotip evoluează rapid spre soluții satisfăcătoare. În plus, necesitatea de-a defini o schemă de dominare complexă și particulară unui cadrul experimental, dispare în abordarea propusă de noi. Simplitatea și flexibilitatea modelului îl face aplicabil unei palete variate de probleme în care se impune selectarea atributelor din seturi de date foarte vaste, depășind spectrul analizei datelor microarray. Algoritmul necesită resurse hardware foarte accesibile, iar timpii de executare sunt acceptabili. De asemenea, flexibilitatea în alegerea clasificatorului supervizat și facilitatea utilizării multiplelor implementări accesibile în Bioconductor pot răspunde unei diversități de contexte experimentale.

Cele două neajunsuri majore ale algoritmului genetic propus în varianta prezentată sunt timpul de execuție, afectat și de necesitatea multiplelor replicări ale unui experiment și tendința AG de-a eșua într-un optim local. Pe parcursul elaborării tezei de doctorat am adresat cele două aspecte, câteva dintre soluțiile dezvoltate în acest scop sunt testate în continuare.

5.3. Evaluarea dominanței incomplete versiunea 2

Testele din subcapitolul precedent au evaluat oportunitatea modelării principiului dominanței incomplete în proiectarea unui algoritm genetic diploid pentru selectarea atributelor în analiza datelor microarray. Rezultatele obținute afirmă eficiența acestei implementări și confirmă abordarea noastră în

implementarea unui algoritm genetic diploid fără a mai fi necesară definirea explicită a unei scheme de dominare particulare pentru maparea genotipului la fenotip. Impactul implicit asupra explorării și exploatării cu metoda propusă, rezidă din generarea aleatorie a populațiilor succesive și eliminarea din fiecare individ a genotipului mai puțin adaptat, după fiecare iterație. Aceste particularități susțin prezența unor genotipuri mai puțin adaptate în generațiile târzii și prin urmare, avantajează explorarea.

Un avantaj major al implementării diploide constă și în posibilitatea implementării unor operatori, modelați după evoluția naturală, cu impact potențial asupra explorării și exploatării. Ne-a interesat în mod special susținerea capacității algoritmului de a părăsi un optim local și a continua căutarea unei soluții noi.

Operatorul pentru elitism oferă o șansă majoră de-a adresa această provocare. În aplicarea elitismului, putem considera adaptabilitatea unui genotip sau a individului pentru ordonarea după performanță. Selectarea celor mai performante genotipuri pentru a fi păstrate în generația viitoare avantajează exploatarea. Perpetuarea în iterația următoare a genotipurilor care au făcut parte din cei mai adaptați indivizi, poate susține explorarea cu AG. Fiecare populație este generată aleatoriu. În consecință, este de așteptat ca genotipuri foarte adaptate să facă parte din indivizi mai puțin adaptați, datorită setului de cromozomi complementari. Această abordare urmează în mod natural implementării dominanței incomplete și promovează în generația următoare genotipuri performante, dar nu neapărat cele mai adaptate, și recombinări între perechile de seturi de cromozomi din indivizii cei mai competitivi.

În subcapitolul anterior am utilizat implementarea clasică a elitismului, mai apropiată de AG haploizi, în care cele mai adaptate fenotipuri erau promovate în generația următoare. Ne propunem să testăm impactul asupra evoluției obținut prin abordarea elitismului la nivelul individului, în comparație cu evaluarea genotipurilor. Algoritmul genetic diploid conceput cu dominanță incompletă evoluează cu sau fără un operator pentru elitism, datorită eliminării implicite a genotipului mai puțin adaptat din fiecare individ după evaluare și al recombinărilor succesive. Așadar, o selecție este intrinsecă în modelul propus. Am decis totuși să numim algoritmul genetic care beneficiază de elitismul la nivelul individului AG cu dominanță incompletă versiunea 2 (DI2), pentru a sublinia această flexibilitate. În realitate, este același algoritm cu dominanță incompletă și abordare diferită a operatorului pentru elitism. Pentru claritate ne vom referi la implementarea inițială a dominanței incomplete în algoritmul genetic diploid propus cu denumirea de versiunea 1 (DI1).

Pentru a obține o imagine acurată a efectelor asupra explorării și exploatării obținute prin elitismul la nivelul individului, vom testa multiplu cele două abordări pe seturi de date identice, pornind de la populații inițiale identice, generate cu același random seed. Am testat evoluțiile pe subsetul de date ALL cu 79 exemple, filtrat specific și nespecific la 628 de atribute. Șansa ca o mutație să apară a fost anulată, pentru a elimina aceste efecte asupra explorării și a obține o evaluare mai corectă a impactului obținut prin utilizarea celor două tipuri de operatori pentru elitism. De asemenea, am utilizat valoarea de 2% pentru elitism față de 5% considerată anterior. Rezultatele obținute cu dominanță incompletă versiunea 2 și implementarea inițială, cu populații generate din același random seed, după 400 de generații, sunt prezentate în Fig. 5.6 și respectiv 5.7.

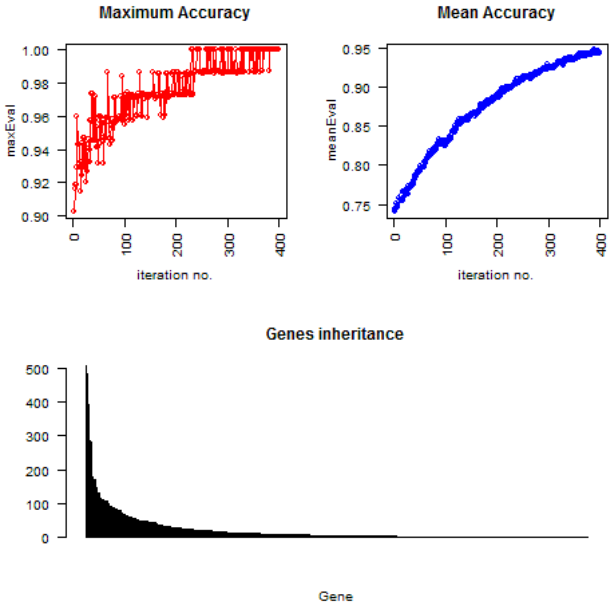


Fig. 5.6 – Rezultatele obținute cu dominanță incompletă versiunea 2.

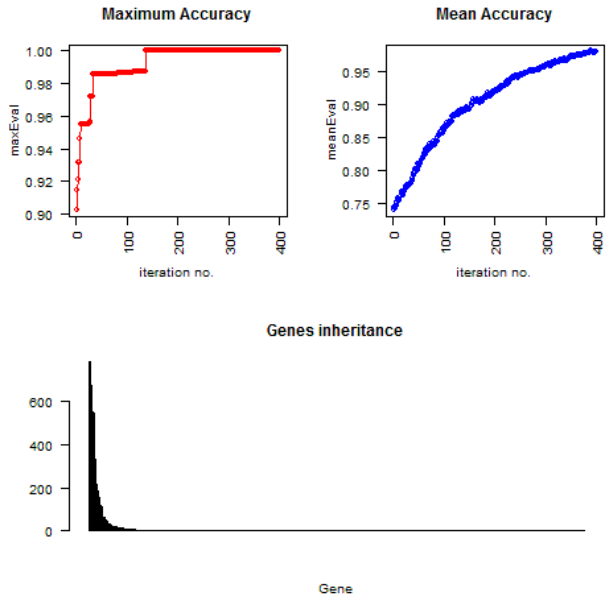


Fig. 5.7 – Rezultatele obținute cu dominanță incompletă versiunea 1.

Inspecția figurilor 5.6 și 5.7 relevă diferențe semnificative între cele două abordări. În mod evident, explorarea este avantajată de versiunea 2, iar AG părăsește cu ușurință un optim local și continuă evoluția fără ca acuratețea medie în populație să fie afectată. Acest aspect este foarte dezirabil în experimentul cu date microarray. Acest comportament este de asemenea indicat de aspectul genelor cel mai frecvent selectate. Un număr mai mare de gene sunt considerate în timpul evoluției cu versiunea 2, dar în continuare, un subgrup restrâns se remarcă prin frecvența transmiterii în generația următoare. Rezultatele comparative în privința celor mai frecvent selectate 5 atribute sunt prezentate în tabelul 5.3. Se observă diferențe semnificative față de rezultatele obținute cu setul de date conținând 628 de atribute în subcapitolul anterior în tabelul 5.2. Aceste diferențe se datorează eliminării șansei ca mutații punctuale să apară, a utilizării valorii 2% pentru elitism față de 5% și a generării populației inițiale din random seed-uri diferite. În mod evident, pornind de la aceeași populație inițială versiunea 2 se dovedește superioară predecesoarei în privința consistenței rezultatelor obținute.

O imagine sugestivă a diferenței în evoluția celor două implementări poate fi obținută prin vizualizarea evoluțiilor acurateței medii (Fig. 5.8) și maxime (Fig. 5.9) în populațiile generate din random seed-uri identice. Examinarea comportamentului AG cu DI1 și DI2 relevă un comportament constant pe toate populațiile inițiale. Se remarcă ușurința cu care AG părăsește un optim local în implementarea DI2, cu prețul evoluției mai lente a acurateței medii în populație față de DI1.

Interpretarea diferențelor dintre cele două abordări trebuie abordată coroborând observațiile cu privire la evoluția algoritmilor și rezultatele obținute. Deși este evident avantajul în privința evoluției acurateței medii în implementarea DI1 față de DI2, rezultatele mai consistente și uniforme obținute prin replicarea experimentelor pe diferitele populații inițiale înclină balanța în avantajul DI2. Ușurința cu care AG părăsește un maxim local, fără ca acuratețea medie să sufere semnificativ este un argument în plus pentru utilizarea DI2. În mod evident, implementarea DI2 avantajează explorarea pe seama exploatării, dar în contextul dat, al selectării atributelor în datele de ADN microarray, oferă un raport profitabil între cele două laturi ale căutării.

Tabel 5.3 - Cel mai frecvent selecționate cinci probe din fiecare set

Nr.	Setul de date cu **628** de probe și dominanță incompletă versiunea 1		Setul de date cu **628** de probe și dominanță incompletă versiunea 2	
	Affymetrix ID	**Gene ID**	**Affymetrix ID**	**Gene ID**
1	$`1635_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"	$`39730_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"
2	$`40480_s_at`	"FYN proto-oncogene, Src family tyrosine kinase"	$`1636_g_at `	"ABL proto-oncogene 1, non-receptor tyrosine kinase"
3	$`675_at`	"interferon induced transmembrane protein 1"	$`38385_at `	"destrin (actin depolymerizing factor)"
4	$`39338_at`	"S100 calcium binding protein A10"	$`1635_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"
5	$`38671_at`	"plexin D1"	$`40480_s_at`	"FYN proto-oncogene, Src family tyrosine kinase"

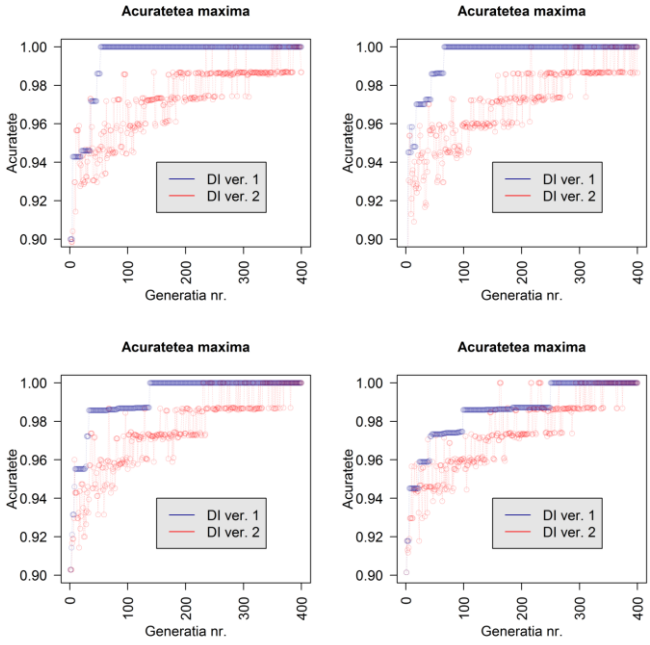


Fig. 5.8 – Evoluția comparativă acurateței maxime pe patru populații inițiale cu DI1 și DI2.

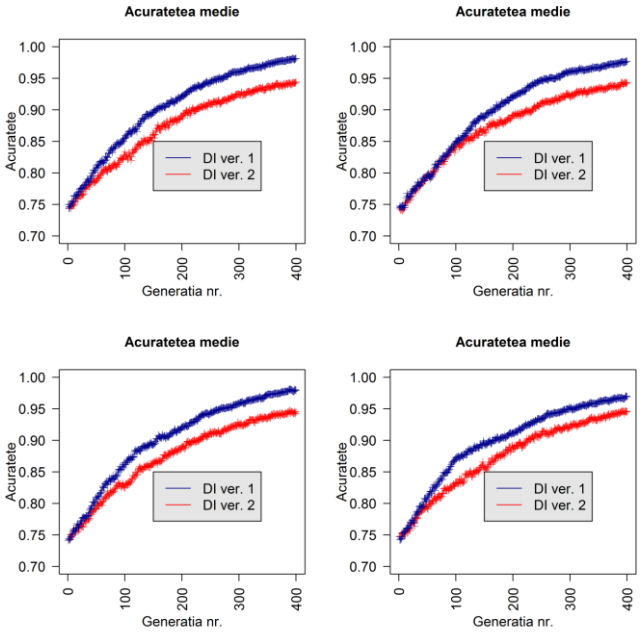


Fig. 5.9 – Evoluția comparativă acurateței medii pe patru populații inițiale cu DI1 și DI2.

Este de așteptat ca sub-grupuri diferite de gene diferențial exprimate să conducă la discriminarea perfectă între cele două clase. Genele diferențial exprimate pot fi consecința altor aspecte din setul de date sau al unor fenomene secundare celui analizat. Analiza cu metodele descoperirii formelor nu urmărește stabilirea unei relații cauzale între un sub-grup dintre atribute și clasele considerate, ci determinarea unui grup de atribute semnificative pentru validare biologică succesivă. Așadar, o abordare care susține explorarea este mai potrivită în acest caz.

5.4. Evaluarea operatorului pentru atribuirea aleatorie a cromozomilor

Ne propunem să testăm operatorul pentru atribuirea aleatorie a cromozomilor (AAC) introdus în capitolul 3.4.2. Pentru a putea implementa operatorul descris, modelat după fenomenele care au loc în timpul meiozei, o condiție necesară o reprezintă separarea genotipurilor într-un număr variabil, în seturi, de cromozomi, așa cum se întâmplă în natură. Operatorul conceput în timpul cercetărilor noastre separă fiecare genom într-un număr de cromozomi, la latitudinea cercetătorului. Mai presus de fundamentarea cadrului în care modelarea atribuirii aleatorii a cromozomilor are loc în meioză, numărul cromozomilor după care se face separarea are un impact major în efectele operatorului. Deoarece o recombinare are loc la nivelul fiecărui cromozom și nu este implementat un argument pentru a stabili șansa ca o operațiune de acest fel să aibă loc, numărul cromozomilor după care se face distribuirea, determină în mod direct și numărul recombinărilor efectuate.

Pentru a obține o imagine cuprinzătoare a impactului operatorului pentru atribuirea aleatorie a cromozomilor am decis să testăm diferite configurații de seturi de cromozomi pe subsetul ALL cu 79 de exemple și 628 de atribute. Ne-am îndreptat atenția asupra a 3 scenarii cu seturi de cromozomi, obținute prin distribuția genelor pe unul, cinci sau douăzeci și doi de cromozomi. Configurația cu un cromozom reprezintă situațiile tratate până acum și este considerată pentru comparație. Situația cu 22 de cromozomi reprezintă situația implementată standard în AG propus. Tabelul 3.1 surprinde regula după care distribuția genelor pe cromozomi are loc și prezice numărul de gene distribuite pe fiecare cromozom într-un context dat. Numărul atributelor distribuite pe fiecare cromozom în situațiile menționate este prezentat în tabelul 5.4.

Algoritmul genetic utilizat pentru testare beneficiază de dominanța incompletă versiunea 1, cu elitism la nivelul seturilor de cromozomi. Versiunea DI2, deși preferată de noi în abordarea datelor de ADN microarray tocmai pentru că favorizează explorarea, este mai dificil de utilizat pentru finalitatea determinării impactului operatorului de atribuire aleatorie a cromozomilor. AAC este conceput pentru a susține evoluția prin sporirea explorării, așadar DI1 oferă un cadru propice pentru a evalua acest impact independent de efectele ID2. Pentru a stimula exploatarea și a evidenția mai clar efectele AAC asupra explorării, valoarea aleasă pentru elitismul de tip ID1 a fost 5% în experimentele noastre.

Tabel 5.4 Distribuția atributelor în scenariile considerate

Nr. cromozom	22 de cromozomi (nr. gene)	5 cromozomi (nr. gene)	1 cromozom (nr. gene)
1	57	176	628
2	48	146	0
3	36	111	0
4	30	93	0
5	32	102	0
6	36	0	0
7	34	0	0
8	26	0	0
9	26	0	0
10	26	0	0
11	38	0	0
12	30	0	0
13	15	0	0
14	23	0	0
15	23	0	0
16	24	0	0
17	30	0	0
18	11	0	0
19	32	0	0
20	17	0	0
21	7	0	0
22	27	0	0

Algoritmii genetici testați au beneficiat de aceeași metodă de evaluare a adaptabilității indivizilor, `knn.cvI(k=8, l=5)`, în context de validare încrucișată leave-one-out și aceeași șansă ca o mutație punctuală să se producă, 0.005. Rezultatele au fost evaluate după ce fiecare algoritm a evoluat pe parcursul a 500 de generații. Cele trei scenarii considerate, cu unul, cinci și douăzeci și doi de cromozomi, au fost evaluate după 20 de replicări, pornind de la 20 de random seed-uri prestabilite și prin urmare 20 de populații inițiale diferite. Aceleași random seed-uri și respectiv populații inițiale au fost testate în fiecare dintre cele trei scenarii. Populațiile inițiale au constat din 200 de indivizi generați fortuit, cu genotipuri conținând 628 de gene și 12 gene activate în fiecare set de cromozomi.

Rezultate obținute în 4 dintre cele 20 de populații inițiale tratate sunt ilustrate în Fig. 5.10 și 5.11. Evoluțiile în cele trei scenarii sunt reprezentate cu culori diferite: roșu, albastru și verde pentru situațiile cu 22, 5 și respectiv 1 cromozom. Am urmărit evoluția acurateței maxime (5.10) în fiecare situație și a acurateței medii (5.11) pe generație. Pe parcursul evoluției, în scenariile cu cinci și douăzeci și doi de cromozomi plus AAC, cel puțin un clasificator cu acuratețe de 100% a fost descoperit în nouăsprezece din cele douăzeci de replicări. În cazul utilizării unui singur cromozom pentru reprezentarea genotipului, în doar jumătate dintre populațiile inițiale au fost găsiți clasificatori cu acuratețe de 100%. În zece dintre cele douăzeci de populații inițiale, am obținut convergență spre soluții cu acuratețe satisfăcătoare, de aproximativ 98.5%, dar inferioară performanței atinse în ambele situații cu cinci și douăzeci și doi de cromozomi și AAC. Așadar, o îmbunătățire a calității evoluției a fost realizată constant prin implementarea AAC. Acuratețea medie a evoluat sensibil diferit. Când un clasificator cu acuratețe maximă de 100% a fost descoperit în contextul cu 1 cromozom, rareori acuratețea medie atinsă a fost ușor superioară valorilor obținute cu AAC. În general, acuratețea medie

a beneficiat de efectele AAC în termeni de valori maxime atinse. În cele mai multe situații, acuratețea medie a evoluat mai rapid până la un anumit nivel în contextul unui singur cromozom, dar a fost depășită în generațiile înaintate de variantele cu AAC. Așa cum era de prevăzut, AAC susține explorarea. Cu cât numărul cromozomilor considerați crește, sporește consecutiv numărul recombinărilor, iar explorarea este facilitată în plus. De asemenea, relația explorare – exploatare este afectată pozitiv de implementarea AAC, cu consecința evoluției spre soluții superioare.

În datele studiate nu am remarcat avantaje în utilizarea a 22 față de 5 cromozomi, cu efectul sporirii numărului recombinărilor. Așa cum se întâmplă cu operatorii pentru recombinări în general, alegerea ratei ideale a recombinărilor depinde de setul de date analizat și necesită o activitate suplimentară, a priori desfășurării experimentelor propriu-zise.

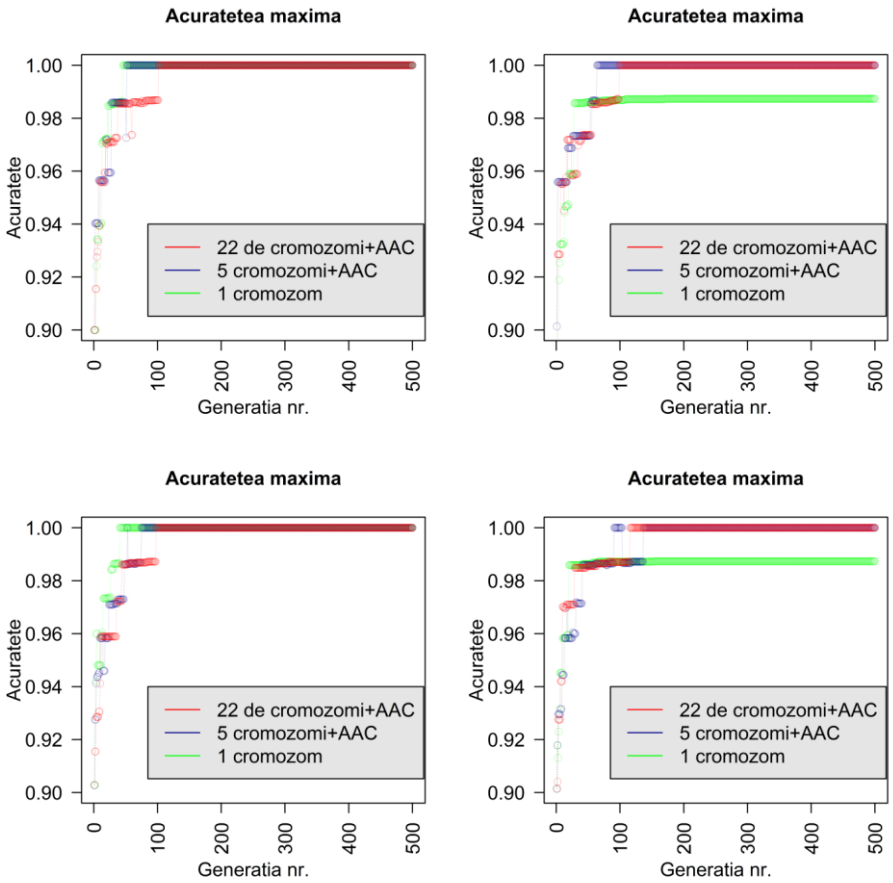


Fig. 5.10 – Evoluția acurateței maxime pe patru populații inițiale.

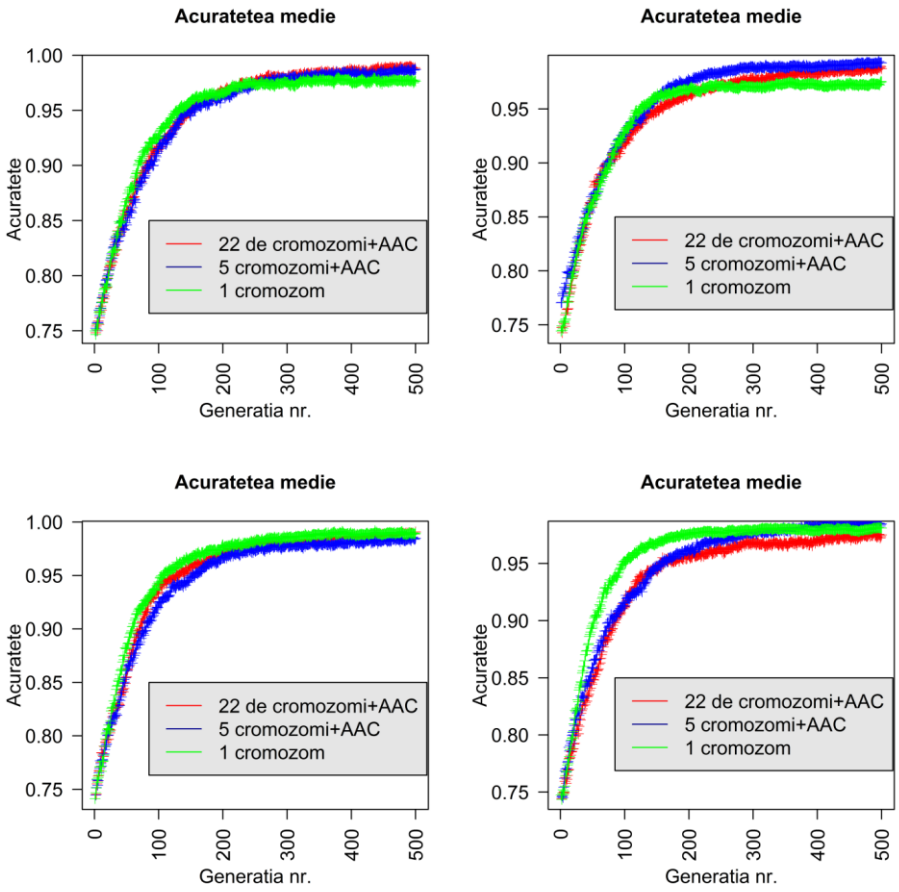
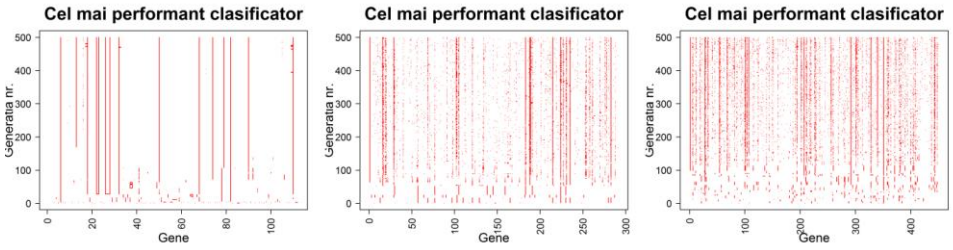


Fig. 5.11 – Evoluția acurateței medii pe patru populații inițiale.

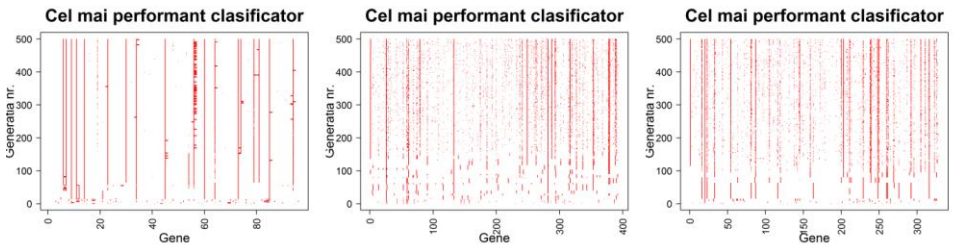
Informații suplimentare despre calitatea evoluției în cele trei scenarii pot fi obținute prin inspecția clasificatorilor cei mai performanți în fiecare generație. Pentru a caracteriza acești clasificatori, ne-am îndreptat atenția asupra atributelor utilizate pentru clasificare de către cel mai acurat clasificator per generație, în fiecare dintre cele trei scenarii analizate. Deoarece ne așteptăm să reprezinte mai sugestiv caracteristicile evoluțiilor, ne-am concentrat în special pe populații inițiale în care algoritmii beneficiind de AAC au descoperit clasificatori cu acurateţe de 100%, iar algoritmul fără AAC a returnat soluții cu acurateţe inferioară. Ilustrăm în Fig. 5.12 și Fig. 5.13 clasificatorii cei mai adaptați pe parcursul celor 500 de generații, în fiecare dintre cele trei contexte experimentale, pentru două populații inițiale, generate cu SEED2 și respectiv SEED4. Evoluțiile corespunzătoare ale acurateței maxime și medii, pentru aceleași populații inițiale, sunt prezentate pe coloanele din dreapta ale Fig. 5.10 și 5.11. Atributele care nu au fost niciodată selectate în cel mai bun clasificator au fost eliminate din grafic pentru o lizibilitate sporită.

Examinarea graficelor din Fig. 5.12 și 5.13 confirmă observațiile anterioare. Variabilitatea genetică menținută în generațiile înaintate este net superioară prin metoda distribuirii genelor pe un număr variabil de cromozomi și aplicarea consecutivă a AAC. Este evidentă tendința AG cu 1 cromozom de-a converge într-un optim local și incapacitatea lui de-a părăsi această configurație pentru a continua căutarea unei soluții alternative. Experimentele cu 5 cromozomi și AAC au surprins capacitatea acestei implementări de-a părăsi un optim local. Totuși, această tendință a fost sporadică și nu poate fi considerată drept o caracteristică în această implementare. Interpretarea evoluției celei mai bune soluții pe parcursul succesiunii generațiilor, coroborat cu evoluțiile acurateței medii și maxime oferă elemente substanțiale în interpretarea genelor cel mai frecvent prezente în iterațiile înaintate.



(a) (b) (c)

Fig. 5.12 - – Evoluția celui mai adaptat clasificator în populația generată cu SEED2. (a) AG cu 1 cromozom, (b) AG cu 5 cromozomi și AAC, (c) AG cu 22 de cromozomi și AAC.



(a) (b) (c)

Fig. 5.13 - Evoluția celui mai adaptat clasificator în populația generată cu SEED4. (a) AG cu 1 cromozom, (b) AG cu 5 cromozomi și AAC, (c) AG cu 22 de cromozomi și AAC.

Metoda standard de analiză supervizată, acompaniată de selectarea atributelor implementată în MLInterfaces este regularized discriminant analysis (RDA), implementată în pachetul R omonim [96]. Pentru a dobândi o imagine mai consistentă asupra eficienței AG implementat, am comparat rezultatele noastre cu analiza aceluiași set de date cu metoda RDA. După 20 de replicări ale experimentelor, RDA a selectat atributele prezentate în tabelul 5.5. și am comparat aceste rezultate cu genele selectate în cele trei scenarii considerate pentru AG propus de noi (tabelul 5.6).

Tabel 5.5 - Atributele selectate cu metoda RDA

No.	Affymetrix ID	Gene ID
1	$`1211_s_at `	"CASP2 and RIPK1 domain containing adaptor with death domain"
2	$`1635_at `	"ABL proto-oncogene 1, non-receptor tyrosine kinase"
3	$`1636_g_at `	"ABL proto-oncogene 1, non-receptor tyrosine kinase"
4	$`2057_g_at `	"fibroblast growth factor receptor 1"
5	$`32186_at `	"solute carrier family 7 member 5"

Tabel 5.6 - Atributele selectate cu AG 1

Nr.	AG cu 1 cromozom		AG cu 5 cromozomi		AG cu 22 de cromozomi și AAC	
	Affymetrix ID	**Gene ID**	**Affymetrix ID**	**Gene ID**	**Affymetrix ID**	**Gene ID**
1	$`39730_at `	"ABL proto-oncogene 1, non-receptor tyrosine kinase"	$`39730_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"	$`39730_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"
2	$`38052_at `	"coagulation factor XIII A chain"	$`1636_g_at `	"ABL proto-oncogene 1, non-receptor tyrosine kinase"	$`1635_at `	"ABL proto-oncogene 1, non-receptor tyrosine kinase"
3	$`39338_at `	"S100 calcium binding protein A10"	$`39338_at`	"S100 calcium binding protein A10"	$`38052_at`	"coagulation factor XIII A chain"
4	$`40480_s_at`	"FYN proto-oncogene, Src family tyrosine kinase"	$`675_at`	"interferon induced transmembrane protein 1"	$`1635_at `	"ABL proto-oncogene 1, non-receptor tyrosine kinase"
5	$`1636_g_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"	$`39837_s_at`	"zinc finger protein 467"	$`38385_at`	"destrin (actin depolymerizing factor)"

Se observă o suprapunere parțială a rezultatelor obținute prin cele două metode. Cu toate acestea, rezultatele returnate de RDA sunt mult mai puțin consistente între replicări succesive. Adesea am observat că atributele selectate cu metoda RDA nu se suprapun în replicări consecutive, aspect mult mai bine adresat de AG. Acuratețea în discriminarea celor două clase a fost, deopotrivă, net inferioară față de AG, cu valori uzual în jurul acurateței de 85%. De asemenea, se remarcă o prezență constantă și mai bine reprezentată în AG, a copiilor multiple reprezentând probe identice. Varianta cu 22 de cromozomi și AAC a oferit rezultatele cele mai uniforme pe diferitele replicări ale experimentelor, deși evoluțiile acurateței maxime și medii nu au oferit indicii semnificative în acest sens.

5.5. Evaluarea operatorului pentru mutația fără sens

Separarea atributelor pe cromozomi de dimensiuni diferite oferă fundamentul modelării unor modalități de apariție al mutațiilor întâlnite în evoluția naturală. Mutația într-un punct are un impact insuficient în selectarea atributelor din genotipuri foarte lungi, cu un număr limitat de alele *1*. Ne-am propus așadar testarea impactului obținut cu operatorii pentru mutații concepuți după modele din genetică, descriși în capitolul 3. Finalitatea urmărită este determinarea unui operator pentru mutație sau a unei combinații de operatori care poate susține explorarea fără a induce efecte distructive majore asupra exploatării, așa cum se observă la aplicarea mutației într-un punct cu șanse de apariție prea sporite.

Pentru a obține comparații cât mai semnificative, am testat fiecare tip de operator pentru mutație într-un cadru uniform. Pe de o parte, am considerat că diversitatea introdusă cu operatorul de atribuire aleatorie al cromozomilor este foarte dezirabilă și va fi utilizată standard în selectarea atributelor cu AG diploizi din datele de microarray. Așadar, am efectuat testele pentru fiecare tip de operator pe setul de date cu 79 exemple și 628 de atribute distribuite pe 22 de cromozomi, valoarea default în pachetul software implementat și AAC activat. Considerăm că superioritatea abordării DI2 față de DI1 impune această metodă pentru contextul de cercetare considerat. Totuși, am decis să utilizăm DI1 în testele noastre deoarece diversitatea genetică menținută de DI2, deși foarte dezirabilă pentru rezultate substanțiale, ar masca efectele diferitelor tipuri de mutații efectuate. Dacă într-un studiu real, DI2 este fără îndoială metoda preferabilă, testarea de față urmărește evaluarea efectelor obținute prin mutații. În acest sens, versiunea DI1 este de elecție.

Am aplicat un cadru experimental identic pentru toate mutațiile testate. În toate experimentele efectuate, am pornit de la populații inițiale cu 200 de indivizi, generate aleatoriu din random seeds prestabilite, pentru a asigura uniformitatea inițializării și o consistență sporită a rezultatelor. Un număr de 12 atribute a fost activat în populația inițială în toate situațiile. Valoarea utilizată pentru elitismul în context DI1 a fost 2%. Condiția de terminare utilizată a fost aceeași, 400 de generații.

Rezultatele obținute vor fi prezentate în fiecare situație prin comparație cu AG-ul diploid identic specificat, cu singura deosebire constând în tipul mutației aplicate. Considerăm că interpretabilitatea evaluărilor este facilitată prin prezentarea în comparație cu metoda standard, mutația într-un punct. Mutații punctuale cu șansa apariției de 0.05%, s-au concretizat în 62 de operațiuni pe populație la fiecare generație. Am ales o șansă de apariție a mutației la o valoare superioară celei dezirabile într-un studiu real, pentru a vizualiza mai bine efectele mutațiilor studiate. Rezultate comparative din punct de vedere al acurateței maxime și medii, obținute prin aplicarea operatorilor pentru mutația fără sens sau mutația punctuală sunt ilustrate în Fig. 5.14 și respectiv 5.15 pentru patru dintre populațiile inițiale generate aleatoriu.

Efectele mutațiilor fără sens sunt comparabile cu cele ale mutațiilor punctuale în privința impactului asupra explorării cu algoritmul genetic. Evoluțiile valorilor maxime ale acurateței pe fiecare generație (Fig. 5.14) sunt comparabile în cele două abordări. Cu toate acestea, evoluția acurateței medii (Fig. 5.15) pe generație relevă un dezavantaj semnificativ la utilizarea mutației fără sens. Valori

semnificativ mai reduse, cu aproximativ 8%, ale adaptabilității medii în populație au fost observate consecvent pentru populațiile inițiale generate din același random seed, în abordarea cu mutația fără sens.

O imagine suplimentară asupra evoluției AG poate fi obținută prin ilustrarea celui mai adaptat set de cromozomi din fiecare generație. Această reprezentare prezintă evoluția celei mai bune soluții, capacitatea algoritmului de a considera căi noi, variabilitatea genetică în generațiile înaintate, dar și numărul alelelor *1* în iterațiile avansate. Numărul genelor active în setul de cromozomi nu crește pe parcursul evoluției când este utilizată mutația fără sens (Fig. 5.16), clasificatorii utilizând un număr constant de atribute pentru discriminare. În cazul utilizării mutației punctuale, pe măsură ce generațiile înaintează, tot mai multe alele *1* apar (Fig. 5.17) în fiecare cromozom. La finalul căutării, numărul atributelor selectate în fiecare clasificator, este semnificativ crescut față de mulțimea atributelor active în populația inițială. În contextul selecției atributelor, acest aspect pledează împotriva utilizării mutațiilor punctuale. Din acest punct de vedere operatorul pentru mutația fără sens este superior celui pentru mutația într-un punct.

Calitatea evoluției nu este îmbunătățită prin utilizarea operatorului pentru mutația fără sens. Deși adresează problema amplificării numărului atributelor active în clasificatorii din iterațiile înaintate, efectul nociv asupra exploatării nu recomandă mutația fără sens pentru aplicația selecției atributelor în datele ADN microarray.

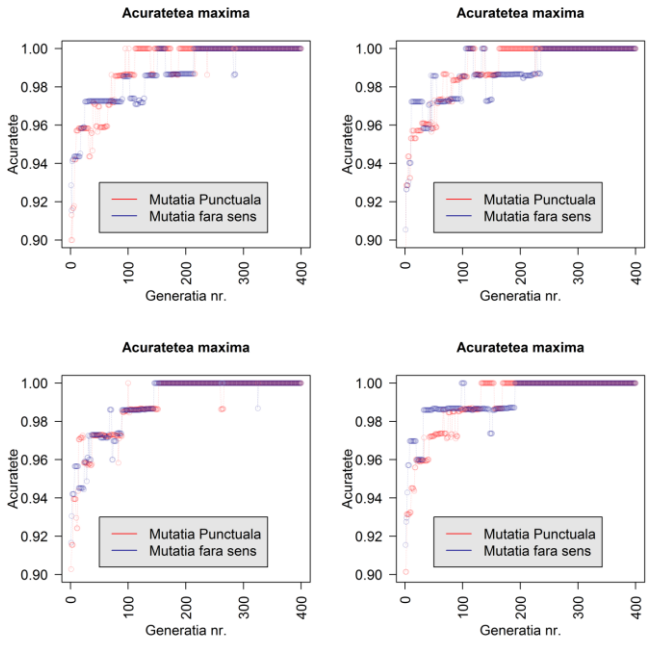


Fig. 5.14 - Evoluția acurateței maxime.

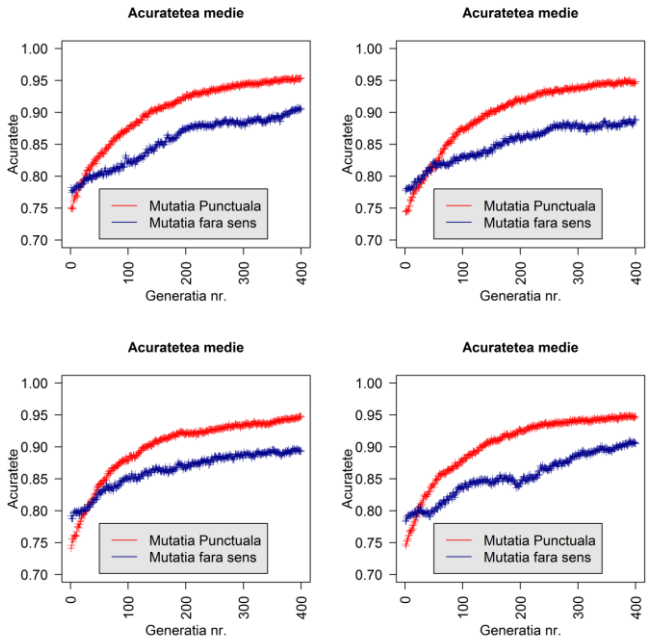


Fig. 5.15 - Evoluția acurateței medii în abordarea cu mutația fără sens în comparație implementarea cu mutația punctuală.

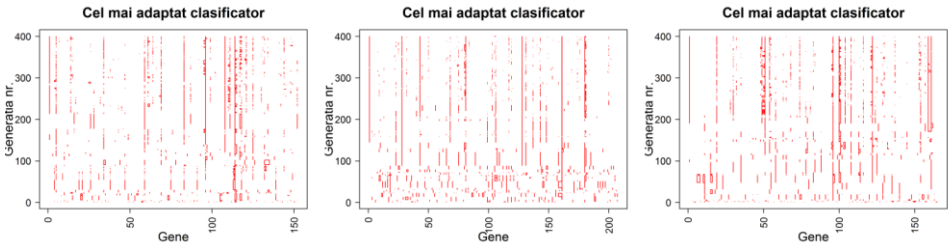


Fig. 5.16 - Evoluția celui mai adaptat clasificator în trei dintre populațiile inițiale, în abordarea cu mutația fără sens.

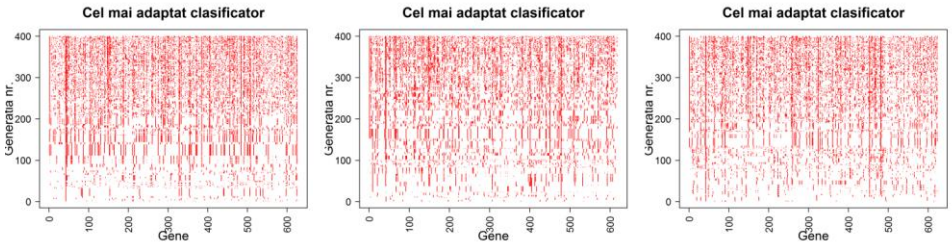


Fig. 5.17 - Evoluția celui mai adaptat clasificator în trei dintre populațiile inițiale, în abordarea cu mutația punctuală.

5.6. Evaluarea operatorului pentru mutația cu deplasare

Teste analoage au fost utilizate pentru evaluarea impactului mutației cu deplasare asupra evoluției. Și în acest caz, am apreciat comparativ contribuțiile mutației într-un punct și cu deplasare. Am utilizat aceleași mijloace de vizualizare pentru a obține o imagine asupra calității evoluției în cele două situații. Rezultatele obținute în timpul revizuirii mutației cu deplasare sunt prezentate în Fig. 5.18-5.19.

În testele noastre, mutația cu deplasare s-a dovedit mai eficientă în a susține tendința AG de a părăsi un optim local și a continua căutarea unei alternative. În unele situații, acuratețea maximă (Fig. 5.18) obținută cu mutația într-un punct a fost superioară optimului găsit prin mutația cu deplasare. De asemenea, media acurateței (Fig. 5.19) în populațiile înaintate, a fost net superioară când a fost utilizat operatorul pentru mutația într-un punct.

Mutația cu deplasare adresează cu succes (Fig. 5.20) neajunsul amplificării alelelor *1* în seturile de cromozomi din populațiile evoluate. Costurile de performanță în privința adaptabilității maxime și medii obținute cu operatorul pentru mutația cu deplasare, nu justifică alegerea lui în defavoarea mutației într-un punct.

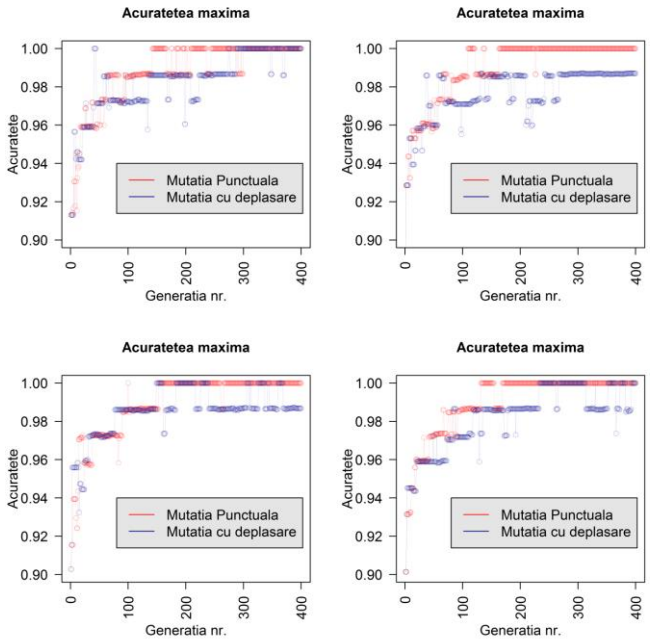


Fig. 5.18 - Evoluția acurateței maxime în abordarea mutație cu deplasare comparativ cu mutația punctuală.

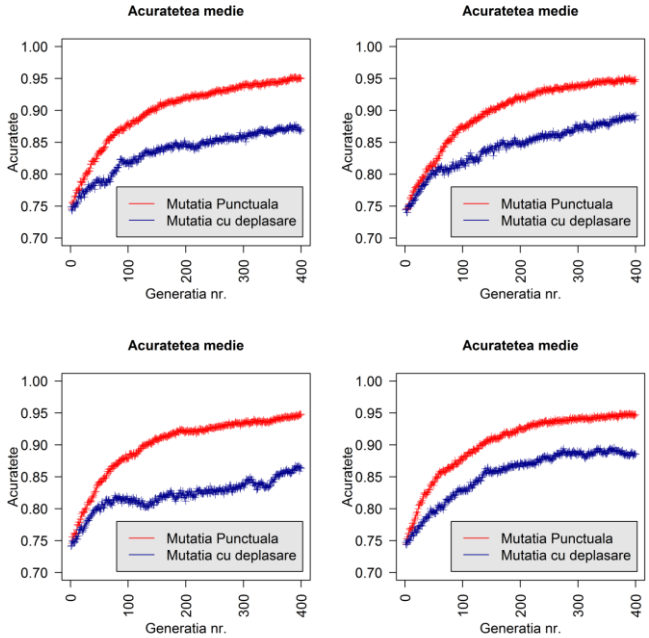


Fig. 5.19 - Evoluția acurateței medii în abordarea mutație cu deplasare comparativ cu mutația punctuală.

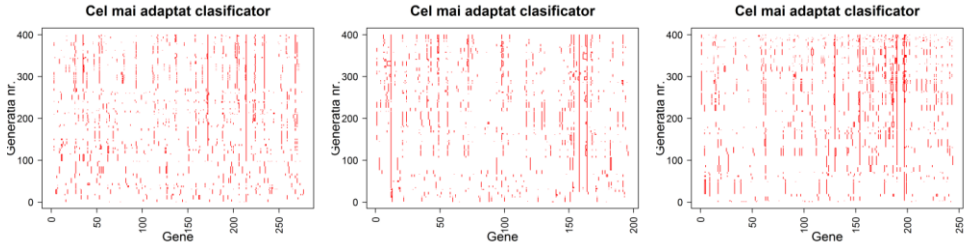


Fig. 5.20 - Evoluția celui mai adaptat clasificator în trei dintre populațiile inițiale, în abordarea mutației cu deplasare.

5.7. Evaluarea operatorului pentru mutația cu ștergerea unui segment

Într-un cadru experimental similar, am testat oportunitatea utilizării operatorului pentru mutație cu ștergerea unui segment de cromozom. Rezultatele comparative în contrast cu mutația într-un punct sunt ilustrate în Fig. 5.21-5.22. Împotriva așteptărilor noastre, mutația cu ștergerea unui segment de cromozom nu a susținut tendința AG de-a părăsi un optim local (Fig. 5.21) mai eficient decât mutația într-un punct. Am remarcat o performanță ușor sporită în privința acurateței medii (Fig. 5.22) în generațiile evoluate, comparativ cu mutația într-un punct. De asemenea nici acest operator nu suferă (Fig. 5.23) de problema majoră a mutației

într-un punct, creșterea numărului atributelor considerate de clasificatorii evaluați în iterațiile înaintate.

Deși operatorul pentru mutația cu ștergerea unui segment de cromozom prezintă avantaje față de mutația clasică, avantajul obținut în explorare nu este suficient de semnificativ pentru a recomanda această abordare pentru selectarea atributelor din datele de ADN microarray.

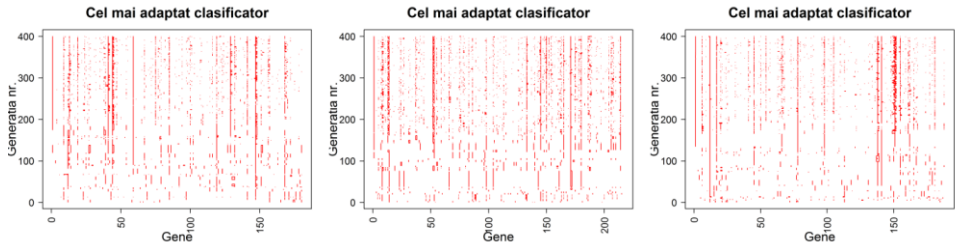


Fig. 5.23 - Evoluția celui mai adaptat clasificator în trei dintre populațiile inițiale, în abordarea mutației cu ștergerea unui segment de cromozom.

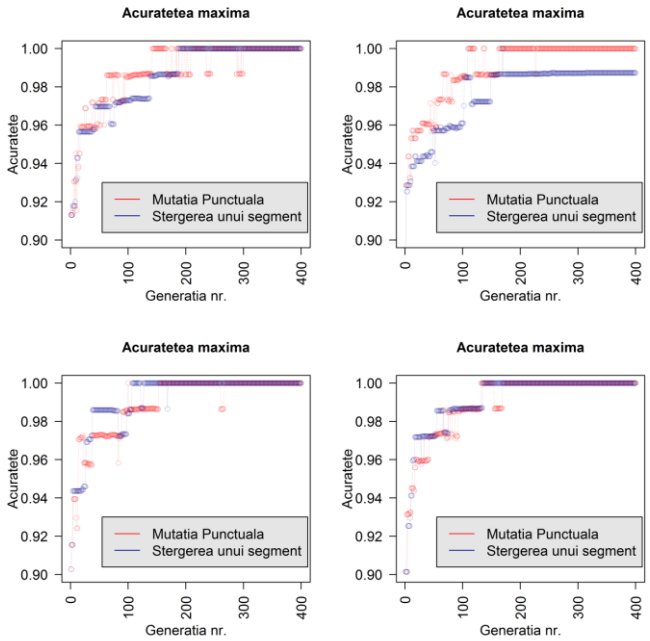


Fig. 5.21 - Evoluția acurateței maxime în abordarea mutație cu ștergerea unui segment de cromozom comparativ cu mutația punctuală.

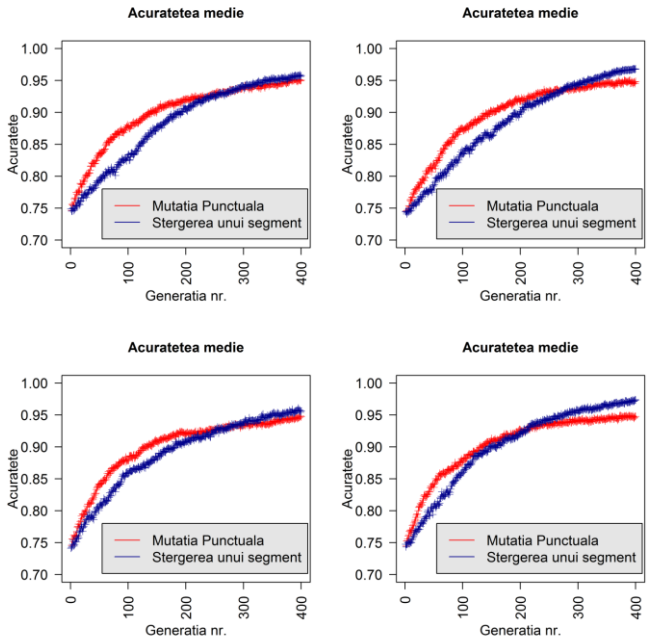


Fig. 5.22 - Evoluția acurateței medii în abordarea mutație cu ștergerea unui segment de cromozom comparativ cu mutația punctuală.

5.8. Evaluarea operatorului pentru mutația cu ștergerea unui cromozom

Calitatea evoluției este influențată pozitiv de operatorul cu ștergerea unui întreg cromozom, comparativ cu situația utilizării mutației într-un singur punct. Tendința de-a părăsi un optim local este deservită în aceeași măsură de cele două variante, iar acuratețea maximă (Fig. 5.24) atinsă în timpul evoluției nu este afectată. De asemenea, am constatat că acuratețea medie (Fig. 5.25) în populațiile evoluate este îmbunătățită sensibil în contextul mutației cu ștergerea unui întreg cromozom. Această abordare nu conduce la creșterea atributelor active (Fig. 5.26) în seturile de cromozomi din generațiile evoluate, așa cum se întâmplă în cazul mutației într-un punct.

Cu toate acestea, o examinare atentă a evoluției clasificatorului cel mai adaptat (Fig. 5.26) relevă tendința AG de-a reveni în același optim local foarte curând după ce o tulburare indusă prin mutația cu ștergerea unui cromozom întreg a avut loc. Deși prin comparație cu mutația clasică, noul operator se dovedește superior, avantajul utilizării lui pentru selecția atributelor în datele de ADN microarray este limitat. Mutația cu ștergerea unui cromozom reușește doar parțial să răspundă provocărilor acestei cercetări.

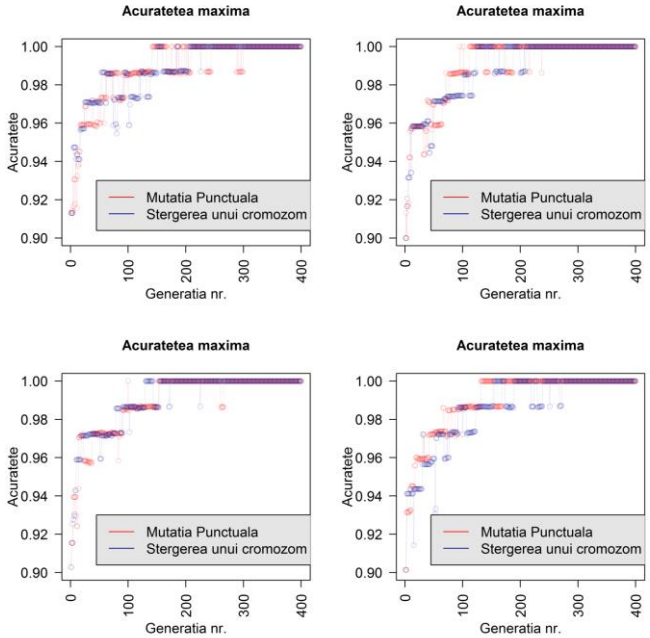


Fig. 5.24 - Evoluția acurateței maxime în abordarea mutație cu ștergerea unui întreg cromozom comparativ cu mutația punctuală.

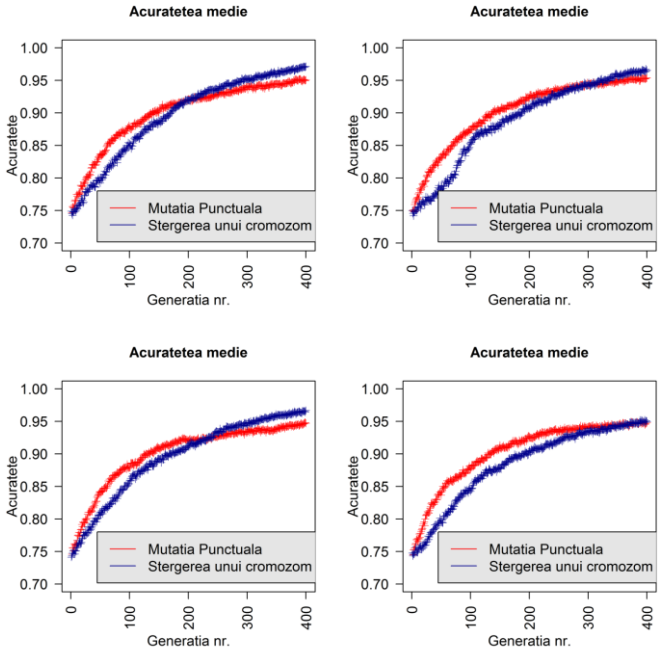


Fig. 5.25 - Evoluția acurateței medii în abordarea mutație cu ștergerea unui întreg cromozom comparativ cu mutația punctuală.

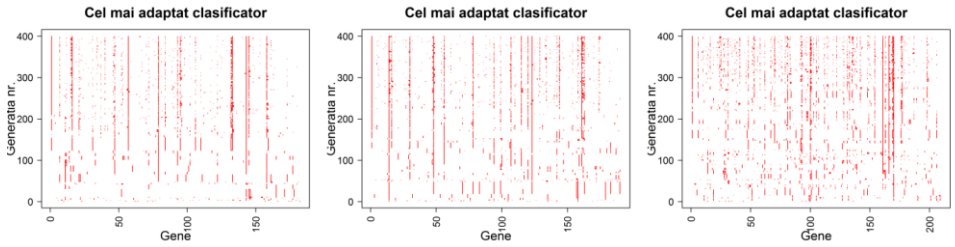


Fig. 5.26 - Evoluția celui mai adaptat clasificator în trei dintre populațiile inițiale, în abordarea mutației cu ștergerea unui întreg cromozom.

5.9. Evaluarea operatorului pentru transpozoni

Aspectul calității evoluției la utilizarea operatorului pentru transpozoni este zugrăvit de ansamblul ilustrațiilor din Fig. 5.27-5.29. Această propunere, deservește în mod superior explorarea cu algoritmul genetic propus, în comparație cu mutația într-un punct. Deși capacitatea AG de-a părăsi un optim local este doar parțial afectată, evoluțiile acurateței maxime și medii sunt susținute de această abordare, iar numărul genelor active în populațiile înaintate nu crește semnificativ, așa cum se întâmplă în cazul mutației într-un singur punct.

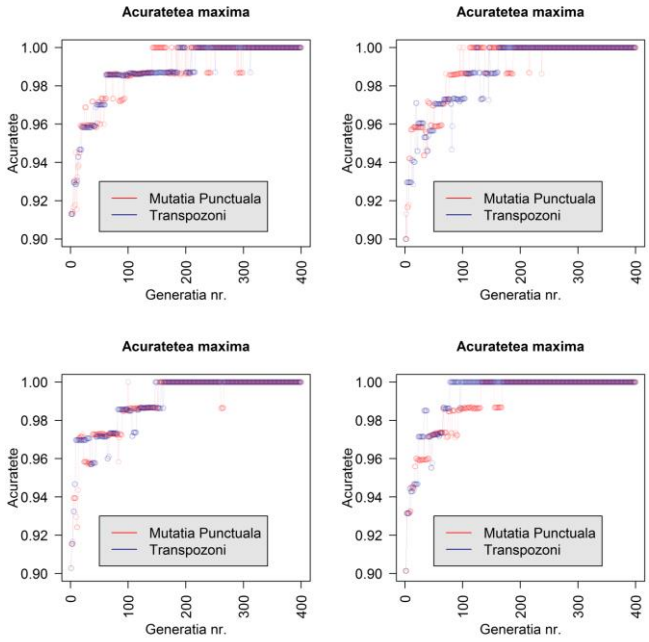


Fig. 5.27 - Evoluția acurateței maxime la utilizarea transpozonilor comparativ cu mutația punctuală.

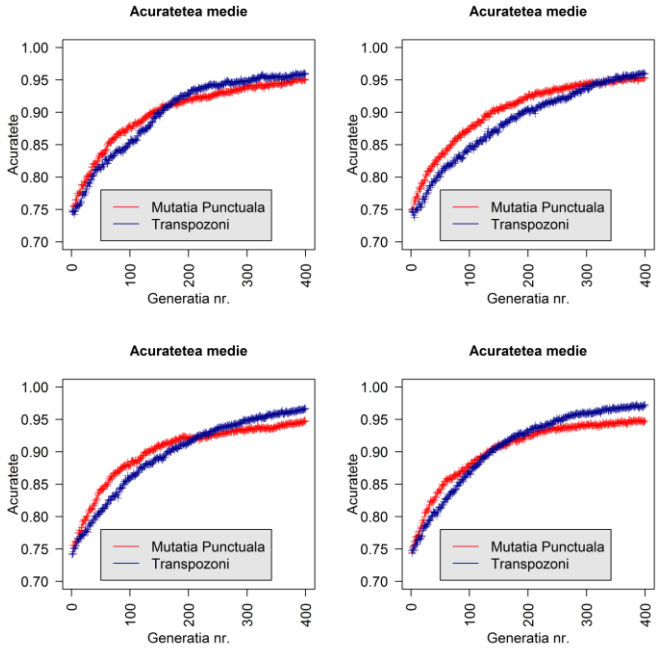


Fig. 5.28 - Evoluția acurateței medii la utilizarea transpozonilor comparativ cu mutația punctuală.

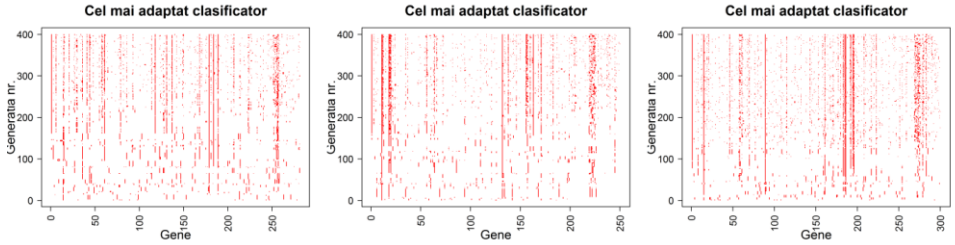


Fig. 5.29 - Evoluția celui mai adaptat clasificator în trei dintre populațiile inițiale, la utilizarea transpozonilor.

5.10. Evaluarea efectelor cumulate ale DI2 și AAC

Testele efectuate pentru evaluarea metodelor propuse, dominanța incompletă în cele două versiuni considerate, operatorul pentru atribuirea aleatorie a cromozomilor și operatorii pentru mutație modelați după fenomene naturale au relevat proprietăți și impact diferit asupra selectării atributelor cu algoritmii genetici.

Argumente pro- și contra- utilizării lor în acest context au fost determinate pentru majoritatea operatorilor pentru mutații. Efecte dezirabile au fost observate în cazurile operatorilor pentru transpozoni și al ștergerii unui întreg cromozom. Cu toate acestea, efectele lor depind semnificativ de șansa ca o mutație să apară, care trebuie determinată empiric. Nu putem concluziona despre nici unul dintre operatorii pentru mutație propuși că este recomandabilă utilizarea lor pentru selecția

atributelor în orice set de date microarray. Putem îndemna la testarea lor în contextul unui experiment similar, pentru evaluarea a priori a efectelor asupra calității evoluției.

Concluziile noastre sunt diferite în raport cu implementarea dominanței incomplete și a operatorului pentru atribuirea aleatorie a cromozomilor. Am remarcat un impact pozitiv important pentru creșterea calității evoluției cu AG pentru selectarea atributelor în ambele cazuri. În special varianta DI2 a dominanței incomplete, a demonstrat caracteristici foarte dezirabile pentru finalitatea propusă. Utilizarea DI2 și AAC au dovedit proprietăți care le recomandă fără echivoc pentru selectarea atributelor din datele de ADN microarray.

Prin urmare, ne propunem să testăm efectul combinat al celor două abordări, DI2 și AAC, pentru selectarea atributelor cu AG. Vom utiliza același context experimental ca și în cazul testării mutațiilor, AG cu specificații identice cu o excepție: șansa ca o mutație de orice tip să apară a fost anulată. În acest cadru, testăm doi algoritmi genetici, unul implementat cu DI2 și AAC, iar celălalt cu DI1 și AAC.

După 400 de iterații, patru dintre populațiile inițiale din cele 20 de replicări ale experimentelor sunt ilustrate în Fig. 5.30 și Fig. 5.31. Evoluțiile acurateței maxime (Fig. 5.30) relevă două aspecte extrem de importante. Pe de o parte, algoritmul cu DI2 și AAC prezintă tendința de-a părăsi un optim local, proprietate observată ocazional la AG cu DI1 și AAC. Implementarea cu AAC și DI1 reușește în marea majoritate a replicărilor experimentale efectuate să atingă o soluție cu acuratețe de 100%. În 19 dintre cele 20 de experimente un grup de atribute cu care un clasificator discriminează perfect între exemple a fost determinat. Totuși, într-o situație această abordare nu a găsit un astfel de subgrup. Pe aceeași populație inițială, generată din același random seeds AG beneficiind de DI2 și AAC a determinat un subgrup de atribute care permit clasificarea cu o acuratețe de 100%.

În ceea ce privește evoluția acurateței medii în populație (Fig. 5.31) pe 400 de generații, remarcăm o ușoară superioritate a implementării cu DI1. Acest aspect a fost observat și la compararea directă a celor două variante de dominanță incompletă (Fig. 5.9), dar introducerea AAC ameliorează sensibil cuantumul acestei diferențe. Tendința sporită de-a părăsi un optim local în favoarea explorării de noi soluții observată cu DI2 și AAC, înclină balanța în această direcție, în ciuda avantajului minor al acurateței medii în populațiile înaintate obținut cu metoda alternativă.

Cele mai selectate 5 gene în fiecare dintre cele două abordări sunt prezentate în tabelul 5.7. Rezultatele obținute prin ambele metode sunt consistente și se intersectează. Prezența grupată a probelor pentru "ABL proto-oncogene 1, non-receptor tyrosine kinase" și selectarea "CD52 molecule", implicată în leucemia limfocitară cronică, subliniază validitatea metodei.

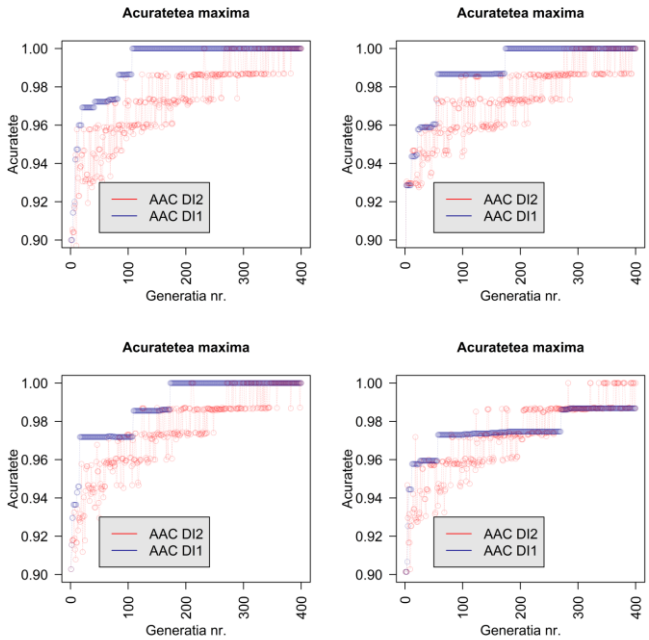


Fig. 5.30 - Evoluția acurateței maxime la utilizarea DI2 plus AAC comparativ cu DI1 combinat cu AAC.

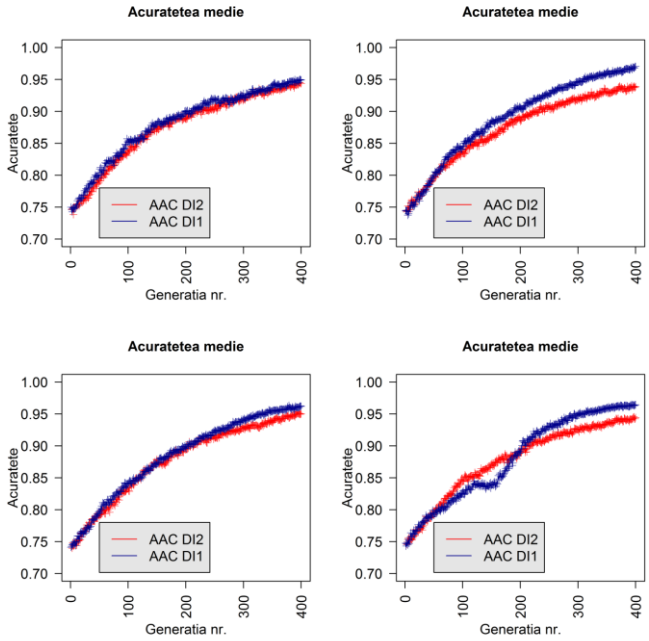


Fig. 5.31 - Evoluția acurateței medii la utilizarea DI2 plus AAC comparativ cu DI1 combinat cu AAC.

Tabel 5.7 - Atributele selectate cu AG

Nr.	AG cu DI2 și AAC		AG cu DI1 și AAC	
	Affymetrix ID	**Gene ID**	**Affymetrix ID**	**Gene ID**
1	$`39730_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"	$`39730_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"
2	$`1636_g_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"	$`1635_at `	"ABL proto-oncogene 1, non-receptor tyrosine kinase"
3	$`1635_at `	"ABL proto-oncogene 1, non-receptor tyrosine kinase"	$`38052_at`	"coagulation factor XIII A chain"
4	`34210_at`	"CD52 molecule"	$`1635_at`	"ABL proto-oncogene 1, non-receptor tyrosine kinase"
5	$`39338_at`	"S100 calcium binding protein A10"	$`38385_at`	"destrin (actin depolymerizing factor)"

5.11. Concluzie

1) Modelul dominanței incomplete

Modalitatea de mapare a genotipului la fenotip propusă de noi susține evoluția în algoritmii genetici utilizați pentru selectarea atributelor din datele de ADN microarray. Modelul propus de noi, este inspirat dintr-un fenomen care fundamentează evoluția naturală, dominanța incompletă. Ne putem aștepta așadar, ca modelarea unui fenomen atât de intim legat de evoluția naturală să-și dovedească eficiența într-o gamă largă de aplicații în care selectarea atributelor este un imperativ.

Implementarea elitismului la nivelul indivizilor în modelarea dominanței incomplete adresează problema convergenței AG într-un optim local. Am constatat că DI2 avantajează explorarea cu efecte pozitive asupra selectării atributelor. Rezultatele încurajatoare obținute în comparația dintre abordările DI1 și DI2 subliniază ameliorarea obținută prin a doua abordare.

2) Operatorul pentru atribuirea aleatorie a cromozomilor

Noul operator pentru atribuirea aleatorie a cromozomilor sprijină diversitatea genetică în generațiile înaintate și susține evoluția spre soluții superioare. Acuratețea maximă atinsă cu utilizarea operatorului AAC a fost consecvent de 100%. Comparativ, în lipsa operatorului AAC, acuratețe perfectă a fost obținută în doar jumătate dintre replicări. Putem concluziona că operatorul influențează benefic relația dintre explorare și exploatare în AG testat. De asemenea, într-o anumită măsură, introducerea atribuirii aleatorii a cromozomilor imprimă o capacitate moderată de-a părăsi un optim local în favoarea descoperirii unei soluții noi. AG cu un singur cromozom are tendința de-a converge mai rapid spre un optim local și nu are capacitatea de-a se îndrepta înspre o alternativă odată

ce o soluție satisfăcătoare a fost găsită. Datorită acestor proprietăți, implementarea cu cromozomi multipli și AAC este mai potrivită pentru studiile de selectare ale atributelor în general și cu date ADN microarray în particular. În acest context, ne interesează atributele reprezentate în generațiile înaintate în defavoarea unei combinații particulare de gene care discriminează eficient clasele. Numeroase combinații de atribute care realizează discriminarea perfectă între clase pot fi determinate. Însă o parte semnificativă dintre aceste gene sunt exprimate consecutiv unor fenomene secundare și nu reprezintă o cauzalitate cu contextul analizat. Calitatea evoluție obținute cu AAC sprijină în mod evident această finalitate. Intersecția sporită a atributelor selectate în replicări succesive reduce numărul repetițiilor necesare pentru a obține rezultate consistente și interpretabile.

3) Operatorii pentru mutații

Operatorii propuși pentru mutații adresează o problemă majoră a mutației într-un punct. La selecția atributelor din genotipuri voluminoase, cu puține alele active, mutația într-un punct are tendința de-a activa un număr mare de alele, efect vizibil în special în generațiile evoluate. Acest efect este defavorabil descoperirii unui sub-grup limitat de atribute importante. O șansă mare ca o mutație într-un punct să apară amplifică acest efect și are un efect distructiv asupra exploatării, în timp ce o rată prea redusă a mutațiilor într-un punct, nu avantajează suficient explorarea. Rata ideală de apariție a fiecărui tip de mutație depinde de setul de date analizat și trebuie determinată în preambulul experimentelor de bază. Nici una dintre mutațiile propuse nu afectează în mod spectaculos calitatea evoluției, dar toate variantele conservă numărul genelor active în clasificatorii evoluați, aspect foarte important pentru o selectare de calitate a atributelor esențiale. În plus, trei dintre propuneri avantajează evoluția acurateței medii. Valori empirice corect determinate, în special pentru mutația cu ștergerea unui întreg cromozom și a transpozonilor, pot afecta pozitiv calitatea evoluției unui AG într-un context similar.

4) Impactul cumulativ al abordărilor propuse

Evaluarea evoluțiilor comparative la utilizarea AAC cu fiecare dintre cele două versiuni de dominanță incompletă au relevat avantaje semnificative în utilizarea combinației DI2 și AAC. Considerăm că această abordare, creează premisele pentru rezultatele superioare în selecția atributelor cu algoritmi genetici din datele de ADN microarray și este metoda de elecție pentru această finalitate. Utilizarea în plus a unuia dintre operatorii propuși pentru mutație, trebuie decisă după evaluarea în prealabil al efectelor asupra evoluției pentru setul de date analizat și stabilirea empirică a șansei ca o mutație să apară.

6. CONCLUZII

Activitatea de cercetare desfășurată în cadrul tezei de doctorat s-a concentrat pe îmbunătățirea metodelor de inteligență artificială care susțin analiza genetică cu tehnologia ADN microarray. Studiul s-a concretizat în abordarea cu un algoritm evoluționist a problematicii selectării atributelor în acest context de cercetare. Am analizat în detaliu elemente de genetică, în lumina progreselor recente din biologie, pentru a surprinde și modela mai fidel principiile care asigură evoluția naturală, în inteligența artificială.

Cercetarea noastră s-a concretizat în modelarea unor principii care stau la baza evoluției naturale și inserarea lor în algoritmii genetici cu scopul ameliorării performanței lor pentru finalitatea selectării atributelor. Rezultatul este elaborarea unui algoritm genetic diploid fundamentat pe **modelul dominanței incomplete** și beneficiind de un **nou operator pentru recombinări**, fasonat după fenomenul **atribuirii aleatorii a cromozomilor** în meioză.

Pe parcursul elaborării tezei de doctorat, am studiat și implementat **modele de apariție a mutațiilor** larg tratate în genetică și am testat impactul asupra evoluției unui algoritm genetic afectat de aceste fenomene.

Pachetul software **dGAselID**, care include implementările modelelor evoluției naturale considerate și testate este integrat în R și Bioconductor, accesibil unei comunități dezvoltate de specialiști în analiză genetică pentru testare și utilizare.

Fenomenele cercetate și modelate sunt testate pe parcursul a miliarde de ani de evoluție naturală, iar rezultatele testării influenței lor asupra selectării atributelor în analiza genetică validează valoarea lor în acest scop. Generalitatea acestor principii și flexibilitatea algoritmilor genetici în a aborda diferite probleme de optimizare și selectare a atributelor ne îndreptățesc să prevedem utilitatea metodei propuse de noi în situații variate, depășind cadrul examinării datelor de ADN microarray sau al analizei genetice.

6.1. Observații finale

Modelul dominanței incomplete din evoluția naturală a fost implementat cu succes pentru a adresa problema mapării genotipului la fenotip în algoritmii genetici diploizi. Modelul elaborat reprezintă o alternativă la constrângerile impuse de abordarea clasică, definirea unei scheme de dominare specifice și restrictive. Testele efectuate au evidențiat impactul pozitiv al acestei abordări asupra evoluției AG diploid, angajat în selectarea atributelor din date microarray. Am dezvoltat și testat două versiuni **DI1** și **DI2**, diferite în privința modului de selecție al indivizilor care persistă în generația succesivă. Ambele variante susțin cu succes evoluția, dar particularitățile diferite le recomandă pentru cadre de cercetare felurite. *În contextul analizei datelor ADN microarray, experimentele noastre au evidențiat superioritatea*

metodei **DI2**, care oferă un raport foarte dezirabil între explorarea și exploatarea cu algoritmul genetic.

Diversitatea genetică este susținută în evoluția naturală prin două fenomene care au loc în timpul meiozei. Unul dintre acestea, intitulat crossing-over, a fost pe larg utilizat în multiple implementări de algoritmi genetici. Al doilea fenomen care susține variabilitatea genetică în natură, atribuirea aleatorie a cromozomilor, a fost trecut cu vederea în elaborarea algoritmilor evoluționiști. **Operatorul AAC**, conceput în timpul elaborării tezei de doctorat, modelează fenomenul omonim din meioză. Testele efectuate confirmă valoarea operatorului în susținerea explorării pe parcursul evoluției AG. Experimentele efectuate validează oportunitatea introducerii acestui operator în analiza datelor ADN microarray. Generalitatea fenomenului modelat ne îndreptățește să considerăm că operatorul conceput de noi își va găsi utilitatea în aplicații multiple, din domenii variate, care angajează algoritmii genetici.

În general, rata recombinărilor este stabilită în algoritmii genetici cu un argument pentru șansa ca o recombinare să apară. Am ajustat rata recombinărilor indirect, prin **defalcarea genomului într-un număr ajustabil de cromozomi** cu dimensiuni diferite, după modelul oferit de genetica umană. Testele noastre au demonstrat un impact pozitiv substanțial al acestei metode în selectarea atributelor cu date ADN microarray. Putem prevedea o influență foarte valoroasă a acestei abordări în cazul *chip-urilor microarray care pot fi adaptate de către cercetător pentru o cercetare particulară*.

Am modelat și implementate câteva fenomene care duc la apariția unor **tipuri diferite de mutații** în genetica umană. Propunerile noastre au adresat cu succes problema amplificării numărului alelelor 1 în seturile de cromozomi, consecutiv utilizării mutației punctuale. Testele noastre au evidențiat proprietățile fiecărui operator în parte. Am discutat impactul lor în selectarea atributelor din datele de ADN microarray și am determinat variantele cele mai avantajoase. Toți operatorii propuși au fost implementați în pachetul software dGAselID, deoarece contexte diferite de cercetare, din analiza genetică sau alt domeniu, pot fi avantajate de un operator sau altul.

După evaluarea de ansamblu a rezultatelor obținute în timpul testărilor efectuate, am concluzionat că abordarea de elecție pentru selectarea unui sub-grup de atribute, care poate eventual oferi o explicație cauzală pentru o anumită patologie, din datele microarray analizate este **algoritmul genetic diploid** cu un **număr multiplu de cromozomi**, **dominanța incompletă** în varianta **DI2** și beneficiind de **operatorul pentru atribuirea aleatorie a cromozomilor**.

6.2. Contribuții personale

Pentru realizarea metodei propuse, am studiat fenomene care fundamentează evoluția naturală. Am pornit de la premiza că, principiile care susțin evoluția naturală sunt validate de succesul pe parcursul a miliarde de ani. Elucidarea acestor legi și implementarea lor în algoritmii evoluționiști va continua să amelioreze metodele existente și vor impulsiona conceperea de abordări noi.

Dominanța incompletă. Modelul propus, prezentată în subcapitolul 3.3, reprezintă o alternativă la definirea unei scheme de dominanță, particulare unui

cadru individual, pentru maparea genotipului la fenotip în implementarea algoritmilor genetici diploizi. Pe parcursul dezvoltării experimentelor s-a dovedit utilă elaborarea a două variante, **DI1** și **DI2**, cu proprietăți semnificativ diferite și aplicabile de elecție în contexte diferite. Experimentele efectuate recomandă implementarea DI2 în cazul selectării atributelor în datele ADN microarray.

Operatorul pentru atribuirea aleatorie a cromozomilor (AAC). Discutat în subcapitolul 3.4, operatorul propus modelează un fenomen care susține diversitatea genetică și are loc în timpul meiozei. Testele efectuate confirmă utilitatea implementării acestui model în algoritmii genetici.

Pachetul software dGAselID. Perfect integrat în R și Bioconductor, pachetul dGAselID facilitează utilizarea metodei propuse pentru selectarea atributelor în contextul analizei genetice și alte domenii de cercetare. Metoda este astfel accesibilă unei comunități foarte diverse de investigatori din mediul academic.

Alternative la mutația punctuală. Am modelat și evaluat impactul unor operatori pentru mutații în analiza datelor ADN microarray. Operatorii pentru mutații implementați în pachetul software dGAselID sunt:

- Operatorul pentru **mutația fără sens**,
- Operatorul pentru **mutația cu deplasare**,
- Operatorul pentru **mutația cu ștergerea unui segment**,
- Operatorul pentru **mutația ștergerea unui cromozom**,
- Operatorul pentru **mutația de tip transpozon**.

Variantele testate adresează tendința de-a amplifica numărul atributelor considerate în generațiile înaintate, dar prezintă și dezavantaje. Operatorii pentru mutația cu ștergerea unui întreg cromozom și mutația modelată după transpozoni s-au dovedit avantajoase pentru analiza datelor studiate.

6.3. Perspectivă de dezvoltare

Pentru viitor, ne propunem următoarele direcții de dezvoltare a direcțiilor urmate pe parcursul realizării tezei de doctorat:

- testarea efectului separării genomului într-un număr variabil de cromozomi în selectarea atributelor din date ADN microarray cu biochip-uri adaptabile unei cercetări particulare,
- evaluarea modelului dominanței incomplete asupra evoluției algoritmilor genetici angajați pentru selectarea atributelor din date aparținând altor domenii de cercetare, în afara spectrului analizei genetice,
- validarea operatorului AAC în algoritmii genetici implicați în problematici variate, din domenii diferite de investigare,
- elaborarea unui operator pentru mutații mai eficient în susținerea evoluției AG pentru selectarea atributelor din date microarray.

BIBLIOGRAFIE

[1] R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[2] W. Huber, V.J. Carey, R. Gentleman, ..., M. Morgan. Orchestrating high-throughput genomic analysis with Bioconductor. Nature Methods, 2015:12, 115.

[3] D.A. Kulesh, D.R. Clive, D.S. Zarlenga și J.J. Greene. Identification of interferon-modulated proliferation-related cDNA sequences. Proc Natl Acad Sci, USA, 84: 8453–8457, 1987.

[4] F.H.C. Crick. On protein synthesis. Symp. Soc. Exp. Biol. XII, 139-163, 1956.

[5] V. Benoit, A. Steel, M. Torres, Y-Y. Yu , H. Yang și J. Cooper. Evaluation of three-dimensional microchannel glass biochips for multiplexed nucleic acid fluorescence hybridization assays. Anal Chem 73:2412–2420, 2001.

[6] A. Binet. The development of the Binet-Simon Scale: New methods for the diagnosis of the intellectual level of subnormals (ES Fite, Trans.). D. Readings in the History of Psychology. NewYork: Appleton-Century-Crofts, 1905.

[7] J. McCarthy. What is artificial intelligence? Stanford University, Stanford, CA 94305, 2007. http://www-formal.stanford.edu/jmc/.

[8] S.J. Russell și P. Norvig. Artificial intelligence : a modern approach. Prentice-Hall, Egnlewood Cliffs, 1995.

[9] S.J. Russell și P. Norvig. Artificial intelligence : a modern approach, Ediţia a III-a. Upper Saddle River, NJ: Prentice Hall, 2009.

[10] P. Larranaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J.A. Lozano, R. Armañanzas, G. Santafé, A. Pérez și V. Robles. Machine learning in bioinformatics. Brief Bioinform. 7(1):86-112, 2006.

[11] I.H. Witten, E. Frank și M.A. Hall. Data mining: practical machine learning tools and techniques, Ediţia a III-a. Burlington, MA: Morgan Kaufmann, 2011.

[12] J.T.L. Wang, M.J. Zaki, H.T.T. Toivonen și D. Shasha (Eds). Data mining in bioinformatics. Springer-Verlag, 2005.

[13] L. Devroye, L. Györfi și G. Lugosi. A probabilistic theory of pattern recognition. Springer-Verlag, 1996.

[14] T. Hastie, R. Tibshirani și J. Friedman. The elements of statistical learning. Springer-Verlag, 2001.

[15] T.M. Mitchell. Machine learning. McGraw-Hill, 1997.

[16] A. Webb. Statistical pattern recognition. Wiley, 2002.

[17] M. Mohri, A. Rostamizadeh și A. Talwalkar. Foundations of machine learning. Cambridge, MA, MIT Press, 2012.

[18] I. H. Witten și E. Frank. Data mining: practical machine learning tools and techniques, Ediţia a II-a. Morgan Kaufmann, San Francisco, 2005.

[19] P. Baldi și S. Brunak. Bioinformatics: the machine learning approach. MIT Press, 2001.

[20] P.A. Pevzner. Computational molecular biology: an algorithmic approach. MIT Press, 2000.

[21] A.R. Webb și K.D. Copsey. Statistical pattern recognition. Hoboken, NJ, Wiley, 2011.

[22] T. Hastie, R. Tibshirani și J. Friedman. The elements of statistical learning: data mining, inference, and prediction. Heidelberg, Germany, Springer, 2009.

[23] C.M. Bishop. Pattern recognition and machine learning. Heidelberg, Germany, Springer-Verlag, 2007.

[24] D. de Ridder, J. de Ridder și M.J.T. Reinders. Pattern recognition in bioinformatics. Brief Bioinform., 14(5):633-47, 2013.

[25] F. Valafar. Pattern recognition techniques in microarray data analysis: a survey. Annals of the NewYork Academy of Sciences, 980:41–64, 2002.

[26] R.O. Duda, P.E. Hart, D.G. Stork. Pattern classification, Ediţia a II-a. Wiley-Interscience, 2001.

[27] P. Vitányi. Information distance in multiples. IEEE Transactions on Information Theory 57 (4), 2451-2456, 2011.

[28] M.M. Deza și E. Deza. Encyclopedia of distances. Springer-Verlag, 2009.

[29] T. Golub. golubEsets: exprSets for golub leukemia data. Pachet R versiunea 1.14.0, 2016.

[30] J.H. Ward. Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association 58 (301): 236–244, 1963.

[31] S. P. Lloyd. Last square quantization in PCM. IEEE Transactions on Information Theory, 28 (2), 129–137, 1982.

[32] G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Journal für die Reine und Angewandte Mathematik, 133:97-178, 1907.

[33] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel și F. Leisch. e1071: misc functions of the department of statistics. Probability Theory Group, TU Viena. Pachet R versiunea 1.6-7, 2015. https://CRAN.R-project.org/package=e1071.

[34] R. Tibshirani, T. Hastie, B. Narasimhan și G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proc. Natl Acad. Sci. USA 99:6567–6572, 2002.

[35] A. Statnikov, C.F. Aliferis, I. Tsamardinos, D. Hardin și S. Levy. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics, 21(5):631–43, 2005.

[36] J.W. Lee, J. B. Lee, M. Park și S.H. Song. An extensive comparison of recent classification tools applied to microarray data. Computational Statistics and Data Analysis, 48 (4), 869-885, 2005.

[37] W. M. Bolstad. Introduction to Bayesian statistics. John Wiley & Sons, 2004.

[38] B. Efron, R. Tibshirani, J.D. Storey și V. Tusher. Empirical Bayes analysis of a microarray experiment. JAm Stat Assoc, 96(456):1151–60, 2001.

[39] S. Theodoridis, K. Koutroumbas. Pattern recognition, Ediţia a III-a. Academic Press, Inc. Orlando, FL, USA, 2006.

[40] M. Merleau-Ponty și C. Smith. Phenomenology of perception, Ediţia a II-a, Routledge, 2002.

[41] A.B. Olshen și A.N. Jain. Deriving quantitative conclusions from microarray data. Bionformatics, 18(7):961–70, 2002.

[42] V. Vapnik și A. Lerner. Pattern recognition using generalized portrait method. Automation and Remote Control, 24 774-780, 1963.

[43] B. E. Boser, I. M. Guyon și V. N. Vapnik. A training algorithm for optimal margin classifiers. 5th Annual ACM Workshop on COLT, pag. 144-152, ACM Press, 1992.

[44] I. Guyon, J. Weston și S. Barnhill. Gene selection for cancer classification using support vector machines. Machine Learning 46:389–422, 2002.

[45] R. Gentleman. Microarray Experiments. 2003. https://www.Bioconductor.org/help/ course-materials/2003/Milan/Lectures/Filtering.pdf

[46] G.K. Smyth. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. Statistical Applications in Genetics and Molecular Biology, 3(1), 2004.

[47] P. Broberg. Statistical methods for ranking differentially expressed genes. GenomeBiol, 4:R41, 2003.

[48] C. Steinhoffand și M.Vingron. Normalization and quantification of differential expression in gene expression microarrays. Brief Bioinform. 7(2):166-77, 2006.

[49] R. Gentleman, V. Carey, W. Huber și F. Hahne. genefilter: genefilter: methods for filtering genes from high-throughput experiments. Pachet R versiunea version 1.54.2, 2016.

[50] W. Pan. A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments. Bioinformatics,18(4):546–54, 2002.

[51] Y. Wang, I.V. Tetko, M.A. Hall, E. Frank, A. Facius, K.F. Mayer și H.W. Mewes. Gene selection from microarray data for cancer classification – a machine learning approach. Computational Biology and Chemistry, 29: 37–46, 2005.

[52] I. Inza, P. Larrañaga, R. Blanco și A.J. Cerrolaza. Filter versus wrapper gene selection approaches in DNA microarray domains. Artificial Intelligence in Medicine, 31(2):91–103, 2004.

[53] L. Li, C.R. Weinberg, T.A. Darden și L.G. Pedersen. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics 17(12):1131–42, 2001.

[54] C.H. Ooi și P. Tan. Genetic algorithms applied to multi–class prediction for the analysis of gene expression data. Bioinformatics 19(1):37–44, 2003.

[55] E.P. Xing, M.I. Jordan și R.M. Karp. Feature selection for highdimensional genomic microarray data. Proceedingsof the Eighteenth International Conference in Machine Learning. ICML, 2001.

[56] B. Krishnapuram, L. Carin, A.J. Hartemink. Joint classifier and feature optimization for comprehensive cancer diagnosis using gene expression data. J Comput Biol, 11(2–3):227–42, 2004.

[57] D.H. Wolpert și W.G.Macready. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1):67–82, 1997.

[58] D.H. Wolpert și W.G.Macready. Coevolutionary free lunches. IEEE Transactions on Evolutionary Computation, 2005.

[59] E. Bair, T. Hastie, D. Paul și R. Tibshirani. Prediction by supervised principal component. J. Amer. Statis. Assoc. 101:119–137, 2006.

[60] G.B. Fogel și D.W. Corne. Evolutionary computation in bioinformatics. Morgan Kaufmann, 2002.

[61] J.H. Holland. Adaptation in natural and artificial systems. University of Michigan Press, 1975.

[62] W.D. Hillis. Co−evolving parasites improve simulated evolution as an optimization procedure. Physica D 42:228–234, 1990.

[63] J.R. Levenick. Inserting introns improves genetic algorithm success rate: taking a cue from biology. Proceedings of the Fourth International Conference on Genetic Algorithms. Morgan Kaufmann, 1991.

[64] M. Mitchell. An introduction to genetic algorithms. The MIT Press, Cambridge, Massachusetts, London, England, 1999.

[65] M.K. Kerr, M. Martin, G.A. Churchill. Analysis of variances from gene expression microarray data. J Comput Biol 2000; 7(6):819–37.

[66] D.E. Goldberg. Genetic algorithms in search, optimization, and machine learning. Addison−Wesley, 1989.

[67] M.A. Bedau și N.H.Packard. Measurement of evolutionary activity, teleology, and life. Artificial Life II. Addison−Wesley, 1992.

[68] L.J. Eshelman, R.A. Caruana și J.D. Schaffer. Biases in the landscape. Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann, 1989.

[69] J.M. Baldwin. A new factor in evolution. American Naturalist 30: 441–451, 536–553, 1986.

[70] K.A. De Jong. An Analysis of the behavior of a class of genetic adaptive systems. Teză de doctorat, University of Michigan, Ann Arbor, 1975.

[71] N.T. Meliță, I. Popescu și Ș. Holban. A genetic algorithm approach to DNA microarrays analysis of pancreatic cancer. Advances in Electrical and Computer Engineering, nr. 2/2008, vol. 8, pag. 43-48, 2008.

[72] D.E. Goldberg și R.E. Smith. Nonstationary function optimization using genetic algorithms with dominance and diploidy. Proc. of the 2nd Int. Conf. on Genetic Algorithms, pag. 59-68, 1987.

[73] K.P. Ng and K.C. Wong. A new diploid scheme and dominance change mechanism for non-stationary function optimisation. Proc. of the 6th Int. Conf. on Genetic Algorithms, pag. 159-166, 1995.

[74] C. Ryan. The degree of oneness. Proc. of the 1994 ECAI Workshop on Genetic Algorithms, 1994.

[75] A.S. Uyar și A.E. Harmanci. A new population based adaptive dominance change mechanism for diploid genetic algorithms in dynamic environments. Soft Computing, 9(11): 803-814, 2005.

[76] R.E. Smith, D.E. Goldberg. Diploidy and dominance in artificial genetic search. Complex Systems, vol. 6, pag.251-285, 1992.

[77] V. Carey, R. Gentleman, J. Mar, cu contribuții din partea lui J. Vertrees și L. Gatto. MLInterfaces: Uniform interfaces to R machine learning procedures for data in Bioconductor containers. Pachet R versiunea 1.52.0, 2016.

[78] J.M. Baldwin. A new factor in evolution. The American Naturalist, vol. 30, nr. 354, pag. 441–451, 1896.

[79] G.J. Mendel. Experiments in plant hybridisation. 1865.

[80] G.J. Tortora și B. Derrickson. Principles of anatomy and physiology, Ediţia a XIV-a. Wiley, 2014.

[81] S. Yang. Learning the dominance in diploid genetic algorithms for changing optimization problems. Proceedings of the 2nd International Symposium on Intelligence Computation and Applications, pag. 157-162, 2007.

[82] A.S. Uyar și A.E. Harmanci. Comparison of major domination schemes for diploid binary, genetic algorithms in dynamic environments. Applications and Science in Soft Computing: Advances in Soft Computing, pag. 75-80, 2004.

[83] J. Lewis, E. Hart și R. Graeme. A comparison of dominance mechanisms and simple mutation on non-stationary problems. Proceedings of Parallel Problem Solving from Nature, Springer Verlag, 1998.

[84] R. Lewis. Human genetics, Ediţia a XI-a. McGraw-Hill Science/Engineering/Math, pag. 46, 2014.

[85] A. Koschmieder, K. Zimmermann, S. Trissl, T. Stoltmann și U. Leser. Tools for managing and analyzing microarray data. Brief Bioinform 13(1):46–60, 2012.

[86] W. Gregory Alvord, J.A. Roayaei,O.A. Quiñones și K.T. Schneider. A microarray analysis for differential gene expression in the soybean genome using Bioconductor and R. Brief Bioinform. 8(6):415-31, 2007.

[87] L. Scrucca. On some extensions to GA package: hybrid optimisation, parallelisation and islands evolution. Submitted to R Journal, 2016. Disponibil online la URL http://arxiv.org/abs/1605.01931.

[88] F. Tenorio. gaoptim: Genetic algorithm optimization for real-based and permutation-based problems. 2013. URL https://cloud.r-project.org/web/packages/gaoptim/index.html.

[89] E. Willighagen și M. Ballings. genalg: R based genetic algorithm, 2015. URL https://cloud.r-project.org/web/packages/genalg/index.html.

[90] C.-S. Tsou. nsga2R: elitist non-dominated sorting genetic algorithm based on R. 2015. URL https://cloud.r-project.org/web/packages/nsga2R/index.html.

[91] M.A. Wolters. A genetic algorithm for selection of fixed-size subsets with application to design problems. Journal of Statistical Software, vol. 68, no. 1, pag. 1-18, 2015.

[92] D. Akdemir, J.I. Sanchez și J.-L.Jannink. Optimization of genomic selection training populations with a genetic algorithm. Genetics Selection Evolution, 47(1): 38, 2015.

[93] D. Kepplinger. gaselect: genetic algorithm (GA) for variable selection from high-dimensional data. 2015. URL https://cloud.r-project.org/web/packages/gaselect/index.html.

[94] T. Pajala. mogavs: multiobjective genetic algorithm for variable selection in regression. 2016. URL https://cloud.r-project.org/web/packages/mogavs/index.html

[95] X. Li. ALL: a data package. Pachet R versiunea 1.14.0, 2009.

[96] Y. Guo, T. Hastie și R. Tibshirani. rda: shrunken centroids regularized discriminant analysis. Pachet R versiunea 1.0.2-2, 2012. URL https://CRAN.R-project.org/package=rda

[97] S. Chiaretti, X. Li, R. Gentleman, A. Vitale, M. Vignetti, F. Mandelli, J. Ritz și R. Foa. Gene expression profile of adult T-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. Blood, 103(7):2771-8, 2004.

[98] S. Faderl, H.M. Kantarjian, M. Talpaz și Z. Estrov. Clinical significance of cytogenetic abnormalities in adult acute lymphoblastic leukemia. Blood, 91(11):3995-4019, 1998.

[99] J.H. Kersey. Fifty years of studies of the biology and therapy of childhood leukemia. Blood, 92(5):1838, 1998.