VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

PREDIKCE SEKUNDÁRNÍ STRUKTURY PROTEINŮ POMOCÍ CELULÁRNÍCH AUTOMATŮ

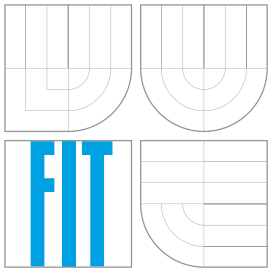
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. VLADIMÍR BRIGANT

BRNO 2013

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

PREDIKCE SEKUNDÁRNÍ STRUKTURY PROTEINŮ POMOCÍ CELULÁRNÍCH AUTOMATŮ

PREDICTION OF SECONDARY STRUCTURE OF PROTEINS USING CELLULAR AUTOMATA

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. VLADIMÍR BRIGANT

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAROSLAV BENDL

BRNO 2013

Abstrakt

Tato práce popisuje návrh metody predikce sekundární struktury proteinů založenou na celulárních automatech (CA) – CASSP. Optimální parametry modelu a přechodové funkce jsou získany pomocí evolučního algoritmu. Predikční model využíva pouze statistických vlastností aminokyselin, takže je velice rychlý. Dosažené výsledky byly porovnány s výsledky existujících metod. Byla také otestováná společná predikce navrženého systému CASSP s existujícím nástrojem PSIPRED. Nepodařilo se však dosáhnout výsledků, ktoré by tento existujíci nástroj převyšovali. Částečné zlepšení se dosáhlo při predikci pouze motivů sekndární struktury α-helix, co může pomoci v případe, že požadujeme co nejpřesenjší predikcii právě těchto motivů. K navrženému systému bylo také vytvořeno webové rozhraní.

Abstract

This work describes a method of the secondary structure prediction of proteins based on cellular automaton (CA) model – CASSP. Optimal model and CA transition rule parameters are acquired by evolutionary algorithm. Prediction model uses only statistical characteristics of amino acids, so its prediction is fast. Achieved results was compared with results of other tools for this purpose. Prediction cooperation with a existing tool PSIPRED was also tested. It didn't succeed to beat this existing tool, but partial improvement was achieved in prediction of only α-helix secondary structure motif, what can be helful if we need the best prediction of α-helices. It was developed also a web interface of designed system.

Klíčová slova

sekundární struktura proteínů, celulární automat, proteinové predikce, evoluční algoritmus

Keywords

secondary protein structure, cellular automata, protein prediction, evolutionary algorithm

Citace

Vladimír Brigant: Predikce sekundární struktury proteinů pomocí celulárních automatů, diplomová práce, Brno, FIT VUT v Brně, 2013

Predikce sekundární struktury proteinů pomocí celulárních automatů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Bendla.

.....
Vladimír Brigant
14. ledna 2014

Poděkování

Chcel by som poďakovať Ing. Jaroslavovi Bendlovi za cenné rady pri tvorbe tejto práce. Veľké poďakovanie takisto patrí za prístup k výpočetným a úložným zariadeniam patriacim do národnej sieťovej infraštruktúry MetaCentrum, vybudovanej v rámci programu „Projekty rozsiahlej infraštruktúry pre výzkum, vývoj a inovácie“ (LM2010005).

Obsah

Kapitola 1

Úvod

Život na Zemi je založený na uhlíku. Chemické vlastnosti tohto prvku, ktorého tvorba by ani nezačala, keby „parametre“ vesmíru boli nastavené o trošku inak, umožňujú vytvárať dlhé uhlíkové polyméry, molekuly s uhlíkovou kostrou. Medzi tie, ktoré zabezpečujú základné funkcie života, patria nukleové kyseliny (prenos genetickej informácie), sacharidy a lipidy (zásobárne energie), a proteíny. Proteíny sú biopolyméry, komplexné molekuly, ich funkcia závisí na poradí aminokyselín, z ktorých sa skladajú. Význam proteínov je enormný, zabezpečujú vnútorné dýchanie, pohyb, či katalyzujú chemické reakcie.

Pokiaľ chceme pochopiť život do najmenších detailov, bez štúdia proteínov sa nezaobídeme. Podľa centrálnej dogmy molekulárnej biológie sa proteíny tvoria na základe génov v DNA. Ich funkcia je definovaná priestorovým usporiadaním jednotlivých aminokyselín (terciárna štruktúra) a naopak. Toto usporiadanie dokážu najpresnejšie (nie však na 100 %) určiť experimentálne metódy. Proteínov je však veľmi veľa a neefektívnosť časovo a finančne náročných experimentálnych metód podnietila vznik strojových predikčných metód štruktúry proteínov. Medzikrokom k terciárnej štruktúre proteínu je sekundárna štruktúra, ktorej správna predikcia významným spôsobom uľahčuje predikciu štruktúry priestorovej. A to je grom tejto práce – vylepšiť predikciu sekundárnej štruktúry proteínov. Bližší popis základných techník predikcie sekundárnej štruktúry proteínov sa nachádza v kapitole 2.

Použitým predikčným modelom je celulárny automat (CA) (viď kapitola 3), ktorý je tvorený mriežkou jednoduchých funkčných jednotiek (buniek). Každá bunka sa nachádza v jednom z viacerých, vopred definovaných stavov. Stav bunky sa môže počas behu CA meniť. Či a ako sa stav bunky zmení, závisí na prechodovej funkcii CA a na stavoch buniek v okolí aktuálnej bunky. Prechodová funkcia definuje správanie CA. Tento jednoduchý výpočetný model by mal zaistiť predovšetkým rýchlosť predikcie. Najväčším problémom CA je vhodné určenie prechodovej funkcie. Nie je možné „odskúšať“ všetky možné a určiť najlepšiu z nich, preto prichádzajú na scénu optimalizačné techniky, v našom prípade evolučný algoritmus, ktorý pri hľadaní suboptimálneho riešenia používa princípy evolučného výberu a genetiky (viac v kapitole 4).

Vlastným návrhom predikčného systému sa zaoberá kapitola 5. V kapitole 6 sa nachádza popis implementácie konzolovej aj webovej verzie systému. Implementovaný model je použitý k predikcii sekundárnej štruktúry proteínov aj ako samostatný systém, aj v spolupráci s nástrojom PSIPRED. Tomuto použitiu predchádza optimalizácia niektorých parametrov vytvoreného modelu. Popis experimentov, ich vyhodnotenie a porovnanie výsledkov s inými metódami ponúka kapitola 7. Záver (kapitola 8) obsahuje zhrnutie práce a výhľad do budúcna.

Kapitola 2

Proteíny a predikcia ich štruktúry

Jacob Berzelius, významný švédsky chemik, v roku 1838 navrhol názov „proteín“ pre zložité organické zlúčeniny bohaté na dusík, ktoré nachádzal v bunkách živočíchov a rastlín. Štruktúra a chémia proteínov sa vyvinula a doladila behom milárd rokov evolúcie. Z chemického hľadiska sú proteíny najzložitejšie a funkčne najdômyselnejšie známe molekuly. Sú tvorené sekvenciou aminokyselín, ktorých väčšina bola objavená v priebehu 19. storočia. Sú pospájané kovalentnou peptidovou väzbou o veľkosti približne 1,32 Å a tvoria väčšinu suchej hmotnosti bunky [3]. Vysoká početnosť funkcií, ktoré proteíny zaisťujú, pramení z obrovského počtu rôznych tvarov, ktorých môžu v priestore nadobúdať – funkcia sleduje štruktúru.

2.1 Štruktúra proteínov a genetický kód

Zásadným krokom v štúdiu proteínov bola práca Jamesa B. Sumnera z roku 1926, ktorý ukázal, že enzýmy možno kryštalizovať a izolovať. V roku 1953 Francis Crick a James D. Watson popísali štruktúru DNA a v roku 1958 prvý menovaný po prvý krát vyslovil slovné spojenie *Centrálna dogma molekulárnej biológie*, ktoré hovorí, že cesta k proteínom je: DNA → RNA → proteín. Prišlo sa na to, že trojice nukleotidov DNA (kodóny) sa mapujú na aminokyseliny [18], prišlo sa na genetický kód (viď obrázok 2.1). Inými slovami, sekvencia nukleotidov určitého génu, ktorý definuje konkrétny proteín, sa v procese translácie „preloží“ na sekvenciu aminokyselín (podrobný popis v [3]). Dá sa povedať, že genetický kód definuje sémantiku DNA reťazca.

GCA GCC GCG GCU	AGA AGG CGA CGC CGG CGU	GAC GAU	AAC AAU	UGC UGU	GAA GAG	CAA CAG	GGA GGC GGG GGU	CAC CAU	AUA AUC AUU	UUA UUG CUA CUC CUG CUU	AAA AAG	AUG	UUC UUU	CCA CCC CCG CCU	AGC AGU UCA UCC UCG UCU	ACA ACC ACG ACU	UGG	UAC UAU	GUA GUC GUG GUU	UAA UAG UGA
Ala	Arg	Asp	Asn	Cys	Glu	Gln	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val	stop
A	R	D	N	C	E	Q	G	H	I	L	K	M	F	P	S	T	W	Y	V	

Obrázok 2.1: Genetický kód (prevzaté z [3]). Pre väčšinu aminokyselín existuje viac kodónov, ktoré ich kódujú. Tri kombinácie nukleotidov sú „vyčlenené“ pre ukončovanie génov (tzv. stop-kodóny).

Spojením viacerých aminokyselín vzniká peptidový reťazec, zvyšky aminokyselín (reziduá) odstupujú od osi reťazca ako tzv. postranné reťazce. O vlastnostiach proteínov roz-

hoduje charakter reziduí aminokyselín a z toho vyplývajúce sily, ktoré medzi nimi pôsobia. Hydrofóbne (nepolárne) aminokyseliny sú priťahované k sebe, tj. do vnútra molekuly (ak sú vo vodnom prostredí), hydrofilné (polárne) sa naopak orientujú na povrch molekuly. Po denaturácii, rozpade natívnej priestorovej štruktúry, peptidového reťazca a následnom odstránení denaturačného rozpúšťadla, sa sekvencia aminokyselín zbalí späť do pôvodného tvaru. Z toho vyplýva, že úplná informácia potrebná k určeniu trojrozmerného tvaru proteínu je obsiahnutá v charaktere jeho aminokyselín a ich poradí v peptidovom reťazci. Štruktúra proteínov je pomerne zložitá, preto má zmysel definovať jej úrovne. Rozlišujeme 4 úrovne štruktúry proteínov – primárnu, sekundárnu, terciárnu a kvartérnu [3] (viď obrázok 2.2).

Primárna štruktúra. Na najnižšej úrovni, na úrovni molekúl, je sekvencia aminokyselín odvodzovaná na základe kódujúcej sekvencie nukleotidov DNA. Dlhú dobu sa sekvenovanie proteínov vykonávalo priamou analýzou ich aminokyselín. Prvým proteínom, ktorého sekvencia bola určená, je inzulín (v roku 1955). Vývoj rýchlych metód sekvenovania DNA v dnešnej dobe umožňuje oveľa jednoduchšiu, nepriamu sekvenáciu proteínov určením poradia nukleotidov v DNA [54].

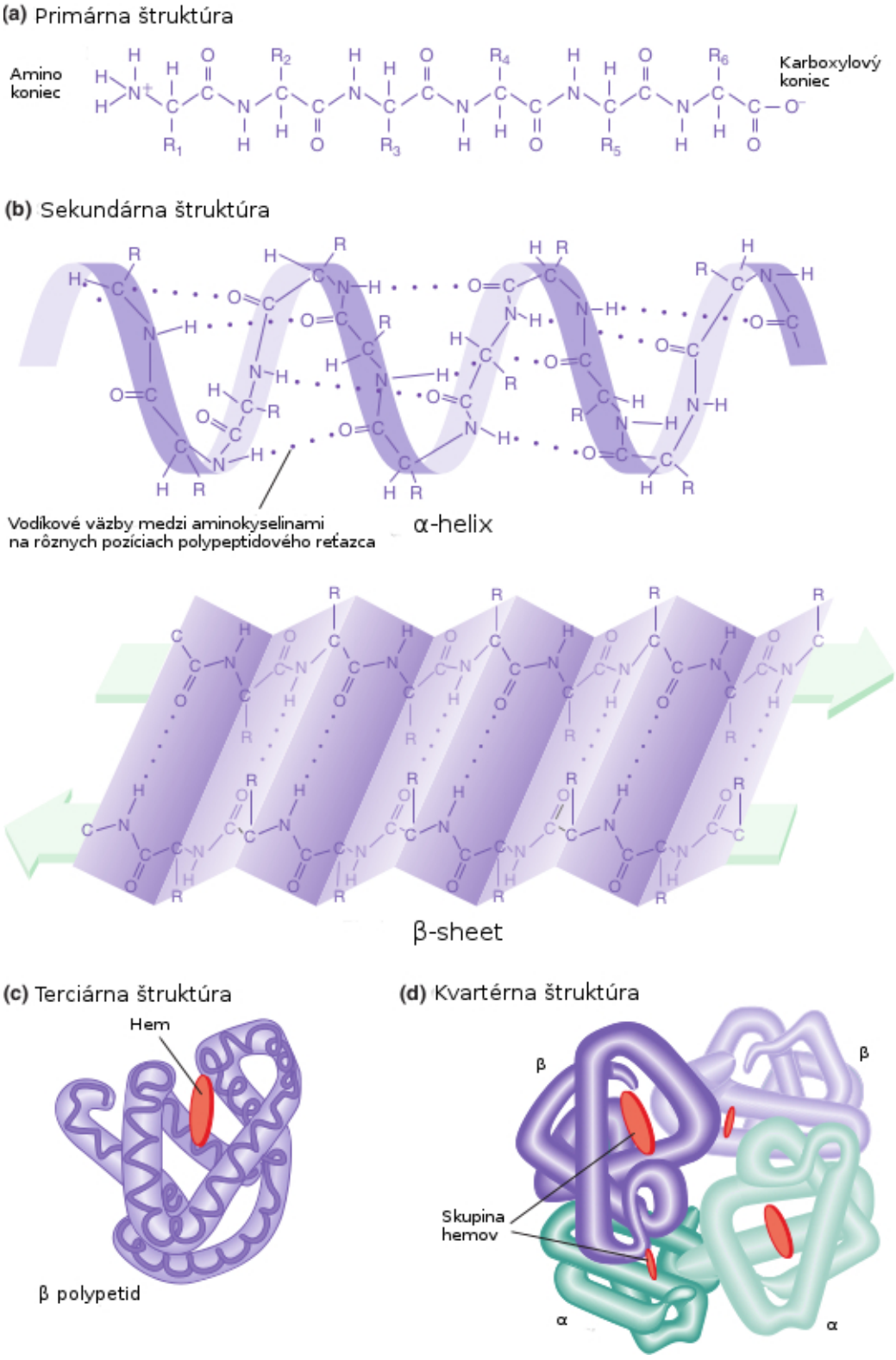
Sekundárna štruktúra. Pri porovnávaní trojrozmerných štrukúr rôznych proteínov vyšlo najavo, že napriek jedinečnosti celkovej konformácie každého proteínu je v nich možné objaviť 2 základné modely skladania. Oba druhy boli objavené asi pred 60 rokmi pri štúdiu vlasov a hodvábia. Prvým z nich je α-helix (H), nájdený v proteíne α-keratín, ktorý sa hojne vyskytuje v koži, vlasoch, nechtoch atď. Druhým typom je β-sheet (E), nájdený v proteíne fibroín, ktorý je hlavnou zložkou hodvábia. Aminokyseliny mimo nich sa označujú ako Coil (C). Oba štruktúrne elementy sú stabilizované vodíkovými mostíkmi. Jadra mnohých proteínov obsahujú rozsiahle oblasti β-sheetov. Tvoria sa zo susedných polypeptidových reťazcov, ktoré majú buď rovnakú alebo opačnú orientáciu, resp. sú paralelné alebo antiparalelné. α-helix vzniká, keď sa jednoduchý polypeptidový režazec ovíja okolo samého seba a tvorý tuhý valec. Vodíkový mostík vzniká medzi každou štvrtou peptidovou väzbou a spája skupinu $C = O$ jednej peptidovej väzby so skupinou $N - H$ inej peptidovej väzby. To dáva vznik pravidelnej skrutkovici s 3,6 aminokyselinovými zvyškami (reziduami) na jednu otáčku. Krátke úseky α-helixov sú obzvlášť hojné v proteínoch umiestnených v bunečných membránach, ako sú transportné proteíny a receptory.

Terciárna štruktúra. Konečná, priestorová konformácia polypeptidového reťazca. Zisťovanie terciárnej štruktúry je metodicky veľmi zložité, používa sa difrakcia röntgenových lúčov na kryštáloch proteínov, nukleárna magnetická rezonancia (NMR) alebo elektrónová mikroskopia [3]. Evolučne príbuzné proteíny majú veľké podobnosti v terciárnej štruktúre.

Kvartérna štruktúra. Niektoré proteíny sú zložené z väčšieho počtu menších molekúl (podjednotiek, protomérov), ktoré sú navzájom viazané nekovalentnými väzbami. Vzájomné priestorové usporiadanie týchto podjednotiek udáva kvartérnu štruktúru proteínu.

Štúdium konformácie, funkcie a evolúcie proteínov tiež prezradilo dôležitosť ďalšej organizačnej jednotky, ktorá sa líši od jednotiek doposiaľ popísaných. Touto jednotkou je *proteínová doména*, ktorá je tvorená ľubovolnou časťou polypeptidového reťazca, ktorá sa môže nezávisle zvinúť do kompaktnej stálej štruktúry. Doména obvykle obsahuje 50–350 aminokyselín a je modulárnou jednotkou. Z domén sú vytvorené všetky väčšie proteíny. Niekedy sa táto štruktúra nazýva *suprasekundárna*.

Molekuly proteínov sa zúčastňujú na všetkých základných životných procesoch. Mnohé bielkoviny sú multifunkčné, napríklad membránové imunoglobulíny imunocytov sú stavebnou súčasťou membrány a súčasne majú funkciu signálnu – rozpoznávajú „svoje“ antigény.



Obrázok 2.2: Úrovne štruktúry proteínu. (a) Primárna štruktúra. (b) Sekundárna štruktúra. Polypeptid môže formovať α-helix alebo β-sheet, ktorý má 2 polypeptidové segmenty usporiadané antiparalelne (znázornené šípkami). (c) Terciárna štruktúra. Hem je neproteínová kruhová štruktúra s atómom železe v strede. (d) Kvartérna štruktúra ilustrovaná proteínom hemoglobín, ktorý je zložený zo štyroch polypeptirových podjednotiek. Obrázok bol prevzatý z [35].

Podľa funkcie môžeme proteíny rozdeliť nasledovne [54]:

- **Stavebné bielkoviny** – sú súčasťou bunkových štruktúr. Informácia pre špecifické usporiadanie podjednotiek je obsiahnutá v štruktúre molekuly, v štruktúre väzbového miesta. Nie je potrebné dodávať ani energiu, pretože nadmolekulárny komplex má nižšiu voľnú energiu ako zmes nepospájaných podjednotiek.
- **Enzýmové bielkoviny** – enzýmové reakcie uskutočňujú takmer všetky chemické reakcie v bunke, a tým celý jej metabolizmus. Enzýmová katalýza je jednou z najdôležitejších funkcií proteínov. Enzýmy umožňujú priebeh aj tých chemických reakcií, ktoré by za podmienok, v ktorých môžu živé systémy existovať, vôbec prebiehať nemohli.
- **Informačné bielkoviny** – regulujú bunkové procesy a medzibunkové vzťahy. Molekuly proteínov hrajú v týchto informačných procesoch 2 role – vystupujú ako signály, ktoré prenášajú informáciu, a receptory, ktoré môžu signály prijímať a transformovať na iné signály.

2.2 Významné projekty súvisiace s analýzou proteínov

Potenciálu štúdia DNA, génov a ich produktov, proteínov, sú si vedomé aj vlády a každoročne investujú do výskumu množstvo finančných prostriedkov. Hlavným dotovateľom najväčších projektov je USA. V roku 1990 bol zahájený medzinárodný výskumný projekt s názvom *Projekt ľudského genómu* (HGP[1]). Cieľom projektu bola sekvenácia ľudského genómu a analýza zhruba 20 000–25 000 génov z fyzikálneho aj funkčného hľadiska. V prvých fázach bol riaditeľom vyššie spomínaný James D. Watson. V roku 2003 bola publikovaná konečná verzia výsledkov a v tom istom roku bol projekt úspešne ukončený [39].

V roku 2011 bol ukončený projekt s názvom *Projekt 1000 genómov* (1000 Genomes Project), ktorý za pár rokov osekvenoval viac než tisíc ľudských genómov rôznych národností, zdravých aj postihnutých jedincov, za účelom možnosti skúmať rôzne variácie v genóme.

Rýchlosť sekvenácie genómu sa zrýchľuje vysokým tempom. HGP za viac než 10 rokov získal sekvenciu genómu jediného človeka, dnešné metódy, nazývané tiež Next Generation metódy, sú schopné zistiť sekvenciu genómu za rádovo dni. Taktiež cena išla nadol z miliárd na menej než $10 000. Sekvencií dát je dostatok, no problémom je, že im príliš nerozumieme, resp. rozumieme len malej časti. Vznikla iniciatíva konkretizovaná do projektu ENCODE [9] [2], ktorého cieľom je nájsť a analyzovať všetky funkčné časti ľudského genómu. Ide o rýdzo americký projekt, pracuje na ňom niekoľko pracovísk, bolo doň investovaných približne $300 miliónov. V septembri 2012 bolo nárazovo publikovaných niekoľko desiatok prác v renomovaných vedeckých časopisoch. Jedným z výsledkov je, že nie je pravda, že väčšia časť DNA je nepotrebná, ale naopak, väčšina má určitú funkciu [59].

2.3 Predikcia sekundárnej štruktúry

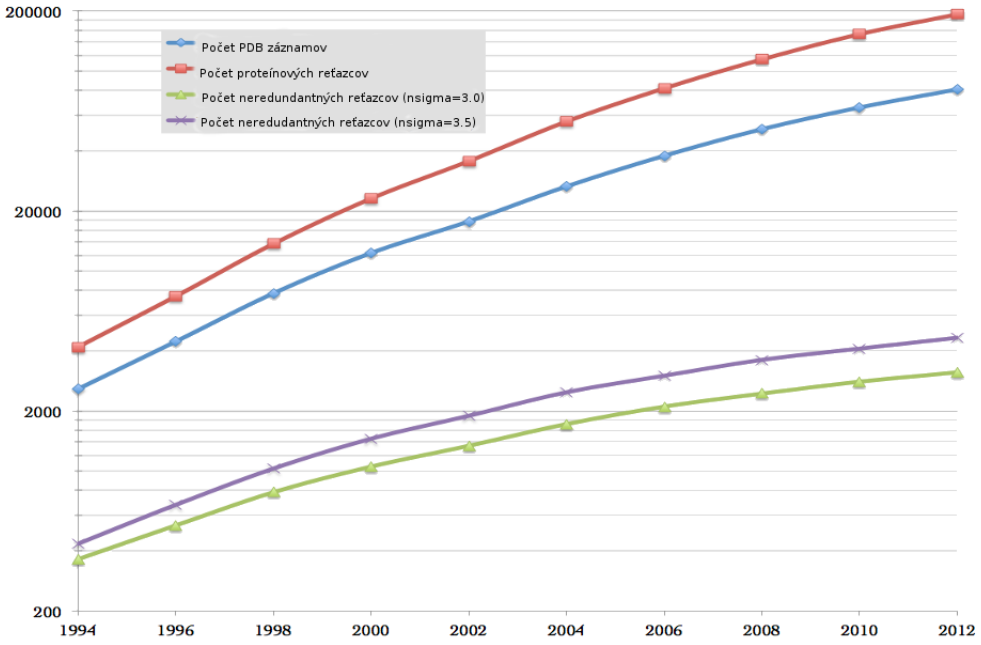
Držiteľ dvoch Nobelových cien (1954, 1962), Linus Pauling, bol prvý, kto predpovedal motívy sekundárnej štruktúry proteínov (SSP) [63]. Koncom 50-tych rokov bola po prvý krát

[1] Z angl. Human Genome Project, domovská stránka projektu: http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml.

[2] Z angl. ENCyclopedia Of DNA Elements, domovská stránka projektu: http://www.genome.gov/10005107.

experimentálne zistená štruktúra proteínu (pomocou röntgenovej kryštalografie[3]). Rozkvet experimentálneho zisťovania štruktúry proteínov však nastal až v 90-tych rokoch 20. storočia vďaka technickému pokroku. Obrázok 2.3 ukazuje nárast počtu záznamov v súčasnosti najväčšej databáze PDB [8]. Krivky zobrazených neredundantných záznamov nemajú tendenciu konvergencie, čo indikuje, že sa ešte ani neblížime úplnému pokrytiu proteínových štruktúr v rámci PDB [34]. V súčasnosti existuje približne 80 000 záznamov experimentálne zistených štruktúr proteínov, dostupná je teda databáza, na ktorú môžeme aplikovať rôzne techniky predikcie (štatistické, strojové učenie atď.).

Experimentálne metódy klasifikujú (štandardizovane podľa DSSP[4]) jednotlivé aminokyseliny do jednej z 8 tried, pri predikcii sa však v odbornej literatúre väčšinou používa redukcia na 3 základné: H, I, G $\rightarrow$ H (Helix), E, B $\rightarrow$ E (Beta Sheet), T, S $\rightarrow$ C (Coil). Vyčerpávajúci popis jednotlivých tried priniesli Wolfgang Kabsch a Christian Sander v [42]. SSP problém teda znie: majme proteínovú sekvenciu s aminokyselinami $\{S_1, S_2, \cdots, S_n\}$; urči pre každú S_i motív sekundárnej štruktúry – α-helix (H), β-sheet (E) alebo Coil (C). Na tento problém bolo aplikovaných veľa rôznych postupov, nasleduje klasifikácia a popis tých najúspešnejších a najprelomovejších.



Obrázok 2.3: S rokmi narastajúci počet PDB záznamov, proteínových reťazcov a neredundantných reťazcov. Úroveň redundancie sekvencií bola určovaná na základe tzv. HSSP funkcie [1]. Obrázok bol prevzatý z [34].

Podľa chronológie možno metódy predikcie SSP rozdeliť do 3 generácii [64] (viď tabuľka 2.1). Úspešnosť metód *1. generácie* nie je vysoká, čo je dané najmä neuvažovaním globálneho kontextu aminokyselinových reziduí ani evolučnej informácie extrahovanej z príslušnej

[3]Metóda zisťovania polohy jednotlivých atómov molekúl za pomoci röntgenového žiarenia, ktoré nám dovoluje „vidieť“ rádovo v jednotkách nanometrov.

[4]Z angl. Define Secondary Structure of Proteins.

rodiny homologických proteínových sekvencií. Prvé metódy sa zameriavali najmä na idenifikáciu α-helixov a boli založené hlavne na modeloch popisujúcich prechody medzi α-helixami a štruktúrami Coil [26]. Neskôr sa motív sekundárnej štruktúry pre určitú aminokyselinu určoval na základe štatistiky, ktorá uprednostňuje ten motív, ktorý je pre danú aminokyselinu najbežnejší, najpravdepodobnejší. Vtedajší nedostatok dát neumožňoval využiť plný potenciál štatistického prístupu. Medzi najvýznamnejšie techniky spadajúce do tejto generácie patrí metóda Chou-Fasman [14] a GOR (Garnier–Osguthorpe–Robson). Chou-Fasman metóda predpovedá motív sekundárnej štruktúry aktuálnej aminokyseliny na základe parametrov, ktoré vyjadrujú schopnosť predĺžiť alebo prerušiť v danom mieste motív sekundárnej štruktúry. GOR prediktor, považovaný za jedného z prvých realizovaných ako počítačový program, využíva poznatky Bayesovkej štatistiky a teórie informácie, ktoré sú aplikované na okno o veľkosti 17 aminokyselín (8 vľavo, 8 vpravo). Pre každú z 20 aminokyselín sa vypočítú frekvencie výskytu na danej pozícii v okne, na základe ktorých sa predikuje motív aminokyseliny v strede. Tento predikčný model však predpokladá, že neexistuje žiadna korelácia medzi konkrétnymi motívami sekundárnej štruktúry aminokyselín v okne 17 aminokyselín a predikovaným motívom v strede okna [30]. GOR II pracuje s rozšírenou databázou, inak je totožná so základnou metódou GOR. Tieto metódy vo vtedajšej dobe vykazovali vyššiu úspešnosť než bola reálna kvôli zahrnutiu trénovacích sekvencií do testovacích [43].

Metódy vyvinuté v 80-tych rokoch 20. storočia možno považovať za *2. generáciu* metód SSP. Vyšší výpočetný výkon dovoloval zložitejšie algoritmy predikujúce motív príslušnej aminokyseliny na základe okolitých aminokyselín v definovanom okne o veľkosti 3–51 aminokyselín. Modelovala sa, na rozdiel od metódy GOR, závislosť motívu predikovanej aminokyseliny na motívoch susedných aminokyselín. Túto koreláciu si uvedomili aj tvorcovia metódy GOR, keď publikovali druhé rozšírenie – GOR III, ktoré sa považuje za najvýznamnejšieho predstaviteľa 2. generácie. Revolúciou v SSP bola dostupnosť rozsiahlých rodín homologickych sekvencií. Kombinácia rozsiahlej databázy sekvencií a sofistikovaných počítačových techník viedla k prekonaniu úspešnosti 70 %.

Na *prelom 2. a 3. generácie* metód predikcie SSP možno zaradiť algoritmy rozšírené o ďalšie informácie o aminokyselinách, napr. tvar, veľkosť alebo fyzikálno-chemické vlastnosti. Patrí sem napríklad metóda najbližších susedov, kde sekundárna štruktúra sa určí na základe štruktúry najpodobnejších sekvenií [31], GOR V [45], ZPRED [33], či PREDATOR [25], ktorý používa metódu najbližších susedov skombinovanú s interakciou so vzdialenejšími aminokyselinami.

Generácia	**Obdobie**	**Úspešnosť [%]**	**Založené na**
1.	1960 – 1980	50–55	predispozíciach jednotlivých aminokyselín
2.	1980 – 1990	55–62	predispozíciach segmentov aminokyselín
3.	1990 – súčasnosť	70–80	evolučnej informácii (zarovnaní sekvencií)

Tabuľka 2.1: Generácie metód predikcie SSP.

Začiatkom 90-tych rokov minulého storočia začali vznikať metódy *3. generácie*, ktorých zásadnou vlastnosťou je využívanie evolučnej informácie – profilov proteínových rodín. Bolo totiž zistené, že všetky prírodne vytvorené proteíny o dĺžke viac než 100 reziduí a s viac než 35 % párovou zhodou má podobnú štruktúru, čo implikuje stabilitu v rámci divergencie sekvencií. Navyše, neutrálne mutácie sú veľmi nepravdepodobné, proteínov teda reálne existuje len malý zlomok, čoho dôsledkom je, úseky o dĺžke povedzme 17 reziduí implicitne

obsahujú dôležité informácie o globálnych interakciách, pretože profily viacnásobného zarovnania reflektujú evolučné obmedzenia [63]. Tieto metódy kombinujú silu väčších databáz a čoraz sofistikovanejších algoritmov založených na strojovom učení. Často sa používajú umelé neurónové siete, skryté Markovove modely, či klasifikátor SVM (z angl. Support Vector Machine). Klasifikátor SVM ukázal sa ako vhodný pre predikciu lokalizácie štruktúr coil, ktoré sú ťažko identifikovatelné štatistickými metódami [60]. Najlepšiu úspešnosť vykazujú metódy zamerané na špecifickú triedu proteínov. Medzi najznámejších predstaviteľov 3. generácie možno zaradiť PSIPRED [41], PHD [67], PHDpsi [66], PROF [65], JPred3 [17], či SSpro [61]. Táto práca sa snaží vylepšiť úspešnosť metódy *PSIPRED*, ktorá používa dvojstupňovú neurónovú sieť. Vstupom sú informácie o zarovnaní sekvencie pomocou nástroja PSI-BLAST [5]. Napriek jednoduchosti modelu je vykazovaná úspešnosť porovnatelná s ostatnými metódami 3. generácie [41].

Súčasný trend vo vývoji prediktorov SSP je vytvárať pomerne zložité modely zložené z viacerých prediktorov, ide o tzv. *konsenzuálne metódy*. Príkladom je hierarchický systém Bingru Yanga a spol., ktorý má 4 vrstvy a vykazuje úspešnosť presahujúcu 80 % [78]. Podľa štúdie z roku 2009 [6], ktorá jednotnou metodológiou analyzovala úspešnost 59 rôznych spôsobov predikcie SSP, existujúce algoritmické techniky nemôžu byť naďalej vylepšované iba pridávaním nehomologických sekvencií do tréningovej dátovej sady, tzn. nové nástroje SSP by sa mali zamerať na navrhovanie nových techník. Dôležité je podotknúť, že nie je možné dosiahnuť úspešnosť blížiacej sa k 100 %. Teoretický horný limit úspešnosti je okolo 90 %, sčasti kvôli neistej DSSP identifikácii blízko koncov sekundárnych štruktúr, kde sa menia lokálne konformácie [22]. Uvedená limitácia je taktiež spôsobená neschopnosťou predikcie sekundárnej štruktúry uvažovať terciárnu štruktúru. Sekvencia predikovaná ako α-helix stále môže nadobúdať konformáciu β-sheet, ak je lokalizovaná v rámci β-sheet regiónu proteínov a jeho postranné reťazce sú združené so susednými reťazcami. Lokálnu sekundárnu štruktúru taktiež môže meniť vlastná funkcia proteínu alebo aj prostredie, v ktorom sa nachádza.

Štruktúra proteínov závisí na nespočetnom množstve parametrov, ktorými sa je potrebné zaoberať, ak chceme dosahovať čoraz lepších výsledkov. Medzi tieto parametre, schopné signifikantným spôsobom zlepšiť výsledky predikcie, patrí napríklad počet kontaktov jednotlivých aminokyselín [47] alebo veľkosť chemických posunov [51].

2.4 Hodnotenie úspešnosti predikcie

Dôležitým prvkom pri vývoji metód predikcie SSP sú postupy merajúce ich úspešnosť. Medzi najpoužívanejšie úspešnostné miery patria Q_3 a SOV. Q_3 udáva pomer správne klasifikovaných reziduí proteínovej sekvencie do jednej z 3 tried (H, E, C) k všetkým reziduám [71]. Táto metodológia je jednoduchá a má určitú výpovednú hodnotu, no presne nezachytáva „užitočnosť“ predikcie elementov sekundárnej štruktúry pre následné využitie pri predikcii terciárnej štruktúry, pretože viac než správne určenie konformačného stavu jednotlivých reziduí je dôležitejšie určenie typu a lokalizácii elementov sekundárnej štruktúry [69].

SOV (z angl. Segment OVerlap) je miera, ktorá sa zameriava práve na správnu predikciu elementov sekundárnej štruktúry proteínov. Pôvodná SOV miera z roku 1994 (SOV'94) [68] nemala definovaný horný limit, čím ju nebolo možné priamo porovnávať s inými mierami (napr. s Q_3). V tejto práci používam upravenú verziu SOV (eliminujúcu nedostatky) definovanú v roku 1999 [69]. Vzhľadom k tomu, že túto mieru budem používať pri hodnotení úspešnosti SSP a jej netriviálnosti, nasledujúca časť sekcie prináša jej podrobný popis.

Nech s_1 a s_2 značia porovnávané segmenty sekundárnej štruktúry v konformačnom stave i (H, E alebo C). Segment s_1 je referenčný (typicky získaný experimentálne), s_2 predikovaný.

Nech (s_1, s_2) je pár prekrývajúcich sa segmentov, $S(i)$ množina všetkých prekrývajúcich sa párov segmentov v stave i a $S'(i)$ množina všetkých segmentov s_1 v stave i, pre ktoré neexistuje žiaden segment s_2 v stave i, ktorý by ich prekrýval, formálne:

$$S(i) = \{(s_1, s_2) : s_1 \cap s_2 \neq \varnothing \ \wedge \ s_1 \text{ a } s_2 \text{ sú v konformačnom stave } i\}$$
$$S'(i) = \{s_1 : \forall s_2, s_1 \cap s_2 = \varnothing \ \wedge \ s_1 \text{ a } s_2 \text{ sú v konformačnom stave } i\}$$

Definícia SOV miery:

$$SOV = \sum_{i\epsilon\{H,E,C\}} SOV(i) = \frac{100}{N} \sum_{i\epsilon\{H,E,C\}} \sum_{S(i)} \left[\frac{\text{minov}(s_1, s_2) + \delta(s_1, s_2)}{\text{maxov}(s_1, s_2)} \times len(s_1) \right], \quad (2.1)$$

kde N je normalizačná hodnota:

$$N = \sum_{i\epsilon\{H,E,C\}} N(i) = \sum_{i\epsilon\{H,E,C\}} \left[\sum_{S(i)} len(s_1) + \sum_{S'(i)} len(s_1) \right], \quad (2.2)$$

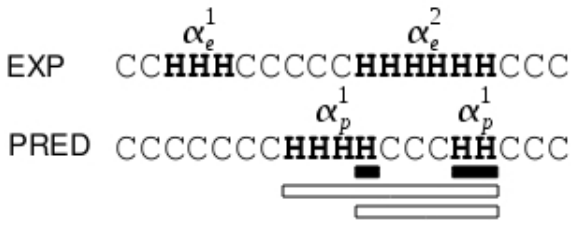
kde $len(s_1)$ vyjadruje počet reziduí v segmente s_1, $minov(s_1, s_2)$ dĺžku aktuálneho prekryvu segmentov s_1 a s_2, $maxov(s_1, s_2)$ rozsah „zjednotenia“ segmentov s_1 a s_2 a $\delta(s_1, s_2)$ je definované nasledovne:

$$\delta(s_1, s_2) = min\{maxov(s_1, s_2) - minov(s_1, s_2); minov(s_1, s_2); \lfloor len(s_1)/2 \rfloor; \lfloor len(s_2)/2 \rfloor\}, \quad (2.3)$$

kde $min\{x_1; x_2; \ldots; x_n\}$ značí minimum z n celých čísel.

Pre predstavu je uvedený príklad výpočtu SOV miery pre konformačný stav H pre dvojicu sekvencií zobrazenej na obrázku 2.4. Hodnota $SOV(H)$ sa na základe rovnice 2.1 vypočíta nasledovne:

$$SOV(H) = \frac{100}{6+6+3} \times \left(\frac{1+1}{10} + \frac{2+1}{6} \right) \times 6 = 28.0$$



Obrázok 2.4: Ilustrácia výpočtu $SOV(H)$. Čierne, resp. biele obdĺžniky reprezentujú *minov* resp. *maxov* hodnoty prekrývajúcich sa segmentových párov z experimentálne zistených (EXP) a predikovaných (PRED) štruktúr.

Kapitola 3

Celulárne automaty

Modelovanie zložitých fyzikálnych javov pomocou počítačových simulácii sa stalo základným nástrojom pri odkrývaní tajov našej Zeme. V prírode sa stretávame s rôznymi príkladmi správania, ktoré vykazuje emergenciu, teda vznik vlastností systému na globálnej úrovni na základe lokálnych interakcií a bez ich explicitnej definície v rámci jednotlivých elementoch systému alebo ich prepojeniach. Ide napríklad o kolónie hmyzu, sietnicu alebo imunitný systém [72]. Jedným z cieľov umelej inteligencie je odhaliť princíp emergentného správania ako takého. Medzi základné prístupy blížiace sa k tomuto cieľu patria agentné systémy, teória chaosu, či teória celulárnych automatov.

Koncept celulárneho automatu (CA) bol vynájdený už mnohokrát pod rôznymi názvami. V matematike ide o oblasť topologickej dynamiky, v elektrotechnike sú to iteračné polia, deti ich môžu poznať ako druh počítačovej hry [73]. Modelovanie pomocou CA je principiálne jednoduché, na základe lokálneho pôsobenia ich elementov je možné vykazovať požadované globálne správanie. V tejto kapitole bude model CA priblížený, načrtnuté historické pozadie jeho výskumu a uvedené krátke pojednanie o jeho aplikáčných doménach.

3.1 Stručná história výskumu celulárnych automatov

Koncept CA uzrel svetlo sveta v 40-tych rokoch 20. storočia vďaka dvojici amerických imigrantov maďarského, resp. poľského pôvodu – John von Neumannovi a Stanislawovi Ulamovi, ktorí tento koncept používali najmä pre výskum logiky života [56]. Von Neumann sa inšpiroval prácami W. McCullocha a W.Pittsa – otcov neurónových sietí [21]. Používal 2 D CA, ktorého bunky sa mohli nachádzať v 1 z 29 stavov, okolie bolo 5-bunkové (neskôr nazývané ako *von Neumannovo*). Tento muž, ktorý pracoval aj na projekte Manhattan[1], dokázal existenciu konfigurácie zloženej z približne 200 000 buniek, ktorá sa dokáže samoreprodukovať. Takýto CA môže simulovať Turingov stroj [29]. Po von Neumannovej smrti (1957) bol jeho dôkaz zjednodušovaný.

Edgar F. Codd vytvoril jednoduchší model s 8 stavmi [16], ktorý ale neimplementoval samoreprodukčné správanie. Tri roky po Coddovej práci, Edwin Roger Banks vytvoril elegantný 4-stavový CA v rámci svojej dizertačnej práce [7], ktorý bol schopný univerzálneho výpočtu, no opäť absentovala samorepredukcia. John Devore vo svojej diplomovej práci významne zredukoval zložitosť Coddovho návrhu, no samoreprodukčný proces si vyžadoval príliš dlhú pásku. Až Christopher Langton upravil Coddov model do podoby schopnej vytvárať tzv. Langtonove slučky reprodukujúce samé seba s minimálnym množstvom pot-

[1] Krycí názov pre utajený americký vývoj atómovej bomby počas 2. svetovej vojny.

rebných buniek, avšak za cenu absencie výpočetnej univerzality [48]. Ďalšími významnými naledovníkmi von Neumanna v štúdii celulárnych automatov boli hlavne A. Burks a jeho študent J. Holland, ktorý je však známejší z oblasti evolučných algoritmov.

V 60-tych rokoch 20. storočia, popri dobových, málo výkonných výpočetných zariadeniach, záujem o CA ustal. O značné spopularizovanie CA sa postaral Martin Gardner, keď v roku 1970 v magazíne *Scientific American* venoval svoj stĺpček celulárnemu automatu Johna Hortona Conwaya s názvom „Game of Life“ [28]. Išlo o 2 D CA, ktorý je pomocou veľmi jednoduchých pravidiel schopný vykazovať, prenesene povedané, známky života.

Začiatkom 80-tych rokov 20. storočia sa začala skúmať otázka, či sú CA schopné modelovať okrem globálnych aspektov nášho sveta aj zákony fyziky ako také. Priekopníkmi v tomto výskume boli Tomasso Toffoli a Edward Fredkin. Hlavnou tézou ich výskumu bola definícia takých fyzikálnych výpočetných modelov, ktoré obsahujú jednu z najzákladnejších vlastností mikroskopickej fyziky – reverzibilitu. Podarilo sa im vytvoriť modely obyčajných diferenciálnych rovníc, akými sú napríklad rovnice prúdenia tepla, vĺn, či Navier-Stokesove rovnice prúdenia tekutín [24].

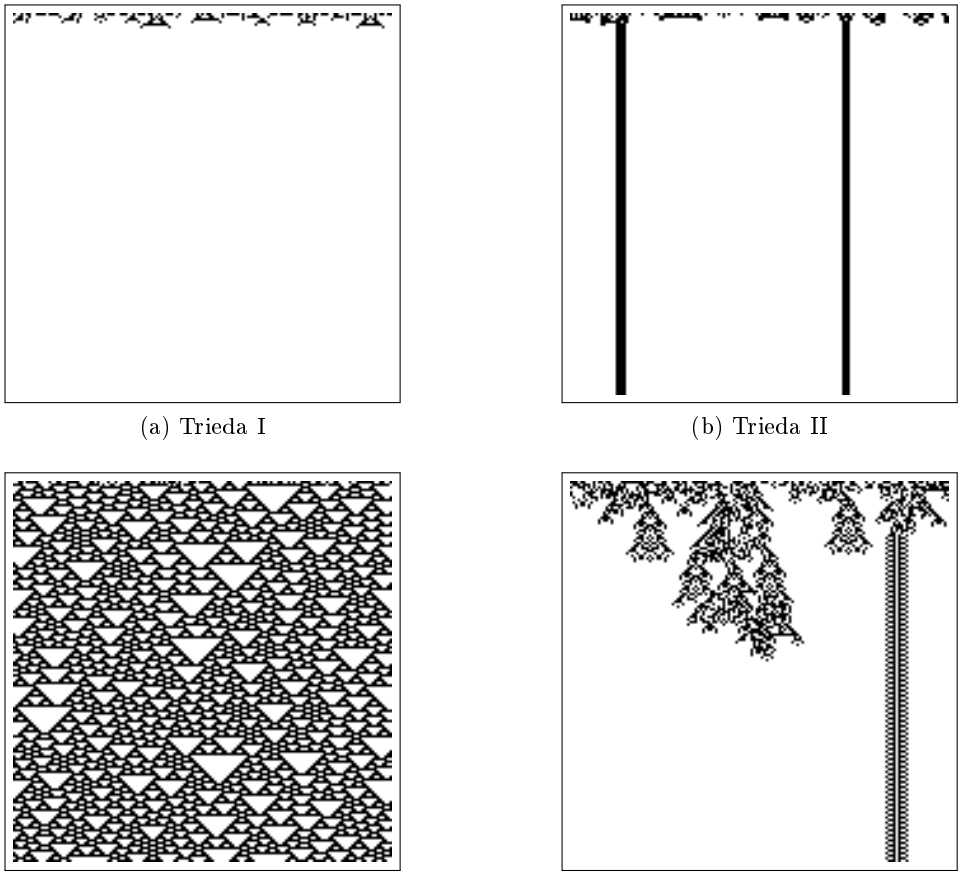
CA je taktiež užitočným modelom pre vetvu teórie dynamických systémov, ktorá sa zaoberá emergenciou takých javov ako je turbulencia, chaos, či fraktálnosť. Stephen Wolfram, o ktorom Terry Sejnowski, odborník na neurónove siete, hovorí ako o jednom z najinteligentnejších vedcov planéty [50], túto oblasť intenzívne a systematicky študoval. Definoval 4 triedy, do ktorých možno rozdeliť celulárne automaty a niektoré ďalšie výpočetné modely [75] (príklady 1 D celulárnych automatov vizualizuje obrázok 3.1). Pomocou týchto tried Wolfram popísal vzťah celulárnych automatov k dynamickým systémom (uvedeným v zátvorkých):

- **Trieda I** – počiatočné konfigurácie evolvujú do stabilného, homogénneho stavu. Akákoľvek náhodnosť počiatočnej konfigurácie mizne (limitné body).
- **Trieda II** – počiatočné konfigurácie konvergujú do jednoducho separovateľných periodických štruktúr (limitné cykly).
- **Trieda III** – vykazuje chaotické neperiodické vzory (chaotické správanie podobné podivným atraktorom).
- **Trieda IV** – vykazuje komplexné vzory (veľmi dlhé prechodné úseky, ktoré nemajú jasnú analógiu v spojitých dynamických systémoch).

Pre ďalší výskum CA bola obrovským prínosom Wolframova publikácia z roku 2002 s názvom „New Kind of Science“ [76], ktorá na 1197 stranách vyčerpávajúco analyzuje potenciál celulárnych automatov a ako názov napovedá, hovorí dokonca o novom druhu vedy. Wolframovým záverom je, že súčasná veda by sa na veci mala začať dívať z iného uhla, samozrejme v rámci možností. Všetky zložité systémy, ktoré dokážeme popísať, väčšinou popisujeme tak, že systém rozložíme na najmenšie elementy a skúmaním týchto častíc sa snažíme popísať systém ako celok. Dokážeme popísať správanie častíc, ale nedokážeme popísať, akým spôsobom tieto častice spolupracujú, akým spôsobom dokážu vytvoriť zložitý systém schopný komplexného správania, komplexného z nášho pohľadu.

3.2 Vlastnosti modelu

Všetky počítačové programy možno považovať v princípe za celulárne automaty, pretože počítač pracuje s obmedzenou aritmetikou aj pamäťou. Väčšina CA však používa stavový



(a) Trieda I

(b) Trieda II

(c) Trieda III

(d) Trieda IV

Obrázok 3.1: Wolframove triedy. Zobrazené sú 1 D celulárne automaty, kde horizontálna os popisje konfiguráciu modelu v určitom čase t a vertikálna os popusje postupné časové kroky (vzrastajúce zhora dole). Bunky celulárnych automatov sú binárne, teda môžu nadubúdať 1 z 2 stavov, okolie je 5-bunkové.

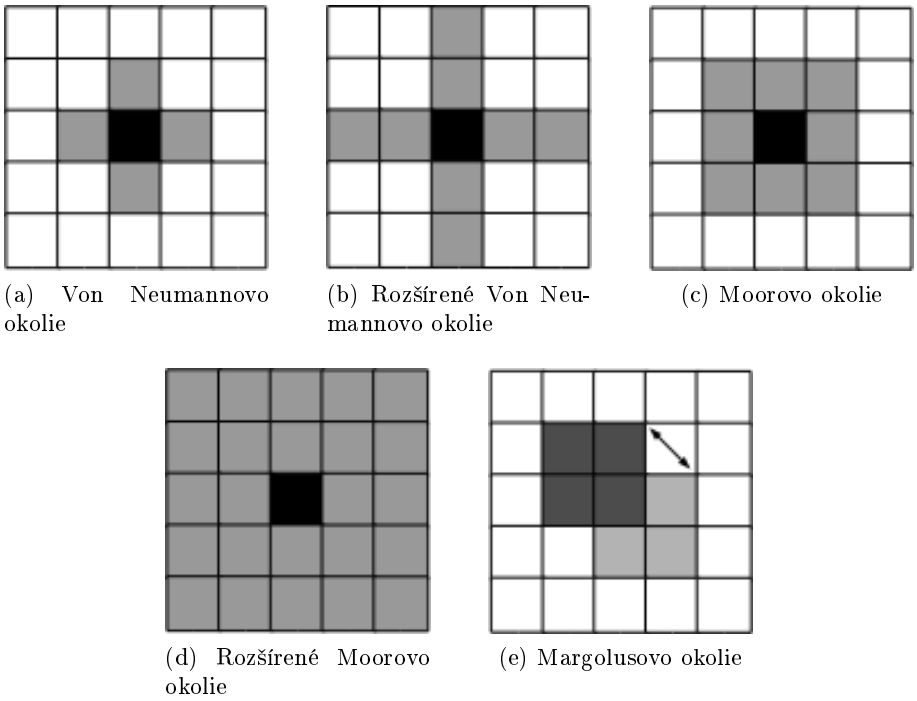
priestor redukovaný na pár stavov (často len 2 stavy – 0/1) [23]. CA je dynamický systém, v ktorom je čas aj priestor diskrétny. Skladá sa z mriežky jednoduchých funkčných jednotiek – buniek, ktoré môžu nadobúdať jeden z viacerých, vopred definovaných stavov. Stavy buniek sú synchrónne aktualizované v každom kroku výpočtu CA na základe prechodovej funkcie. Formálne [27]:

$$s^{(t+1)} = f(s^{(t)}, s_N^{(t)}), \forall i, j, \tag{3.1}$$

kde $s^{(t+1)}$ reprezentuje stav bunky danej pozíciami i a j v čase $t+1$, $s^{(t)}$ vyjadruje stav bunky v čase t, f je prechodová funkcia CA a $s_N^{(t)}$ značí stavy okolitých buniek.

Okolie buniek CA môže byť špecifikované rôzne, obrázok 3.2 zobrazuje najčastejšie používané. Všetky vizualizované okolia sú intuitívne jasné, až na Margolusove okolie (3.2e). Tento typ okolia je špecifický tým, že plochu s bunkami je nutné rozdeliť na štvorce o veľkosti 2×2 a stav všetkých buniek v štvorci závisí len na 4 bunkách ohraničených daným štvorcom. Navyše, všetky bunky v jednotlivých štvorcových plochách majú rovnaký stav. Aby sa však nebránilo propagácii stavov buniek, celá sieť 2×2 plôch sa posunie v každom párnom

kroku evolúcie automatu o jednu bunku v ose x aj y a v každom nepárnom kroku zase späť. Popísaný, pomerne zvláštny typ okolia sa úspešne využíva napríklad pri približnom riešení, už skôr spomínaných, Navier–Stokesových rovníc [73].



(a) Von Neumannovo okolie

(b) Rozšírené Von Neumannovo okolie

(c) Moorovo okolie

(d) Rozšírené Moorovo okolie

(e) Margolusovo okolie

Obrázok 3.2: Rôzne typy okolia CA.

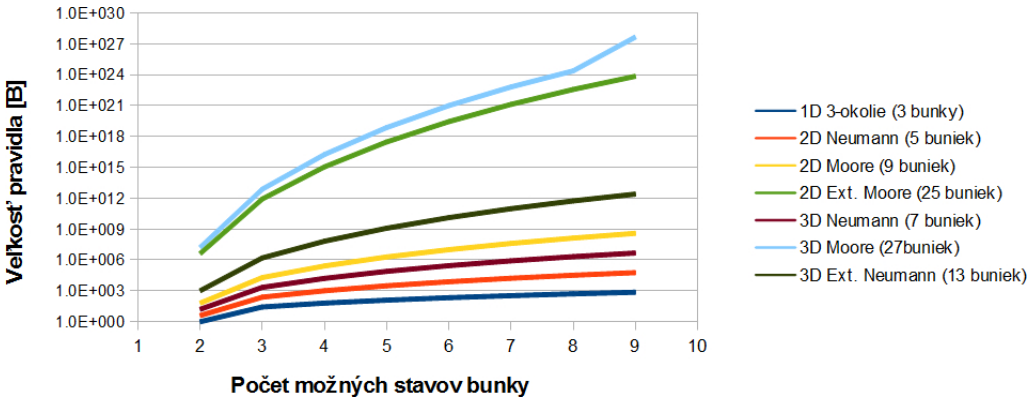
Hlavné rysy CA sú:

- **paralelizácia** – jednotlivé stavy buniek je možné počítať paralelne,
- **lokalita** – nový stav bunky závisí na stave aktuálnej a okolitých buniek,
- **homogenita** – všetky bunky používajú rovnakú prechodovú funkciu, to však platí len pre klasické celulárne automaty (uniformné),
- **diskrétnosť** – časová aj priestorová.

Každá navrhnutá abstrakcia reality – model, má na základe svojej špecifikácie výhody a nevýhody. Medzi svetlé stránky modelu CA patrí najmä:

- **jednoduchosť** – či už ide o jednoduchosť principiálnu alebo implementačnú,
- **paralelizmus** – stavy buniek v nasledujúcom časovom úseku je možné počítať paralelne, čo predstavuje obrovskú výhodu a rýchlostný potenciál evolúcie CA,
- **vizuálnosť** – výstupy sú väčšinou vizualizovatelné a jednoducho interpretovateľné.

Pamäťová náročnosť modelu však stúpa pri nepatrnom zväčšení okolia či zvýšení počtu stavov, ktorých môžu jednotlivé bunky nadobúdať. Porovnanie náročnosti modelu v závislosti na počte stavov a type resp. veľkosti okolia je znázornené na obrázku 3.3. Priestorová a časová diskrétnosť sa v niektorých prípadoch nehodí a môže byť kontraproduktívna.



Obrázok 3.3: Pamäťová náročnosť modelu celulárneho automatu v závislosti na počte stavov a type okolia.

Sňáď najpopulárnejším celulárnym automatom je už spomínaný celulárny automat s názvom „Game of Life“. Je uvedený takmer v každej spisbe týkajúcej sa CA, pripomína vývoj spoločenstva živých organizmov. Existuje veľké množstvo implementácii[2], kde každá bunka sa nachádza v 1 z 2 stavov – mrtvá alebo živá. Okolie je 5-bunkové, von Neumannove. Pravidlá hry resp. prechodová funkcia je veľmi jednoduchá:

- bunky s menej než 2 živými susednými bunkami zomierajú
- živé bunky s 2 alebo 3 živými susedými bunkami prežívajú
- živé bunky s viac než 3 živými susednými bunkami zomierajú
- mrtvé bunky s práve 3 živými susednými bunkami ožívajú

Paralelný výpočet buniek CA dal vznik mnohým hardwarovým riešeniam šitým na mieru špecifickým problémom, príkladom je CAM (Cellular Automata Machines) vyvinutý na MIT (Massachusetts Institute of Technology), za ktorej vývojom stoja Norman H. Margolus a Tomasso Toffoli. V dobe písania tejto práce bola aktuálna verzia CAM-8[3].

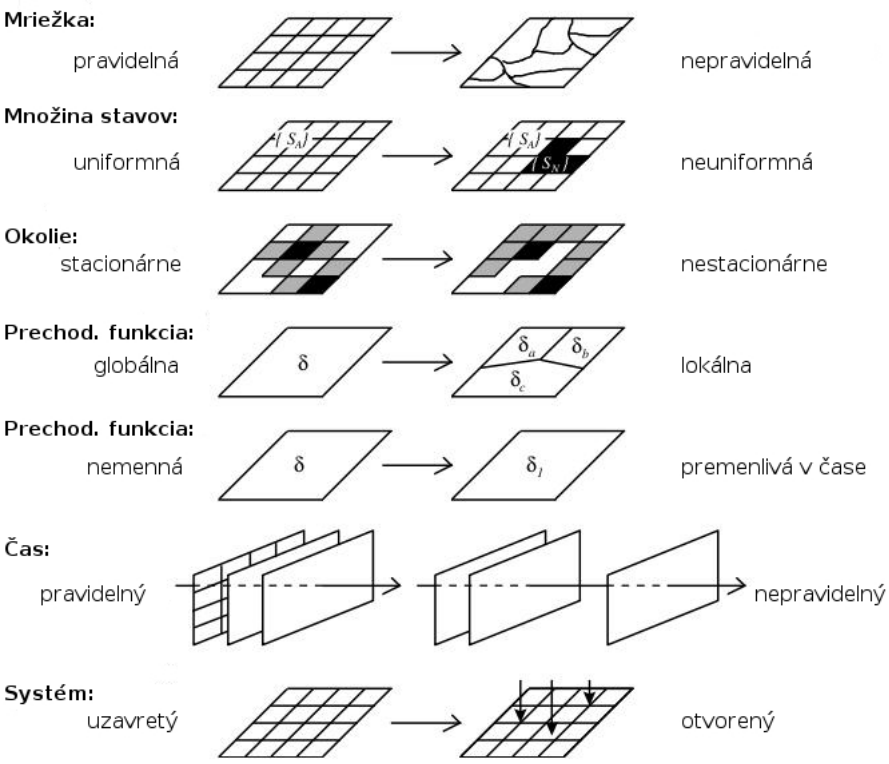
3.3 Hlavné aplikačné domény

Pomerne úzka aplikačná doména celulárnych automatov sa postupom času rozrástla až do takej miery, že sa CA začali používať v každej oblasti, kde je potrebné simulovať nejaké dynamické priestorové deje, ktoré sú charakteristické svojou komplexnosťou. Začal sa používať v rôznych oblastiach života – výpočetné úlohy, fyzikálne, chemické, biologické procesy, sociologické procesy (segregácia) atď. V praktických experimentoch sa CA využívajú v rôznych modifikáciach, málokedy sa použije model klasického celulárneho automatu. Príklady modifikácii CA sú znázornené na obrázku 3.4.

Významné postavenie má CA v modelovaní biochemických javov, príkladom je predikcia miesta pôsobenia proteínov v bunke. Cieľom je vytvoriť automatizovanú metódu spoľahlivého určovania polohy skúmaných proteínov. Informácia o polohe proteínu dokáže výrazne

[2] Interaktívnu implementáciu v podobe appletu možno nájsť na adrese: http://www.bitstorm.org/gameoflife/.

[3] Pre bližšie informácie o CAM-8 navštívte http://www.ai.mit.edu/projects/im/cam8.



Obrázok 3.4: Klasický vs. modifikovaný CA, prevzaté z [12].

urýchliť proces určovania jeho biologickej funkcie. CA sa v tomto prípade používa na tvorbu „obrázkov“, na ktoré sa aplikujú metódy rozpoznávania vzorov v obraze [77].

Model „bunkového“ automatu možno použiť aj na modelovanie biologických dráh, konkrétne na modelovanie signálnych dráh mitogénom aktivovaných proteínkynáz [44]. Ide o signálnu dráhu, po ktorej sú signály vysielané z cytoplazmatickej membrány do cytoplazmy a jadra. Použitý CA modeluje 3 rôzne substráty a 4 enzýmy. Počiatočné koncentrácie enzýmov sú popísané parametrami buniek celulárneho automatu. Každej bunke je priradený stav, ktorý hovorí, či je bunka prázdna, či je tvorená substrátom, enzýmom alebo ich produktom. Obsah bunky môže uniknúť z okupujúcej bunky alebo prejsť do susednej okupovanej bunky. Tieto trajektórie sú popísané pomocou pravdepodobnostných pravidiel na začiatku behu celulárneho automatu, aby reflektovali predpokladaný vzťah medzi prvkami systému. Pravidlá sú nasledovne aplikované náhodne na každú bunku, až kým všetky bunky nemajú korektne prepočítané svoje stavy a trajektórie.

Kapitola 4

Evolučné algoritmy

Charles Darwin bol Angličan, syn významného lekára. Vyštudoval teológiu, po štúdiu sa zaoberal geologickými formáciami v horách Walesu. Koncom roka 1831 však odišiel na 5-ročnú výskumnú cestu okolo sveta. Loď HMS Beagle ho zaviedla aj na Galapágy, ekvádorské súostrovie 19 sopečných ostrovov vo východnej časti Tichého oceánu, kde zhromaždil podľa jeho slov najcennejšiu časť prírodovedeckého materiálu, ktorý použil vo svojom najväčšom diele publikovanom v roku 1859 – *O vzniku druhov prírodným výberom alebo uchovávanie prospešných plemien v boji o život* [20]. Hlavou plnou prírodovedných informácií, získaných z okružnej plavby okolo sveta, v ňom vrhá ucelený pohľad na vývoj druhov oslobodený od spirituality a náboženských predstáv svojej a predchádzajúcich dôb. Darwin vysvetľuje vznik rôznych druhov organizmov na základe prirodzeného výberu, teda schopnosti prežiť len tých najschopnejších. Významným argumentom pre jeho teóriu boli aj stratigrafické[1] výsledky geológa Charlesa Lyella, ktoré podporovali rodiacu sa evolučnú teóriu v oblasti „časovej zložitosti“. Aplikované princípy klasickej evolučnej teórie s mnohými „vylepšeniami“ hrajú významnú úlohu vo fonde vedomostí ľudstva.

Evolučné algoritmy (EA), ktoré sú postavené na myšlienkach evolučnej teórie, začali vznikať už v 50-tych rokoch 20. storočia. Výraznejší záujem však nastal až približne o 30 rokov neskôr, kedy David Goldberg významne rozšíril prácu Johna Hollanda o genetických algoritmoch (z roku 1975 [37]) v práci publikovanej v roku 1989 [32]. Značným impulzom pre popularizáciu EA bola prvá väčšia práca o genetickom programovaní, ktorej autorom je John Koza [46].

4.1 Biologické pojmy v kontexte evolučných algoritmov

Vo zvyšku tejto práce sa budú vyskytovať pojmy pochádzajúce z biológie, no sémantika nie všetkých je zhodná so sémantikou v kontexte EA. Pre ujasnenie pojmov je uvedený ich krátky prehľad.

Medzi základné pojmy Darwinovej evolučnej teórie patrí populácia. Populácia je množina jedincov, ktorí sú reprezentovaní svojím genetickým materiálom. V tejto práci budú voľne zamieňané pojmy genetický materiál, genóm a chromozóm, aj keď z pohľadu biológie tieto pojmy nie sú rovnocenné. Genotyp je vlastné zakódovanie genetickej informácie do určitej štruktúry. Spôsob, akým sa genotyp v danom prostredí interpretuje, ako dobre rieši nastolený problém, sa nazýva fenotyp. Jedinec s rovnakým genotypom môže mať v inom prostredí odlišnú schopnosť prežitia, inými slovami, odlišné prostredie spôsobí odlišnú inter-

[1] Stratigrafia je geologický vedný obor, který študuje vek sedimentárnych vrstiev hornín.

pretáciu genotypu na fenotyp. Genetický materiál sa skladá z lineárne usporiadaných génov, v kontexte EA jeden gén kóduje jednu vlastnosť. Konkrétna vlastosť, hodnota génu, sa nazýva alela. V rámci počítačovej terminológie môžeme povedať, že každý gén reprezentuje určitý dátový typ a alely sú hodnotami daného dátového typu, génu.

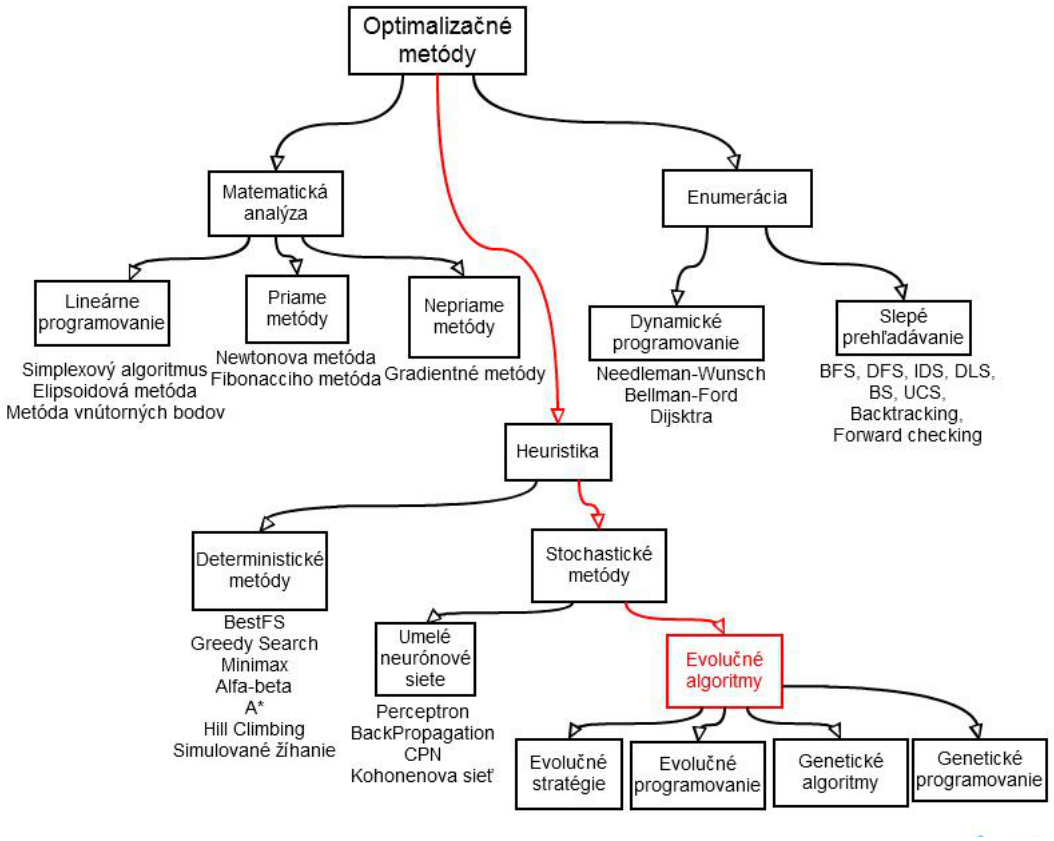
4.2 Klasifikácia evolučných algoritmov

Existujú problémy, ktorých exaktný model sa nedá, nevieme alebo je veľmi náročné zostaviť, a navyše počet riešení daného problému, resp. stavový priestor úlohy je obrovský. Kvôli takýmto problémom vznikla nová vetva prístupov k riešeniu úloh označovaná pojmom *softcomputing*. Ide o spôsob riešenia problémov „s rozumom“, teda nie hrubou (hard) silou ako je to prípade klasických algoritmov na prehľadávanie stavového priestoru (napr. BFS, DFS). Softcomputing rieši problémy pomocou určitej heuristiky resp. heuristickej funkcie, ktorou algoritmus „myslí“. Potrebné je dodať, že metódy softcomputingu väčšinou nenájdu najlepšie riešenie, ale len suboptimálne, ktoré však takmer vždy postačuje. Obrázok 4.1 zobrazuje zasadenie evolučných algoritmov do kontextu všetkých významných optimalizačných techník.

Evolučné algoritmy sú spoločným vyjadrením pre množinu moderných matematických postupov, ktoré využívajú modely evolučných procesov v prírode založených na Darwinovej evolučnej teórii popísanej na začiatku tejto kapitoly. Jednotlivé riešenia, ktoré tvoria populáciu, sa vyvíjajú na základe klasických evolučných a genetických operátorov ako je selekcia, kríženie či mutácia. EA stavajú a súčasne aj padajú na fitness funkcii, ktorá definuje schopnosť jedinca prežiť v danom prostredí resp. schopnosť riešenia riešiť daný problém. Evolučné algoritmy sú v poslednej dobe veľmi rozšírené. Prílišná popularizácia so sebou však prináša aj nerealistické očakávania. Podobne ako evolučné algoritmy, tak všeobecne aj všetky optimalizačné techniky nefungujú ako univerzálne metódy, ale každá z nich sa hodí na inú oblasť problémov. Zistiť, ktorá technika je najlepšia, prípadne s akými parametrami, je už úlohou inžinirskeho návrhu.

Podstatným rozdielom oproti klasickým optimalizačným metódam je práca nie s jedným, ale s množinou riešení, na ktorú je možno aplikovať genetické a iné operátory. Princíp evolúcie jednotlivých riešení v rámci populácie popisuje nasledovná všeobecná schéma evolučného algoritmu [40]:

1. Vynuluj hodnotu počítadla generácii $t = 0$.
2. Náhodne vygeneruj počiatočnú populáciu $P(0)$.
3. Vypočítaj ohodnotenie ($fitness$) každého jedinca v počiatočnej populácii $P(0)$.
4. Vyber dvojice jedincov z populácie $P(t)$ a vytvor ich potomkov $P'(t)$.
5. Vytvor novú populáciu $P(t+1)$ z pôvodnej populácie $P(t)$ a množiny potomkov $P'(t)$.
6. Zväčši hodnotu počítadla generácii o jedna ($t := t + 1$).
7. Vypočítaj ohodnotenie ($fitness$) každého jedinca v populácii $P(t)$.
8. Ak je t rovné maximálnemu počtu generácii alebo je splnené iné ukončovacie kritérium, vráť ako výsledok populáciu $P(t)$; inak pokračuj krokom číslo 4.



Obrázok 4.1: Klasifikácia evolučných algoritmov v kontexte optimalizačných techník. Soft-computing spadá medzi stochastické heuristické metódy.

Keď v roku 1963 začali Hans-Paul Schwefel a Ingo Rechenberg na Technickej univerzite v Berlíne s napodobňovaním vývoja v prírode, boli presvedčení, že ich metóda najlepšie aproximuje evolúciu v živej prírode. Preto svoju metódu nazvali, celkom všeobecne, *evolučné stratégie* (ES). Postupom času sa však ukázalo, že tento spôsob rieši len určitý typ úloh, hlavne v stavebnom a strojnom inžinierstve. Genetické algoritmy nie sú teda podradené evolučným stratégiam, ako sa domnievali, ale naopak, svojou popularitou ich zatieňujú [49]. Genóm v rámci ES je zložený z génov, ktoré sú reprezentované reálnymi číslami, z čoho vyplýva implementačná diferenciácia genetických operátorov. Mutačný operátor je väčšinou implementovaný pomocou pripočítania hodnoty podľa Gaussovej funkcie. Takýto spôsob mutácie rieši jeden problém genetických algoritmov, ktorý znie: malé zmeny genotypu nemusia viesť k malým zmenám fenotypu[2].

4.3 Evolučné operátory

Evolučné procesy z biológie boli aplikované a svojím spôsobom interpretované v teórii evolučných algoritmov. Ide o pekný príklad medzioborového transféru informácií. V evolučnom procese je nutná značná variácia genómu, ktorá je zabezpečovaná rekombinačnými (gene-

[2] Tento problém sa dá obísť napríklad pomocou Grayovho kódovania, v ktorom sa každé dve po sebe idúce hodnoty líšia v bitovom vyjadrení len na jednej pozícii.

tickými) operátormi. Následný výber najlepších jedincov reprezentuje operátor selekcie.

Selekcia je evolučný operátor, ktorý určuje, ktoré riešenie v populácii riešení prežije, a ktoré nie. Reprezentuje prirodzený výber popísaný Darwinom. Rozoznávame 3 najpoužívanejšie typy selekcie a ich varianty [40]:

- **Koleso šťastia** – jednotlivým jedincom sa priradí pravdepodobnosť výberu do ďalšej generácie na základe hodnoty fitness funkcie, „lepší" jedinci budú vyberaný s vyššou pravdepodobnosťou.
- **Turnaj** – je založený na náhodnom výbere n-tíc jedincov a ich súboji, v ktorom sú zbraňami hodnoty fitness funkcie, víťaz je vo väčšine variant tejto selekcie vybraný do ďalšej generácie vždy, no môže byť vyberaný s pravdepodobnosťou menšou než 1.
- **„Najlepší vyhráva"** – je najjednoduchším typom selekcie, v každej generácii preferuje len tých najstatnejších jedincov, jedincov umiestnených na čelných priečkach rebríčka zostavovaného komisiou, ktorá hodnotí statnosť jedinca na základe hodnoty fitness funkcie. Popisovaný spôsob výberu je vhodný v prípade, keď fitness funkcia nemá veľa extrémov, pretože evolučné algoritmy založené na tomto type selekcie nevedia riešiť tzv. klamné problémy a multimodálne funkcie, pretože v populácii sa nezachováva diverzita jedincov, hodnôt fitnes funkcie. Evolučné algoritmy založené na tomto type selekcie častkrát predbežne konvergujú, uviaznu v lokálnom extréme fitness funkcie.

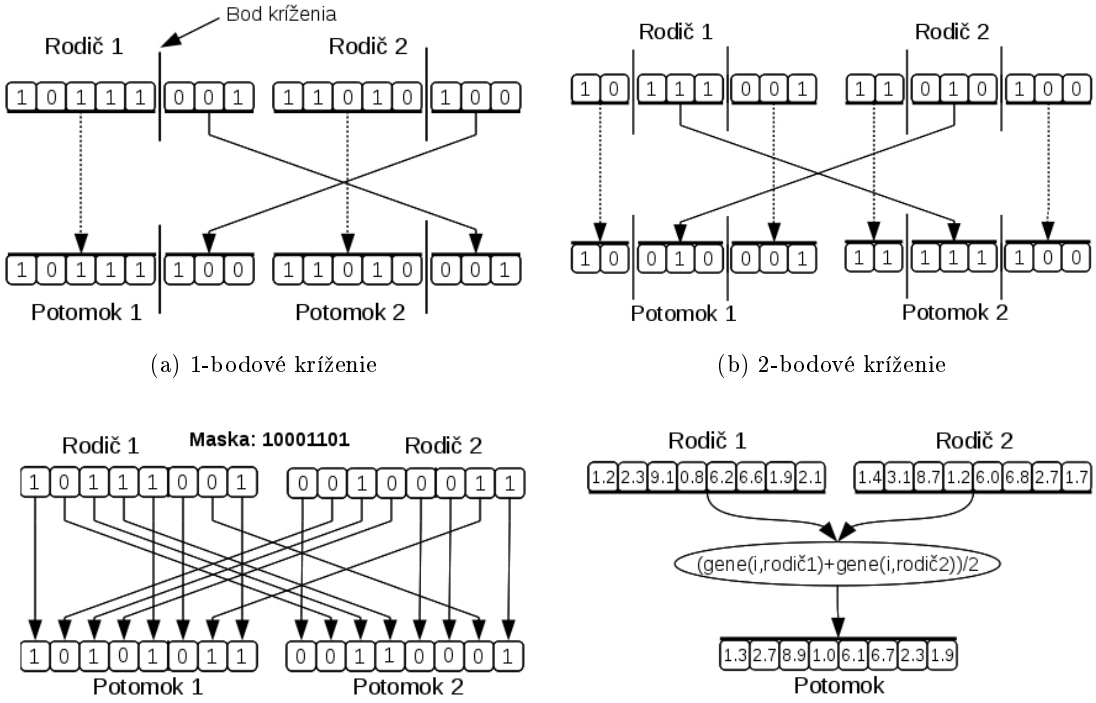
Pri praktických aplikáciach je len málokedy možné sa stretnúť s určitým typom selekcie v základnej podobe, takmer vždy sa používajú ich možné variácie a kombinácie. Pre priblíženie, existujú aj prístupy, ktoré pracujú s dvoma populáciami kvôli zachovaniu rôznorodosti populácie a dochádza k migrácii medzi populáciami. Každá populácia je však založená na inej fitness funkcii [58]. So selekciou úzko súvisí obnova populácie. Po vyhodnotení hodnôt fitness funkcie jedincov populácie a selekcii jedincov, ktorí „prežijú", máme viac možností ako nahradiť aktuálnu populáciu. Rozlišujeme 2 základné prístupy:

- **Úplná obnova populácie** – dochádza k vymieraniu rodičov, teda celá generácia je nahradená novou.
- **Čiastočná obnova populácie** (steady state) – potomkami sa nahradi len určitá časť jedincov.

Kríženie je základným rekombinačným operátorom, pomocou ktorého sa mieša genetická informácia 2 jedincov. Rôzne techniky kríženia majú spoločnú tú vlastnosť, že ide vždy o vzájomnú výmenu častí chromozómov. V niektorých prípadoch môže byť však potrebné a užitočné dať jedincom možnosť prežiť bezo zmeny a uchovať pre budúcu populáciu kópie rodičovských jedincov. Preto sa operátor kríženia aplikuje s istou pravdepodobnosťou a v ostatných prípadoch sú za potomkov rodičovského páru prehlásené ich priame kópie. Pravdepodobnosť použitia operátoru kríženia je obvykle relatívne vysoká (0,75–0,95 [70]). Kríženie umožňuje rýchlu výmenu relatívne veľkého množstva genetickej informácie a do značnej miery ovplyvňuje efektívnosť evolučného algoritmu [40]. K tomuto operátoru možno pristupovať viacerými spôsobmi (vizuálna podoba je zobrazená na obrázku 4.2):

- **N-bodové kríženie** – najpoužívanejší typ kríženia, väčšinou sa používa kríženie jednobodové alebo dvojbodové, závisí samozrejme na danom probléme a veľkosti chromozómu.

- **Kríženie maskou** – náhodne sa vygeneruje bitová maska, ktorej dĺžka je zhodná s dĺžkou chromozómu a noví jedinci sa tvoria tak, že gény na pozíciach obsahujúcich „0“ zdedí prvý potomok a gény na pozíciach obsahujúcich „1“ zdedí potomok druhý.
- **Aritmetické kríženie** – využíva sa najmä pri evolučných stratégiach, kde sú gény reprezentované reálnymi číslami, noví jedinci sa tvoria na základe aplikácie nejakého aritmetického operátoru (väčšinou priemer) na gény rodičov.



(a) 1-bodové kríženie

(b) 2-bodové kríženie

(c) Kríženie maskou

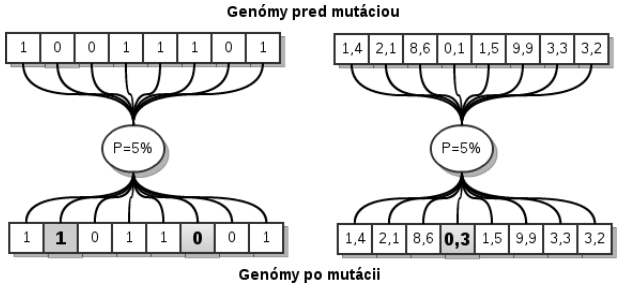
(d) Aritmetické kríženie (pomocou priemeru)

Obrázok 4.2: Rôzne typy evolučného operátoru kríženia, prevzaté z [11].

Operátor *mutácie* väčšinou veľmi jednoduchým spôsobom a s relatívne malou pravdepodobnosťou (0,001–0,05 [70]) náhodne mení hodnotu jednotlivých génov. V prípade binárneho kódovania to konkrétne znamená, že vybraný gén v danom chromozóme zmení svoju hodnotu z nuly na jednotku a naopak. Slúži ako nástroj, ktorý bráni príliš rýchlemu „zjednotvárneniu“ vlastností v rámci populácie, strate užitočného genetického materiálu a predčasnej konvergencie populácie [40]. Rozlišujeme v zásade 2 typy mutácie (ilustruje obrázok 4.3):

- **Inverzia génu** – vhodné a použitelné len pri binárnej reprezentácii génov, teda génov, ktoré môžu existovať len v 2 navzájom rôznych alelách.
- **Pripočítanie hodnoty rozloženia pravdepodobnosti** – využíva sa pri reprezentácii génov reálnymi číslami, k hodnote génu sa pripočíta hodnota daná určitým rozdelením pravdepodobnosti.

Špecificky definované chromozómy jedincov si vyžadujú špecifické operátory kríženia a mutácie. Rozšírené je tzv. *permutačné kódovanie* kandidátneho riešenia. Využíva sa väčšinou pri problémoch, kedy je riešením permutácia určitých hodnôt. Podrobne študovaným



Obrázok 4.3: Mutácia genómu, ľavá časť obrázku reprezentuje aplikáciu operátoru mutácie pomocou inverzie génu, pravá časť pomocou pripočítania hodnoty rozloženia pravdepodobnosti.

je napríklad problém obchodného cestujúceho[3]. Mutácia takto zakódovaných jedincov sa najčastejšie rieši prehodením hodnôt dvoch náhodne vybraných génov mutovaného chromozómu. Alternatívnym prístupom je obrátenie podsekvencie chromozómu (angl. Reverse Sequence Mutation), kedy sa vezme dvomi náhodne vygenerovanými pozíciami obmedzená podsekvencia chromozómu a poradie génov v sekvencii sa otočení [2]. Kríženie je zložitejšie, existujú 3 najpoužívanejšie prístupy [2] (pre zamedzenie konfúzii sú ponechané ich originálne anglické názvy):

- **Order 1 crossover (OX)** – náhodne sa vygenerujú body kríženia, gény medzi bodmi kríženia 1. rodiča sa skopírujú do potomka, následne sa postupne prechádzajú jednotlivé gény 2. rodiča (od 2. bodu kríženia) a do potomka sa kopírujú len tie gény, ktoré sa nenachádzajú v časti medzi bodmi kríženia 1. rodiča.
- **Partially-mapped crossover (PMX)** – náhodne sa vygenerujú body kríženia, gény medzi bodmi kríženia 1. rodiča sú skopírované do potomka. Prechádzamjú sa gény medzi bodmi kríženia 1. rodiča zľava doprava, aktuálny gén sa „spojí“ s protiľahlým génom a s génom s rovnakou hodnotou v 2. rodičovi. Ak protiľahlý gén nemá rovnakú hodnotu ako aktuálny, skopírujú sa tieto 2 gény a prehodia ich pozície, následné sa postupným posunom doprava tento algoritmus opakuje. Následne sa do potomka skopírujú len tie gény z 2. rodiča, ktoré sú mimo body kríženia, pričom sa zachováva pozícia génov. V poslednom kroku sa prechádzajú jednotlivé gény 2. rodiča, začína sa génom za 2. bodom kríženia a do potomka sa kopírujú tie gény, ktorých hodnota sa nenachádza v časti medzi bodmi kríženia 1. rodiča alebo v potomkovi.
- **Cycle crossover (CX)** – kríženie sa začína s génom na 1. pozícii 1. rodiča (alebo na inej štartovacej pozícii) a skopíruje sa na 1. pozíciu potomka. Potomok teda nemôže dediť 1. gén z 2. rodiča, takže tento gén musí byť vyhľadaný v 1. rodičovi a skopírovaný do potomka. Nech tento gén má pozíciu x v 1. rodičovi. Potom je zdedený potomkom na pozícii x a nemôže byť teda zdedený od 2. rodiča. Tento proces sa opakuje, kým sa nevytvorí cyklus, teda kým sa nedosiahne gén, ktorý už potomok zdedil. Potom sa vyberie gén z 2. rodiča, a vytvorí sa analogicky cyklus, tentokrát však potomok dedí gény 2. rodiča. Nevýhodou tohto prístupu môže byť to, že podsekvencie dedených génov nemusia byť súvislé.

[3] Problém obchodného cestujúceho (angl. Travelling Salesman Problem) je optimalizačný problém nájdenia najkratšej možnej cesty prechádzajúcej všetkými zadanými bodmi na mape. Matematicky ide o nájdenie Hemiltonovskej cesty v grafe G s najnižšou cenou C.

Kapitola 5

Návrh predikčného systému

Navrhnutý systém z veľkej časti vychádza z dvoch prác venujúcich sa SSP s využitím CA [13] [74]. Modifikované sú tie časti systému, ktoré majú potenciál zlepšiť úspešnosť predikcie SSP. Ide hlavne o spôsob klasifikácie jednotlivých reziduí do jednej z troch motívov sekundárnej štruktúry a parametrizácie modelu CA.

Ako bolo spomenuté v sekcii 2.3, výsledná štruktúra proteínov závisí na veľkom množstve známych či neznámych parametrov. Niektoré parametre je možné získať iba experimentálne. No môžeme hodnoty týchto parametrov nahradiť hodnotami najpodobnejšej sekvencie získanej zarovnaním (napríklad pomocou algoritmu BLAST [4]), pretože konzervácia častí sekvencie spôsobuje, že proteíny v rámci jednej rodiny majú podobné vlastnosti. Aminokyselinová sekvencia však musí byť porovnávaná s celou databázou sekvencií, navyše, používané algoritmy zarovnávania majú linárnu časovú zložitosť v závislosti na dĺžke sekvencie. Jednoduchšie je využiť informácie v dátach, tzn. protriedkami štatistiky dáta nejakým spôsobom popísať, ideálne z pohľadu nastoleného problému. Takéto získanie informácii je samozrejme obmedzené pokrytím dát. Takýmto spôsobom však možno získať aj také informácie, ktoré sú neznáme, skryté, no nejakým spôsobom agregované do korelácii atp. Jedným z cieľov návrhu je, aby bola predikcia systému rýchla a aby bol systém robustný, tzn. mal by byť schopný predikovať sekundárnu štruktúru ľubovolnej aminokyselinovej sekvencie, teda mal by fungovať bez nutnosti získavania ďalších potrebných informácii o predikovanej sekvencií. Na to tu sú vhodnejšie, sofistikované prediktory využívajúce evolučné informácie získané na základe zarovnania sekvencie v rámci určitej proteínovej rodiny, alebo chemické posuny, kde je taktiež potrebné zarovnanie a hľadanie najpodobnejšej aminokyselinovej sekvencie. Uvažovaním týchto parametrov do navrhovaného systému by model CA strácal zmysel a vhodnejšie by sa javili iné predikčné modely. Samozrejme, dosahovaná úspešnosť nebude závratná, no adekvátna požadovaným vlastnostiam modelu.

Predikčným modelom je spomínaný CA, ktorého prechodová funkcia je suboptimálne parametrizovaná pomocou evolučného algoritmu, konkrétne pomocou evolučnej stratégie. Boli navrhnuté 2 prechodové funkcie. Prvá je zhodná s prechodovou funkciu vytvorenou Choprom a Benderom v [13], bola implementovaná najmä kvôli porovnávaniu s druhou, rozšírenou verziou prechodovej funkcie, ktorá využíva okrem klasickým Chou-Fasmanových koeficientov aj tzv. konformačné koeficienty (viď sekcia 5.1), ktoré štatisticky popisujú pravdepodobnosť výskytu určitej aminokyseliny v určitom konformačnom stave resp. motíve sekundárnej štruktúry.

5.1 Štatistický popis reziduí

V návrhu systému sú využité 2 štatistické vlastnosti aminokyselín, *Chou-Fasmanove koeficienty*, ktoré aminokyseliny charakterizujú z pohľadu miery výskytu v určitom motíve sekundárnej štruktúry, a *konformačné preferencie*, ktoré popisujú jednotlivé aminokyseliny na základe predispozície nachádzať sa na začiatku, resp. na konci určitého motívu sekundárnej štruktúry.

Jednou z metód predikcie SSP prvej generácie je metóda Chou-Fasman (viď 2.3 pre prehľad predikčným metód), v ich práci z roku 1974 [15] je definovaný tzv. parameter konformačnej predispozície aminokyseliny j ku konformačnému stavu i, Chou-Fasmanov koeficient P_j^i:

$$P_j^i = \frac{f_j^i}{\langle f_j^i \rangle}, \tag{5.1}$$

kde f_j^i je relatívna frekvencia aminokyseliny j v konformačnom stave i daná vzťahom 5.2 a $\langle f_j^i \rangle$ priemerná relatívna frekvencia konformačného stavu i v rámci všetkých aminokyselín vyjadrená vzťahom 5.3. Kvôli konzistencii s odkazovanými prácami sa v systéme s Chou-Fasmanovými koeficientami P_j^i pracuje v percentuálnej podobe, tzn $P_j^i \cdot 100$.

$$f_j^i = \frac{n_j^i}{n^i} \tag{5.2}$$

kde n_j^i je počet reziduí j v konformačnom stave i a n^i je celkový počet reziduí v konformačnom stave i.

$$\langle f_j^i \rangle = \frac{\sum\limits_{\forall k \, \epsilon \, AK} f_k^i}{n_j} \tag{5.3}$$

Na základe práce Guang-Zheng Zhanga a spol. [79] definujeme *konformačnú triedu* pre všetky aminokyseliny a všetky konformačné stavy (H, E, C). Nech $P = p_1, p_2, \ldots, p_n$ je primárna štruktúra proteínu (sekvencia aminokyselín) a $S = s_1, s_2, \ldots, s_n$ odpovedajúca sekundárna štruktúra proteínu dĺžky n. Ak s_i a s_{i+1} sú rôzne konformačné stavy, napríklad $s_i = H$ a $s_{i+1} = E$, hovoríme o tzv. štruktúrnom prechode (ST[1]), v tomto prípade ST_{HE}. Týmto spôsobom môžeme definovať ostatných 5 štruktúrnych prechodov: ST_{HC}, ST_{EH}, ST_{EC}, ST_{CH} a ST_{CE}.

Na základe uvedených štruktúrnych prechodov definujeme konformačnú preferenciu aminokyselín nachádzať sa na začiatku resp. konci určitého motívu sekundárnej štruktúry. Štruktúrny prechod ST_{HE} môžeme chápať ako ukončenie H a súčasne ako začiatok B. V kontexte všetkých 6 ST, počet všetkých ukončení a začiatkov H určíme nasledovne:

$$N_{\alpha\text{-ukončenie}} = N_{ST_{HB}} + N_{ST_{HC}} \tag{5.4}$$

$$N_{\alpha\text{-začiatok}} = N_{ST_{BH}} + N_{ST_{CH}} \tag{5.5}$$

[1] Z angl. Structure Transition.

kde $N(\cdot)$ reprezentuje počet rôznych štruktúrnych prechodov. Počet ukončení a začiatkov B a C sa určí analogicky. Definujeme konformačnú preferenciu CP ukončenia resp. začiatku určitého konformačného stavu aminokyseliny i, konkrétne $CP_{j,\alpha\text{-ukončenie}}$, $CP_{j,\alpha\text{-začiatok}}$, $CP_{j,\beta\text{-ukončenie}}$, $CP_{j,\beta\text{-začiatok}}$, $CP_{j,\text{Coil}-\text{ukončenie}}$ a $CP_{j,\text{Coil-začiatok}}$. Výpočet $CP_{j,\alpha\text{-ukončenie}}$ (ostatné konformačné preferencie sa získajú analogicky):

$$CP_{j,\alpha\text{-ukončenie}} = \frac{P_{j,\alpha\text{-ukončenie}}}{P_j} \tag{5.6}$$

kde $P_{j,\alpha\text{-ukončenie}}$ a P_j sa získa nasledovne:

$$P_{j,\alpha\text{-ukončenie}} = \frac{N_{j,\alpha\text{-ukončenie}}}{\sum_{i=1}^{20} N_{i,\alpha\text{-ukončenie}}} \tag{5.7}$$

kde $N_{j,\alpha\text{-ukončenie}}$ vyjadruje počet reziduí ukončujúcich H.

$$P_j = \frac{N_j}{N} \tag{5.8}$$

kde N_j vyjadruje celkový počet reziduí aminokyseliny j a N celkový počet reziduí. Uvažujúc reziduum j a jeho motív sekundárnej štruktúry, α-helix (H), definujeme konformačnú triedu (CC) konformačného stavu rezidua j (obdobne možno vyjadriť konformačné triedy pre B a C):

$$CC_{j,\alpha} = \begin{cases} b & \text{ak } CP_{j,\alpha\text{-ukončenie}} \geq 1 \ \wedge \ CP_{j,\alpha\text{-začiatok}} < 1 \\ f & \text{ak } CP_{j,\alpha\text{-začiatok}} \geq 1 \ \wedge \ CP_{j,\alpha\text{-ukončenie}} < 1 \\ n & \text{inak} \end{cases} \tag{5.9}$$

Použité znaky b, f, n značia v tomto poradí triedy *Breaker*, *Former* a *Neutral*. Trieda *Breaker* reprezentuje reziduá, ktoré väčšinou ukončujú určitý motív sekundárnej štruktúry, *Former* reprezentuje reziduá, ktoré ho začínajú, a do triedy *Neutral* spadajú reziduá väčšinou nachádzajúce sa mimo jeho okrajov. Konformačná klasifikácia rezidua j je vyjadrená 3-znakovým kódom – $CC_{j,\alpha}CC_{j,\beta}CC_{j,Coil}$. Pre potreby modelu nie je použitá vlastná konformačná klasifikácia $CC_{j,\alpha}$, ale konformačné preferencie, na základe ktorých sa klasifikuje, tzn. $CP_{j,\alpha\text{-ukončenie}}$ resp. $CP_{j,\alpha\text{-začiatok}}$.

5.2 1 D celulárny automat ako model aminokyselinovej sekvencie

Sekvenciu aminokyselín proteínov reprezentuje 1 D CA, ktorého bunky modelujú jednotlivé reziduá aminokyselín. Bunky môžu nadobúdať jeden z troch stavov (H, E, C). Štatistické vlastnosti aminokyselín sú modelované parametrami buniek CA. Veľkosť okolia nie je optimalizovaná pomocou evolučného algoritmu (ale je parametrizovateľná), najmä kvôli možnej paralelizácii výpočtu a relatívnej zložitosti genetických operátorov navyšujúcej čas evolúcie optimálneho pravidla. V rámci inicializácie sú každej bunke pridelené Chou-Fasmanove koeficienty, ktoré sú pri prechode do ďalšej konfigurácie CA upravované a vyjadrujú predispozície bunky nachádzať sa v určitom stave. Prechodová funkcia CA môže mať podobu základnú alebo rozšírenú. *Základná prechodová funkcia* [13] má nasledovný tvar:

$$S_{t+1,j} = \max R^i_{t+1,j} \quad i \,\epsilon\, \{H, E, C\} \tag{5.10}$$

kde $S_{t+1,j}$ je stav bunky j v čase $t+1$ a parameter $R^i_{t+1,j}$ vyjadruje mieru príslušnosti bunky resp. aminokyseliny j v kroku $t+1$ ku konformačnému stavu i (H, E, C):

$$R^i_{t+1,j} = P^i_{t+1,j} \tag{5.11}$$

$P^i_{t+1,j}$ vyjadruje Chou-Fasman koeficient bunky j v čase $t+1$ pre konformačný stav i, ktorý je váhovaným súčtom jednotlivých Chou-Fasmanových koeficientov P^i_{j-k} v okolí o:

$$P^i_{t+1,j} = \frac{\sum_{k=-o}^{o} w_k P^i_{j-k}}{\sum_{k=-o}^{o} w_k} \tag{5.12}$$

Rozšírená prechodová funkcia sa od základnej líši v definícii predispozícii bunky nachádzať sa v danom stave $R^i_{t+1,j}$:

$$R^i_{t+1,j} = \alpha \cdot R^i_{t,j} + \beta \cdot CP_{j,i\text{-začiatok}} + \gamma \cdot CP_{j,i\text{-ukončenie}} \tag{5.13}$$

kde α, β a γ sú váhy troch definovaných parametrov, ktoré sú optimalizované evolučným algoritmom. Ide o rekurentný zápis nelineárnej funkcie, čo zvyšuje potenciál úspešnejšej klasifikácie, a teda predikcie sekunárnej štruktúry proteínov.

5.3 Okrajové podmienky a inicializácia modelu

Pri modelovaní javov pomocou celulárneho automatu je dôležitá definícia okrajových podmienok popisujúcich situácie, kedy okolie aktuálnej bunky nie je kompletné – týka sa väčšinou buniek na okraji automatu. Podľa Chopra a Bendera, na základe expermentov, ktoré vykonali, je vhodné použiť bunky okolia mimo automatu v štruktúre Coil [13]. Vojtěch Šalanda sa v rámci svojej bakalárskej práce takýmito bunkami zaoberal [74], experimentoval s reálnymi aj fiktívnymi, reálne neexistujúcimi aminokyselinami, a zistil, že ako najvhodnejšia sa javí fiktívna aminokyselina s označením X300, ktorej Chou-Fasmanove parametre sú 0-0-300, tzn. tendencia bunky nachádzať sa v štruktúre Coil je nenulová, má hodnotu 300.

Na inicializácii jedincov v rámci evolučného algoritmu v podstate nezáleží, no pre rýchlosť konvergencie je dôležité sa zaoberať aj touto časťou systému. Je intuitívne jasné, že vplyv reziduí v tesnom okolí predikovanej aminokyseliny bude vyšší než vplyv reziduí vzdialenejších. Platí to najmä pri motívoch α-helixu, no motívy β-sheet často vznikajú globálnou interakciou, ktorú by sa mal, vzhľadom k charaktere modelu, pokúsiť emulovať navrhnutá model celulárneho automatu.

Použitá je inicializácia hodnôt vplyvu jednotlivých okolitých reziduí (váh) na základe normalizovanej Gaussovej funkcie pre strednú hodnotu $\mu = 0$ a smerodatnú odchýlku $\sigma = 0.399$ (hodnota $f(0) \doteq 1$):

$$\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{5.14}$$

5.4 Optimalizácia vektoru parametrov pomocou evolučnej stratégie

Motorom CA je jeho prechodová funkcia, ktorej expertné určenie nie je jednoduché, preto sa na jej určenie využívajú rôzne optimalizačné techniky. Keďže ide o optimalizáciu vektoru celých a reálnych čísel, je použitý algoritmus evolučnej stratégie, ktorý je podmnožinou väčšej triedy optimalizačných techník, evolučných algoritmov. Stavový priestor prechodových funkcií, ktoré sú parametrizované reálnymi číslami je teoreticky nekonečný, čo opodstatňuje použitie optimalizačných techník. Evolvovaný chromozóm základného resp. rozšíreného pravidla, C_z resp. C_r má tvar:

$$\begin{aligned} C_z &= [s, w_{-r}, w_{-r+1}, \ldots, w_{r-1}, w_r] \\ C_r &= [s, \alpha, \beta, \gamma, w_{-r}, w_{-r+1}, \ldots, w_{r-1}, w_r] \end{aligned}$$

kde s vyjadruje počet krokov CA, α vplyv predchádzajúcej predispozície na stav bunky (tým je zaistená nelinearita, viď rovnica 5.13), β vplyv konformačného koeficientu, ktorý vyjadruje schopnosť určitej aminokyseliny začínať určitý motív sekundárnej štruktúry, γ vplyv konformačného koeficientu, ktorý vyjadruje schopnosť určitej aminokyseliny končiť určitý motív sekundárnej štruktúry, a w_i pre $i \epsilon \{-r, \ldots, r\}$ váhy jednotlivých buniek okolia.

Fitness funkciou evolučného algoritmu je jedna z dvoch funkcií porovnávajúcich podobnosť sekvencií (viď sekcia 2.4), Q_3 alebo SOV. Prehľad spôsobu implementácie evolučných operátorov ponúka tabuľka 5.1.

Evolučný operátor	**Spôsob implementácie**
Selekcia	Koleso šťastia
Kríženie	1-bodové
Mutácia	Gaussovo rozdelenie pravdepodobnosti
Náhrada populácie	Čiastočná (steady state)

Tabuľka 5.1: Navrhnuté spôsoby implementácie evolučných operátorov.

Kapitola 6

Implementácia predikčného systému

Implementácia prediktora, nazvaného *CASSP* (Cellular Automaton Secondary Strucure Predictor), predstavuje prepis návrhu systému do podoby núl a jedničiek, do podoby inštrukcií procesora. Tento prepis je vcelku priamočiary proces. Ide najmä o hľadanie čo najvhodnejších implementačných prostriedkov, ktoré budú čo najlepšie reprezentovať navrhnutý model.

Prediktor je implementovaný v jazyku Java JRE (Java Runstime Environment) 1.6, zaistená je bezproblémová funkčnosť aj pre JRE 1.7. Na implementáciu evolučného algoritmu bola použitá voľne dostupná knižnica JGAP [52].

6.1 Konfigurácia a API systému

Výsledný program má dve varianty – konzolovú a webovú. Konzolová aplikácia je konfigurovateľná pomocou konfiguračného súboru a argumentov príkazovej riadky s tým, že konfigurácia parametrov v príkazovej riadke má vyššiu prioritu než konfigurácia v konfiguračnom súbore, čím je zabezpečená určitá flexibilita konfigurácie programu. Veľké množstvo vlastností systému je parametrizovaných, kompletné možnosti konfigurácie možno nájsť v dokumentácii k systému. Systém má definované API, tzn. možno ho tiež použiť ako knižnicu. Príklad použitia API (použitie cross-validácie):

```
SimConfig config = new SimConfig(<conf_file_path>);
config.setDataPath(<data_path>);
config.setCrossProb(0.75);
config.setPop(100);

CASSP predictor = new CASSP(config);
predictor.crossValidate(10); // 10-stupňová cross-validácia

predictor.createEvolutionImage("evolution");
predictor.createAccClassesImage("accuracy");
predictor.createReliabImage("reliability");
```

6.2 Vlastnosti systému

Výhodou systému je možná paralelizácia výpočtu pri trénovaní prechodovej funkcie pomocou cross-validácie, ktorá je implementovaná pomocou vláken. Pre možnú vlastnú definíciu prechodovej funkcie CA je implementovaná abstraktná trieda `CARule`. Prepísaním

```
FVNQHLCGSHLVEALYLVCGERGFFYTPKA
CCCCCCCCHHHHHHHHHHHHHHHCECCCCCC
...
```

(a) Formát dát pre systém CASSP.

```
FVNQHLCGSHLVEALYLVCGERGFFYTPKA
CCCCCCCCHHHHHHHHHHHHHHHCECCCCCC
CCCCCCHHHHHHHHHHHHHHCCCCEEECCCC
95401326789999999708622662589
...
```

(b) Formát dát pre systém CASSP využívajúci nástroj PSIPRED.

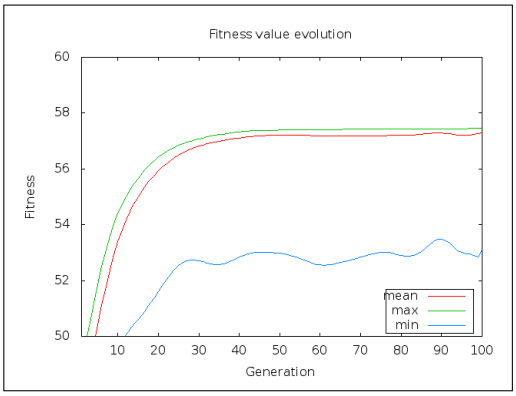
Obrázok 6.1: Formát dátových súborov pre samostatný nástroj CASSP (a) – aminokyselinová sekvencia, referenčná sekvencia sekundárnej štruktúry, a pre systém spolupracujúci s nástrojom PSIPRED (b) – aminokyselinová sekvencia, referenčná sekvencia sekundárnej štruktúry, sekvencia sekundárnej štruktúry predikovaná nástrojom PSIPRED, koeficienty spoľahlivosti predikcie nástroja PSIPRED.

metód `toChromosome`, `fromChromosome` a `nextState` možno dosiahnuť požadované správanie prechodovej funkcie. Spustením metód modulu CASSP – `createEvolutionImage`, `createReliabImage` a `createAccClassesImage` sa vytvoria obrázky popisujúce dosiahnutý výsledok (viď obrázok 6.2) vo formáte PNG [62]. Dáta potrebné pre tvorbu obrázkov sú uložené do textového súboru pre vlastné zobrazenie týchto dát. Pre neskoršie použitie získaného pravidla je implementovaná jeho serializácia, metóda `loadRule` pravidlo načíta, metóda `saveRule` pravidlo uloží do požadovaného súboru. V rámci systému je vytvorený model zapúzdrujúci nástroj PSIPRED triedou `Psipred`. Pri trénovaní/testovaní CASSPu s nástrojom PSIPRED je však nutný iný formát dát (viď obrázok 6.1).

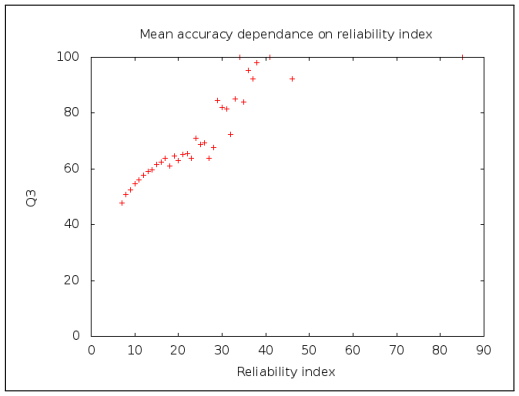
6.3 Webové rozhranie

Webové rozhranie je minimalistické, no spĺňa účel. V rámci webového rozhrania nemožno trénovať nové prechodové pravidlá, pre každý spôsob predikcie je nastavené pravidlo dosahujúce najlepšej úspešnosti predikcie. Bol použitý open-source framework Google Web Toolkit[1] (GWT). ktorý poskytuje nástroje potrebné pre jednoduchú tvorbu a správu JavaScriptových front-endových aplikácií. Ide o server-klient komunikáciu, na strane serveru je použitá technológia Java servletov, na strane klienta sú preložené funkcie aplikačného rozhrania do JavaScriptového kódu. Webovú adresa a bližší popis jednotlivých častí je možné dohľadať v dokumentácii k systému.

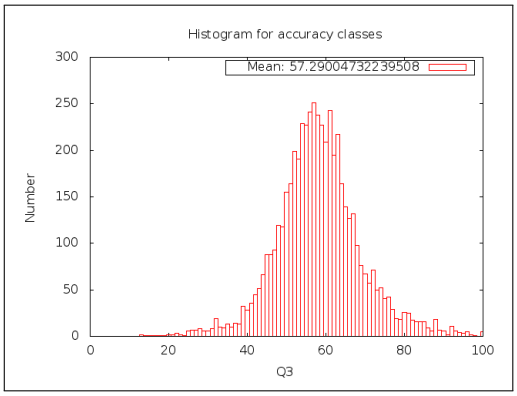
[1] Domovská stránka Google Web Toolkitu: https://developers.google.com/web-toolkit/.



(a) Evolúcia hodnoty fitness funkcie (`createEvolutionImage`).



(b) Úspešnosť predikcie v závislosti na hodnoty spoľahlivosti predikcie (`createReliabImage`).



(c) Počet sekvencií v závislosti na úspešnosti predikcie (`createAccClassesImage`).

Obrázok 6.2: Obrázky popisujúce „výkonnosť“ evolučného algoritmu a vlastnosti predikcie zvolenej dátovej sady.

Kapitola 7

Experimenty

Primárnou snahou experimentovania s navrhnutým modelom bolo zlepšiť úspešnosť predikcie nástroja PSIPRED, ktorý sa radí do tretej generácie metód predikcie sekundárnej štruktúry proteínov (viď sekcia 2.3). Systém je však otestovaný aj ako samostatný prediktor a jeho úspešnosť porovnaná s ostatnými metódami.

Pre dosiahnutie čo najlepších výsledkov je dôležitá rozumná parametrizácia modelu. Pre jej dosiahnutie bola najskôr vykonaná optimalizácia parametra veľkosti okolia, ktoré uvažuje prechodová funkcia CA. Následne bol zistený optimálny maximálny počet krokov CA, ktoré môžu pri evolúcii pravidla CA jednotlivé riešenia nadobúdať. Po správnom „nastavení“ týchto 2 parametrov boli vykonané experimenty predikujúce sekundárnu štruktúru proteínu v spolupráci s nástrojom PSIPRED. Uvažované boli 2 varianty:

1. Primárna predikcia pomocou navrhnutého systému CASSP a následná oprava nie príliš vierohodných predikcii pomocou nástroja PSIPRED.

2. Primárna predikcia pomocou nástroja PSIPRED a následná oprava nie príliš vierohodných predikcií pomocou navrhnutého systému CASSP.

V oboch prípadoch je dôležité správne stanoviť prah opravy primárnej predikcie pomocou predikcie sekundárnej. Pre zistenie vhodného prahu bola vykonaná jeho optimalizácia pre dve varianty:

1. Použitie sekundárneho prediktora pre reziduá, ktorých vierohodnosť predikcie je nižšia než zadaný prah.

2. Použitie sekundárneho prediktora pre celú proteínovú sekvenciu, ak priemerná vierohodnosť reziduí v rámci opravovanej sekvencie je nižšia než zadaný prah.

7.1 Trénovacie a testovacie dátové sady

Pri experimentoch boli použité 3 dátové sady – RS126, CB513 a PDBselect. Dátová sada RS126 bola prvý krát použitá v článku Burkharda Rosta a Chrisa Sandera z roku 1993 [67]. Podľa ich slov ide o nehomológnu dátovú sadu. Nehomológnosť definovali tak, že žiadne 2 proteíny v dátovej sade nesmú mať viac než 25 %-nú zhodu v sekvenciách pri ich dĺžke presahujúcej 80 reziduí. Nevýhodou tohto súboru 126 sekvencií (okrem malého počtu) je, že obsahuje páry proteínových sekvencií, ktoré sú podobné pri provnávaní inými, sofistikovanejšími metódami než obyčajnou percentouálnou zhodou. Výpočet percentuálnej zhody

je totiž závislý na dĺžke zarovnania a zložení sekvencií, takže 2 sekvencie podobného, ale nezvyčajného aminokyselinového zloženia, môžu mať vysokú percentuálnu zhodu, aj keď nie sú evolučne príbuzné [19].

Dátovú sadu CB513 vytvorili páni Geoffrey Barton a James Cuff v rámci svojej štúdie z roku 1999 [19]. Zhodu dvoch aminokyselinových sekvencií, označme ich A a B, neurčovali na základe percentuálnej zhody, ale pomocou metódy, ktorá najskôr zarovná sekvencie štandardným algoritmom dynamického programovania (napríklad pomocou algoritmu Needleman-Wunsch [55]) a získa sa skóre zarovnania V. Poradie jednotlivých aminokyselín v každej proteínovej sekvencii je náhodne zmenené a následne je vykonané zarovnanie pomocou spomínaného algoritmu dynamického programovania. Tento proces sa opakuje typicky aspoň 100 krát, následne sa vypočíta priemer $\bar{x}$ a smerodatná odchýlka σ jednotlivých skóre zarovnania. Výsledná hodnota „podobnosti“ sekvencií A a B, SD, je určená nasledovne: $SD(A, B) = (V - \bar{x})/\sigma$.

Z počiatočnej dátovej sady sa odstránili multisegmentové domény, odstránené boli aj sekvencie, ktorých štruktúry získané pomocou röntgenovej kryštalografie nemali dostatočné rozlíšenie (aspoň 2,5 Å). Ďalej neuvažované boli tiež sekvencie, ktorých podobnosť s nejakou sekvenciu z dátovej sady RS126 bola $SD >= 5$, a sekvencie, ktoré nemali úplnú DSSP definíciu. Táto precízne prefiltrovaná množina sekvencií bola spojená so 126 sekvenciami dátovej sady RS126. Použitá definícia podobnosti však podľa autorov nie je schopná zachytiť všetky homológne sekvencie, na ďalšie porovnávanie a filtrovanie sekvencií bol použitý algoritmus SCOP [53]. Výsledkom bola dátová sada CP513. Pre potreby tejto práce, teda pre jednotné počítanie v sekcii 5.1 bližšie popísaných Chou-Fasmanovych a konformačných koeficientov boli nejednoznačné aminokyseliny (B, Z, X) nahradené priemernými hodnotami: pre B je to priemer hodnôt asparagínu (N) a kyseliny asparagovej (D), pre Z priemer hodnôt glutamínu (Q) a kyseliny glutámovej (E) a pre J priemer hodnôt leucínu (L) a izoleucínu (I).

Treťou použitou dátovou sadou je rozsiahly súbor približne 5 300 proteínových sekvencií z databázy PDB, ktorý bol získaný pomocou nástroja PDBselect [34]. Ide o zoznam reprezentatívnych proteínovych sekvencií s nízkou sekvenčnou podobnosťou (počítanou pomocou HSSP funkcie [1]), ktorý bol vytvorený pre potreby objektívneho štatistického vyhodnodocania okrem iného aj predikcie štruktúry proteínov. Na zarovnanie sekvencií bol využitý rýchly Huang-Miller algoritmus [38]. Jednotlivé selekcie zo zoznamu boli získané pomocou webovej služby MRS [36]. Vzhľadom k rozsiahlosti dátovej sady bola v kontexte tejto práce využívaná iba na testovanie.

Keďže vývoj v oblasti bioinformatiky je veľmi rýchly a počet záznamov v PDB rastie exponenciálne, možno považovať databázy RS126 a CB513 za zastaralé a pre reálne praktické použitie by sa siahlo po novšej, viac aktualizovanej databáze – napríklad spomínanej PDBselect. No pre základnú charakteristiku navrhnutého modelu sú tieto 2 dátové sady dostačujúce, navyše, takmer všetky nástroje predikcie sekundárnej štruktúry spomenuté v kapitole 2 pracujú práve s týmito dátovými sadami, takže je možné priame porovnanie úspešností.

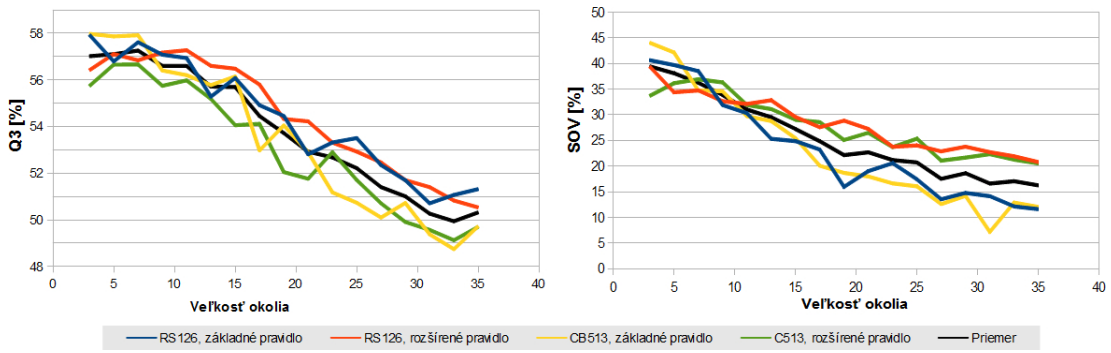
V rámci korektného popisu prediktora je veľmi dôležitý spôsob vyhodnocovania jeho úspešnosti. Azda najdôležitejšou podmienkou je, aby trénovacie a testovacie dáta nekorelovali, čo v reálnych podmienkach nie je jednoduché dosiahnúť, kedy často nie je dostatok dát a ich rozloženie je neznáme. Na mieste je teda otázka, ako ideálne rozdeliť dátovú sadu na trénovaciu a testovaciu tak, aby sme získali čo možno najdôveryhodnejšiu hodnotu úspešnosti resp. chybovosti predikcie? Simone Borra a Agostino Di Ciaccio vo svojej práci [10] došli k záveru, že najvernejšiu hodnotu chyby prediktoru pre reálne dátové sady vykazuje 10-stupňová cross-validácia pre viac než 100 vzorkov. Leave-One-Out (LOO) cross-validácia

teoreticky vykazuje objektívnejší výsledok, no pri testovaní vzhľadom k tomu, že testovacie dátové sady obsahujú iba 1 prvok, sa prejavuje veľká variabilita, čo robí problémy pri selekcii najlepšieho dielčieho modelu. Prístup 10-stupňovej cross-validácie bude teda použitý (pokiaľ nebude povedané inak) pri vyhodnocovaní úspešnosti jednotlivých experimentov. Pri získanie lepšej vierohodnosti je cross-validácia spúšťaná 3-krát a spriemerovaním je získaná výsledná úspešnosť. Na vlastné vyhodnotenie podobnosti referenčných a predikovaných sekundárnych štrukúr proteínových sekvencií boli použité 2 miery - Q_3 a SOV, bližšie popísané v sekcii 2.4.

7.2 Optimalizácia parametrov modelu

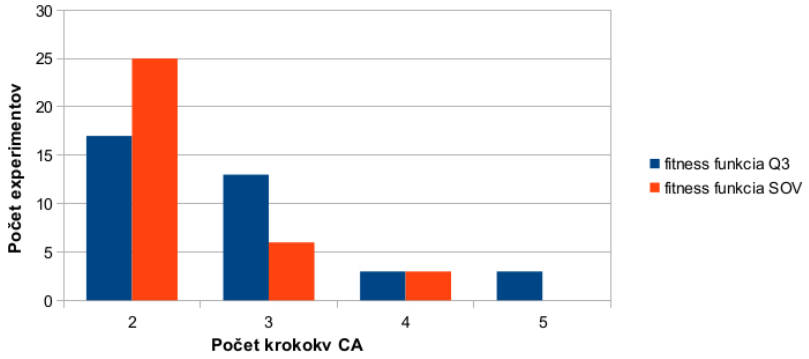
Ako bolo uvedené v úvode tejto kapitoly, pre získanie lepších výsledkov sú optimalizované dva základné parametre navrhnutého modelu – veľkosť okolia, s ktorým pracuje prechodová funkcia celulárneho automatu, a maximálny počet krokov CA (po ktorých sa získa predikovaná sekvencia), ktorý je dosiahnuteľný v rámci evolúcie evolučného algoritmu.

Ako je možné vidieť na obrázku 7.1, so zväčšujúcim sa okolím úspešnosť predikcie klesá, čo je čiastočne zrejme zapríčinené tým, že nebol dostatočný čas na trénovanie, predsalen stavový priestor pri väčších hodnotách okolia značne narastá. Ale na druhej strane to sčasti potvrdzuje slová štúdie z roku 1999 [57], ktorá hovorí, že na determináciu motívu centrálneho rezidua stačí okolie 14–17 a prídavná informácia môže byť nadbytočnou a kontraproduktívnou. Ako optimálne sa v tomto prípade javí *okolie o veľkosti 7*, ktoré bude použité v nasledujúcich experimentoch.

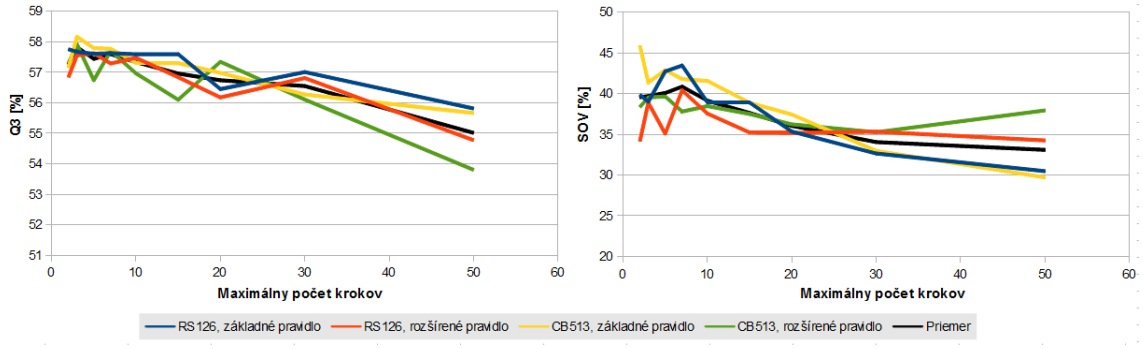


Obrázok 7.1: Klesajúca tendencia úspešnosti predikcie. Natrénované boli modely so základným aj rozšíreným pravidlom CA, s dátovou sadou RS126 aj CB513.

Dôležitým parametrom navrhnutého modelu je počet krokov, po ktorých celulárny automat určuje motívy sekundárnej štruktúry aminokyselinovej sekvencie. Dá sa predpokladať, že pri vyššom počte krokov má celulárny automat šancu zahrnúť do svojej predikcie potenciálne globálne interakcie jednotlivých buniek a tým pádom zlepšiť úspešnosť predikcie modelu. No z výsledkov vizualizovaných na obrázku 7.2 vyplýva, že viac počet krokov CA úspešnosti nepomáha. Navyše, obrázok 7.3 ukazuje, že potrebný počet krokov CA je naozaj minimálny, a teda že evolučný algoritmus nikdy nedokonvergoval do stavu, kedy by sa javil ako optimálny počet krokov celulárneho automaty vyšší než 5. Preto bude v ďalšom skúmaní nastavený *maximálny počet krokov na 5*, tým sa zmenší možný stavový priestor evolučného algoritmu a tým pádom môžeme potenciálne získať lepšie riešenia v kratšej dobe.



Obrázok 7.2: Počet experimentov v závislosti na natrénovanom počte krokov.



Obrázok 7.3: Závislosť úspešnosti predikcie na nastavenom maximálnom počte krokov CA v rámci evolučného algoritmu.

7.3 Systém ako samostatný prediktor

V rámci tejto časti je otestovaná výkonnosť samostatného prediktoru CASSP. Trénovanie prebiehalo na dátových sadách RS126 a CB513 s optimalizovanými parametrami modelu z predchádzajúcej sekcie 7.2. V rámci cross-validácie vzniklo viacero predikčných modelov, ktorých úspešnosť predikcie bola spriemerovaná. Najlepšie predikčné modely boli otestované na dátovej sade získanej pomocou nástroja PDBselect, ktorá je omnoho väčšia a poskytuje vierohodnejší výsledok o úspešnosti predikcie. Táto dátovú sada bude v ďalšom texte označovaná ako „PDBselect“. Ideálne by bolo na tejto dátovej sade vykonať cross-validáciu, z časových dôvodov to však nebolo reálne, no výsledky najlepších prechodových funkcií CA dávajú dobrú informáciu o limitoch navrhnutého systému.

V tabuľke 7.1 sú uvedené výsledky predikcie pre použitú dátovú sadu RS126 a CB513. Tabuľka 7.2 uvádza úspešnosť predikcie najlepších modelov v rámci cross-validácie pri použití týchto sád. Grafická podoba porovnania jednotlivých úspešností je zobrazená na obrázku 7.4. Úspešnosti vyhodnocované mierou podobnosti Q_3 sa pohybujú na rovnakej úrovni, no úspešnosť predikcie vyhodnocovanej pomocou miery SOV dáva pre dátovú sady PDBselect omnoho lepšie výsledky.

V tabuľke 7.3 sú pre najlepšie úspešnosti predikcie pre mieru Q_3 a SOV (zvýraznené v tabuľke 7.2) uvedené prechodové funkcie. Obrázok 7.5 ilustruje vlastnosti predikcie z pohľadu úspešnosti pre jednotlivé triedy vierohodnosti predikcie (7.5b, 7.5d) a z pohľadu

Dátová sada	RS126				CB513			
Pravidlo	základné		rozšírené		základné		rozšírené	
Podobnosť	Q3	SOV	Q3	SOV	Q3	SOV	Q3	SOV
Celkovo	57, 125	42, 688	57, 414	38, 363	57, 976	42, 696	56, 938	38, 244
Coil	63, 044	42, 031	63, 661	36, 670	61, 792	40, 860	61, 924	38, 040
β-sheet	46, 669	44, 244	47, 363	40, 600	47, 247	41, 498	47, 334	38, 372
α-helix	56, 254	39, 850	55, 251	34, 621	59, 606	40, 136	57, 442	33, 535

Tabuľka 7.1: Úspešnosť predikcie systému CASSP ako samostatného prediktora.

Dátová sada	PDBselect (RS126)				PDBselect (CB513)			
Pravidlo	základné		rozšírené		základné		rozšírené	
Podobnosť	Q3	SOV	Q3	SOV	Q3	SOV	Q3	SOV
Celkovo	**57, 276**	**48, 949**	57, 163	48, 547	57, 208	48, 831	57, 142	48, 810
Coil	61, 660	49, 672	63, 975	50, 517	61, 704	49, 346	61, 337	50, 214
β-sheet	45, 563	48, 066	47, 898	50, 562	49, 098	50, 170	49, 170	51, 326
α-helix	60, 201	48, 729	55, 153	45, 794	57, 579	47, 672	57, 778	46, 314

Tabuľka 7.2: Úspešnosť predikcie najlepších modelov získaných cross-validáciou dátových sád RS126 a CB513. Úspešnosť je v tomto prípade vyhodnocovaná na dátovej sade PDBselect.

rozloženia počtu sekvencií vzhľadom k ich úspešnosti (7.5a, 7.5c). V tabuľke 7.4 je uvedené porovnanie s vybranými metódami predikcie sekundárnej štruktúry proteínov, úspešnosť je uvádzaná pre dátovú sadu RS126.

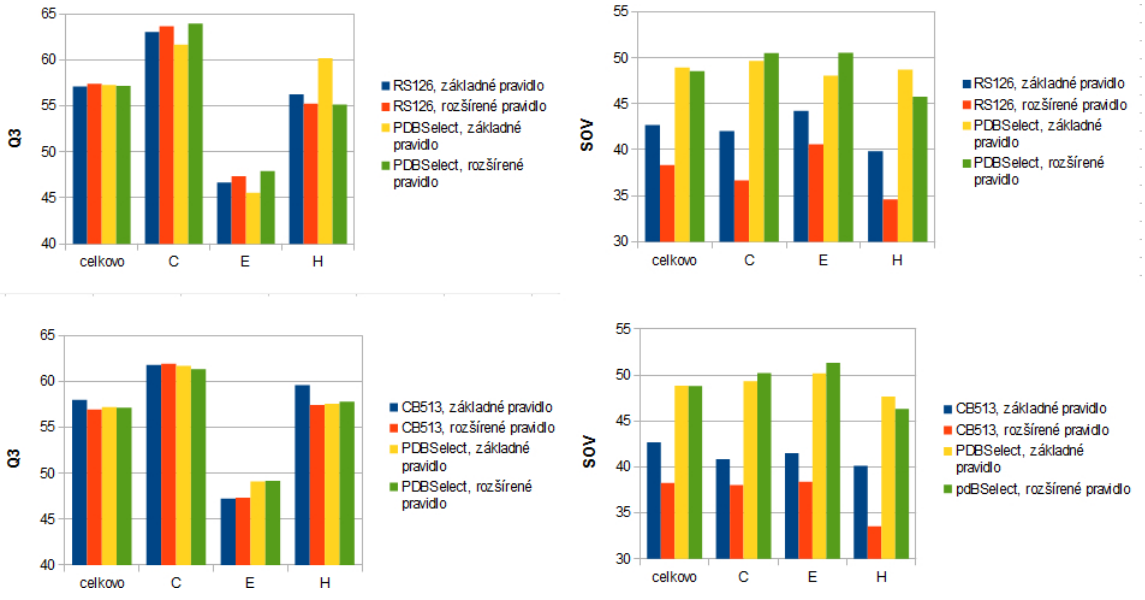
Najlepšie pravidlo pre mieru Q_3 ($Q_3 = 57, 276$)							
Počet krokov	w_{-3}	w_{-2}	w_{-1}	w_0	w_1	w_2	w_3
2	0, 059	0, 328	0, 758	0, 988	0, 726	0, 318	0, 104
Najlepšie pravidlo pre mieru SOV ($SOV = 48, 949$)							
Počet krokov	w_{-3}	w_{-2}	w_{-1}	w_0	w_1	w_2	w_3
2	0, 056	0, 318	0, 777	0, 984	0, 767	0, 372	0, 082

Tabuľka 7.3: Parametre najlepších pravidiel pre úspešnostné miery Q_3 a SOV.

7.4 Systém ako primárny prediktor

V tejto časti je overovaná hypotéza, že navrhnutý systém s optimalizovanými parametrami veľkosti okolia a maximálneho počtu krokov CA je schopný v spolupráci s nástrojom PSIPRED zlepšiť úspešnosť tohto nástroja samotného.

Ako názov sekcie napovedá, primárnym prediktorom bude v tomto prípade systém CASSP a sekundárnym nástroj PSIPRED. Počet vierohodnostných tried je možné v implementovanom systéme meniť, ako vhodný bol určený tento počet na 1000. Nebola uvažovaná



Obrázok 7.4: Porovnanie úspešností získaných cross-validáciou dátových sád RS126 a CB513 s úspešnosťou najlepších modelov v rámci cross-validácie na dátovej sade PDBselect. Sú zobrazené celkové úspešnosti, ale aj úspešnosti jednotlivých motívov sekundárnej štruktúry, teda motívu α-helix (H), β-sheet (E) a Coil (C).

Metóda	Q_3	SOV	Princíp metódy
CASSP	$57,4$	$42,7$	Model celulárneho automatu
PREDATOR	$70,3$	$69,9$	Neurónové siete
PHD	$70,8$	$73,5$	Neurónové siete
DSC	71.1	$71,6$	Štatistická metóda
NNSSP	$72,7$	$70,6$	Metóda najbližších susedov

Tabuľka 7.4: Porovnanie úspešnosti vybraných metód predikcie sekundárnej štruktúry proteínov.

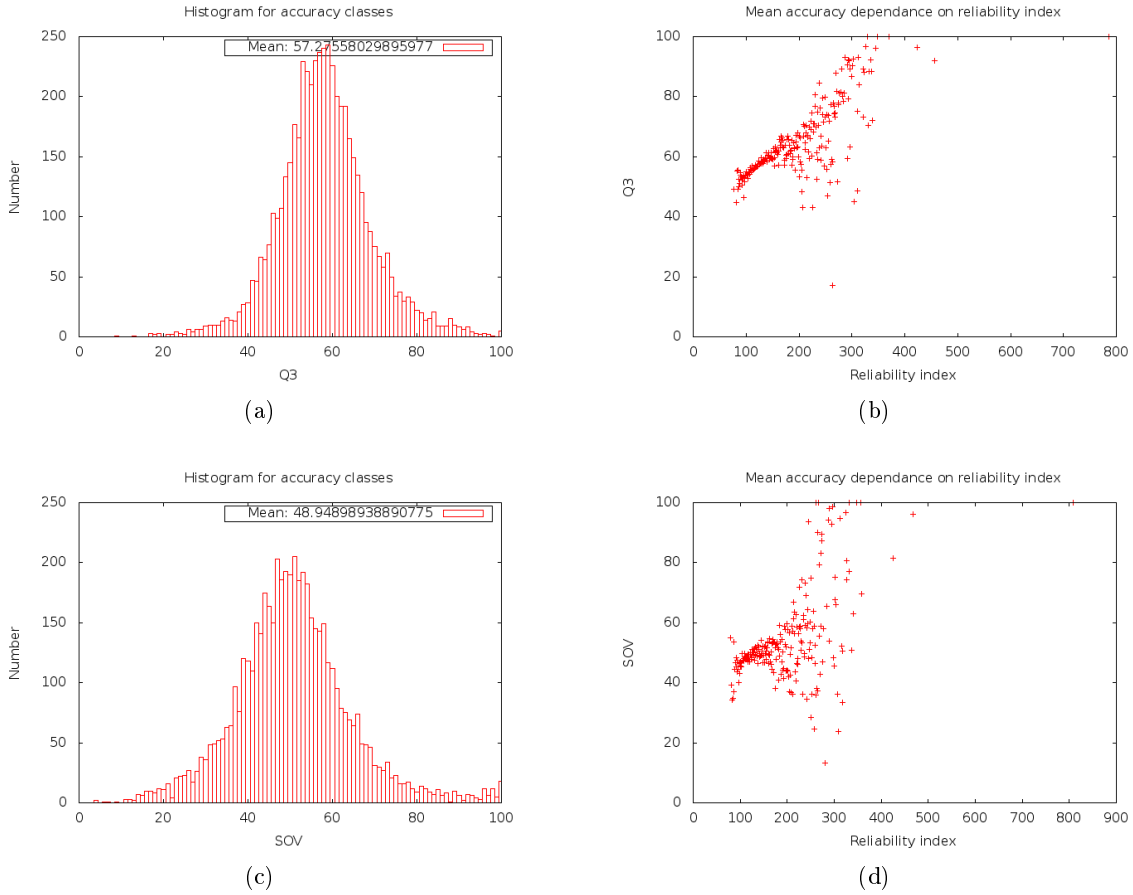
rozšírená prechodová funkcia vzhľadom k jej nelinearite a tým pádom aj nepredvídateľnosti maximálnej hodnoty vierohodnosti. Na základe empirickej analýzy je výpočet maximálnej vierohodnosti MV pre základnú prechodovú funkciu nasledovný:

$$MV = \frac{\max CF \cdot n}{3}, \tag{7.1}$$

kde $\max CF$ je maximálna hodnota Chou-Fasmanových koeficientov a n veľkosť okolia, ktoré uvažuje prechodová funkcia CA.

7.5 Systém ako sekundárny prediktor

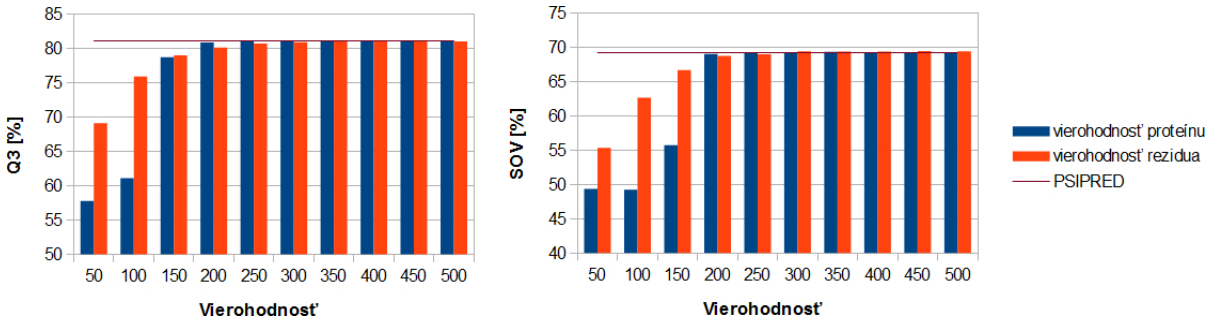
Druhým spôsobom ponímania spolupráce systémov CASSP a PSIPRED je určiť ako primárny prediktor PSIPRED a ako sekundárny CASSP. PSIPRED udáva vierohodnosť svojej



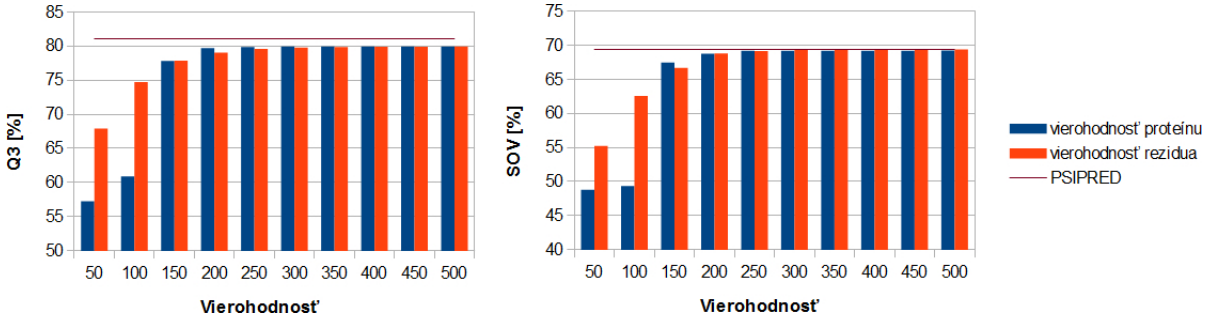
Obrázok 7.5: (a) a (c) ilustruje vlastnosti predikcie z pohľadu rozloženia počtu sekvencií vzhľadom k ich úspešnosti, (b) a (d) z pohľadu úspešnosti pre jednotlivé triedy vierohodnosti predikcie. (a) a (b) reprezentujú najlepšie pravidlo pre mieru Q_3, (c) a (d) pre mieru SOV.

predikcie v škále od 0 do 9. Úlohou je opäť vhodné nastavenie prahu tak, aby málo vierohodné predikcie dokázal systém CASSP čo možno najsprávnejšie opravovať. Opäť sú uvažované dva spôsoby opravy – na základe priemernej vierohodnosti celej proteínovej sekvencie alebo na základe vierohodnosti každého rezidua zvlášť.

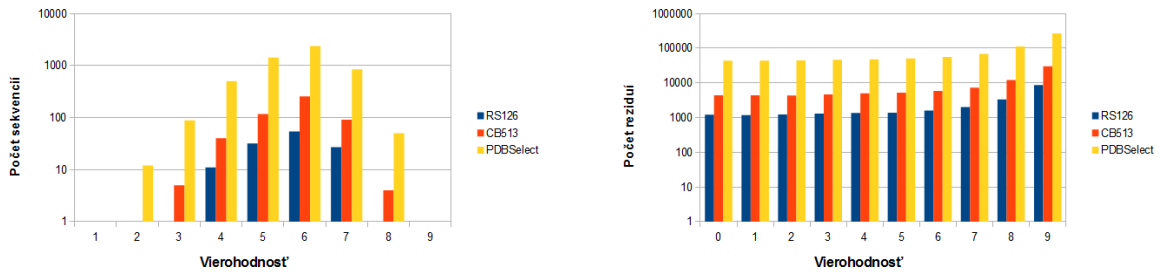
Obrázok 7.8a ukazuje, že nie všetky intervaly hodnôt vierohodnosti proteínových sekvencií majú signifikantné zastúpenie, preto nemá zmysel sa nimi zaoberať. No pre opravu jednotlivými reziduami má zmysel sa zaoberať všetkými úrovňami prahu vierohodnosti (viď obrázok 7.8b). Prediktor CASSP sa trénoval na dátovej sade CB513. Úspešnosť v porovnaní s príslušnou úspešnosťou nástroja PSIPRED zobrazuje obrázok 7.9. Najlepšie modely boli opäť otestované na dátovej sade PDBselect (obrázok 7.10), no zlepšenia sa však nepodarilo dosiahnúť. Určité zlepšenie však je vidielné pri predikcii motívov sekundárnej štruktúry α-helix pri použití miery Q_3 (viď obrázok 7.11).



Obrázok 7.6: Porovnanie úspešností systému CASSP (ako primárneho prediktora) získaných cross-validáciou dátovej sady CB513 so samostatným nástrojom PSIPRED (celkovo).

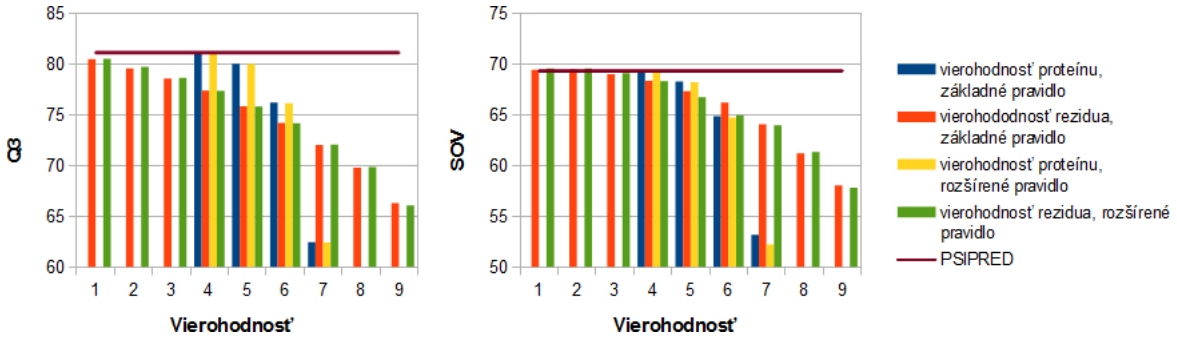


Obrázok 7.7: Úspešnosť predikcie najlepších modelov (CASSP ako primárny prediktor) získaných cross-validáciou dátovej sady CB513. Úspešnosť je v tomto prípade vyhodnocovaná na dátovej sade PDBselect.

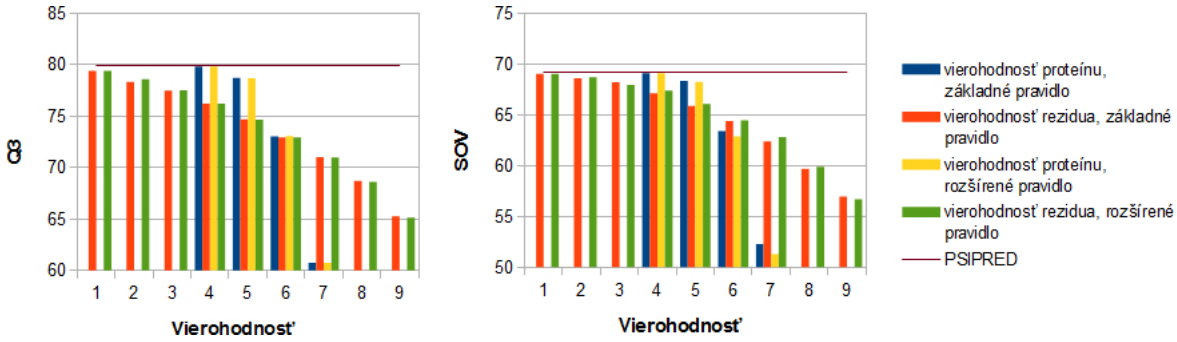


(a) (b)

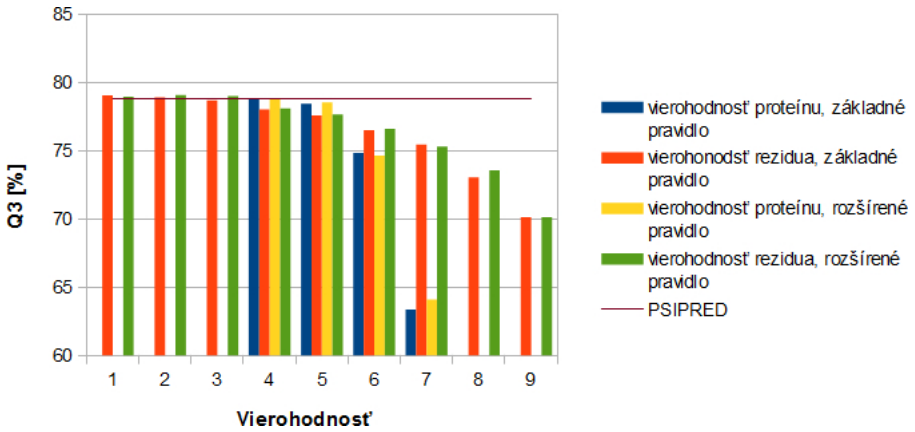
Obrázok 7.8: Počet sekvencií (a) resp. počet reziduí (b) v závislosti na prahu vierohodnosti. Pre celé sekvencie je vierohodnosť počítaná ako priemer vierohodností všetkých jej reziduí.



Obrázok 7.9: Porovnanie úspešností (CASSP ako sekundárny prediktor) získaných cross-validáciou dátovej sady CB513 so samostatným nástrojom PSIPRED (celkovo).



Obrázok 7.10: Úspešnosť predikcie najlepších modelov (CASSP ako sekundárny prediktor) získaných cross-validáciou dátovej sady CB513. Úspešnosť je v tomto prípade vyhodnocovaná na dátovej sade PDBselect.



Obrázok 7.11: Porovnanie úspešností získaných cross-validáciou dátovej sady PDBselect so samostatným nástrojom PSIPRED (len pre motívy α-helix).

Kapitola 8

Záver

Proteíny sú základnými stavebnými kameňmi života na Zemi, starajú sa o podstatnú časť biologických funkcí a ich reguláciu. Funkcia proteínov je určená ich štruktúrou a predikciou (sekundárnej) štruktúry sa venovala táto práca. Bol navrhnutý predikčný model založený na modeli celulárneho automatu, na ktorého parametre (počet krokov, váhy okolitých buniek atď.) boli kvôli netriviálnej optimalizácii využité služby evolučných algoritmov, konkrétne evolučných stratégii. Boli navrhnuté dve prechodové funkcie modelu celulárneho automatu, *základná*, využívajúca prístup pánov Chopru a Bendera [13] a ich Chou-Fasmanove koeficienty, a *rozšírená*, ktorá okrem Chou-Fasmanových koeficientov využíva tzv. konformačné preferencie, ktoré popisujú preferencie jednotlivých aminokyselín začínať alebo končiť určitý motív sekundárnej štruktúry [79]. Na základe vykonaných experimentov možno povedať, že výsledky oboch pravidiel sa líšili minimálne. Pridané štatistické vlastnosti sú zrejme nejakým spôsobom obsiahnuté v Chou-Fasmanových koeficientoch uvažovaných v základom pravidle, ktoré charakterizujú mieru výskytu jednotlivých aminokyselín v motívoch sekundárnej štruktúry, a tým pádom neprinášajú takmer žiadne nové informácie, ktoré by klasifikácii reziduí pomohli.

Systém bol otestovaný ako samostatný prediktor, bola získaná úspešnosť $Q_3 = 57,414\,\%$ resp. $SOV = 42,688\,\%$ pre dátovú sadu RS126, $Q_3 = 57,796\,\%$ resp. $SOV = 42.696\,\%$ pre dátovú sadu CB513. Bola tiež vytvorená nová dátová sada pomocou nástroja PDBselect, ktorá však vzhľadom k svojej rozľahlosti nebola použitá na trénovanie, no otestované na nej boli najlepšie pravidlá, ktoré vykázali maximálnu úspešnosť $Q_3 = 57,276\,\%$ resp. $SOV = 48,949\,\%$.

Zaujímavou sa javila myšlienka predikcie navrhnutého systému CASSP v spolupráci s nástrojom PSIPRED. Boli uvažované dve varianty – CASSP ako primárny prediktor a PSIPRED ako prediktor sekundárny, a naopak. Spôsom, akým sa nahradzujú primárne predikcie sekundárnymi bol uvažovaný dvojaký – náhrada sekvencií ako celku na základe priemernej vierohodnosti predikcie ich jednolivých reziduí, a náhrada jednotlivých reziduí na základe ich vierohodnosti predikcie. Výsledkom však v oboch prípadoch spoločnej predikcie nebolo zlepšenie úspešnosti nástroja PSIPRED pre všetky motívy sekundárnej štruktúry. Malé zlepšenie (v desatinách percenta) vykázala úspešnosť predikcie motívu Coil. Išlo o rozšírenú prechodovú funkciu, náhrada predikcií bola na úrovni jednotlivých reziduí, prah vierohodnosti bol 2 (alebo 3) a CASSP bol sekundárnym prediktorom. Z tohto dôvodu je možné použiť toto „combo“ pri predikciách, v ktorých požadujeme čo najpresnejšie určenie motívu α-helix.

Návrhy na možné pokračovanie vo výskumu tohto prístupu k predikcii sekunárnej štruktúry proteínov:

- navrhnutie iného spôsobu spolupráce nástroja PSIPRED s navrhnutým prediktorom CASSP, prípadne zmena externého nástroja
- zmena jednoduchej klasifikačnej funkcie, ktorá ako motív sekundárnej štruktúry vyberie ten s najvyššou hodnotou predispozícii ním byť
- inovatívny návrh prechodovej funkcie celulárneho automatu, zavedenie dômyselnejších nelinearít
- uvažovanie ďalších informácii o proteínovej sekvencii, napríklad chemické posuny, prípadne uvolučné informácie, čo by ale spomalilo predikciu, no vhodná kombinácia prídavných informácii by mohla signifikantne zlepšiť úspešnosť predikcie

Literatúra

[1] Abagyan, R. A.; Batalov, S.: Do aligned sequences share the same fold? *Journal of Molecular Biology*, ročník 273, č. 1, 1997: s. 355–368, ISSN 0022-2836.

[2] Abdoun, O.; Abouchabaka, J.: A comparative study of adaptive crossover operators for genetic algorithms to resolve the traveling salesman problem. *International Journal of Computer Applications*, ročník 31, č. 11, 2011.

[3] Alberts, B.; Bray, D.; Johnson, A.; aj.: *Základy buněčné biologie: úvod do molekulární biologie buňky*. Ústí nad Labem: Espero Publishing s.r.o, druhé vydání, 2005, ISBN 978-80-902906-2-0, 740 s.

[4] Altschul, S. F.; Gish, W.; Miller, W.; aj.: Basic local alignment search tool. *Journal of Molecular Biology*, ročník 215, č. 3, 1990: s. 403–410, ISSN 0022-2836.

[5] Altschul, S. F.; Madden, T. L.; Schäffer, A. A.; aj.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, ročník 25, č. 17, 1997: s. 3389–3402.

[6] Bagos, P. G.; Tsaousis, G. N.; Hamodrakas, S. J.: How many 3D structures do we need to train a predictor? *Genomics, Proteomics & Bioinformatics*, ročník 7, č. 3, 2009: s. 128–137, ISSN 1672-0229.

[7] Banks, E. R.: Information processing and transmission in cellular automata. Technická zpráva, Cambridge, MA, USA, 1971.

[8] Berman, H. M.; Westbrook, J.; Feng, Z.; aj.: The Protein Data Bank. *Nucleic Acids Res*, ročník 28, 2000: s. 235–242.

[9] Birney, E.; Stamatoyannopoulos, J. A.; Dutta, A.; aj.: Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature*, ročník 447, č. 7146, 2007: s. 799–816.

[10] Borra, S.; Ciaccio, A. D.: Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods. *Computational Statistics & Data Analysis*, ročník 54, č. 12, 2010: s. 2976–2989, ISSN 0167-9473.

[11] Brigant, V.: *Evoluční návrh simulátoru založeného na celulárních automatech*. Bakalářská práce, FIT VUT v Brně, Brno, 2011.

[12] Cecchini, A.; Rinaldi, E.: The multi-cellular automaton: a tool to build more sophisticated models. A theoretical foundation and a practical implementation. 1999.

[13] Chopra, P.; Bender, A.: Evolved cellular automata for protein secondary structure prediction imitate the determinants for folding observed in nature. *In Silico Biol*, ročník 7, č. 1, 2007: s. 87–93.

[14] Chou, P. Y.; ; Fasman, G. D.: Prediction of protein conformation. *Biochemistry*, ročník 13, č. 2, jan 1974: s. 222–245.

[15] Chou, P. Y.; Fasman, G. D.: Conformational parameters for amino acids in helical, β-sheet, and random coil regions calculated from proteins. *Biochemistry*, ročník 13, č. 2, jan 1974: s. 211–222.

[16] Codd, E. F.: *Cellular automata*. Orlando, FL, USA: Academic Press, Inc., 1968, ISBN 978-0-1217-8850-4.

[17] Cole, C.; Barber, J. D.; Barton, G. J.: The Jpred 3 secondary structure prediction server. *Nucleic Acids Research*, ročník 36, č. 2, 2008: s. 197–201.

[18] Crick, F. H.; Barnett, L.; Brenner, S.; aj.: General nature of the genetic code for proteins. *Nature*, ročník 192, Prosinec 1961: s. 1227–1232, ISSN 0028-0836.

[19] Cuff, J. A.; Barton, G. J.: Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, ročník 34, č. 4, 1999: s. 508–519, ISSN 1097-0134.

[20] Darwin, C.: *Pôvod druhov*. Kalligram, 2006, ISBN 978-80-7149-745-2, 542 s.

[21] Delorme, M.; Mazoyer, J.: *Cellular Automata: a parallel model*, ročník 460. Springer, 1998, ISBN 978-0-7923-5493-2.

[22] Dor, O.; Zhou, Y.: Achieving 80 % ten-fold cross-validated accuracy for secondary structure prediction by large-scale training. *Proteins: Structure, Function, and Bioinformatics*, ročník 66, č. 4, 2007: s. 838–845, ISSN 1097-0134.

[23] Ermentrout, B. G.; Edelstein-Keshet, L.: Cellular automata approaches to biological modeling. *Journal of Theoretical Biology*, ročník 160, č. 1, jan 1993: s. 97–133, ISSN 0022-5193.

[24] Fredkin, E.; Toffoli, T.: Collision-based computing. kapitola Conservative logic, London, UK, UK: Springer-Verlag, 2002, ISBN 978-1-85233-540-8, s. 47–81.

[25] Frishman, D.; Argos, P.: Incorporation of non-local interactions in protein secondary structure prediction from the amino acid sequence. *Protein engineering*, ročník 9, č. 2, 1996: s. 133–142.

[26] Froimowitz, M.; Fasman, G. D.: Prediction of the secondary structure of proteins using the helix-coil transition theory. *Macromolecules*, ročník 7, č. 5, 1974: s. 583–589.

[27] Fuqiang, D.: Mining dynamic transition rules of cellular automata in urban population simulation. In *Proceedings of the 2010 Second International Conference on Computer Modeling and Simulation - Volume 02*, ICCMS '10, Washington, DC, USA: IEEE Computer Society, 2010, ISBN 978-0-7695-3941-6, s. 471–474.

[28] Gardner, M.: Mathematical Games The fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, ročník 223, 1970: s. 120–123.

[29] Gardner, M.: *Wheels, life, and other mathematical amusements*. Freeman, 1983, ISBN 978-0-7167-1589-4.

[30] Garnier, J.; Gibrat, J. F.; Robson, B.: GOR method for predicting protein secondary structure from amino acid sequence. *Methods Enzymol*, ročník 266, 1996: s. 540–553.

[31] Ghosh, A.; Parai, B.: Protein secondary structure prediction using distance based classifiers. *International Journal of Approximate Reasoning*, ročník 47, č. 1, Leden 2008: s. 37–44, ISSN 0888-613X.

[32] Goldberg, D. E.: *Genetic algorithms in search, optimization and machine learning*. Boston, MA, USA: Addison-Wesley Longman Publishing, první vydání, 1989, ISBN 978-0-2011-5767-5.

[33] Granseth, E.; Viklund, H.; Elofsson, A.: ZPRED: Predicting the distance to the membrane center for residues in alpha-helical membrane proteins. In *ISMB (Supplement of Bioinformatics)*, 2006, s. 191–196.

[34] Griep, S.; Hobohm, U.: PDBselect 1992-2009 and PDBfilter-select. *Nucleic Acids Research*, ročník 38, 2010: s. 318–319.

[35] Griffiths, A. J. F.; Wessler, S. R.; Lewontin, R. C.; aj.: *Introduction to genetic analysis*. W. H. Freeman, 9 vydání, Únor 2007, ISBN 978-0-7167-6887-9.

[36] Hekkelman, M. L.; Vriend, G.: MRS: a fast and compact retrieval system for biological data. *Nucleic Acids Res*, ročník 33: s. 766–769.

[37] Holland, J. H.: *Adaptation in natural and artificial systems*. Ann Arbor, MI, USA: University of Michigan Press, 1975.

[38] Huang, X.; Miller, W.: A time-efficient, linear-space local similarity algorithm. *Advances in Applied Mathematics*, ročník 12, č. 3, 1991: s. 337–357, ISSN 0196-8858.

[39] Human Genome Program: *Genomics and its impact on science and society: The Human Genome Project and beyond*. U.S. Department of Energy, 2008.

[40] Hynek, J.: *Genetické algoritmy a genetické programování*. Praha 7: Grada Publishing a.s., 2008, ISBN 978-80-247-2695-3.

[41] Jones, D. T.: Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, ročník 292, 1999: s. 195–202.

[42] Kabsch, W.; Sander, C.: Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, ročník 22, č. 12, dec 1983: s. 2577–2637, ISSN 0006-3525.

[43] Kabsch, W.; Sander, C.: How good are predictions of protein secondary structure? *FEBS Letters*, ročník 155, č. 2, 1983: s. 179–182, ISSN 0014-5793.

[44] Kier, L. B.; Bonchev, D.; Buck, G. A.: Modeling biochemical networks: A cellular-automata approach. *Chemistry & Biodiversity*, ročník 2, č. 2, 2005: s. 233–243, ISSN 1612-1880.

[45] Kloczkowski, A.; Ting, K.; Jernigan, R.; aj.: Combining the GOR V algorithm with evolutionary information for protein secondary structure prediction from amino acid sequence. *Proteins*, ročník 49, č. 2, 2002: s. 154–166.

[46] Koza, J. R.: *Genetic programming: On the programming of computers by means of natural selection (complex adaptive systems)*. The MIT Press, první vydání, December 1992, ISBN 978-0-262-11170-5.

[47] Lakizadeh, A.; Marashi, S. A.: Addition of contact number information can improve protein secondary structure prediction by neural networks. *EXCLI Journal*, ročník 8, 2009: s. 66–73.

[48] Langton, C. G.: Self-reproduction in cellular automata. *Physica D: Nonlinear Phenomena*, ročník 10, č. 1–2, 1984: s. 135–144, ISSN 0167-2789.

[49] Lažanský, J.; Mařík, V.; Štěpánková, O.: *Umělá inteligence (3)*. Academia, 2001, ISBN 978-80-200-0472-6, 328 s.

[50] Malone, M. S.: God, Stephen Wolfram, and everything else. *Forbes ASAP*, november 2000: s. 162–180, ISSN 1078-9901.

[51] Mechelke, M.; Habeck, M.: A probabilistic model for secondary structure prediction from protein chemical shifts. *Proteins: Structure, Function, and Bioinformatics*, 2012, ISSN 1097-0134.

[52] Meffert, K.; Rotstan, N.; Knowles, C.; aj.: Jgap-java genetic algorithms and genetic programming package. 2011.
URL http://jgap.sf.net

[53] Murzin, A. G.; Brenner, S. E.; Hubbard, T.; aj.: SCOP: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, ročník 247, č. 4, 1995: s. 536–540, ISSN 0022-2836.

[54] Nečas, O.: *Obecná biologie pro lékařské fakulty*. H & H Vyšehradská, 2000, ISBN 978-80-86022-46-8.

[55] Needleman, S. B.; Wunsch, C. D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, ročník 48, č. 3, 1970: s. 443–453, ISSN 0022-2836.

[56] von Neumann, J.: *Theory of self-reproducing automata*, ročník 160. Illinois: University of Illinois Press, 1966, ISBN 978-0-598-37798-0.

[57] Pan, X. M.; Niu, W. D.; Wang, Z. X.: What is the minimum number of residues to determine the secondary structural state? *Journal of protein chemistry*, 1999: s. 579–584.

[58] Park, T.; Ryu, K. R.: A dual-population genetic algorithm for adaptive diversity control. *Evolutionary Computation, IEEE Transactions on*, ročník 14, č. 6, december 2010: s. 865–884, ISSN 1089-778X.

[59] Pennisi, E.: Genomics. ENCODE project writes eulogy for junk DNA. *Science*, ročník 337, č. 6099, 2012: s. 1159–1161, ISSN 1095-9203.

[60] Pham, T. H.; Satou, K.; Ho, T. B.: Support Vector Machines for prediction and analysis of beta and gamma-turns in proteins. *Journal of Bioinformatics and Computational Biology*, ročník 03, č. 02, 2005: s. 343–358.

[61] Pollastri, G.; Przybylski, D.; Rost, B.; aj.: Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins: Structure, Function, and Bioinformatics*, ročník 47, č. 2, 2002: s. 228–235, ISSN 1097-0134.

[62] Roelofs, G.: PNG Documentation. 2010, [online],[cit. 2012-05-11].
URL http://www.libpng.org/pub/png/pngdocs.html

[63] Rost, B.: Review: Protein secondary structure prediction continues to rise. *J. Struct. Biol*, ročník 134, 2001: s. 204–218.

[64] Rost, B.: Protein Prediction - Part 1: Structure. University Lecture, 2011, [online],[cit. 2012-05-12].

[65] Rost, B.; Eyrich, V. A.: EVA: Large-scale analysis of secondary structure prediction. ročník 5, 2001: s. 192–199.

[66] Rost, B.; Sander, C.: Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, ročník 90, č. 16, 1993: s. 7558–7562, ISSN 0027-8424.

[67] Rost, B.; Sander, C.: Prediction of protein secondary structure at better than 70 % accuracy. *Journal of Molecular Biology*, ročník 232, 1993: s. 584–599.

[68] Rost, B.; Sander, C.; Schneider, R.: Redefining the goals of protein secondary structure prediction. *Journal of Molecular Biology*, ročník 235, č. 1, 1994: s. 13–26, ISSN 0022-2836.

[69] Rost, B.; Zemla, A.; Fidelis, K.; aj.: A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins: Structure, Function, and Genetics*, ročník 34, 1999: s. 220–223.

[70] Schaffer, J. D.; Caruana, R.; Eshelman, L. J.; aj.: A study of control parameters affecting online performance of genetic algorithms for function optimization. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, ISBN 1-55860-066-3, s. 51–60.

[71] Schulz, G. E.; Pain, R. H.; Schirmer, R. H.: Principles of protein structure. *Biochemical Education*, ročník 8, č. 4, 1980: s. 108–130, ISSN 1879-1468.

[72] Sipper, M.: *Evolution of parallel cellular machines: The cellular programming approach*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2001, ISBN 978-3-540-62613-8.

[73] Toffoli, T.; Margolus, N.: *Cellular automata machines, a new environment for modeling*. Cambridge, MA: MIT Press, 1987.

[74] Šalanda, V.: *Predikce sekundární struktury proteinu pomocí celulárního automatu.* Bakalářská práce, FIT VUT v Brně, Brno, 2012.

[75] Wolfram, S.: Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, ročník 10, č. 1–2, 1984: s. 1–35, ISSN 0167-2789.

[76] Wolfram, S.: *A new kind of science.* Wolfram Media, January 2002, ISBN 978-1-57955-008-8, 1197 s.

[77] Xiao, X.; Shao, S.; Ding, Y.; aj.: Using cellular automata images and pseudo amino acid composition to predict protein subcellular location. *Amino Acids*, ročník 30, 2006: s. 49–54, ISSN 0939-4451.

[78] Yang, B.; Hou, W.; Xie, Y.; aj.: The research of protein secondary structure prediction system based on KDTICM. *Proceedings of The World Congress on Engineering and Computer Science*, 2009: s. 47–51.

[79] Zhang, G. Z.; Huang, D. S.; Zhu, Y. P.; aj.: Improving protein secondary structure prediction by using the residue conformational classes. *Pattern Recogn. Lett.*, ročník 26, č. 15, nov 2005: s. 2346–2352, ISSN 0167-8655.