VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

KLASIFIKACE MALÝCH NEKÓDUJÍCÍCH RNA

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. TOMÁŠ ŽIGÁRDI
AUTHOR

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

KLASIFIKACE MALÝCH NEKÓDUJÍCÍCH RNA

CLASSIFICATION OF SMALL NONCODING RNAS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. TOMÁŠ ŽIGÁRDI
AUTHOR

VEDOUCÍ PRÁCE — Ing. IVAN VOGEL
SUPERVISOR

BRNO 2015

Abstrakt

Tato diplomová práce opisuje návrh a implementaci nástroje pro klasifikaci rostlinných microRNA bez genomu. Použity jsou vlastnosti mature a star sekvencí v microRNA duplexech. Implementována metoda je založena na shlukování RNA sekvencí (nástrojem CD–HIT), hlavne pro redukci jejich počtu. Vybraní reprezentanti z jednotlivých shluků jsou klasifikováni použitím support vector machine. Výkonnost klasifikace je víc než 96% (na základe metody cross–validation, využitím trénovacích dat).

Abstract

This masters's thesis contains description of designed and implemented tool for classification of plant microRNA without genome. Properties of mature and star sequences in microRNA duplexes are used. Implemented method is based on clustering of RNA sequences (with CD–HIT) to mainly reduce their count. Selected representants from each clusters are classified using support vector machine. Performance of classification is more than 96% (based on cross-validation method using the training data).

Klíčová slova

malé nekódující RNA, miRNA, centrální dogma molekulární biologie, CD–HIT, next–generation sekvenování, zhlukování sekvencí, cross–validation, support vector machine, klasifikace rostlinných miRNA, duplex

Keywords

small noncoding RNAs, miRNA, central dogma of molecular biology, CD–HIT, next–generation sequencing, sequences clustering, cross–validation, support vector machine, classification of plant miRNA, duplex

Citace

Tomáš Žigárdi: Klasifikace malých nekódujících RNA, diplomová práce, Brno, FIT VUT v Brně, 2015

Klasifikace malých nekódujících RNA

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně pod vedením Ing. Ivana Vogela. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

. . . . . .
Tomáš Žigárdi
27. května 2015

Poděkování

Chcel by som poďakovať vedúcemu práce, Ing. Ivanovi Vogelovi, za jeho odbornú pomoc, poskytnuté rady, konzultácie, ochotu a čas, ktorý mi pri tvorbe práce venoval.

Obsah

Kapitola 1

Úvod

Táto práca sa zaoberá klasifikáciou malých nekódujúcich RNA (konkrétne microRNA) rastlín, bez znalosti ich genómu. Klasifikácia je založená na analýze vlastností sekvencií, ktoré tvoria microRNA.

Prvá kapitola obsahuje stručný opis centrálnej dogmy molekulárnej biológie, teda procesov ako replikácia, transkripcia a translácia DNA. Ďalej sú opísané nekódujúce RNA, konkrétne microRNA (miRNA) a small interfering RNA (siRNA), ich vznik, štruktúra a ich funkcia v organizmoch.

Nasleduje kapitola venovaná next–generation sekvenovaniu, teda spôsobu, akým je možné previesť sekvencie nukleotidov do digitálnej podoby. Priestor je venovaný hlavne metódam Ion Torrent Personal Genome Machine (PGM) a Illumina MiSeq. Nasleduje opis najpoužívanejších zhlukovacích algoritmov, ktoré slúžia na rozdelenie dát do skupín podľa ich vlastností.

Ďalej sú opísané konkrétne programy, ktoré slúžia na zhlukovanie sekvencií, konkrétne malých RNA sekvencií. Nasledujúca kapitola obsahuje postup, akým bol určený program, ktorý je použitý počas samotnej klasifikácie, výsledky tohto postupu a ich zhodnotenie.

Ďalej je opísaná hypotéza o využití komplementarity sekvencií, z ktorých sa skladá microRNA. Bol navrhnutý postup na klasifikovanie microRNA pomocou zhlukovania, overenie tejto hypotézy pomocou navrhnutých krokov a zhodnotenie tejto hypotézy.

Nasleduje schéma a opis navrhnutej a implementovanej metódy, ktorá využíva zhlukovanie na klasifikáciu a zároveň na zmenšenie počtu dát, ktoré je nutné spracovať. Opísané a analyzované sú aj vlastnosti microRNA, ktoré boli použité pri klasifikácii sekvencií nástrojom *support vector machine* (SVM). Nástroj je použitý na vybrané sekvencie, ktorým nebola určená príslušnosť k microRNA pomocou zhlukovania. Opísaný je aj postup vytvorenia trénovacích a vstupných dát pre SVM model a vytvorenie modelu.

Ďalšia kapitola obsahuje zistenie výkonnosti klasifikačného modelu pomocou metód *cross–validation* a *grid–search* a potvrdenie výkonnosti pomocou testovania na náhodne vybraných dátach. Nasledujúca časť sa zaoberá prácou s implementovanou aplikáciou, teda jej spustenie a použitie potrebných pomocných nástrojov.

V záverečnej kapitole sú opísané možnosti ďalšieho vývoja implementovanej aplikácie.

Kapitola 2

Centrálna dogma molekulárnej biológie a malé nekódujúce RNA

2.1 Centrálna dogma molekulárnej biológie

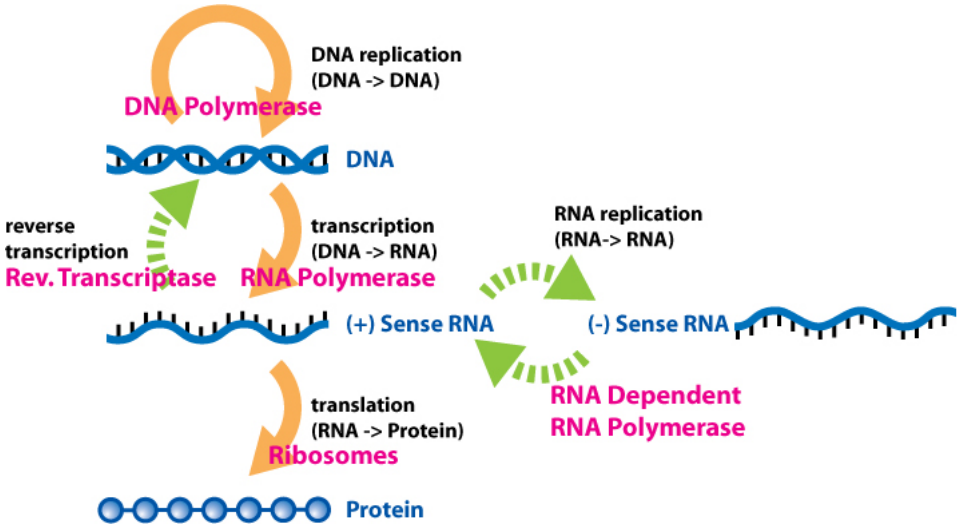
Centrálna dogma molekulárnej biológie opisuje spôsob prenosu genetickej informácie v živých organizmoch [3]. DNA (deoxyribonukleová kyselina) je pravotočivá dvojzávitnica obsahujúca kompletnú genetickú informáciu, ktorá definuje štruktúru a funkciu organizmov. Nachádza sa v jadre a je to prírodný polymér zložený z nukleotidov. Každý nukleotid pozostáva z troch zložiek:

- Fosfátový zvyšok kyseliny fosforečnej.
- Molekula deoxyrobózy.
- Dusíkatá báza: adenín (A), cytozín (C), guanín (G) a tymín (T).

Podľa genetického kódu, ktorý je uložený v DNA sa vytvárajú proteíny. Na šírení, dedení a úprave genetickej informácie z jednej podoby do druhej sa podieľajú tri hlavné procesy [3]:

- Replikácia: Dvojzávitnica DNA je rozdelená na dva samostatné reťazce. Ku každému sa vytvorí komplementárne vlákno. To zabezpečuje enzým DNA polymeráza, ktorá postupne prechádza vláknom a k nukleotidom vytvára ich komplement. Z jednej molekuly DNA takto vzniknú 2 identické molekuly DNA.
- Transkripcia: Informácia obsiahnutá v DNA sa prepisuje do poradia nukleotidov RNA (ribonukleová kyselina). Tento proces ja rovnako ako replikácia, založený na párovaní komplementárnych báz. Po rozpletení dvojzávitnice DNA sa na oblasť pred začiatkom génu naviaže enzým RNA–polymeráza. Tento enzým postupuje po reťazci nukleotidov a prepisuje ich do RNA (konkrétne mRNA). Jediný rozdiel v nukleotidoch je pri tymíne, ktorý sa prepíše na uracil (U). RNA je teda jednovláknový reťazec zložený z nukleotidov A,C,G a U.
- Translácia: Preklad genetickej informácia z poradia nukleotidov v mRNA do poradia aminokyselín v polypeptidovom reťazci. To je zabezpečené pomocou ribozómov, ktoré postupne čítajú trojice nukleotidov v mRNA (kodóny) a prekladajú ich na aminokyseliny. Druh aminokyseliny určuje kodón mRNA a komplementárny antikodón tRNA. Výsledný reťazec aminokyselín je proteín.

Na nasledujúcom obrázku sú zobrazené tieto procesy.



Obrázok 2.1: Centrálna dogma molekulárnej biológie [29].

2.2 Malé nekódujúce RNA

Malé nekódujúce RNA (miRNA a siRNA) sú regulačné molekuly, ktoré sa uplatňujú vo väčšine eukaryotických organizmoch. Ich funkciou je napríklad eliminácia DNA, zhromažďovanie heterochromatínu, štiepenie mRNA a represia translácie [4]. Tieto procesy patria medzi epigenetické mechanizmy bunky a zaoberá sa nimi odbor epigenetika.

2.2.1 miRNA

miRNA (microRNA) sú malé nekódujúce RNA molekuly, ktoré majú dĺžku od 20 do 27 nukleotidov [4]. Nachádzajú sa v rastlinách, živočíchoch a v niektorých vírusoch. Ich funkciou je regulácia génovej expresie po transkripcii. Vznikajú transkripciou génov v DNA, ale ich translácia už nenastane. Namiesto toho sa primárny transkript miRNA (pri–miRNA) páruje s niektorými vlastnými komplementárnymi bázami a mení sa na miRNA. Tie sú čiastočne komplementárne k určitým molekulám mRNA a sú schopné regulovať výrobu proteínov, ktoré tieto mRNA kódujú. To môže prebiehať viacerými spôsobmi [4]. Napríklad:

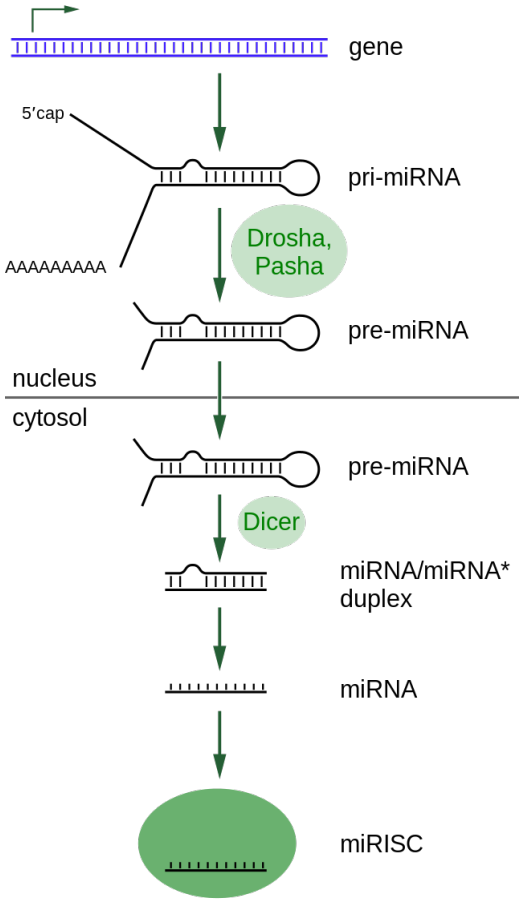
- Štiepenie mRNA na dva kusy.
- Destabilizácia mRNA skrátením jej poly(A) konca
- Menej účinná translácia mRNA na proteíny ribozómami.

Živočíšne miRNA je väčšinou komplementárne k regiónu 3' UTR, teda k časti mRNA, ktorá nekóduje proteíny, ale vykonáva regulačné funkcie súvisiace s danou molekulou RNA. Rastlinná miRNA je komplementárna ku kódujúcim oblastiam mRNA. Po spárovaní reťazcov miRNA a mRNA je zakázaná translácia mRNA na proteín. Niekedy je uľahčený rozklad mRNA. V tomto prípade vznik dvojvláknovej RNA spôsobuje v bunke proces podobný RNA interferencii, ktorý spôsobujú siRNA.

2.2.2 Vznik miRNA

Z génov je prepísaných asi 70 nukleotidov do reťazca pri–miRNA s čapičkou na 5' konci a poly–A koncom na 3' konci [4]. Pomocou proteínového komplexu (*Microprocessor complex*) nastáva prvá úprava. Je zložený z nukleázy Drosha a proteínu Pasha, ktoré sa viažu na dvojvláknovú RNA. Tento komplex mení pri–miRNA na pre–miRNA.

Potom pre–miRNA vstupuje do cytoplazmy, kde interaguje s endonukleázou Dicer, čím vzniká miRNA, ktorá sa viaže do komplexu RISC (RNA–*induced silencing complex*). Ten je schopný utlmovať expresiu génov (RNA interferencia). Tento postup prebieha u živočíchov. U rastlín je postup mierne odlišný. Nenachádza sa u nich proteín Drosha a jeho úlohu plní Dicer.



Obrázok 2.2: Vznik miRNA [30].

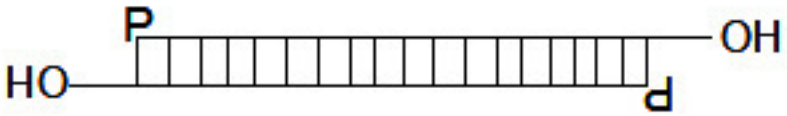
Funkcia miRNA sa u rastlín a živočíchov líši. Rastlinné miRNA majú obvykle takmer presné napojenie na ich cieľové mRNA. Preto je na génovú represiu uplatnené štiepenie mRNA na kusy.

U zvierat sú rozpoznávané cieľové mRNA pomocou 6–8 nukleotidov na 5' konci miRNA, čo je nedostačujúce na štiepenie [7].

2.2.3 siRNA

siRNA (small interfering RNA) sú dvojvláknové RNA, ktoré majú dĺžku 20–25 nukleotidov. Uplatňujú sa podobne miRNA v RNA interferencii, a teda ovplyvňujú expresiu génov. Môžu slúžiť nielen na ochranu pred vírusmi, ale ovplyvňujú aj priestorovú štruktúru chromatínu.

siRNA majú dobre definovanú stavbu. Na jednej strane každého vlákna prečnievajú dva nukleotidy. Tie sú nespárované s nukleotidmi druhého vlákna [4].



Obrázok 2.3: Štruktúra siRNA [31].

Na 5' konci je fosfátová skupina, na 3' konci je hydroxylová skupina. Táto štruktúra je určená enzymatickým účinkom enzýmu Dicer.

Tento typ RNA vzniká štiepením pomerne dlhých dvojvláknových molekúl RNA (napríklad transpozonových alebo vírusových), alebo prepisom časti genómu, napríklad v centromérických alebo repetitívnych oblastiach DNA. Niektoré siRNA môžu vzniknúť aj štiepením častí molekúl mRNA [4].

siRNA sa podobne ako miRNA viaže s proteínovým komplexom RISC a smeruje ho ku konkrétnemu úseku mRNA, ktorý je komplementárny s siRNA. Po naviazaní na RISC nasleduje rozpletenie a zahodenie priameho vlákna s endonukleázami. Komplementárne vlákno sa naviaže na cieľovú mRNA a tým sa inicializuje umlčanie. RISC potom katalyzuje rozštiepenie tejto cieľovej mRNA. Takto dochádza k posttrankripčnému umlčaniu (*silencing*) génu. Gén sa prepisuje, ale jeho mRNA je štiepená, čím je zabránené vytvoreniu proteínu. siRNA môže blokovať aj prepis génu pomocou mechanizmov, ktorými navodzuje vznik heterochromatínu, ktorý nie je prepisovaný.

Kapitola 3

Next–generation sekvenovanie

Sekvenovanie je metóda na určenie presného poradia nukleotidov, ktoré sa nachádzajú v DNA alebo RNA molekulách [6]. Sekvenovanie sa používa viac ako v minulosti a je stále prístupnejšie pre výskumné a klinické centrá po celom svete. Prvým míľnikom DNA sekvenovania bol výskum *Human Genome Project*, ktorý bol ukončený v roku 2003. Stál 3 miliardy dolárov a trval 13 rokov. Sekvenovanie prebiehalo takzvanou Sangerovou metódou. Ide o metódu sekvenovania prvej generácie. Bola vyvinutá v roku 1975 a stala sa štandardom na nasledujúcich 25 rokov.

Po dokončení projektu však bolo potrebné získať rýchlejšie a lacnejšie metódy. Preto boli vyvinuté metódy druhej generácie, alebo *next–generation sequencing* (NGS). Tieto metódy sú založené na paralelnom spracovaní a umožňujú osekvenovanie celého genómu za jeden deň a oveľa lacnejšie ako v prípade Sangerovej metódy.

Slúžia na sekvenovanie molekúl DNA, RNA a krátkych nekódujúcich RNA [22], ktoré sú pre túto prácu podstatné. Medzi nevýhody však patrí napríklad nepresné sekvenovanie homopolymérových úsekov (opakujú sa v nich nukleotidy). To platí hlavne pre Ion Torrent PGM. Ďalším nedostatkom je krátka dĺžka výsledných získaných sekvencií, čo môže viesť k nepresnostiam.

V súčastnosti patria medzi najpoužívanejšie metódy vo výskume a klinických laboratóriách Ion Torrent Personal Genome Machine (PGM) a Illumina MiSeq [6].

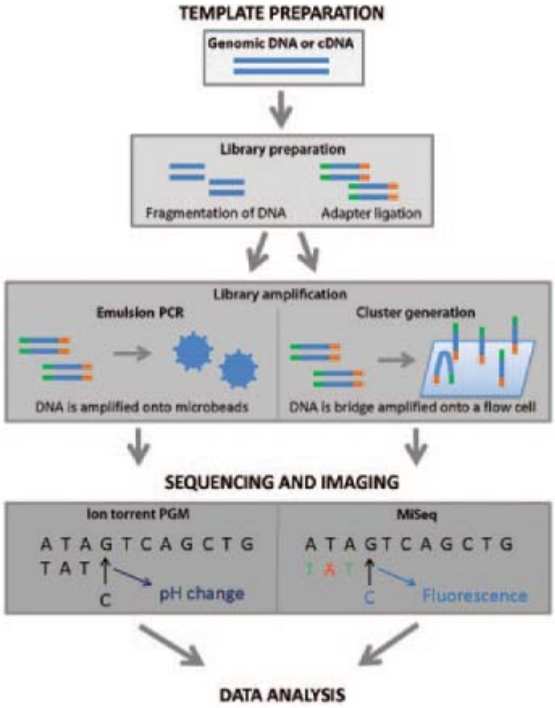
Obe metódy majú podobnú metodológiu, ktorá sa skladá z troch krokov:

- Pripravenie formy
- Sekvenovanie a *imaging*
- Analýza dát

Odlišujú sa však v detailoch jednotlivých krokov. Postup je zobrazený aj na obrázku 3.

3.1 Pripravenie formy

Príprava formy pozostáva z vytvorenia knižnice nukleových kyselín (DNA alebo komplementárne DNA) a jej amplifikácie. Knižnica je vytvorená fragmentovaním DNA vzoriek a ligáciou adaptérových sekvencií na koniec DNA fragmentov. Po vytvorení sú knižnice amplifikované a pripravené na sekvenovanie.



Obrázok 3.1: Postup pri next–gen sekvenovaní [6].

3.2 Sekvenovanie a *imaging*

Na získanie sekvencií nukleových kyselín z knižníc používajú obe metódy sekvenovanie syntézou. Fragmenty z formy plnia úlohu vzorov, z ktorých sa syntetizujú nové DNA fragmenty. Po pripojení nukleotidov do rastúceho DNA reťazca sú digitálne získane ako sekvencia. V tomto kroku sa metódy odlišujú [6].

PGM vykonáva sekvenovanie pomocou polovodičov, ktoré je založené na detekcii zmien pH hodnoty. Po pripojení nukleotidu k reťazcu je vytvorená kovalentná väzba medzi nukleotidmi a je uvoľnený pyrofosfát a kladne nabitý vodíkový ión. Tento ión spôsobuje zmenu pH hodnoty a následne je určený nukleotid.

MiSeq je založený na detekcii fluorescencie, ktorá je generovaná pripojením fluorescenčne označených nucleotidov do rastúceho reťazca DNA.

3.3 Analýza dát

Po dokončení sekvenovania musia získané dáta prejsť niekoľkými krokmi analýzy. Patrí sem odstránenie adaptérových sekvencií a nekvalitných vzoriek. Ďalej mapovanie dát na referenčný genóm alebo *de novo* zarovnanie sekvenovaných vzoriek a analýza skompilovanej sekvencie.

Tá zahŕňa napríklad detekciu SNP (polymorfizmus nukelotidu), vloženia alebo odstránenia báz, detekciu nových génov alebo regulačných elementov. Môže sem patriť aj identifikácia somatických a zárodočných mutačných udalostí, čím sa môže podieľať napríklad na diagnóze rôznych chorôb [6].

3.4 Dátové formáty

Získané dáta sa najčastejšie ukladajú do formátov FASTA [11] a FASTQ [23]. Ďalšími možnými formátmi sú napríklad SAM, BAM (slúžia prevažne na uloženie zarovnanie NGS dát) [2], FASTG, SFF a VCF [17].

Nasleduje opis dvoch najpoužívajších, teda FASTQ a FASTA.

3.4.1 FASTQ

Tento formát obsahuje pre každú sekvenciu 4 riadky:

1. Znak „@“ a identifikátor sekvencie. Môže byť pridaný aj opis sekvencie.
2. Samotná sekvencia.
3. Znak „+“, ktorý môže byť nasledovaný rovnakým identifikátorom sekvencie a jej opisom, ako v prvom riadku.
4. Reťazec určujúci skóre kvality sekvenácie pre jednotlivé nukleotidy. Má rovnakú dĺžku ako sekvencia.

Súbor vo FASTQ formáte má väčšinou koncovku „.fq“ alebo „.fastq“. Príklad takéhoto záznamu:

@EAS20
CGCGTAACAAAAGTGTCTATAATCACGGC
+EAS20
HHHHHHHHHFHGGHHHHHHHHHHHHHHHH

3.4.2 FASTA

Formát FASTA je podobný formátu FASTQ avšak neobsahuje jeho tretí a štvrtý riadok:

1. Znak „>“ a identifikátor sekvencie. Môže byť pridaný aj opis sekvencie.
2. Samotná sekvencia rozdelená do riadkov, ktorých dĺžka môže byť 60 alebo 80 nukleotidov.

Súbor vo FASTA formáte má väčšinou koncovku „.fa“ alebo „.fasta“.

Kapitola 4

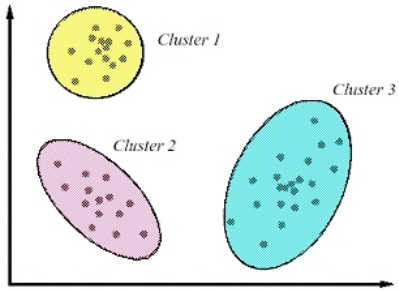
Zhlukovanie RNA sekvencií

4.1 Zhlukovanie dát

Zhlukovanie je proces, ktorého cieľom je rozdelenie sady objektov do skupín (zhlukov) tak, že v každej skupine sa nachádzajú objekty, ktoré majú podobné vlastnosti [5]. Veľké objemy sú teda reprezentované malou dátovou sadou, v ktorej je každý zhluk reprezentovaný jedným reprezentantom alebo obmedzením. Rozdiely medzi objektami v zhluku by mali byť čo najmenšie a rozdiely medzi samotnými zhlukmi by mali byť čo najväčšie.

Pretože dáta, s ktorými sa bude pracovať obsahujú veľké množstvo sekvencií, cieľom zhlukovania v tejto práci bude redukcia dátovej sady v snahe minimalizovať stratu informačnej hodnoty. Algoritmy, ktoré riešia túto úlohu pracujú na rôznych princípoch.

Zhlukovanie môže byť definované ako viackriteriálny optimalizačný problém. Výber zhlukovacieho algoritmu a parametrov ako vzdialenostná funkcia, počet očakávaných zhlukov a podobne, záleží na konkrétnych dátach a následnom využití výsledkov. Na nasledujúcom obrázku je zobrazený príklad dát, ktoré sú rozdelené do zhlukov na základe ich vzájomnej vzdialenosti.



Obrázok 4.1: Príklad dátových zhlukov [25].

Nasleduje opis najpoužívanejších zhlukovacích algoritmov.

4.1.1 Hierarchické zhlukovanie

Je založené na myšlienke, že objekty, ktoré sú podobnejšie sú menej vzdialené ako objekty menej podobné. Zhlukovanie teda prebieha na základe ich vzdialeností. Zhluk môže byť

určený maximálnou vzdialenosťou, ktorá je potrebná na prepojenie jeho objektov. Pri rôznych vzdialenostiach sa vytvárajú ďalšie zhluky. Výsledkom analýzy je hierarchia zhlukov, ktoré sa navzájom spájajú podľa ich vzdialenosti.

Jednotlivé algoritmy sa odlišujú podľa spôsobu akým sú vypočítané vzdialenosti. Môže to byť napríklad podľa maximálnej alebo minimálnej vzdialenosti objektov. Zložitosť je $O(n^3)$, kvôli čomu sú príliš pomalé pre veľké dátovej sady [5].

4.1.2 *Centroid–based* zhlukovanie

V tomto prípade sú zhluky reprezentované centrálnym vektorom, ktorý ale nemusí byť členom dátovej sady. Pokiaľ je počet žiadaných zhlukov pevne určený na hodnotu k, tak ide o *k–means* zhlukovanie. Cieľom je potom nájsť k zhlukových stredov a priradenie objektov k najbližšiemu zhlukovému stredu. Teda aby boli štvorce vzdialeností od zhluku čo najmenšie.

Tento problém je NP–ťažký, preto sa využíva hľadanie približných riešení. Známy je Lloydov algoritmus [18], ktorý ponúka lokálne optimum a často sa vykonáva viac krát s rôznymi náhodnými inicializáciami.

Variácie *k–means* často obsahujú optimalizácie ako výber najlepšieho výsledku z niekoľkých chodov, určenie, že stredy zhlukov musia byť členovia daných zhlukov. Alebo výber mediánov (*k–median* zhlukovanie), výber počiatočných stredov menej náhodne (*k–medoidy*) alebo povolenie fuzzy priradenia zhlukov (*Fuzzy c–means*) [5].

Väčšina algoritmov často vyžaduje, aby bol počet zhlukov k definovaný, čo je považované za ich najväčšiu nevýhodu. Okrem toho algoritmy uprednostňujú zhluky približne rovnakej veľkosti. To často vedie k nesprávnemu upraveniu hraníc medzi zhlukmi, pretože algoritmus optimalizuje zhlukové centrá, nie ich hranice.

4.1.3 Zhlukovanie založené na mriežke

Priestor objektov je rozdelený na konečný počet buniek, ktoré tvoria mriežku [5]. Operácie zhlukovania prebiehajú nad touto mriežkovou štruktúrou.

Výhodou je rýchla doba spracovania, ktorá nie je závislá na počte objektov, ale na počte buniek mriežkovej štruktúry.

Patrí sem metóda WaveCluster. Postup metódy:

1. Rozdelenie dátového priestoru pomocou mriežky.
2. Každá bunka obsahuje informácie o skupine objektov, ktoré sú do nej mapované.
3. Na hodnoty v mriežke sa aplikuje viacúrovňová vlnková transformácia. Tá zdôrazňuje oblasti, v ktorých dochádza k zhlukovaniu bodov v priestore. Potláča slabšie informácie, ktoré sa nachádzajú za hranicami zhlukov. Tým sa zvýrazňujú zhluky a odstraňujú odľahlé hodnoty.

Metóda má zložitosť $O(n)$, dokáže efektívne spracovať veľké dátové množiny a nájsť zhluky ľubovolného tvaru. Nevyžaduje špecifikáciu vstupných parametrov.

4.1.4 Zhlukovanie založené na modeloch

Pri týchto metódach nastáva optimalizácia zhody medzi dátovou množinou a matematickým modelom. Hľadajú sa zhluky, ktoré maximálne odpovedajú danému modelu [5]. Dáta sú generované na základe pravdepodobnostnej funkcie.

Patrí sem napríklad metóda *Expectation–Maximalization*, konceptuálne zhlukvanie a metódy založené na neurónových sietiach (SOM).

V metóde *Expectation–Maximalization* je zhluk reprezentovaný pomocou parametrizovanej pravdepodobnostnej distribučnej funkcie. Dátová množina je zmes týchto distribučných funkií. Modelom je k pravdepodobnostných distibučných funkcií, pričom každá reprezentuje jeden zhluk. Metóda umožňuje nájsť parametry distibučných funkcií. Metóda obvykle konverguje, ale nemusí nájsť globálne optimum.

4.1.5 Zhlukovanie založené na hustote

Zhluky sú definované ako oblasti s väčšou hustotou ako zvyšok dátovej sady [5]. Objekty v riedkych oblastiach sú považované za šum alebo hraničné body.

Najpopulárnejšou metódou je DBSCAN. Jej zhlukovací model je založený na spájaní bodov v určitého vzdialenostného prahu. Avšak spája iba body, ktoré spĺňajú hĺbkové kritériá (v pôvodnej verzii definované ako minimálny počet iných objektov v rámci rádiusu).

V každom behu algoritmu sú výsledky rovnaké, preto nie je nutné metódu opakovať a porovnávať výsledky. Nevýhodou je, že metódy očakávajú pokles hustoty na detekovanie hraníc zhlukov. Taktiež nemôžu detekovať vnútorné štruktúry zhlukov, ktoré sú bežné vo väčšine reálnych dát.

4.2 Klasifikácia RNA sekvencií

Obrovský nárast počtu nasekvenovaých sekvencií vyvinul tlak na zhlukovacie algoritmy, čo sa týka hlavne ich schopnosti spracovať veľké množstvo dát v rozumnom čase.

Programy, ktoré sú opísané v tejto kapitole využívajú takzvaný hladný (*greedy*) inkrementálny zhlukovací algoritmus [27] a patria pod *centroid–based* zhlukovacie programy. V každom programe sa nachádzajú určité odlišnosti, ale základný algoritmus pracuje nasledovne:

1. Sekvencie sú najskôr zoradené od najdlhších po najkratšie.
2. Najdlhšia sekvencia je reprezentatívna pre prvý zhluk.
3. Každá ďalšia je porovnaná s existujúcimi reprezentujúcimi sekvenciami.
4. Ak je podobnosť vyššia ako zadaný prah, je sekvencia priradená do daného zhluku. Inak je vytvorený nový zhluk a sekvencia sa stáva jeho reprezentatívnou sekvenciou.

Nasleduje opis najpoužívanejších programov, ktoré sa dajú použiť na klasifikáciu RNA sekvencií.

4.2.1 CD–HIT

CD–HIT je voľne dostupný program (*open source*), ktorý slúži na zhlukovanie DNA, RNA sekvencií a proteínov [26]. Dokáže pracovať paralelne na viacjadrových systémoch. Pre každé porovnanie sekvencií je aplikované filtrovanie znakov, aby sa zistilo či je podobnosť nižšia ako prah zhluku. Ak to nemôže byť potvrdené, je vykonané zarovnanie sekvencií. Aby sa zrýchlil výpočet zhlukov, CD–HIT používa heuristiky založené na štatistickom *k–mer* filtrovaní.

Filter pracuje nasledovne. Dve sekvencie, ktoré sa čiastočne zhodujú, musia mať určitý počet zhodných dipeptidov, tripeptidov a podobne. Napríklad, ak majú dve sekvencie proteínov 85 percentnú identitu na 100 znakoch, mali by mať aspoň 70 identických dipeptidov, 55 identických tripeptidov a 25 identických pentapeptidov. CD–HIT vynecháva väčšinu párových zarovnaní, pretože pomocou jednoduchého spočítania slov vie, že podobnosť dvoch sekvencií je pod určitým prahom.

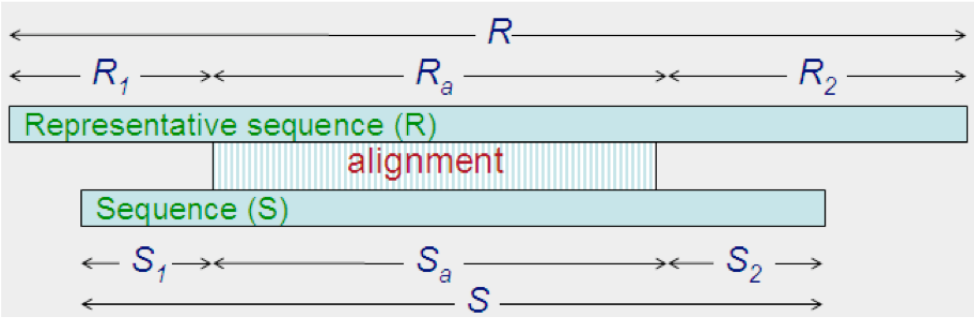
Ďalšie zrýchlenie prináša použitie indexovacej tabuľky. Využíva veľmi krátke slová s dĺžkou 2 až 5. Napríklad celkový počet možných pentapeptidov pri proteínoch je 215 a idexovacia tabuľka vyžaduje iba 4 milióny záznamov, čo je pre súčasné počítače prijateľná hodnota. Tabuľka umožňuje účinné zrátanie krátkych slov. Zrátanie dlhých slov je ešte účinnejšie.

Medzi nevýhody filtru patrí, že ho nie je možné použiť pre nižšie zhlukovacie prahy. V najhoršom prípade, keď sú nezhody medzi sekvenciami rozložené po celom zarovnaní, počet spoločných slov je minimálny. Takže teoreticky môžu byť pentapeptidy, tetrapeptidy, tripeptidy a dipeptidy byť použité iba pre prahy vyššie ako 80, 75, 66, 67 a 50 percent. Tento najhorší prípad je medzi skutočnými dátami pomerne výnimočný vďaka evolúcii, ktorá uprednostňujú viac konzervované a rozmanité oblasti [26].

Ďalší problém súvisí s hladným inkrementálnym zhlukovaním. Napríklad môže byť sekvencia priradený do zhluku A, aj keď je sa viac zhoduje so zhlukom B. Situácia môže nastať iba kvôli tomu, že nastalo porovnávanie najprv s zhlukom A. Táto nedokonalosť sa dá minimalizovať viackrokovým zhlukovaním.

Tieto príklady boli zamerané na zhlukovanie proteínových sekvencií, ale zhlukovanie DNA a RNA sekvencií týmto programom funguje obdobne využitím podprogramu CD–HIT–EST.

Na obrázku je zobrazené porovnanie sekvencií týmto programom:



Obrázok 4.2: Porovnanie sekvencií programom CD–HIT–EST [1].

4.2.2 Uclust

Algoritmus môže byť použitý na zhlukovanie DNA, RNA sekvencií a proteínov [12]. Jeho binárne súbory sú voľne dostupné. Uclust využíva podobne ako CD–HIT hladný inkrementálny prístup. Na rozdiel od CD–HIT, na rýchle porovnávanie sekvencií používa heuristiku nazvanú Usearch. Rýchlosť je získaná porovnávaním iba niekoľkých sekvencií namiesto celej databáze.

Databázové sekvencie sú zoradené podľa počtu znakov, v ktorých sa zhodujú (v poradí od najväčšej zhody). Je to založené na:

- Ak existuje zhoda v databázy, je pravdepodobné, že to bude medzi prvými kandidátmi.
- Pravdepodobnosť, že existuje zhoda rapídne klesá počtom neúspešných pokusov.

Hľadanie teda môže byť ukončené po preskúmaní malého počtu kandidátov bez velkej straty citlivosti. Párové porovnávanie používa štandardné rýchle zarovnávacie techniky, ako bezmedzerové detekciu segmentových párov a dynamické programovanie.

4.2.3 DNACLUST

DNACLUST je volne dostupný program (open source). Taktiež využíva hladný inkrementálny prístup a suffixové pole na indexovanie vstupnej sady dát [20].

Výber najdlhšej sekvencie bez zhluku a jej určenie za zhlukové centrum je nutné na zabezpečenie správnosti zhlukovania v prípade, ak sa dĺžky sekvencií nezhodujú. Ak by boli dve sekvencie, ktoré sú dlhšie ako zhlukové centrum, určené do jedného zhluku spolu, nebolo by možné zaručiť, že boli priradené správne. Bolo by možné, že sa výrazne odlišujú.

Výpočet vzdialenosti pri prehľadávaní nepriradených sekvencií môže byť založený na globálnom, alebo semi–globálnom zarovnaní. V tom prípade sa cena medzery na jednom alebo oboch koncoch kratšej sekvencie ignoruje. Toto prehľadávanie je krok, ktorý zaberá algoritmu najviac času [20].

Algoritmus môže byť pomerne jednoducho modifikovaný tak, aby vytváral dostatočne vzdialené zhluky:

- Vždy, keď je vytvorený nový zhluk s polomerom r, označia sa všetky sekvencie bez zhlukov so vzdialenosťou menšou ako $2r$ od stredu centra nového zhluku.
- Označené sekvencie nemôžu byť v nasledujúcich iteráciách určené ako centrá nových zhlukov.
- Môžu sa však priradiť do zhlukov, ktoré vznikli okolo neoznačených zhlukových centier.

Tento prístup zaručuje, že vzdialenosť medzi 2 centrami zhlukov nebude menšia ako dvojnásobok polomeru zhluku.

4.2.4 SEED

Na rozdiel od vyššie spomenutých programov pracuje SEED iba s DNA sekvenciami [13]. Hodí sa na zhlukovanie vysoko podobných sekvencií. Pracuje iba so vzorkami získanými metódou Illumina a identifikuje iba sekvencie, ktoré sa líšia v maximálne 3 znakoch, alebo 3 prevísajúcich znakoch.

Používa otvorenú hašovaciu techniku a špeciálnu triedu rozmiestnených semienok (*seeds*), ktoré sa nazývajú blokové rozmiestnené semienka. Po uložení vzoriek do hašovacej tabuľky, ich zhlukovanie prebieha vytvorením virtuálnej centrálnej sekvencie pre každý zhluk a hľadaním vzoriek, ktoré spĺňajú prah podobnosti k centrálnej sekvencii.

Indexovanie prebieha v nasledujúcich krokoch [13]:

1. Inicializácia indexovania ak sa najdlhšia a najkratšia sekvencia nelíšia v dĺžke o viac ako 5 báz.

2. Použitie prvého semienka vo vybranej sade blokových semienok na hašovanie sekvencií do hašovacej tabuľky

3. Opakovanie bodu 2 s každým blokom semienok danej sady a uloženie ich výsledkov do oddelených hašovacích tabuliek.

Zhlukovanie prebieha nasledovne:

1. Výber ľubovolnej sekvencie, identifikovanie všetkých sekvencií s počtom nezhôd s touto sekvenciou menším ako dvojnásobok prahu a výpočet ich virtuálnych centier.

2. Nájdenie všetkých sekvencií pre dané virtuálne centrum s povoleným počtom prevísajúcich báz, alebo nezhôd. Nasleduje vymazanie týchto sekvencií z hašovacej tabuľky.

3. Opakovanie bodov 1 a 2, pokiaľ nie je hašovacia tabuľka prázdna.

V nasledujúcich častiach práce je opísaná analýza týchto zhlukovacích nástrojov vykonaná na reálnych dátach a výber jedného z nástrojov. Ten bude použitý pri klasifikácii miRNA. Podrobný postup využitia tohto nástroja je taktiež opísaný v ďalších kapitolách.

Kapitola 5

Redukcia dát a klasifikácia

Bolo potrebné rozhodnúť, ktorý program na zhlukovanie sekvencií sa použije v nasledujúcich častiach práce na redukciu dátovej sady. Výber najvhodnejšieho programu prebehol nasledovne.

5.1 Vstupné dáta

Bolo určených 5 testovacích dátových sád. Sú to reálne dáta, ktoré vznikli sekvenovaním technológiou Illumina MiSeq, ktorá je opísaná v kapitole 3. Ide o sadu malých nekódujúcich RNA rastliny *Silene latifolia* (Silenka biela), ktoré obsahujú veľký počet sekvencií. Konkrétne to boli malé RNA zo samčích listov (10915932 sekvencií), samičích listov (8452609 sekvencií), neoplodnených piestikov (9261945 sekvencií), oplodnených piestikov (11454754 sekvencií) a RNA z peľu (11345890 sekvencií).

Ako referenčné boli použité jednotlivé rodiny miRNA sekvencií z referenčnej sady. Tá bola získaná z databáze miRBase, verzia 17[1]. Boli z nej použité iba vybrané (rastlinné) sekvencie. Takýchto sekvencií bolo 8496. Všetky spomenuté sekvencie boli pomerne krátke, nepresahovali dĺžkou 30 nukleotidov.

Nasledovala identifikácia pomocou zhlukovania. Najprv boli zlúčené referenčné sekvencie s jednotlivými testovacími sadami a spustenie zhlukovacích programov s danými zlúčenými sekvenciami.

Ideálny výsledok by nastal v nasledujúcom prípade:

- Ak by vo výsledných zhlukoch boli spolu v jednom zhluku všetky referenčné sekvencie z rovnakej skupiny spolu so sekvenciami z testovacích dát.
- Referenčné sekvencie z iných skupín by sa v danom zhluku nemali nachádzať.

Program, ktorý splní tieto 2 podmienky najlepšie, je určený ako najvhodnejší pre ďalšiu prácu.

5.2 Priebeh experimentu

Na testovanie boli určené programy opísané v minulej kapitole, teda CD–HIT, Uclust, DNACLUST a SEED.

[1] http://www.mirbase.org/

- SEED sa ukázalo ako nevhodný kvôli jeho obmedzeniam. Je schopný zhlukovať iba sekvencie, ktoré sa odlišujú v dĺžke o maximálne 3 sekvencie. Kedže vstupné dáta mali rozptyl dĺžok väčší, nebolo možné tento program použiť.

Ostatné programy nemajú podobné obmedzenia a zadané dáta dokázali spracovať a poskytnúť výsledky na ďalšiu analýzu.

- DNAclust bol spustený s parametrom *no–k–mer* filter, ktorý je vhodný na zhlukovanie veľmi krátkych sekvencií s vysokou podobnosťou, ako v tomto prípade.
- CD–HIT–EST, ktorý funguje rovnako ako CD–HIT, avšak je určený na zhlukovanie DNA a RNA sekvencií, bol spustený s prahom 90 percent.
- Program Uclust bol taktiež spustený s prahom 90 percent.

Tieto výpočty sú pomerne časovo a výpočtovo náročné, preto boli vykonané na Metacentre[2]. Ide o distribuovanú výpočtovú štruktúru, ktorá pozostáva z vlastných i zverených výpočtových a úložných kapacít akademických centier Českej republiky. Po vykonaní výpočtov vznikli pre každú z piatich dátových sád tri rôzne rozdelenia do zhlukov, podľa programov, ktoré ich vykonali.

5.3 Analýza a zhodnotenie výsledkov

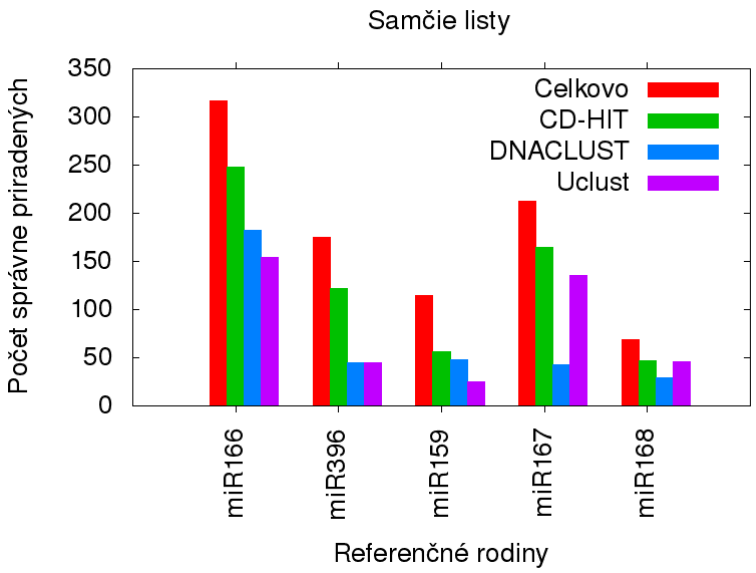
Najprv bolo získaných 1000 najväčších zhlukov z každého výstupu. Potom boli každom zhluku nájdené iba referenčné sekvencie. Následne boli analyzované odpovedajúce zhluky z každého z troch programov. Ideálnym výsledkom pre zhluk by bolo, ak by obsahoval všetky sekvencie z referenčnej sady, ktoré patria do rovnakej rodiny. Tento stav však nebol nikdy dosiahnutý. Ďalším kritériom bol počet sekvencií, ktoré patrili do inej referenčnej rodiny, ale boli priradené nesprávne do daného zhluku.

Tieto dve informácie boli získané pre 5 najväčších zhlukov, ktoré obsahovali referenčné sekvencie z rovnakých rodín vo výstupe z každého programu. Tento postup bol vykonaný pre každú z piatich dátových sád. Jediná sada, v ktorej nebolo v tisíc najväčších zhlukoch päť takých, ktoré by obsahovali rovnaké rodiny referenčných sekvencií bola posledná, teda RNA z peľu. Pri nej boli nájdené iba štyri takéto zhluky.

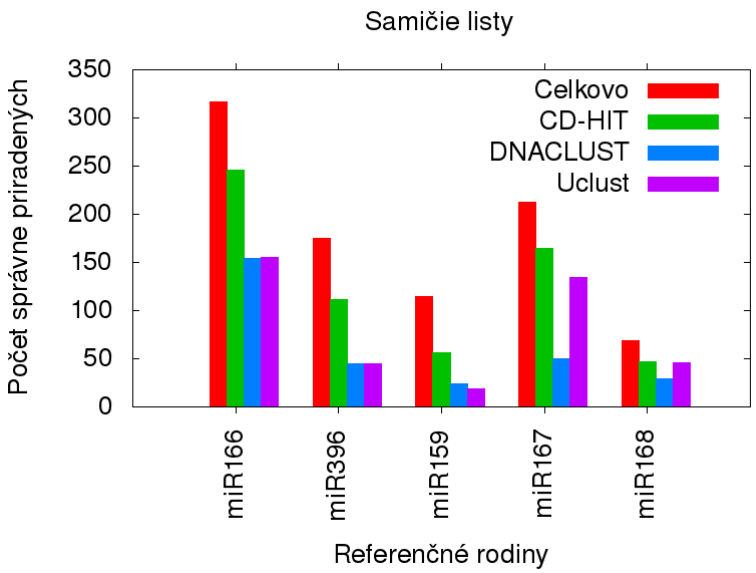
Výsledky pre každú dátovú sadu sú zobrazené na nasledujúcich grafoch. Prvý stĺpec určuje celkový počet sekvencií v danej rodine. Ostatné určujú počet správne priradených sekvencií z danej rodiny.

Počet sekvencií, ktoré boli do daného zhluku priradené z nesprávej rodiny bol minimálny. V malom počte prípadov ich bolo nesprávne priradených 5 a raz 12. Takéto sekvencie sa našli iba pri programe CD–HIT–EST. Táto chyba je zanedbateľná a nie je ani zobrazená v grafoch.

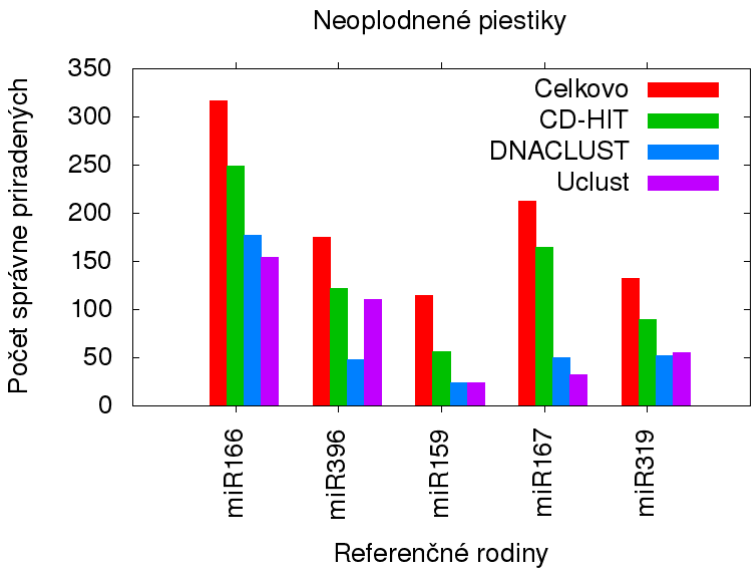
[2] https://metavo.metacentrum.cz/



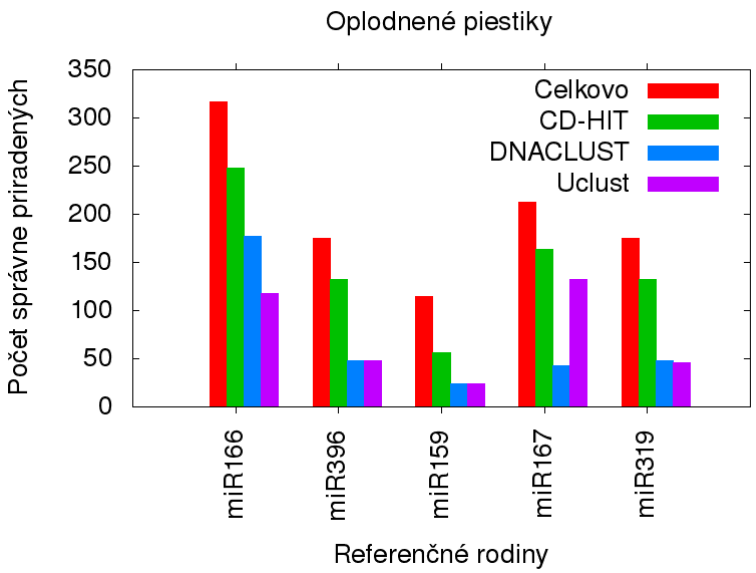
Obrázok 5.1: Porovnanie výsledkov pre samčie listy.



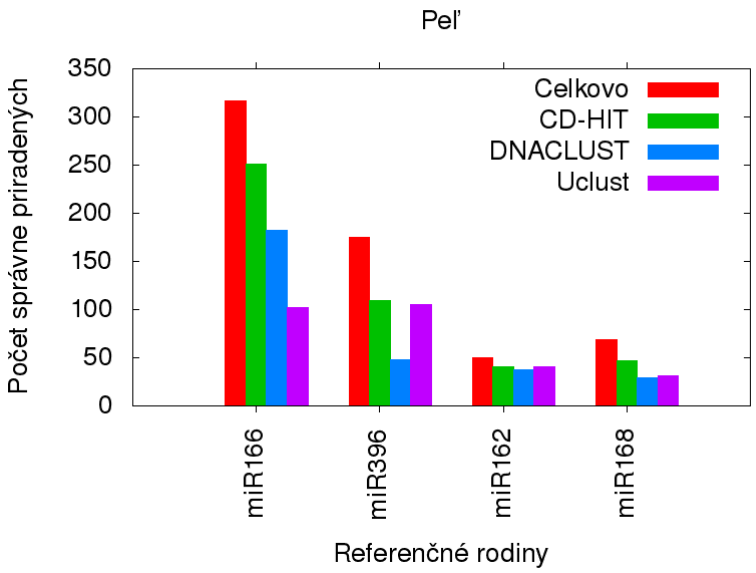
Obrázok 5.2: Porovnanie výsledkov pre samičie listy.



Obrázok 5.3: Porovnanie výsledkov pre neoplodnené piestiky.



Obrázok 5.4: Porovnanie výsledkov pre oplodnené piestiky.



Obrázok 5.5: Porovnanie výsledkov pre peľ.

Nesprávne priradené referenčné sekvencie sa objavovali iba pri použití programu CD–HIT–EST a to iba pri niektorých zhlukoch. Ostatné programy sa tejto chybe vyhli. Avšak aj napriek tomu je je CD–HIT najvhodnejším prostriedkom na redukciu dát bez výraznej straty informácie. Pretože počet správne priradených referenčných sekvencií sa často blíži celkovému počtu týchto sekvencií a v každom zhluku tento počet prevyšuje výsledok z ostatných dvoch programov.

Kapitola 6

Klasifikácia miRNA bez referenčného genómu

Hlavným cieľom práce bola klasifikácia malých nekódujúcich RNA, konkrétne rastlinných microRNA, bez ich referenčného genómu. Nástroj na klasifikáciu živočíšnych miRNA bez referenčného genómu existuje (Mirplex [10]), ale nástroj na klasifikáciu rastlinných miRNA ešte nebol implementovaný.

Tieto miRNA by mali byť klasifikované nie na základe ich vlastností a polohe v genóme, ale na základe vlastností a štruktúry sekvencií, z ktorých sú zložené. Pokrok v sekvenovacích technológiách umožňuje zisk kompletného obsahu malých RNA organizmu alebo vlákna [10]. Tie typicky pozostávajú z miliónov krátkych sekvencií alebo vlákien s dĺžkou od 16 do 25 nukleotidov. Často sa niektoré sekvencie nachádzajú v datasete viackrát. Preto sú uložené iba odlišné sekvencie spolu s údajom o ich početnosti.

S využitím takéhoto datasetu je možné identifikovať známe *mature* miRNA a to porovnaním sRNA sekvencií s databázou známych miRNA (miRBase). Na to môže slúžiť software ako miRProf [21] a miRNAKey [24]. Aj keď je takýto postup pomerne jednoduchý, je limitovaný počtom už identifikovaných miRNA. Pokročilejšie nástroje ako miRCat [21] a miRDeep [15] umožňujú predikciu nepreskúmaných miRNA mapovaním sRNA sekvencií na referenčný genóm a identifikovaním domnelých miRNA vlásenkových štruktúr [10].

Genómy však často nie sú dostupné a to aj napriek pokrokom v sekvenovaní a ich zostavovaní. A preto je potrebné predikovanie nových miRNA využitím sRNA datasetov.

Jednou z metód na predikovanie miRNA z sRNA datasetov bez genómu je využitie evolučného konzervovania *mature* miRNA vlákien [10]. miRNA sú užitočnými fylogenetickými ukazovateľmi. Pretože:

- Do genómov sa postupne pridávajú nové miRNA rodiny.
- Je nepravdepodobné, že rôzne miRNA s rovnakými *mature* sekvenciami sa vyvinú separovane vďaka náhode.

MirMiner [28] zisťuje evolučné konzervovanie miRNA a vytvára zoznam konzervovaných vlákien medzi skupinami organizmov so spoločnými vlastnosťami alebo zoznam unikátnych sekvencií pre každú skupinu. Výsledky môžu byť overené pomocou zarovnania genómov alebo metódou *genome walking*.

Pokrok v sekvenačných technológiách taktiež ponúka ďalší prístup na predikovanie miRNA bez genómu. Tým je využitie dlhších vlákien na zachytenie celej pre–miRNA sekvencie do jedného vlákna. Avšak stabilita pre–miRNA je oveľa nižšia ako stabilita *mature*

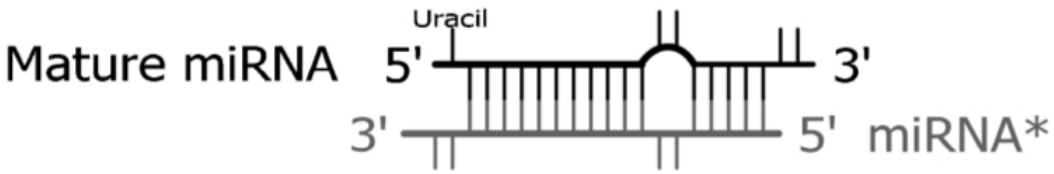
sekvencií, čo znamená, že nemusí byť možné získať sekvenciu prekurzorov málo exprimovaných miRNA [10]. Okrem toho môžu vlásenky prekročiť maximálnu dĺžku niektorých bežne používaných sekvenovacích platforiem. To platí hlavne pri prekurzoroch rastlinných miRNA, ktoré môžu byť dlhé niekoľko stoviek báz.

V práci je opísaná alternatívna metóda na predikciu miRNA bez genómu. Využíva vlastnosti miRNA duplexov.

6.1 miRNA Duplex

Duplex je dvojzávitnicová RNA (dsRNA) s vysokou komplementaritou vlákien a dvoma prevísajúcimi nukleotidmi na 3' konci [10]. Jedno vlákno je typicky dominantnejšie. Je to *mature* sekvencia, ktorá býva použitá komplexom RISC vďaka proteínu Argonaut. Druhé vlákno, miRNA* (*star* vlákno) býva degradované alebo menej abundantné. U živočíchov sa RISC neviaže na cieľ silnou väzbou. Väčšinou je perfektná komplementarita iba na 3' konci miRNA. To vedie k represii translácii, na rozdiel od degradácie a štiepenia, ktoré sú typické pre rastliny. Konkrétny postup vzniku miRNA bol opísaný v časti 2.2.2 a zobrazený na obrázku 2.2.2.

Typické miRNA duplexy majú niekoľko rozlišovacích vlastností, ako dĺžka sekvencií, komplementarita medzi vláknami a ďalšie štrukturálne vlastnosti. Obe vlákna miRNA duplexov sú väčšinou prítomné v sRNA datasete, pričom *mature* vlákno zvykne mať vysokú početnosť. Naopak väčšina miRNA* má nižšiu početnosť v datasete.



Obrázok 6.1: Duplex, prevzaté z [10].

Navrhnutá a implementovaná metóda využíva zhlukovanie na zníženie počtu miRNA kandidátnych duplexov z sRNA datasetov. Tie sú ďalej spracované nástrojom *support vector machine* (SVM), ktorý im priradí skóre a určí, či sú to skutočne miRNA duplexy.

Jedným z cieľov bolo zníženie počet sekvencií, ktoré je potrebné klasifikovať pomocou niektorej z metód strojového učenia. Vstupné dáta môžu často obsahovať desiatky miliónov sekvencií. Ak by sa vytvárali duplexy kombinovaním každej sekvencia so všetkými ostatnými, počet duplexov by mohol prekročiť 10^{13}.

6.2 *Support vector machine*

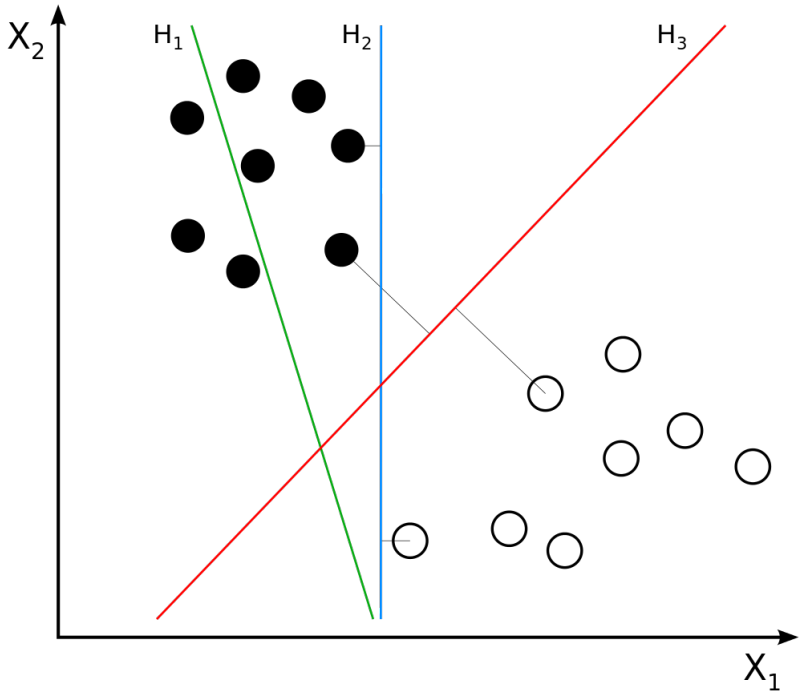
Support vector machine (SVM) je model, ktorý pomocou analýzy dát, rozpoznávania vzorov a postupov, slúži na klasifikáciu dát a regresnú analýzu [8]. Model je potrebné natrénovať pomocou trénovacích dát. Tie sú rozdelené do dvoch skupín, ktoré zodpovedajú dvom kategóriám. Jedna obsahuje dáta patriace do prvej kategórie a druhá skupina obsahuje dáta patriace do druhej kategórie. Trénovací algoritmus vytvorí model, ktorý dáta priradí do určených kategórií. Je to teda binárny lineárny klasifikátor.

SVM model reprezentuje dáta ako body v priestore. Tie sú namapované tak, že skupiny trénovacích dáta sú oddelené pomocou medzery, ktorá by mala byť čo najširšia, aby boli skupiny dát čo najvzdialenejšie od seba navzájom. Nové dáta, ktoré sú na vstupe natrénovaného modelu, sú namapované do rovnakého priestoru a je predikovaná ich príslušnosť k jednej z kategórií. A to vzhľadom k tomu, na ktorej strane medzery sa dáta nachádzajú.

Hlavnými problémami tejto metódy je nájdenie najvhodnejšieho umiestnenia medzery, ktorá oddeľuje dáta v priestore, aby pri klasifikovaní neznámych dát bola metóda najpresnejšia.

Druhým problémom je vysporiadanie sa s mnohorozmerným priestorom. Je to dané tým, že každý člen v modeli je definovaný množinou vlastností (rysov). Každá z ich je vložená do modelu v číselnej podobe. Výhodou oproti iným technikám strojového učenia je v tom, že SVM dokáže rozoznať závislosť medzi vloženými rysmi a výpočtová zložitosť nie je závislá na počte použitých rysov.

Príklad akým SVM mapuje a rozdeľuje dáta v priestore je zobrazený na nasledujúcom obrázku.



Obrázok 6.2: Rozdelenie priestoru modelom SVM, prevzaté z [32].

Kapitola 7

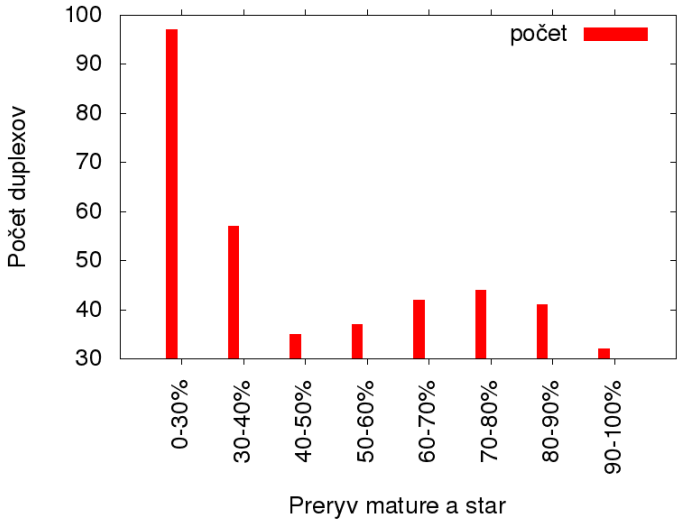
Využitie zhlukovania pri klasifikácii

V nasledujucom texte je opisaná sada experimentov, ktorá skúma, ako je možné využiť zhlukovacie algoritmy pri samotnej klasifikacii miRNA duplexov. Experimenty sú založené na využití vysokej kompementarity *mature* a *star* sekvencií v duplexe. Tá by mala zabezpečiť priradenie týchto sekvencií do rovnakého zhluku pri zhlukovní.

V ideálnom prípade by tieto zhluky boli vytvorené s dostatočne vysokým prahom a obsahovali by menej členov. Teda šanca, že v jednom zhluku by sa nachádzali iba prislúchajúce *mature* a *star* sekvencie miRNA duplexu by bola vyššia. Na zhlukovanie bol využitý program CD–HIT, ktorý v predchádzajúcich častiach práce vykazoval najlepšie výsledky pre požadované úlohy (5.3).

7.1 Zistenie prekryvu pomocou BLAST

Pred samotným zhlukovaním však boli pomocou nástroja BLAST zarovnané *mature* a *star* sekvencie miRNA duplexov, ktoré boli použité aj v predchádzajúcich častiach práce (získané z databáze miRBase, verzia 17). Výsledok je zobrazený v nasledujúcom grafe.



Obrázok 7.1: Zistenie prekryvu *mature* a *star* sekvencií z miRBase ver. 17 nástrojom BLAST.

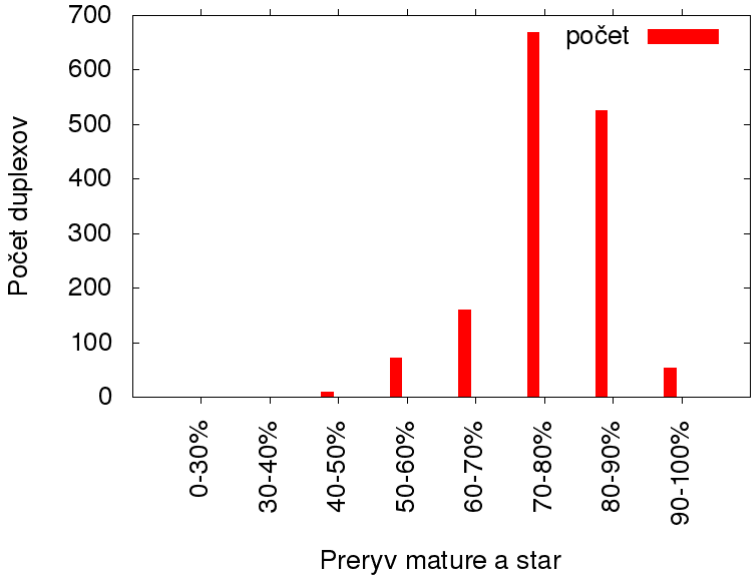
Aspoň 60 percentný prekryv nukleotidov *mature* a *star* sekvencií bol dosiahnutý len v 41 percentách z 387 porovnávaných duplexov. Na overenie, či bude počet prislúchajúcich *mature* a *star* priradených do rovnakého zhluku dostatočný, bolo vykonané zhlukovanie.

7.2 Zistenie prekryvu pomocou Biopython

Z referenčných miRNA z databáze miRBase verzia 17 sa dalo v tomto experimente vytvoriť iba 387 duplexov. Pri ostatných nebolo úplne jasné, ktorá sekvencia s nimi vytvára duplex. Po naštudovaní novej anotácie, ktorú využíva miRBase verzia 21 na označenie *mature* a *star* sekvencií, bolo možné využiť túto databázu. V tejto verzii sú *mature* sekvencie označené koncovkou „-5p“ a *star* sekvencie koncovkou „-3p“[1] .

Na rozdiel od predchádzajúcej verzie, zhoda sekvencií v novej verzii miRNA duplexov bola overená modulom pairwise2 z balíka Bio[2], ktorý ponúka pre jazyk Python rôzne funkcie pre úlohy z bioinformatiky.

Tento modul umožňuje pomocou dynamického programovania získať najlepšie globálne alebo lokálne zarovanie dvoch sekvencií. Dá sa určiť aké skóre bude priradené zhode, nezhode alebo medzere v zarovnaní. Sekvencie duplexov boli zarovnané lokálne, za zhodu bolo pridané skóre 1, za nezhodu a medzeru v zarovnaní nebolo skóre znížené. Zarovnaná bola vždy *mature* sekvencia duplexu a reverzný komplement prislúchajúcej *star* sekvencie. Výsledok tohto zarovnania je zobrazený v nasledujúcom grafe.



Obrázok 7.2: Zistenie prekryvu *mature* a *star* sekvencií z miRBase ver. 21 nástrojom Bio.pairwise2.

Percentuálna zhoda je v tomto prípade oveľa vyššia. Pri zarovnaní nástrojom BLAST boli výsledky nedostatočné. Je to zrejme dané metódou akou tento nástroj zarovnáva sekvencie.

[1] http://www.mirbase.org/blog/2011/04/whats-in-a-name/
[2] http://biopython.org/DIST/docs/api/Bio.pairwise2-module.html/

7.3 Navrhnutá metodika hierarchického zhlukovania

Použitých bolo 5 vstupných súborov, ktoré sa využívali aj v predchádzajúcich častiach práce (RNA samčích listov, samičích listov, neoplodnených piestikov, oplodnených piestikov a peľu). Ku každému z týchto súborov boli pridané referenčné miRNA sekvencie. Týchto upravených 5 súborov poslúžilo ako vstup pre hierarchické zhlukovanie s programom CD–HIT.

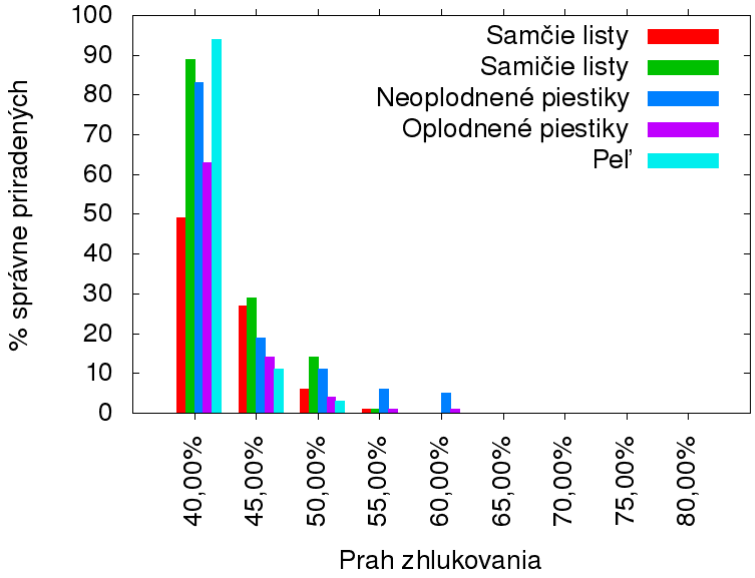
Program CD–HIT–EST nebol použitý, pretože umožňuje nastaviť prah zhlukovania minimálne na 75 percent, čo sa ukázalo ako nedostatočné pre zisk použiteľných výsledkov zhlukovania. Vo väčšine prípadov bolo nutné znížiť prah zhlukovania až na 40 percent, pretože počet *mature* a *star* sekvencií jedného duplexu priradených do jedného zhluku bol príliš nízky pri vyšších prahoch. Navrhnutá metodika využívajúca hierarchické zhlukovanie na zisk potrebných výsledkov prebiehal nasledovným postupom:

1. Zhlukovanie vstupného súboru s prahom 99 percent.
2. Zhlukovanie iba reprezentantov z výstupného súboru predchádzajúceho kroku s prahom 95 percent.
3. Opakovanie kroku 2 so stále nižšími prahmi (90%, 85%, 80%, 75%, 70%, 65%, 60%, 55%, 50%, 45%, 40%). Vstupom sú zakaždým reprezentanti z výstupu práve dokončeného zhlukovania s vyšším prahom.
4. Priradenie čísla zhluku každej sekvencii vo výstupoch z jednotlivých zhlukovaní.
5. Nájdenie referenčných miRNA sekvencií vo výstupe zo zhlukovania s prahom 99%.
6. Podľa čísla zhluku, do ktorého patria referenčné miRNA sa nájde reprezentant ich zhluku a jednotlivé miRNA s reprezentantom sa zapíšu do pomocného súboru spolu s číslom zhluku.
7. Vo výstupnom súbore zo zhlukovania s nižším prahom sa vyhľadá zhluk, v ktorom je reprezentant zhluku s vyšším prahom a číslo zhluku. (Príklad: *mature* sekvencia sa pri zhlukovaní s prahom 99% nachádza v zhluku 99_5 s reprezentantom 99_5. V tomto kroku sa vyhľadá reprezentant 99_5 vo výstupe zo zhlukovania s prahom 95% a zistí sa, že je v zhluku 95_3 s reprezentantom 95_3).
8. Do nového pomocného súboru sa zapíšu rovnaké *mature* a *star* sekvencie ako v predchádzajúcom pomocnom súbore, avšak namiesto reprezentantov pôvodných zhlukov sa k nim priradia reprezentanti zhlukov, ktoré boli nájdené v predchádzajúcom kroku a číslo tohto zhluku (teda namiesto reprezentanta 99_5 a zhluku 99_5 bude so sekvenciou zapísaný reprezentant 95_3 a číslo zhluku 95_3).
9. Opakovanie krokov 7 a 8.
10. V pomocných súboroch, ktoré vznikli v kroku 7 sa vyhľadajú jednotlivé *mature* sekvencie, k nim prislúchajúce *star* sekvencie a podľa čísla ich zhluku sa zistí, či boli priradené do rovnakého zhluku (či *mature* sekvencia z predchádzajúceho súboru a k nemu prislúchúca *star* sekvencia majú číslo zhluku 99_5 v prvom pomocnom súbore alebo 95_3 v druhom pomocnom súbore a tak ďalej).
11. Po zistení koľko percent *mature* a *star* sekvencií z jedného miRNA duplexu sa nachádza v jednom zhluku pre každý prah zhlukovania je algoritmus ukončený.

12. Opakovanie predchádzajúcich krokov s ďalším vstupným súborom.

7.3.1 Zhodnotenie výsledkov

Výsledky sú zobrazené v nasledujúcom grafe.

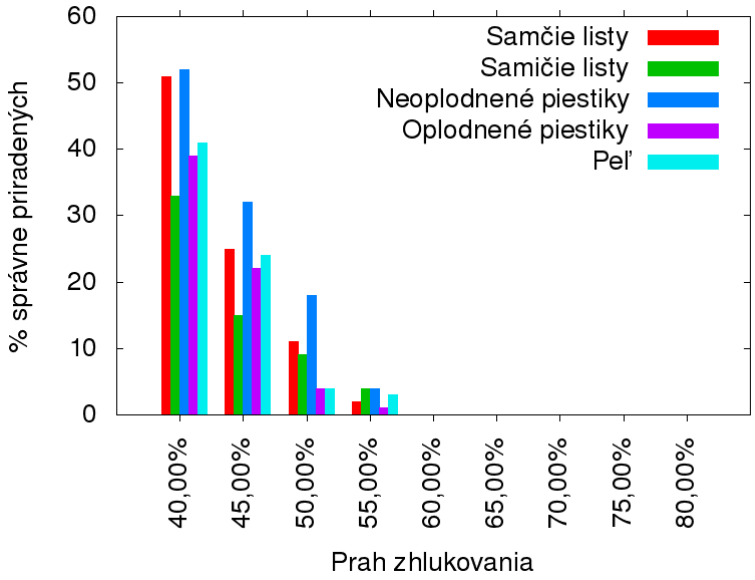


Obrázok 7.3: Porovnanie výsledkov pre prvý experiment so zhlukovaním so sekvenciami z miRBase ver. 17.

Tieto výsledky potvrdili výsledky z programu BLAST. Počet správne priradených *mature* a *star* sekvencií bol postačujúci až pri prahu približne 40 percent. V tomto prípade bol však počet sekvencií v zhluku príliš vysoký a počet zhlukov príliš nízky. Touto navrhnutou metódou sa nezískala takmer žiadna výhoda a očakávané zníženie počtu sekvencií, ktoré treba spracovať.

Po získaní prvých výsledkov boli referenčné miRNA z databáze miRBase verzia 17 nahradené miRNA z databáze miRBase vezia 21. Počet jednoznačných duplexov vytvorených z tejto verzie databáze bol 1487.

Experiment sa teda potom opakoval rovnakým spôsobom, ale referenčné miRNA boli nahradené novou verziou. Výsledky tohto experimentu sú zobrazené v nasledujúcom grafe.



Obrázok 7.4: Porovnanie výsledkov pre prvý experiment so zhlukovaním so sekvenciami z miRBase ver. 21.

Aj napriek vyššiemu počtu referenčných miRNA sekvencií, ktoré tvoria duplex sa výsledok experimentu príliš nezlepšil.

7.4 Upravenie navrhnutej metodiky zhlukovania

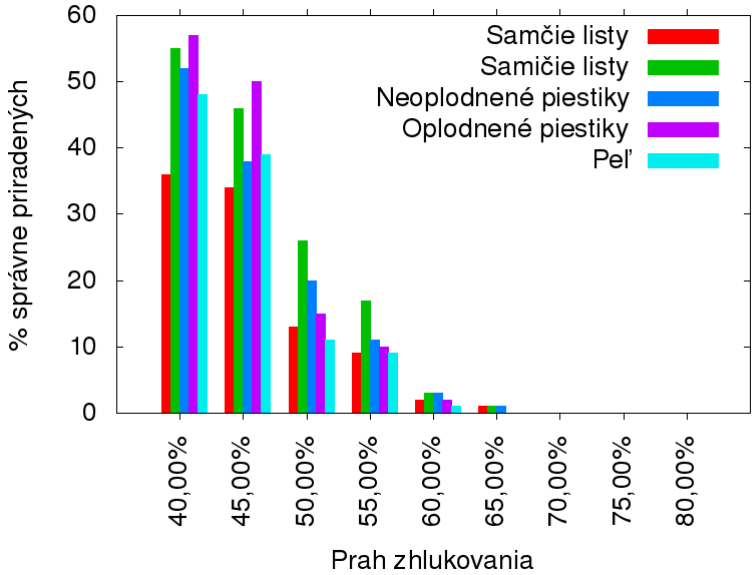
Ďalším spôsobom, akým by bolo možné znížiť počet sekvencií je tiež zhlukovanie pomocou CD–HIT, ale nie hierarchické. V experimente teda bola vždy zhlukovaná pôvodná dátová sada (s pridanými novými referenčnými miRNA), ale postupne sa znižoval prah zhlukovania.

Rozdiel oproti predchádzajúcemu experimentu bol v tom, že predtým sa zhlukovali s nižším prahom iba reprezentanti zhlukov s vyšším prahom. Na vstupe bolo rovnakých 5 súborov ako v predchádzajúcom experimente. Postup nového zhlukovania bol nasledovný:

1. Zhlukovanie vstupného súboru s prahom 99 percent.
2. Priradenie čísla zhluku každej sekvencii vo výstupoch z jednotlivých zhlukovaní.
3. Porovnanie čísla zhluku *mature* a *star* sekvencií z jedného duplexu.
4. Percentuálne vyhodnotenie počtu prislúchajúcich *mature* a *star* sekvencií, ktoré sa nachádzajú v rovnakom zhluku.
5. Opakovanie krokov 1,2,3 a 4 so stále nižšími prahmi (90%, 85%, 80%, 75%, 70%, 65%, 60%, 55%, 50%, 45%, 40%). Vstupom je vždy rovnaký vstupný súbor.
6. Opakovanie predchádzajúcich krokov s ďalším vstupným súborom.

7.4.1 Zhodnotenie výsledkov

Výsledky experimentu sú zobrazené v nasledujúcom grafe.



Obrázok 7.5: Porovnanie výsledkov pre tretí experiment so zhlukovaním so sekvenciami z miRBase ver. 21.

Ani v tomto prípade výsledky neboli dostačujúce pre cieľ, ktorý mali splniť. Preto bol navrhnutý nový spôsob, akým by sa dali klasifikovať miRNA vďaka zhlukovaniu alebo aspoň zmenšiť počet sekvencií, ktoré musia byť spracované iným spôsobom. Ten je opísaný v nasledujúcej kapitole.

Kapitola 8

Návrh implementácie

Nová navrhnutá a implementovaná metóda je založená na tom, že *mature* sekvencia a revezný komplement *star* sekvencie majú pomerne vysoké percento zhody. Preto by sa mohli nachádzať v jednom zhluku, pri dostatočne nízkom prahu. Avšak prah nesmie byť príliš nízky, aby neobsahoval priveľa sekvencií.

Sekvencie ktoré sa teda nachádzajú v jednom zhluku s *mature* alebo *star* sekvenciami a majú s nimi vysokú percentuálnu zhodu by mohli byť označené ako miRNA sekvencie. Sekvencie, ktoré sa nenachádzajú v zhlukoch s *mature* alebo *star* sekvenciami sú určené ako vstup strojovému učeniu, ktoré určí pravdepodobnosť, či patria alebo nepatria medzi miRNA. Výber týchto sekvencií prebieha s cieľom znížiť ich počet na prijateľnú hodnotu.

V experimente boli znovu použité rovnaké vstupné súbory a nové referenčné miRNA sekvencie (z miRBase ver. 21). Na zhlukovanie bol použitý program CD–HIT, ktorý umožňuje nastaviť prah na nižšie hodnoty ako program CD–HIT–EST. Kedže program CD–HIT neumožňuje nastaviť, aby sa vstupné sekvencie pridávali do zhlukov aj na základe percentuálnej zhody reverznej sekvencie, tieto sekvencie musia byť vytvorené skriptom a pridané k ostatným vstupným sekvenciám. Na rozlíšenie priamych a reverzných sekvencií bol pridaný prefix k názvu aj priamych aj reverzných sekvencií.

8.1 Metodika výberu dát pomocou zhlukovania

1. Sekvenciám zo vstupného súboru a referenčným miRNA sekvenciám je pridaný prefix, ktorý určuje, že sú to priame sekvencie.

2. K vstupným a miRNA sekvenciám sú vytvorené reverzné sekvencie a k nim je pridaný prefix, ktorý určuje túto ich vlastnosť.

3. Všetky priame a reverzné vstupné a priame a reverzné miRNA sekvencie sú určené ako vstup pre zhlukovanie.

4. K sekvenciám vo výstupnom súbore zo zhlukovania je pridané číslo zhluku v ktorom sa nachádzajú.

5. Prechádzajú sa jednotlivé priame a reverzné referenčné sekvencie a zisťuje sa číslo ich zhluku.

6. Sekvencie, ktoré sa nachádzajú v rovnakom zhluku ako niektoré z referenčných miRNA sekvencií a zároveň s nimi majú vysokú percentuálnu zhodu, sú určené ako miRNA

sekvencie. Zároveň je zistená rodina referenčnej miRNA sekvencie. Do tejto rodiny pravdepodobne patrí aj získaná sekvencia.

7. Prechádzajú sa zhluky, ktoré neobsahujú priame ani reverzné referenčné miRNA sekvencie.

8. Zistí sa početnosť dĺžok sekvencií v týchto zhlukoch (dĺžka každej sekvencie je zobrazená v riadku danej sekvencie).

9. Početnosť sa zoradí od najvyššiu po najnižšiu.

10. Nájde sa sekvencia, ktorá má najpočetnejšiu dĺžku (ale menšiu ako 50 nukleotidov) a zároveň sto percentnú zhodu s reprezentantom zhluku, v ktorom sa nachádza (percentuálna zhoda s reprezentantom je tiež zobrazená v riadku danej sekvencie).

11. Podla prefixu v názve sa zistí, či je to sekvencia priama alebo reverzná.

12. V prípade, že je to sekvencia priama, vyhľadá sa k nej reverzná sekvencia. V opačnom prípade sa k nej vyhľadá priama sekvencia. Hľadaná sekvencia musí mať s pôvodnou aspoň 70 percentnú zhodu (keďže pôvodná sekvencia má 100 percentnú zhodu s reprezentantom, môže sa pri hľadaní opačne orientovanej sekvencii použiť údaj o jej zhode s reprezentantom zhluku).

13. Sekvencia s najvyššou dĺžkovou početnosťou (možná *mature* sekvencia) je spolu s novou nájdenou sekvenciou (možná *star* sekvencia) zapísaná do výstupného súboru.

14. Kroky 10, 11, 12 a 13 sa opakujú, avšak nepoužije sa už sekvencia s najpočetnejšou dĺžkou, ale s druhou, treťou a štvrtou najpočetnejšou dĺžkou (musia však byť kratšie ako 50 nukleotidov). Tak vznikli z každého zhluku 4 možné duplexy.

15. Predchádzajúce kroky sa opakujú pre všetkých päť vstupých súborov.

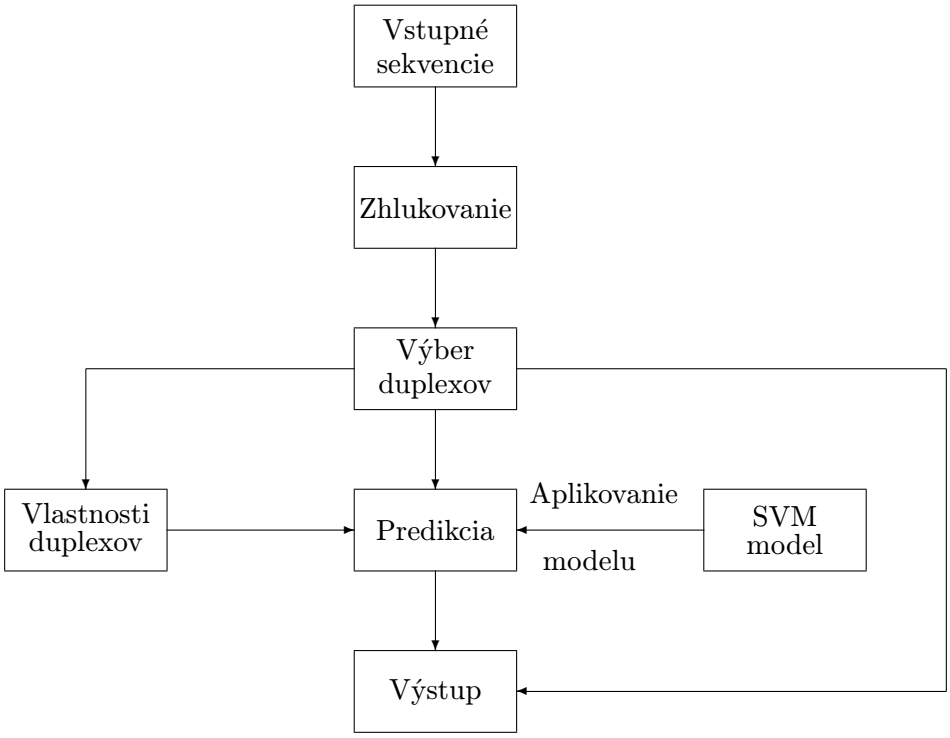
16. Získané možné duplexy sa stanú vstupom pre natrénovaný model strojoveho učenia, ktorý určí, či sa jedná o miRNA sekvencie alebo nie.

V postupe sa vyhľadávajú najpočetnejšie dĺžky sekvencií z toho dôvodu, že *mature* sekvencie sa nachádzajú v datasetoch s oveľa vyššou početnosťou ako *star* sekvencie. Početnosť dĺžok v zhluku teda slúži ako nástroj na nájdenie *star* sekvencií.

8.2 Schéma postupu

Vstupné dáta sú teda zhlukované programom CD–HIT. Podľa príslušnosti ku zhluku, ktorý obsahuje referenčné miRNA sekvencie (a vysokej podobnosti s nimi) sú určené pravdepodobné miRNA sekvencie a ich možná rodina.

Zo zhlukov, ktoré neobsahujú referenčné miRNA sekvencie sú určené sekvencie, ktoré by mohli by *mature*, k nim sú priradené možné *star* sekvencie. Ziskajú sa ich vlastnosti a pomocou natrénovaného SVM modelu je určené, či sa jedná o miRNA duplex alebo nie. Tento postup je zobrazený na nasledujúcom obrázku



Obrázok 8.1: Postup pri klasifikácii.

8.3 Použité vlastnosti duplexov a ich analýza

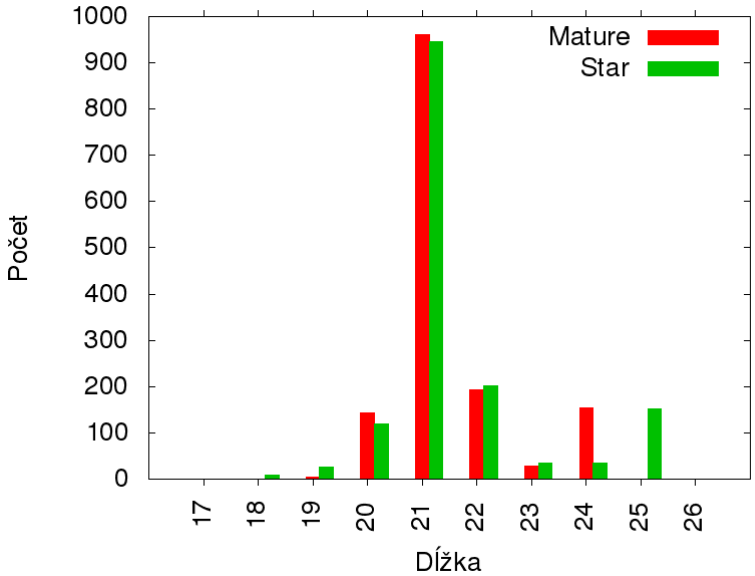
Vlastnosti duplexov, ktoré boli použité pri predikcii pomocou *support vector machine* sa dajú rozdeliť do 3 skupín: veľkosť, stabilita a zloženie [10]. Tieto kategórie sú zobrazené v tabuľke 8.3 a opísané v nasledujúcej časti práce. Rozdelenie vlastností je inšpirované autormi nástroja MirPlex, ale ich analýza bola vykonaná na rastlinných miRNA.

Dĺžka	**Stabilita**	**Zloženie**
Dĺžka *mature*	Skóre komplementarity	*mature* 1. báza
Dĺžka *star*	Počet nezhôd	*star* 1. báza
Rozdiel dĺžok	Počet G/U párov	*mature* G obsah
	Vypukliny	*mature* C obsah
	3´ prevísajúci koniec *mature*	*mature* A obsah
	3´ prevísajúci koniec *star*	*mature* U obsah
	Minimálna voľná energia	*star* G obsah
		star C obsah
		star A obsah
		star U obsah

Tabulka 8.1: Vlastnosti duplexu, prevzaté z [10].

8.3.1 Veľkosť

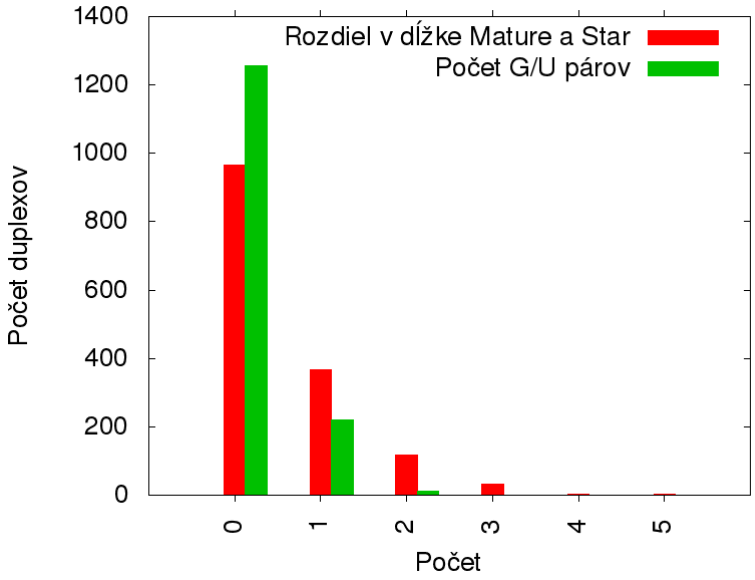
Dicer má silnú tendenciu rozdeliť pre-miRNA na špecifické dĺžky a produkuje *mature* a *star* sekvencie s dĺžkou približne 21-23 nukleotidov. Analýza rozdelenia dĺžok *mature* miRNA sekvencií z databázy miRBase verzia 21 je zobrazená v nasledujúcom grafe. Sú v ňom zobrazené dĺžky *mature* a *star* sekvencií, ktoré boli použité pri vytváraní pozitívneho datasetu pri trénovaní SVM. Najpočetnejšia dĺžka oboch sekvencií duplexov trénovacieho setu je jednoznačne 21 nukleotidov.



Obrázok 8.2: Dĺžky miRNA sekvencií z pozitívneho datasetu, verzia 21.

Jednou z použitých vlastností teda bola dĺžka *mature* a *star* sekvencií. Štruktúra enzýmu Dicer určuje, že je pravdepodobnejšie, že odrežie miRNA* sekvencie podobnej dĺžky ako *mature* sekvencie. Preto by rozdiel medzi ich dĺžkou mal byť nulový. Výnimkou sú duplexy, ktoré obsahujú asymetrickú vypuklinu (bulge). Teda rozdiel dĺžok môže poslúžiť ako ďalší parameter. Na nasledujúcom grafe je zobrazený rozdiel dĺžok *mature* a *star* sekvencií duplexov trénovacej sady. Tiež počet G/U párov v týchto duplexoch.

Najčastejší prípadom je rovnaká dĺžka oboch sekvencií v duplexe. Vo vyše 200 prípadoch je rozdiel v dĺžke jeden nukleotid. Podobne je to aj pri počte G/U párov. Väčšinou sa v duplexe nenachádza ani jeden. Vo vyše 200 prípadoch je to práve jeden pár. Ostatné hodnoty sú zanedbateľné.



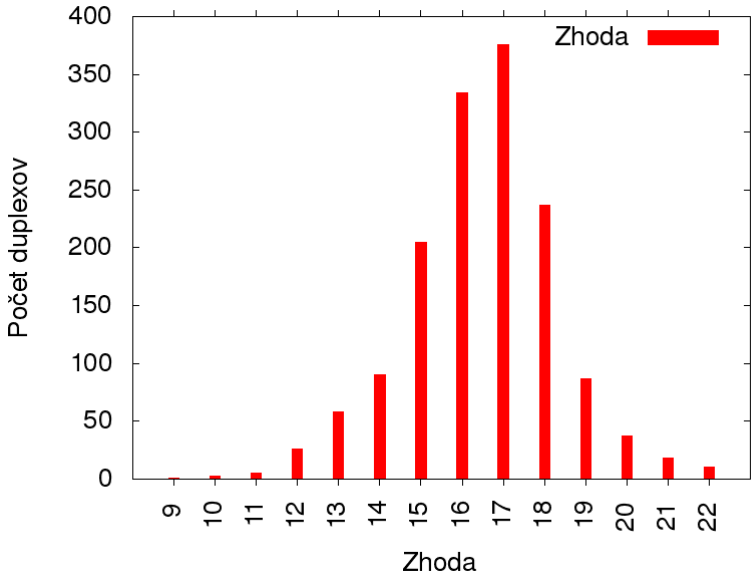
Obrázok 8.3: Rozdiel dĺžok a počet G/U párov *mature* a *star* sekvencií v duplexoch z pozitívneho datasetu.

8.3.2 Stabilita

Dicer pri vytváraní miRNA necháva dva prečnievajúce nukleotidy na *mature* a *star* vlákne. Preto bol počet prečnievajúcich nukleotidov na oboch vláknach použitý ako ďalší parameter. Typický duplex taktiež obsahuje nezhody (*mismatches*) medzi nukleotidmi. Tým vznikajú symetrické alebo asymetrické vypukliny.

Medzi ďalšie parametre teda patrí počet nezhôd a výskyt vypuklín. miRNA sú tiež energeticky stabilnejšie a majú nižšiu volnú energiu ako náhodné sekvencie. Táto energia určuje stabilitu na základe spárovania nukleotidov a samotnej štruktúry duplexu [9]. Majú teda vyššiu tendenciu udržať si stabilnú sekundárnu štruktúru. Ako posledné parametre v tejto skupine boli teda pridané: volná energia, počet G/U párov a celková zhoda sekvencií duplexu.

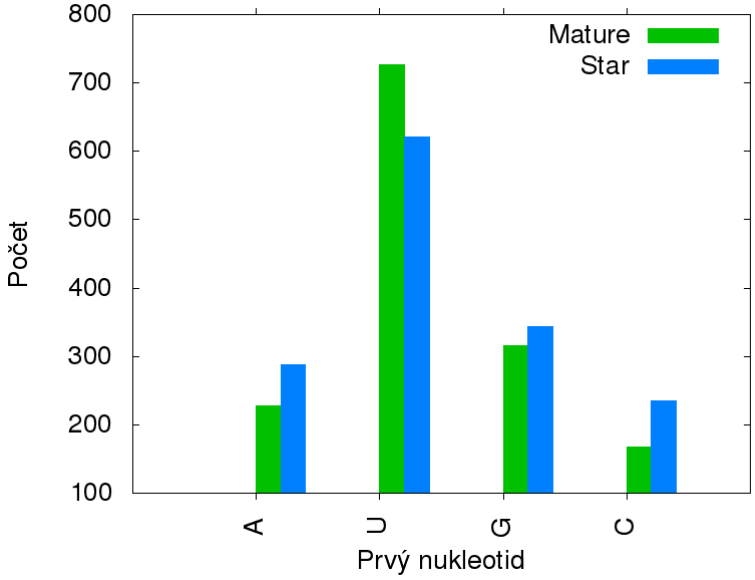
Na nasledujúcom grafe je zobrazená zhoda jednotlivých nukleotidov *mature* a *star* sekvencií v duplexe. Najčastejšie sú hodnoty 17 a 16. Vyše 200 krát sa v trénovacích dátach nachádzajú aj zhody 15 a 16 nukleotidov.



Obrázok 8.4: Zhoda *mature* a *star* sekvencií v duplexoch z pozitívneho datasetu.

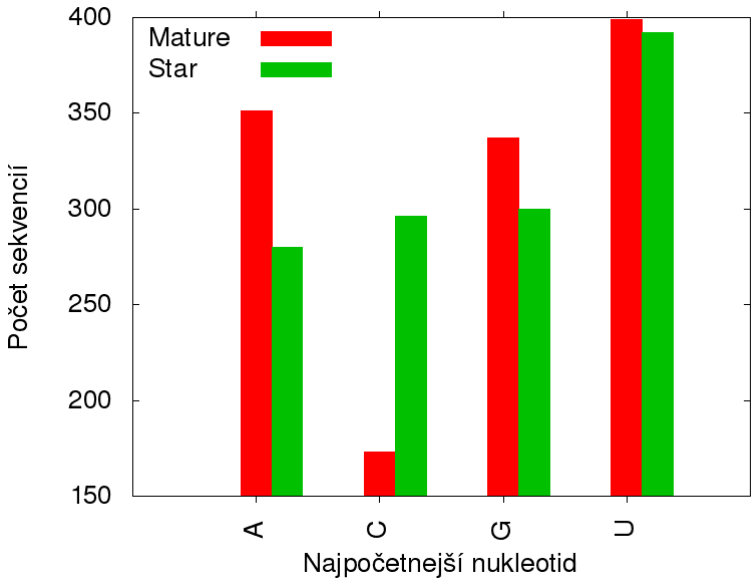
8.3.3 Zloženie

Väčšina *mature* sekvencií obsahuje prvý nukleotid na 5' konci uracil. Na nasledujúcom grafe je zobrazená analýza prvých nukleotidov *mature* a *star* sekvencií z databáze miRBase v21, ktoré boli použité ako pozitívny dataset. Uracil je najčastejšie prvým nukleotidom pri *mature* aj *star* sekvenciách.



Obrázok 8.5: Početnosť nukleotidov na prvej pozícii miRNA sekvencií z pozitívneho datasetu.

Percento výskytu jednotlivých nukleotidov v *mature* a *star* sekvenciách a prvý nukleotid v sekvencii boli použité ako ďalšia vlastnosť. Pre zjednodušenie je na nasledujúcom grafe zobrazený počet, koľko *mature* a *star* sekvencií obsahuje ako najpočetnejší nukleotid A, C, G, alebo U. V oboch typoch sekvencií je najpočetnejší väčšinou uracil. Zaujímavý údajom je malý počet *mature* sekvencií, ktoré majú najpočetnejší nukleotid cytozín.



Obrázok 8.6: Najpočetnejší nukeleotid v sekvenciách.

8.4 Vytvorenie modelu SVM

8.4.1 Metóda tvorby negatíveho a pozitívneho datasetu

Na vytvorenie SVM modelu a klasifikáciu vstupných dát bola použitá knižnica dostupná aj pre jazyk Python, LIBSVM[1], ktorá umožnuje škálovanie vstupných dát, trénovanie modelu, nájdenie najlepších parametrov, samotnú klasifikáciu a vykonanie ďalší úloh.

Trénovacie dáta SVM modelu obsahovali negatívny a pozitívny dataset. Dôležité bolo, aby počet pozitívnych a negatívnych duplexov počas trénovania modelu bol rovnaký, teda aby bola trénovacia dátová sada vyvážená. Negatívny dataset by sa taktiež nemal príliš odlišovať od pozitívneho, ale nemal by byť ani príliš podobný. Preto boli v nasledujúcom postupe hľadané sekvencie pre negatívny dataset, ktoré majú podobné dĺžky ako sekvencie v pozitívnom datasete.

Pozitívny dataset bol vytvorený z miRNA sekvencií z databáze miRBase verzie 21. Duplexy boli vytvorené na základe názvu *mature* a *star* sekvencií. V tejto verzii databáze miRBase sú *mature* sekvencie označené koncovkou „5p“ a *star* sekvencie koncovkou „3p“. Duplex bol bol teda vytvorený nájdením prislúchajúcich „5p“ a „3p“ sekvencií. Napríklad k *mature* sekvencii s názvom „ath-miR156a-5p“ bola priradená *star* sekvencia s rovnakým názvom, ale inou koncovkou: „ath-miR156a-3p“. Týmto postupom bolo vytvorených 1487 duplexov, ktoré boli použité ako pozitívny dataset pre učenie SMV modelu.

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Sekvencie tvoriace negatívny dataset boli pôvodne vytvorené nasledujúcim navrhnutým postupom, ktorý sa v niektorých bodoch podobá postupu na nájdenie sekvencií, ktoré sú použité ako vstup pre SVM model:

1. Sekvenciám zo vstupného súboru je pridaný prefix, ktorý určuje, že sú to priame sekvencie.
2. K sekvenciám sú vytvorené reverzné sekvencie a k nim je pridaný prefix, ktorý určuje túto ich vlastnosť.
3. Všetky priame a reverzné vstupné sekvencie sú určené ako vstup pre zhlukovanie.
4. K sekvenciám vo výstupnom súbore zo zhlukovania je pridané číslo zhluku v ktorom sa nachádzajú.
5. Prechádzajú sa jednotlivé priame a reverzné referenčné sekvencie a zisťuje sa číslo ich zhluku.
6. Prechádzajú sa jednotlivé zhluky.
7. Zistí sa početnosť dĺžok sekvencií v týchto zhlukoch (dĺžka každej sekvencie je zobrazená v riadku danej sekvencie).
8. Početnosť sa zoradí od najvyššiu po najnižšiu.
9. Nájde sa sekvencia, ktorá má najpočetnejšiu dĺžku (ale menšiu ako 50 nukleotidov) a zároveň sto percentnú zhodu s reprezentantom zhluku, v ktorom sa nachádza (percentuálna zhoda s reprezentantom je tiež zobrazená v riadku danej sekvencie).
10. Podla prefixu v názve sa zistí, či je to sekvencia priama alebo reverzná.
11. V prípade, že je to sekvencia priama, vyhľadá sa k nej reverzná sekvencia. V opačnom prípade sa k nej vyhľadá priama sekvencia. Hľadaná sekvencia musí mať s pôvodnou menej ako 70 percentnú zhodu (keďže pôvodná sekvencia má 100 percentnú zhodu s reprezentantom, môže sa pri hľadaní opačne orientovanej sekvencii použiť údaj o jej zhode s reprezentantom zhluku). Dôvod je ten, že väčšina sekvencií v miRNA duplexoch má zhodu vyššiu ako 70 percent a preto sa pre negatívny dataset hľadajú sekvencie s nižšou zhodou. Zníži sa tak podobnosť pozitívneho a negatívneho datasetu.
12. Sekvencia s najvyššou dĺžkovou početnosťou (možná *mature* sekvencia) je spolu s novou nájdenou sekvenciou (možná *star* sekvencia) zapísaná do výstupného súboru.
13. Kroky 9, 10, 11 a 12 sa opakujú avšak nepoužije sa už sekvencia s najpočetnejšou dĺžkou, ale s dtuhou, treťou a štvrtou najpočetnejšou dĺžkou (musia však byť kratšie ako 50 nukleotidov). Tak vzniknú z každého zhluku 4 možné duplexy.
14. Predchádzajúce kroky sa opakujú pre všetkých päť vstupých súborov.
15. Získané možné duplexy sa stanú negatívnym datasetom pre trénovanie modelu strojového učenia.

Tento postup vyprodukoval príliš málo sekvencií duplexov, ktorých dĺžka bola podobná dĺžkam referenčných duplexov, ktoré tvorili pozitívny dataset a zároveň podobnosť hľadaných negatívnych *mature* a *star* sekvencií bola nižšia ako 70 percent. Preto bol postup mierne upravený.

Jednotlivé kroky postupu zostali rovnaké, avšak na vstupe zhlukovacieho algoritmu nebolo postupne 5 rozdielnych súborov s priamymi a reverznými sekvenciami. Na vstupe bol súbor, v ktorom bolo spojených všetkých päť vstupných súborov s ich priamymi a reverznými sekvenciami. Vznikol tak viac ako 8 GB súbor. Tento krok je dôležitý aj z toho dôvodu, aby sa efektívne spracovali aj sekvencie, ktoré sú silne exprimované len v niektorom z orgánov (v niektorom z piatich vstupných súborov). Tento výsledok bol pridaný k už získanému negatívnemu datasetu.

8.4.2 Prevod vlastností do formátu vhodného pre SVM

Po získaní negatívneho a pozitívneho datasetu nasledovalo získanie vlastností týchto duplexov a ich prevod do vhodnej podoby pre SVM. Pri učení SVM, ale aj pri samotnej klasifikácii musia byť jednotlivé záznamy, ktoré majú byť klasifikované, na samostatných riadkoch. Prvým údajom je príslušnosť k jednej z dvoch skupín (negatívny alebo pozitívny dataset). Pre pozitívny dataset bol použitý údaj „+1“, pre negatívny „-1“. Nasleduje 22 znakov, ktoré charakterizujú daný duplex:

- Dĺžka *mature* a *star* sekvencie.
- Absolútna hodnota rozdielu dĺžky *mature* a *star* sekvencie.
- Minimálna voľná energia duplexu.
- Prvý nukleotid *mature* a *star* sekvencie.
- Počet prevísajúcich nukleotidov na 5’ a 3’ konci *mature* a *star* sekvencie.
- Počet G/U párov a nezhôd (mismatch) v duplexe.
- Skóre zhody v duplexe.
- Binárny údaj o existencii vypuklín v duplexe (hodnota 1, ak existuje v duplexe jedna alebo viac vypuklín).
- Percentá výskytu jednotlivých A,C,G a U nukleotidov v *mature* sekvencii.
- Percentá výskytu jednotlivých A,C,G a U nukleotidov v *star* sekvencii.

Poslednými údajmi záznamu sú jednotlivé nukleotidy *mature* a *star* sekvencie, prevedené do numerickej podoby. Vlastnosti duplexov, teda nie iba jednej z ich sekvencií (Počet G/U párov, skóre zhody, počet nezhôd a podobne) boli získané spracovaním výsledkov modulu pairwise2. Tento modul jednotlivé *mature* a *star* sekvencie zarovnal do duplexov, ktoré boli ďalej spracované a boli z nich získané potrebné informácie.

Minimálna voľná energia bola získaná z výstupu programu RNAfold[2], ktorý zo vstupného súboru s duplexami zistí ich minimálnu voľnú energiu a zapíše ju do výstupného súboru. Z tohto súboru boli potrebné údaje získané. Rovnakým spôsobom získava tieto informácie implementovaná aplikácia pri spracovaní neznámych dát.

[2] http://www.tbi.univie.ac.at/RNA/RNAfold.html

8.4.3 Trénovanie SVM modelu

Pozitívny a negatívny dataset bol následne spojené do súboru, ktorý sa použil pri trénovaní SVM modelu. Konkrétne rysy boli následne škálované, aby ich hodnoty boli v určitom rozsahu. Je to dôležitý krok, pretože sa tým zabráni dominancii rysov s vyššími číselným rozsahom nad rysmi s nižšími číslenými rozsahom.

Ďalšou výhodou je vyhnutie sa numerickým problémom počas výpočtu. Vyššie hodnoty vlastností by mohli spôsobiť určité komplikácie. Hodnoty preto boli prevedené do rozsahu [-1,1] pomocou nástroja svm–scale, ktorý je súčasťou LIBSVM knižnice. Daný rozsah hodnôt bol uložený a je používaný pri škálovaní vstupných hodnôt pri klasifikácii. Vďaka tomuto kroku sa zvyšuje presnosť SVM modelu.

Pri trénovaní modelu bolo použité RBF (*radial basis function*) jadro. Toto jadro dokáže, na rozdiel od lineárneho jadra, namapovať vzorky do viazrozmerného priestoru aj v prípade, že vzťah medzi označením triedy a vlastnosťami je nelineárny.

Ďalším dôvodom je počet hyperparametrov, ktoré ovplyvňujú komplexitu selekcie modelu. Polynomiálne jadro má viac hyperparametrov ako RBF jadro.

Pri RBF jadre sú podstatné 2 parametre: C a γ [8]. Pri vyššej hodnote C je cieľom vyššia presnosť klasifikácie s využitím viacerých vzoriek ako podporných vektorov. γ definuje aký veľký vplyv má na klasifikáciu jeden trénovací prvok. Nízka hodnota γ určuje väčší vplyv jedného trénovacieho prvku na klasifikáciu. Naopak vysoká hodnota γ určuje nižší vplyv jedného trénovacieho prvku na klasifikáciu.

Nie je vopred jasné, aké hodnoty týchto parametrov sa majú nastaviť. Cieľom je teda identifikovať najlepšiu kombináciu týchto parametrov pre trénovacie dáta, aby model klasifikoval neznáme dáta čo najlepšie. Na to sa využíva metóda *cross–validation* a *grid–search*. Opis, použitie a výsledky týchto metód sú uvedené v nasledujúcej kapitole.

Kapitola 9

Výsledky

9.1 *Cross–validation* a *grid–search*

Všeobecná stratégia *cross–validation* je založená na rozdelení trénovacích dát na dve časti, z ktorých je jedna považovaná za neznámu. Úspešnosť predikcie „neznámych“ prvkov zobrazuje presnejšie výkonnosť pri klasifikácii nezávislého datasetu. *Cross–validation* zabraňuje pretrénovaniu (*overfitting problem*), teda vzniku náhodných chýb alebo šumu vo výstupe. Najpoužívanejšie sú nasledovné 3 metódy [16].

9.1.1 metódy *cross–validation*

- *Holdout metóda*: Najjednoduchšia forma *cross–validation*. Dáta sú rozdelené do 2 častí. Jedna z nich je trénovacia, druhá testovacia. Model natrénovaný pomocou trénovacej časti sa použije na predikciu testovacích dát. Počet nesprávne odhadnutých prvkov slúži na ohodnotenie modelu. Výhodou je rýchlosť tejto metódy. Nevýhodou je presnosť, pretože výsledok závisí na tom, ktoré dáta sa dostanú do trénovacej a testovacej sady. Teda presnosť ovplyvňuje spôsob, ktorým sú dáta rozdelené.

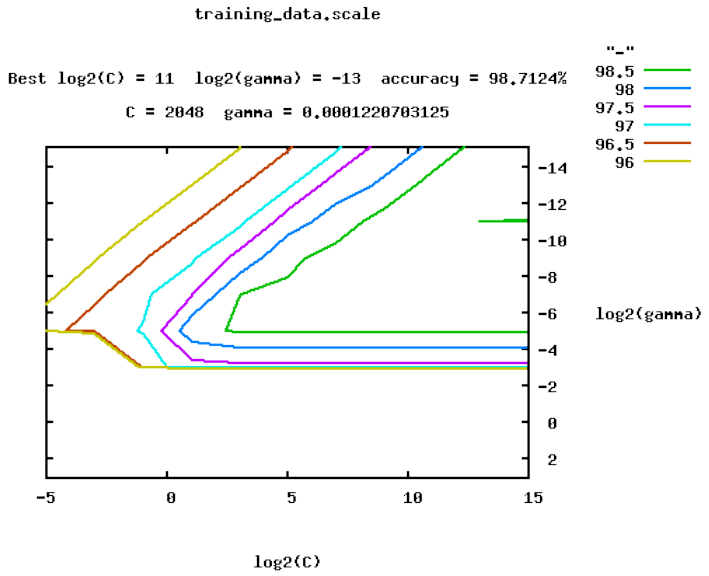
- *v–fold cross–validation*: Ide o vylepšenie predchádzajúcej metódy. Dáta sa rozdelia do v častí rovnakej veľkosti. Postupne je každá časť testovaná modelom, ktorý bol natrénovaný pomocou zvyšných v-1 častí. Takto sa predikuje každá časť trénovacích dát a presnosť *cross–validation* je určená ako percento dát, ktoré boli klasifikované správne. Výhodou je, že menej záleží na spôsobe rozdelenia dát. Každý prvok je testovaný práve raz a v trénovacej sade je v-1 krát. Nevýhodou môže byť to, že trénovací algoritmus musí byť spustený odznova v krát, čím sa ohodnotenie modelu stáva v krát pomalšie ako v predchádzajúcej verzii. Variantom tejto metódy je náhodné rozdelenie dát na trénovacie a testovacie v krát. Výhodou je možnosť výberu veľkosti každej testovacej časti.

- *Leave–one–out cross–validation*: Ide o *v–fold cross–validation*, kde v sa rovná počtu prvok v dátovej sade. Model je teda trénovaný N krát s všetkými dátami okrem jedného prvku, ktorý model následne predikuje. Presnosť ohodnotenia modelu je vysoká a pomerne jednoducho realizovateľná.

9.1.2 *Grid–search*

Metóda *grid–search* na nájdenie C a γ v tomto prípade využíva *cross–validation*. Sú určené rôzne kombinácie C a γ a kombinácia, ktorá poskytne najlepšie *cross–validation* výsledky, bude použitá pri trénovaní modelu. *Grid–search* umožňuje aj paralizovanie výpočtu, pretože C a γ sú navzájom nezávislé.

Na nasledujúcom grafe je zobrazený výsledok tejto metódy, teda výkonnosť klasifikátora na základe kombinácie parametrov C a γ, získaný pomocou knižnice LIBSVM. Najvhodnejšou kombináciou je pre dané trénovacie dáta C=2048.0 a γ=0.0001220703125.



Obrázok 9.1: *Výstup metódy grid–search vykonanej na trénovacích dátach.*

9.2 Meranie výkonnosti klasifikácie

Správnosť klasifikácie môže byť ohodnotená počtom správne odhadnutých členov triedy (*true positive*, *tp*), nesprávne priradených členov do triedy (*false positive*, *fp*), správne odhadnutých členov, ktorí nepatria do triedy (*true negative*, *tn*) a nesprávne odhadnutých členov, ktorí patria do triedy, ale neboli do nej priradení (*false negative*, *fn*).

9.2.1 *Cross–validation*

Hodnoty boli zistené pre každý z 10 behov *10–fold cross–validation* na trénovacej dátovej sade. Klasifikátor bol v každom behu natrénovaný dátovou sadou s 2520 duplexami. Každá testovacia sada bola zložená z 280 duplexov. Počet negatívnych a pozitívnych duplexov v testovacej a trénovacej sade bol v každom behu vyrovnaný. Získané hodnoty sa nachádzajú v nasledujúcej tabuľke.

Ďalšie kritéria, ktoré charakterizujú binárnu klasifikáciu sú [19]:

Beh	**Trénovacia sada**	**Testovacia sada**	***False Positive***	***True Negative***	***True Positive***	***False Negative***
1.	2520	280	2	138	138	2
2.	2520	280	6	134	140	0
3.	2520	280	6	134	138	2
4.	2520	280	3	137	136	4
5.	2520	280	10	130	140	0
6.	2520	280	1	139	132	8
7.	2520	280	5	135	133	7
8.	2520	280	8	132	138	2
9.	2520	280	10	130	138	2
10.	2520	280	0	140	137	3

Tabulka 9.1: Výsledok *10–fold cross–validation.*

- Presnosť (*Accuracy*): Celková efektívnosť klasifikátora. Výpočet:

$$\frac{tp + tn}{tp + fn + fp + tn}$$

- Akosť (*Precision*): Správnosť určenia dát, ktoré boli klasifikované ako pozitívne. Výpočet:

$$\frac{tp}{tp + fp}$$

- Citlivosť (*Sensitivity*): Efektívnosť klasifikátoru identifikovať pozitívny dataset. Výpočet:

$$\frac{tp}{tp + fn}$$

- *F–score*: Vzťah medzi pozitívnym označením a klasifikátorom priradeným označením. Výpočet:

$$2 * \frac{Precision * Sensitivity}{Precision + Sensitivity}$$

- Špecifickosť (*Specificity*): Určenie ako efektívne dokáže klasifikátor identifikovať negatívny dataset. Výpočet:

$$\frac{tn}{fp + tn}$$

- *AUC* (*Area Under the Curve*), teda oblasť pod krivkou: Schopnosť klasifikátoru predísť nesprávnemu označeniu. Výpočet:

$$\frac{1}{2}(\frac{tp}{tp + fn} + \frac{tn}{tn + fp})$$

Vypočítané hodnoty týchto kritérií sú v nasledujúcej tabuľke.

Vlastnosť	Hodnota
Presnosť (*Accuracy*)	97.11 %
Akosť (*Precision*)	96.41 %
Citlivosť (*Sensitivity*)	97.86 %
F–score	97.13 %
Špecifickosť (*Specificity*)	96.36 %
AUC	97.11 %

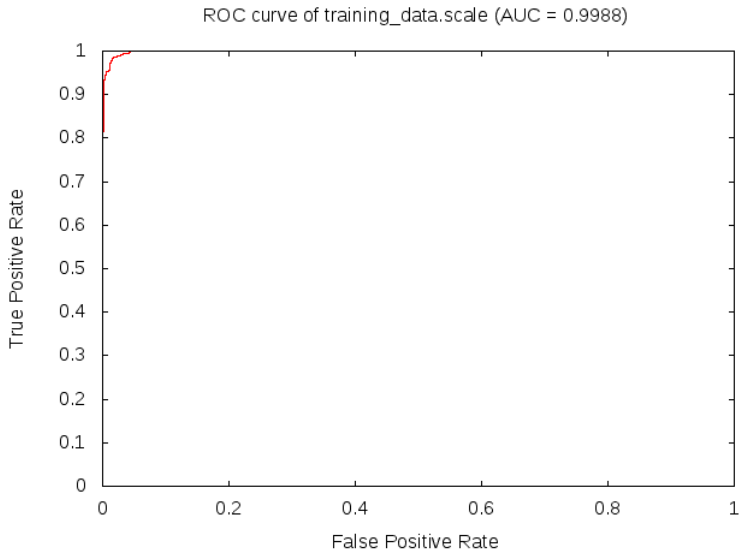
Tabulka 9.2: Vypočítané vlastnosti klasifikátora.

9.2.2 ROC krivka

Na grafické zobrazenie výkonnosti binárneho klasifikátoru slúži ROC (*receiver operating characteristic*) krivka. Krivka zobrazuje hodnotu *true positive* v závislosti od hodnoty *false positive* [14]. Takže na vytvorenie tejto krivky sú potrebné iba 2 výstupné hodnoty každého behu cross-validation.

Čím je metóda na klasifikáciu úspešnejšia, teda čím viac členov je určených správne za pozitívnych a čím menej členov je určených nesprávne za pozitívnych, tým je krivka bližšie k ľavému hornému rohu grafu. Ak by binárny klasifikátor určoval prvky náhodne, jeho krivka by mala tvar diagonály z ľavého dolného rohu do pravého horného rohu grafu. Body krivky, ktoré sa nachádzajú nad touto diagonálou predstavujú dobré výsledky klasifikácie (alebo aspoň lepšie ako náhodné) a body ktoré sa nachádzajú pod touto diagonálou predstavujú zlé výsledky klasifikácie (horšie ako náhodné).

ROC krivka vytvorená z výstupu *10–fold cross–validation* na trénovacích dátach je zobrazená na nasledujúcom grafe. Na vytvorenie bol použitý nástroj plotroc.py z knižnice LIBSVM[1].



Obrázok 9.2: ROC krivka vytvorená z výstupu *10–fold cross–validation*.

[1] http://www.csie.ntu.edu.tw/cjlin/libsvmtools/

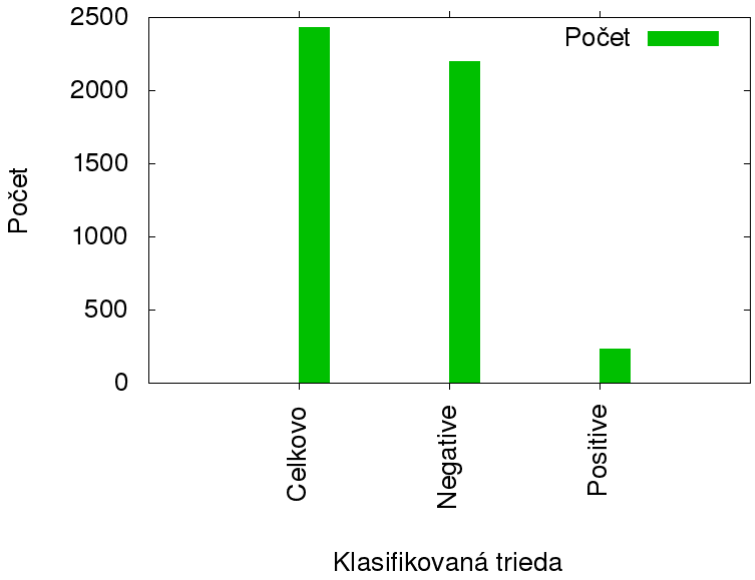
Krivka sa nachádza blízko pri ľavom hornom rohu grafu. To potvrdzuje dobré výsledky zobrazené v predchádzajúcich tabuľkách a teda aj pomerne vysokú presnosť klasifikátora na trénovacích dátach.

Vysoká presnosť na trénovacích dátach však môže znamenať, že negatívny dataset použitý pri trénovaní SVM modelu bol príliš odlišný od pozitívneho datasetu. To by mohlo viesť k nepresnostiam pri klasifikovaní neznámych dát. V tom prípade by ako možné vylep-šie nástroja v budúcnosti, mohlo patriť zlepšenie výberu negatívneho datasetu. Na overenie funkčnosti klasifikácie na neznámych dátach bol vykonaný nasledujúci experiment.

9.3 Klasifikácia náhodne vytvorených párov sekvencií

Na ďalšie, aj keď menej presné overenie výkonnosti a presnosti klasifikátora boli získané náhodné dvojice sekvencií z piatich súborov, ktoré boli použité v predchádzajúcich častiach práce (RNA zo samčích a samičích listov, neoplodnených a oplodnených piestikov a z peľu). Z každého súboru bolo získaných náhodných 500 párov sekvencií. Z výsledných 2500 párov boli vyfiltrované identické dvojice. Zostalo 2432 párov sekvencií, ktoré boli klasifikované pomocou natrénovaného SVM klasifikátora.

Predpoklad bol, že náhodne kombinované sekvencie, ktoré majú s vysokou pravdepodobnosťou rôzne dĺžky, nižšiu vzájomnú zhodu a ďalšie vlastnosti, vytvárajú duplexy, ktoré nie sú miRNA. Teda väčšinu vytvorených dvojíc by mal klasifikátor označiť za negatívne. Výsledok je zobrazený v nasledujúcom grafe.



Obrázok 9.3: Výstup po klasifikovaní náhodne vytvorených sekvencií.

Predpoklad sa potvrdil, väčšina (90.34 %) vytvorených párov bola ohodnotená, tak, že to nie sú miRNA duplexy. Niekoľko párov bolo označených za miRNA duplexy. To môže byť spôsobené tým, že to buď sú miRNA duplexy, alebo vysokou podobnosťou s miRNA duplexami a niektoré mohli byť označené aj na základe o niečo nižšej ako 100 percentnej presnosti klasifikátora.

9.4 Implementovaná aplikácia

Navrhnutá a implementovaná aplikácia sa spúšťa tradičným spôsobom v prostredí bash, teda „./aplikacia.sh“. Aplikáciu je možné spustiť s nasledujúcimi parametrami:

- „./aplikacia.sh -h“: Vypísanie nápovedy.
- „./aplikacia.sh -file vstupný_súbor“: Vstupný súbor je spracovaný podľa krokov, ktoré boli opísané v minulých kapitolách. Teda jednotlivým sekvenciám sú vytvorené reverzné sekvencie, súbor je spojený s referenčnými miRNA sekvenciami a nasleduje Zhlukovanie pomocou CD-HIT. Vybraný kandidáti z hlukov sú určení ako miRNA sekvencie alebo ako vstup pre SVM klasifikátor. Ak sú určené pre SVM klasifikátor, sú upravené do vhodnej podoby, škálované a následne klasifikované. Výstupom je súbor so zoznamom sekvencií, ktoré boli určené ako miRNA sekvenie na základe zhlukovania a druhý súbor, v ktorom sú duplexy, ktoré boli pomocou SVM klasifikované ako miRNA duplexy.
- „./aplikacia.sh -seq sekvencia_1 sekvencia_2“: Tieto sekvencie sa spracujú do vhodnej formy pre SVM, následne sú škálované a klasifikované. Vynecháva sa teda zhlukovanie pomocou CD-HIT, pretože nie je nutné znižovať počet vstupných sekvencií. Výstupom je informácia o tom, či sú tieto sekvencie klasifikované ako miRNA alebo nie.
- „./aplikacia.sh ... -r 0/1“: Vymazanie pomocných súborov, ktoré vznikli počas práce aplikácie, ak je -r 1. Prednastavená hodnota je 0, teda ponechanie týchto súborov.
- „./aplikacia.sh -filesvm vstupný_súbor“: Ak je zadaný tento parameter, dáta vo vstupnom súbore sa nebudú zhlukovať, ale budú priamo klasifikované pomocou SVM. Vo vstupnom súbore ale musia byť vždy *mature* a *star* sekvencie duplexu na jednom riadku.

Jednotlivé kroky algoritmu sú rozdelené do niekoľkých skriptov, ktoré sú napísané v jazyku Python. Externé nástroje sú použité na zhlukovanie (CD–HIT), SVM klasifikáciu (LIBSVM), zistenie minimálnej energie duplexu (RNAfold) a zarovnanie *mature* a *star* sekvencií v duplexoch (Bio.pairwise2). Podrobný návod na spustenie aplikácie je uvedený v manuáli (README.txt).

9.4.1 Výstup

V prípade, že sú na vstupe aplikácie iba sve sekvencie, výstupom je informácia o tom ako tento duplex klasifikoval SVM systém. Teda či to pravdepodobne je (výstup je „1“) alebo nie je (výstup je „-1“) miRNA duplex.

Rovnaký výstup poskytne aplikácia spustená s parametrom „-filesvm“ pre vstupný súbor, v ktorom je zoznam duplexov. Výstupom je teda vstupný súbor doplnený v každom riadku o hodnotu 1, ak bol duplex klasifikovaný ako miRNA duplex, alebo o hodnotu -1 v opačnom prípade.

V prípade, že je na vstupe vstupný súbor s neznámymi sekvenciami a nebol určený parameter „-filesvm“, výstup je rozdelený na 2 súbory:

- Prvý súbor obsahuje zoznam sekvencií, ktoré boli pomocou zhlukovania určené ako miRNA sekvencie. Pridaná k nim je aj pravdepodobná rodina, do ktorej patria (na základe podobnosti s referenčnými miRNA sekvenciami).

- Druhý súbor obsahuje zoznam duplexov, ktoré boli vytvorené zo určených sekvencií v jednotlivých zhlukoch a zároveň boli určené ako miRNA duplexy pomocou SVM klasifikácie.

V poslednom prípade teda výstup obsahuje iba pravdepodobné miRNA sekvencie a pravdepodobné miRNA duplexy. Informácia o tom, ktoré z duplexov neboli klasifikovaná ako miRNA nie je uvedená.

Kapitola 10

Záver

V práci bol navrhnutý a implementovaný nástroj, ktorý slúži na klasifikáciu malých nekódujúcich RNA (konkrétne microRNA) rastlín a to bez analýzy ich vlastností v genóme. Využité boli vlastnosti *mature* a *star* sekvencií v duplexe microRNA. Na klasifikáciu, ale hlavne na zmenšenie dátovej sady, bol použitý zhlukovací nástroj CD–HIT, ktorý určí niektoré sekvencie ako microRNA na základe ich príslušnosti k zhlukom s referenčnými microRNA. Z ostatných zhlukov je určených niekoľko sekvencií, ktoré vytvoria možný duplex a tie sú vstupom *support vector machine* (SVM), ktorý určí, či ide skutočne o microRNA duplex.

V práci boli opísané malé nekódujúce RNA (miRNA, siRNA), ich štruktúra, vznik a funkcia v organizme. Ďalej bola opísaná metóda ich získania (next–generation sekvenovanie) a analýza algoritmov na ich zhlukovanie.

Bola vykonaná analýza konkrétnych zhlukovacích algoritmov (CD–HIT, DNACLUST, SEED, Uclust). Bol vykonaný experiment a analýza jeho výsledkov, ktoré slúžili na výber konkrétneho programu, ktorý bol použitý počas samotnej klasifikácie (CD–HIT).

Nasledovala analýza hypotézy, založenej na vysokej komplementarite *mature* a *star* sekvencií duplexu. Na klasifikáciu malo stačiť iba zhlukovanie, ktoré by prislúchajúce sekvencie rozdelilo do rovnakých zhlukov. Hypotéza po otestovaní navrhnutých metód nebola potvrdená. Preto bola navrhnutá výsledná metóda, v ktorej je zhlukovanie použité čiastočne na klasifikáciu, ale hlavne na zmenšenie dátovej sady, ktorá je vstupom pre SVM model.

Ďalej boli analyzované vlastnosti referenčných microRNA duplexov, postup ich využitia pri tvorbe a trénovaní SVM modelu. Tiež návrh a implementácia samotného nástroja na klasifikáciu vstupných sekvencií. Vlastnosti charakterizujúce výkonnosť klasifikácie boli získané pomocou metódy *10–fold cross–validation* a všetky dosahujú viac ako 96 percentné hodnoty na testovacích dátach. Klasifikácia prebehla aj na náhodne kombinovaných sekvenciách a potvrdila predpoklad o jej výsledku.

Medzi vylepšenia nástroja by mohlo patriť rozšírenie funkcionality o klasifikáciu živočíšnych microRNA. Teda implementovaný nástroj by bol schopný klasifikovať nielen rastlinné, ale aj živočíšne microRNA. Tiež by bolo možné rozšíriť funkcionalitu o klasifikáciu siRNA.

Možné je i doplnenie grafického používateľského rozhrania, alebo implementovanie vlastných nástrojov, namiesto používania externých nástrojov (napríklad na získanie voľnej energie duplexov). Ďalším možným zlepšením je rozšírenie na operačný systém Windows.

Dôležitým vylepšením by mohlo byť spresnenie klasifikácie neznámych dát, teda zlepšenie výberu negatívneho datasetu do trénovacích dát SVM modelu, alebo doplnenie nových vlastností duplexov, ktoré sú použité pri SVM klasifikácii.

Literatúra

[1] *CD-HIT User's Guide*. 2013, [Online; cit. 2015-01-11].
URL <http://weizhongli-lab.org/cd-hit/wiki/doku.php?id=cd-hit_user_guide#cd-hit-est>

[2] Sequence Alignment/Map Format Specication. December 2014: str. 17.

[3] Alberts, B.: *Základy buněčné biologie*. Espero, 2005, ISBN 8090290620, 740 s.

[4] Allison C Mallory, H. V.: Functions of microRNAs and related small RNAs in plants. *Nature Genetics*, , č. 38, June 2006: str. 7.

[5] Andritsos, P.: Data Clustering Techniques. 2002: str. 34.

[6] Ayman Grada, K. W.: Next-Generation Sequencing: Methodology and Application. *Journal of Investigative Dermatology*, 2013: str. 4.

[7] Bartel, D. P.: MicroRNA Target Recognition and Regulatory Functions. *NIHPA Author Manuscripts*, , č. 38, October 2013: str. 33.

[8] Chih-Wei Hsu, C.-J. L., Chih-Chung Chang: A Practical Guide to Support Vector Classifcation. 2010: str. 16.

[9] Dan Tulpan, S. L., Mirela Andronescu: Free energy estimation of short DNA duplex hybridizations. *BMC Bioinformatics*, , č. 11, February 2010: str. 22.

[10] Daniel Mapleson, T. D., Simon Moxon: MirPlex: A Tool for Identifying miRNAs in High–Throughput sRNA Datasets Without a Genome. *Proceedings of Experimental Zoology*, , č. 320:B, 2013: s. 47–56.

[11] D.J. Lipman, W. P.: Rapid and sensitive protein similarity searches. *Science*, , č. 227, March 1985: str. 7.

[12] Edgar, R. C.: Search and clustering orders of magnitude faster than BLAST. In *Bioinformatics*, 19, August 2010, str. 2.

[13] Ergude Bao, I. K., Tao Jiang: SEED: efficient clustering of next-generation sequences. In *Bioinformatics*, 18, August 2011, str. 8.

[14] Florkowski, C. M.: Sensitivity, Specificity, Receiver-Operating Characteristic (ROC) Curves and Likelihood Ratios: Communicating the Performance of Diagnostic Test. *The Clinical Biochemist Reviews*, , č. 29, August 2008: str. 5.

[15] Friedländer M.R., A. C., Chen W.: Discovering microRNAs from deep sequencing data using miRDeep. *Nature Biotechnology*, , č. 26, 2008: s. 407–415.

[16] Kohavi, R.: A Study of Cross–Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference on Artificial Intelligence*, 1995: str. 7.

[17] Lindenbaum, P.: Next Generation Sequencing File Formats.

[18] Lloyd, S. P.: Least squares quantization in PCM. *IEEE Transactions on Information Theory*, , č. 28, March 1982: s. 129–137.

[19] Marina Sokolova, G. L.: A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, , č. 45, May 2009: s. 427–437.

[20] Mohammadreza Ghodsi, M. P., Bo Liu: DNACLUST: accurate and efficient clustering of phylogenetic marker genes. , č. 271, 2011: str. 11.

[21] Mohorianu, I.; Stocks, M. B.; Wood, J.; aj.: CoLIde: A bioinformatics tool for CO-expression based small RNA Loci Identification using high-throughput sequencing data. *RNA biology*, ročník 10, č. 7, June 2013, ISSN 1555-8584. URL http://www.ncbi.nlm.nih.gov/pubmed/23851377

[22] Olena Morozova, M. A. M.: Applications of next-generation sequencing technologies in functional genomics. *Genomics*, , č. 92, 2008: s. 255–264.

[23] Peter J. A. Cock, N. G., Christopher J. Fields: The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research*, , č. 38, December 2009: str. 5.

[24] Roy Ronen, S. M., Ido Gan: miRNAkey: a software for microRNA deep sequencing analysis. *Bioinformatics*, , č. 26, 2010: s. 2615–2616.

[25] Srivastava, T.: Getting your clustering right (Part I). 2013, [Online; cit. 2015-01-11]. URL <http://www.analyticsvidhya.com/blog/2013/11/getting-clustering-right/>

[26] Weizhong Li, A. G.: Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. In *Bioinformatics*, 22, Burnham Institute for Medical Research, 2006, str. 2.

[27] Weizhong Li, B. N., Limin Fu: Ultrafast clustering algorithms for metagenomic sequence analysis. In *Briefings in Bioinformatics*, ročník 6, Oxford Journals, 2012, str. 13.

[28] Wheeler, B. M.: *Automating the Annotation and Discovery of MicroRNA in Multi-species High-throughput 454 Sequencing*. Diplomová práce, Graduate Faculty of North Carolina State University, 2008.

[29] Wikipedia: Central dogma of molecular biology – Wikipedia, The Free Encyclopedia. 2015, [Online; cit. 2015-01-11]. URL <http://en.wikipedia.org/wiki/Central_dogma_of_molecular_biology>

[30] Wikipedia: microRNA – Wikipedia, The Free Encyclopedia. 2015, [Online; cit. 2015-01-11]. URL <http://en.wikipedia.org/wiki/MicroRNA>

[31] Wikipedia: Small interfering RNA – Wikipedia, The Free Encyclopedia. 2015, [Online; cit. 2015-01-11].
URL <http://en.wikipedia.org/wiki/Small_interfering_RNA>

[32] Wikipedia: Support vector machine – Wikipedia, The Free Encyclopedia. 2015, [Online; cit. 2015-01-11].
URL <http://en.wikipedia.org/wiki/Support_vector_machine>

Príloha A

Obsah CD

- Technická správa vo formáte PDF.
- Zdrojové texty technickej správy.
- Implementovaný nástroj.
- Pomocné nástroje a moduly.
- Ukážkové testovacie dáta.
- Manuál.