Univerzita Komenského v Bratislave
Fakulta Matematiky, Fyziky a Informatiky

Algoritmy na rozpoznávanie neuropeptidov

Diplomová práca

Bratislava, 2014 **Bc. Andrej Ridzik**

Univerzita Komenského v Bratislave
Fakulta Matematiky, Fyziky a Informatiky

Algoritmy na rozpoznávanie neuropeptidov

Diplomová práca

Študijný program:	Informatika
Študijný odbor:	2508 Informatika
Školiace pracovisko:	Katedra informatiky FMFI
Školiteľ:	Mgr. Bronislava Brejová, PhD.

Bratislava, 2014 **Bc. Andrej Ridzik**

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Andrej Ridzik
Študijný program: informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.2.1. informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Algoritmy na rozpoznávanie neuropeptidov
Algorithms for Identification of Neuropeptides

Cieľ: Neuropeptidy sú krátke proteíny používané na posielanie signálov medzi neurónmi v mozgu. Cieľom práce je vyvinúť systém založený na technikách strojového učenia rozpoznávajúci gény kódujúce tieto proteíny v genóme.

Vedúci: Mgr. Bronislava Brejová, PhD.
Katedra: FMFI.KI - Katedra informatiky
Vedúci katedry: doc. RNDr. Daniel Olejár, PhD.

Dátum zadania: 14.10.2011

Dátum schválenia: 02.11.2011 prof. RNDr. Branislav Rovan, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Čestne prehlasujem, že som túto diplomovú prácu vypracoval samostatne s použitím citovaných zdrojov a pod dohľadom mojej školiteľky.

Rád by som sa poďakoval mojej školiteľke, Mgr. Bronislave Brejovej, PhD.,
za jej odborné vedenie, inšpiratívne rady a cenné pripomienky.
Ďakujem taktiež mojej rodine a priateľom za ich podporu a povzbudenie.

Abstrakt

Základnou jednotku genetickej informácie sú gény, ktoré určujú, aké proteíny si dokáže daný organizmus vytvárať. Tieto proteíny však nie vždy musia byť finálnym produktom, ale sa po vytvorení podrobujú rôznym biologickým procesom, ktorými vzniká žiadaný produkt. Jedným z príkladov takýchto produktov sú aj neuropeptidy, ktoré majú viaceré veľmi dôležité funkcie, ako napríklad vzájomnú komunikáciu neurónov. Existuje už niekoľko algoritmov na rozpoznávanie neuropeptidov, ktorých úlohou je identifikovať, aké neuropeptidy si skúmaný organizmus dokáže vytvárať. Tieto metódy sa však zaoberajú riešením len čiastkových problémov, a preto sme navrhli systém založený na metódach strojového učenia, ktorým je možné rozpoznať produkované neuropeptidy priamo z genetickej informácie organizmu, resp. z proteínov, ktoré sú v nej zakódované.

Kľúčové slová: proteín, prekurzor, neuropeptid, strojové učenie, support vector machines, conditional random fields

Abstract

A gene is the basic unit of genetic information that contains the information needed to make a specific protein. These proteins are not always the final product, and they can undergo various biological processes resulting in the desired product. An example of such products are neuropeptides with a number of very important functions, e.g. communication between neurons. There already exist several algorithms for identification of neuropeptides, whose task is to identify what neuropeptides the studied organism is able to produce. However, these methods deal only with several partial problems, and therefore we have designed a system based on machine learning methods to identify neuropeptides directly from an organism's genetic information, i.e. out of proteins which the genome encodes.

Keywords: protein, precursor, neuropeptide, machine learning, support vector machines, conditional random fields

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

Neuropeptidy sú krátke proteíny, ktoré zabezpečujú vzájomnú komunikáciu neurónov, modulujú ich rôzne funkcie a zapájajú sa do viacerých mozgových činností ako príjem potravy, učenie, zvládanie stresu, či pociťovanie bolesti. Nakoľko ide o pomerne krátke proteíny, vznikajú štiepením dlhšieho proteínu na niekoľko menších častí. Jedným z dôležitých súčasných problémov je zistiť, aké neuropeptidy dokáže skúmaný organizmus vytvárať. Zložité experimentálne metódy sú veľmi náročné a často ani nie vždy realizovateľné. Práve z týchto dôvodov sa viaceré vedecké skupiny pokúšajú o nové metódy, ktorými by sa tento problém dal riešiť výpočtovo, bez nutnosti zdĺhavých a náročných experimentov v biologickom laboratóriu.

K tomuto účelu bolo navrhnutých už niekoľko postupov, no doposiaľ sa všetky zaoberali len čiastkovými problémami, ktorým je napríklad problém zistenia, či zo skúmaného proteínu neuropeptidy vznikajú [7]. Kompletný problém pozostávajúci z takejto identifikácie až po samotné rozpoznanie neuropeptidov doteraz nebol dostatočne skúmaný, a preto sme sa mu rozhodli venovať v našej práci. Za týmto účelom sme navrhli systém využívajúci metódy strojového učenia, ktorý má za úlohu riešiť žiadaný problém, pričom je schopný využiť rôzne druhy informácií.

V prvej kapitole tejto práce najprv predstavíme niektoré základné biologické pojmy, ktorým je za účelom pochopenia problému dôležité rozumieť. Samotnému problému rozpoznávania neuropeptidov sa podrobne venuje druhá kapitola, v ktorej uvádzame aj niekoľko súčasných riešení čiastkových problémov. V závere tej istej kapitoly stručne predstavujeme nami navrhnutý systém, ktorému je venovaná ostatná časť tejto práce. V tretej kapitole sa bližšie venujeme problému rozpoznávania miest štiepenia proteínov a predstavíme spôsob, akým je táto informácia naším systémom získavaná. Výsledky tohto čiastkového problému taktiež porovnáme s aktuálne používanou webovou aplikáciou NeuroPred [4]. V poslednej kapitole tejto práce sa venujeme samotnému problému rozpoznávania neuropeptidov a podrobne popisujeme činnosť nášho systému založeného na modeli strojového učenia s názvom *conditional random fields* [13]. V tejto kapitole sa taktiež sústreďujeme na opis použitia nášho systému pri riešení problémov spojených s rozpoznávaním neuropeptidov a uvádzame niekoľko experimentov, ktoré boli realizované za účelom testovania úspešnosti systému z viacerých hľadísk.

Kapitola 1

Základné biologické pojmy

Na to, aby sme mohli definovať problém rozpoznávania neuropeptidov a zaoberať sa prístupmi na jeho riešenie, je nutné najprv vysvetliť niektoré biologické pojmy. V tejto kapitole opíšeme základné pojmy, ktoré sú nevyhnutné na pochopenie ostatného textu.

1.1 DNA

DNA je skratka pre deoxyribonukleovú kyselinu (z angl. deoxyribonucleic acid). Je nositeľkou genetickej informácie bunky, ktorá riadi rast, delenie a regeneráciu bunky. DNA je tvorená polynukleotidovými vláknami a je väčšinou uložená v bunke v podobe dvojzávitnicovej špirály.

Základnou stavebnou jednotkou DNA sú *nukleotidy*. Nukleotid v DNA sa skladá z troch zložiek a to z fosfátovej zložky, sacharidovej zložky (deoxyribóza) a purínovej alebo pyrimidínovej dusíkatej bázy. V nukleotidoch, nachádzajúcich sa v DNA, sa vyskytujú štyri druhy dusíkatých báz: *adenín*, *guanín*, *cytozín* a *tymín*. Jednotlivé nukleotidy sa od seba líšia len dusíkatými bázami, a preto sa aj zvyknú označovať začiatočnými písmenami daných báz, teda A, G, C a T.

Celková genetická informácia bunky sa nazýva *genóm* a je zakódovaná práve poradím nukleotidov A, G, C, T v molekulách DNA, ktoré sa nazývajú *chromozómy*. Proces, pri ktorom sa z jednotlivých molekúl DNA získava genetická informácia (teda poradie dusíkatých báz) sa nazýva *sekvenovanie*. Na sekvenovanie DNA sekvencií bolo vyvinutých už niekoľko technologických postupov, ktoré sa od seba líšia predovšetkým rýchlosťou a cenovou dostupnosťou. V súčastnosti sa podarilo úspešne osekvenovať už veľké množstvo DNA sekvencií rôznych organizmov, z ktorých sú mnohé prístupné i verejnosti prostredníctvom databázy GenBank [26].

1.2 Proteíny

Proteíny (alebo bielkoviny) sú nevyhnutnými zložkami všetkých rastlinných i živočíšnych buniek a plnia niekoľko funkcií. V podobe enzýmov sú nenahraditeľné pri regulácii rôznych biochemických reakcií, plnia stavebnú funkciu bunky, v podobe protilátok sa podieľajú na obranyschopnosti organizmu, môžu regulovať tvorbu ďalších proteínov a taktiež slúžia organizmu ako rezervné látky. V súčasnosti napríklad vieme, že v ľudskom tele sa vyskytuje až vyše 10000 rôznych proteínov.

Proteíny sú tvorené reťazcom *aminokyselín*, ktorý môže byť rôznej dĺžky. Poznáme 20 základných druhov aminokyselín a každej je priradené jedno písmeno abecedy (tab. 1.1). Jednotlivé aminokyseliny sa medzi sebou líšia viacerými vlastnosťami a podľa tých môžu byť rozdelené do skupín. Existujú viaceré menej či viac špecifické rozdelenia, no my v tabuľke uvádzame to rozdelenie, s ktorým sme neskôr v práci pracovali.

Ľubovoľnú postupnosť aminokyselín nazývame *peptid*. Definícia proteínov nie je jednoznačná, ale štandardne sa za proteíny považujú predovšetkým dlhšie peptidy spĺňajúce vzájomnú dohodu biológov.

Skratka	Aminokyselina	Skupina	Kodóny
A	Alanín	alifatické	GCA, GCC, GCG, GCT
C	Cysteín	obsah. síru	TGC, TGT
D	Kys. asparágová	kyslé	GAC, GAT
E	Kys. glutámová	kyslé	GAA, GAG
F	Fenylalanín	aromatické	TTC, TTT
G	Glycín	alifatické	GGA, GGC, GGG, GGT
H	Histidín	bázické	CAC, CAT
I	Izoleucín	alifatické	ATA, ATC, ATT
K	Lyzín	bázické	AAA, AAG
L	Leucín	alifatické	CTA, CTC, CTG, CTT, TTA, TTG
M	Metionín	obsah. síru	ATG (Štart kodón)
N	Asparagín	amidické	AAC, AAT
P	Prolín	alifatické	CCA, CCC, CCG, CCT
Q	Glutamín	amidické	CAA, CAG
R	Arginín	bázické	CGA, CGC, CGG, CGT, AGA, AGG
S	Serín	hydroxylické	TCA, TCC, TCG, TCT, AGT, AGC
T	Treonín	hydroxylické	ACA, ACC, ACG, ACT
V	Valín	alifatické	GTA, GTC, GTG, GTT
W	Tryptofán	aromatické	TGG
Y	Tyrozín	aromatické	TAC, TAT
*	Stop		TAA, TAG, TGA (Stop kodóny)

Tabuľka 1.1: Tabuľka aminokyselín a im prislúchajúcich kodónov. Preložením jednotlivých kodónov v géne vzniká reťazec aminokyselín, t. j. proteín.

1.3 Gén

Gén je základnou jednotkou genetickej informácie. Je to časť DNA sekvencie, z ktorej sa zložitými biochemickými procesmi vytvárajú proteíny, prípadne len RNA sekvencie. Jeden gén pozostáva z niekoľkých striedajúcich sa exónov a intrónov. Exóny tvoria úseky, ktoré kódujú jednotlivé aminokyseliny, pričom každá aminokyselina je kódovaná trojicou nukleotidov, tzv. *kodónom*. Intróny predstavujú neprekladané úseky génu, ktoré sa pred vytváraním proteínu vystrihujú. Pospájaním všetkých exónov génu a následným preložením kodónov podľa tabuľky 1.1 vznikne postupnosť aminokyselín, t. j. proteín, ktorý je daným génom kódovaný.

Proteín, ktorý je nejakým génom kódovaný, nemusí byť vždy koncovým produktom, ale môže byť len medziproduktom, z ktorého sa následne vytvára iná látka, resp. niekoľko ďalších látok. Takýto medziprodukt sa štandardne nazýva *prekurzor*. Modifikácie, ktoré z prekurzoru vytvoria konkrétny finálny produkt, závisia od jeho konkrétneho

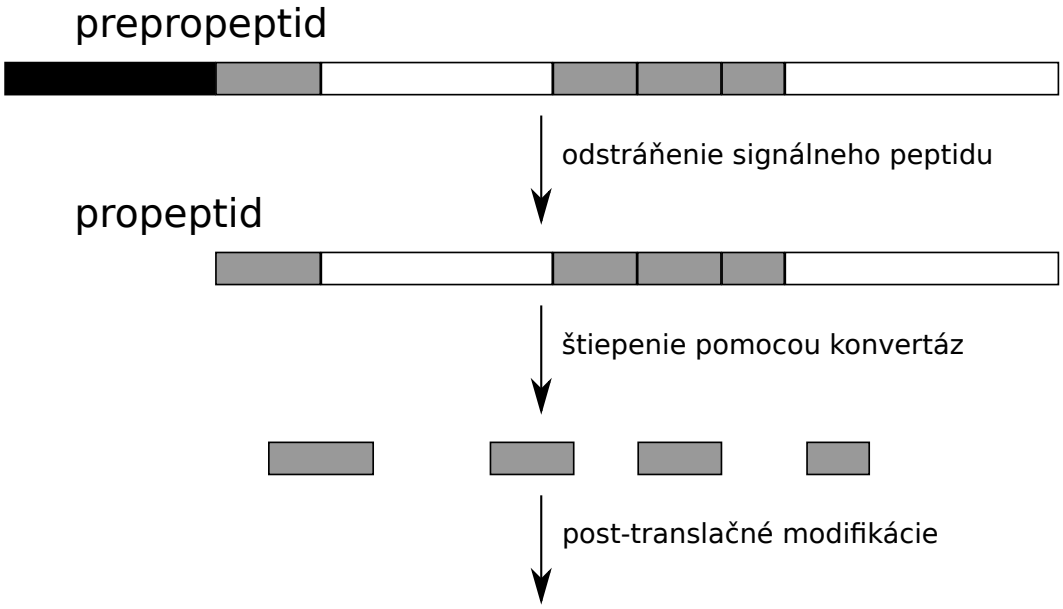
typu a môžu pozostávať z odstránenia niektorých aminokyselín na jeho koncoch alebo z naviazania nejakej chemickej skupiny.

1.4 Neuropeptidy

Neuropeptidy sú krátke peptidy (reťazce aminokyselín), ktoré sú produkované a následne vylučované z neurónov. Neuropeptidy vo všeobecnosti modulujú ich rôzne funkcie. Zabezpečujú komunikáciu medzi neurónmi, ovplyvňujú mozgovú aktivitu a zapájajú sa do rôznych mozgových funkcií ako príjem potravy, učenie, pamäť, zvládanie stresu, pociťovanie bolesti, no môžu mať taktiež mnoho ďalších účinkov.

Nakoľko ide o pomerne krátke reťazce aminokyselín, ktoré sú dokonca následne vylučované z bunky von, tak pre ne neexistuje samostatný gén, ale vznikajú z dlhých neuropeptidových prekurzorov prostredníctvom zložitého enzymatického procesu zahŕňajúceho štiepenie na jeden alebo viacero krátkych úsekov a rôzne *post-translačné modifikácie* (skratka PTMs), ako napríklad odstránenie niekoľkých koncových aminokyselín alebo naviazanie určitej konkrétnej chemickej skupiny. Prekurzor, ktorý vznikol prekladom kodónov jeho génu predstavuje tzv. *prepropeptid*. Tento prepropeptid začína postupnosťou niekoľkých aminokyselín tvoriacich *signálny peptid*. Podľa signálneho peptidu vie bunka prepropeptid premiestniť do endoplazmatického retikula, kde sa signálny peptid odstráni, a tak vznikne kratší peptid, tzv. *propeptid*. Propeptid pôvodného prekurzoru následne putuje ďalej do Golgiho aparátu, kde pomocou *konvertáz* dochádza na určitých miestach k štiepeniu, a tým vznikne z pôvodného jedného prekurzoru niekoľko kratších peptidov (obr. 1.1) [2]. Týchto konvertáz existuje niekoľko druhov a líšia sa ich špecifickosťou štiepenia. Vo všeobecnosti sa dá povedať, že k štiepeniu dochádza na mieste za bázickou aminokyselinou, konkrétnejšie za aminokyselinou arginín (značka R), no nie po každom výskyte a niekedy i po aminokyseline lyzín (značka K). Jednou z významných konvertáz je enzým furín, ktorého miesta štiepenia sú v súčasnosti pomerne dobre preskúmané. Po štiepení propeptidu na kratšie sekvencie sa niektoré z nich odstránia (budeme ich naďalej nazývať *propeptid*) a ostatné pokračujú ďalej v procese, ktorým z nich vzniknú neuropeptidy. Tento proces štandardne zahŕňa odstránenie bázických aminokyselín na koncoch jednotlivých novovzniknutých peptidov a následne prípadné odstránenie koncovej aminokyseliny glycín za účelom naviazania nejakej chemickej látky (viď obrázok 1.1).

Napríklad u ľudí existuje okolo 90 génov, ktoré kódujú rôzne prekurzory neuropeptidov. Čo sa týka mozgu cicavcov, tak sa v súčasnosti vie o približne 100 rôznych peptidoch, ktoré sú vylučované rôznymi skupinami neurónov [23].



Obr. 1.1: Tvorba neuropeptidov z ich prekurzoru. Najprv sa z prepropeptidu odstráni signálny peptid a následne dochádza na príslušných miestach k štiepeniu zvyšného propeptidu. Čierny úsek na obrázku predstavuje signálny peptid a sivé úseky jednotlivé časti, z ktorých na konci procesu vznikajú neuropeptidy. Biele časti predstavujú úseky, ktoré sú po štiepení odstránené.

Jednou zo základných vlastností prekurzorov neuropeptidov teda je, že obsahujú na začiatku svojej aminokyselinovej sekvencie signálny peptid, ktorý sa v procese tvorby neuropeptidov odštiepi a odstráni. Po úspešnom odstránení signálneho peptidu následne dochádza k štiepeniu propeptidu. Tento jav už bol doteraz mnohokrát študovaný a stále podlieha viacerým skúmaniam. Existuje však konsenzus (Nakayama, 1997), že konvertáza furín zabezpečuje štiepenie na mieste za štvoricou aminokyselín R-X-K/R-R, kde jednotlivé písmená predstavujú aminokyseliny z tabuľky 1.1 a symbol X vyjadruje ľubovoľnú aminokyselinu. Hoci je tento konsenzus pomerne presný, stále nie je úplne zrejmé, kedy za výskytom tejto štvorice k štiepeniu dochádza a kedy nie, nakoľko je to mnohokrát ovplyvňované i aminokyselinami pred i za týmto výskytom a i malá zmena niektorej aminokyseliny v okolí môže štiepenie zmeniť. Dokonca sa zistilo, že v niektorých prípadoch dochádza k štiepeniu i na miestach mimo tohto konsenzu, ako je to napríklad pri organizme mušky *Drosophila*. Na druhej strane, podmienky, na akých miestach k štiepeniu dochádza, sa môžu taktiež líšiť v závislosti od druhu organizmu. V prípade stavovcov dokonca existuje niekoľko proteínov, ktoré majú funkciu slúžiť ako konvertázy neuropeptidov a majú rôznu špecifickosť štiepenia. Štiepenie prekurzoru na konkrétne neuropeptidy teda priamo závisí na aktuálnom zložení bunky, od konvertáz, ktoré práve obsahuje. U stavovcov preto neexistujú všeobecné deterministické pravidlá štiepenia a prakticky ani existovať nemôžu [2].

1.5 Homológia a zarovnania

Vplyvom evolúcie sa DNA môže modifikovať a to zmenou nejakej bázy na inú (substitúcia), vložením novej bázy (inzercia) alebo odstránením niektorej bázy (delécia). Takto teda môže aj z jedného úseku DNA časom vzniknúť viacero jeho modifikácií v rôznych organizmoch, ktoré sa navzájom líšia. Konkrétne z génu, kódujúceho nejaký proteín, môže takto vzniknúť viacero génov, ktoré kódujú líšiace sa proteíny. Proteíny, ktoré vznikli z pôvodne rovnakého génu, a teda majú rovnakého predka, označujeme ako *homologické* alebo *homológy*. Nakoľko tieto proteíny vznikli z jedného rovnakého predka, tak sú zväčša veľmi podobné, čiže ich tvoriace sekvencie aminokyselín sa líšia len málo.

Na porovnanie podobnosti dvoch alebo viacerých sekvencií sa používajú *zarovnania*. Zarovnaním sekvencií sa rozumie vloženie medzier na niektorých miestach v jednotlivých sekvenciách za účelom spárovania podobných úsekov. Na obrázku 1.2 uvádzame ukážku zarovnania viacerých sekvencií aminokyselín, ktoré predstavujú úseky homologických proteínov. Jedným z najrozšírenejších nástrojov na tvorbu zarovnaní je program BLAST [19].

Salamander	`ECSKDCAACTYR-PGLRA-DINPLACTLECEGKLPSSKAWDTCKELLQII`
Myš	`ECSQDCAKCSYR-LVRPG-DINFLACTLECEGQLPSFKIWETCKDLLQVS`
Človek	`ECSQDCATCSYR-LVRPA-DINFLACVMECEGKLPSLKIWETCKELLQLS`
Bombina Orientalis	`DCVSQCFSCSQQIRDSDTIQMNPLACSLECEGSLISTDEWDWCRKILEGD`
Xenopus laevis	`DCVSKCFSCSLQMKALSA-KFNPLVCSLQCEGSLLQDDEWERCGQLLSSQ`

Obr. 1.2: Zarovnanie častí niekoľkých homologických proteínov z rôznych organizmov. Sekvencie aminokyselín v jednotlivých riadkoch predstavujú zarovnávané sekvencie. Pomlčka na niektorom mieste v sekvencii znamená, že u daného organizmu došlo na príslušnom mieste k delécii, respektíve, že v ostatných organizmoch došlo k inzercii. Rôzne aminokyseliny v tom istom stĺpci predstavujú substitúciu na danom mieste.

Kapitola 2

Rozpoznávanie neuropeptidov

Vďaka súčasným moderným technológiám a pokrokom v oblasti sekvenovania sa vybudovala už pomerne veľká databáza genómov rôznych organizmov. Taktiež už existuje niekoľko rôznych programov na hľadanie génov, ktoré sa v jednotlivých genómoch nachádzajú. Týmito pokrokmi sme už teda schopní získať informáciu o tom, aké proteíny si je daný organizmus schopný tvoriť. Problém však nastáva, ak ide len o prekurzor nejakej látky, nakoľko v tomto prípade o danej látke väčšinou stále veľa nevieme. V ďalšom texte sa budeme zaoberať problémom zisťovania, či daný gén kóduje prekurzor nejakého neuropeptidu a taktiež samotným rozpoznávaním neuropeptidov, ktoré sa z neho vyrábajú.

2.1 Problém rozpoznávania

Ako už bolo spomenuté, neuropeptidy majú rôzne dôležité funkcie, a preto je zaujímavé sa nimi zaoberať. Medzi dôležité problémy patrí i otázka, aké neuropeptidy dokáže skúmaný organizmus vytvárať. Jedno z riešení je experimentálne overovanie, ktoré však nie je vždy možné. Dokonca i v prípade, že toto overenie je realizovateľné, ide o veľmi finančne a predovšetkým časovo náročný proces postupného testovania výskytu všetkých možných neuropeptidov [2]. Preto je potrebná i identifikácia prekurzorov z genomických sekvencií, ktorými v súčastnosti už disponujeme. Vďaka takejto identifikácii by sme boli schopní nájsť neuropeptidy i v organizmoch, ktoré sú experimentálne len veľmi málo preskúmané. Zaujímame sa preto o nejaký algoritmus, ktorý by túto otázku vedel zodpovedať bez nutnosti experimentálnych metód v biologickom laboratóriu. Cieľom je zistiť, či daný gén kóduje nejaký prekurzor neuropeptidov a následne rozpoznať, aké konkrétne neuropeptidy sa z neho tvoria. V súvislosti s touto problematikou môžeme rozlišovať tri základné problémy:

Identifikácia prekurzorov neuropeptidov predstavuje problém, pri ktorom máme daný proteín a úlohou je určiť, či ide o prekurzor neuropeptidov alebo nie, pričom

nás nezaujíma, aké neuropeptidy sa z neho vytvárajú. Dve súčasné metódy, ktoré sa pokúšajú tento problém riešiť uvádzame v častiach 2.2.3 a 2.2.4.

Identifikácia miest štiepenia prekurzoru je problém, pri ktorom je daný proteín, o ktorom vieme, že je prekurzorom neuropeptidov a úlohou je zistiť, na ktorých jeho miestach dochádza k štiepeniu na kratšie úseky (viď sekciu 1.4). Aktuálne je za týmto účelom viacerými výskumnými skupinami používaná aplikácia *NeuroPred*, ktorej prístup je bližšie popísaný v časti 2.2.2.

Rozpoznávanie neuropeptidov spája a zároveň rozširuje predchádzajúce dva problémy. Vstupom je proteín, ktorý môže a nemusí byť prekurzorom a úlohou je rozhodnúť, či prekurzorom neuropeptidov je a v tomto prípade taktiež identifikovať signálny peptid, miesta štiepenia a o každom vzniknutom úseku rozhodnúť, či predstavuje úsek, z ktoré vzniká neuropeptid alebo nie. Takémuto popisu jednotlivých častí prekurzoru budeme v nasledujúcom texte hovoriť *anotácia*. Práve tomuto poslednému zadefinovanému problému sme sa rozhodli venovať v našej práci.

2.2 Súčasné metódy

V minulosti bolo vyvinutých niekoľko metód na riešenie problémov spojených s hľadaním neuropeptidov. Autori sa venovali identifikácii prekurzorov, identifikácii miest štiepenia, ale i samotnému problému rozpoznávania neuropeptidov. V nasledujúcej časti uvedieme niektoré z týchto metód, ktoré boli doposiaľ zrealizované.

2.2.1 SignalP

Všeobecnejším problémom, avšak problémom, ktorý súvisí s rozpoznávaním neuropeptidov, je taktiež identifikácia signálneho peptidu. Za týmto účelom bol vyvinutý program *SignalP* [18], ktorý umožňuje identifikovať signálny peptid na začiatku skúmaného proteínu. Ide o natrénovaný model založený na umelých neurónových sieťach, ktorý pre každú pozíciu skúmanej sekvencie vráti skóre určujúce, či sa na danom mieste končí signálny peptid alebo nie. SignalP je pomerne kvalitný nástroj, no jeho predikcie nie sú vždy dokonalé. Existujú napríklad sekvencie, ktoré sú prekurzorom neuropeptidov, a teda aj obsahujú na svojom začiatku signálny peptid, ale program SignalP nevyhodnotí žiadnu pozíciu ako tú, kde signálny peptid končí, konkrétne všetky pozície určí ako málo pravdepodobné. Na druhej strane sme sa pri pozorovaniach stretli i s tým, že v niektorých prípadoch boli dokonca viaceré pozície určené ako miesta, kde signálny peptid môže končiť a to s pomerne vysokými hodnotami skóre. Avšak tento jav nemusí byť vždy dôsledok len nekvalitnej predikcie programu, ale môže mať aj reálne biologické vysvetlenie. Napríklad neuropeptid SIFamid sa medzi organizmami

vyskytuje v niekoľkých podobách, ktoré majú rôzne odstrihnuté začiatky, čo môže byť spôsobené i odštiepením rôznych skrátených foriem signálneho peptidu pred nimi [2]. V tomto prípade program SignalP nevie jednoznačne povedať, kde signálny peptid končí, nakoľko existujú jeho dva varianty, a preto priradí viacerým pozíciám vysoké hodnoty skóre.

2.2.2 NeuroPred

V súčasnosti existuje aplikácia s názvom *NeuroPred* [4], ktorej autori nemali za cieľ riešiť celkovú otázku rozpoznávania neuropeptidov, ale zaoberali sa len samotným problémom identifikácie štiepenia prekurzoru neuropeptidov. Ide o webovú aplikáciu vytvorenú v jazyku Python, v ktorej užívateľ zadá sekvenciu aminokyselín predstavujúcu prekurzor neuropeptidov a ona mu následne nájde miesta, na ktorých s veľkou pravdepodobnosťou dochádza k štiepeniu. Užívateľ preto musí už vopred disponovať informáciou, že skúmaná sekvencia je prekurzorom. Program NeuroPred je aktuálne schopný predpovedať miesta štiepenia na základe viacerých rôznych modelov, ktoré boli vytvorené z dát špecifickej skupiny organizmov. Ak teda užívateľ vie, že skúmaná sekvencia pochádza napríklad z priadky morušovej, vyberie si na predikciu model, ktorý je tomuto organizmu najbližší, v tomto prípade model natrénovaný na dátach hmyzu, nakoľko priadka morušová je hmyzom.

Jeden z modelov je špeciálny tým, že je súhrnom rôznych empirických poznatkov získaných štúdiom miest, na ktorých konvertázy prekurzor štiepia. Ide o súhrn viacerých motívov (postupností aminokyselín), ktoré sa v okolí miest štiepenia bežne vyskytujú. Jedna z takýchto možností už bola spomenutá vyššie v texte v časti 1.4 pre enzým furín. Ak sa teda v okolí skúmaného miesta vyskytuje jeden alebo viacero motívov, ktoré boli pozorované aj v známych sekvenciách, potom sa dané miesto bude považovať za miesto, kde dochádza k štiepeniu. Tento model autori nazvali *Known Motif* model [3] a použili ho pri predikcii miest štiepenia pre hmyz, cicavce, vtáky, ryby a iné živočíchy.

Ostatné modely sú založené na prístupe využívajúcom metódy strojového učenia a boli vytvorené pomocou trénovacích dát konkrétnej skupiny organizmov (napr. mäkkýše, hmyz, cicavce, ľudia). Trénovacie dáta predstavujú známe prekurzory neuropeptidov spolu s informáciou o miestach, na ktorých dochádza k štiepeniu, ktorá bola experimentálne overená.

V článku Southey a kol, 2008 [5] autori opisujú, ako vytvorili modely na predikciu štiepnych miest prekurzorov neuropeptidov hmyzu. Použili pritom štyri rôzne prístupy: Known Motif model, binárnu logistickú regresiu, umelé neurónové siete a metódu k-tich najbližších susedov. Ako trénovacie dáta boli použité experimentálne overené prekurzory z organizmov *Apis mellifera* (včela medonosná) a *Drosophila melanogaster* (octová muška) a vytvorili preto dve trénovacie množiny *Apis* a *Drosophila*.

Ďalej vytvorili dve testovacie množiny dát, ktoré mali s trénovacími množinami

prázdny prienik. Prvá z týchto množín bola testovacia množina s názvom *Various* a obsahovala kompletné prekurzory hmyzu rôznych druhov. Druhá, *Insuline-like*, bola tvorená prekurzormi peptidov podobných inzulínu. Táto druhá množina bola vytvorená preto, lebo sekvencie týchto prekurzorov si sú navzájom veľmi podobné, dokonca i naprieč organizmami. Taktiež neobsahovala žiadne homológy prekurzorov v množinách *Apis*, *Drosophila* ani *Various*. Poslednou vlastnosťou týchto množín bolo, že navzájom medzi sebou neobsahovali žiadnu rovnakú sekvenciu prekurzorov pochádzajúcu z toho istého organizmu [5].

Následne prešiel každý prekurzor v jednotlivých množinách procesom, ktorým sa získali dáta potrebné na natrénovanie jednotlivých modelov. Tento proces pozostával jednak z predikcie signálneho peptidu pomocou programu SignalP, ktorý bol taktiež odstránený, a následnej tvorby dát zo zostávajúcej sekvencie. Okolo všetkých potenciálnych miest štiepenia, ktoré obsahovali arginín alebo lyzín, boli vytvorené okná veľkosti $2k$. Ak však išlo o štiepenie na mieste za niekoľkými za sebou idúcimi aminokyselinami arginín alebo lyzín, potom bolo miesto štiepenia priradené tej najviac vpravo. Okná sa teda mohli prekrývať a všetky boli rozdelené na základe známych poznatkov medzi pozitívne príklady, ak išlo o okno, v ktorom dochádza k štiepeniu, a negatívne príklady, ak v nich k štiepeniu nedochádzalo.

Model binárnej logistickej regresie, umelých neurónových sietí a k-tich najbližších susedov boli následne natrénované na množine dát *Apis* a *Drosophila*, pričom každá bola analyzovaná samostatne, a tým sa získali dva rôzne modely. Ako model logistickej regresie bola použitá kombinácia metód backward, forward a stepwise model-selection. Umelé neurónové siete boli implementované ako viacvrstvový perceptrón s jednou skrytou vrstvou pozostávajúcou z 500 vrcholov. Model k-tich najbližších susedov bol realizovaný pomocou algoritmu k-tich najbližších susedov s euklidovskou metrikou a za účelom získania optimálnej hodnoty k boli testované jej viaceré hodnoty.

Následne boli tieto modely testované na všetkých množinách dát a navzájom porovnané. Ak bola pravdepodobnosť štiepenia na danej pozícii menšia ako 0,5, potom sa považovala za neštiepnu (*negative*) a naopak, ak bola táto pravdepodobnosť väčšia alebo rovná 0,5, potom sa daná pozícia prehlásila za miesto štiepenia (*positive*). Vzhľadom na autormi uvažované pozorovania dosahovali všetky vytvorené modely porovnateľné výsledky. Čo sa týka Known Motif modelu, tak spomedzi všetkých modelov dosahoval celkovo najlepšie hodnoty počtu správne predikovaných pozícií štiepenia pre všetky množiny dát. Na druhej strane taktiež dosahoval vysoké hodnoty nesprávne predikovaných pozícií štiepenia, čo sa zobrazilo na nízkej pozitívnej presnosti, keďže zo všetkých predikovaných pozícií štiepenia bolo až 30-50% nesprávnych. Tieto vlastnosti Known Motif modelu mali za dôsledok, že celkovo, až na senzitivitu, dosahoval najhoršie výsledky vo všetkých autormi uvažovaných pozorovaniach.

2.2.3 Identifikácia prekurzorov

Ako bolo spomenuté v texte vyššie, samotný program NeuroPred nerieši problém zistenia, či skúmaný proteín je prekurzorom neuropeptidov alebo nie, ale už to vopred predpokladá. Tomuto postupu preto predchádza nejaký iný, ktorý by vedel túto informáciu zistiť. Týmto problémom sa venovali aj autori programu NeuroPred v článku od Southey a kol. [6], pričom skúmali opicu makak (*Macaca mulatta*). Tento organizmus si vybrali preto, lebo hoci sa často používa ako modelový organizmus, napriek tomu boli dovtedy známe len štyri gény prekurzorov, pričom ľudských je známych vyše 90.

Ich postup pozostáva z dvoch základných krokov, pričom v prvom využili dva známe poznatky. Jednak to, že pre niektoré cicavce už existuje rozsiahlejšia databáza prekurzorov neuropeptidov a taktiež, že v súčasnosti disponujeme pomerne rozsiahlou databázou génov, resp. proteínov makaka, ktoré sme získali ako výsledok programov na predikciu génov. Z dôvodu relatívne blízkej evolučnej príbuznosti použili celé prekurzory neuropeptidov u človeka[1] a každý zarovnali so všetkými proteínmi makaka. Na získanie zarovnaní použili štandardný program BLAST [19].

Následne boli v druhom kroku všetky relevantné výskyty zarovnané k zodpovedajúcim sekvenciám viacerých iných organizmov. Pre jednotlivé potenciálne prekurzory teda vytvorili viacnásobné zarovnanie (program T-Coffee [20]) a tie následne ručne preskúmali. V databáze predikovaných proteínov makaka takto našli 67 prekurzorov neuropeptidov.

Takáto identifikácia prekurzorov z už známych predikovaných proteínov, ale zlyháva v situácii, keď bol niektorý proteín predikovaný neúplne alebo nebol predikovaný vôbec. Je preto potrebné zistiť, či sa nejaký konkrétny prekurzor u človeka vyskytuje i v genóme makaka. Na takéto účely sa používa program Wise2 [21], ktorý predikuje štruktúru génu pomocou porovnávania sekvencie proteínu s DNA sekvenciou a používa pritom i model na predikciu génov. Týmto spôsobom bolo nájdených ďalších 17 prekurzorov. Táto kombinácia použitia programov BLAST a Wise2 sa preto ukázala byť dobrá, lebo každý našiel i také prekurzory, ktoré sa tomu druhému nájsť nepodarilo.

Týmto postupom boli autori schopní identifikovať proteíny, ktoré sú prekurzormi neuropeptidov a na tie následne nasadiť program NeuroPred, ktorý bol schopný nájsť miesta štiepenia, prípadne boli tieto miesta predikované už len na základe zarovnaní, pomocou ktorých boli získané. Napriek tomuto pomerne komplikovanému postupu je však dôležité si uvedomiť, že týmto postupom autori Southey a kol. [6] identifikovali len také prekurzory, pre ktoré v ľudskom genóme existuje homológ. Neumožňuje teda nájsť nejaký iný alebo nový prekurzor. Tento postup tiež nie je možné uplatniť na genómy, ktoré nemajú blízko príbuzný genóm s už nájdenými neuropeptidmi.

[1] Úsek so signálnym peptidom a jednotlivé neuropeptidy

2.2.4 NeuroPID

Iným postupom, ktorý rieši problém identifikácie prekurzorov neuropeptidov, sa zaoberali Ofer a kol. [7]. Vytvorili nástroj *NeuroPID*, pomocou ktorého sa pokúšali identifikovať prekurzory neuropeptidov u cicavcov, pričom ich postup nebol obmedzený len na už doposiaľ známe prekurzory ako tomu bolo v prípade autorov NeuroPred-u. Ide o tradičnú schému strojového učenia, v ktorej si vytvorili niekoľko atribútov (angl. features) opisujúcich sekvenciu aminokyselín, podľa ktorých sa následne rozhodne, či ide o sekvenciu prekurzoru neuropeptidov alebo nie. Vytvorili pomerne veľký zoznam obsahujúci takmer 600 rôznych atribútov, ktoré použili na popis sekvencie. Logicky ich možno rozdeliť do troch základných skupín:

- Biofyzikálne kvantitatívne vlastnosti
  - molekulová hmotnosť, dĺžka, isoelektrický bod, bigramy aminokyselín, výskyt aminokyselín s nábojom, výskyt aromatických aminokyselín a iné

- Binárne atribúty
  - zachytávajú informáciu o nie náhodnom rozložení konkrétnych skupín aminokyselín v oknách veľkosti 5. Pre všetky binárne možnosti okna ($2^5 = 32$) sa zráta počet výskytov tohto okna v sekvencii, pričom na mieste jednotiek sa nachádzajú aminokyseliny z danej skupiny. Takto sa vytvorilo 32 atribútov pre skupiny $[G, K, R]$, $[K, R]$, aminokyseliny s nábojom a iné.

- Informačné štatistiky
  - rôzne štatistiky motivované oblasťou vyhľadávania informácií (angl. information retrieval) ako entropia aminokyselín a autokorelácia

Po zadefinovaní jednotlivých atribútov autori vytvorili trénovacie dáta, pozostávajúce zo známych prekurzorov neuropeptidov a negatívnych sekvencií, t. j. proteínov, o ktorých sa vie, že nie sú prekurzorom neuropeptidov. Na týchto dátach vykonali Kolmogorov-Smirnov test, čím zistili, ktoré z ich atribútov sú štatisticky najviac informatívne pri probléme identifikácie prekurzorov neuropeptidov. Takto napríklad zistili, že počet výskytov aromatických aminokyselín zohráva pomerne významnú úlohu. Konkrétne, že sekvencie, ktoré sú prekurzormi neuropeptidov majú zväčša zvýšený výskyt aromatických aminokyselín.

Trénovacie dáta potom zakódovali do všetkých uvažovaných atribútov, ktoré ich popisujú a následne na nich natrénovali viaceré modely strojového učenia ako *support vector machines* (SVM), *gradient boosting* alebo *random forests*. Všetky tieto modely viedli k porovnateľným výsledkom a to 82-89% správnej klasifikácie pri krížovej validácii (angl. cross-validation). Nakoľko však použili veľmi veľké množstvo atribútov, rozhodli sa vynechať také atribúty, ktoré neprinášajú dostatok informácií potrebných

na klasifikáciu. Realizovali to postupným vynechávaním atribútov a pozorovaním úspešnosti klasifikácie, a taktiež pomocou metódy analýzy hlavných komponentov (angl. principal component analysis). Týmto postupom sa im podarilo znížiť počet vysoko informatívnych atribútov z pôvodných 561 až na 23, pričom sa zachovala stále pomerne dobrá miera úspešnosti.

2.3 Návrh systému

V texte vyššie boli spomenuté niektoré súčasné prístupy riešenia problému rozpoznávania neuropeptidov. Program NeuroPred [4] sám o sebe tento problém nerieši, ale aktuálne sa stal akýmsi štandardom, ktorý vo väčšej alebo menšej časti využívajú aj viaceré iné metódy ako napríklad práce autorov Ofer a kol. (2014) [7], Delfino a kol. (2010) [8], Clynen a kol. (2010) [9] alebo Xie a kol. (2010) [10]. Použitie programu NeuroPred je v skutočnosti možné len ak o skúmanej sekvencii vieme, že je prekurzorom neuropeptidov. Dokonca i v tom prípade, keď touto informáciou disponujeme, stále môžu nastať komplikácie, nakoľko musíme dopredu poznať pozíciu, resp. ukončenie signálneho peptidu. Ako už bolo spomenuté v časti 2.2.1, program SignalP túto čiastkovú úlohu rieši, avšak aj tak nie úplne dokonalo. Ak teda nevieme presne určiť, kde sa v nami skúmanej sekvencii končí signálny peptid alebo takýchto miest máme dokonca viacero, potom aj miesta štiepenia predikované programom NeuroPred nemusia byť úplne správne. Môže ich byť viac alebo menej, vzhľadom na chybné skrátenie alebo predĺženie potenciálneho signálneho peptidu, ale môžu byť niektoré miesta aj úplne pozmenené. Nepresnosť predikcií signálneho peptidu však nie sú jedinou slabosťou NeuroPred-u. Hoci nám je schopný určiť miesta potenciálneho štiepenia prekurzoru na menšie časti, neurčuje však, ktoré z týchto častí následne v bunke prebehnú sériou modifikácií a budú z nich vytvorené neuropeptidy. Získame teda zoznam menších úsekov, no nevieme im priradiť ich biologickú vlastnosť, a preto sme v princípe úlohu nájdenia neuropeptidov nevyriešili.

Ako bolo spomenuté, už samotné zistenie, či je skúmaná sekvencia prekurzorom, je dôležitý problém, ktorému sa vedci venujú aj nezávisle od predikcie miest štiepenia alebo neuropeptidov samotných. V našej práci sme sa rozhodli nevenovať týmto problémom oddelene, ale vnímame ich ako jeden väčší problém, ktorý by sa mal riešiť jednotne. Poznatok, že skúmaná sekvencia je prekurzorom nám totiž dodáva informáciu o tom, že ide o prekurzor neuropeptidov, a teda, že sa štiepi na menšie časti. Na druhej strane, ak sa sekvencia štiepi na menšie časti, potom nám to pridáva na istote, že je naozaj prekurzorom. Navrhli sme preto systém, ktorého úlohou je o skúmanej sekvencii zozbierať viaceré informácie, ktoré môžu byť pri probléme rozpoznávania neuropeptidov relevantné, tieto informácie následne spracovať a vyhodnotiť metódami strojového učenia. Informácie uvažované pri návrhu nášho systému môžeme rozdeliť do štyroch

základných skupín:

- *Informácia o signálnom peptide*
  Každej pozícii v skúmanej sekvencii chceme priradiť určité skóre, ktoré hovorí o tom, či sa na danom mieste končí signálny peptid alebo nie. Ide teda o funkciu, od ktorej vyžadujeme, aby v ideálnom prípade priradila koncovej pozícii signálneho peptidu vysokú hodnotu a ostatným pozíciám hodnotu nízku. Toto priradenie hodnôt vieme dosiahnuť použitím programu SignalP.

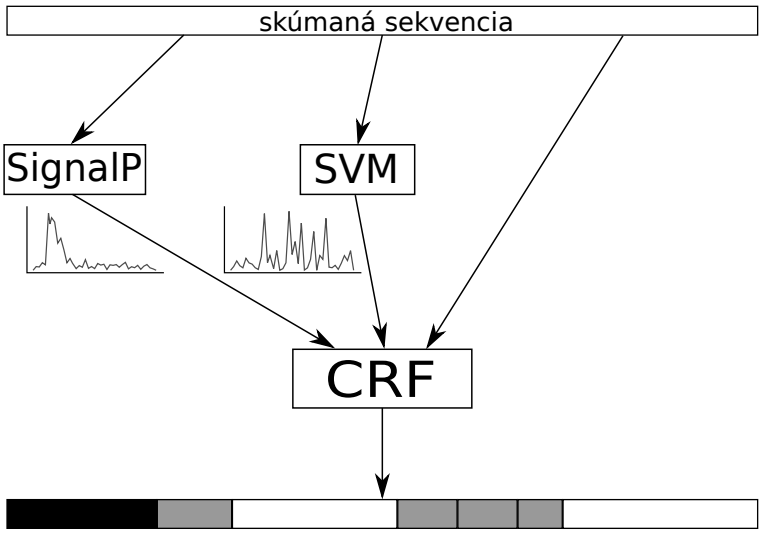
- *Informácia o miestach štiepenia*
  Každej pozícii v skúmanej sekvencii chceme priradiť skóre hovoriace ako veľmi je pravdepodobné, že za danou pozíciou dochádza k štiepeniu. Toto vieme dosiahnuť aj využitím dostupnej aplikácie NeuroPred, no nevýhodou je jednak to, že ide výhradne o webovú aplikáciu a i fakt, že pri používaní si máme možnosť vybrať len z veľmi obmedzeného počtu natrénovaných modelov. Rozhodli sme sa preto implementovať vlastný nástroj určený na predikciu miest štiepenia, ktorý je založený na podobnom princípe ako NeuroPred. Výhodou je, že užívateľ si je schopný vopred natrénovať svoj vlastný model na známych, dobre anotovaných sekvenciách a následne ho použiť na predikciu štiepnych miest skúmanej sekvencie. Na vytvorenie modelu sme použili metódu strojového učenia nazývanú *support vector machines* [11] (SVM), ktorá bude bližšie opísaná v nasledujúcej kapitole.

- *Informácia o dĺžke*
  Dĺžky úsekov prekurzoru, ktoré predstavujú neuropeptidy sú vo všeobecnosti kratšie ako propeptidy. Informácia o dĺžke uvažovaného úseku preto môže istým spôsobom podávať informáciu o tom, či je daný úsek neuropeptid alebo nie. Taktiež signálny peptid zvykne mať istú štandardnú dĺžku. Tieto informácie boli využité aj v našom systéme a ich využitie je bližšie popísané v časti 4.2.2.

- *Štatistické informácie získané zo sekvencie*
  Do tejto skupiny patria všetky ostatné informácie, ktoré sa dajú získať priamo zo sekvencie. V našich experimentoch sme použili niekoľko z týchto možností, ktoré sú bližšie opísané v kapitole 4.2.2, ale vo všeobecnosti je možné doplnenie nášho systému aj o ďalšie informácie.

Získané informácie o sekvencii sú následne spracované a na základe nich je skúmaná sekvencia anotovaná. Na anotáciu sme sa rozhodli použiť metódu strojového učenia s názvom *conditional random fields* [13] (CRF) určenej na segmentáciu a anotáciu sekvenčných dát. Ide o pravdepodobnostný model, ktorý vo viacerých aspektoch prekonáva doposiaľ známe iné modely ako napríklad skryté markovovské modely (angl. hidden Markov models), markovovské modely maximálnej entropie alebo stochastické

gramatiky. Navyše ide o model, ktorý sa veľmi hodí na nami navrhnutý systém, nakoľko ho vieme vytvoriť presne tak, aby dokázal spracovávať nami navrhnuté informácie o sekvencii a na základe nich ju anotovať. Obrázok 2.1 schematicky znázorňuje priebeh nami navrhnutého systému. V prvom kroku sa o skúmanej sekvencii zozbierajú jednotlivé informácie. Menovite informácia o signálnom peptide pomocou programu SignalP, informácia o miestach štiepenia pomocou dopredu natrénovaného modelu SVM a ostatné informácie získané priamo zo sekvencie. Tieto informácie sú následne využité prostredníctvom modelu CRF, ktorý bol predom natrénovaný na známych správne anotovaných sekvenciách. V zásade rozlišujeme tri základné anotácie a to *signal*, *neuropeptid* a *propeptid*. Avšak nakoľko má byť náš systém schopný správne pracovať aj bez informácie, či je skúmaná sekvencia naozaj prekurzorom neuropeptidov, je nutné uvažovať aj ďalšie dodatočné anotácie, podľa ktorých bude možné zistiť, že skúmaná sekvencia prekurzorom neuropeptidov nie je. Týmto problémom sa však budeme podrobnejšie venovať až v kapitole 4 spolu s celkovým popisom anotácie.



Obr. 2.1: Schéma ilustrujúca nami navrhnutý systém na rozpoznávanie neuropeptidov. Zo skúmanej sekvencie sú extrahované informácie viacerých druhov a v konečnom štádiu je sekvencia rozdelená na menšie anotované úseky pomocou pravdepodobnostného modelu conditional random fields (CRF). Úsek zvýraznený čiernou farbou predstavuje predikovaný signálny peptid, sivé úseky predstavujú jednotlive neuropeptidy a biele úseky označujú nekódujúce časti, t. j. propeptidy.

Kapitola 3

Rozpoznávanie miest štiepenia

V predchádzajúcej kapitole sme v časti 2.3 predstavili nami navrhnutý systém pozostávajúci z niekoľkých častí. V tejto kapitole sa budeme venovať časti, ktorá má na starosti predikciu potenciálnych miest štiepenia sekvencie, ktorá je na obrázku 2.1 označená skratkou SVM. V nasledujúcom texte najprv predstavíme metódu strojového učenia *support vector machines* [11], ktorej táto skratka prislúcha, a následne popíšeme ako bola táto metóda v našom systéme využitá za účelom predikcie miest štiepenia. Nami navrhnutý postup trénovania predikcie štiepenia taktiež porovnáme s predikciami dostupnej webovej aplikácie NeuroPred [4].

3.1 Úvod do SVM

Support vector machines [11] (SVM) je metóda strojového učenia, ktorá sa používa pri riešení problému klasifikácie i regresie, no bližšie sa budeme venovať len binárnej klasifikácii, teda klasifikácii dát do dvoch tried.
Nech $\mathcal{D} = \{(\mathbf{x_i}, y_i)|x_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\}_{i=1}^n$ predstavuje trénovaciu množinu dát pozostávajúcu z n bodov v p rozmernom priestore, pričom každý bod je priradený do jednej z dvoch tried, -1 alebo 1.

3.1.1 Lineárna klasifikácia

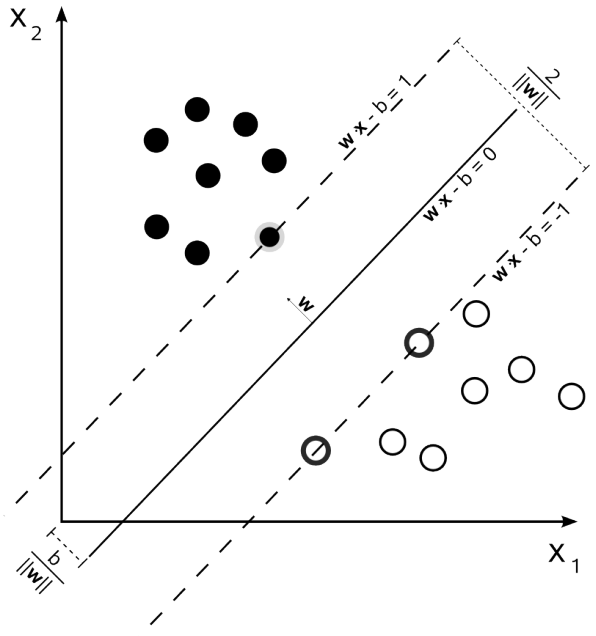
V prípade lineárnej SVM metódy sa hľadá nadrovina v p-rozmernom priestore, ktorá separuje body $\mathbf{x_i}$ v trénovacej množine $\mathcal{D}$ tak, že body patriace do jednej triedy ležia na jednej strane nadroviny a body patriace do druhej triedy na jej druhej strane. Navyše sa hľadá optimálna separácia, teda nadrovina, ktorá má čo najväčšiu vzdialenosť k nej najbližšiemu bodu. Každú nadrovinu priestoru $\mathbb{R}^p$ možno zapísať ako množinu bodov $\mathbf{x}$ spĺňajúcich rovnicu

$$\mathbf{w} \cdot \mathbf{x} - b = 0,$$

kde $\cdot$ predstavuje skalárny súčin a $\mathbf{w}$ je normálový vektor danej nadroviny. Hodnota $\frac{b}{||\mathbf{w}||}$ určuje vzdialenosť nadroviny od začiatku súradnicového systému, t. j. bodu $\mathbf{0}$. Ak hľadáme nadrovinu, ktorá rozdeľuje body trénovacej množiny optimálne, treba preto zobrať dve navzájom rovnobežné nadroviny, ktoré dané body rozdeľujú a zároveň požadovať, aby vzdialenosť medzi nimi bola čo najväčšia. Tieto dve nadroviny možno reprezentovať ako $\mathbf{w} \cdot \mathbf{x} - b = 1$ a $\mathbf{w} \cdot \mathbf{x} - b = -1$. Obrázok 3.1 ilustruje danú situáciu, pričom je na ňom znázornená jednak samotná separujúca nadrovina, ale i táto dvojica nadrovín, tvoriaca tzv. *hranicu*. Snažíme sa teda maximalizovať vzdialenosť medzi týmito dvomi nadrovinami, ktorá je $\frac{2}{||\mathbf{w}||}$, t. j. minimalizovať $||\mathbf{w}||$, pričom musí stále platiť, že obe rozdeľujú body trénovacej množiny. Tieto podmienky možno zjednodušene zapísať v tvare

$$y_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1,$$

kde $(\mathbf{x_i}, y_i)$ sú jednotlivé body trénovacej množiny $\mathcal{D}$. Metóda lineárnej SVM minimalizuje hodnotu $||\mathbf{w}||$ pri dodržaní týchto podmienok využitím bodov ležiacich na jednej z dvoch separujúcich nadrovín (hraníc), nakoľko len nimi sa má zmysel zaoberať, a tieto body nazýva podporné vektory (angl. support vectors).



Obr. 3.1: Optimálne rozdelenie bodov v dvojrozmernom priestore do dvoch tried. Plná čiara predstavuje nadrovinu, ktorá dané body rozdeľuje tak, že body jednotlivých tried ležia na samostatnej strane nadroviny. Prerušované čiary predstavujú hranicu danej separácie a body na nich sú tzv. podporné vektory. (Zdroj www.wikipedia.org)

Nakoľko ale nie vždy existuje taká nadrovina, ktorá by rozdeľovala všetky body $\mathbf{x_i}$ správne, môžeme dovoliť len jemnú hranicu, a teda, že nemusia byť body dokonale separované. V tomto prípade sa uvažuje nejaká konkrétna penalizácia zlého priradenia do triedy, pričom ale samotný algoritmus SVM sa naďalej pokúša maximalizovať vzdia-

lenosť separujúcej nadroviny a najbližšieho správne klasifikovaného bodu. Hľadanie optimálnej nadroviny je preto hľadanie akejsi rovnováhy medzi čo najväčšou hranicou a čo najmenšou chybou, vzhľadom na zvolenú penalizáciu jednotlivých bodov.

3.1.2 Nelineárna klasifikácia

Ak sú body z $\mathcal{D}$ lineárne separovateľné, tak lineárne SVM nájde ich optimálnu separáciu. Ak ale lineárne separovateľné nie sú, potom sa môže použiť metóda, ktorá dané body transformuje do nejakého viacrozmerného priestoru, v ktorom už následne lineárne separovateľné byť môžu. Táto transformácia pochopiteľne nemôže byť lineárna, nakoľko by body ostali naďalej lineárne neseparovateľné. Tento postup sa však takto priamo nepoužíva, ale využíva sa tzv. *jadrový trik* (angl. kernel trick). Celý výpočet lineárnej SVM sa dá totiž vykonať tak, že sa body $\mathbf{x_i}$ vo výpočte vyskytujú vždy len v nejakom skalárnom súčine s iným bodom $\mathbf{x_j}$ alebo samým sebou. Ak by sme preto chceli použiť spomínanú metódu, ktorá transformuje body do nejakého viac rozmerného priestoru prostredníctvom nelineárnej transformácie $\phi(\mathbf{x})$, všade vo výpočte sa budú vyskytovať len skalárne súčiny $\phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j})$, ktoré môžeme označiť ako $k(\mathbf{x_i}, \mathbf{x_i})$ a nazývať *jadro*. Spomínaný jadrový trik teda spočíva v tom, že sa vo výpočte SVM namiesto skalárnych súčinov jednotlivých bodov $\mathbf{x_i}, \mathbf{x_i}$ vždy použije jadrová funkcia $k(\mathbf{x_i}, \mathbf{x_i})$, ktorá predstavuje lineárny súčin daných bodov po transformácii do nejakého viacrozmerného priestoru. Tento jadrový trik sa taktiež použije nielen pri trénovaní, ale i následnom použití SVM za účelom klasifikácie, kde sa tiež vykonávajú len súčiny $\phi(\mathbf{x}) \cdot \phi(\mathbf{x_i})$ vstupných dát $\mathbf{x}$ s podpornými vektormi $\mathbf{x_i}$.

V praxi sa používa viacero jadrových funkcií, no spomenieme len tzv. *radial basis function* (RBF), ktorá je definovaná ako

$$k(\mathbf{x_i}, \mathbf{x_i}) = \exp(-\gamma ||\mathbf{x_i} - \mathbf{x_i}||^2),$$

kde γ je parameter danej funkcie. Výhodou použitia takejto funkcie je fakt, že RBF predstavuje skalárny súčin v určitom nekonečno-rozmernom priestore a kladie len minimálne podmienky na dáta. Táto funkcia navyše zabezpečuje pomerne veľkú presnosť a pri trénovaní taktiež rýchlu konvergenciu k optimálnemu riešeniu [7]. Výhodou je taktiež jej malý počet parametrov.

3.2 Predikcia miest štiepenia

Ako už bolo spomenuté v predchádzajúcom texte, v našom systéme sme použili metódu SVM na klasifikáciu miest štiepenia sekvencií na menšie časti. Na riešenie tohto problému sme použili prístup podobný prístupu autorov programu NeuroPred [4]. Pomocou trénovacích dát sme vytvorili SVM umožňujúce klasifikáciu jednotlivých pozícií

v skúmanej sekvencii do dvoch tried podľa toho, či na danej pozícii (resp. za ňou) dochádza k štiepeniu.

3.2.1 Tvorba dát potrebných na klasifikáciu

Ako bolo spomenuté v predchádzajúcom texte, model určený na predikciu miest štiepenia má byť vytvorený z užívateľom vybraných trénovacích dát. Tieto trénovacie dáta pozostávajú zo sekvencií, ktoré sú anotované informáciou o štiepení. Teda každej pozícii v sekvencii aminokyselín je priradená anotácia, či za danou aminokyselinou v sekvencii dochádza k štiepeniu alebo nie. Na to, aby sme vedeli použiť metódu klasifikácie miest štiepenia pomocou SVM, je nutné z trénovacích dát vytvoriť dáta požadovaného typu, teda body v niekoľko-rozmernom priestore, ktoré sú zaradené do dvoch rôznych tried.

Jednotlivé aminokyseliny reprezentujeme ako bázové vektory 20-rozmerného priestoru $\mathbb{R}^{20}$, nakoľko ich je presne 20 (viď tabuľku 1.1), teda alanín ako vektor majúci prvú súradnicu jednotkovú a ostatné nulové, cysteín majúci druhú súradnicu jednotkovú a ostatné nulové atď. Sekvenciu pozostávajúcu z k aminokyselín v našom prístupe reprezentujeme vektorom v $20k$-rozmernom priestore $\mathbb{R}^{20k}$, ktorý vznikol karteziánskym súčinom vektorov reprezentujúcich jednotlivé aminokyseliny na prvej až k-tej pozícii. V našom prípade sme z jednej sekvencie aminokyselín dĺžky n vytvorili n vektorov prislúchajúcich k jednotlivým pozíciám. Ak by sme neuvažovali žiadne okolie (teda okno veľkosti 1), získali by sme tak n vektorov v 20-rozmernom priestore, teda každý vektor by bol vektorom aminokyseliny na danej pozícii. My však uvažujeme aj okolie veľkosti k naľavo i napravo od potenciálneho miesta štiepenia (teda okno veľkosti $2k+1$), a preto dostávame vektory v $20 * (2k+1)$-rozmernom priestore. Aby sme týmto postupom dostali všetky vektory rovnakej dimenzie, v prípade krajných pozícií sme okno doplnili o neurčené aminokyseliny X, ktoré boli následne reprezentované nulovým vektorom, t. j. $\mathbf{0} \in \mathbb{R}^{20}$.

Počiatočné trénovacie dáta boli anotované informáciou o miestach štiepenia, a preto jednotlivé okná vieme rozdeliť do dvoch tried podľa toho, či predstavujú okno so stredom v mieste štiepenia alebo nie. Týmto postupom sme získali dáta vhodné na klasifikáciu pomocou metódy SVM. Avšak, ako bolo už spomenuté v časti 1.4, vieme, že k štiepeniu dochádza len za aminokyselinami lyzín a arginín, a preto len vektory zodpovedajúce štiepeniam na týchto miestach boli použité pri trénovaní modelu. Podobne v prípade samotnej predikcie miest štiepenia pomocou natrénovaného modelu, pozície, na ktorých je iná aminokyselina než lyzín alebo arginín môžu byť automaticky považované za nie štiepne. Pre ostatné pozície sa následne vytvoria korešpondujúce okná a k nim zodpovedajúce vektory a tie môžu byť na základe klasifikácie pomocou natrénovaného SVM modelu anotované ako štiepne alebo nie. Konkrétne, každej takejto pozícii SVM priradí pravdepodobnosť s akou je štiepna, a teda nami zvolenou hranicou vieme nastaviť citlivosť anotácie štiepenia.

3.2.2 Trénovanie modelu

Po získaní anotovaných dát vo formáte požadovanom SVM, je možné samotné trénovanie modelu. V našej práci sme sa rozhodli použiť štandardne používanú knižnicu LibSVM [12] a ako jadrovú funkciu zvoliť RBF, ktorej výhody boli opísané v časti 3.1. Tento postup zahŕňa použitie dvoch hyper-parametrov γ a C. Prvý z nich predstavuje parameter v definícii jadrovej funkcie RBF a druhý penalizáciu zlého priradenia do triedy. Za účelom trénovania modelu s čo najlepšou presnosťou sme zvolili postup nájdenia ideálnych hodnôt hyper-parametrov prehľadávaním viacerých možností. Použili sme postup krížovej validácie (angl. cross-validation), pri ktorom sme rozdelili trénovacie dáta na päť častí, na štyroch z nich bol model trénovaný s konkrétnym nastavením γ a C a na piatej bol validovaný. Teda pre každú kandidátsku dvojicu (γ, C) sme natrénovali 5 rôznych modelov, ktoré boli vždy trénované i validované na iných dátach a následne bola každej tejto dvojici nastavení priradená hodnota úspešnosti klasifikácie v podobe priemernej presnosti klasifikácie všetkých piatich modelov. Finálny model bol napokon natrénovaný na celých trénovacích dátach použitím tých nastavení parametrov γ a C, ktoré dosiahli v predchádzajúcich validáciách najlepšiu úspešnosť.

3.3 Experimenty

V predchádzajúcom texte sme opísali ako využiť model SVM na predikciu miest štiepenia sekvencií aminokyselín. Hoci je tento krok len čiastkovým krokom nami navrhnutého systému na predikciu neuropeptidov opísaného v sekcii 2.3, testovali sme presnosť predikcie porovnaním s aktuálne dostupnou webovou aplikáciou NeuroPred [4].

Porovnávali sme výsledky modelu *Apis* aplikácie NeuroPred, ktorý vznikol natrénovaním na anotovaných 16-tich prekurzoroch neuropeptidov získaných z genómu včely medonosnej, ktoré sú voľne dostupné. Tieto sekvencie spolu celkovo obsahujú 70 výskytov štiepenia z celkového počtu 2931 pozícií [1]. Na tých istých sekvenciách bol trénovaný i náš model postupom, ktorý bol podrobne opísaný v predchádzajúcom texte, pričom sme použili rôzne veľkosti okien. Ako testovacie dáta sme zvolili sekvencie *Drosophila*, ktoré boli už spomínané v časti 2.2.2, pozostávajúce z 21 prekurzorov neuropeptidov získaných z genómu octovej mušky, obsahujúcich 87 výskytov štiepenia z celkového počtu 3948 pozícií. Tieto dáta sme použili pri klasifikácii miest štiepenia aplikáciou NeuroPred a nášho modelu a následne sme ich predikcie navzájom porovnali. Za účelom merania úspešnosti klasifikácie sme uvažovali nasledovné počty:

true positive (TP): počet správne predikovaných pozícií štiepenia

false positive (FP): počet nesprávne predikovaných pozícií štiepenia

[1] Súčet dĺžok všetkých 16-tich sekvencií *Apis*

true negative (TN): počet správne predikovaných pozícií, na ktorých nedochádza k štiepeniu

false negative (FN): počet nesprávne predikovaných pozícií, na ktorých nedochádza k štiepeniu

positive (P): celkový počet pozícií štiepenia, t. j. TP+FN

negative (N): celkový počet pozícií, na ktorých nedochádza k štiepeniu, t. j. TN+FP

na základe ktorých sú vyjadriteľné štandardne používané štatistiky, použité v našich meraniach:

Presnosť klasifikácie: pomer počtu pozícií so správnou predikciou ku všetkým predikciám, t. j. (TP+TN)/(P+N)

Senzitivita: pomer počtu TP k celkovému počtu pozícií štiepenia, t. j. TP/P

Špecificita: pomer počtu TN k celkovému počtu pozícií, na ktorých nedochádza k štiepeniu, t. j. TN/N

Pozitívna presnosť: pomer počtu TP k celkovému počtu predikovaných pozícií štiepenia, t. j. TP/(TP+FP)

Negatívna presnosť: pomer počtu TN k celkovému počtu predikovaných pozícií, na ktorých nedochádza k štiepeniu, t. j. TN/(TN+FN)

Na základe viacerých experimentov, vykonaných taktiež na iných dátach, sme sa rozhodli pri tvorbe dát na klasifikáciu použiť okná veľkosti 11, teda okrem aminokyseliny na danej pozícii sa berie do úvahy aj 5 aminokyselín pred a za danou pozíciou. Táto veľkosť sa ukázala byť dostatočná na generalizáciu problému, pričom sa stále dosahovali dostatočne dobré výsledky presnosti klasifikácie.

V tabuľke 3.1 uvádzame porovnanie predikcií nami natrénovaného modelu s modelom *Apis* aplikácie NeuroPred. Z tohto porovnania vidno, že nami natrénovaný model dosahoval porovnateľné výsledky, pričom dosiahol o niečo lepšie hodnoty presnosti klasifikácie, špecificity i pozitívnej presnosti. Uvedené hodnoty boli získané použitím základnej hraničnej hodnoty klasifikácie, a teda miesto v sekvencii je považované za štiepne, ak je jeho príslušnosť (pravdepodobnosť priradenia) do triedy štiepnych miest väčšia ako do neštiepnych. Zmena tejto hraničnej hodnoty môže viesť i k lepším výsledkom, no tie neboli v našich experimentoch porovnávané. Avšak nami navrhnutý systém na predikciu neuropeptidov, ktorý výsledky tejto klasifikácie len využíva (viď sekciu 2.3), nepracuje s týmto pevným priradením do tried na základe konkrétnej hodnoty, ale s jemným percentuálnym priradením.

	NeuroPred	Náš model
Presnosť klasifikácie	90,75%	91,14%
Senzitivita	63,22%	62,07%
Špecificita	96,30%	96,99%
Pozitívna presnosť	77,46%	80,60%
Negatívna presnosť	92,86%	92,70%

Tabuľka 3.1: Porovnanie natrénovaného modelu s aplikáciou NeuroPred. Ako trénovacie dáta boli zvolené sekvencie z množiny *Apis* popísané v článku autorov Southey a kol. [5]. Jednotlivé modely boli testované na množine sekvencií *Drosophila* popísanej v tom istom článku.

Z výsledkov tohto experimentu, ale i iných, ktoré boli vykonané, môžeme povedať, že náš postup trénovania modelu na predikciu miest štiepenia vedie k podobným výsledkom ako aplikácia NeuroPred. Podotýkame však, že naším cieľom nebolo dosiahnuť výrazne lepšie výsledky, ale len porovnateľné, nakoľko samotný problém predikcie miest štiepenia riešený týmto postupom predstavuje len súčasť nami navrhnutého systému na rozpoznávanie neuropeptidov.

Kapitola 4

Rozpoznávanie a anotácia neuropeptidov

V tejto kapitole podrobne vysvetlíme postup, akým v našom systéme dochádza k anotácii sekvencií. Ako bolo už naznačené pri návrhu nášho systému v časti 2.3, ide o model strojového učenia s názvom *conditional random fields* [13] (CRF), ktorý umožňuje využitie rôznych druhov informácií za účelom anotácie sekvencií, v našom prípade sekvencií aminokyselín (t. j. proteínov). V nasledujúcom texte najprv vysvetlíme túto metódu i s jej zovšeobecnením *semi-Markov conditional random fields* (semi-CRF), ktorá bola v našom systéme využitá. Po zadefinovaní modelu popíšeme, ako bol náš konkrétny problém modelovaný a aké rôzne informácie je schopný využiť pri anotácii. Na záver kapitoly taktiež uvedieme niekoľko experimentov, ktoré boli vykonané a popíšeme ich výsledky.

4.1 Úvod do CRF

Model strojového učenia *conditional random fields* nemodeluje spoločnú pravdepodobnosť dát a ich anotácií, ale podmienenú pravdepodobnosť anotácií vzhľadom na dáta. Stále však poskytuje možnosť ovplyvňovania anotácie sekvencie jednej časti druhými, čo vedie ku globálne optimálnej anotácii. Navyše, pomocou CRF je možné zapracovať viaceré štatisticky korelované vlastnosti dát a využiť ich v podobe *atribútov* pri anotácii. CRF môže byť štandardne použité pre rôzne definované grafové modely, avšak my v našej práci pracujeme so sekvenčnými dátami, a preto aj definície prispôsobíme týmto účelom.

Ak uvažujeme sekvenčné dáta, nech $\mathbf{X} = (x_1, x_2, \cdots, x_n)$ je sekvencia a $\mathbf{Y} = (y_1, y_2, \cdots, y_n)$ jej anotácia. CRF modeluje podmienenú pravdepodobnosť $\Pr(\mathbf{Y}|\mathbf{X})$. Definujme teraz K *lokálnych atribútov* $f_1, f_2, \cdots, f_K$, pričom každý lokálny atribút priradí dvojici $(\mathbf{X}, \mathbf{Y})$ a indexu $i \in \{1, \cdots, n\}$ hodnotu $f_k(i, \mathbf{X}, y_{i-1}, y_i) \in \mathbb{R}$. Jednotlivé hodnoty $f_k(i, \mathbf{X}, y_{i-1}, y_i)$ teda predstavujú hodnoty lokálnych atribútov pre pozíciu

i v sekvencii $\mathbf{X}$, ktorej anotácia je y_i, pričom anotácia predchádzajúcej pozície je y_{i-1}. *Globálny atribút* F_k definujeme ako súčet hodnôt lokálneho atribútu f_k pre všetky pozície v sekvencii,

$$F_k(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{|\mathbf{X}|} f_k(i, \mathbf{X}, y_{i-1}, y_i).$$

Potom autori Lafferty a kol. [13] definujú pravdepodobnosť $\Pr(\mathbf{Y}|\mathbf{X})$ v CRF ako

$$\Pr(\mathbf{Y}|\mathbf{X}, \mathbf{w}) = \frac{1}{Z_{\mathbf{w}}(\mathbf{X})} \exp\left(\sum_{k=1}^{K} w_k F_k(\mathbf{X}, \mathbf{Y})\right), \tag{4.1}$$

kde $\mathbf{w} = (w_1, w_2, \cdots, w_K,)$ je vektor váh jednotlivých globálnych atribútov a $Z_{\mathbf{w}}(\mathbf{X})$ predstavuje normalizačnú konštantu, teda

$$Z_{\mathbf{w}}(\mathbf{X}) = \sum_{\overline{\mathbf{Y}}} \exp\left(\sum_{k=1}^{K} w_k F_k(\mathbf{X}, \overline{\mathbf{Y}})\right). \tag{4.2}$$

Po zadefinovaní lokálnych atribútov teda samotné trénovanie modelu CRF spočíva v nájdení takých váh $\mathbf{w}$, ktoré maximalizujú výraz (4.1). Z definície taktiež vyplýva, že vďaka normalizácii pomocou $Z_{\mathbf{w}}(\mathbf{X})$ je možné definovať hodnoty atribútov úplne ľubovoľne a nemusia predstavovať žiadne pravdepodobnostné hodnoty.

Avšak nevýhodou takto zadefinovaného modelu je, že niektoré vlastnosti dát sa pomocou uvedených lokálnych atribútov veľmi ťažko definujú. Ide o vlastnosti, ktoré sú špecifické pre dlhší úsek skúmanej sekvencie, ktorý je anotovaný rovnakou anotáciou, a nie len pre úseky jednotkovej dĺžky. Existuje preto zovšeobecnenie, ktoré uvažuje anotáciu sekvencie ako anotáciu segmentov sekvencie $\mathbf{S} = (s_1, s_2, \cdots, s_p)$, kde p je počet segmentov, na ktoré je sekvencia $\mathbf{X}$ segmentovaná pomocou jej anotácie $\mathbf{Y}$. Teda anotáciu sekvencie $\mathbf{X}$ možno vyjadriť v tvare $\mathbf{S} = (b_i, e_i, a_i)_{i=1}^{p}$, kde b_i predstavuje pozíciu začiatku segmentu s_i, e_i jeho koniec a a_i jeho anotáciu. Platí preto:

$$b_1 = 1; \quad e_i \geq b_i; \quad e_{i-1} + 1 = b_i; \quad e_p = n; \quad a_{i-1} \neq a_i$$

a zároveň anotácie segmentov sedia s anotáciou celej sekvencie. Konkrétne

$$y_{b_i} = y_{b_i+1} = y_{b_i+2} = \cdots = y_{e_i} = a_i.$$

Použitím takto definovaného modelu možno funkciu predstavujúcu globálny atribút definovať ako

$$G_k(\mathbf{X}, \mathbf{S}) = \sum_{i=1}^{p} g_k(a_{i-1}, b_i, e_i, a_i, \mathbf{X}), \tag{4.3}$$

kde g_k predstavuje lokálny atribút závislý od pozície segmentu s_i, jeho anotácie a_i,

anotácie predchádzajúceho segmentu a_{i-1} a celej sekvencie $\mathbf{X}$. Pre podmienenú pravdepodobnosť $\Pr(\mathbf{S}|\mathbf{X})$ platí podobný vzťah ako (4.1) len s použitím tejto novej definície globálnych atribútov

$$\Pr(\mathbf{S}|\mathbf{X}, \mathbf{w}) = \frac{1}{Z_{\mathbf{w}}(\mathbf{X})} \exp\left(\sum_{k=1}^{K} w_k G_k(\mathbf{X}, \mathbf{S})\right), \tag{4.4}$$

kde opäť $\mathbf{w}$ predstavuje vektor váh jednotlivých globálnych atribútov a $Z_{\mathbf{w}}(\mathbf{X})$ je normalizačnou konštantou, v tomto prípade

$$Z_{\mathbf{w}}(\mathbf{X}) = \sum_{\overline{\mathbf{S}}} \exp\left(\sum_{k=1}^{K} w_k G_k(\mathbf{X}, \overline{\mathbf{S}})\right). \tag{4.5}$$

Toto zovšeobecnenie CRF sa štandardne nazýva *semi-Markov conditional random fields* (semi-CRF) [15], avšak v nasledujúcom texte budeme pod skratkou CRF vždy myslieť práve tento typ.

4.1.1 Inferencia

Po vyššie uvedenom definovaní modelu semi-CRF možno samotnú inferenciu v tomto modeli definovať ako nájdenie optimálnej segmentácie sekvencie $\mathbf{X}$ pri použití váh $\mathbf{w}$, teda $\arg\max_{\mathbf{S}} \Pr(\mathbf{S}|\mathbf{X}, \mathbf{w})$, kde pravdepodobnosť $\Pr(\mathbf{S}|\mathbf{X}, \mathbf{w})$ je definovaná vzťahom (4.4). Pri použití definícií (4.4) a (4.3) preto dostávame, že inferencia zodpovedá maximalizácii

$$\arg\max_{\mathbf{S}} \Pr(\mathbf{S}|\mathbf{X}, \mathbf{w}) = \arg\max_{\mathbf{S}} \sum_{k=1}^{K} \left(w_k \sum_{i=1}^{p} g_k(a_{i-1}, b_i, e_i, a_i, \mathbf{X})\right).$$

Pri danej sekvencii $\mathbf{X}$, váhach atribútov $\mathbf{w}$ a hornej hranici dĺžky segmentov, sa následne dá získať optimálna segmentácia S algoritmom podobným štandardnému Viterbiho algoritmu [15].

4.1.2 Trénovanie modelu

Pri trénovaní modelu semi-CRF máme dané trénovacie dáta $T = \{(\mathbf{X}_i, \mathbf{S}_i)\}_{i=1}^{N}$ obsahujúce sekvencie aj ich správne segmentácie. Cieľom je maximalizovať podmienenú pravdepodobnosť jednotlivých segmentov v modeli, pričom táto pravdepodobnosť je daná váhovaním $\mathbf{w}$. Nakoľko je funkcia logaritmu neklesajúca, tak maximalizácia tejto pravdepodobnosti je ekvivalentná maximalizácii jej logaritmickej hodnoty. Vzhľadom na viaceré výhody sa preto maximalizuje logaritmická hodnota pravdepodobnosti tré-

novacích dát a môže byť pomocou vzťahu (4.4) vyjadrená v tvare

$$L_T(\mathbf{w}) = \sum_i \log \Pr(\mathbf{S}_i|\mathbf{X}_i, \mathbf{w}) = \sum_i \left(\sum_{k=1}^{K} w_k G_k(\mathbf{X}_i, \mathbf{S}_i) - \log Z_{\mathbf{w}}(\mathbf{X}_i) \right). \tag{4.6}$$

Cieľom trénovania je nájdenie takého váhovania $\mathbf{w}$, ktoré vedie k maximálnej hodnote $L_T(\mathbf{w})$. Užitočnou vlastnosťou funkcie L_T je jej konkávnosť, vďaka ktorej je optimálne váhovanie možné nájsť gradientovými metódami alebo mnohými podobnými metódami. Nami použitá implementácia za týmto účelom využíva *kvázi-Newtonovu metódu* s obmedzenou pamäťou [15], kde je gradient vyjadrený v tvare

$$\begin{aligned} \nabla L_T(\mathbf{w}) &= \sum_i \mathbf{G}(\mathbf{X}_i, \mathbf{S}_i) - \frac{\sum_{\overline{\mathbf{S}}} \mathbf{G}(\mathbf{X}_i, \overline{\mathbf{S}}) e^{\mathbf{w}\mathbf{G}(\mathbf{X}_i, \overline{\mathbf{S}})}}{Z_{\mathbf{w}}(\mathbf{X}_i)} \\ &= \sum_i \mathbf{G}(\mathbf{X}_i, \mathbf{S}_i) - \mathrm{E}_{\Pr(\overline{\mathbf{S}}|\mathbf{w})} \mathbf{G}(\mathbf{X}_i, \overline{\mathbf{S}}), \end{aligned} \tag{4.7}$$

kde $\mathbf{G}(\mathbf{X}, \mathbf{S})$ predstavuje vektor globálnych atribútov $\mathbf{G} = (G_1, G_2, \cdots, G_K)$. Prvý člen vzťahu (4.7) sa vyráta priamočiaro a očakávanú hodnotu v druhej časti možno vyrátať pomocou metódy dynamického programovania [15].

4.2 Model na anotáciu proteínových sekvencií

Vyššie popísaný model CRF sme sa rozhodli použiť pri riešení nášho problému anotácie sekvencií aminokyselín. Na základe popisu nami navrhnutého modelu v časti 2.3 je naším cieľom navrhnúť model tak, aby bol schopný anotovať jednotlivé úseky (segmenty) skúmaných sekvencií, na základe ktorých bude možné rozlíšiť, či je skúmaná sekvencia prekurzorom neuropeptidov. V tomto prípade taktiež vyžadujeme, aby bolo možné zo získaných anotácií rozlíšiť, ktoré časti skúmanej sekvencie predstavujú neuropeptidy.

V našej práci sme sa rozhodli použiť knižnicu, pomocou ktorej je možné vytvárať a používať model CRF bez nutnosti implementácie algoritmov trénovania a inferencie. Avšak vzhľadom na to, že chceme pracovať so zovšeobecnením semi-CRF, nájdenie dostupnej implementácie nie je veľmi jednoduché. Nakoniec sme však našli implementáciu Prof. Sunita Sarawagi, ktorá umožňuje pracovať aj s nami požadovaným modelom. Ide o projekt v jazyku Java, ktorý je voľne dostupný na stránkach projektu [17].

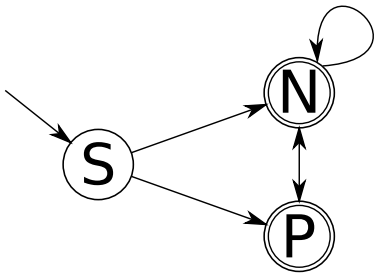
Za účelom použitia modelu CRF pri riešení problému rozpoznávania neuropeptidov je v prvom rade nutné navrhnúť množinu anotácií, ktorými budeme jednotlivé segmenty skúmaných sekvencií označovať. Po voľbe týchto anotácií je následne nevyhnutné navrhnúť atribúty, tvoriace neoddeliteľnú súčasť modelu CRF a na základe ktorých je samotná anotácia realizovaná. V nasledujúcom texte tieto dve časti návrhu modelu bližšie popíšeme a opíšeme ich realizáciu v našom systéme.

4.2.1 Topológia modelu

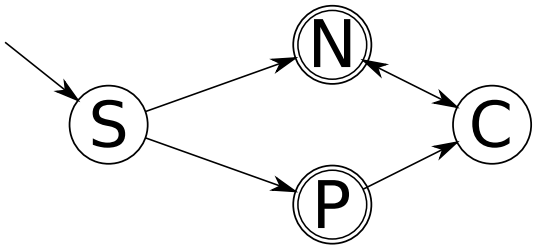
Prvým krokom pri používaní CRF modelu je voľba anotácií, ktoré chceme pri anotovaní používať. Tieto anotácie by mali taktiež popisovať štruktúru sekvencií, ktorými sa chceme zaoberať, a preto je vhodné ich reprezentovať stavovým diagramom, ktorého stavy reprezentujú jednotlivé anotácie a prechodová hrana idúca zo stavu u do v symbolizuje, že segment anotovaný anotáciou v môže nasledovať za segmentom anotovaným ako u.

Uvažujme najprv jednoduchší prípad, keď chceme navrhnúť model CRF, ktorý bude správne anotovať prekurzory neuropeptidov, pričom má zaručené, že vstupná sekvencia je prekurzor. Intuitívne vieme navrhnúť stavový diagram znázornený na obrázku 4.1a, kde vrchol S reprezentuje signálny peptid nachádzajúci sa na začiatku každej prípustnej sekvencie, neuropeptidy sú reprezentované vrcholom N a ostatné časti vrcholom P. Pod slovom *neuropeptid* však nebudeme myslieť neuropeptid definovaný v časti 1.4, ale časť prekurzoru neuropeptidov, ktorá vznikla po štiepení prekurzoru a z ktorej sa následne modifikáciami v bunke neuropeptid vytvorí (viď sekciu 1.4). Nakoľko sú tieto modifikácie dobre preskúmané a vieme ich dobre simulovať, je problém rozpoznávania neuropeptidov ekvivalentný s problémom rozpoznania týchto dlhších úsekov.

Stavový diagram na obrázku 4.1a však nie je možné priamo použiť pri tvorbe modelu CRF, nakoľko obsahuje slučku, teda hranu vedúcu do vrchola, z ktorého taktiež vychádza. Toto je vlastnosť, ktorá je v spore s definíciou semi-CRF, podľa ktorej nemôžu mať dva po sebe idúce segmenty rovnakú anotáciu (viď sekciu 4.1), a preto nie je štandardnými implementáciami semi-CRF podporovaná. Na druhej strane, diagram na obrázku 4.1b neobsahuje žiadne slučky, a preto je korektný vzhľadom na CRF. Sémantika tohto modelu je taká, že vrchol so symbolom C reprezentuje segment jednotkovej dĺžky na pozícii, za ktorou dochádza k štiepeniu. Sémantika ostatných vrcholov je rovnaká ako v predchádzajúcom diagrame.



(a) Jednoduchý stavový diagram popisujúci štruktúru prekurzoru neuropeptidov, avšak obsahujúci slučku.



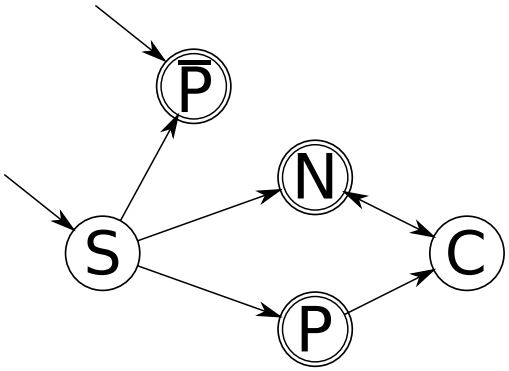
(b) Stavový diagram popisujúci štruktúru prekurzoru neuropeptidov bez nutnosti použitia slučiek. Vrchol C reprezentuje anotáciu aminokyseliny (t. j. úseku jednotkovej dĺžky), za ktorou dochádza k štiepeniu.

Obr. 4.1: Stavové diagramy popisujúce anotácie prekurzoru neuropeptidov. Prechodová hrana vedúca do vrchola S nemá začiatok, a preto predstavuje začiatok sekvencie (každý prekurzor neuropeptidov začína signálnym peptidom). Vrcholy zvýraznené dvojitým krúžkom symbolizujú konečné stavy, teda anotácie, ktorými môže prekurzor neuropeptidov končiť.

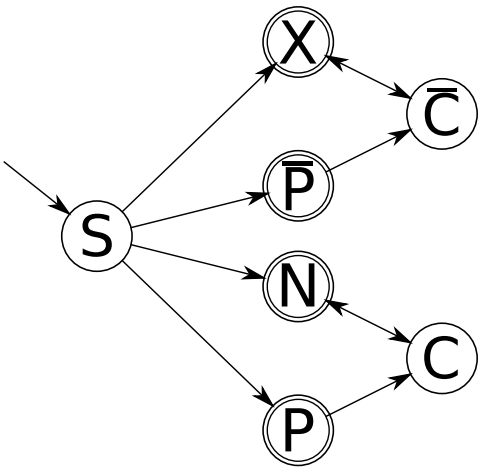
Diagram z obrázku 4.1b dobre popisuje štruktúru prekurzorov neuropeptidov, avšak cieľom nášho systému je, aby bol schopný pracovať taktiež so sekvenciami, ktoré takýmto prekurzorom nie sú. Je preto nutné uviesť iný stavový diagram, ktorý nám zároveň umožní rozlišovať aj tieto skupiny sekvencií. Priamočiarym rozšírením vyššie uvedeného diagramu, ktorý by spĺňal požadované vlastnosti, je stavový diagram znázornený na obrázku 4.2a. Tento diagram umožňuje okrem sekvencií prekurzorov neuropeptidov popísať taktiež iné proteínové sekvencie, ktorých anotáciu označujeme symbolom $\overline{P}$, ale i sekvencie, ktoré začínajú signálnym peptidom, avšak nie sú prekurzorom neuropeptidov[1]. Takto zvolený model anotácií má však jeden veľký nedostatok. Problémom je, že prekurzory neuropeptidov nie sú jedinými proteínmi, u ktorých dochádza k štiepeniu na menšie časti. V prípade, kedy by sme sa týmto stavovým diagramom pokúšali anotovať prekurzor štiepiaci sa na menšie časti, z ktorých sa následne tvoria iné látky než neuropeptidy, by mohlo ľahko dôjsť k chybnej anotácii. Konkrétne, skúmaná sekvencia by mohla byť anotovaná ako prekurzor neuropeptidov, nakoľko spĺňa jeho viaceré vlastnosti ako výskyt miest štiepenia a úsekov propeptidov P. Z týchto dôvodov sme sa preto rozhodli zvoliť stavový diagram anotácií, ktorým by bolo možné dobre rozlišovať medzi prekurzormi neuropeptidov a prekurzormi iných látok. Tento diagram je znázornený na obrázku 4.2b. Ako vidno z obrázku, uvažujeme, že každá skúmaná sekvencia začína signálnym peptidom a následne sa v diagrame vydá do jednej z dvoch častí, jednej pre prekurzory neuropeptidov (vrcholy N, P a C majú rovnaký význam ako v predchádzajúcich modeloch) a druhej pre ostatné sekvencie, pričom vrcholom X reprezentujeme anotáciu funkčných častí, ktoré sú po štiepení na miestach $\overline{C}$ po-

[1]Viaceré proteínové sekvencie môžu začínať nejakým signálnym peptidom, v závislosti od konkrétneho typu. Význam signálneho peptidu bol bližšie popísaný v časti 1.4.

važované za cieľové produkty (podobne ako neuropeptidy N v prípade prekurzorov neuropeptidov). Takto zvoleným stavovým diagramom je preto možné dobre rozlišovať prekurzory neuropeptidov a prekurzory iných látok, ktoré taktiež vznikajú štiepením na menšie časti. Síce nie sme schopní správne anotovať sekvencie, ktoré nezačínajú signálnym peptidom, avšak toto nie je veľkým problémom, nakoľko sa zaoberáme problémom rozpoznávania neuropeptidov, ktorých prekurzory signálny peptid obsahujú a zároveň už existuje dobrý nástroj na ich identifikáciu (viď sekciu 2.2.1). Ak máme danú skúmanú sekvenciu, pre ktorú chceme riešiť problém rozpoznania neuropeptidov, tak v prvom kroku v nej skúsime nájsť signálny peptid programom SignalP. Ak pre všetky uvažované konce potenciálneho signálneho peptidu dostaneme veľmi nízke skóre, potom môžeme predpokladať, že skúmaná sekvencia signálny peptid neobsahuje, a preto ani nie je prekurzorom neuropeptidov. V opačnom prípade môžeme použiť nami navrhnutý diagram anotácií na popísanie jej štruktúry.



(a) Stavový diagram popisujúci štruktúru proteínov. Je ním možné popísať aj proteíny, ktoré nie sú prekurzorom neuropeptidov (vrchol s anotáciou $\overline{P}$), avšak je náchylný napomáhať tvorbe zlých anotácií prekurzorov iných látok než neuropeptidov.



(b) Stavový diagram popisujúci štruktúru proteínov obsahujúcich signálny peptid. Na rozdiel od grafu na obrázku 4.2a ním možno dobre rozlišovať prekurzory neuropeptidov a prekurzory iných látok.

Obr. 4.2: Stavové diagramy popisujúce anotáciu proteínov. Význam vrcholov a prechodových hrán je rovnaký ako v obrázku 4.1.

4.2.2 Tvorba atribútov modelu

Po zvolení množiny anotácií, ktorými budeme skúmanú sekvenciu popisovať, je samotná anotácia sekvencie pomocou CRF závislá jedine od zvolených atribútov (viď 4.1). Preto taktiež v našom systéme zohráva výber atribútov kľúčovú úlohu pri správnom riešení problému rozpoznania neuropeptidov. Pri ich tvorbe sme však museli brať ohľad aj na obmedzenia, ktoré boli kladené samotnou implementáciou CRF, ktorú sme použili. Ide najmä o fakt, že algoritmy trénovania a inferencie sú z dôvodu lepších presností počítané v log-priestore, čo má za následok, že hodnoty atribútov môžu byť

len nezáporné, nakoľko logaritmická funkcia nie je pre záporné hodnoty definovaná. Ďalším z väčších obmedzení je, že táto implementácia je schopná pracovať len s takými diagramami anotácií, ktoré predstavujú kompletné orientované grafy, a preto je nutné požadované vlastnosti zabezpečiť vhodným výberom atribútov, ktoré umožnia len prípustné anotácie proteínov.

Nami vytvorené atribúty možno rozdeliť do piatich základných skupín, ktoré budú následne presne popísané. K väčšine atribútov taktiež uvedieme, aká by mala byť v ideálnom prípade hodnota váhy daného atribútu po natrénovaní modelu. V zásade však možno len predpokladať, či bude táto váha záporná alebo kladná, nakoľko jej veľkosť veľmi závisí od ostatných použitých atribútov a toho, ako medzi nimi dochádza ku vzájomnej rovnováhe.

Obmedzenia topológie modelu

Nakoľko je nami použitá implementácia CRF schopná pracovať len s kompletnými grafmi anotácií, pri jej použití je nám umožnené zvoliť len počet vrcholov, ktoré má nami zvolený diagram z obrázku 4.2b obsahovať, avšak nie je možné definovať povolené prechody medzi anotáciami. Jediný spôsob, ktorým je možné zabezpečiť správnu anotáciu vzhľadom na zvolený diagram, je výber vhodných atribútov. Následným natrénovaním modelu CRF, a teda získaním optimálnych váh týchto atribútov, je možné „naučenie sa“ topológie modelu bez nutnosti explicitného definovania. Vytvorili sme preto atribúty, ktorých natrénované váhy zabezpečia správne začiatočné a koncové stavy (anotácie prvého a posledného segmentu skúmaných sekvencií) a taktiež správne prechody.

Pre každú anotáciu v boli vytvorené dva lokálne atribúty g_{begin_v} a g_{end_v}. Prvý mal za úlohu zachytiť informáciu o začiatočných stavoch a jeho hodnota sa rovnala

$$g_{begin_v}(a', b, e, a, \mathbf{X}) := \begin{cases} 1 & a = v \quad \wedge \quad b = 1 \\ 0 & \text{inak.} \end{cases} \tag{4.8}$$

Teda atribút g_{begin_v} má hodnotu 1 len v prípade, ak sekvencia X začína anotáciou v. Podobným spôsobom boli vytvorené aj atribúty g_{end_v}, ktoré zachytávajú informáciu o koncových stavoch diagramu. Konkrétne

$$g_{end_v}(a', b, e, a, \mathbf{X}) := \begin{cases} 1 & a = v \quad \wedge \quad e = n \\ 0 & \text{inak,} \end{cases} \tag{4.9}$$

kde n je dĺžka skúmanej sekvencie $\mathbf{X}$. Natrénovaním váh atribútov g_{begin_v} (resp. g_{end_v}) preto dostaneme váhy, pre ktoré očakávame, že budú kladné pre anotácie (t. j. stavy) v nachádzajúce sa na začiatku (resp. konci) nejakej trénovacej sekvencie. Ostatným atribútom budú priradené váhy záporné (teoreticky záporné nekonečné, avšak prakticky

ide len o nejaké veľmi nízke hodnoty).

Analogicky sme vytvorili i atribúty, ktorých úlohou je zachytiť informáciu o prechodoch z anotácie u do anotácie v. Pre každú dvojicu stavov (anotácií) sme preto vytvorili lokálny atribút

$$g_{edge_{(u,v)}}(a', b, e, a, \mathbf{X}) := \begin{cases} 1 & a' = u \quad \wedge \quad a = v \\ 0 & \text{inak.} \end{cases} \tag{4.10}$$

Rovnako ako v predchádzajúcom prípade, natrénovaním váh na trénovacích dátach očakávame kladné hodnoty v prípade prechodov nachádzajúcich sa dostatočne často v dátach, a teda prechodov prípustných vzhľadom na náš graf anotácií 4.2b.

Informácia o signálnom peptide

Ako už bolo spomenuté v predchádzajúcom texte, za účelom správnej anotácie signálneho peptidu sme použili program SignalP, ktorý bol popísaný v časti 2.2.1. Pre danú skúmanú sekvenciu $\mathbf{X}$, označme

$$\mathbf{signal}(\mathbf{X}) = (signal_1(\mathbf{X}), signal_2(\mathbf{X}), \cdots, signal_n(\mathbf{X}))$$

výstup programu SignalP pre túto sekvenciu. Vyššie hodnoty $signal_i(\mathbf{X})$ predstavujú vyššiu pravdepodobnosť, že signálny peptid končí na pozícii i. Na základe takýchto výstupov programu SignalP sme vytvorili atribút g_{signal}, ktorý ich mal za úlohu využiť, a tým bol jedným z hlavných atribútov zabezpečujúcich správnu anotáciu signálneho peptidu. Konkrétne

$$g_{signal}(a', b, e, a, \mathbf{X}) := \begin{cases} signal_e(\mathbf{X}) & b = 1 \quad \wedge \quad a = \mathtt{S} \\ 0 & \text{inak,} \end{cases} \tag{4.11}$$

kde výrazom $a = \mathtt{S}$ myslíme, že anotácia a predstavuje anotáciu signálneho peptidu S. Teda pre segment sekvencie začínajúci sa na jej začiatku, bude výstupom atribútu skóre programu SignalP pre poslednú pozíciu segmentu. Natrénovaním váhy tohto atribútu očakávame nejakú kladnú váhu, nakoľko vyššia hodnota $signal_i(\mathbf{X})$ pre koncovú pozíciu i znamená vyššiu pravdepodobnosť, že tu bude končiť signálny peptid.

Informácia o miestach štiepenia

Dôležitou informáciou, ktorú by sme chceli využiť pri anotácii, je informácia o miestach štiepenia. Za týmto účelom bol skonštruovaný náš model SVM, ktorý sme popísali v časti 3.2. Vďaka tomuto modelu sme schopní každej pozícii skúmanej sekvencie priradiť hodnotu, ktorá predstavuje pravdepodobnosť, že za touto pozíciou dochádza k štiepeniu. Vzhľadom na nami definované diagramy anotácií 4.1b a 4.2b znamená fakt,

že za danou pozíciou i dochádza k štiepeniu, to, že táto pozícia i má byť anotovaná anotáciou C, resp. $\overline{C}$. Na zachytenie tejto skutočnosti sme zostrojili atribúty g_{cleave_C} a $g_{cleave_{\overline{C}}}$, ktorých hodnota závisí od aktuálneho segmentu a predikcie natrénovaného modelu SVM. Teda, ak označíme výstup modelu M na predikciu štiepenia v sekvencii $\mathbf{X}$

$$\mathbf{cleavage}^M(\mathbf{X}) = (cleavage_1^M(\mathbf{X}), cleavage_2^M(\mathbf{X}), \cdots, cleavage_n^M(\mathbf{X})),$$

tak tieto atribúty možno definovať ako

$$g_{cleave_v}(a', b, e, a, \mathbf{X}) := \begin{cases} cleavage_e^{M(v)}(\mathbf{X}) & a = v \quad \wedge \quad b = e \\ 0 & \text{inak}, \end{cases} \tag{4.12}$$

kde $v \in \{C, \overline{C}\}$ a $M(v)$ predstavuje použitý model pre anotáciu v. Hodnota získaná pomocou modelu SVM je teda priradená len vtedy, keď anotácia a predstavuje anotáciu C alebo $\overline{C}$, a zároveň ide o segment jednotkovej dĺžky. Na získanie hodnôt $\mathbf{cleavege}^{M(v)}(\mathbf{X})$ pre jednotlivé anotácie C a $\overline{C}$ možno použiť buď rovnaký model (teda $M(C) = M(\overline{C})$), alebo aj dva rôzne. Natrénovaním modelu CRF sa nájde optimálna váha atribútu g_{cleave_v}, ktorá vlastne modelu na predikciu miest štiepenia priradí dôležitosť, resp. mieru informatívnosti, vzhľadom na celkový problém anotácie. Očakávame preto kladnú výslednú hodnotu natrénovanej váhy.

Informáciu o miestach štiepenia sme sa však rozhodli použiť i pri tvorbe ďalších atribútov. Atribúty g_{cleave_C} a $g_{cleave_{\overline{C}}}$ síce sú navrhnuté tak, aby zabezpečili správnu anotáciu segmentu anotáciou C (resp. $\overline{C}$) len ak je jednotkovej dĺžky a dochádza na ňom k štiepeniu, avšak chceli by sme pridať aj atribúty, ktoré by zabezpečovali, že segment s inou anotáciou nebude disponovať pozíciou s vysokou pravdepodobnosťou štiepenia. Vzájomnou rovnováhou týchto atribútov by sme chceli docieliť, aby boli pozície s väčšou pravdepodobnosťou štiepenia anotované ako C (resp. $\overline{C}$), avšak také, ktoré taktiež vedú k nízkym pravdepodobnostiam štiepenia v ostatných vzniknutých segmentoch. Za týmto účelom sme sa rozhodli pre každú anotáciu v inú než C alebo $\overline{C}$ skonštruovať atribút $g_{nocleave_v}$. Bolo by taktiež možné navrhnúť pre všetky anotácie v jeden spoločný atribút, ale týmto spôsobom by sa počas trénovania trénovala len jeho jedna váha. Nami zvolený postup však umožňuje väčšiu voľnosť a zabezpečuje využitie rôznych vlastností segmentov anotovaných inými anotáciami, nakoľko každej je priradená samostatná váha, ktorá bude optimalizovaná pre túto konkrétnu anotáciu.

Jednou z možností ako atribút $g_{nocleave_v}$ definovať je, že segmentom s anotáciou v priradí hodnotu predstavujúcu *maximum hodnôt* $\mathbf{cleavage}(\mathbf{X})$ v tomto segmente. Avšak nevýhodou tejto metriky je, že zohľadňuje len jednu pozíciu a nie celý segment. Uvažujme napríklad segment, v ktorom sa nachádza jedna pozícia s veľmi vysokou pravdepodobnosťou štiepenia c_{max} a ostatné pozície majú tieto hodnoty minimálne. Tento segment je väčšinou oveľa menej nežiaduci ako taký, ktorého dokonca viaceré

pozície disponujú tou istou pravdepodobnosťou štiepenia c_{max}, hoci majú oba rovnakú maximálnu hodnotu. Preto výber maximálnej hodnoty nie je postačujúci, nakoľko pri dvoch segmentoch s rovnakými maximálnymi hodnotami nie vždy platí, že sú rovnako zlé, a podobne nie je možné na základe porovnania rôznych takýchto hodnôt povedať, ktorý z im prislúchajúcich segmentov je lepší. Je preto nutné uvažovať inú metriku, ako napríklad *súčet hodnôt* **cleavage**($\mathbf{X}$) v segmente. Táto metrika síce rieši spomínaný problém, avšak len v prípade porovnávania segmentov rovnakej dĺžky. Je prirodzené, že dlhší segment má zväčša súčet hodnôt pravdepodobností štiepenia väčší než segment kratší, ale túto skutočnosť metrika súčtu nezohľadňuje.

Prirodzenou modifikáciou metriky súčtu je *priemer hodnôt* **cleavage**($\mathbf{X}$) v segmente, ktorý vznikne normalizáciou súčtu hodnôt dĺžkou daného segmentu. My sme sa však rozhodli v našej práci použiť *priemer druhých mocnín hodnôt*. Na rozdiel od obyčajného priemeru totiž nejde o lineárnu funkciu hodnôt **cleavage**($\mathbf{X}$) v danom segmente, čo má za dôsledok, že takto definovaná metrika jednak spĺňa vlastnosti, ktoré chýbali metrike využívajúcej len maximálnu hodnotu, ale navyše tiež priraďuje vyšším hodnotám ešte väčšiu váhu (nie lineárne, ale kvadraticky). Ak preto chceme minimalizovať celkový súčet priemeru druhých mocnín hodnôt jednotlivých segmentov, týmto prístupom je vyvolávaný väčší dôraz na zamedzenie segmentov s vysokými hodnotami. Nami vytvorené atribúty sme navrhli v tvare

$$g_{nocleave_v}(a', b, e, a, \mathbf{X}) := \begin{cases} \frac{1}{(e-b+1)} \sum_{i=b}^{e} (cleavage_i^{M(v)}(\mathbf{X}))^2 & a = v \\ 0 & \text{inak.} \end{cases} \tag{4.13}$$

Podobne ako pri vytváraní atribútov g_{cleave_v} môže byť pre rôzne anotácie v použitý iný model predikcie štiepení $M(v)$, alebo pre všetky anotácie ten istý. Nakoľko vyššie hodnoty atribútu $g_{nocleave_v}$ predstavujú horšie vlastnosti daného segmentu, natrénovaním modelu CRF sa pre tento atribút očakáva váha záporná.

Informácia o dĺžke

Užitočnou informáciou, ktorá môže byť použitá pri anotácii segmentov je aj informácia o ich štandardnej dĺžke. Jedným z dobrých prístupov je vytvorenie atribútu g_{length_v} pre každú uvažovanú anotáciu, ktorý by priradil segmentu anotovanému ako v hodnotu podľa toho, akú má dĺžku. Čím štandardnejšia dĺžka segmentu, tým vyššia hodnota atribútu. Hodnoty takéhoto atribútu preto môžu byť napríklad vytvorené na základe hodnôt rozdelenia pravdepodobností dĺžok, čiže hodnota atribútu pre segment dĺžky l anotovaný ako v bude pravdepodobnosť, že segment anotovaný anotáciou v má dĺžku l v tomto rozdelení. Aproximácia rozdelenia dĺžok sa dá vypočítať na základe známych, dobre anotovaných sekvencií, avšak na dobrú aproximáciu treba sekvencií veľa.

Bohužiaľ, v našom prípade takýchto sekvencií veľa nie je (trénovacie dáta popi-

sujeme v časti 4.4), boli sme preto nútení zvoliť iný prístup na zachytenie informácie o dĺžkach. Pre každý segment sme vždy na základe literatúry alebo odhadom určili, aké dlhé jednotlivé segmenty môžu približne byť, pričom ale stále pripúšťame, že segment môže byť kratší alebo dlhší. Napríklad v prípade neuropeptidov vieme, že až 98% doteraz známych nie je dlhších ako 60 a zároveň vieme, že sú dĺžky aspoň 3 [9]. Jednotlivým anotáciám segmentov v, preto priradíme hodnoty min_v a max_v, ktoré znamenajú, že štandardná dĺžka segmentov anotovaných ako v leží v intervale $[min_v, max_v]$. Na základe týchto intervalov sme pre každú anotáciu v vytvorili tri atribúty g_{short_v}, g_{normal_v} a g_{long_v} a definovali ich nasledovne:

$$g_{short_v}(a', b, e, a, \mathbf{X}) := \begin{cases} 1 & a = v \quad \wedge \quad (e - b + 1) < min_v \\ 0 & \text{inak} \end{cases} \tag{4.14}$$

$$g_{normal_v}(a', b, e, a, \mathbf{X}) := \begin{cases} 1 & a = v \quad \wedge \quad (e - b + 1) \in [min_v, max_v] \\ 0 & \text{inak} \end{cases} \tag{4.15}$$

$$g_{long_v}(a', b, e, a, \mathbf{X}) := \begin{cases} 1 & a = v \quad \wedge \quad (e - b + 1) > max_v \\ 0 & \text{inak} \end{cases} \tag{4.16}$$

Pri zvolení vhodných jednotlivých hodnôt min_v a max_v teda vieme rozlišovať, ktoré dĺžky sú vzhľadom na uvažovanú anotáciu príliš krátke alebo dlhé. Konzervatívnym prístupom, pri ktorom nastavíme hodnotu min_v veľmi malú a hodnotu max_v veľmi veľkú, a následným natrénovaním modelu získame váhy atribútov g_{short_v} a g_{long_v} záporné a veľmi nízke, čo je dôsledkom toho, že takýchto krátkych alebo dlhých segmentov je veľmi málo. Použitie takto natrénovaného modelu pri anotácii potom zabezpečí, že v zásade všetky segmenty anotované ako v budú mať dĺžku v rozmedzí min_v a max_v. Na druhej strane, pri použití malého intervalu $[min_v, max_v]$, ktorý bude pokrývať len najviac frekventované dĺžky, získame natrénovaním modelu vysokú kladnú váhu atribútu g_{normal_v}, čo má za následok vysokú presnosť správnej anotácie segmentov štandardnej dĺžky. Hoci teda má nastavenie min_v a max_v vplyv na výslednú anotáciu, ťažko určiť nejaké ich konkrétne hodnoty vedúce k ideálnemu modelu. V zásade preto postačuje, aby boli zvolené tak, aby nimi tvorený interval pokrýval najviac frekventované dĺžky.

Štatistické informácie získané zo sekvencie

Poslednou uvažovanou skupinou atribútov sú tie, ktoré akýmsi spôsobom popisujú vnútornú štruktúru jednotlivých segmentov skúmaných sekvencií. Práve tieto atribúty sa hlavnou mierou podieľajú na úlohe rozlišovania anotácií jednotlivých segmentov, pričom na tvorbe samotných segmentov sa podieľajú najmä atribúty zo skupiny atribútov

využívajúcich informáciu o miestach štiepenia. Ako bolo taktiež spomenuté v predchádzajúcom texte v časti 2.2.4, z pozorovaní výskumnej skupiny Ofer a kol. [7] napríklad vyplýva, že pre sekvencie predstavujúce prekurzory neuropeptidov je typická zvýšená frekvencia aromatických aminokyselín (viď tabuľku 1.1). Práve toto pozorovanie nás viedlo k tomu, aby sme vytvorili atribúty, ktoré budú zachytávať informáciu o frekvencii aromatických aminokyselín v jednotlivých segmentoch. Podobne ako pri tvorbe predchádzajúcich atribútov sme zvolili prístup, pri ktorom boli pre jednotlivé anotácie v vytvorené samostatné atribúty g_{aroma_v}, ktorých váhy môžu byť trénované špecificky pre konkrétnu anotáciu. Vytvorili sme teda atribúty, ktorých hodnoty boli definované nasledovne:

$$g_{aroma_v}(a', b, e, a, \mathbf{X}) := \begin{cases} \sum_{i=b}^{e} [x_i \text{ je aromatická aminokyselina}] & a = v \\ 0 & \text{inak.} \end{cases} \tag{4.17}$$

Teda atribút g_{aroma_v} počíta počet výskytov aromatických aminokyselín v sekvencii $\mathbf{X}$ v uvažovanom segmente[2]. Takto definované atribúty sa neskôr počas testovaní ukázali byť užitočné, pričom sme zistili, že po natrénovaní modelu je váhe prislúchajúcej ku atribútu g_{aroma_N} vždy priradená výrazne vyššia kladná hodnota. To znamená, že zvýšená frekvencia aromatických aminokyselín v segmente je dobrým ukazovateľom toho, že ide o segment prislúchajúci neuropeptidu. Pozorovanie zvýšenej frekvencie aromatických aminokyselín v prekurzoroch neuropeptidov, ktoré bolo zistené výskumnou skupinou Ofer a kol. je preto len dôsledkom toho, že neuropeptidy vo všeobecnosti sú typicky tvorené zvýšeným počtom týchto aminokyselín.

Za účelom zachytenia vnútornej štruktúry segmentov sme sa rozhodli vytvoriť aj ďalšie atribúty zohľadňujúce rôzne frekvencie jednotlivých aminokyselín v segmentoch jednotlivých anotácií. Uvažujme na chvíľu situáciu, pri ktorej máme danú sekvenciu aminokyselín $\mathbf{s} = (s_1, s_2, \cdots, s_n)$ a zároveň funkciu všeobecnej distribúcie aminokyselín. Potom pravdepodobnosť uvažovanej sekvencie vzhľadom na danú distribúciu je $\prod_{i=1}^{n} \Pr(s_i)$. Nech je teraz daná sekvencia spolu s jej segmentáciou a anotáciou jednotlivých častí $\mathbf{s} = (b_i, e_i, a_i)_{i=1}^{p}$, kde p je počet jej segmentov a b_i, e_i, a_i predstavujú začiatok, koniec a anotáciu i-tého segmentu. Ak máme pre každú anotáciu danú taktiež jej distribúciu aminokyselín, potom následne vieme vyjadriť pravdepodobnosť sekvencie $\mathbf{s}$ vzhľadom na tieto distribúcie v tvare $\Pr(\mathbf{s}) = \prod_{i=1}^{p} \prod_{j=b_i}^{e_i} \Pr(s_j|a_i)$. Práve na základe týchto úvah sme pre každú anotáciu v vytvorili samostatný atribút g_{emit_v}, ktorý jednotlivým segmentom priradí logaritmickú hodnotu pravdepodobnosti sekvencie amino-

[2]Notácia [A] predstavuje hodnotu 1, ak je výrok A pravdivý a hodnotu 0, ak je nepravdivý.

kyselín tohto segmentu, vzhľadom na jeho anotáciu. Teda konkrétne

$$g_{emit_v}(a', b, e, a, \mathbf{X}) := \begin{cases} \sum_{i=b}^{e} \log \Pr(x_i|v) & a = v \\ 0 & \text{inak.} \end{cases} \tag{4.18}$$

V prípade, kedy by sme ručne nastavili váhy všetkých takýchto atribútov na hodnotu 1, potom zo vzťahu (4.4) vyplýva, že je ich použitie ekvivalentné použitiu pravdepodobnosti $\Pr(\mathbf{s})$ z vyššie uvedeného príkladu. Avšak možnosť trénovania modelu, a tým nájdenia optimálnych hodnôt jednotlivých váh, dodáva modelu CRF možnosť použiť rôznu mieru významnosti tejto informácie, čím získava väčšiu silu učenia sa.

Na použitie atribútov g_{emit_v} je však ešte potrebné disponovať pravdepodobnosťou $\Pr(x_i|v)$. Jej aproximáciu je možné dostať z trénovacích dát alebo na základe iných pozorovaní. V tomto prípade, na rozdiel od dĺžok segmentov, pri ktorých nebolo možné vytvoriť dostatočne dobrú aproximáciu ich distribúcie, takýto problém nemáme, nakoľko sa zaoberáme distribúciou samotných aminokyselín, pre ktoré dostatok výskytov v dostupných dátach máme[3].

Takto vytvorenými atribútmi je náš model schopný zakomponovať do anotácie i informáciu o frekvencii jednotlivých aminokyselín. Avšak, ako sa napríklad ukázalo v prípade aromatických aminokyselín, niekedy nemusí vždy záležať na tom, aká presne aminokyselina sa v sekvencii vyskytuje, ale jej vlastnosti sú ovplyvňované výskytom ľubovoľnej aminokyseliny z nejakej väčšej skupiny. Jedno z možných rozdelení aminokyselín do skupín, na základe viacerých chemických a biologických vlastností, bolo uvedené v tabuľke 1.1. Rozhodli sme sa preto vytvoriť ďalšie atribúty g_{emitg_v} podobným spôsobom ako g_{emit_v}, avšak nebudeme uvažovať distribúciu jednotlivých aminokyselín v segmentoch rôznych anotácií, ale distribúciu skupín aminokyselín. Tieto atribúty preto možno vyjadriť v tvare

$$g_{emitg_v}(a', b, e, a, \mathbf{X}) := \begin{cases} \sum_{i=b}^{e} \log \Pr(group(x_i)|v) & a = v \\ 0 & \text{inak,} \end{cases} \tag{4.19}$$

kde $group(x)$ predstavuje jednu z uvažovaných skupín, do ktorej patrí aminokyselina x_i, a teda pravdepodobnosť $\Pr(group(x_i)|v)$ je pravdepodobnosť výskytu niektorej aminokyseliny z tejto skupiny v segmente anotovanom ako v. Aproximácie týchto distribúcií je možné získať, podobne ako v predchádzajúcom prípade, z anotovaných trénovacích dát.

[3]Celkový počet segmentov v dostupných dátach je oveľa menší ako celkový počet jednotlivých aminokyselín.

4.3 Implementácia

V predchádzajúcich častiach textu sme popísali viaceré metódy a navrhli ako ich využiť pri probléme rozpoznávania neuropeptidov. Pri implementovaní týchto metód sme použili voľne dostupnú knižnicu CRF [17] určenú primárne na problém segmentácie textu, vďaka ktorej bolo vykonané samotné trénovanie modelu na trénovacích dátach a následná inferencia. Hoci zatiaľ neexistuje kompletná dokumentácia tejto knižnice implementovanej v jazyku Java, vybrali sme si ju na základe toho, že poskytuje možnosť pracovať so zovšeobecneným modelom semi-Markov CRF, ktorý je nevyhnutný pre implementáciu nami navrhnutého postupu. Táto knižnica poskytuje viaceré metódy na prácu s rôznymi typmi abstraktných tried, ktoré je nutné implementovať za účelom možnosti definovania konkrétneho modelu, jeho trénovania i následného použitia pri anotácii sekvencií. Nakoľko zvolená knižnica pri jej použití kladie viaceré obmedzenia, z ktorých boli niektoré spomenuté v časti 4.2.2, boli sme sa im nútení prispôsobiť pri tvorbe modelu i jeho atribútov.

4.3.1 Použitie navrhnutého systému

Fungovanie nami navrhnutého systému, ktoré bolo stručne vysvetlené v časti 2.3, pozostáva z trénovania modelu CRF a následného použitia pri anotácii. Pred samotným trénovaním modelu je však nutné získať všetky informácie potrebné na správne fungovanie navrhnutých atribútov. Uvažujme preto trénovaciu množinu dát pozostávajúcu z anotovaných sekvencií spĺňajúcich podmienky opísané v časti 4.2.1. V prvom kroku boli tie sekvencie, ktoré predstavujú prekurzory neuropeptidov, použité na natrénovanie modelu predikcie miest štiepenia M, ktorého tvorba bola popísaná v časti 3.2. Vzhľadom na to, že ostatné sekvencie predstavujú zmes rôznych proteínov, ktoré vo všeobecnosti nezdieľajú rovnaké vlastnosti, rozhodli sme sa pre ne netrénovať nový model na predikciu štiepenia, ale použiť model M pri tvorbe všetkých atribútov využívajúcich informáciu o miestach štiepenia.

Po vytvorení modelu M bolo následne nutné vytvoriť i ostatné informácie potrebné pre použitie všetkých vytvorených atribútov modelu CRF. Z trénovacích dát boli vytvorené aproximácie distribúcií jednotlivých aminokyselín v špecificky anotovaných segmentoch, teda jedna aproximácia pre každú uvažovanú anotáciu. Podobne boli vytvorené aj aproximácie distribúcií jednotlivých skupín aminokyselín, pričom bolo použité rozdelenie aminokyselín do skupín na základe tabuľky 1.1, uvedenej v časti venujúcej sa opisu základných biologických pojmov.

Pre použitie atribútov g_{length_v} zachytávajúcich informáciu o dĺžke jednotlivých segmentov, je nutné zvoliť parametre min_v a max_v. Vzhľadom na pozorovania výskumnej skupiny Clynen a kol. [9], sme sa rozhodli pre segmenty neuropeptidov použiť hodnoty $min_N = 5$ a $max_N = 60$. O niečo vyššiu hodnotu min_N sme použili z dôvodu,

že nami anotované segmenty neuropeptidov nepredstavujú len neuropeptidy, ale ich o niečo dlhšie verzie (viď 4.2.1). Tieto hodnoty taktiež dobre opisujú štandardné dĺžky signálnych peptidov, a preto boli zvolené aj pre hodnoty min_S a max_S [18]. Nakoľko o dĺžkach inak anotovaných segmentov nemáme dostatok informácií, použili sme rovnaké hodnoty pre všetky anotácie. Ak náhodou existuje anotácia, ktorej štandardné dĺžky sú v protiklade s týmito hodnotami, potom sa očakáva, že natrénovaním modelu CRF budú prislúchajúcim atribútom priradené váhy blízke nule, a teda výsledný model je ekvivalentný takému, ktorý tieto atribúty nevyužíva. Preto by nemalo zvolenie rovnakých hodnôt min_v (resp.max_v) pre všetky anotácie v výrazne zhoršiť výsledné anotácie modelu CRF.

Samotná časť anotácie sekvencií následne pracuje jednak so sekvenciou, ale i vytvorenými aproximáciami rozdelení aminokyselín potrebných atribútmi g_{emit_v} a g_{emitg_v}. Každej sekvencii aminokyselín $\mathbf{X}$ sú taktiež priradené dve sekvencie rovnakej dĺžky. Prvá z týchto sekvencií predstavuje výstup programu SignalP, použitého na identifikáciu signálneho peptidu v $\mathbf{X}$. Táto sekvencia pozostáva z reálnych čísel opisujúcich skóre jednotlivých pozícií uvažovanej sekvencie a bola v predchádzajúcom texte označovaná $\mathbf{signal}(\mathbf{X})$. Druhá z priradených sekvencií je zase výstupom natrénovaného modelu M prislúchajúcej sekvencie, teda sekvenciou reálnych čísel $\mathbf{cleavage}^M(\mathbf{X})$, ktorá je používaná atribútmi definovanými v časti 4.2.2. Všetky tieto informácie sú následne použité v kombinácii so zvolenou knižnicou CRF, čoho výsledok je natrénovanie modelu určeného na anotáciu. V prípade samotnej inferencie, a teda anotácie testovacej sekvencie, je opäť použitý ten istý model M i všetky požadované aproximácie rozdelení, pričom je aj tvorba spomínaných dvoch pomocných sekvencií rovnaká ako v prípade trénovania modelu.

4.3.2 Výpočtová zložitosť a jej zlepšenie

Trénovanie modelu CRF je vo všeobecnosti časovo veľmi náročný iteratívny proces, pri ktorom sa hľadá optimálne priradenie váh jednotlivým lokálnym atribútom (viď popis v časti 4.1). Za účelom zlepšenia výpočtovej zložitosti jedného kroku tohto procesu sme sa rozhodli modifikovať spôsob výpočtu niektorých atribútov. Pri implementácii atribútov $g_{nocleave_v}$, g_{aroma_v}, g_{emit_v} a g_{emitg_v} priamočiarym spôsobom podľa ich definícií v časti 4.2.2, a teda nutnosti prechádzania hodnôt prislúchajúcich segmentov, je zložitosť výpočtu jedného atribútu lineárna od dĺžky segmentu. Takže pri sekvencii dĺžky n je zložitosť výpočtu spomenutých atribútov $\mathrm{O}(n)$ a nie $\mathrm{O}(1)$, ako sa štandardne pri dokazovaní výpočtovej zložitosti trénovania, ale i inferencie, uvažuje. Z tohto dôvodu sme ešte pred vykonaním výpočtov CRF predspracovali vstupné dáta sekvencie a modifikovali výpočty atribútov tak, aby pracovali s takto predspracovanými dátami, pričom výsledné hodnoty atribútov zostali nezmenené a ich výpočet bol realizovateľný v konštantnom čase $\mathrm{O}(1)$. Samotné predspracovanie vstupných dát spočíva vo vytvo-

rení prefixových súm

- sekvencie hodnôt predikcie štiepenia

$$\mathrm{PS}_i^{\mathrm{cleave}}(\mathbf{cleavage}^M(\mathbf{X})) = \sum_{j=1}^{i} cleavage_j^M(\mathbf{X})$$

- sekvencie výskytu aromatickej aminokyseliny

$$\mathrm{PS}_i^{\mathrm{aroma}}(\mathbf{X}) = \sum_{j=1}^{i} [x_i \text{ je aromatická aminokyselina}]$$

- sekvencií log-pravdepodobností jednotlivých anotovaných aminokyselín

$$\mathrm{PS}_i^{\mathrm{emit}_v}(\mathbf{X}) = \sum_{j=1}^{i} \log \Pr(x_j|v)$$

- sekvencií log-pravdepodobností jednotlivých anotovaných skupín

$$\mathrm{PS}_i^{\mathrm{emitg}_v}(\mathbf{X}) = \sum_{j=1}^{i} \log \Pr(group(x_j)|v)$$

Výpočty jednotlivých atribútov $g_{nocleave_v}$, g_{aroma_v}, g_{emit_v} a g_{emitg_v} vieme jednoducho nahradiť výpočtami využívajúcimi tieto prefixové sumy, za využitia vlastnosti

$$\sum_{i=b}^{e} \mathrm{A}_i = \mathrm{PS}_e - \mathrm{PS}_{b-1},$$

kde $\mathbf{A} = (\mathrm{A}_1, \mathrm{A}_2, \cdots, \mathrm{A}_n)$ predstavuje sekvenciu čísel s prefixovými sumami $\mathbf{PS} = (\mathrm{PS}_1, \mathrm{PS}_2, \cdots, \mathrm{PS}_n)$, pričom $\mathrm{PS}_0 = 0$.

Po tejto modifikácii výpočtu atribútov je časová zložitosť výpočtu všetkých uvažovaných atribútov konštantná, čím bola výrazne zlepšená i zložitosť samotného trénovania i inferencie.

4.3.3 Identifikácia prekurzorov neuropeptidov

Náš systém na rozpoznávanie neuropeptidov možno použiť aj na riešenie jednoduchšieho problému identifikácie prekurzorov neuropeptidov. Ide teda o problém, pri ktorom sa nezaujímame o konkrétnu anotáciu skúmanej sekvencie, ale úlohou je len zistiť, či je daná sekvencia prekurzorom neuropeptidov. Táto informácia sa dá ľahko zistiť priamo z predikovanej anotácie modelom CRF. Budeme hovoriť, že anotácia je pozitívna, ak prislúcha k anotácii prekurzoru neuropeptidov. V opačnom prípade ju budeme označovať ako negatívnu. Avšak nevýhodou tohto prístupu je, že nám neposkytuje žiadnu informáciu o spoľahlivosti identifikácie, nakoľko každú skúmanú sekvenciu striktne priradí do jednej z dvoch skupín, a to buď, že je prekurzorom neuropeptidov alebo nie

(angl. hard assignment). Navrhli a zrealizovali sme preto postup, pri ktorom by bolo možné jemné priradenie (angl. soft assignment).

Normalizačná konštanta $Z_{\mathbf{w}}(\mathbf{X})$ zo vzťahu (4.5) predstavuje súčet skóre všetkých anotácií danej sekvencie $\mathbf{X}$ v natrénovanom modeli, t. j. pri váhovaní globálnych atribútov $\mathbf{w}$. Pri inferencii sa hľadá optimálna anotácia, a teda anotácia s najväčším skóre. Z definície tejto hodnoty skóre vyplýva, že pri veľmi nízkej hodnote váhy prislúchajúcej k atribútu (potenciálne nekonečne zápornej) a nezápornej hodnote tohto atribútu, je celkové skóre tejto anotácie nulové, resp. veľmi nízke. Ak preto po natrénovaní modelu zmeníme získané váhy prislúchajúce atribútom popisujúcim prechody z anotácie S do anotácií N a P, teda váhy atribútov $g_{edge_{S,N}}$ a $g_{edge_{S,P}}$, na veľmi nízke hodnoty, bude hodnota normalizačnej konštanty $Z_{\mathbf{w}'}(\mathbf{X})$ rovná súčtu skóre všetkých anotácií, ktoré neobsahujú tieto prechody, a teda predstavujú negatívne anotácie sekvencie. Taktiež preto platí, že podiel R_{neg} týchto dvoch hodnôt približne predstavuje prínos negatívnych anotácií do celkovej hodnoty skóre všetkých anotácií sekvencie $\mathbf{X}$, konkrétne

$$R_{neg} = \frac{Z_{\mathbf{w}'}(\mathbf{X})}{Z_{\mathbf{w}}(\mathbf{X})}. \tag{4.20}$$

Ak je skúmaná sekvencia prekurzorom neuropeptidov, potom očakávame, že prínos pozitívnych anotácií do celkového skóre bude vyšší než prínos anotácií negatívnych, a teda, že hodnota R_{neg} bude nízka. Totiž, ak je optimálna anotácia pozitívna, potom jednak ona prispieva značne veľkou časťou do celkového skóre a zároveň je očakávateľné, že taktiež veľa suboptimálnych anotácií bude pozitívnych. Z týchto dôvodov je tento pomer, resp. hodnotu $R_{pos} = 1 - R_{neg}$, možné použiť ako ukazovateľ spoľahlivosti identifikácie prekurzorov neuropeptidov, pričom je taktiež možné zvoliť konkrétnu prahovú hodnotu, na základe ktorej sa vykoná rozhodnutie o tom, či skúmaná sekvencia je, alebo nie je takýmto prekurzorom.

Nami použitá knižnica CRF nám umožňuje pre jednotlivé sekvencie získať nielen optimálne anotácie, ale i hodnotu $Z_{\mathbf{w}}(\mathbf{X})$, a preto sme boli schopní túto metódu identifikácie prekurzorov použiť a testovať jej presnosť (viď časť 4.5.5).

4.4 Tvorba anotovaných dát

Jednotlivé časti nášho systému na rozpoznávanie neuropeptidov boli implementované, no k samotnému vytvoreniu je nutné trénovanie na správne anotovaných dátach. Tieto dáta by mali jednak pozostávať z anotovaných sekvencií prekurzorov neuropeptidov, ktoré nazývame *pozitívne dáta*, nakoľko práve tieto sekvencie sú tie, ktoré má náš systém za cieľ rozpoznať a následne anotovať. Keďže má byť náš systém schopný rozpoznávať tieto sekvencie, musia zvolené trénovacie dáta obsahovať taktiež sekvencie, ktoré prekurzormi neuropeptidov nie sú. Tieto sekvencie nazývame *negatívne dáta*. Vzhľadom na nami zvolenú topológiu modelu, ktorá bola popísaná v časti 4.2.1, mu-

sia taktiež všetky anotácie vybraných sekvencií korešpondovať so zvoleným stavovým diagramom z obrázku 4.2b.

Vzhľadom na to, že v súčasnosti neexistuje žiadna databáza anotovaných sekvencií, ktoré by sme mohli priamo použiť ako naše pozitívne alebo negatívne dáta, boli sme nútení si ich vyrobiť sami pomocou rozsiahlej databázy UniProt [24]. Táto databáza pozostáva z anotovaných proteínových sekvencií, vytvorených ručne na základe literatúry a experimentov, alebo automatizáciou. V nej dostupné anotácie predstavujú súbor rôznych informácií o funkciách, vlastnostiach a štruktúre daného proteínu, ktoré sú doposiaľ známe. Ukážku takejto anotácie uvádzame na obrázku 4.3.

Z databázy UniProt sme pomocou dotazov uvedených na obrázku 4.4 získali záznamy, na základe ktorých sme vytvorili dve množiny anotovaných dát postupom, ktorý je opísaný v nasledujúcich častiach. Prvú z vytvorených množín anotovaných dát nazývame *Arthropoda* a obsahuje sekvencie živočíšneho kmeňa článkonožcov (angl. Arthropoda). Druhá vytvorená množina dát je rozšírením prvej a obsahuje sekvencie celej ríše živočíchov, nazývame ju *Metazoa* (angl. Metazoa).

4.4.1 Pozitívne dáta

Pozitívne dáta množiny *Arthropoda* boli získané z databázy UniProt a následným filtrovaním. V prvom kroku sme z databázy získali všetky proteínové sekvencie prislúchajúce organizmom patriacim do kmeňa článkonožcov, ktoré mali v svojom popise molekulárnej funkcie uvedený popis „Neuropeptide“, pričom sme ignorovali tie s popisom „Receptor“, nakoľko receptory neuropeptidov nie sú sekvenciami prekurzorov neuropeptidov. Konkrétny použitý dotaz na databázu uvádzame na obrázku 4.4.

Výsledných 1340 sekvencií však obsahuje veľké množstvo rovnakých alebo veľmi podobných sekvencií, a preto sme ich zoskupili do zhlukov na základe ich vzájomnej podobnosti. K tomuto účelu sme použili program CD-HIT [25], pričom do jedného zhluku boli priradené sekvencie so vzájomnou podobnosťou minimálne 90%. Týmto postupom sme získali 430 zhlukov, z ktorých sme z každého vybrali len jedného reprezentanta, ku ktorému existuje v databáze UniProt najviac údajov. Záznamy 430-tich sekvencií sme následne filtrovali a vyhodili tie, ktoré buď nepredstavovali sekvencie prekurzorov neuropeptidov, alebo ich nebolo možné na základe dostupných údajov anotovať. Takto vyradenými sekvenciami boli také, ktoré napríklad neobsahovali (alebo nemali anotovaný) signálny peptid, alebo neobsahovali (alebo nemali anotovaný) žiaden neuropeptid, alebo ktoré neobsahovali vôbec žiadnu anotáciu. Za neuropeptid sme v dostupných anotáciách databázy UniProt považovali len tú časť sekvencie, ktorá bola anotovaná heslom „peptide“. Vzhľadom na nami uvažovanú sémantiku anotácie N sme tieto časti rozšírili o prípadný glycín a bázické aminokyseliny lyzín a arginín (viď sekcie 1.4 a 4.2.1). Všetky uvažované sekvencie, ktorých anotácie boli nekompletné alebo

```
ID   HUGIN_DROME             Reviewed;         191 AA.
AC   Q9VG55;
DT   19-SEP-2002, integrated into UniProtKB/Swiss-Prot.
DT   01-MAY-2000, sequence version 1.
DT   16-APR-2014, entry version 100.
...
OS   Drosophila melanogaster (Fruit fly).
OC   Eukaryota; Metazoa; Ecdysozoa; Arthropoda; Hexapoda; Insecta;
OC   Pterygota; Neoptera; Endopterygota; Diptera; Brachycera; Muscomorpha;
OC   Ephydroidea; Drosophilidae; Drosophila; Sophophora.
...
DR   GO; GO:0005179; F:hormone activity; NAS:FlyBase.
DR   GO; GO:0016084; F:myostimulatory hormone activity; IDA:FlyBase.
DR   GO; GO:0071855; F:neuropeptide receptor binding; IPI:FlyBase.
DR   GO; GO:0005102; F:receptor binding; ISS:FlyBase.
DR   GO; GO:0018990; P:ecdysis, chitin-based cuticle; IMP:UniProtKB.
DR   GO; GO:0030536; P:larval feeding behavior; IMP:FlyBase.
DR   GO; GO:0007218; P:neuropeptide signaling pathway; NAS:UniProtKB.
PE   1: Evidence at protein level;
KW   Amidation; Cleavage on pair of basic residues; Complete proteome;
KW   Direct protein sequencing; Neuropeptide; Reference proteome; Secreted;
KW   Signal.
FT   SIGNAL        1     24
FT   PROPEP       25    119
FT                                /FTId=PRO_0000029908.
FT   PEPTIDE     121    137       Hug-gamma (Potential).
FT                                /FTId=PRO_0000029909.
FT   PEPTIDE     140      ?       Hug-peptide (Potential).
FT                                /FTId=PRO_0000029910.
FT   PROPEP        ?    171
FT                                /FTId=PRO_0000029911.
FT   PEPTIDE     174    181       PK-2.
FT                                /FTId=PRO_0000029912.
FT   PROPEP      185    191
FT                                /FTId=PRO_0000029913.
FT   MOD_RES     137    137       Leucine amide (Potential).
FT   MOD_RES     181    181       Leucine amide.
SQ   SEQUENCE   191 AA;  20623 MW;  78F4A9811C57D932 CRC64;
     MCGPSYCTLL LIAASCYILV CSHAKSLQGT SKLDLGNHIS AGSARGSLSP ASPALSEARQ
     KRAMGDYKEL TDIIDELEEN SLAQKASATM QVAAMPPQGQ EFDLDTMPPL TYYLLLQKLR
     QLQSNGEPAY RVRTPRLGRS IDSWRLLDAE GATGMAGGEE AIGGQFMQRM VKKSVPFKPR
     LGKRAQVCGG D
```

Obr. 4.3: Ukážka časti anotácie získaná z databázy UniProt. Anotácia prislúcha sekvencii s označením Q9VG55, získanej z octovej mušky (Drosophila melanogaster), ktorá bola získaná dotazom pre tvorbu pozitívnych dát množiny *Arthropoda*. Riadky začínajúce označením FT popisujú jednotlivé časti sekvencie. Uvedená sekvencia pozostáva zo signálneho peptidu, úsekov neuropeptidov (PEPTIDE) a úsekov propeptidov (PROPEP), ale nebola vybraná do našej množiny dát, nakoľko nie je presne známy koniec jej druhého neuropeptidu (symbol ?).

```
A: keyword:"Neuropeptide [KW-0527]" AND taxonomy:"Arthropoda [6656]"
   NOT keyword:"Receptor [KW-0675]"

B: keyword:"Neuropeptide [KW-0527]" AND taxonomy:"Metazoa [33208]"
   NOT keyword:"Receptor [KW-0675]"

C: NOT keyword:"Neuropeptide [KW-0527]" AND taxonomy:"Arthropoda [6656]"
   AND keyword:"Reference proteome [1185]" AND reviewed:yes

D: NOT keyword:"Neuropeptide [KW-0527]" AND taxonomy:"Metazoa [33208"
   AND keyword:"Reference proteome [1185]" AND reviewed:yes
```

Obr. 4.4: Dotazy na databázu UniProt použité pri tvorbe anotovaných dát. Dotazy A a B boli použité pri tvorbe pozitívnych dát množín Arthropoda a Metazoa. Pri tvorbe negatívnych dát boli použité dotazy C a D.

viedli k nejednoznačnosti[4], boli taktiež odstránené. Týmto filtrovaním sme z pôvodných 1340 sekvencií získali 76 dobre anotovaných sekvencií prekurzorov neuropeptidov.

Z dôvodu malého počtu takto získaných dát sme podobným postupom zostrojili aj väčšiu množinu pozitívnych dát množiny *Metazoa*, ktorá bola získaná z proteínových sekvencií všetkých organizmov patriacich do ríše živočíchov (teda aj článkonožce). Z databázy UniProt bolo podobným dotazom ako v predchádzajúcom prípade získaných 2029 sekvencií. Tieto sekvencie sme následne zoskupili do zhlukov a získali 841 reprezentantov, z ktorých prešlo naším filtrovaním a anotovaním 175 sekvencií.

4.4.2 Negatívne dáta

Negatívne dáta pozostávajú z anotovaných sekvencií, ktoré nie sú prekurzormi neuropeptidov. Nakoľko sme sa rozhodli použiť topológiu modelu CRF opísanú v časti 4.2.1, bolo možné tieto dáta zostrojiť spôsobom podobným tomu, ktorým boli zostrojené aj pozitívne dáta. Pri tvorbe negatívnych dát množiny *Arthropoda* sme uvažovali všetky proteínové sekvencie článkonožcov, ktoré nemali v popise ich molekulárnej funkcie heslo „Neuropeptide“. Z dôvodu zvýšenia kvality týchto dát sme taktiež požadovali, aby tieto sekvencie pochádzali z referenčných organizmov a boli označené ako „reviewed“, čím sme odfiltrovali tie, ktoré doposiaľ neboli dostatočne skúmané (viď obrázok 4.4). Takto sme získali 4297 sekvencií, ktoré sme, rovnako ako v prípade pozitívnych dát, zoskupili a získali 4013 sekvencií. Väčšina z týchto sekvencií však nespĺňa kritéria kladené naším stavovým diagramom modelu CRF 4.2b, nakoľko väčšina napríklad neobsahuje signálny peptid. Použitím rovnakého filtrovania, ako v prípade pozitívnych dát, sme preto získali len 35 kvalitných anotovaných sekvencií, ktorých anotácie korešpondujú s nami navrhnutým diagramom.

[4]Napríklad, ak boli viacerým prekrývajúcim úsekom priradené rôzne anotácie.

Rovnakým postupom boli vytvorené i negatívne dáta množiny *Metazoa*, pričom bol použitý obdobný dotaz na databázu UniProt. Z výsledných 74446 sekvencií bolo pomocou programu CD-HIT získaných 50140 sekvencií, ktoré prešli následným filtrovaním. Takto vznikla množina negatívnych dát množiny *Metazoa* pozostávajúca z 255 sekvencií.

4.5 Experimenty

V tejto kapitole sa budeme venovať niektorým experimentom, ktorými sme testovali náš systém na rozpoznávanie neuropeptidov, popísaný v predchádzajúcich častiach. Vzhľadom na pomerne malé množstvo anotovaných dát *Arthropoda* i *Metazoa* sme sa rozhodli nerozdeliť tieto dáta na trénovacie a testovacie, ale zvolili sme iný, štandardne používaný, postup krížovej validácie, pričom sme celú množinu dát rozdelili na 5 častí s rovnakým počtom sekvencií (angl. 5-fold cross-validation). Experiment bol následne vykonaný tak, že jedna časť dát bola vyhradená ako testovacia a na zvyšných štyroch prebiehalo trénovanie modelu. Tento postup bol vykonaný päťkrát, pričom vždy bola za testovaciu množinu vybraná iná časť. Metódou krížovej validácie zistíme výslednú úspešnosť celého experimentu na základe úspešnosti jednotlivých piatich experimentov. Podotýkame, že v priebehu jednotlivých experimentov nebola zvolená testovacia časť nijako použitá počas trénovania systému. Jednotlivé informácie potrebné k fungovaniu nášho systému boli zostavené len výlučne z ostatných štyroch častí. Preto aj samotné trénovanie modelu SVM určenému na predikciu miest štiepenia, ktoré pozostáva z nájdenia optimálnych hyper-parametrov pomocou krížovej validácie, nijakým spôsobom nevyužívalo informácie zo sekvencií testovacej časti (viď sekciu 3.2.2).

V našich experimentoch sme používali dva rôzne typy modelov CRF. Prvý, *kompletný model*, je určený na anotáciu sekvencií za účelom riešenia problému rozpoznávania neuropeptidov, a teda je založený na topológii znázornenej stavovým diagramom na obrázku 4.2b. Druhý z použitých modelov je založený na topológii znázornenej na obrázku 4.1b, a preto je určený na anotáciu sekvencií prekurzorov neuropeptidov. Využitie tohto modelu preto spočíva v identifikácii jednotlivých neuropeptidov v skúmanom prekurzore a budeme ho označovať ako *redukovaný model*.

4.5.1 Porovnanie použitých atribútov

Ako prvý z experimentov uvádzame porovnanie úspešnosti anotácie prekurzorov neuropeptidov použitím viacerých podmnožín nami implementovaných atribútov modelu CRF. Úlohou tohto experimentu bolo zistiť, či boli jednotlivé atribúty zvolené tak, aby pridávali nášmu systému silu. Na tento účel postačuje uvažovať len pozitívne dáta, a preto sme zvolili náš redukovaný model. Naším cieľom bolo taktiež len merať schopnosť učenia nášho systému, a nie nutne generalizácie, a preto sme za trénovacie i tes-

Použité atribúty	**S**		**P**		**N**		**Prk**
	Ppr	Sen	Ppr	Sen	Ppr	Sen	
Aroma	99,46	97,59	64,31	56,53	67,72	74,83	**72,41**
Emit	99,30	98,29	81,61	85,56	87,43	84,32	**87,40**
Aroma, Emit	99,46	98,29	81,44	86,18	87,90	84,12	**87,53**
Aroma, Emit, EmitG	99,46	98,29	81,05	87,47	88,80	86,05	**87,71**
Celý model	99,46	98,66	83,44	85,83	88,00	86,21	**88,41**

Tabuľka 4.1: Porovnanie úspešnosti anotácie použitím rôznych atribútov. Prvý stĺpec popisuje použité atribúty, pričom všetky uvedené modely využívali taktiež všetky atribúty skupín: obmedzenia topológie modelu, informácia o signálnom peptide a informácia o miestach štiepenia. Stĺpce S, P a N označujú hodnoty pre anotácie signal, propeptid a neuropeptid. Posledný stĺpec popisuje celkovú presnosť klasifikácie príslušného modelu. Ppr = pozitívna presnosť, Sen = senzitivita, Prk = presnosť klasifikácie.

tovacie dáta zvolili rovnakú množinu dát, a to pozitívne dáta množiny *Arthropoda*. V tabuľke 4.1 uvádzame porovnanie úspešnosti anotácie niekoľkých modelov CRF používajúcich rôzne atribúty. Všetky uvedené modely používali všetky atribúty skupín: obmedzenia topológie modelu, informácia o signálnom peptide a informácia o miestach štiepenia, nakoľko tie jednoznačne predstavujú neoddeliteľnú súčasť nášho modelu CRF a ich použitie určite prispieva k zlepšeniu. Z výsledkov presnosti klasifikácie jednotlivých modelov vyplýva, že všetky zvolené atribúty prispievajú k zvýšeniu presnosti klasifikácie, a preto boli všetky ponechané. V tabuľke neuvádzame výsledky anotácie miest štiepenia C, nakoľko tieto miesta boli vo všetkých prípadoch anotované správne.

4.5.2 Anotácia sekvencií

Náš systém, ktorý využíval všetky implementované atribúty popísané v časti 4.2.2, sme testovali metódou krížovej validácie. Nakoľko nám nie je známy žiaden iný prístup na riešenie problému rozpoznávania neuropeptidov, nemôžeme správnosť našich predikcií porovnať s inými metódami. Porovnávali sme preto len úspešnosti predikcií nášho systému na základe viacerých dát. Testovali sme jednak schopnosť anotácie prekurzorov neuropeptidov redukovaným modelov CRF na pozitívnych dátach množín *Arthropoda* a *Metazoa* a taktiež náš kompletný systém na rozpoznávanie neuropeptidov na celých dátach *Arthropoda* a *Metazoa*. V tabuľke 4.2 uvádzame presnosti klasifikácie modelov v jednotlivých experimentoch, spolu s mierou F_1 pre jednotlivé anotácie. Miera F_1 predstavuje kombináciu senzitivity (Sen) a pozitívnej presnosti (Ppr), pričom je vyjadrená v tvare

$$F_1 = 2 \cdot \frac{\mathrm{Sen} \cdot \mathrm{Ppr}}{\mathrm{Sen} + \mathrm{Ppr}}.$$

Ako bolo očakávané, nakoľko úlohou redukovaného modelu je len anotovať sekvencie prekurzorov, a nie všeobecnejších sekvencií, dosahoval tento model lepšiu presnosť

Model	S	P	N	C	$\overline{\text{P}}$	X	$\overline{\text{C}}$	Prk
	F_1	F_1	F_1	F_1	F_1	F_1	F_1	
Arthropoda_p	95,58	78,56	81,29	88,57	-	-	-	**82,82**
Metazoa_p	96,02	71,50	71,96	88,67	-	-	-	**75,86**
Arthropoda	98,08	64,14	75,95	78,48	34,66	56,51	53,70	**70,36**
Metazoa	95,89	53,96	65,28	81,24	72,39	56,20	48,61	**69,03**

Tabuľka 4.2: Výsledky presnosti anotácie jednotlivých modelov. Riadky Arthropoda_p a Metazoa_p predstavujú výsledky krížovej validácie redukovaného modelu trénovaného a testovaného na sekvenciách prekurzorov neuropeptidov, t. j. pozitívnych dátach. Riadky označené Arthropoda a Metazoa predstavujú výsledky krížovej validácie kompletného modelu za použitia všetkých sekvencií príslušných množín dát. Posledný stĺpec znázorňuje presnosť klasifikácie jednotlivých modelov, ostatné stĺpce znázorňujú hodnotu F_1 pre jednotlivé anotácie.

klasifikácie než jeho zložitejšia verzia, pri ktorej je nutná aj identifikácia prekurzoru. Z tabuľky taktiež vidíme, že presnosť klasifikácie redukovaného modelu na dátach *Arthropoda* je výrazne vyššia než na väčšej množine dát *Metazoa*. Toto je pravdepodobne spôsobené tým, že táto väčšia množina pozostáva zo sekvencií veľkej skupiny organizmov, ktoré sa od seba už príliš líšia. V prípade menšej skupiny článkonožcov, ktoré sú si podobné, je preto problém naučenia sa anotácie jednoduchší. Na druhej strane, pri porovnaní celkovej presnosti klasifikácie jednotlivých experimentov použitím kompletného modelu sme výrazný rozdiel nezaznamenali. Tento malý rozdiel je zjavne dôsledkom zlej úspešnosti anotácie dát *Arthropoda*, pri ktorej sme častejšie pozorovali chybu, kedy bola pozitívnou anotáciou anotovaná negatívna sekvencia (nie prekurzor neuropeptidov). Pozorovaním jednotlivých anotácií sme zistili, že náš systém mal v tomto prípade problém zistiť, že je skúmaná sekvencia prekurzorom neuropeptidov, pričom z predikovaných negatívnych anotácií bolo až vyše 30% prekurzorov neuropeptidov, ktoré nedokázal rozpoznať (v prípade *Metazoa* len cca 22%).

Čo sa týka presnosti predikcií jednotlivých anotácií, tak z výsledkov vidíme, že presnosť predikcií miest štiepenia v prekurzoroch neuropeptidov je výrazne vyššia než predikcia miest štiepenia v negatívnych sekvenciách. Toto pozorovanie je pravdepodobne dôsledkom použitia rovnakého modelu SVM na predikciu miest štiepenia v oboch prípadoch (anotácia C a $\overline{C}$). Tieto miesta preto zrejme nezdieľajú rovnaké vlastnosti a bolo by dobré pre ne použiť rôzny model SVM.

4.5.3 Predikcia miest štiepenia

Náš systém na rozpoznávanie neuropeptidov sme taktiež testovali na schopnosť predikcie miest štiepenia prekurzorov neuropeptidov, pričom sme použili pozitívne dáta množín *Arthropoda* a *Metazoa*. Prostredníctvom metódy krížovej validácie sme naším redukovaným systémom získali anotácie jednotlivých sekvencií prekurzorov neuro-

	Arthropoda			Metazoa			
	SVM	CRF	NeuroPred	SVM	CRF	NeuroPred	NeuroPred M
Prk	99,58	99,53	99,14	99,40	99,34	98,95	98,44
Sen	87,35	83,79	75,10	87,86	84,51	79,65	79,19
Špe	99,86	99,88	99,68	99,77	99,81	99,55	99,04
Ppr	93,25	94,22	84,07	92,12	93,12	84,75	72,03
Npr	99,72	99,64	99,44	99,62	99,52	99,37	99,35

Tabuľka 4.3: Výsledky porovnania úspešnosti predikcie miest štiepenia nášho systému a programu NeuroPred na pozitívnych dátach množín *Arthropoda* a *Metazoa*. Stĺpce SVM a CRF predstavujú úspešnosti predikcií na základe nášho modelu predikcie miest štiepenia SVM a modelu rozpoznávania neuropeptidov CRF. Stĺpce NeuroPred (resp. NeuroPred M) prislúchajú predikciám modelu NeuroPred pri použití modelu Insect (resp. Mammalian). Prk = presnosť klasifikácie, Sen = senzitivita, Špe = špecificita, Ppr = pozitívna presnosť, Npr = negatívna presnosť.

peptidov a extrahovali z nich informáciu o anotáciách C, ktoré symbolizujú miesta štiepenia. Úspešnosť predikcií týchto miest sme následne porovnali s úspešnosťou predikcií programu NeuroPred, avšak nakoľko bol program NeuroPred trénovaný na iných dátach, nemôže byť toto porovnanie vykonané dôkladne. V prípade porovnania predikcií na dátach *Arthropoda* sme zvolili model NeuroPred Insect, ktorý bol trénovaný na sekvenciách hmyzu. Hoci predpokladáme, že zvolený model NeuroPred bol trénovaný i na niektorých sekvenciách *Arthropoda*, a teda na tých, na ktorých bol nami testovaný, napriek tomu náš systém dosiahol väčšinou porovnateľné výsledky, a v prípade senzitivity a pozitívnej presnosti dokonca výrazne lepšie výsledky[5]. Výsledky tohto porovnania uvádzame v tabuľke 4.3 spolu s porovnaním nášho systému so samotným modelom SVM, ktorého výstupy boli modelom CRF používané. Ako vidno, až na hodnotu senzitivity, náš systém dosahuje veľmi podobné výsledky. Na základe tohto pozorovania preto usudzujeme, že predikcie miest štiepenia modelu SVM sú výrazne najdôležitejšou informáciou o hraniciach segmentov, na základe ktorých model CRF segmenty vytvára. Ostatné použité atribúty teda nijako výrazne neovplyvňujú jednotlivé segmenty, ale len ich anotácie.

Podobné porovnanie sme vykonali aj pre pozitívne dáta množiny *Metazoa*. Nakoľko však webová aplikácia NeuroPred neposkytuje použitie modelu trénovaného na podobných dátach (dátach všetkých živočíchov), porovnávali sme predikcie jeho dvoch modelov trénovaných na sekvenciách hmyzu (Insect) a sekvenciách cicavcov (Mammalian). Predikcie týchto modelov, vykonané na dátach *Metazoa*, viedli k porovnateľným výsledkom, až na nižšiu hodnotu pozitívnej presnosti modelu Mammalian. V porovnaní s týmito predikciami náš systém dosahoval podobné výsledky ako v predchádzajúcom

[5]Podotýkame, že úspešnosť nášho modelu je vyjadrená z výsledkov krížovej validácie, a teda v našom prípade nebol model testovaný na žiadnej sekvencii, ktorá by sa vyskytovala aj v jeho trénovacej množine.

Zdroj hraníc	Model	S	P	N	C	$\overline{\mathbf{P}}$	X	$\overline{\mathbf{C}}$	Prk
		F_1	F_1	F_1	F_1	F_1	F_1	F_1	
Správne	Arthropoda_p	100	80,86	84,70	100	-	-	-	**86,14**
hodnoty	Arthropoda	100	77,91	80,46	95,16	50,78	58,55	74,23	**79,13**
SignalP	Arthropoda_p	95,58	78,56	81,29	88,57	-	-	-	**82,82**
+SVM	Arthropoda	98,08	64,14	75,95	78,48	34,66	56,51	53,70	**70,36**

Tabuľka 4.4: Porovnanie výsledkov anotácie za použitia správnej informácie o signálnom peptide a miestach štiepenia. Hodnota v prvom stĺpci určuje, či sa pri trénovaní i testovaní modelu použili informácie získané pomocou programu SignalP a modelu SVM na predikciu miest štiepenia, alebo boli použité presné správne hodnoty. Riadky Arthropoda_p predstavujú výsledky krížovej validácie redukovaného modelu trénovaného a testovaného na pozitívnych dátach *Arthropoda* a riadky označené Arthropoda predstavujú výsledky krížovej validácie kompletného modelu za použitia všetkých sekvencií týchto dát. Posledný stĺpec vyjadruje celkovú presnosť klasifikácie pozícií sekvencií.

porovnaní. Rovnako podobne dopadlo aj porovnanie predikcií získaných naším systémom a predikcií nášho modelu na predikciu miest štiepenia.

4.5.4 Anotácia segmentov sekvencií

Ďalší z experimentov mal za úlohu zistiť, aká je schopnosť nášho modelu anotovať jednotlivé segmenty. Za týmto účelom neboli modelu CRF dané informácie o signálnom peptide a miestach štiepenia na základe programu SignalP a nášho modelu SVM, ale presné informácie získané priamo zo správnych anotácií jednotlivých sekvencií. Konkrétne, informácia o signálnom peptide bola vyjadrená sekvenciou **signal**(**X**) obsahujúcou samé nulové hodnoty až na jednu jednotkovú, ktorá zodpovedala správnemu ukončeniu signálneho peptidu v **X**. Podobným spôsobom boli skonštruované aj sekvencie **cleavage**(**X**), pričom hodnoty 1 prislúchali miestam štiepenia. Tieto zdroje informácií o hraniciach jednotlivých segmentov sekvencií boli použité počas trénovania i testovania.

V tabuľke 4.4 uvádzame porovnanie úspešnosti tejto anotácie so štandardným nastavením, pri ktorom sa využívajú informácie programu SignalP a natrénovaného modelu SVM. Ako sme očakávali, úspešnosť predikcií týmto spôsobom je vyššia než štandardným spôsobom, čo je dôsledkom toho, že model CRF dokázal naplno využiť správnu informáciu o hraniciach segmentov. Správnou predikciou segmentov sa následne zvýšila i presnosť predikcie ich anotácií. Výsledky tohto experimentu taktiež naznačujú, nakoľko by dokonalejšie predikcie SignalP a miest štiepenia mohli pomôcť zvýšiť presnosť celého systému.

		Hard	Soft					
Arthropoda	**Sen**	86,8	94,7	90,8	85,5	80,3	75,0	69,7
	Špe	65,7	62,9	65,7	68,6	80,0	80,0	85,7
Metazoa	**Sen**	73,8	95,3	90,7	85,0	80,4	75,7	70,1
	Špe	63,6	65,6	72,8	81,5	83,4	88,1	90,7

Tabuľka 4.5: Tabuľka senzitivity (Sen) a k nej prislúchajúcej špecificity (Špe) identifikácie prekurzorov neuropeptidov prostredníctvom krížovej validácie na dátach *Arthropoda* a *Metazoa*. Stĺpec Hard označuje hodnoty striktnej identifikácie na základe optimálnej anotácie modelom CRF. Stĺpce označené ako Soft označujú hodnoty identifikácie využívajúcej informácie aj o suboptimálnych anotáciách, pri použití rôznych prahových hodnôt.

4.5.5 Identifikácia prekurzorov neuropeptidov

Ako bolo vysvetlené v časti 4.3.3, navrhnutý systém na rozpoznávanie neuropeptidov možno taktiež použiť pri riešení jednoduchšieho problému identifikácie prekurzorov neuropeptidov. Použitím priamočiarej metódy, ktorá rozhodne, či je daná sekvencia týmto prekurzorom len na základe optimálnej anotácie, sme na dátach *Arthropoda* úspešne identifikovali takmer 87% prekurzorov, no pomerne často boli sekvencie negatívnych dát chybne považované za pozitívne, čoho dôsledkom bola hodnota špecificity len necelých 66%. V prípade rozsiahlejších dát *Metazoa* sa dosiahla ešte nižšia miera senzitivity než v predchádzajúcom prípade, nakoľko bolo až vyše 40% predikovaných prekurzorov predikovaných nesprávne. V oboch prípadoch teda dochádzalo ku chybným identifikáciám prekurzorov, no v jednotlivých experimentoch sa, vzhľadom na rôzne pomery veľkostí pozitívnych a negatívnych dát, tento jav prejavil na iných mierach (viď sekciu 4.4).

V tabuľke 4.5 uvádzame porovnanie tohto prístupu identifikácie s postupom založeným na pomere normalizačných konštánt modelu CRF, umožňujúcim nastavenie prahovej hodnoty identifikácie (viď časť 4.3.3). Uvádzame niekoľko výsledkov úspešnosti anotácie použitím rôznych prahov. Z tohto porovnania vidno, že predikcie založené len na optimálnej anotácii dosahujú o niečo horšie výsledky ako tie založené aj na suboptimálnych anotáciách. Z tohto porovnania vyplýva, že náš model CRF priraďuje vyššie hodnoty skóre nielen optimálnej anotácii, ale aj anotáciám jej podobným, a preto možno túto informáciu využiť. Ďalšou jednoznačnou výhodou druhej z metód je možnosť nastavenia prahu identifikácie, resp. možnosť usporiadania jednotlivých sekvencií na základe im prislúchajúcej hodnoty R_{neg}. V praxi teda možno postupne laboratórne testovať jednotlivé predikcie prekurzorov od najpravdepodobnejších (najnižšia hodnota) k menej pravdepodobným (vyššia hodnota).

Záver

V tejto práci sme navrhli systém určený k riešeniu problému rozpoznávania neuropeptidov. Použitím nami vytvoreného systému sme schopní identifikovať proteíny, ktoré predstavujú prekurzory neuropeptidov, a zároveň ich jednotlivé úseky anotovať za účelom rozpoznania takých, z ktorých vznikajú konkrétne neuropeptidy. Náš systém je založený na modeli strojového učenia *conditional random fields* [13] (CRF), ktorý pri segmentácii a anotácii skúmanej sekvencie využíva viaceré druhy informácií. Tieto informácie môžu byť získané buď priamo zo sekvencie alebo použitím iných nástrojov, ako napríklad pomocou programu SignalP [18] na identifikáciu signálneho peptidu, ktorý bol použitý i v našej práci.

Jednou z dôležitých častí nášho systému je i využitie informácie o miestach štiepenia skúmanej sekvencie, ktorej získavaniu bola venovaná tretia kapitola práce. V tejto kapitole sme jednak popísali model strojového učenia *support vector machines* [11] (SVM), ale tiež opísali, ako bol v našom systéme využitý pri úlohe identifikácie miest štiepenia prekurzoru. Nami navrhnutý a natrénovaný model na predikciu štiepenia sme taktiež porovnali s aktuálne rozšírenou webovou aplikáciou NeuroPred [4] a zaznamenali podobné alebo v niektorých ohľadoch dokonca lepšie výsledky. V štvrtej kapitole našej práce sme sa venovali samotnému problému rozpoznávania neuropeptidov a popísali náš konkrétny návrh topológie a atribútov modelu CRF, ktorý bol na tento účel zvolený. V našej práci sme taktiež navrhli a popísali samotný proces vytvárania anotovaných dát, nakoľko v súčasnosti neexistuje vhodná databáza, ktorá by sa na náš účel dala použiť priamo. Po implementovaní nášho systému a vytvorení týchto dát sme vykonali niekoľko rôznych experimentov, ktoré boli realizované z dôvodu testovania nášho systému. Zaoberali sme sa jednak použitím systému pri riešení kompletného problému rozpoznávania neuropeptidov, ale i viacerými čiastkovými problémami, ako napríklad identifikáciou miest štiepenia a identifikáciou prekurzorov neuropeptidov.

Z našich experimentov vyplýva, že náš systém možno priamo použiť pri riešení viacerých otázok spojených s problémom rozpoznávania neuropeptidov, avšak vzhľadom na jeho návrh je taktiež veľký priestor na jeho rozšírenie, a tým zlepšenie jeho výsledkov. Jednou z možností je napríklad použitie rôznych modelov SVM na predikciu miest štiepenia v pozitívnych a negatívnych dátach, ktorých predikcie sú využívané niektorými atribútmi modelu CRF. Hoci negatívne dáta, na rozdiel od pozitívnych, nepred-

stavujú nejakú konkrétnu skupinu sekvencií, ktoré by mali zdielať spoločné vlastnosti, je možné, že použitie samostatného modelu SVM pre tieto sekvencie dokáže zlepšiť presnosť predikcie miest ich štiepenia, a tým aj presnosť celého systému. Avšak okrem tejto malej modifikácie existujúcich atribútov je taktiež možné implementovanie ďalších nových atribútov modelu CRF, ktoré by využívali aj iné druhy informácií a boli by pri anotácii užitočné. Takouto informáciou môže byť napríklad podobnosť určitého segmentu skúmanej sekvencie s nejakým už známym neuropeptidom. V súčasnosti taktiež existuje databáza tzv. *proteínových domén*, ktoré popisujú základnú štruktúru niektorých špecifických skupín proteínov. Na základe tejto databázy a vhodne vytvorených atribútov by bolo možné identifikovať niektoré segmenty skúmanej sekvencie ako segmenty spĺňajúce štruktúru domény proteínov s inou vlastnosťou než neuropeptid a toto pozorovanie následne využiť v podobe atribútov pri anotácii.

Literatúra

[1] Li C, Kim K. Neuropeptides. April 2008. *WormBook*, ed. The C. elegans Research Community, WormBook, `http://www.wormbook.org`.

[2] Veenstra JA. Proteolytic processing of arthropod regulatory peptide precursors. 2011. In *Progress in Neurobiology, manuscript.*

[3] Southey BR, Rodriguez-Zas SL, Sweedler JV. 2006. Prediction of neuropeptide prohormone cleavages with application to RFamides. In *Peptides*, vol. 27, p. 1087-1098.

[4] Southey BR, Amare A, Zimmerman TA, Rodriguez-Zas SL, Sweedler JV. July 2006. NeuroPred: a tool to predict cleavage sites in neuropeptide precursors and provide the masses of the resulting peptides. In *Nucleic Acids Res.*, vol. 34, Web Server issue, p. 267–272.

[5] Southey BR, Sweedler JV, Rodriguez-Zas SL. March 2008. Prediction of neuropeptide cleavage sites in insects. In *Bioinformatics*, vol. 24, no. 6, p. 815–825.

[6] Southey BR, Sweedler JV, Rodriguez-Zas SL. December 2008. A Python analytical pipeline to identify prohormone precursors and predict prohormone cleavage sites. In *Frontiers in Neuroinformatics*, vol. 2, no. 7.

[7] Ofer D, Linial M. March 2014. NeuroPID: A Predictor for Identifying Neuropeptide Precursors from Metazoan Proteomes. In *Bioinformatics*, vol. 30, no. 6.

[8] Delfino KR, Southey BR, Sweedler JV, Rodriguez-Zas SL. February 2010. Genome-wide census and expression profiling of chicken neuropeptide and prohormone convertase genes. In *Neuropeptides*, vol. 44 (1), p. 31–44.

[9] Clynen E a kol. 2010. Bioinformatic approaches to the identification of novel neuropeptide precursors. In *Peptidomics*, p 357-374.

[10] Xie F a kol. 2010. The zebra finch neuropeptidome: prediction, detection and expression. In *BMC Biology*, vol. 8:28.

[11] Cortes C, Vapnik V. 1995. Support-vector networks. In *Machine Learning*, vol. 20, no. 3.

[12] *LibSVM*. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[13] Lafferty J, McCallum A, Pereira F. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, p. 282–289.

[14] DeCaprio D, Vinson JP, Pearson MD, Montgomery P, Doherty M, Galagan JE. September 2007. Conrad: gene prediction using conditional random fields. In *Genome Res.*, vol 17 (9), p. 1389–98.

[15] Sarawagi S, Cohen WW. 2005. Semi-markov conditional random fields for information extraction. In *Advances in neural information processing systems*.

[16] Sarawagi S. Efficient inference on sequence segmentation models. June 2006. In *Proceedings of the 23rd international conference on Machine learning*, p. 793-800.

[17] Sarawagi S. *Conditional Random Fields*. `http://crf.sourceforge.net`.

[18] Bendtsen JD, Nielsen H, von Heijne G, Brunak S. 2004. Improved prediction of signal peptides: SignalP 3.0. In *Journal of Molecular Biology*, vol. 340, p. 783-795.

[19] Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z,Miller W, Lipman DJ. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. In *Nucleic Acids Res.*, vol. 25, p. 3389-3402.

[20] Notredame C, Higgins D, Heringa J. 2000. *T-Coffee: A novel method for multiple sequence alignments*. `http://www.ebi.ac.uk/Tools/msa/tcoffee/`.

[21] *Wise2*. `http://www.ebi.ac.uk/Tools/Wise2/`.

[22] Zvelebil M, Baum JO. 2007. *Understanding bioinformatics*. Garland Science. ISBN 0-8153-4024-9.

[23] *The Neuropeptide Database*. `http://www.neuropeptides.nl`.

[24] *UniProt*. `http://www.uniprot.org`.

[25] *CD-HIT*. `http://weizhong-lab.ucsd.edu/cd-hit`.

[26] *GenBank*. `http://www.ncbi.nlm.nih.gov/sites/entrez?db=nucleotide`.