VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

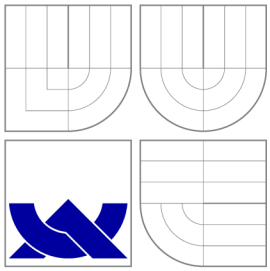
PREDIKCE SEKUNDÁRNÍ STRUKTURY PROTEINŮ POMOCÍ CELULÁRNÍHO AUTOMATU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

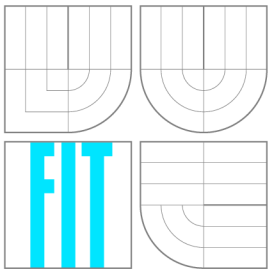
AUTOR PRÁCE
AUTHOR

VOJTĚCH ŠALANDA

BRNO 2012



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 CELULÁRNÍHO AUTOMATU

PREDICTION OF THE SECONDARY STRUCTURE OF PROTEINS BY CELLULAR AUTOMATON

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — VOJTĚCH ŠALANDA
AUTHOR

VEDOUCÍ PRÁCE — Ing. JAROSLAV BENDL
SUPERVISOR

BRNO 2012

Abstrakt

Tato práce přináší nový přístup k predikci sekundární struktury proteinů. K existujícím přístupům k této problematice zavádí novou metodu, založenou na celulárním automatu a jeho charakteristických vlastnostech. Hlavním cílem práce je zvýšení rychlosti predikce i za cenu mírného snížení její úspěšnosti. Pro nalezení optimálních parametrů běhu celulárního automatu je využito genetického algoritmu s vhodně definovanými genetickými operátory. Dosažené výsledky byly porovnány s výsledky existujících metod. Parametry predikce, se kterými bylo dosaženo nejvyšší úspěšnosti, byly použity ve výpočetním frameworku pro predikci sekundární struktury libovolného analyzovaného proteinu.

Abstract

This thesis presents a new approach to the prediction of the secondary structure of proteins. It employs a new method based on cellular automata and its characteristic properties. The main objective is to increase speed of the prediction even at the cost of slight decrease of overall accuracy. Optimal parameters of cellular automata was found by genetic algorithm using suitable genetic operators. These parameters are incorporated into developed application for prediction. Finally, the results was compared with results of other tools for this purpose.

Klíčová slova

Celulární automat, genetický algoritmus, predikce sekundární struktury, protein.

Keywords

Cellular automata, genetic algorithm, secondary structure prediction, protein.

Citace

Vojtěch Šalanda: Predikce sekundární struktury proteinů pomocí celulárního automatu, bakalářská práce, Brno, FIT VUT v Brně, 2012

Predikce sekundární struktury proteinů pomocí celulárního automatu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením ing. Jaroslava Bendla a uvedl jsem všechny literární prameny, ze kterých jsem čerpal.

.....
Vojtěch Šalanda
14. května 2012

Poděkování

Vřele děkuji svému vedoucímu panu ing. Jaroslavu Bendlovi za vedení při přípravě a vypracování této práce. Jeho důsledný přístup, připomínky a rady mi pomohly v řešení mnohých problémů.

Obsah

Kapitola 1

Úvod

Hlavním úkolem bioinformatiky je studium, analýza a zpracování biologických dat. Velmi zásadní část těchto dat tvoří proteiny, u kterých se zkoumají zejména jejich struktury. Predikce těchto struktur a vývoj nástrojů pro predikci je jedním z nejdůležitějších úkolů při zkoumání vlastností a využití proteinů.

Navzdory pokroku ve výzkumu nástrojů pro predikci proteinových struktur lze stále optimalizovat jejich postupy za účelem zkvalitnění výsledků v závislosti na využitých prostředcích. Důležitým aspektem je v tomto případě rychlost predikce. V této práci se zabývám predikcí sekundární struktury, přičemž hlavními cíli jsou rychlost a dostatečná přesnost. Přesnost predikce vychází z porovnání s experimentálně zjištěnými výsledky. Samotná sekundární struktura je mezikrok pro predikci vyšších struktur, jejichž složitost je natolik vysoká, že znalost sekundární struktury značně urychlí navazující výpočty.

Existuje mnoho metod a popsaných postupů, jak predikovat sekundární strukturu. Využití celulárního automatu je zcela nový přístup. Celulární automat jako dynamický systém poskytuje výhody, kterých lze využít pro efektivní výpočet a možné zpřesnění již existujících metod. Jeho souvislost s biologií a schopnost vzájemné interakce sousedních buněk automatu jsou vhodnými prostředky, kterých lze v této práci využít.

Cílem určování struktur je schopnost zařadit daný protein do určité proteinové rodiny. Podle těchto rodin lze určit vnější a vnitřní chování obsažených proteinů. Tyto vlastnosti slouží jako zdroj dat pro biologické databáze a následně pro širší využití při dalším výzkumu.

Ve druhé kapitole se zabývám celulárními automaty. Tyto automaty jsou použity jako prostředek pro predikci sekundární struktury proteinů. Tento abstraktní model pracuje v diskrétním čase a s diskrétními hodnotami. Automat je N-rozměrná mřížka buněk, které interagují mezi sebou. Každá buňka v diskrétních časových okamžicích mění svůj stav na nový a tento přechod je ovlivněn stavy okolních buněk.

Třetí kapitola je věnována genetickému algoritmu, na jehož principech je založeno hledání nejlepších vstupních parametrů pro simulační model tak, aby bylo dosaženo co nejlepších výsledků. Toto hledání parametrů je založeno na jejich vzájemné kombinaci.

Ve čtvrté kapitole popisuji proteiny a jejich struktury. Podrobně jsou zmíněny biologické údaje, ze kterých vyplývají důležité znalosti o proteinu jako simulačním modelu. Dále popisuji známé a používané metody pro predikci sekundární struktury a jejich charakteristiky.

Pátá kapitola je věnována vlastní implementaci výpočetního frameworku. Uvedeny jsou použité techniky přístupu k okrajovým podmínkám celulárního automatu, řešení genetického algoritmu a použité datasety a techniky porovnání úspěšnosti.

Šestá kapitola je věnována experimentům, prováděných za účelem nalezení optimálních parametrů predikce. Dále zde provádím hodnocení dosažených výsledků a srovnání s vý-

sledky jiných metod. Toto srovnání a hodnocení a návrh dalších možných postupů je shrnuto v závěru společně s celkovým shrnutím přístupu k celé problematice.

Kapitola 2

Celulární automaty

Celulární automat je dynamický diskrétní, spojitý či stochastický systém vzájemně se ovlivňujících buněk. Typicky je to diskrétní systém diskrétních hodnot v diskrétním čase i prostoru. V N-rozměrném prostoru ho definujeme jako pravidelnou strukturu prvků (buněk). Struktury nebo též pole buněk (lattice) mohou být konečné i nekonečné. Tato pole rovnoměrně rozdělují prostor. Pokud se jedná obecně o N-dimenzionální prostor, hovoříme o entitách automatu jako o prvcích. Pojem buňka je nejčastěji spojen s 2D celulárním automatem, kdy jsou buňky orientovány do čtvercové mřížky.

Každý z prvků automatu je konečný K-stavový automat, tedy každý z prvků se v daném diskrétním časovém okamžiku nachází v jednom z K stavů. V následujícím časovém okamžiku se pro každý prvek v celé struktuře paralelně vypočítá nový stav na základě lokální přechodové funkce, která je pro každý prvek automatu stejná. Výsledek je ovlivněn vstupními argumenty, jimiž jsou aktuální stav počítaného prvku a aktuální stavy sousedících prvků v obecně libovolně velkém okolí (neighbourhood). V 1D automatu určuje počet sousedů po obou stranách tzv. poloměr. V případě 2D automatu lze popsat několik typů okolí.

Podle obrázku 2.1 definujeme 2 základní typy okolí. Von Neumannovo okolí sestává ze 4 sousedů po stranách a Moorovo okolí ze 4 sousedů po stranách a 4 sousedů na diagonálách. Obě tato okolí lze rozšířit na širší (extended) varianty [23]. Ve vícerozměrných okolích jsou tyto typy obdobné. V obecném pojetí vyplňuje struktura celý nekonečný prostor, po omezení prostoru předpokládáme okrajové buňky. Tyto buňky lze nastavit jako nulové, nebo lze vytvořit smyčku v případě 1D automatu nebo anuloid v případě 2D okolí.

Na základě těchto specifik lze celulární automaty charakterizovat třemi nejdůležitějšími vlastnostmi [17]:

- paralelismem: výpočet nového stavu probíhá současně pro všechny buňky ve struktuře,
- lokalitou: nový stav prvku se vypočítá pouze z jeho původního stavu a z hodnot stavů okolních prvků,
- homogenitou: pro všechny buňky struktury platí stejná lokální přechodová funkce.

2.1 Historie celulárních automatů

Počátky vývoje celulárních automatů sahají do 40. let 20. století, kdy americký vědec John von Neumann sestrojil ze svého kinematického modelu [21], který popisoval vývoj od jednodušších struktur po složitější, první celulární - buněčný automat. Byl sestaven asi z 200 000

buněk, z nichž se každá nacházela právě v jednom z 29 existujících stavů. Podle koncepce kinematického modelu bylo pole buněk rozděleno na 3 složky (továrna, duplikátor, počítač) a do dlouhého výrůstku, což byla analogie pásky Turingova stroje. Schopnost sebereplikace automatu je potom řízen procesy na zmíněném výrůstku, ale tímto problémem se ve své práci nezabývám. Podstatná vlastnost von Neumannova automatu byla jeho dynamika, která spočívá v rozložení globální složitosti celé struktury na jednoduché lokální chování dílčích buněk.

Už i von Neumannův automat vykazoval svými vnějšími znaky vztah k biologickým procesům a tento vztah se naplno projevil v jeho zjednodušené verzi, jejíž autor byl Horton Conway. Tento automat byl pojmenován Game of Life [8] (Hra života). Možné stavy automatu byly omezeny na 2, které reprezentovaly existující buňku a uhynulou buňku. Na základě pravidel, která věrně napodobují procesy v přírodě, může buňka vzniknout v ideálním prostředí pro vznik nebo se udržet ve svém stavu. Pokud je buňka opuštěná nebo se nachází v přehuštěném prostředí, pak uhyne. Conway svým automatem udal podnět pro řadu experimentů, které prováděli i další vědci. Tyto experimenty vedly k popsání struktur schopných přenosu informace. Vznikl tak velice zjednodušený komunikační model, který měl velmi úzký vztah k obdobným procesům v biologii.

Díky experimentům s celou řadou dalších automatů došlo i k pokusům o vytvoření reverzibilních automatů, které neztrácejí informaci o předchozím kroku. Je tedy možno v jakémkoliv časovém okamžiku vrátit předchozí stavy všech buněk. V souvislosti s reverzibilitou automatu vznikl první paralelní počítač CAM (Cellular Automata Machine). Vycházel z předchozích experimentů s vytvářením komunikujících entit. Svojí rychlostí předčil tehdejší superpočítač Cray. Jedna z novějších verzí CAM-6 byla dostupná jako přídavná karta do PC. Samotná rychlost výpočtu je jedním z hlavních aspektů, proč je vhodné celulární automat použít jako prostředek pro tuto práci.

2.2 Formální definice CA

Existuje více možností, jak matematicky popsat celulární automaty. Protože se jedná o diskrétní modely jak v prostoru tak v čase, lze k jejich popisu využít množiny, které budou obsahovat konečný nebo spočetný počet prvků. Matematickou definici demonstruji na jednorozměrném automatu, který lze definovat jako sedmici [25]:

$$A = (Q,\ N,\ R,\ z,\ b_1,\ b_2,\ c_0)$$

Q	binární množina stavů
N	používané okolí buňky
z	počet buněk
$b_1,\ b_2$	hraniční hodnoty
c_0	počáteční konfigurace

$$R:\ S \to \delta$$

Množina R je zobrazení z množiny S do množiny δ. Množina S obsahuje jednotlivé buňky v mřížce $S = \{1,\ 2,\ \ldots,\ z\}$. Množina δ je množina lokálních přechodových funkcí $\delta_1,\ \delta_2,\ \ldots,\ \delta_z$. Takto popsané lokální přechodové funkce jsou předpisem, který množině stavů Q přiřazuje právě jeden stav z množiny Q. $\delta_i = Q^N \to Q$.

Příkladem aktuální konfigurace může být automat, jehož okolí bude mít poloměr 2, tedy $N = \{-2,\ -1,\ 0,\ 1,\ 2\}$. Definujeme globální přechodovou funkci, která bere v úvahu

okrajové hodnoty b_1 a b_2, jako zobrazení $G : Q^S \to Q^S$. Každá konfigurace $c \in Q^S$takto přiřazuje všem buňkám v množině S stav z množiny Q. Zápis pravidel je:

$$G(c) = \begin{cases} i = 3, \ldots, z-2 & \delta_i\,(c_{i-2},\, c_{i-1},\, c_i,\, c_{i+1},\, c_{i+2}) \\ i = 1 & \delta_1\,(b_1,\, b_1,\, c_1,\, c_2,\, c_3) \\ i = 2 & \delta_2\,(b_1,\, c_1,\, c_2,\, c_3,\, c_4) \\ i = z & \delta_z\,(c_{z-2},\, c_{z-1},\, c_z,\, b_2,\, b_2) \\ i = z-1 & \delta_{z-1}\,(c_{z-3},\, c_{z-2},\, c_{z-1},\, c_z,\, b_2) \end{cases}$$

Takto definovaná globální funkce obecně popisuje pro všechny buňky $i \in S$ ze stavu z množiny Q do jiného nebo stejného stavu z množiny Q. Okrajové podmínky vstupují do funkce v případě, že počet buněk je konečný. Buňky, které nemají dostatečně velké okolí pro výpočet nového stavu, využijí pro jeho výpočet okrajových podmínek.

2.3 Jednorozměrný celulární automat

Celulární automaty lze dělit do kategorií podle několika kritérií. Nejobecnější rozdělení je podle počtu dimenzí jejich pravidelné mřížky. Nejjednodušší je podle tohoto rozdělení 1D automat, který lze reprezentovat jako řetězec buněk.

Příkladem takového automatu může být Wolframův automat [27]. Pro buňku a její bezprostřední okolí, pokud uvažujeme buňky jako pouze dvoustavové automaty, existuje $2^3 = 8$ různých kombinací hodnot stavů, tedy 8 různých pravidel, jimiž se řídí přechodová funkce v celulárním automatu. Těchto 8 pravidel může celkem nabývat $2^8 = 256$ různých výsledků. Tedy 256 různých celulárních automatů, z nichž každý využívá jinou sadu 8 pravidel. Wolfram kategorizoval tyto automaty čísly od 1 do 256. Okrajové možnosti 1 a 256 nemají smysl uvažovat, protože hned v prvním kroku dojde k nastavení stejných hodnot ve všech buňkách celulárního automatu.

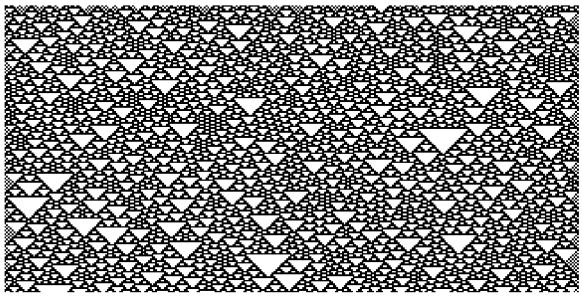
Aktuální stav (se sousedy)	111	110	101	100	011	010	001	000
Nový stav	0	1	1	1	1	0	1	0

Tabulka 2.1 Pravidlo (automat) 122.

Tabulka 2.1 popisuje různé konfigurace buňky s jejím bezprostředním okolím. Na druhém řádku je příklad automatu (pravidlo 122). Binární podoba čísla 122 odpovídá výsledným stavům buňky po jenom kroku automatu.

Na jednorozměrném automatu lze nejjednodušeji popsat okrajové buňky. Pokud je, jako v předešlém případě, poloměr 1, musí mít každá buňka oba dva sousedy. Na obou koncích proto přibude 1 buňka. Stavy těchto buněk jsou konstantní a nejčastěji nulové. V jiném případě lze považovat za okraj automatu jeho začátek. Vznikne tak smyčka, se kterou lze získat výrazně odlišné hodnoty výsledků. Tyto odlišnosti se prohloubí zvětšením poloměru.

Zajímavostí je využití 1D automatu v praxi. Samotný řetězec se v diskrétním čase mění. Pokud ale zaznamenáváme jednotlivé kroky pod sebe, je možné pozorovat zajímavé jevy a obrazce (viz obrázek 2.1).



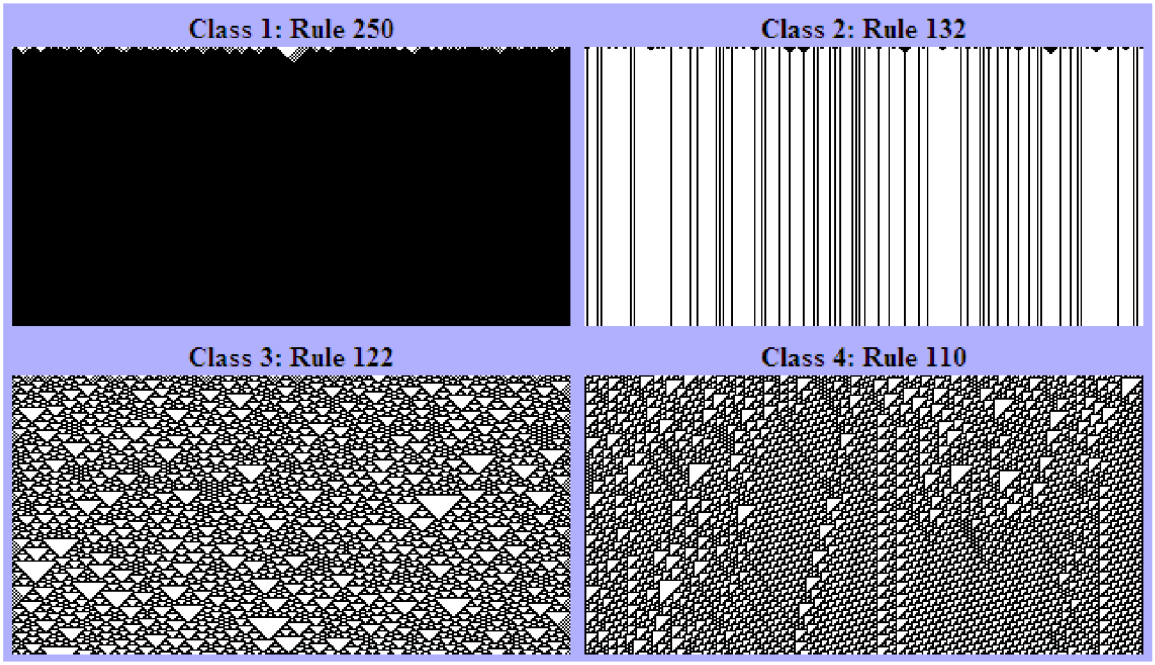
Obrázek 2.1 Průběh pravidla 122 [17].

Na obrázku je vidět průběh již zmíněného pravidla 122. Můžeme pozorovat tzv. fraktálové obrazce, což jsou zdánlivě velmi složité geometrické útvary, které se však při bližším zkoumání skládají z často jednoduchých geometrických primitiv. Podobné obrazce můžeme pozorovat i v přírodě na lasturách některých živočichů či povrchu rostlin.

2.4 Wolframovy třídy

Wolframovy automaty, tedy všech 256 kombinací 8 pravidel pro jednorozměrný automat s poloměrem sousedů velikosti 1, mají různou složitost. Tuto složitost můžeme pozorovat na různých úrovních složitosti fraktálových obrazců při současném zobrazení několika kroků automatu. Na základě těchto zobrazení stanovil Wolfram 4 třídy, do kterých rozdělil 256 automatů. Každé třídě lze přiřknout vhodnost použití pro modelování různě složitých situací. Rozdělení je následující [27]:

- 1. třída: Do této třídy patří nejjednodušší automaty. Nezávisle na počáteční konfiguraci dochází v několika prvních krocích k ustálení jednoho ze stavů 0 nebo 1 ve všech buňkách mřížky.
- 2. třída: Třída zahrnuje automaty, ve kterých dochází k ustálení určité konfigurace. Stabilní shluky tedy nahradí počáteční aktivitu v jednotlivých částech automatu. Na rozdíl od 1. třídy má počáteční konfigurace vliv na pozici ustálených shluků.
- 3. třída: Obrazce automatů, které spadají do této třídy, jsou na první pohled nepravidelné. Zobrazení fraktálových obrazců je chaotické a bez náznaku periodicity.
- 4. třída: Dvourozměrné automaty v této třídě jsou schopny realizovat obecný počítač [17]. Příkladem je Hra života (Game of Life). Je to systém shluků, které jsou schopny vytvářet, přenášet a také přijímat informace v různých tvarech. Obecně lze třídu popsat jako chaotické uspořádání pravidelných struktur.

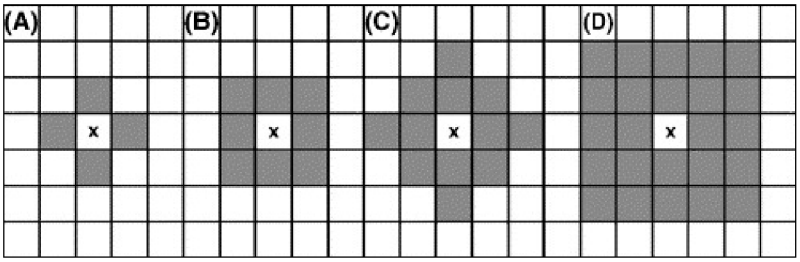


Obrázek 2.2 Příklady automatů ve Wolframových třídách [27].

Na obrázku 2.2 jsou příklady automatů jednotlivých tříd. Každý příklad ve své skupině znázorňuje typické chování automatu své třídy [27].

2.5 Vícerozměrné celulární automaty

Pro řešení některých složitějších problémů je třeba využít více rozměrů než pouze řetězec buněk. Často používané jsou 2D automaty, jejichž znázorněním je čtvercová soustava buněk.



Obrázek 2.3 Příklady okolí u 2D automatu jsou von Neumannovo (A), Moorovo (B) a jejich extended varianty (C, D) [23].

Na obrázku 2.3 jsou znázorněna základní okolí s jednoduchými rozšířeními. Varianty lze podle velikosti poloměru rozšiřovat i dál, čímž roste počet parametrů přechodové funkce.

U okrajových buněk 2D automatů je zajímavá alternativa jejich spojení. Vzniká tak útvar zvaný toroid a tato varianta značně ovlivní chování automatu.

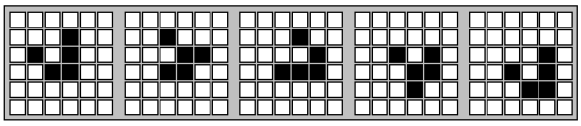
Velmi zásadní vlastností 2D automatů je sebereplikace [3], která byla a je předmětem zkoumání, jehož výsledkem je například Coddův automat nebo Langtonovy Q-smyčky. Automat je schopen vytvářet kopie sebe sama bez vnějšího řízení.

Ze složitějších automatů lze použít i 3D automat k modelování prostorových závislostí. Toto jsou však velmi vzácné a také velmi složité problémy. V těchto automatech lze také využít prostorovou variantu Moorova a von Neumannova okolí. Těmto problémům se v této práci nevěnuji.

2.6 Emergence

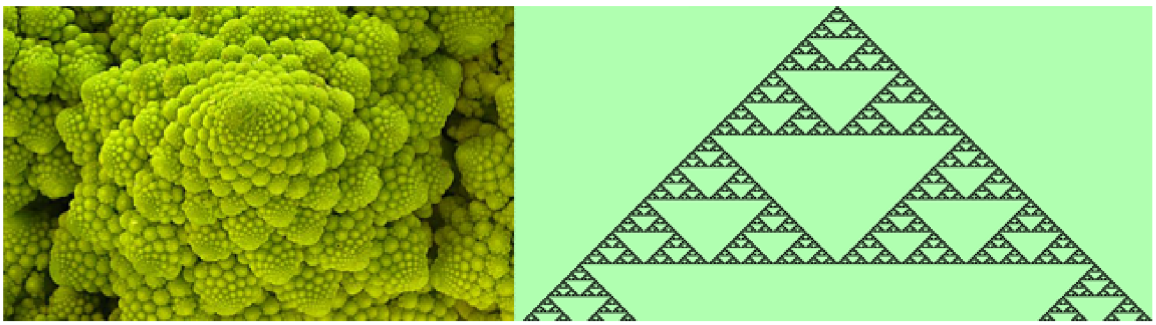
Pokud buňky v systému interagujících mezi sebou, mohou tak vytvářet velmi složité chování systému, které přímo nevychází z jejich vlastního, obvykle velmi jednoduchého chování. Toto chování je popsáno jako emergence [17]. Příkladem emergence mohou být molekuly ve vodě, neurony v mozku, lidé či auta ve městě a také celulární automat, který je schopen na určité úrovni abstrakce předešlé případy modelovat. Důležitou vlastností je fakt, že toto složité chování vyplývá pouze ze vzájemné interakce dílčích entit v systému a není nijak ovlivněno vnějším zásahem.

Automatem, který demonstruje emergentní jevy, je Conwayova Hra života (Game of Life) [8]. Ve dvourozměrné pravidelné mřížce vznikají interakcí na buněčné úrovni složitější struktury, které jsou také schopny interakce. Svojí složitostí spadá do 4. třídy podle Wolframova rozdělení a vykazuje tedy neperiodické, místy chaotické chování. V průběhu vývoje byly objeveny struktury, které jsou schopny pohybu ve dvourozměrné mřížce, shluky, které jsou schopné tyto pohybující struktury vytvářet, a také shluky, které tyto pohybující struktury začleňují do sebe. Jedna ze struktur, které jsou schopny transportu, je na obrázku 2.4. Je na něm znázorněna struktura zvaná „kluzák" a rozdíl oproti první a páté generaci je v umístění v mřížce automatu. Všechno toto chování probíhá bez vnějšího řízení.



Obrázek 2.4 Kluzák v automatu Hra života [8].

Důvodem pro název tohoto automatu (Hra života) je jeho skutečný vztah k reálnému životu. V přírodě najdeme ekosystémy, které se vyvíjejí a vzájemně kooperují bez zásahu řídícího prvku. Příkladem může být hejno ptáků (Reynoldsův model [17]), mraveniště, či tvary fraktálových obrazců u rostlin. Příkladem těchto jevů v jednorozměrných automatech může být Wolframův automat 126. Na obrázku 2.5 je znázorněna podobnost fraktálů Wolframova automatu 126 a fraktálů na rostlině v přírodě.



Obrázek 2.5 Srovnání fraktálů v automatu a v přírodě.

2.7 Shrnutí

Celulární automat je vhodný nástroj pro vytváření simulačního modelu zkoumaných přírodních procesů. V našem případě půjde o řetězec aminokyselin, které budou popsány ve 4. kapitole. Emergence je velmi žádanou vlastností pro studium vzájemně integrujících buněk bez vnějšího zásahu jakýchkoliv řídících příkazů. Stavy jednotlivých buněk jsou řízeny stavovými automaty a nový stav je ovlivněn i buňkami ve svém okolí, což také odpovídá předpokladům pro použití. Výhodou je také paralelní zpracování buněk automatu při přechodu mezi stavy, které úspěšně modeluje přírodní procesy.

Kapitola 3

Evoluční algoritmy

Evoluční výpočetní techniky jsou přírodou a přírodními vědami inspirované algoritmy, které se snaží svými postupy napodobit evoluci v přírodě. Vychází z darwinovského principu, který je založen na hledání nejoptimálnějšího či dostatečně vyhovujícího řešení problému. Tyto algoritmy jsou v prostředí jejich využití funkční, tedy tak nepřímo dokazují platnost teorie, z níž vycházejí, nicméně Darwinova myšlenka má z filozofického, morálního či jiného hlediska mnoho odpůrců. Tato práce se však nevěnuje ideologickému základu, ale praktickému využití evoluce pro nalezení vyhovujícího řešení.

Počátky pokusů napodobit při řešení problémů procesy v přírodě sahají do 50. let 20. století, kdy však tehdejší stroje neměly dostatečnou výpočetní sílu ani rychlost, a proto vývoj evolučními procesy inspirovaných paradigmat řešení problémů nebyl nijak výrazný. Dalším důvodem byl nedostatečný metodologický přístup [6]. Koncem 20. století se však evoluční techniky, což jsou stochastické optimalizační metody vycházející z darwinovského principu evoluce, dostaly do popředí zájmu vědecké společnosti i praktických specialistů v širokém spektru oborů. Za základní prvek lze považovat genetický algoritmus, který bude pro účely nalezení správných parametrů v této práci využit. Z hlediska terminologie nejsou všechny pojmy zcela vymezeny a některé techniky svým charakterem přesahují rozsah chápání genetického algoritmu.

3.1 Darwinova evoluční teorie

Základem pro evoluční teorii je kniha Charlese Darwina z roku 1859 s názvem O vzniku druhů přírodním výběrem [5]. V 19. století, které paradoxně nebylo podobným revolučním názorům nakloněno, se teorii přírodního výběru věnovalo více vědců. Ve spolupráci s nimi se Darwinovi podařilo zjistit, že fosílie, které přivezl ze svých objevitelských plaveb za oceán, pochází ve většině případů z vyhynulých živočichů a rostlin. Ze své cesty Darwin také přivezl žijící exempláře flóry a fauny. Podle Darwina se populace rostlin a živočichů vyvíjela po mnoho generací podle principu přirozeného výběru a přežití těch nejschopnějších a nejpřizpůsobivějších jedinců. Své pozorování a studium publikoval Darwin ve svém již zmíněném díle O vzniku druhů přírodním výběrem, které se na dlouho stalo předmětem sporů a diskuzí, zda lze jeho obsahu věřit či nikoliv.

3.2 Historie evolučních výpočetních technik

Počátky metod inspirovaných přírodními procesy sahají do 50. a první poloviny 60. let. Na univerzitě v Berlíně bylo využito kombinování dvojic různých typů konstrukcí převodovek za účelem zisku optimálního řešení s lepšími užitnými vlastnostmi [13]. Odpovídajícím přírodním jevem je křížení v rámci biologického druhu. Na této myšlence založené a již vědomě použité metody se objevily koncem 60. let jako evoluční programování. Jako evoluční prostředek byl použit konečný automat se schopností adaptace [12]. Jednalo se o vývoj technologie predikce změn prostředí, ve kterém se automat nachází. Další metody a postupy byly vyvíjeny na základě teoretických biologických publikací. Nejslavnější kniha Johna Hollanda [12] pojednává o příčině velkých odlišností mezi jedinci téhož biologického druhu a stala se teoretickým základem pro evoluční algoritmy.

Jednotlivé techniky se vyvíjely většinou nezávisle na sobě, a proto byly dílčí postupy v některých krocích odlišné. Jejich společný a obecný tvar publikoval ve své knize David Goldberg v roce 1989 [10]. Genetické algoritmy, jak byly tyto postupy pojmenovány, jsou zde popsány z technického pohledu a způsobem, který lze aplikovat na širokou škálu různých úloh.

Výraznější modifikace genetických algoritmů se objevily až koncem 90. let 20. století. Jak již bylo zmíněno, tyto nové možnosti potřebovaly rychlejší výpočetní stroje a sofistikovanější metodologický přístup, který do té doby nebyl dostatečný. Těmito modifikacemi se zabýval Zbigniew Michalewicz [19].

Za zakladatele genetického programování je považován John Koza [15]. Genetické programování je vyvrcholením užití genetický algoritmů, na které navazuje, a cílem je pomocí evoluční automatizace generovat počítačové programy. Tyto programy jsou schopny řídit konkrétní úlohy či jejich zobecnění v určité skupině.

3.3 Genetický algoritmus

Genetický algoritmus je řídící struktura, která za účelem dosažení optimálního nebo suboptimálního řešení využívá evolučního principu procesů známých z přírody. Řešení probíhá formou soutěže postupně se vyvíjejících jedinců v populaci. Důležitou vlastností genetického algoritmu je vhodné vyhodnocení této soutěže, aby se zjistilo, který jedinec v rámci populace má lepší předpoklady pro další vývoj a tak i pro hledání lepšího řešení. Ohodnocení jedince se nazývá fitness. Jedinec s vyšším fitness nese lepší vlastnosti populace, které zůstávají zachovány v průběhu vývoje. Tímto vývojem se stanoví lokální optima, což jsou dílčí úspěchy v různých generacích populace. Když dojde k ukončení genetického algoritmu, z celé populace je vybráno výsledné řešení, které se prohlásí za optimální, pokud jsou výsledky uspokojivé. Může však dojít k degeneraci, kdy nejlepší jedinec bude obsahovat pouze lokální optimum, které se liší od optima globálního. Abychom dokázali určit výsledek, uchovávají se statistiky během vývoje populace zejména o nejlepších, nejhorších a průměrných hodnotách.

Výpočetní síla genetického algoritmu je v jeho robustnosti a univerzálnosti. Na rozdíl od klasických optimalizačních metod využívá celé populace vyvíjejících se řešení namísto jediného počátečního řešení, což vede k paralelizaci celého výpočtu.

Genetický algoritmus je algoritmem slepým. Znamená to, že kromě samotného ohodnocení jedinců v populaci funguje zcela bez znalosti konkrétního problému, který řeší. Díky této vlastnosti je velmi univerzální a má široké spektrum použití. Pokud se však pro řešení konkrétního problému navrhne specializovaná heuristika, dosáhne mnohdy lepších výsledků

než genetický algoritmus. Nevýhodou specializované heuristiky je ale její omezenost na konkrétní úlohu. Naproti tomu genetický algoritmus lze využít pro téměř jakoukoliv úlohu. V této práci byl použit jako metoda pro nalezení optimálních parametrů pro přechodovou funkci celulárního automatu.

Rozšířením genetických algoritmů je již zmíněné genetické programování. Výsledkem genetického programování je hierarchicky strukturovaný program, který (přibližně) řeší daný problém. Jedná se o jakési pěstování počítačových programů. Úspěchy genetického programování jsou například znovuobjevení i vylepšení různých patentů [16].

3.4 Reprezentace jedinců

Každá populace se skládá z individuálních jedinců. Reprezentace těchto jedinců se nazývá genotyp nebo též chromozom. V biologii je genotypem pojmenován soubor chromozomů, ale pro účely této práce a obecně pro genetické algoritmy platí, že pro reprezentaci jedince stačí pouze jeden chromozom. Záměna těchto dvou pojmů je možná pouze v oblasti genetických algoritmů. Jednotlivé chromozomy lze dále dělit na geny, které jsou uspořádány lineárně. Pokud tedy dva chromozomy stejného typu obsahují gen pro určitou vlastnost, nachází se jeho konkrétní forma na stejných místech v odpovídajících genech. Konkrétní forma genu se nazývá alela a jedná se v zásadě o konečný stavový automat, jehož stav udává formu genu. Popis jednotlivých částí a terminologie odpovídá biologickým podkladům, kterými se blíže zabývá genové inženýrství.

Všechny chromozomy mají pevně stanovenou délku, protože pocházejí ze stejné populace. Toto pravidlo vychází z faktu, že u každého člověka je stejný počet 23 párů chromozomů v každé buňce jeho těla. Způsob zakódování chromozomů, tedy souboru jednotlivých genů má zásadní vliv na úspěšnost genetického algoritmu. Nevhodným způsobem reprezentace se lze připravit o důležitá data. Historicky nejstarším způsobem kódování je binární kód [12]. Jeho hlavní výhodou je jednoduchost, proto je stále jedním z nejpoužívanějších reprezentací [20]. Znamená to, že stavový automat genu nabývá pouze dvou stavů: 0 a 1.

Přesnost zakódování informace do binárního kódu udává následující příklad, kdy je cílem nalézt minimum funkce $f(x)$ na intervalu $\langle r, s\rangle$. Požadujeme-li přesnost ϵ, čili absolutní přesnost je lepší než ϵ, pak délka binárně reprezentovaného chromozomu se určí takto:

$$L = \left\lceil \log_2 \frac{|s-r|}{\epsilon} \right\rceil,$$

kde proměnná x ve funkci $f(x)$ je pro přesnost $\epsilon = 10^{-6}$ na intervalu $\langle -2, 2\rangle$ zakódována do 22 bitů podle následujícího vzorce:

$$x = r + x' \frac{|s-r|}{2^{22}-1},$$

kde proměnná x' je dekadická hodnota binárního čísla.

3.5 Pseudokód genetického algoritmu

Genetické algoritmy se mezi sebou liší obvykle jen tím, jak se vytváří nová populace z té předchozí. To, co však mají společné, se dá popsat následujícím pseudokódem [13]:

1. `Vynulování hodnoty počítadla generací na` $t = 0$

2. `Inicializace počáteční populace` $P(0)$
3. `Každé individuum v` $P(0)$ `musí mít počáteční ohodnocení (fitness)`
4. `Rozdělení individuí do dvojic, ze kterých vznikne populace potomků` $P'(t)$
5. `Nově vzniklá individua získají ohodnocení`
6. `Nová populace vznikne sloučením` $P(t+1) = P'(t)$
7. `Inkrementace počítadla generací` $t = t + 1$
8. `Ohodnotí se nově vzniklá generace` $P(t+1)$
9. `Pokud je splněna ukončující podmínka, pak výpočet končí`
10. `Pokračování krokem 4`

Ukončovacích podmínek může být více druhů. V obecném případě pokračuje bez vnějšího zásahu evoluce nekonečnou dobu a zabere nekonečně velký prostor. Jedna z podmínek ukončení spočívá v omezení počtu generací. Další možností ukončení výpočtu je kontrola dosažených výsledků. Jakmile dosáhne ohodnocení individuí určité hranice, výpočet se ukončí.

Samotný algoritmus potom začíná inicializací počáteční populace. Při náhodném generování dojde k vytvoření množiny řetězců stejné délky. Samotné ohodnocení (fitness) nelze udělat bez znalosti konkrétního problému, na jehož řešení je genetický algoritmus použit. Dalším krokem algoritmu je výběr vhodných dvojic pro reprodukci a tento výběr by měl být přirozený podle darwinovského principu. Zdatnější jedinci jsou v tomto případě ti, kteří mají z hlediska řešení problému nejlepší vlastnosti. Nejpoužívanější metodou výběru je ruletový mechanismus [13], který se od klasické rulety liší v nestejně velkých výsečích kola. Každá výseč odpovídá jednomu jedinci. Princip vytvoření tohoto kola spočívá ve výpočtu ohodnocení p_i podle následujícího vzorce:

$$p_i = \frac{f_i}{\sum f_i} \, i \in \{1, \ldots, N\},$$

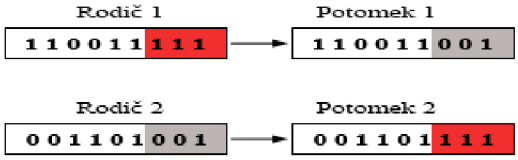
kde proměnná N značí velikost populace a f_i je vlastní ohodnocení jedince. Náhodně vygenerujeme číslo $r \in \langle 0, 1 \rangle$ a i-tý jedinec je vybrán, právě když platí:

$$\sum_{j=1}^{i-1} p_j < r \leq \sum_{j=1}^{i} p_j \; i \in \{1, \ldots, N\}.$$

Po výběru dvojic jedinců dochází k vytvoření potomků, ke kterému jsou využívány techniky genetických operátorů, kterým je věnována následující kapitola. Použijeme-li tzv. generační strategii, nová generace přepíše generaci původní. Původní generace v tomto případě zaniká a nahrazuje ji nová. Tím končí jeden cyklus generačního algoritmu. Dojde ke kontrole ukončujících podmínek a v případě jejich splnění se vývoj ukončí.

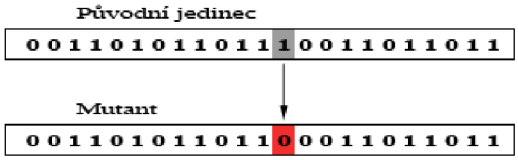
3.6 Genetické operátory

Existují dva základní genetické operátory - křížení a mutace. Při křížení je na výběr mnoho možností, jak dva chromozomy kombinovat. Společnou vlastností všech technik je výměna genetického materiálu. Nově vzniklý jedinec dědí vlastnosti po obou rodičích, tedy vzniká kombinací rodičovských chromozomů a výměnou jejich částí. Nejjednodušší možnost křížení se nazývá jednobodové křížení [13]. Spočívá v určení bodu v chromozomu, od kterého začíná část určená ke křížení. Tato část může mít rozdílnou velikost v každém procesu křížení. Jedna z existujících strategií také připouští zachování původních jedinců bez použití genetického operátoru. Znázornění křížení je na obrázku 3.1.



Obrázek 3.1 Operace křížení.

Druhým operátorem je operátor mutace. Proces mutace je aplikován na jedince vzniklé křížením. V přírodě pozorujeme ekvivalentní proces pod týmž názvem, kdy při křížení dvou jedinců může dojít k poškození vznikajícího genu. Použití mutace jako genetického operátoru se provádí za účelem vylepšení vlastností jedince. Dalším cílem mutace je zabránění příliš rychlé konvergenci populace.



Obrázek 3.2 Operace mutace.

Existují i jiné genetické operátory, ale v této práci nejsou použity. K použití operátorů typicky nedochází ve všech případech. Typicky probíhá křížení v 75 - 95% [24] případů. Proces křížení převládá nad procesem mutace, ke kterému dochází jen velmi zřídka (podle [24] 0,1 - 5%). Někdy může zdegenerovat celá populace a výsledek může být zahozen. Příklady udávají nejjednodušší způsob výměny genetického materiálu, ovšem přesně daný standard pro genetické operátory neexistuje. Algoritmus pro vytvoření nového jedince spočívá v kombinaci obou jeho rodičů a zcela zásadní roli hraje vhodné zakódování informace do chromozomu.

3.7 Shrnutí

Evoluční metody jsou metody stochastické, které se nacházejí mezi heuristickými a úplnými metodami umělé inteligence. Genetický algoritmus využívá celou populaci jedinců, tedy populaci možných řešení problému, na který je aplikován. Jeho schopnost tato řešení kombinovat a vytvářet nové jedince v rámci populace je vhodnou vlastností při hledání optimálních parametrů pro nalezení optimálního řešení problému. Vytváření nového jedince probíhá s využitím genetických operátorů. Takto lze nalézt řešení i v místě, ve kterém nejsou očekávána, protože občas přežívají i méně zdatní jedinci a mutací je tento postup zpomalen. Výsledku je dosaženo pomocí soutěže různě zdatných jedinců, kdy po několika generacích můžeme nalézt optimální řešení. Během vývoje jedinců je vhodné zaznamenávat průběžné statistiky maxima, minima a průměrného výsledku.

Kapitola 4

Proteiny a predikce sekundární struktury

Bílkoviny neboli proteiny vytvářejí látkový základ všech živých organismů. V těle vyšších organismů a člověka plní proteiny mnoho funkcí, z nichž nejvýznamnější je funkce stavební. Podíl proteinů v tkáních je až 80%. U rostlin je zastoupení proteinů nižší a rozdíl doplňují polysacharidy. Z hlediska bioinformatiky jsou proteiny předmětem zájmu hlavně z důvodu zkoumání jejich biologických funkcí, složení a jejich struktur. Proteiny jsou řetězce složené z aminokyselin, což jsou molekuly, které se spojují peptidickou vazbou a vytvářejí tak řetězce proteinů. Proteinové struktury jsou rozděleny podle složitosti od primární po kvartérní. Na základě různého uspořádání a tvarů těchto struktur vykazují proteiny určitý typ chování a vnější vlastnosti. Cílem této práce je predikce sekundární struktury na základě znalosti posloupnosti aminokyselin, což slouží ke stanovení vlastností a chování zkoumaného proteinu. Samotná sekundární struktura, která je popsána v kapitole 4.2, není sice pro toto stanovení dostatečná, ale je prvním krokem, který značně usnadní následující upřesnění. Na základě podobnosti proteinových struktur vznikají tzv. proteinové rodiny [22], což jsou skupiny proteinů, nesoucí stejné nebo podobné vlastnosti.

Jak již bylo zmíněno v úvodní kapitole, zkoumání makromolekulárních řetězců je obecně jedním z nejdůležitějších úkolů bioinformatiky. Pro určování a predikci struktur existuje mnoho různých nástrojů a variant výpočtu, které budou popsány v kapitole 4.3. V této práci porovnávám mnou dosažené výsledky s výsledky těchto metod. Cílem je co největší urychlení výpočtu, a to i za cenu menší přesnosti oproti sofistikovanějším, avšak značně pomalejším metodám.

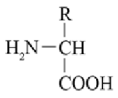
4.1 Proteiny

Proteiny patří k nejdůležitějším látkám v lidském těle. Z biochemického hlediska patří mezi biopolymery, což znamená, že jsou to řetězce molekul (polymery) a zároveň přírodní látky (bio). V těle se proteiny štěpí a přetváří se na jiné, které jsou z hlediska jejich funkce potřeba. Proteiny tak rozdělujeme do skupin podle biologických funkcí:

- stavební (kolagen),
- katalyzační (enzymy),
- regulační (hormony),

- transportní (transport látek v těle - hemoglobin, transferin),
- obranná (různé protilátky),
- pohybová, zásobní, signální, receptorová, regulace a exprese genů.

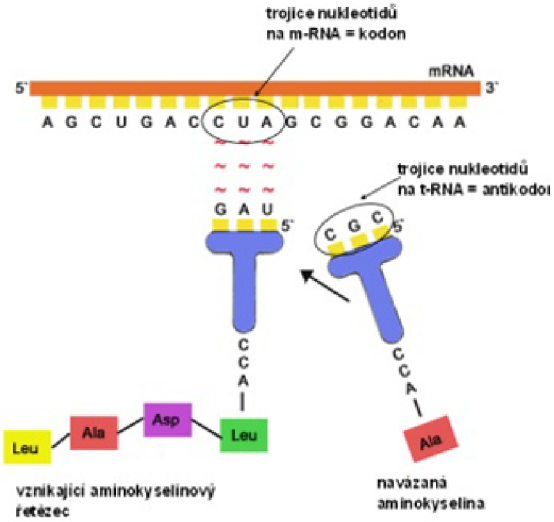
Tyto funkce a jiné vlastnosti jsou přímo ovlivněny složením proteinového řetězce. Formální definicí z pohledu teoretické informatiky lze proteiny popsat jako řetězce nad abecedou aminokyselin. Aminokyseliny jsou obecně molekuly, které se skládají z určitých jednodušších chemických molekul. Na obrázku 4.1 je obecná chemická struktura aminokyselin.



Obrázek 4.1 Obecná chemická struktura AK.

Aminokyseliny se skládají z karboxylové skupiny ($-COOH$), aminoskupiny (H_2N-), alfa-uhlíku uprostřed a postranního řetězce R, kterým se od sebe odlišuje 20 známých aminokyselin [1].

Proces vzniku proteinů se nazývá proteosyntéza a probíhá uvnitř buněk. Pro jednoduchost předpokládáme lineární vlákno mRNA, které je složeno z tzv. nukleotidů. Nukleotidy se mezi sebou liší jednou ze čtyř dusíkatých bází. Báze se označují počátečními písmeny svých názvů (Adenin, Guanin, Cytosin, Uracil). Každé tři po sobě následující nukleotidy tvoří tzv. triplet (kodón). Aminokyseliny jsou trasportní strukturou tRNA dopraveny k vláknu mRNA, kde na principu komplementarity na sebe nasedají kodóny mRNA a antikodóny tRNA a podle obrázku 4.2 vzniká vlákno aminokyselin. Proces se nazývá translace [22].



	U		C		A		G	
U	UUU	fenylalanin	UCU	serin	UAU	tyrosin	UGU	cystein
	UUC	fenylalanin	UCC	serin	UAC	tyrosin	UGC	cystein
	UUA	leucin	UCA	serin	UAA	**stop**	UGA	**stop**
	UUG	leucin	UCG	serin	UAG	**stop**	UGG	tryptofan
C	CUU	leucin	CCU	prolin	CAU	histidin	CGU	arginin
	CUC	leucin	CCC	prolin	CAC	histidin	CGC	arginin
	CUA	leucin	CCA	prolin	CAA	glutamin	CGA	arginin
	CUG	leucin	CCG	prolin	CAG	glutamin	CGG	arginin
A	AUU	izoleucin	ACU	treonin	AAU	asparagin	AGU	serin
	AUC	izoleucin	ACC	treonin	AAC	asparagin	AGC	serin
	AUA	izoleucin	ACA	treonin	AAA	lysin	AGA	arginin
	AUG	**metionin**	ACG	treonin	AAG	lysin	AGG	arginin
G	GUU	valin	GCU	alanin	GAU	kys.	GGU	glycin
	GUC	valin	GCC	alanin	GAC	asparagová	GGC	glycin
	GUA	valin	GCA	alanin	GAA	kys.	GGA	glycin
	GUG	valin	GCG	alanin	GAG	glutamová	GGG	glycin

Obrázek 4.2 Translace v buňce a tabulka AK kódovaných kodóny mRNA [22].

Každá aminokyselina je kódována trojicí vybranou ze 4 možných nukleotidů, čili celkový počet možných kombinací je $4^3 = 64$. Některým aminokyselinám však odpovídá více kódů.

Tabulka AK[1] je na obrázku 4.2, odkud je zřejmé, že například leucin je kódován 6 různými kodóny. Vznikající aminokyselinový řetězec není v žádném případě náhodný jev. Posloupnost kodónů je předpisem pro vznik konkrétního proteinu, jehož AK jsou propojeny tzv. peptidickou vazbou. Tato vazba je výsledkem reakce aminoskupiny s karboxylovou skupinou sousední AK, kde vedlejším produktem je voda (H_2O). Tím je dovršen proces translace a vzniká protein. Jelikož jsou proteiny přírodní látky, liší se od polymerů vzniklých chemickou cestou přesně danou strukturou, která je hlavním předmětem zkoumání v této práci.

AK lze rozdělit takto:

- hydrofóbní AK (jsou nesolubilní, proto se v proteinech vyskytujících se ve vodě nachází zpravidla uvnitř molekuly),
- polární AK (jsou solubilní, proto se v proteinech vyskytujících se ve vodě nachází zpravidla na povrchu molekuly).

Z rozdělení vyplývá, že některé AK podporují vznik vazeb s okolními molekulami vody více než jiné.

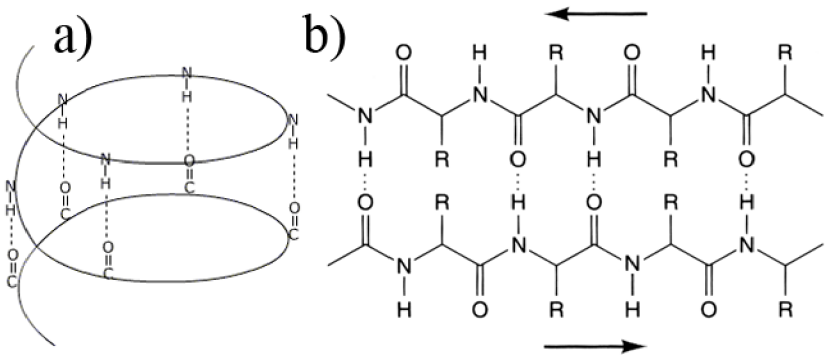
4.2 Proteinové struktury

U proteinů existují 4 popsané struktury [1]:

- primární (posloupnost aminokyselin),
- sekundární (geometrické uspořádání polypeptidického[2] řetězce[1]),
- terciární (uspořádání sekundárních struktur v prostoru),
- kvartérní (vzájemné prostorové uspořádání více polypeptidických řetězců).

Primární struktura je tvořena sekvencí nukleotidů na vlákně mRNA, které je předlohou pro vznikající protein. Tato struktura podmiňuje biologické funkce a chemické a fyzikální vlastnosti proteinů. Na základě znalostí těchto vlastností se dá přibližně určit výskyt elementů sekundární struktury. O těchto technikách je více psáno v kapitole 4.3.

V minulosti bylo zjištěno několik geometrických útvarů, které lze na proteinovém řetězci nalézt a které se považují za sekundární struktury.



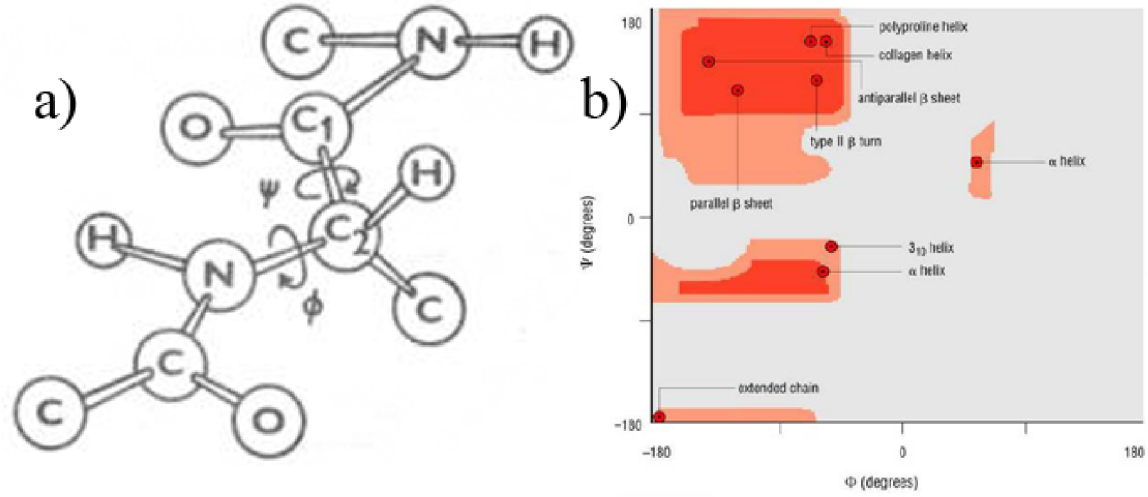
Obrázek 4.3 Struktury α-helix (a) a β-sheet (b).

[1] AK je běžně užívaná zkratka pro aminokyseliny

[2] peptid je protein čítající obvykle cca 100 AK, polypeptidický řetězec je protein čítající stovky AK) [22]

Prvním elementárním útvarem je α-helix [1], který je znázorněn na obrázku 4.3a. Jedná se o pravotočivou šroubovici, kde na jednu otáčku této struktury je třeba 3,6 AK. Jeho postranní řetězce jsou natočeny směrem ven. Tento útvar většinou nebývá dlouhý. Struktura α-helix se velmi často nachází v tzv. membránových[3] („amfipatických") proteinech [1]. Tyto proteiny zpravidla prochází skrz membránu buněk a svými hydrofóbními AK dotváří její povrch. Na vnitřní straně membrány jsou naopak hydrofilní AK, což vede k vytváření polopropustné membrány.

Dalším možným útvarem je β-sheet [1] neboli tvar skládaného listu, který je znázorněn na obrázku 4.3b. Tento útvar může být tvořen dvěma či více paralelními či antiparalelními vlákny. Antiparalelní vlákna mají vzájemně opačný směr, paralelní vlákna mají směr stejný. Speciálním případem antiparalelismu je jedno vlákno, které se přes smyčku vrací podél svého původního vinutí. Spojující smyčka je v tomto případě popsána jako β-turn a v buňkách většinou tvoří jejich povrch. Spolu s dalšími možnými strukturami smyček spadá do skupiny, která je popsána jako „coil". Tato práce ji používá jako strukturu pro popis všech jiných útvarů sekundární struktury, než jsou α-helix a β-sheet.



Obrázek 4.4 Torzní úhly otočení vazby (a) a Ramachandranova mapa [1] (b).

Již bylo zmíněno, že uspořádání elementů sekundární struktury závisí na struktuře primární, která ustanovuje fyzikální a chemické vlastnosti proteinu. Na úrovni chemie jsou to vazby tzv. vodíkových můstků, což jsou relativně slabé vazebné interakce mezi vodíkem a silně elektronegativním prvkem, jakým je v tomto případě kyslík. Tato vodíková vazba spočívá v částečném odčerpání elektronové hmoty z kyslíku směrem k vodíku. Vznik této vazby je důkazem velmi důležitých interakcí mezi jednotlivými AK v řetězci proteinu, které jsou v této práci modelovány jako interakce buněk v celulárním automatu. Vzájemné interakce vodíkovými můstky jsou naznačeny na obrázku 4.3.

Z hlediska fyzikálních vlastností se jedná zejména o hodnoty tzv. torzních úhlů. V řetězci jsou popsány pro každou AK tři torzní úhly ϕ, ψ a ω. Poslední ω se však neuvažuje, protože je tzv. planární, což znamená, že má úhel 0° nebo 180°. Oba zbývající úhly ϕ a ψ jsou na obrázku 4.4 znázorněny jako úhly, které udávají, o kolik stupňů je jedna AK otočena vzhledem ke druhé ve vazbě mezi nimi. V širším úhlu pohledu jde o úhel, který mezi sebou svírají dvě roviny v prostoru. Vedle naznačených úhlů se nachází Ramachandranova

[3]membránové proteiny tvoří nejčastěji povrch buněk, ale oddělují i její vnitřní části

mapa [1], která udává sekundární strukturu pro konkrétní kombinaci úhlů ϕ a ψ. I v tomto případě lze pozorovat vzájemné interakce sousedních AK, protože velikosti úhlů závisí na jejich posloupnosti.

Terciární struktura udává uspořádání sekundárních struktur v prostoru. Výsledné rozložení může být fibrilární (tvar vlákna) nebo globulární (tvar klubka) [1]. Toto rozložení se může vlivem vnějších podmínek měnit. Příčinou těchto změn jsou i vlastnosti AK, které například reagují na vodu či elektrický náboj. Z hlediska procesu uvnitř vlákna se jedná o vratné i nevratné změny, ale těmto strukturám a kvartérní struktuře se tato práce přímo nevěnuje.

4.3 Predikce sekundární struktury

Z předchozí kapitoly vyplývá jasná souvislost mezi jednotlivými strukturami, které se vzájemně ovlivňují od nejjednodušší směrem ke složitějším. Stanovení vyšších struktur je důležité z hlediska zjištění biologických funkcí a vlastností proteinů a pro následné zařazení do různých proteinových rodin. Tyto rodiny, jak již bylo zmíněno, jsou skupiny proteinů, které sdružují jejich typické vlastnosti a z nich vyplývající funkce. Velký význam mají tyto znalosti například pro genetické inženýrství, které se zabývá genovou expresí a regulací. Výsledkem pak je zařazení konkrétního proteinu podle vlastnosti, který gen je spouštěn či regulován a jakými znaky se projeví [1].

Významným přínosem nástrojů predikce jsou nízké náklady. Experimentální zkoumání složitějších proteinových struktur je velmi drahé. Navíc jsou používané metody náročné na zkoumaný materiál. Aby bylo možné zjistit přímo sekundární strukturu, je třeba mít krystal proteinu, jehož získání je časově náročné a drahé.

Predikční metody, které se tímto problémem zabývají, lze rozdělit takto:

- statistické,
- knowledge-based,
- metody založené na strojovém učení,
- konsenzuální.

Statistické metody jsou založeny na předpokladu, že konkrétní aminokyselina se bude nacházet v té sekundární struktuře, pro kterou predikční metoda spočítala nejvyšší pravděpodobnost. Tato směrodatná pravděpodobnost výskytu ve struktuře je zjištěna ze znalosti sekundárních struktur velkého množství proteinů. Výhodou těchto metod je rychlost, protože nepoužívají žádné sofistikované výpočty. Jejich nevýhodou je velká nepřesnost. Samotná statistika upřednostňující strukturu, která je pro danou AK nejběžnější, však neobsahuje dostatek informací pro úspěšnou predikci. Příkladem je metoda Chou-Fasman, která využívá relativní frekvenci f výskytu AK ve struktuře α-helix, β-sheet a coil a následný tzv. parametr konformační preference [1] podle vzorců:

$f_i^h = \frac{n_i^h}{n^h}$,

kde n_i^h je počet výskytu dané i AK ve struktuře h, n^h je počet všech AK ve struktuře h

$P_i^h = \frac{f_i^h}{\langle f_i^h \rangle}$,

kde f_i^h je relativní frekvence výskytu i AK ve struktuře h, $\langle f_i^h \rangle$ je průměrná relativní frekvence výskytu všech AK ve struktuře h.

Obdobně se určí parametry konformační preference pro všechny zkoumané struktury. Tyto parametry udávají svojí hodnotou schopnost prodloužit nebo přerušit v daném místě strukturu a jsou tak využity pro další výpočet. Hodnoty těchto parametrů jsou užity jako inicializační parametry pro výpočet struktury v této práci. Tato metoda má úspěšnost průměrně 50–60 %. Mezi nejznámější statistické metody patří metoda GOR [9]. Metoda GOR využívá teorii informace, kdy se zkoumá pravděpodobnost výskytu elementu sekundární struktury, jestliže je na konkrétním místě daná AK. Nevýhodou je nutnost shromáždit dostatečně velkou trénovací množinu a absence vlivu sousedů. Nová verze metody GOR III sice počítá s vlivem sousedů na výsledek, ale stále zůstává nutnost trénovacích množin.

Další skupina metod využívá dodatečné znalosti o AK. Jedná se zejména o fyzikální a chemické vlastnosti a evoluční informace. Patří sem následující metody ZPRED [11], PREDATOR [7] nebo metoda nejbližších sousedů [14]. Jejich nevýhodou může být absence dodatečných informací a některé z nich neberou v úvahu interakce mezi AK.

Nejpřesnější metody jsou založené na strojovém učení. Jejich nevýhodou je opět potřeba velké trénovací množiny, což značně zpomaluje a zesložiťuje výpočet. Využívá se technik neuronových sítí nebo skrytých markovských modelů. Výhodou je jejich úspěšnost, která u neuronových sítí dosahuje až 80%. Zástupci jsou PSIPRED [18], PHD [14] nebo NNSSP [14].

Do poslední skupiny se řadí například JPRED [4]. Metody v této skupině predikují strukturu pomocí více metod a za správný je považován ten výsledek, který je predikován nejčastěji. Výhody i nevýhody tak přebírají z metod, které k výpočtu využijí.

Pro hodnocení přesnosti těchto metod slouží dvě techniky. V této práci použitá metoda Q_3 je standardem pro hodnocení přesnosti. Méně častá je metoda SOV, která klade důraz na správný počet, typ a pořadí elementů sekundární struktury. Metoda Q_3 určuje poměr správně predikovaných AK a celkového počtu AK.

4.4 Shrnutí

Existuje mnoho metod a přístupů pro predikci sekundární struktury proteinů. Predikce této struktury ušetří finanční i časové prostředky a je mezikrokem pro stanovení vyšších struktur, které mají význam pro lékařské, biologické, chemické či jiné vědecké disciplíny. Zároveň lze ze znalosti struktur zařadit protein do určité proteinové rodiny, jejíž zástupci vykazují stejné vlastnosti a biologické funkce.

Kapitola 5

Implementace

Cílem praktické části této práce byl návrh výpočetního frameworku pro predikci sekundární struktury proteinů. Jako teoretický model byl použit celulární automat, který svými klíčovými vlastnostmi, jimiž jsou paralelismus, lokalita změn a homogenita, vyhovuje jako vhodný nástroj pro simulaci vzájemných interakcí AK v proteinech. Vznik sekundární struktury v proteinu nemá vnější centrální řízení. Obdobné chování lze pozorovat i v celulárním automatu. Tento jev se nazývá emergence [17].

Pro nalezení optimálních parametrů pro běh celulárního automatu byl využit genetický algoritmus. Z původní množiny možných kombinací parametrů vznikly postupně další varianty kombinací. Tyto nové kombinace vznikly za použití genetických operátorů křížení a mutace a jejich vložením do výpočetního frameworku bylo dosaženo lepších výsledků než při použití parametrů z předchozích množin.

Aby bylo možné prohlásit tyto parametry za optimální, bylo nutné použít pro jejich otestování více proteinových datasetů. Na výběru datových sad s proteinovými řetězci velmi záleží, protože je nutné, aby množina obsahovala řetězce různých délek s různými sekundárními strukturami. Čím větší odlišnosti jsou mezi řetězci, tím je výsledek důvěryhodnější. Pro každý dataset platí, že při porovnání řetězců způsobem každý s každým nesmí podobnost zkoumaných dvojic přesáhnout 25 % [26].

Výsledný program se jmenuje CELLPRED. Jedná se o konzolovou aplikaci, napsanou v objektově orientovaném jazyce C++. Program je multiplatformní, tedy plně přenositelný.

5.1 Trénovací datasety

Pro testování úspěšnosti predikce sekundární struktury byly zvoleny dva datasety. Dataset je soubor řetězců AK, u kterých byla experimentální metodou DSSP zjištěna jejich sekundární struktura. Každý záznam v mnou vybraných datasetech obsahuje aminokyselinový řetězec, experimentálně určenou sekundární strukturu (DSSP) [26] a další pomocné a zjištěné údaje, mezi které patří i predikované sekundární struktury pomocí různých metod. Obecně tyto množiny řetězců obsahují větší počet vzájemně se odlišujících řetězců. Důležitými aspekty jsou dostatečná průměrná délka řetězců a vhodné zastoupení jednotlivých elementů sekundární struktury.

V této práci byly využity dva datasety podle vzoru [2]. Prvním z nich je dataset RS126 s počtem 126 řetězců a druhým je dataset CB513 s 513 řetězci. Příklad údajů o jednom řetězci je následující:

```
OrigSeq:TTCCPSIVARSNFNVCRLPGTPEAICATYTGCIIIPGATCPGDYAN
cons:-------------EEEE---------------EEE-----------
dsc:-----------HHHHHH---------------EEE-----------
mul:---E----------EEE-------EEE----EEEE-----------
nnssp:-------------------------------EEE-----------
phd:-------------EEEE------EEEE----EEEEE----------
pred:------------------------------EEE-----------
zpred:--EEE-------EEEEEE------EEE----EEEE-----------
access:EEBBEE-BBEEBBBBBEBEE-EEBBBBBBBBBEBBBBBEBEEE-EE
PHD:98767545545522342589897265413652877558899899889
Pred:00999076677765566799999855555998885889990000000
dssp:-EE---HHHHHHHHHH-----HHHHHHHH--EE-------HHH--
define:EEEE-HHHHHHHHHHHHH---HHHHHHHHHEEEEEE-EEEE-----
stride:-EE---HHHHHHHHHHHH----HHHHHHHH--EE------------
```

První řádek, začínající prefixem „OrigSeq", udává primární strukturu a ostatní řádky označují sekundární strukturu zjištěnou různými metodami. Referenční metodou pro porovnání je metoda DSSP. Nejedná se o metodu predikce, nýbrž jde o experimentální zjištění sekundární struktury, zpravidla pomocí magnetické rezonance nebo rentgenových paprsků. Na základě zjištěných údajů lze definovat 8 tříd (typů) sekundární struktury. Tyto třídy jsou v této práci zjednodušeny na tři základní podle následující tabulky [26]:

DSSP třídy	symboly pro 8	symboly pro 3	Název třídy
3_{10}-helix	G	H	Helix
α-helix	H		
π-helix	I		
β-strand	E	E	Sheet
izolovaný β-bridge	B	C	Loop
Bend	S		
Turn	T		
Ostatní (spoje)	-		

Tabulka 5.1 Redukce 8 tříd na 3 třídy DSSP [26].

5.2 Celulární automat

Inicializaci, průběh a získané výsledky celulárního automatu lze ovlivnit několika parametry, které lze libovolně měnit. Inicializace samotného celulárního automatu proběhla vždy na základě empirických údajů z tabulky Chau-Fassman (tabulka 5.2). Na rozdíl od již zmíněné Chau-Fassmanovy metody je v této práci kladen větší důraz na interakce vzdálených sousedů na primární struktuře, které však mohou být v prostoru velmi blízko u sebe. Každá buňka automatu je reprezentována třemi údaji z této tabulky. V programu udávají tyto údaje schopnost buňky nacházet se v jednom ze tří cílových elementů sekundární struktury (α-helix, β-sheet, coil).

Jelikož má automat konečnou délku, je třeba se zabývat okrajovými podmínkami. Proteiny zpravidla netvoří kruhové útvary, proto ani propojení konců automatu nemá smysl. Proto vzniknou na obou koncích automatu pomocné buňky, které umožňují korektní práci automatu. Vlastnosti těchto okrajových buněk silně ovlivňují výpočet. Již bylo zmíněno, že tyto buňky bývají v běžných aplikacích CA většinou nulové, ale v této práci je nutné pro výpočet stanovit jiné hodnoty. Tento požadavek vychází z podstaty přechodové funkce, která bude popsána dále. Pokud by buňky byly nulové, podléhaly by velmi snadno vlivu svých sousedů. Stejnou nevýhodu vykazuje i případ duplikace buněk na začátku a konci

řetězce. Tato možnost byla zobecněna na přidání libovolných (i fiktivních) AK na oba konce řetězce. Jejich počet udává, jako i v předešlých případech, poloměr sousedů.

AK	A	R	D	N	C	E	Q	G	H	I
α-helix	142	98	101	67	70	151	111	57	100	108
β-sheet	83	93	54	89	119	37	110	75	87	160
coil	66	95	146	156	119	74	98	156	95	47
AK	L	K	M	F	P	S	T	W	Y	V
α-helix	121	114	113	57	77	83	108	69	106	50
β-sheet	130	74	105	138	55	75	119	137	147	170
coil	59	101	60	60	152	143	96	96	114	50

Tabulka 5.2 Parametry konformační preference v Chau-Fassmanově tabulce.

Důležitou fází simulace je výpočet nového stavu buňky. Tento výpočet musí zachovávat paralelismus celulárního automatu, který spočívá v souběžném výpočtu nových stavů pro všechny buňky. Aby bylo možné toto provést sekvenčně, obsahuje každá buňka aktuální stav, který je použit pro výpočet, a nový stav, do kterého se uloží výsledek. Jakmile jsou vypočteny všechny nové stavy, jsou prohlášeny za aktuální a automat je připraven k dalšímu kroku.

Samotná přechodová funkce spočívá v určení nejpravděpodobnějšího typu sekundární struktury v daném místě podle následujícího vzorce [2]:

$Prediction_{t+1} = max\,(H_{t+1},\, B_{t+1},\, C_{t+1})$,

kde t je číslo kroku automatu a jednotlivé složky pro každý typ sekundární struktury se spočítají podle následujících vzorců [2]:

$$H_{t+1,\,k} = \left(\sum_{j=-x}^{x} H_{t,\,k-j} \cdot P_j\right) / \left(\sum_{j=-x}^{x} P_j\right),$$

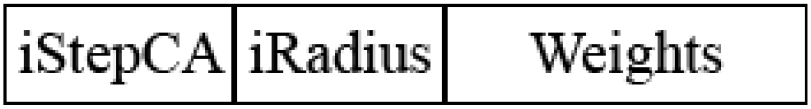
$$B_{t+1,\,k} = \left(\sum_{j=-x}^{x} B_{t,\,k-j} \cdot P_j\right) / \left(\sum_{j=-x}^{x} P_j\right),$$

$$C_{t+1,\,k} = \left(\sum_{j=-x}^{x} C_{t,\,k-j} \cdot P_j\right) / \left(\sum_{j=-x}^{x} P_j\right),$$

kde k je index konkrétní buňky v automatu, x je poloměr vstupující do přechodové funkce a P je váhový vektor, obsahující váhy jednotlivých sousedů.

5.3 Genetický algoritmus

Již bylo zmíněno, že pro genetický algoritmus, který hledá optimální kombinaci parametrů pro běh celulárního automatu, je zcela zásadní uspořádání dat v chromozomu. V této práci se chromozom skládá ze tří složek podle obrázku 5.1. První částí je počet kroků celulárního automatu. Druhým parametrem je poloměr sousedů a třetí parametr je vektor vah sousedů.



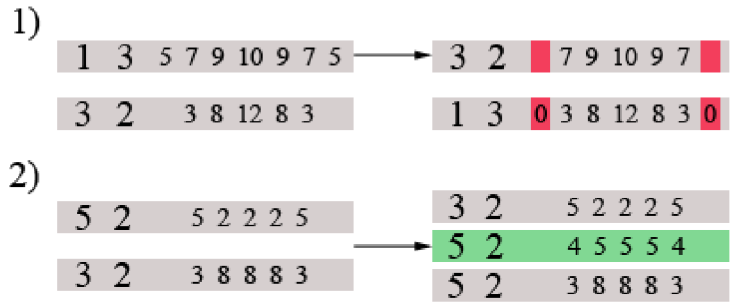
Obrázek 5.1 Chromozom se svými složkami.

Při vzniku nové generace dochází k selekci jedinců pro přežití pomocí ruletového mechanismu [13]. Jedinci jsou ohodnoceni podle tabulky 5.2, kde je každému jedinci přiřazena hodnota na základě jeho úspěšnosti při výpočtu sekundární struktury. Výběr pak probíhá náhodným generováním čísel od 1 do celkového součtu všech ohodnocení v rámci populace a podle vygenerovaného čísla se vybere první jedinec, jehož součet vlastního a všech předchozích ohodnocení je větší nebo roven vygenerovanému číslu. Platí pravidlo, že jedinci musí být dva, čili druhý jedinec se vybírá tak dlouho, dokud se neliší od prvního.

$Q_3 > 60$	50
$Q_3 > 55$	10
$Q_3 > 50$	1

Tabulka 5.2 Ohodnocení jedinců v závislosti na úspěšnosti predikce.

Tyto složky lze při vzniku nových generací ovlivnit použitím genetických operátorů. Operace křížení spočívá ve výměně genetického materiálu u dvou vybraných jedinců. Výměna první složky proběhne vždy. V 50% případů dojde k výměně druhé složky. Druhý případ vykazuje jistá specifika. Tedy pokud je poloměr různý u křížících se jedinců, je buďto zkrácen vektor vah na obou koncích, nebo je doplněn hodnotami 0, pokud poloměr vzrostl. Speciální případ nastane, pokud jsou poloměry stejné. V takovém případě jsou oba rodiče zachováni pro další populaci a navíc vznikne jedinec, který dědí první dvě složky z rodiče, který byl vybrán jako první a vektor vah je vytvořen z průměrných hodnot vah rodičů. Případy křížení jsou na obrázku 5.2. K mutaci dochází jen u jednoho jedince v populaci. Je vybrán vždy ten nejlepší chromozom z populace a dojde k inkrementaci jeho první složky podle obrázku 5.3. Posílí se tak vliv celulárního automatu na výsledek a do populace se tak může zavést jedinec, který má jedinečnou první složku, jež nebyla vygenerována v inicializačním kroku populace.



Obrázek 5.2 Křížení - výměna 1. a 2. složky (1) a vznik nového jedince průměrem (2).

Obrázek 5.3 Mutace nejlépe ohodnoceného jedince.

Četnost výskytu použití jednotlivých operátorů je inspirována zdrojem [24] upřednostňujícím křížení, ke kterému dochází v tomto případě vždy v jiné intenzitě, před mutací, která pouze zpomaluje konvergenci populace. V případě mutace dochází v určitých případech k zavedení unikátního počtu kroků celulárního automatu.

5.4 Kongruentní generátor pseudonáhodných čísel

Při simulaci dochází v určitých místech k jevům, které závisí na velikosti vygenerovaného náhodného čísla. Příkladem je inicializace populace, kde první složka je pseudonáhodné číslo od 1 do 4. Další využití pseudonáhodného generátoru je v rozhodovacím mechanismu pro křížení. Tady se generují čísla 1 nebo 2, z nichž každé má pravděpodobnost výskytu 50%, čímž lze ovlivnit vznikající jedince. Z těchto potřeb algoritmu vyplývá, že je nutné zavést generátor pseudonáhodné veličiny s rovnoměrným rozložením pravděpodobnosti [23]. Kongruentní generátor generuje konečnou posloupnost čísel, protože použitím operace modulo se posloupnost opakuje. Generátor je algoritmický, proto jsou čísla pseudonáhodná. Princip generátoru vychází z následující rovnice:

$x_{i+1} = (A \cdot x_i + B)\ mod\, M$,

kde A je konstanta s hodnotou 69 069, B je konstanta s hodnotou 1 a M je konstanta o velikosti 2^{32} podle [23]. Tento generátor generuje číslo od 0 do M-1 a dalšími úpravami lze umístit čísla do konkrétního intervalu.

Kapitola 6

Experimenty

Provádění experimentů lze rozdělit na 3 etapy. První etapa experimentů spočívala v nalezení optimálních okrajových buněk celulárního automatu. Ve druhé etapě byl pomocí genetického algoritmu nalezen optimální poloměr pro lokální přechodovou funkci celulárního automatu, což je počet sousedů po obou stranách na datasetu RS126. Současně s tím byl stanoven optimální počet kroků automatu a vektor vah sousedů. Aby byla dosažená úspěšnost predikce co nejpřesnější, ve třetí etapě se opět pomocí genetického algoritmu provedla predikce na datasetu CB513.

6.1 Okrajové buňky

Je nutné, aby nastavené okrajové buňky minimálně ovlivňovaly vnitřní buňky automatu. Podle experimentů [2] je vhodné přiřadit buňkám strukturu coil. V tabulce Chau-Fassman (tabulka 5.2) jsou to AK, které mají vysokou hodnotu konformační preference pro strukturu coil. Při experimentech bylo náhodně vybráno 5 různě dlouhých váhových vektorů. Pro každý řetězec z datasetu RS126 byla provedena predikce sekundární struktury a byly zaznamenány výsledky od 1. do 11. kroku automatu. Predikce proběhla nad každým řetězcem s využitím všech 5 váhových vektorů. Další údaje se pak týkají konkrétního vektoru a s ním dosažených hodnot.

Cílem experimentu bylo zjistit, která AK se nejvíce hodí pro použití jako okrajová buňka, s jejímž použitím se dosáhne při predikci nejvyšší úspěšnosti. V následujícím příkladu je zkoumána úspěšnost predikce při použití aminokyseliny Alanin (A) jako okrajové buňky. Následující výstup z testu udává výsledky. První řádek je pro výběr nejvhodnější okrajové buňky směrodatný. První i druhá hodnota udávají průměrný nejlepší výsledek na datasetu RS126 za postupného použití všech váhových vektorů. Další údaje se pak týkají konkrétního vektoru vah a s ním dosažených hodnot.

```
A: 0.531626, 0.564363
Výpočty na datasetu RS126
# Váhový vektor 0.000145794 0.140442 0.720968 1 0.720968 0.140442 0.000145794
# Průměrná úspěšnost Q3: 0.538053 (včetně cross-validace)
# Průměrná úspěšnost Q3: 0.568363
# Průměrná délka kroku: 4
```

V tomto experimentu nejlépe uspěly AK se schopností vytvářet strukturu coil, které jsou shrnuty v následující tabulce:

AK	Zkratka	Q_3(cross-validace)	Q_3
Asparagin	N	0.563249	0.597539
Glycin	G	0.564187	0.597914
Prolin	P	0.564840	0.598628
Serin	S	0.563566	0.597615
X300	X	0.564303	0.597436

Tabulka 6.1 Nejvhodnější kandidáti AK na okrajové buňky.

V tabulce 6.1 se nachází aminokyselina s označením X300. Jedná se o fiktivní, reálně neexistující AK, která nese pouze vlastnosti, které podporují tvorbu struktury coil. Průměrný součet hodnot konformační preference (viz kapitola 5.2) pro všechny struktury jedné aminokyseliny je 302,85. Byly vytvořeny i další AK, které zachovávají tento průměr. Označení je voleno podle hodnoty P(c). V následující tabulce 6.2 jsou výsledky predikce za použití jednotlivých fiktivních AK jako okrajových buněk:

AK	P(a)	P(b)	P(c)	Q_3(cross-validace)	Q_3
X300	0	0	300	0.564303	0.597436
X240	30	30	240	0.564284	0.598393
X200	50	50	200	0.557175	0.589829
X136	95	70	136	0.555441	0.592116
X100	100	100	100	0.544395	0.578646

Tabulka 6.2 Fiktivní AK jako kandidáti na okrajové buňky.

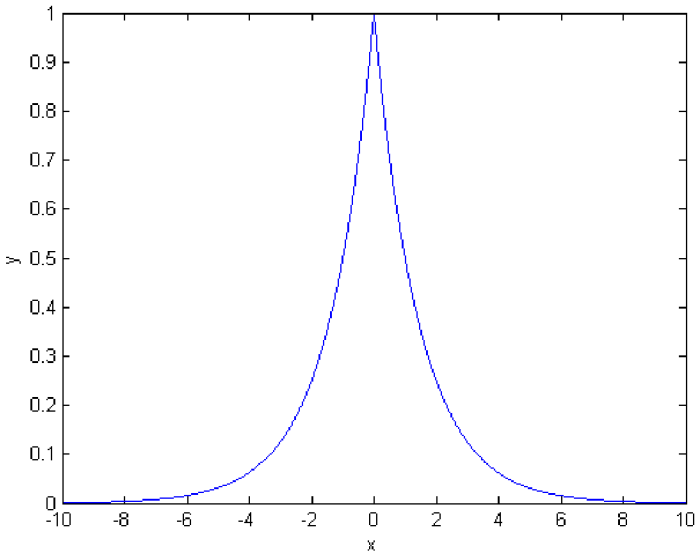
Tyto výsledky poskytují základ pro druhý experiment, ve kterém jsem hledal optimální vektor vah sousedů pro lokální přechodovou funkci, počet kroků automatu a velikost poloměru sousedů pomocí genetického algoritmu.

6.2 Parametry predikce

Cílem experimentů je nalezení optimálního počtu kroků a počtu sousedů v celulárním automatu a také optimální vektor vah pro tyto sousedy pomocí genetického algoritmu. Prvním krokem algoritmu je inicializace počáteční populace. Již bylo zmíněno složení chromozomu. Počet kroků automatu je generováno zmíněným generátorem od 1 do 4. Velikost okolí se dopočítá z konkrétního váhového vektoru. Zdroje čísel pro váhový vektor jsou dva. Prvním způsobem lze vygenerovat celkem 185 vektorů, jejichž čísla leží v oboru hodnot následující funkce:

$y = a^{-|x|}$,

kde $a \in \{2, 2.5, 3, \cdots, 20\}$. Prostřední číslo je vždy 1 a počet sousedů po obou stranách je od 3 do 7. Nejdelší váhový vektor tedy obsahuje 15 čísel. Proměnná x závisí na vzdálenosti souseda od centrální AK. Graf funkce, který je vytvořen pomocí programu MATLAB, je na následujícím obrázku 6.1:



Obrázek 6.1 Graf funkce, jejíž obor hodnot H(f) obsahuje čísla vektoru vah.

Z vlastností funkce plyne pro přechodovou funkci pravidlo, že vzdálenější buňky se ovlivňují méně než buňky blízké. Příklad váhového vektoru v tabulce 6.3:

Sousedé	-3	-2	-1	0	1	2	3
Váha	0.03	0.48	0.88	1	0.88	0.48	0.03

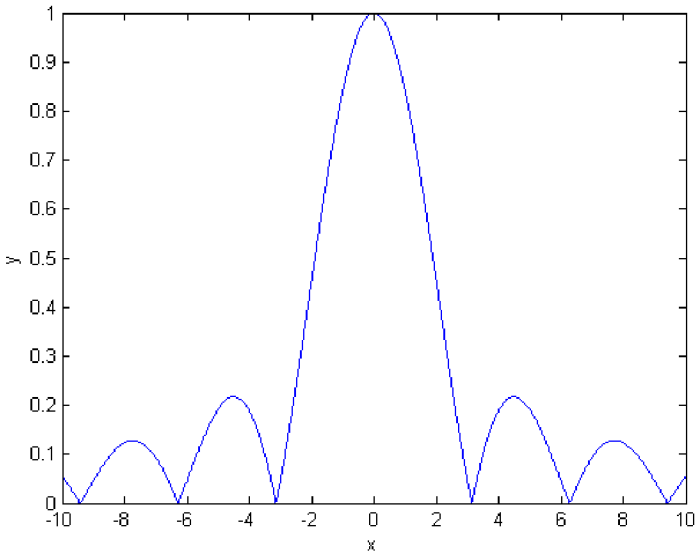
Tabulka 6.3 Příklad váhového vektoru.

Druhou možností, jak vytvořit váhový vektor, je ruční vyzkoušení různých jiných možností. V této práci byly použity vektory, jejichž čísla leží na funkci $sinc(x)$ nebo funkci jí podobné. Vhodnost těchto funkcí spočívá v existenci sekundární struktury, vlivem které mohou být některé AK blíž k sobě, než kdyby byl protein lineární bez náznaku sekundárních struktur. Příklad takového vektoru je v tabulce 6.4:

Sousedé	-5	-4	-3	-2	-1	0	1	2	3	4	5
Váha	0.02	0.14	0.31	0.2	0.96	1	0.96	0.2	0.31	0.14	0.02

Tabulka 6.4 Příklad váhového vektoru pro funkci sinc(x).

Vlastnosti této funkce nejlépe ukazuje následující obrázek 6.2:



Obrázek 6.2 Graf funkce sinc(x), jejíž vlastnosti jsou použity pro generování vektorů vah.

Příkladem samotného výpočtu je následující tabulka 6.5, která zaznamenává výsledky v každé generaci. Vektor je z důvodu úspory místa zkrácený na polovinu, váhy jsou vždy symetrické. Testovacím datasetem byl RS126 a na okrajové podmínky byla použita AK X200.

Generace	StepsCA	Q_3	Q_a	Q_b	Q_c	Vektor
1, 2	3	0.5734	0.3894	0.4530	0.6134	0.00237037 0.260991 0.799413 1
3, 4	2	0.5755	0.3959	0.4416	0.6169	0.13 0.05 0 0.41 0.9 0.96

Tabulka 6.5 Seznam výsledků za použití populace 88 chromozomů.

V případě další úpravy operátoru křížení a ohodnocení jedinců v rámci populace už ke zlepšení nedošlo. Tyto testy byly úspěšnější než při použití jiných okrajových podmínek. Se stejnými parametry, které platily pro tabulku 6.5, proběhl test pro okrajové buňky s AK X100, X136, X240, X300. Výsledky těchto testů jsou v následující tabulce 6.6. Pokud je na jednom řádku zaznamenáno více generací, jde pouze o úsporu místa a ostatní parametry byly ve všech těchto sjednocených generacích stejné.

Generace	StepsCA	Special	Q_3	Vektor
1, 2, 3, 4	3	X300	0.5745	0.000512 0.185664 0.755303 1
1, 2, 3	3	X240	0.5742	0.000512 0.185664 0.755303 1
4	2	X240	0.5757	0.13 0.05 0 0.41 0.9 0.96
1, 2, 3, 4, 5	3	X136	0.5711	0.00364133 0.287116 0.812225 1
1, 2, 3	2	X100	0.5569	0.00364133 0.287116 0.812225 1

Tabulka 6.6 Seznam výsledků s okrajovými buňkami X100, X136, X240, X300.

Testy byly provedeny i za použití menšího počtu 13 chromozomů, které dosahovaly nejvyšší úspěšnosti. Účelem omezení bylo prohloubit změny pouze u těchto nejlepších jedinců. Výsledek se však oproti původnímu nelišil, jak ukazuje tabulka 6.7:

Generace	StepsCA	Special	Q_3	Vektor
3	3	X240	0.5738	0.00594025 0.213781 0.758241 1
5	2	X200	0.5755	0.13 0.05 0 0.41 0.9 0.96

Tabulka 6.7 Nejlepší výsledky za použití populace 13 chromozomů.

AK X136 vznikla jako průměrná AK pro dataset RS126, která bere v úvahu rozdílné počty jednotlivých elementů sekundární struktury. Navzdory předpokladům nevykazovala výraznější úspěšnost oproti ostatním fiktivním AK. Lze tedy podle těchto výsledků používat AK X200 nebo X240 i na jiných datasetech bez rizika, že procentuální zastoupení jednotlivých útvarů sekundární struktury zásadně ovlivní úspěšnost predikce. Rozšířením experimentů na jiné datasety se zabývám ve 3. etapě.

6.3 Změna datasetu

Třetí etapa experimentů, prováděných za účelem nalezení optimálních parametrů predikce, je zaměřena na otestování doposud nejoptimálnějších parametrů na jiném datasetu. Tímto datasetem je po vzoru [2] dataset CB513 s 513 řetězci. Předpokladem bude nižší úspěšnost Q_3, protože je k dispozici větší počet řetězců. Díky tomuto počtu však bude stanovená úspěšnost přesnější než na datasetu RS126. V následující tabulce 6.8 jsou výsledky dosažené s konkrétními okrajovými podmínkami, které vykazovaly úspěšnost na datasetu RS126. Populace genetického algoritmu obsahovala 88 chromozomů.

Generace	StepsCA	Special	Q_3	Vektor
2	2	X300	0.5610	0.0911126 0.738173 1.64495 2
1	3	X240	0.5635	0.000512 0.185664 0.755303 1
2, 3	3	X240	0.5635	0.000657516 0.196276 0.762333 1
4, 5	2	X240	0.5637	0 0.41 0.9 0.96
1, 2	3	X200	0.5632	0.000512 0.185664 0.755303 1
3	3	X200	0.5633	0.000578704 0.190786 0.758736 1

Tabulka 6.8 Výsledky dosažené na datasetu CB513.

Při aplikaci menší sady chromozomů v počáteční populaci už k žádným dalším zlepšením nedošlo. Za použití symetrických váhových vektorů, tedy těch, které mají prostřední člen a dvojice stejných členů po obou stranách, jsem pomocí experimentů došel ke konkrétním výsledkům, které lze porovnat s ostatními metodami.

6.4 Porovnání výsledků

Každá metoda predikce sekundární struktury proteinů je, jak již bylo zmíněno, založená na jiném přístupu a má své výhody i nevýhody. V této práci byl kladen důraz zejména na rychlost predikce. V porovnání s metodami strojového učení nebo konsenzuálními metodami je nesrovnatelně rychlejší. Podle zdroje [18] se průměrná doba predikce sekundární struktury jednoho proteinu na běžném počítači pohybuje okolo 30 minut. Program CELLPRED predikuje sekundární strukturu řádově v desítkách milisekund.

Další výhodou je skutečnost, že oproti těmto extrémně pomalým metodám nepotřebuje CELLPRED žádnou trénovací množinu. Sestavení vhodné trénovací množiny je podobné

jako u sestavení datasetu. Vhodně zvolené kombinace aminokyselinových řetězců tvořících dataset a jejich počet má zásadní vliv na výsledek. V této práci proběhlo testování na datasetech pouze ve fázi experimentování s různými parametry a při predikci již není dalších řetězců třeba.

Další výhodou oproti metodám statistickým, mezi které patří například metody GOR, je využití vlivu sousedů na celkový výsledek predikce. Novější verze metody GOR III už tento deficit nemá.

Nevýhodou zůstává nižší úspěšnost predikce oproti sofistikovanějším metodám, protože výrazné zrychlení predikce bylo provedeno na úkor přesnosti. Vliv celulárního automatu však i přes tento nedostatek zůstává pozitivní, protože při srovnání úspěšnosti před použitím automatu a po použití došlo ke zlepšení predikce více jak o 10%. V tabulce 6.9 je souhrn výsledků nad datasetem RS126. První řádek udává predikci za použití prosté inicializace pomocí Chau-Fassmanovy tabulky. V tabulce je i výsledek metody GOR IV [2]:

Metoda	Q_3	Přístup
Init	0.4678	Inicializace tabulkou
CELLPRED	0.5742	Celulární automat
GOR IV	0.5330	Statistická metoda
CONS	0.7497	Neuronové sítě
DSC	0.7120	Statistická metoda
NNSSP	0.7282	Metoda nejbližších sousedů
PHD	0.7364	Neuronové sítě
PRED	0.7037	Neuronové sítě
ZPRED	0.6280	Dodatečné informace

Tabulka 6.9 Srovnání výsledků metod predikce sekundární struktury.

Pro metodu CELLPRED, jejíž výsledky jsou v tabulce 6.9, platí následující parametry:

Kroky CA	Váhový vektor	Special
3	0.000512 0.185664 0.755303 1	X240

Tabulka 6.10 Optimální parametry predikce.

Kapitola 7

Závěr

Cílem této práce bylo vytvoření výpočetního frameworku pro predikci sekundární struktury proteinů. Hlavním prostředkem predikce je celulární automat, jehož charakteristické vlastnosti a výhody ho staví jako vhodného kandidáta pro modelování proteinu. Samotný výpočet sekundární struktury probíhal po inicializaci bez centrálního řízení určený počet iterací, kdy spolu vzájemně interagovaly sousední buňky automatu. Pro nalezení optimálního řešení byl použit genetický algoritmus. Z počáteční populace chromozomů vznikaly novější populace, které obsahovaly jiné chromozomy s parametry predikce a vykazovaly vyšší úspěšnost po aplikaci ve výpočetním frameworku. Nové kombinace parametrů byly generovány použitím genetických operátorů - křížení a mutace. Aby bylo dosaženo co nejoptimálnějších hodnot, byly použity 2 různé datasety pro testování vhodných parametrů predikce. Na základě výsledků nad oběma datasety byla vybrána nejoptimálnější kombinace těchto parametrů.

Při srovnání výsledků s jinými metodami bylo zjištěno, že úspěšnost predikce není tak vysoká jako u sofistikovanějších metod. Výhodou je však její vysoká rychlost, protože predikce trvá na běžném počítači řádově několik desítek milisekund. Navíc metoda bere v úvahu i děje a vazební síly v proteinech, které celou sekundární strukturu utvářejí. Výběr ideální kombinace parametrů byl proveden s ohledem na úspěšnost nad oběma zkoumanými datasety. Při experimentech nad těmito datasety se v obou případech prosadila v rámci populace kombinace parametrů v tabulce 6.10. Tato kombinace parametrů byla prohlášena za optimální a je používána pro predikci sekundární struktury jakéhokoliv dalšího proteinového řetězce.

Dalším možným postupem při zdokonalování přístupu k predikci pomocí celulárního automatu by mohla být změna fáze inicializace celulárního automatu. Chau-Fassmanova tabulka vychází ze statistických výpočtů, které nejsou vždy přesné. Celulární automat dokázal po několika krocích zpřesnit tuto inicializaci o více než 10%. Nejvyšší úspěšnost, které bylo dosaženo, byla 57,57% na datasetu RS126 a 56,37% na datasetu CB513. Vzhledem k tomu, že se výsledky liší daleko méně než v [2], kde je rozdíl téměř dvojnásobný, je zřejmé, že při rozšíření testů na jiné datasety nebude docházet k výraznějším výkyvům predikce.

Metoda predikce sekundární struktury proteinů pomocí celulárního automatu má stejně jako jiné metody výhody i nevýhody. Její využití je vhodné zejména v situacích, kdy záleží na čase, protože je rychlejší než jiné metody.

Literatura

[1] Branden, C.: *Introduction to Protein Structure.* Garland Science, druhé vydání, 1999, ISBN 978-0815323051.

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

[3] Codd, E. F.: *Cellular Automata.* Academic Press, New York, 1968.

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

[5] Darwin, C.: *O vzniku druhů přírodním výběrem čili zachováním vhodných odrůd v boji o život.* F. Klapálek, 1914.

[6] Fogel, D.: *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence.* Piscataway, NJ: IEEEE Press, třetí vydání, 2006.

[7] Frishman, D.; Argos, P.: Seventy-five percent accuracy in protein secondary structure prediction. *PROTEINS: Structure, Function, and Genetics*, ročník 27, 1997: s. 329–335.

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

[9] Garnier, J.; Gibrat, J. F.; Robson, B.: GOR secondary structure prediction method version IV. *Methods in Enzymology*, ročník 266, 1996: s. 540–553.

[10] Goldberg, D.: *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Weasley, Reading, MA, 1989.

[11] Granseth, E.; Viklund, H.; Elofsson, A.: ZPRED: predicting the distance to the membrane center for residues in alpha-helical membrane proteins. *Bioinformatics*, ročník 22, 2006: s. 191–196.

[12] Holland, J.: *Adaptation in Natural and Artifical Systems.* The University of Michigan Press, Ann Arbor, Michigan, 1975.

[13] Hynek, J.: *Genetické algoritmuy a genetické programování.* Grada Publishing, a.s., 2008, ISBN 978-80-247-2695-3.

[14] King, R.; Ouali, M.; Strong, A.; aj.: Is it better to combine predictions? *Protein Engineering*, ročník 13, 2000: s. 15–19.

[15] Koza, J.: *Genetic Programming. On the Programming of Computers by Means of Natural Selection.* Cambridge, MA: MIT Press, 1994.

[16] Koza, J.; Keane, M.; Steeter, M.; aj.: *Genetic Programming IV: Routine Human-Competitive Machine Inteligence.* Kluwer Academic Publishers, 2003.

[17] Mařík, V.; Štěpánková, O.; Lažanský, J.; aj.: *Umělá inteligence 3.* Academia, 2001, ISBN 80-200-0472-6.

[18] McGuffin, L.; Bryson, K.; Jones, D.: The PSIPRED protein structure prediction server. *Bioinformatics*, ročník 14, 2000: s. 404–405.

[19] Michalewicz, Z.; Fogel, D.: *How to Solve It: Modern Heuristics.* Springer, Berlin, 2000.

[20] Mitchell, M.: *An Introduction to Genetic Algorithms.* Cambridge, MA: MIT Press, 1996.

[21] von Neumann, J.: *Theory of self-reproducing automata.* (A. W. Burks, ed.), University of Illinois Press, Urbana and London, 1966.

[22] Nečas, O.; kolektiv: *Obecná biologie pro lékařské fakulty.* H&H, 2000, ISBN 80-86022-46-3.

[23] Peringer, P.: Modelování a Simulace [online]. `http://www.fit.vutbr.cz/study/courses/IMS/public/prednasky/IMS.pdf`, 2011-09-22 [cit. 2012-05-01].

[24] Schaffer, J.; Caruana, R.; Eshelman, L.; aj.: *A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization.* In: Schaffer, J. (ed): Proceedings of the Third International Conference on Genetic Algorithms, San Mateo, California, Morgan Kaufmann, 1989, p. 51-60.

[25] Sekanina, L.: *Evolvable components: From theory to hardware implementations.* Springer, 2004, ISBN 978-3-540-40377-7.

[26] Wang, G.; Li, T.; Grzymala-Busse, J.; aj.: *Rough Sets and Knowledge Technology.* Springer, 2008, ISBN 3-540-79720-3.

[27] Wolfram, S.: *A New Kinf of Science.* Wolfram Media, Inc, 2002, ISBN 1-57955-008-8.

Příloha A

Obsah CD

Přílohový disk CD obsahuje zdrojové kódy programu, této práce, dokumentaci a další užitečné a doplňkové soubory.

/cellpred

V této složce se nachází zdrojové kódy programu CELLPRED. Existuje zde také Makefile, kterým lze program přeložit v prostředí linux pomocí volání „make". Pro vyzkoušení běhu predikce slouží příkaz „make run" a vyzkoušení simulace pomocí genetického algoritmu příkaz „make sim".

/vysledky

Tato složka obsahuje logy všech provedených testů a simulací, rozdělených po složkách. Vnitřní složky s logy mají název odvozený od názvu konkrétní simulace.

/cellpred/RS126 a /cellpred/CB513

V těchto složkách se nachází pro program CELLPRED upravené datasety, na kterých byly testovány parametry predikce. Pro zopakování experimentů obsahují také soubor *files.input* se seznamem názvů souborů.

/cellpred/neighbours

Složka obsahující soubory s váhovými vektory, které sloužili jako vstupní množina parametrů pro genetický algoritmus.

/cellpred/proteins.input

Soubor se vstupními řetězci, oddělenými znakem konce řádku, pro predikci sekundární struktury.

/latex

Složka obsahuje zdrojové soubory pro vytvoření tohoto dokumentu včetně obrázků.

/doc

V této složce se nachází dokumentace, vygenerovaná pomocí programu Doxygen. Jedná se o popis tříd, jejichž objekty zapouzdřují data a operace nad nimi.

Příloha B

Návod ke spuštění

Program CELLPRED umožňuje spuštění ve dvou různých režimech, které závisí na parametrech příkazové řádky. Pokud se spustí program bez parametrů, potom se na standardní výstup vypíše nápověda pro spuštění. První možností běhu je predikce sekundární struktury, za jejímž účelem byla aplikace napsána. Následující příklad je pro spuštění v prostředí linux:

./cellpred proteins.input,

kde textový soubor *proteins.input* obsahuje řetězce aminokyselin, oddělené znakem konce řádku. AK jsou reprezentovány standardně jedním velkým písmenem podle tabulky 5.2. Lze také použít příkaz „make run", který provede totéž.

Druhý mód běhu programu umožňuje další zkoumání predikce sekundární struktury a provádí simulaci s využitím genetického algoritmu. Spuštění opět v prostředí linux je následující:

./cellpred RS126 files.input 5 neighbours/mala_mnozina.input,

kde *RS126* je název datasetu pro experimentování a zároveň název složky, ve které se nachází řetězce, *files.input* je název souboru, který obsahuje seznam názvů souborů s řetězci, a *neighbours/mala_mnozina.input* je soubor, který obsahuje váhové vektory. Za každým váhovým vektorem je znak konce řádku a za každým číslem je vždy mezera. Pokud tento parametr chybí, program vygeneruje své vlastní váhové vektory do souboru *neighbours.input*. Číslo 5, nebo jakékoliv jiné přirozené číslo od 1 do $2^{32} - 1$, udává počet dovolených populací genetického algoritmu. Simulaci lze spustit také přímo pomocí příkazu „make sim".