Szegedi Tudományegyetem
Mesterséges Intelligencia Kutatócsoport

Protein Classification in a Machine Learning Framework

PhD értekezés tézisei

Kertész-Farkas Attila

Témavezetők:
Dr. Csirik János
Dr. Kocsor András

Szeged
2008

„Elméletileg nincs különbség
elmélet és gyakorlat között.
Gyakorlatilag meg van."
Jan L.A. van de Snepscheut

Bevezetés

Bár a bioinformatika területét nehéz lenne pontosan körülhatárolni, de elmondhatjuk, hogy a biológia, a matematika és az informatika egy közös része [1; 2]. A hetvenes-nyolcvanas években a molekuláris biológia és kémia területén jelentek meg a sok adatot termelő technikai újítások, másrészt elérhetővé váltak az olcsó és hatékony számítógépek is. Ezek együttesen vezettek egy új tudományág, a bioinformatika kialakulásához, amelynek egyik fő és talán legszélesebb ága a kutatólaborokban termelt adatok feldolgozása, archiválása, rendezése, rendszerezése, valamint az adatok kiértékelése és a bennük rejlő összefüggések feltárása lett.

A bioinformatika egyik fő feladata a fehérje-szekvenciához tartozó fehérje molekula funkciójának és térszerkezetének megállapítása. Erre három fő módszertan alakult ki:

(i) A fehérjék térszerkezetének vizsgálata kémiai kísérletekkel és fizikai eszközökkel. Ezek a módszerek bonyolult és drága berendezéseket igényelnek, és nem is adnak mindig megbízható eredményt.

(ii) A fehérje-szekvencia ismeretében számítógépes algoritmusokkal modellezik az adott szekvenciához tartozó térszerkezetet. A probléma NP-nehéz, ezért különböző heurisztikus algoritmusukat fejlesztettek ki.

(iii) A fehérje-szekvenciát összehasonlítják már ismert térszerkezetű és funkciójú szekvenciákkal, és a hozzá leghasonlóbb szekvencia tulajdonságaiból következtethetünk a kérdéses tulajdonságokra, vagy gépi tanulási algoritmusokkal sorolják be egy már jól ismert fehérjeosztályba.

Jelen disszertáció témája is a (iii) területre sorolható; a következő fejezetekben összefoglaljuk a disszertáció tézispontjait.

Fehérje-osztályozási adatbázis

Egy fehérje molekula aminosav molekulák lineárisan összekapcsolt sorozata és így ábrázolható egy olyan ábécé (amelyet nevezzük itt aminosav-ábécének) feletti sztringként, ahol az ábécé minden eleme egy-egy aminosavhoz van rendelve. Az aminosavak száma – és így az ábécé mérete – 20. Egy fehérjét kódoló sztring első néhány eleme például a következő: „`rintvrgpit iseagftlth ehicgssagf lraw peffgs ...`"[1]. Az ilyen sztringet a bioinformatika területén szokásosan fehérje-szekvenciának nevezik, és a fehérje-szekvencia nem tévesztendő össze fehérjék sorozatával. A fehérje szekvenciák abban specifikusak (különböznek tetszőleges sztringektől), hogy (1) az ábécé mérete 20, (2) a hosszuk átlagosan néhány tíztől néhány ezerig terjed, (3) a fehérje-szekvenciák gyakorta hosszú ismétlődő szakaszokat tartalmazhatnak, (4) továbbá minden fehérje-szekvenciához valós fehérje molekula tartozik, viszont nem tartozik minden aminosav-ábécé feletti sztringhez valós fehérjemolekula. A dolgozatban vizsgált fehérje-szekvenciák valós fehérjékhez tartoznak.

[1]Megjegyezzük, hogy fehérje molekulák más módon is reprezentálhatók, például a fehérjét kódoló DNS szakasz nukleotid sorrendjével, vagy a fehérjét felépítő atomok típusának és 3-dimenziós koordinátáinak megadásával.

A fehérjék biológiai hasonlóságuk alapján osztályokba csoportosíthatók. Például a különböző élőlények vörös vértestében az oxigén megkötésére szolgáló fehérjék a hemoglobin osztályba sorolhatók. Ezek a fehérjék élőlényenként és fajonként kissé eltérőek lehetnek, de térszerkezetük és funkciójuk hasonló. Így egy újonnan meghatározott fehérje- szekvencia funkciója és térszerkezete megjósolható ismert fehérje-osztályba sorolással. Címkézzük a szekvencia-osztályokat rendre $1, 2, \ldots, N$ természetes számokkal, ahol N jelöli az osztályok számát. Jelölje az (s, y) páros az s fehérje-szekvenciát és a szekvencia y osztályát $(1 \leq y \leq N)$.

A mintaosztályozás célja általában, hogy egy s mintát a mintának megfelelő y osztályba sorolja, azaz egy olyan F függvény meghatározása, amelyre az $F(s) = y$ teljesül. A gépi tanulás egyik feladata ilyen F függvények paramétereinek automatikus tanulása (beállítása) úgy, hogy a helytelenül osztályozott minták száma[2] ($|\{s \mid F(s) \neq y\}|$) minimális legyen [3]. A disszertációban az osztályozás során kitüntetünk egy osztályt, amelyet pozitív osztálynak (vagy cél osztálynak) és az elemeit pozitív elemeknek nevezzük, míg a többi osztályba tartozó mintát egy osztályként kezeljük és negatív osztálynak, elemeit negatív elemeknek nevezzük. Erre azért van szükség, mert az osztályozó algoritmusok többsége két osztályra definiált [3]. A tanítás elvégzéséhez a pozitív és a negatív osztályt két-két nem-üres, diszjunkt részre osztják, egy tanuló és egy tesztelő halmazra, így kapunk pozitív tanuló, pozitív tesztelő, negatív tanuló, negatív tesztelő halmazokat és az ilyen felosztását az adatoknak osztályozási feladatnak nevezzük. A tanuló halmazt használják az F osztályozó algoritmus paramétereinek beállítására. A tesztelés során (a tanulás eredményességének mérésére) a tesztelő halmaz elemeinek az osztálycímkéje az osztályozó algoritmus számára rejtve marad (csak a kiértékeléshez alkalmazzák), és a címke meghatározását az F osztályozó algoritmustól várjuk. A disszertációban az osztályozás kiértékeléséhez az ún. ROC analízist [4] használtuk, amelyről bővebben az Olvasó a disszertációban olvashat. Erről ebben a tézisfüzetben elegendő annyit tudni, hogy a ROC analízissel kapott érték a [0,1] intervallumba eshet, és a jobb módszerhez a nagyobb érték tartozik.

A osztályozó algoritmusok jelentős része megkívánja az osztályozandó adatok egy rögzített dimenziójú vektorként való ábrázolását. A disszertációban a fehérje-szekvenciák ábrázolásához az ún. „Empirical Feature Mapping" (EFM) módszert [5] használtuk, amely egy s szekvenciához az alábbi numerikus vektort rendeli:

$$F_s = [f_{x_1}, f_{x_2}, \ldots, f_{x_n}], \tag{1}$$

ahol $x_1, x_2, \ldots, x_n$ a tanuló halmazban lévő szekvenciák és f_{x_i} az s szekvenciának az x_i szekvenciához számított hasonlósági értéke egy előre rögzített hasonlósági módszer szerint. Azért ezt a módszer választottuk, mert ezzel jelentősen jobb fehérje ábrázolás érhető el osztályozási szempontból, mint más módszerekkel [6].

A fehérjék gyakran hierarchikusan kategorizálhatók, hasonlóan az élővilág rendszertanához (faj, nemzetség, család, rend, osztály, törzs, ország). Az ilyen rendezés reprezentálható egy H magasságú, gyökerezett fával, ahol a H famélység jelöli a hierarchiák számát, a fa gyökere reprezentálja az egész fehérjehalmazt, a fa levelei reprezentálják magukat a fehérjéket, továbbá minden belső pontja a fának a pontból leszármazott fehérjék halmazát jelöli és e halmazokat kategóriáknak nevezzük. A fa azonos magasságán lévő kategóriák az adatbázis egy particionálását adják. Egy ilyen fa ábrázolása az 1 ábra A részén látható. Megjegyezzük, hogy e fa kiegyensúlyozott[3], mert minden objektum beletartozik a hierarchia minden szintjének valamely kategóriájába, azaz például nincs olyan, hogy egy faj nem

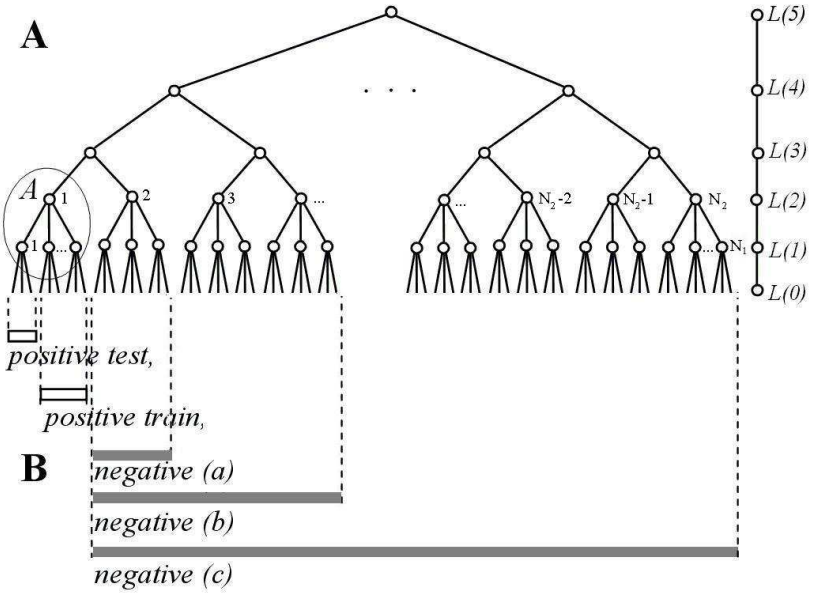
[2] Az osztályozás helyességének mérésére finomabb módszerek is elterjedtek.

[3] Itt egy fa kiegyensúlyozottságon azt értjük, hogy a fában az összes a gyökérből a levélbe vezető út hossza egyforma.

tartozik bele egyik nemzetsége se és így tovább.

Indexeljük a fa azonos magasságon lévő csúcsait $1, \ldots, N_p$ természetes számokkal, ahol N_p jelöli a p $(0 \leq p \leq H)$ magasságban lévő pontok számát. Megjegyezzük, hogy N_0 pontosan az adatbázis elemeinek számával egyenlő, míg $N_H = 1$, mert a H. szint csupán a gyökércsúcsot tartalmazza, továbbá $N_H \leq N_{H-1} \leq \cdots \leq N_1 \leq N_0$. Jelölje most az (s, y) páros a fa egy s levelét (egy fehérje-szekvenciát) és egy $y \in N^{H+1}$ vektort, ahol N a természetes számok halmazát jelöli, amely a gyökérből az s-be vezető utat kódolja úgy, hogy y_i $(\leq N_i)$ vektorkomponens jelöli az úton az i. csúcs indexét az i. szinten a fában. Ekkor az adatbázis elemeiből a következőképpen készíthető pozitív és negatív osztály. Rögzítsünk egy i szintet a hierarchiában, majd ezen belül jelöljünk ki egy j csúcsot (kategóriát) és legyen a pozitív osztály $P = \{(s, y) \mid y_i = j\}$, míg a negatív osztály $N = \{(s, y) \mid y_i \neq j\}$. Ekkor a pozitív és a negatív halmazokból osztályozási feladatok a klasszikus módszerekkel (kereszt-validáció, One-Leave-Out, stb. [3]) megkapható. Megjegyezzük, hogy az adatbázis elemei nincsenek előre pozitív és negatív oszátlyokba sorolva, hanem különböző oszátlyozási feladatokat generáltunk a hierarchia i szintjének és azon belül a $j(< N_i)$ kategória variálásával, igy egy adatbázis elem az egyik oszátlyozási feladatban pozitív oszátlyba tartozhat, míg egy másikban a negatív osztályohoz.

A gyakorlatban előfordul, hogy egy olyan fehérje-szekvenciával van dolgunk, amely egy ismert kategórián belül egy új részkategóriába tartozik. Ha az osztályozási feladat létrehozása során a pozitív osztályon (kategórián) belül a tanuló és a tesztelő halmazokat véletlenszerű felosztással határozzuk



1. ábra. A felügyelt kereszt-validáció alkalmazása egy sematikus, 5 szintű, hierarchikusan rendszerezett adatbázisra. (A) Pozitív halmaznak az A-val (j=1) jelölt csoportot választottuk az $L(2)$ szinten (i=2) és ezen belül a tanuló és tesztelő halmazok az $L(1)$ szinten definiáltuk úgy, hogy egy kategóriát jelölünk ki pozitív tesztnek (k=1), a többi a pozitív tanuló halmaznak. Így a tanuló és a tesztelő halmazok diszjunkt fehérjekategóriákból állnak és a tanuló algoritmusnak a tesztelés során egy új részkategória elemeit kell helyesen osztályoznia. (B) A negatív halmazok szintén definiálhatók kategorikusan a kiválasztott A pozitív csoporthoz. A negatív (a), (b) és (c) halmazok rendre az A csoportól a hierarchiában egyre távolabbi kategóriát tartalmaznak.

meg, akkor (nagy valószínűséggel) a tanuló és a tesztelő halmaz ugyanazon részkategóriák elemeiből áll. Ekkor a tesztelés során nem kapunk információt arra, hogy az osztályozó algoritmus menynyire képes helyesen osztályozni a kategórián belül a tanulás során nem látott részkategóriába tartozó elemeket. Egyik eredményünk, hogy kifejlesztettünk egy általános módszertant osztályozási feladatok konstruálására hierarchikusan kategorizált fehérje-adatbázisokban, ahol a pozitív osztályon (kategórián) belül az elemek szétválogatása tanuló- és tesztelő halmazokba a részkategóriák figyelembevételével történik. A módszer a következő. Legyen a $P = \{(s, y) \mid y_i = j\}$ egy pozitív osztály (a j. kategória az i. szinten), és ezen belül pozitív teszthalmaznak válasszunk ki egy k részkategória elemeit a j kategórián belül, azaz a fában a k csúcs a j csúcsnak fia, míg a pozitív tanulóhalmaz legyen a többi részkategória eleme a j kategórián belül. A negatív halmaz általában több kategóriát tartalmaz, és e kategóriák tesztelő- illetve tanuló halmazba sorolását véletlenszerű besorolással végezzük el, mert így az osztályozó algoritmusnak ismert negatív kategóriákon belül negatív részkategóriát is helyesen kell osztályoznia. A módszert felügyelt kereszt-validációnak neveztünk el [7], mert a tanuló és a tesztelő halmazok készítésénél kihasználtuk azt az információt, hogy az elemek hierarchikusan rendezettek. Ezzel a felosztással véleményünk szerint pontosabb becslést kaphatunk arra, hogy egy betanított osztályozó algoritmus mennyire képes helyesen osztályozni egy új, de még nem látott részcsoport elemeit egy már ismert csoporton belül, és ez tekinthető a betanított osztályozó algoritmus általánosító képességének is. Ezt a módszert összehasonlítottuk a hagyományos kereszt-validációs és a One-Leave-Out módszerekkel [3] is és azt kaptuk, hogy a felügyelt kereszt-validációs módszerrel készített osztályozási feladatokon az osztályozó algoritmusok átlagos teljesítmény gyengébb, mint a hagyományos módszerekkel készített osztályozási feladatokon. Azaz a hagyományos validációs technikák túlértékelik az osztályozó valódi teljesítményét valós gyakorlati problémákon.

Megvizsgáltuk, hogy hogyan módosulnak az osztályozási eredmények ha a hierarchikus kategorizálást kihasználva a negatív osztályból kategóriákat kihagyunk a negatív osztály méretének csökkentése céljából. Ugyanis a fehérje-osztályozási feladatoknál a pozitív osztály elemszáma néhány tíztől néhány százig terjed, viszont a negatív osztály mérete tíz- vagy százezres nagyságú is lehet, ami nemcsak feleslegesen lassítja az előfeldolgozó és a tanuló algoritmusok futását hanem az ún. „class-imbalanced" problémához is vezethet [8]. A hierarchikus kategorizálást kihasználva legyen a negatív osztály az $N_k = \{(s, y) \mid y_i \neq j, y_k = \phi(i, j, k)\}$ halmaz, ahol $k > i$ és a $\phi(i, j, k)$ függvény a k. szinten annak a kategóriának az indexét adja vissza, amelyikbe az i. szinten lévő j. kategória beletartozik. Így a k paraméterrel szabályozhatjuk, hogy a pozitív osztályhoz legfeljebb milyen távoli elemek tartozzanak, ahol a távolságot két elem között a hierarchiában a legrövidebb út lépésszámával definiáljuk. $k = H$ esetben a negatív osztály a pozitív osztály komplementere az egész adatbázisra nézve, valamint ha a $k = i$-t megengednék akkor erre az esetre a negatív osztály az üres halmaz lenne. Megjegyezzük, hogy minél nagyobb k értéke, annál bővebb negatív osztályt kapunk. Nevezzük ezt a módszert felügyelt kategórikus szűrésnek, és a 1 ábra B része szemlélteti.

Ezt a módszert összehasonlítottuk két másik módszerrel, nevezetesen: (i) a véletlenszerű egyenletes eloszlás szerinti kiválasztást (pl. az eredeti 10% vagy 20% megtartását), (ii) a pozitív osztályhoz legközelebbi negatív elemek kiválasztását, egy fehérjehasonlósági mérték szerint [8]. Az összehasonlító tesztek alapján az (i) módszer használatát javasoljuk, mert ezzel az osztályozási eredmény az eredetihez (azaz amikor a negatív osztály a pozitív osztály komplementere a teljes adatbázisra nézve) képest közel változatlan marad, továbbá a gyakorlatban az elemek eloszlása azonos marad. A (ii) módszerrel túlságosan nehéz osztályozási feladatot kaphatunk. A felügyelt kategorikus szűréssel

1. táblázat. Fehérje-osztályozási eredmények.

	1NN	RF	SVM	ANN	LogReg
BLAST	0.7577	0.6965	0.9047	0.7988	0.8715
SW	0.8154	0.8230	0.9419	0.8875	0.9063
NW	0.8252	0.8030	0.9376	0.8834	0.9175
LAK	0.7343	0.8344	0.9396	0.9022	0.8766
PRIDE	0.8644	0.8105	0.9361	0.9073	0.9029
DALI	0.9892	0.9941	0.9946	0.9897	0.9636

Az osztályozáshoz a szekvenciákat az általunk elkészített adatbázis, az ún. 'SCOP40' bejegyzéséből vettük és az osztályozási feladatok az adatbázis honlapján elérhető. Az osztályozási eredményeket a ROC analízissel értékeltük ki. Az osztályozáshoz használt paraméter-beállítások a disszertációban megtalálható. Az osztályozás során az osztályozó algoritmusnak a szekvenciákat térszerkezete alapján kellett a megfelelő osztályba sorolnia.

kapott osztályozási eredmények rosszabbak az eredetihez képest, de jobbak a (ii) módszerrel kapott eredményekhez képest, viszont a negatív halmaz nem reprezentálja a valós negatív fehérje-univerzumot, mert számos kategória kimaradhat. Habár ezen eredmény negatívnak tekinthető, érdemesnek tartjuk megjegyezni, mert a hierarchikusan rendezett adatbázisok jellemzése osztályozási feladatok készítése szempontjából véleményünk szerint így teljesebb.

Létrehoztunk egy fehérje-osztályozási adatbázist [9], amellyel a gépi tanulási algoritmusok és szekvencia-hasonlósági módszerek egységes adatbázison összehasonlíthatók. Továbbá az adatbázisban az osztályozási feladatokat a felügyelt-keresztvalidációs technikával készítettük. Az adatbázis fehérje-szekvenciák halmazából és a felügyelt kereszt-validációs módszerrel képzett kétosztályos osztályozási feladatokból áll. A fehérje-szekvenciákat több publikus, interneten elérhető, biológiai adatbázisból (3PGK[10], CATH[11], COG[12], SCOP[13]) válogattuk össze, és a kiválogatott szekvenciák száma eléri a 40000-et, míg a legyártott osztályozási feladatok száma közel 9500. Az adatbázisban a szekvenciák ábrázolásához az EFM módszert használtuk a BLAST [14], a Smith-Waterman (SW) [15], a Needleman-Wunsch (NW) [16], a Local Alignment Kernel (LAK) [17], a PRIDE [18] és a DALI [19] hasonlósági módszerekkel. Az adatbázis tartalmazza továbbá az osztályozási feladatokon elért eredményeket a mesterséges neurális hálózat (ANN) [20], szupport vektor gép (SVM) [21], random forest (RF) [22], legközelebbi szomszéd módszer (1NN) [3] és logisztikus regresszió (LogReg) [23] módszerekkel. Ezen eredmények mintegy baseline-ként használhatók fel későbbi összehasonlításokban, illetve tartalmazza még a felhasznált módszerek rövid leírását és a használt paraméter-beállításokat. Az adatbázis ingyenesen használható és érhető el a `http://hydra.icgeb.trieste.it/benchmark` címen.

Az 1. táblázat néhány osztályozási eredményt közöl. A táblázat oszlopaiban a használt osztályozó algoritmus neve, míg a sorokban a vektorizálás során használt hasonlósági módszer neve szerepel. Az osztályozási eredmények kiértékeléséhez az ún. ROC analízist [4] használtuk.

Ennek a tézispontnak az újdonsága, hogy elkészítettünk egy fehérje osztályozási adatbázist fehérje szekvenciák osztályozásához, amelyen a gépi tanulási eszközök és szekvencia hasonlósági algoritmusok egy egységes adatbázison hasonlíthatók össze, valamint kifejlesztettük a felügyelt-keresztvalidációs módszert, amellyel pontosabb becslést kaphatunk az osztályozó algoritmus általánosító képességének mérésére. Ez tekinthető a "fehérje-szekvencia univerzum" egyfajta jellemzésének is.

Likelihood-ratio közelítés fehérje-szekvenciák osztályozása

Egy új fehérje-szekvencia biológiai tulajdonságaira (térszerkezet, funkció) gyakran a hozzá leghasonlóbb és már tanulmányozott fehérje-szekvenciák tulajdonságaiból következtethetünk, és a hasonlóság kiértékelése szekvencia-hasonlító algoritmusokkal – például a Smith-Watermannal (SW) [15] – elvégezhető. Pontos és gyors szekvencia-hasonlító algoritmusoknak kulcs szerepük van itt és egyre pontosabb és kifinomultabb szekvencia hasonlító algoritmusok kifejlesztésére került sor, mint például BLAST[14], PSI-BLAST [24], FASTA[25] algoritmusok.

Jelölje S fehérje-szekvenciák egy halmazát, legyen $T \subset S$ az ismert és tanulmányozott fehérje-szekvenciák halmaza, $P \subset T$ fehérje szekvenciák egy osztálya[4] és $s : S \times S \to R^+$ egy hasonlósági függvényt (amelytől elvárjuk, hogy biológiailag hasonló fehérje szekvenciákhoz nagyobb értéket rendel, míg kevésbé hasonlókhoz kisebbet). Definiáljuk azt, hogy egy $x \in S$ szekvencia mennyire tartozik a P osztályhoz az x szekvenciához leghasonlóbb P osztálybeli szekvencia hasonlósági értékével, a legközelebbi szomszéd módszer elvéhez hasonlóan[3]. Formálisan:

$$POS(x, P) = \max_{z \in P} \{s(x, z)\}. \tag{2}$$

Ekkor az x szekvenciát a P osztályba soroljuk – és az x szekvencia tulajdonságaira a P osztályba tartozó szekvenciák jellemzőiből következtethetünk – ha a fenti érték elér egy bizonyos előre meghatározott küszöbszámot, különben nem. Megjegyezzük, hogy a gyakorlatban e küszöbszám kalibrálása gyakran priori információk alapján vagy heurisztikus algoritmusokkal történik.

Tekintsük az alábbi

$$LRA(x, P, N) = \left\{ \frac{POS(x, P)}{POS(x, N)} \right\}$$

hányadost, ahol N a P halmaz komplementere a T halmazra. A módszer ötlete a képfeldolgozásból származik [26], viszont a [27] publikációban összehasonlító tesztek elvégzésével mi mutattuk meg először, hogy ezzel a módszerrel jelentősen jobb fehérje-osztályozás érhető el a POS módszerhez képest, valamint megmutattuk, hogy ez a módszer azonos a Likelihood-ratio [3] módszerrel [27]. Teszteredmények a 2. táblázatban találhatók.

2. táblázat. Az LRA és a POS módszerek összehasonlítása a BLAST és a SW hasonlósági függvényekkel és a ROC analízissel kiértékelve.

		SCOP	3PGK	COG
SW	POS	0.850	0.791	n.a
	LRA	0.932	0.944	n.a.
BLAST	POS	0.825	0.792	0.987
	LRA	0.892	0.941	0.999

A SCOP adatbázist a [28] publikációból vettük, a 3PGK és a COG fehérje adatbázisok az általunk elkészített adatbázisból valók. A módszereknél használt paraméter beállítások a disszertációban találhatók meg.

Megjegyezzük, hogy az $POS(.,.)$ módszer helyett tetszőleges olyan algoritmus használható, amely becslést ad arra, hogy egy objektum mennyire tartozik egy osztályhoz. Viszont a gyakorlatban nem minden módszer alkalmazható tetszőleges osztályra. Például az $POS(., P)$ mérésére egy profile-HMM[29]

[4]Megjegyezzük, hogy a gyakorlatban a fehérjék egy osztálya bizonyos biológiai tulajdonságuk – például azonos térszerkezetük vagy azonos funkciójuk – alapján összetartozó fehérjék halmaza.

modell alkalmas lenne, de a gyakorlatban profile-HMM nem kalibrálható az N osztályhoz, mert az többfajta szekvencia-csoport egyvelege.

Tömörítő-alapú távolság (CBD) vizsgálata fehérje-szekvenciákon

Az információs távolság egy a kilencvenes években kifejlesztett univerzális metrika sztringek távolságának mérésére [30], amely a következő képlettel kapható meg:

$$E(x,y) = \frac{\max\{K(x \mid y), K(y \mid x)\}}{\max\{K(x \mid \lambda), K(y \mid \lambda)\}}, \tag{3}$$

ahol λ az üres sztringet jelöli és $K(a \mid b)$ az a sztring feltételes Kolmogorov-bonyolultsága a b sztringre vonatkozóan, ami a legrövidebb olyan bináris programnak a hosszát jelöli, amelyet a b paraméterrel futtatva az a sztringet adja eredményül. Mivel a Kolmogorov-bonyolultság nem kiszámítható Turing értelemben – és így az információs távolság sem – ezért ennek becslésére hagyományos tömörítő algoritmusokat alkalmaznak az alábbi formulával[31]:

$$CBD(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}, \tag{4}$$

ahol $C(z)$ jelöli a z szekvenciának a C tömörítővel (például gzip, arj) tömörített hosszát és xy az x és az y szekvenciák konkatenációját.

A CBD távolságmértéket (tudomásom szerint) először filogenetikus fák felépítésére használták 2001-ben [31]. Ezt követte például hierarchikus klaszterezésre a [32; 33], természetes nyelvek hierarchikus osztályozására a [34; 35] és zene osztályozásra a [36] publikációk. A [37] cikkünk volt az első, amely a fehérje szekvenciák osztályozása szempontjából vizsgálta a CBD módszer alkalmazhatóságát, majd később a [38] cikkben további vizsgálatok eredményét közöltük.

Az eredményeink 3 pontban foglalhatók össze:

(1) A CBD metrikákat alkalmaztuk fehérje-szekvenciák ábrázolására az (1) módszerrel és a tömörítő algoritmusok közül egy adaptív Huffman tömörítőt[39] (AH), egy LZW (Lempel-Ziv-Welch) tömörítőt [40], egy PPM (Prediction by Partial Matching) tömörítőt [41], és a GenCompress (GC) tömörítőt [42] (amely speciálisan DNS illetve fehérje szekvenciák tömörítésére lett kifejlesztve) használtuk. Majd az általunk elkészített adatbázist használva az elterjedtebb osztályozó algoritmusokat kiértékeltük, és az eredmények a 3. táblázatban találhatók. Az algoritmusoknál alkalmazott paraméterbeállítások a disszertációban megtalálhatók. A kísérleti eredmények azt mutatják, hogy a CBD metrikák kevésbé teljesítenek olyan jól, mint a rész-szekvencia alapú összehasonlító algoritmusok, mint például a legjobbnak tartott SW. Ez magyarázható azzal, hogy a SW tartalmaz biológiai tudást, ami lényegében az aminosav-helyettesítési mátrixba van kódolva, míg a CBD alapú távolságmódszerek nem alkalmaznak semmilyen priori tudást [37; 38].

(2) A CBD metrikák hatékonyságán, szabadon fogalmazva, azt értjük, hogy az egy osztályba tartozó fehérjeszekvencia-párhoz kisebb távolságértéket ad, míg a különböző osztályba tartozókhoz nagyobb távolságértéket rendel. Ennek mérésére a ROC analízist használtuk. Megvizsgáltuk, hogy hogyan változik a CBD metrikák hatékonysága a fehérje-szekvenciákat reprezentáló ábécé méretének függvényében. A szekvencia adatbázist az általunk elkészített adatbázisból vettük, tömörítő algoritmusnak egy Huffman, egy LZW, egy PPM és a GenComress tömörítőt használtuk. Aminosav-ábécé csökkentéskor az azonos típusú aminosavakat, míg az ábécé növelésekor a betű-ketteseket és

3. táblázat. Fehérje-szekvencia osztályozási eredmények CBD metrikával.

Módszer név[1]	1NN	SVM	RF	LogReg	ANN	Átlag
AH	.711	.877	.824	.751	.800	.793
GC	.644	.775	.691	.753	.769	.726
LZW	.751	.856	.821	.718	.794	.788
PPM	.798	.851	.800	.813	.583	.823
SW[2]	.815	.942	.823	.906	.888	.875

A teszteléshez a szekvenciákat az általunk elkészített adatbázis 'SCOP40' rekordjából vettük. Az eredmények kiértékeléséhez ROC analízist alkalmaztunk. Az alkalmazott paraméter-beállítások a disszertációban megtalálhatók. [1](1) EFM módszerben alkalmazott mértékek. CBD esetében az alkalmazott tömörítő eljárás nevét tüntettük fel. [2]Az összehasonlítás végett feltüntetjük azt az esetet is, amikor SW-nal alkalmaztuk az EFM módszert.

betű-hármasokat (bi-gram, tri-gram) ábrázoltuk egy új karakterrel [43]. Az azonos típusú aminosav-csoportokat az aminosavak kémiai tulajdonságai alapján határozzák meg [44]. Itt jegyezzük meg, hogy az ábécé csökkentésével a szekvencia reprezentálása egyszerűsödik – azaz információt vesztünk el – és ebben az esetben tekinthetünk a CBD-re úgy, mintha veszteséges tömörítővel alkalmaznánk. Az elvégzett összehasonlító tesztek alapján eredményül azt kaptuk, hogy általánosságban nem volt megfigyelhető számottevő összefüggés a szekvenciákat reprezentáló ábécé mérete és az osztályozási eredmények között. Például bizonyos tömörítőknél javulás volt elérhető, ha a szekvenciákat reprezentáló ábécé méretét csökkentettük, míg más típusú tömörítőkkel romlott a CBD metrikák hatékonysága [38]. Részletes vizsgálatok a disszertációban találhatók.

(3) Megvizsgáltuk, hogy a CBD metrikát egy gyors, de alkalmazás-specifikus heurisztikával kombinálva a kapott összetett mérték milyen hatékonyságú fehérje-osztályozásban. A mértékek kombinálását a alábbi módon végeztük el:

$$F(x,y) = \left(1 - \frac{BLAST(x,y)}{BLAST(x,x)}\right) \cdot CBD(x,y), \tag{5}$$

ahol $BLAST(x,y)$ jelöli az x és az y szekvenciák távolságát a BLAST módszerrel. Azért a BLAST algoritmust választottuk, mert a gyakorlatban ez az algoritmus a legelterjedtebb fehérje szekvenciák hasonlóságának mérésére. A képlet első tagja a BLAST mértéket normalizálja [0,1] intervallumba és konvertálja a hasonlósági értéket távolságértékké, hogy konzisztens legyen a CBD távolságmértékkel. A tömörítő algoritmusok közül a PPM és az LZW algoritmusokat vizsgáltuk, és a tesztek elvégzéséhez a szekvencia adatbázist a [28] publikációból vettük. E kombinált módszerrel sikerült jobb osztályozási eredményt elérni a legjobbnak tartott, de költséges Smith-Waterman módszernél, és jobb eredmény érhető el két rejtett-Markov model alapú módszernél (Fisher kernel[45], SAM[46]) is. Teszteredmények a 4. táblázatban találhatók. Megjegyezzük, hogy ez a módszer nem tartalmaz semmilyen állítható paramétert, így bizonyára kifinomultabb módszerekkel további javítás érhető el [37].

4. táblázat. Osztályozási eredmények fehérje domain adatbázison[a] több fehérje szekvencia hasonlító módszerrel, Az osztályozáshoz az SVM módszert, míg a kiértékeléshez a ROC analízist használtuk.

SW	BLAST	LZW	PPMZ	LZW + BLAST	PPMZ + BLAST	SVM-Fisher[a]	SAM[b]
0.901	0.884	0.869	0.787	0.907	0.884	0.686	0.657

[a]Az adatbázist a [28] publikációból vettük. [b]SVM-Fisher módszer [45], [c]SAM egy profile-HMM alapú osztályozó [46]. Az SVM-Fisher és a SAM algoritmussal kapott eredményeket a [28] publikációból vettük, nem mi értékeltük ki.

Ekvivalencia-tanulás fehérje-szekvencia osztályozásra

Különböző adatosztályok (fehérjecsoportok) nagyban különbözhetnek jellemzőikben, mint például: osztályméret, osztályon belüli és osztályok közötti hasonlóság, stb. Egy tetszőleges módszer, amelyik jobban működik egy osztályon, kevésbé működhet jól egy másikon, és viszont. A hasonlóság ill. távolság függvények tanulása felügyelt módon egy általános módszertant nyújthat hasonlósági ill. távolság függvények specifikus osztályokhoz igazításához.

Távolság metrika tanulására (TMT) az első felügyelt módszert a [47] publikáció adta, amely módszer mátrix diagonalizáció és saját érték/vektor felbontáson alapszik. TMT finomításával például a [48–52] publikációk foglalkoznak. A bioinformatika területén hasonlóság ill. távolság tanulásával [53; 54] publikációk javasolnak módszereket fehérjék közötti biológiai kölcsönhatás tanulására. Az idevonatkozó szakirodalom részletesebb áttekintése a disszertációban található. Szekvencia illesztések tanulására a [55; 56] publikációk adnak eljárásokat a szekvencia-illesztés helyettesítési mátrixának optimalizálására. Szerző legjobb tudomása szerint a [57; 58] saját publikációi az elsők, amelyek fehérje szekvenciák hasonlóságának tanulásával foglalkozik.

A Szerző eredménye egy fehérje-szekvencia hasonlóságának tanulására kifejlesztett módszer, amely egy kétosztályos tanuló-algoritmuson és ekvivalens fehérjepárok (amelyek egy osztályba tartoznak) és nem-ekvivalens párok (amelyek különböző osztályba tartoznak) használatán alapszik, és amellyel az elvégzett összehasonlító tesztek alapján mondhatjuk, hogy jobb fehérje osztályozási eredményeket kapunk [57]. Továbbá megmutattuk, hogy bizonyos feltételek mellett metrika vagy kernel[5]-függvény tanulható meg. A metrikák és a kernel függvények tekinthetők olyan távolság- és hasonlóság-függvényeknek (külön-külön), amelyek további speciális matematikai tulajdonsággal bírnak, amelyek kihasználása más algoritmsokban további lehetőségeket adhat. Ezt az általunk kifejlesztett módszertant fogjuk most vázolni.

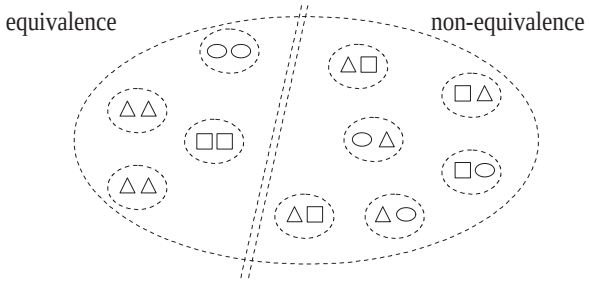
Mivel az osztályozási feladat célja, általában, egy F függvény megtanulása úgy, hogy $F(s) = y$, ahol y jelöli az s objektum osztálycímkéjét, ezért egy ilyen osztályozás természetes módon definiál egy δ relációt az objektumok felett a következőképpen:

$$\delta(s,t) = \begin{cases} 1 & F(s) = F(t) \text{ azaz } s \text{ és } t \text{ ugyanabba az osztályba tartozik,} \\ 0 & \text{különben,} \end{cases}$$

Könnyen megmutatható, hogy ez a reláció reflexív, szimmetrikus és tranzitív, azaz ekvivalenciareláció. Ennek a relációnak a tanulását ekvivalencia-tanulásnak (EL) neveztük el [57] és az 2. ábra szemlélteti.

Mivel a gépi tanulási módszerek többsége numerikus vektorokon alapszik, kérdés az objektumpárok reprezentálása. Jelöljön $P_C^\phi : S \times S \to R^n$ egy olyan leképezést, amely szekvencia-párokhoz egy valósértékű, n-dimenziós vektort rendel, formálisan: két tetszőleges s és t szekvenciákra: $P_C^\phi(s,t) = C(\phi(s), \phi(t))$, ahol S jelöli a szekvenciák halmazát. $\phi : S \to R^n$ egy leképezés, amely bármely szekvenciához egy rögzített komponensű valós értékű vektort rendel, amely lehet például az EFM (1) módszer, $C : R^n \times R^n \to R^n$ egy kétváltozós, vektorokon értelmezett operátor, ún. kompozíciós operátor. A disszertációban alkalmazott kompozíciós operátor módszereket a 5. táblázat definiálja és foglalja össze. Ekkor az ekvivalencia-tanulás felírható az $EL(s,t) = F(P_C^\phi(s,t)) = y$ függvény tanulásként, ahol $y = \{1, 0\}$ jelöli azt, hogy ugyanabba az osztályba tartoznak-e vagy nem (azaz

[5] A kernel függvény egy kétváltozós, szimmetrikus és pozitív definit függvény, amely tekinthető a belsőszorzat (skalárszorzat) egy általánosításának is.



2. ábra. Az ekvivalencia-tanulás alapelve. Az ekvivalencia-tanulás objektumpárok kétosztályos osztályozásának tanulási feladata. Az ábrán objektum párok láthatók, amelyek csoportját itt $\triangle$, $\circ$ és $\square$ jelekkel ábrázoltuk, és csoportosítottuk őket annak alapján, hogy az objektum pár mindkét tagja egy osztályba tartozik (ekvivalens) vagy nem (nem ekvivalens).

5. táblázat. Az alkalmazott kompozíciós operátorok.

Név	Formula		
Sum	$C_+(u,v)$	$=$	$u+v$
Product	$C_\bullet(u,v)$	$=$	$u \cdot v$
Quadratic	$C_Q(u,v)$	$=$	$(u-v)^2$
Hellinger	$C_H(u,v)$	$=$	$(\sqrt{u}-\sqrt{v})^2$
Dombi	$C_D(u,v)$	$=$	$u \cdot v + (1-u)\cdot(1-v)$

Itt az egyszerűbb jelölés miatt a fenti operátorok a vektorokon koordinátaként definiáltuk, azaz bármely $u, v \in R^n$ vektorokra, $(u \cdot v)_i = u_i v_i$, $(\sqrt{v})_i = \sqrt{v_i}$ and $(v^n)_i = (v_i)^n$.

$y = \delta(s,t)$). A továbbiakban tekintsünk a δ relációra, mint egy 0-1 értékű hasonlóság függvényre, és a hozzá tartozó EL tanulásra tekintsünk, mint hasonlóság függvény tanulására. Az EL módszer paraméterei a tanuláshoz használt F osztályozó algoritmus, a ϕ vektorizációs módszer, C kompozíciós operátor illetve ezen módszerek paraméterei. Most megvizsgáljuk, hogy az EL módszerrel hogyan kapható kernel függvény.

Tekintsük az $f(z) = \langle w, z\rangle + b$ döntési felületet (hipersíkot) két osztály között az n-dimenziós valós térben, ahol z n-dimenziós vektor, w a döntési felület normál vektora, b egy konstans és $\langle x, y\rangle = \sum_{i=1}^n x_i y_i$ az n-dimenziós vektorok skalár szorzatát jelöli. Az SVM módszer azt az f döntési felületet határozza meg a két osztály között, amelyhez a két osztály legközelebbi elemei maximális távolságra vannak tőle[6]. A kapott döntési felület felírható az

$$f(z) = \sum_{i=1}^{n} \alpha_i \langle z, x_i\rangle + b \tag{6}$$

alakban, ahol x_i ún. szupport vektorok, míg α_i a hozzá tartozó Lagrange konstans. A tézis egyik fő eredménye, hogy megmutattuk, hogy ha az (6) egyenletet lecseréljük az alábbi függvények bármelyikével

$$SVK_{P^\phi_{C_\bullet}}(s,t) = \sum\nolimits_i \alpha_i \exp(\sigma\langle P^\phi_{C_\bullet}(s,t), x_i\rangle), \tag{7}$$

$$SVK_{P^\phi_{C_+}}(s,t) = \sum\nolimits_i \alpha_i \exp(\sigma\langle P^\phi_{C_+}(s,t), x_i\rangle), \tag{8}$$

$$SVK_{P^\phi_{C_D}}(s,t) = \sum\nolimits_i \alpha_i \exp(\sigma\langle P^\phi_{C_D}(s,t), x_i\rangle), \tag{9}$$

[6]Itt az egyszerűség kedvéért feltettük, hogy a két osztály adatai lineárisan szétválaszthatók.

6. táblázat. Az ekvivalencia-tanulással (EL) kapott osztályozási eredmények[a] összehasonlítása több különböző hasonlósági módszerrel és osztályozó algoritmussal.

Módszerek[b]	1NN	SVM	RF	ANN	LogReg	*Átlag*
BLAST	0.863	0.953	0.852	0.958	0.953	*0.921*
SW	0.861	0.953	0.866	0.955	0.948	*0.916*
LAK	0.860	0.955	0.876	0.956	<u>0.959</u>	*0.922*
EL[c]	<u>0.964</u>	<u>0.966</u>	<u>0.948</u>	<u>0.963</u>	0.927	*<u>0.953</u>*

Minden oszlopban aláhúzással jelöltük a legnagyobb értéket. [a]Az osztályozás kiértékeléséhez ROC-analízist használtunk. [b]Az oszlopban fehérje-hasonlósági módszerek nevei találhatók, míg a sorban osztályozó algoritmusoké. [c]Az EL tanulásához az RF osztályozót, a C_D kompozíciós operátort alkalmaztuk, míg a szekvenciák vekotrizálásához az EFM (1) módszert, amelyhez a tanuló halmazban lévő szekvenciákat használtuk.

$$SVK_{P^{\phi}_{C_Q}}(s,t) = \sum\nolimits_i \alpha_i \exp(-\sigma\langle P^{\phi}_{C_Q}(s,t), x_i\rangle), \quad (10)$$

$$SVK_{P^{\phi}_{C_H}}(s,t) = \sum\nolimits_i \alpha_i \exp(-\sigma\langle P^{\phi}_{C_H}(s,t), x_i\rangle), \quad (11)$$

ahol σ egy tetszőlegesen beállítható pozitív paraméter és a tanulásához egyosztályos SVM-et vagy nemnegatív legkisebb négyzetek módszerét alkalmazzuk akkor a fenti SVK függvények mindegyike kernel függvény lesz a fehérje szekvenciák felett. Az SVK tanulásával kapott teszteredmények a disszertációban találhatók.

Az alábbi 6. táblázat osztályozási eredményeket tartalmaz. A táblázat sorai különböző hasonlósági módszerekhez tartoznak, míg az oszlopok osztályozó algoritmusokhoz. A teszteléshez az osztályozási feladatokat az általunk kifejlesztett adatbázisból vettük. Minden oszlopban kiemeltük a legjobb osztályozási értéket, amelyek mutatják, hogy az EL módszerünkkel jobb osztályozási eredmények kaphatók. A tesztelés során alkalmazott paraméterek a disszertációban találhatók meg. A módszer viselkedését további szempontok alapján is megvizsgáltuk viszont ezen tapasztalatok leírása mellett itt helyhiány miatt eltekintünk, de a disszertációban megtalálhatók.

Kálmán szűrő DNS-chip adatokra

A disszertáció utolsó fejezete tartalmában eltér az előzőektől. E fejezet DNS-chipekkel foglalkozik, amelyek többféle típusú (általában daganatos vagy egészséges, vagy többféle daganatos betegségű) szövetekből (sejtekből) nyert génexpressziós adatokat tartalmaznak [59]. A DNS chip adatai mátrix formában rendezettek, ahol a mátrix egy oszlopa egy mintához, míg egy sora egy génhez tartozik, és a mátrix egy eleme mutatja, hogy az adott génről egy adott mintában mennyi fehérje íródik át (azaz keletkezik, amit génexpressziós értéknek neveznek). Ekkor a minták tekinthetők egy olyan térben, ahol a jellemzők (feature-ök) a génhez tartozó expressziós értékek. A feladat itt az, hogy minél pontosabban azonosítsuk azokat a géneket (mátrix sorait), amelyekkel a különböző típusú minták a legjobban osztályozhatók (mátrix oszlopai). Átfogalmazva: válasszuk ki azt a legszűkebb génhalmazt, amellyel még jól osztályozhatók a különböző típusú minták. Egy ilyen kiválasztott génhalmazt biomarkernek nevezzük és ezek ismeretében pontosabb diagnózis és hatékonyabb kezelés állítható fel a beteg számára.

A DNS chipek gyakorlati alkalmazását megnehezíti, hogy a módszerrel kapott génexpressziós ada-

tok zajjal terheltek [60]. Az egyik eredményünk, hogy megmutattuk, hogy a Kálmán-szűrő (KF) [61] sikeresen használható DNS chipek több típusán (affymetrix, cDNS, oligonukleotid), mert az eredményül kapott szűrt DNS chipeken az osztályozó algoritmusokkal jobb osztályozási eredményeket értünk el már kisebb génhalmaz használatával is, mint az eredeti szűrés nélküli DNS chipeken. Továbbá a módszer előnye a PCA zajszűrővel [62] szemben az, hogy a Kálmán szűrő az adatokat az eredeti jellemző (feature) térben tartja. Ezáltal a jellemzők továbbra is a génekkel azonosíthatók és a szűrés után ezek felhasználhatók az eredmények biológiai értelmezésére, míg ez a PCA esetében nem mondható el.

A Kálmán-szűrő paramétereinek beállításának kulcsszerepe van a szűrő sikeres alkalmazásában. Erre megadtunk egy új automatikus módszert, amelyet most fogunk ismertetni. A Kálmán-szűrő leírása mellett eltekintünk terjedelmi okok miatt, viszont a szűrő leírása a disszertáció 7.2 fejezetében vagy a [61]-ban megtalálható. Jelöljük a Kálmán-szűrő két paraméterét Q-val és R-rel, ahogy a disszertációban is jelöltük. Mindkettő egy-egy mátrix, a Q paraméter a modellezési hibát reprezentálja, míg az R paraméter a mérési zajt. Ennek a két paraméternek a helyes beállítása kulcsfontosságú a jó eredmények elérése érdekében. Az általunk javasolt paraméterezés a következő. Legyen $Q = \overline{Q} + qI$ és $R = \overline{R} + rI$, ahol $\overline{Q}$ az osztályon belüli adatok kovariancia mátrixa és $\overline{R}$ az osztályok közötti kovariancia mátrix. A másik két additív tag egy-egy regularizációs paraméter a túltanulás elkerülése végett. Az összehasonlító tesztek elvégzése alapján azt találtuk, hogy az affymetrix típusú DNS chipek esetében az $r = \overline{R_{11}}$ és $q = \overline{Q_{11}}$ választások míg, ritka mátrixú, cDNS vagy oligonukleotid típusú DNS chipek esetében az $r = tr(\overline{R})$ és $q = tr(\overline{Q})$ paraméter-választással kaptunk jó eredményeket, ahol A_{11} az A mátrix első sorának az első eleme, $tr(A)$ az A mátrix nyomát jelöli és I az egységmátrixot.

A módszert teszteltük 7 publikus, daganatos eredetű többosztályos DNS-chip adatbázisokon és az elterjedtebb osztályozó módszereket lefuttattuk a szűrt és a nem szűrt adatokon egyaránt. Teszteredmények a 7. táblázatban találhatók, amelyek igazolják, hogy a Kálmán filter javítja az osztályozhatóságot. Az adatbázisok forrása és az osztályozó algoritmusokhoz használt paraméterbeállítások a disszertációban megtalálhatók. A disszertációban új biomarkerek azonosítását is tárgyaljuk, annak bizonyítására, hogy a szűrt expressziós adatokkal már kisebb elemszámú génhalmaz is predikciós

7. táblázat. Osztályozási eredmények az eredeti és a KF-rel szűrt DNS chip adatbázison több osztályozó módszerrel. Az osztályozási eredmények kiértékeléséhez ROC analízist használtunk.

Adatbázis	SVM			ANN		1NN		RF	
neve	Eredeti	PCA[a]	KF	Eredeti	KF	Eredeti	KF	Eredeti	KF
ALL-AML	0.99	0.99	0.99	0.97	0.99	0.73	1.00	0.92	0.95
Breast Cancer	0.88	0.81	0.70	0.67	0.74	0.23	0.68	0.64	0.68
Lung Cancer	1.00	0.99	0.99	1.00	0.99	0.59	0.99	0.99	0.99
MLL	1.00	1.00	1.00	1.00	1.00	0.87	1.00	0.92	0.98
Leukeamia	0.97	0.96	0.98	0.90	0.98	0.60	0.88	0.94	0.96
SRBCT	0.99	0.99	1.00	0.99	1.00	0.66	1.00	0.99	1.00
Tumours	0.95	0.91	0.94	0.90	0.94	0.72	0.92	0.84	0.87

A táblázat sorai különböző adatbázisokhoz, míg az oszlopok különböző osztályozási algoritmusokhoz tartoznak. Megjegyezzük, hogy itt a táblázat adatai alapján az SVM módszer esetében mintha a KF nem hozna javulást, de a disszertációban lévő további vizsgálatok alapján az SVM esetében KF-rel már kisebb elemszámú biomarkerrel jobb eredmény érhető el, mint az eredeti adatokon. [a]Az SVM esetében a PCA módszerrel kapott zajszűrést is közöljük.

potenciállal bír az osztályozásra nézve, illetve három különböző grafikai ábrázolást alkalmaztunk annak demonstrálására, hogy az egyes osztályok szemmel láthatóan elkülönülnek egymástól.

Konklúzió

A disszertáció témája gépi tanulási módszerek alkalmazása a fehérje-osztályozásban. Röviden összegezve az eredményeket az mondható el, hogy minél több alkalmazható biológiai ismeretet tartalmaz az osztályozó algoritmus, annál pontosabb eredményeket ad. Ebből a nézőpontból foglaljuk most össze az előbbi fejezeteket.

A dolgozat első része írja le a fehérje-osztályozási eredményeket. Általánosságban elmondhatjuk, hogy az osztályozási eredmények kevésbé függnek az alkalmazott gépi tanulási módszertől, sokkal inkább a fehérje-szekvencia ábrázolásától. Itt a biológiai tudást a fehérje-hasonlító módszerek tartalmazzák, amit a fehérjék reprezentálására használtunk az ún. „Empirical Feature Mapping" módszerrel. A legtöbb biológiai információt a 3D térszerkezeti (pl. DALI, PRIDE) és a szekvencia-illesztésen alapuló módszerek (pl. BLAST, Smith-Waterman, Needleman-Wunsch) tartalmazzák. A CBD módszerek nem vesznek figyelembe biológiai információt, ezek csupán rész-szekvenciák ismétlődésén és eloszlásán alapulnak, de még ezeket a részszekvenciákat sem súlyozzák biológiai fontosságuk szerint. Az n-gram (pl. karakter kettes ($n = 2$), és hármas ($n = 3$) összetétel) módszer szintén gyenge fehérje-reprezentációt ad, mivel az aminosav-összetétel alapján nem mondható meg sem a fehérje térszerkezete, sem a funkciója. Ahogy az várható, a CBD és az n-gram technikák általános teljesítménye fehérje-osztályozásban gyengébb, mint a 3D térszerkezeten vagy szekvencia-illesztésen alapuló hasonlósági algoritmusoké, viszont sebességük nagyságrendekkel gyorsabb.

Hasonló következtetést vonhatunk le az LRA esetében is. Itt a hasonlósági algoritmusnak a fehérje-osztályozási képessége egy pozitív P osztályra vonatkozóan nem csupán a P osztály elemeit, hanem az N osztály elemeit is tekintetbe veszi, amely további információt jelent az osztályozás számára.

A felügyelt tanulómódszerek előnye az, hogy mindig az aktuális részproblémához igazítják a modell paramétereit. Az ekvivalencia-tanulási módszer azt tanulja meg, hogy vajon két szekvencia egy osztályba tartozik-e vagy sem, és itt a szekvenciapárok egymáshoz való viszonyából nyerhető további ismeret. A Kálmán-szűrő egy zaj-szűrő algoritmus, amely a paramétereit ugyancsak felügyelt tanulási módon állítja be, így a Kálmán-szűrő az aktuális osztályozandó csoporthoz igazodik.

Sajnos a felügyelt módszerek érzékenyek a tanuló- és a tesztadatok eloszlására és könnyen túltanulhatnak, ami csökkenti a tanuló algoritmus általánosító képességét. A felügyelt kereszt-validációs módszer egy valósabb becslést ad arra, hogy egy tanuló algoritmus mennyire képes felismerni egy új részcsoportot a már ismert csoporton belül, ami tekintehő a osztályozó algoritmus általánosító képességének is.

Az eredmények a disszertációban további kérdéseket vethetnek fel. Vajon a Kálmán-szűrő alkalmazható-e fehérje hasonlósági-mértékekre? Tudunk-e használható biológiai tudást építeni a CBD módszerekbe? Meg tudjuk-e ezt tenni úgy, hogy a CBD megőrizze a metrikatulajdonságokat? Láthattuk, hogy két gyors hasonlósági mérték egyszerű kombinálásával jobb osztályozási eredmény érhető el a legjobbnak tartott, de lassú Smith-Watermannál. Vajon érhető-e el jobb eredmény a 3D szerkezeti hasonlító algoritmusnál a gyors mértékek kombinálásának tanulásával?

Eddigi tapasztalataink alapján fontosnak tartjuk a biológiai tudás reprezentálását az algoritmusban, ám a fordított irány is hasonlóan fontos lehet. Ugyanis ezen biológiai információk matematikai

formalizálásával és modellezésével más nézőpontból kaphatunk betekintést a biológiai folyamatok megértéséhez.

Az eredmények tézisszerű összefoglalása

A következőkben összegezzük a Szerző eredményeit négy fő tézispontba rendezve. A 8. táblázat tartalmazza a kutatásokból származó publikációkat, valamint azok tartalmának viszonyát az egyes tézispontokhoz.

I. A fehérje-osztályozási adatbázis

a) A Szerző részt vett egy ingyenesen hozzáférhető fehérje- osztályozási adatbázis elkészítésében, amelyen az újonnan kifejlesztett gépi tanulási algoritmusok és szekvencia-hasonlósági módszerek kiértékelhetők és összehasonlíthatók. Az osztályozási feladatok száma eléri a 9500-at. A Szerző feladata volt a napjainkban leginkább használatos state-of-the-art gépi tanulási és hasonlósági algoritmusok beállítása és kiértékelése az összes osztályozási feladaton. Az algoritmusok paraméterei és az eredmények az adatbázis weblapján elérhetők, így ezek az adatok felhasználhatók összehasonlító tesztek elvégzésekor új algoritmusok kifejlesztésénél [9].

b) A Szerző kifejlesztett egy általános matematikai keretet a hierarchikusan rendszerezett fehérje adatbázisokra az osztályozási feladatok létrehozásakor a tanuló- és teszt elemek kiválasztására. Ezt a módszert felügyelt kereszt-validációnak nevezte el. Eredményként egy megbízhatóbb becslést kapunk arra, hogy egy betanított osztályozó algoritmus mennyire képes felismerni egy csoportban egy új, de még nem látott részcsoportot. Ez tekinthető az osztályozó általánosító képességének is. A szerző megtervezte és elvégezte a szükséges teszteket az összehasonlításhoz és azt tapasztaltuk, hogy az így kapott osztályozási feladatok nehezebbek, de véleményünk szerint sokkal valósabb eredményt adnak a hagyományos, kereszt-validációs módszerekkel szemben (pl. 10-fold, leave one out) [7].

A Szerző megvizsgálta, hogy hogyan módosulnak az osztályozási eredmények, ha a hierarchikus rendezést kihasználva kategóriákat kihagyunk a negatív osztályból az előfeldolgozó és a tanuló algoritmusok futási idejének rövidítése és az ún. „class-imbalanced" probléma elkerülése céljából. A Szerző megtervezte és kiértékelte az összehasonlító teszteket, amelyek alapján nem javasolja a módszer alkalmazását, mert a kapott negatív halmaz nem reprezentálja a valós negatív fehérje-univerzumot [7]. Habár ezen eredmény negatívnak tekinthető, érdemesnek tartjuk megjegyezni, mert a hierarchikusan rendezett adatbázisok jellemzése osztályozási feladatok készítése szempontjából véleményünk szerint így teljesebb.

II. Likelihood ratio közelítés

a) A szerző megvizsgálta, hogy a likelihood ratio módszer alkalmazása hogyan változtatja meg a hasonlósági módszereken alapuló osztályozást. A Szerző megtervezte és kiértékelte az összehasonlító teszteket, és arra a megállapításra jutott, hogy ezzel a módszerrel jelentős javulás érhető el fehérje-osztályozás terén [27].

III. Tömörítő alapú távolság (CBDs)

a) A Szerző megvizsgálta a tömörítő alapú távolságok (CBD) viselkedését fehérje-szekvenciákon. A CBD metrikák hatékonyságán azt értjük, hogy ugyanabba az osztályba tartozó fehérje-szekvenciákhoz kicsi, míg különbözőkbe tartozókhoz nagy távolságértéket rendel. A kísérleti eredmények azt mutatják, hogy a CBD metrikák kevésbé teljesítenek olyan jól, mint a rész-szekvencia alapú összehasonlító algoritmusok, mint például a legjobbnak tartott Smith-Waterman. Ez magyarázható azzal, hogy a Smith-Waterman tartalmaz biológiai tudást, ami lényegében az aminosav-helyettesítési mátrixba van kódolva, míg a CBD alapú távolságmódszerek nem alkalmaznak semmilyen priori tudást [37; 38]. A Szerző megvizsgálta, hogy a CBD metrikák hatékonysága hogyan változik a fehérje-szekvenciákat reprezentáló ábécé méretének függvényében. A fehérje-ábécé csökkentéskor az azonos típusú aminosavakat, míg az ábécé növelésekor a betű-ketteseket és betű-hármasokat (bi-gram, tri-gram) ábrázoltuk egy új karakterrel. A Szerző megtervezte és kiértékelte a kísérleteket, ami alapján számottevő összefüggés nem volt megfigyelhető sem aminosav-szekvenciákon sem nukleotid-szekvenciákon [38]. Ezek az eredmények nem pozitív eredmények, de mint észrevételek segíthetik a bioinformatikai alkalmazásokat, tehát nem érdemes figyelmen kívül hagyni.

b) A Szerző megvizsgálta, hogy a CBD metrikát egy gyors, de alkalmazás-specifikus heurisztikával (BLAST) kombinálva a kapott összetett mérték milyen hatékonyságú fehérje-klasszifikációban. A Szerző megtervezte és kiértékelte az összehasonlító teszteket. Eredményül megállapíthatjuk, hogy kombinált CBM és BLAST mértékkel közel azonos osztályozási eredmény érhető el, mint a legjobbnak tartott, de költséges Smith-Waterman módszerrel, továbbá jobb eredmények érhetők el, mint két rejtett-Markov model alapú (Fisher kernel, SAM) mértékkel [37].

IV. Ekvivalencia tanulása

a) A Szerző bevezette az ekvivalencia-tanulás fogalmát, mint egy új típusú hasonlóságtanulást. A Szerző megtervezte és kiértékelte az összehasonlító teszteket, amelyek azt mutatják, hogy ekvivalencia-tanulással jobb fehérje-osztályozást sikerült elérni [57].

b) A Szerző új típusú kernel függvények osztályát – Szupport Vektor Kernel (SVK) – is definiálta, és elméleti úton megmutatta, hogy az SVK kielégíti a kernel-függvényekre vonatkozó feltételeket. A szerző megadott két módszert is az SVK tanulására, megtervezte és elvégezte az összehasonlító teszteket. [57; 58].

V. Zajszűrés DNS-chipekre.

a) A szerző hozzájárulása ehhez a tanulmányhoz az összehasonlító tesztek megtervezése és kiértékelése a génkiválasztási és minta osztályozási feladatokra DNS-chip adatbázisokon. A szerző tervezett egy automatikus paraméter-behangolási módszert is a Kálmán-szűrőhöz, amely közös és oszthatalan eredménye az első szerzővel [63].

A disszertációban szereplő eredmények több cikkben kerültek publikálásra. A 8. táblázat összegzi, hogy mely tézispontot mely publikáció közli.

	[9]	[7]	[27]	[37]	[38]	[57]	[58]	[63]
I	a	b						
II			a					
III				a,b	a			
IV						a	a,b	
V								a

8. táblázat. Tézispontok és a Szerző publikációinak viszonya.

Hivatkozások

[1] Arthur M. Lesk. *Introduction to Bioinformatics*. Oxford University Press, 2002.

[2] Zoe Lacroix and Terence Critchlow. *Bioinformatics – Managing Scientific Data*. Morgan Kaufmann Publishers, 2003.

[3] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley Interscience, 2 edition, 2000.

[4] M. Gribskov and N. Robinson. Use of receiver operating characteristic (roc) analysis to evaluate sequence matching, comput. chem. (20), pp. 25–33, 1996.

[5] K. Tsuda. Support vector classifier with asymmetric kernel function in european symposium on artificial neural networks (esann), pp. 183–188, 1999.

[6] Hiroto Saigo, Jean-Philippe Vert, Nobuhisa Ueda, and Tatsuya Akutsu. Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682–1689, Jul 2004.

[7] Attila Kertész-Farkas, Somdutta Dhir, Paolo Sonego, Mircea Pacurar, Sergiu Netoteia, Harm Nijveen, Arnold Kuzinar, Jack Leunissen, András Kocsor, and Sándor Pongor. Benchmarking protein classification algorithms via supervised cross-validation. *J Biochem Biophys Methods*, 35:1215–1223, 2007.

[8] János Murvai, Kristian Vlahoviek, Csaba Szepesvári, and Sándor Pongor. Prediction of protein functional domains from sequences using artificial neural networks. *Genome Research*, 11(8):1410–1417, 2001.

[9] Paolo Sonego, Mircea Pacurar, Somdutta Dhir, Attila Kertész-Farkas, András Kocsor, Zoltán Gáspári, Jack A. M. Leunissen, and Sándor Pongor. A protein classification benchmark collection for machine learning. *Nucleic Acids Research*, 35(Database-Issue):232–236, 2007.

[10] J.D. Pollack, Q. Li, and D.K. Pearl. Taxonomic utility of a phylogenetic analysis of phosphoglycerate kinase proteins of archaea, bacteria, and eukaryota: insights by bayesian analyses. *Mol. Phylogenet. Evol.*, 35:420–430, 2005.

[11] L. H. Greene, T. E. Lewis, S. Addou, A. Cuff, T. Dallman, M. Dibley, O. Redfern, F. Pearl, R. Nambudiry, A. Reid, I. Sillitoe, C. Yeats, J. M. Thornton, and C. A. Orengo. The cath

domain structure database: new protocols and classification levels give a more comprehensive resource for exploring evolution. *Nucleic Acids Res*, 35(Database issue), January 2007.

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

[13] A. Andreeva, D. Howorth, and C. Brenner. Scop database in 2004: refinements integrate structure and sequence family data, 2004.

[14] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3):403–410, October 1990.

[15] Temple F. Smith and Michael S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[16] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–53, 1970.

[17] Jean-Philippe Vert, Hiroto Saigo, and Tatsuya Akutsu. Local alignment kernels for biological sequences. In Bernhard Schoelkopf, Koji Tsuda, and Jean-Philippe Vert, editors, *Kernel Methods in Computational Biology*, Cambridge, MA, 2004. MIT Press.

[18] Zoltán Gáspári, Kristian Vlahovicek, and Sándor Pongor. Efficient recognition of folds in protein 3d structures by the improved pride algorithm. *Bioinformatics*, 21(15):3322–3323, 2005.

[19] L. Holm and J. Park. Dalilite workbench for protein structure comparison. *Bioinformatics*, 16(6):566–567, June 2000.

[20] Christopher M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, Oxford, UK, 1996.

[21] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[22] S.K. Remlinger. Introduction and application of random forest on high throughput screening data from drug discovery, 2003.

[23] J C Rice. Logistic regression: An introduction. In B Rhompson, editor, *Advances in social science methodology*, volume 3, pages 191–245. JAI Press, Greenwich, 1994.

[24] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–3402, Sep 1997.

[25] W. R. Pearson. Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods Enzymol*, 183:63–98, 1990.

[26] David Claus and Andrew W. Fitzgibbon. Reliable fiducial detection in natural scenes. In Tomás Pajdla and Jiri Matas, editors, *ECCV (4)*, volume 3024 of *Lecture Notes in Computer Science*, pages 469–480. Springer, 2004.

[27] László Kaján, Attila Kertész-Farkas, Dino Franklin, Neli Ivanova, András Kocsor, and Sándor Pongor. Application of a simple likelihood ratio approximant to protein sequence classification. *Bioinformatics*, 22(23):2865–2869, 2006.

[28] Li Liao and William Stafford Noble. Combining pairwise sequence similarity and support vector machines for remote protein homology detection. In *RECOMB '02: Proceedings of the sixth annual international conference on Computational biology*, pages 225–232, New York, NY, USA, 2002. ACM.

[29] Anders Krogh and Søren Kamaric Riis. Hidden neural networks. *Neural Comput.*, 11(2):541–563, 1999.

[30] Ming Li and Paul Vitanyi. *An introduction to kolmogorov complexity and its applications*. Springer-Verlag, 2 edition, 1997. read.

[31] M. Li, J. H. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang. An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, 17(2):149–154, Feb 2001.

[32] R. Cilibrasi and P. M. B. Vitanyi. Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005.

[33] Alexander Kraskov, Harald Stögbauer, Ralph G. Andrzejak, and Peter Grassberger. Hierarchical clustering using mutual information. *CoRR*, q-bio.QM/0311037, 2003.

[34] Dario Benedetto, Emanuele Caglioti, and Vittorio Loreto. Zipping out relevant information. *Computing in Science and Engg.*, 5(1):80–85, 2003.

[35] Ming Li, Xin Chen, Xin Li, Bin Ma, and Paul Vitányi. The similarity metric. In *SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 863–872, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics.

[36] Rudi Cilibrasi, Paul Vitányi, and Ronald De Wolf. Algorithmic clustering of music based on string compression. *Comput. Music J.*, 28(4):49–67, 2004.

[37] András Kocsor, Attila Kertész-Farkas, László Kaján, and Sándor Pongor. Application of compression-based distance measures to protein sequence classification: a methodological study. *Bioinformatics*, 22(4):407–412, 2006.

[38] Attila Kertész-Farkas, András Kocsor, and Sándor Pongor. The application of the data compression-based distances to biological sequences. In Frank Emmert-Streib and Matthias Dehmer, editors, *Information Theory and Statistical Learning*, Lecture Notes in Computer Science. Springer, 2008.

[39] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. John Wiley and Sons, Inc., 1991.

[40] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, May 1977.

[41] J. Cleary and I. Witten. Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications*, 32(4):396–402, Apr 1984.

[42] Xin Chen, Sam Kwong, and Ming Li. A compression algorithm for DNA sequences and its applications in genome comparison. In *RECOMB*, page 107, 2000.

[43] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, July 1999.

[44] Edward Susko and Andrew J. Roger. On reduced amino acid alphabets for phylogenetic inference. *Mol Biol Evol*, 24:2139–2150, Sep 2007.

[45] T. Jaakkola, M. Diekhaus, and D. Haussler. Using the fisher kernel method to detect remote protein homologies. *7th Intell. Sys. Mol. Biol.*, pages 149–158, 1999.

[46] K. Karplus, C. Barrett, and R. Hughey. Hidden markov models for detecting remote protein homologies. *Bioinformatics*, 14(10):846–856, 1998.

[47] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning, with application to clustering with side-information, 2003.

[48] Ivor Tsang and James Kwok. Distance metric learning with kernels, 2003.

[49] James T. Kwok and Ivor W. Tsang. Learning with idealized kernels. In Tom Fawcett and Nina Mishra, editors, *ICML*, pages 400–407. AAAI Press, 2003.

[50] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 209–216, New York, NY, USA, 2007. ACM.

[51] Kilian Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1473–1480. MIT Press, Cambridge, MA, 2006.

[52] Alexander Zien and Cheng Soon Ong. Multiclass multiple kernel learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 1191–1198, New York, NY, USA, 2007. ACM.

[53] Jean-Philippe Vert, Jian Qiu, and William S Noble. A new pairwise kernel for biological network inference with support vector machines. *BMC Bioinformatics*, 8(Suppl 10):S8, 2007.

[54] Y. Yamanishi, J.-P. Vert, and M. Kanehisa. Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, 20(1):363–370, 2004.

[55] Maricel Kann and Richard A. Goldstein. Optima: A new score function for the detection of remote homologs. In Concettina Guerra and Sorin Istrail, editors, *Mathematical Methods for Protein Structure Analysis and Design*, volume 2666 of *Lecture Notes in Computer Science*, pages 99–108. Springer, 2003.

[56] Hiroto Saigo, Jean-Philippe Vert, and Tatsuya Akutsu. Optimizing amino acid substitution matrices with a local alignment kernel. *BMC Bioinformatics*, 7:246, 2006.

[57] Attila Kertész-Farkas, András Kocsor, and Sándor Pongor. Equivalence learning in protein classification. In Petra Perner, editor, *MLDM*, volume 4571 of *Lecture Notes in Computer Science*, pages 824–837. Springer, 2007.

[58] József Dombi and Attila Kertész-Farkas. Using fuzzy technologies for equivalence learning in protein classification. *Accepted for publication in Journal of Computational Biology*, 2008.

[59] M. Schena. *DNA microarrays: A practical approach*, volume 205 of *Practical Approach Series*. Oxford Univ. Press., Oxford, 1999.

[60] Y. Tu, G. Stolovitzky, and U. Klein. Quantitative noise analysis for gene expression microarray experiments. *Proc Natl Acad Sci U S A*, 99(22):14031–14036, Oct 2002.

[61] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*, D(82):35–45, 1960.

[62] I. T. Jolliffe. *Principal Component Analysis*. Springer, second edition, October 2002.

[63] János Z. Kelemen, Attila Kertész-Farkas, András Kocsor, and László G. Puskás. Kalman filtering for disease-state estimation from microarray data. *Bioinformatics*, 22(24):3047–3053, 2006.