



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

DIPLOMOVÁ PRÁCE

Bc. Rastislav Galváneš

Predikce terciární struktury RNA s využitím více vzorů

Katedra teoretické informatiky a matematické logiky

Vedoucí diplomové práce: RNDr. David Hoksza, Ph.D

Studijní program: Informatika

Studijní obor: IUI

Praha 2019

Prohlašuji, že jsem tuto diplomovou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V Praze dne

Podpis autora

Poděkování.

Název práce: Predikce terciární struktury RNA s využitím více vzorů

Autor: Bc. Rastislav Galvánek

Katedra teoretické informatiky a matematické logiky: Katedra teoretické informatiky a matematické logiky

Vedoucí diplomové práce: RNDr. David Hoksza, Ph.D, Katedra softwarového inženýrství

Abstrakt: Abstrakt.

Klíčová slova: klíčová slova

Title: RNA tertiary structure prediction using multiple templates

Author: Bc. Rastislav Galvánek

Department of Theoretical Computer Science and Mathematical Logic: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: RNDr. David Hoksza, Ph.D, Department of Software Engineering

Abstract: Abstract.

Keywords: key words

Obsah

Úvod	3
1 RNA štruktúra	4
1.1 RNA	4
1.2 Druhy a funkcie RNA	4
1.3 Reprezentácia a práca s RNA	5
1.4 Význam a získavanie terciárnej štruktúry	7
2 Metódy výpočetnej predikcie	9
2.1 Ab initio predikcia	9
2.2 Knowledge based de novo predikcia	10
2.3 Alignment sekvencií	11
2.4 Homológne modelovanie	13
2.5 Prehľad existujúcich nástrojov	13
2.6 ModeRNA	13
2.7 FARFAR	15
3 Algoritmus z bakalárskej práce	17
3.1 Používané typy súborov	17
3.2 Kostra algoritmu	17
3.3 Popis algoritmu	18
3.4 Popis implementácie	19
3.5 Hlavné problémy algoritmu a jeho implementácie	22
4 Automatizácia predikcie a porovnanie s ModeRNA	23
4.1 Automatické vyhľadanie template štruktúry	23
4.2 Automatizácia predikcie	24
4.3 Porovnanie s ModeRNA	25
4.4 Porovnanie predikcie dlhých štruktúr s ModeRNA	26
4.5 Úprava vstupných dát	26
5 Predikcia sekundárnej štruktúry	28
5.1 Príprava dát	28
5.2 Predikcia sekundárnej štruktúry	28
5.3 Integrácia do existujúceho algoritmu	28
5.4 Experiment	28
5.5 Výsledky	28
6 Použitie viacerých template štruktúr pri predikcii	29
6.1 Varianty a prístupy	29
6.2 Výber vhodných štruktúr	29
6.3 Algoritmus	29
6.4 Integrácia do existujúceho algoritmu	29
6.5 Experiment	29
6.6 Výsledky	29

Závěr	30
Seznam použité literatury	31
Seznam obrázků	33
Seznam tabulek	34
Seznam použitých zkratek	35
A Přílohy	36
A.1 První příloha	36

Úvod

Následuje několik ukázkových kapitol, které doporučují, jak by se měla diplomová práce sázet. Primárně popisují použití T_EXové šablony, ale obecné rady poslouží dobře i uživatelům jiných systémů.

1. RNA štruktúra

V tejto práci sa venujeme automatickej predikcii priestorovej štruktúry ribonukleovej kyseliny, preto sa budeme v prvej kapitole venovať jej významu z biologického hľadiska. Ďalej preberieme možnosti, akým spôsobom a aké podrobné informácie o štruktúre RNA dokáže súčasná veda získať experimentálnym spôsobom a aká je motivácia pre počítačovú predikciu RNA. Nakoniec uvedieme možnosti, ako RNA štruktúru reprezentovať vo formáte textových súborov a aké informácie o štruktúre jednotlivé typy súborov uchovávajú.

1.1 RNA

Ribonukleová kyselina slúži na prenos alebo uchovávanie genetickej informácie vo všetkých živých organizmoch. Najznámejšia je jej úloha v Centrálnnej dogme molekulárnej biológie Crick (1970), kde slúži pri syntéze proteínov z DNA na prenášanie genetickej informácie.

RNA je tiež, ako DNA tvorená štyrmi typmi nukleotidov (báz). Sú to adenín (A), guanín (G), cytozín (C) a uracyl (U). Na rozdiel od DNA sa v DNA namiesto uracylu vyskytuje báza tymín (T). Jednotlivé nukleotidy sú chemicky naviazané na cukre - ribóze, ktorý ich spája do vlákna (v prípade DNA sa jedná o deoxiribózu). Dĺžka vlákna môže byť v závislosti od typu RNA od jednotiek až po tisíce nukleotidov. Pre DNA následne platí, že sa vodíkovými väzbami spájajú dve komplementárne vlákna do špirály, čo čiastočne určuje pravidelný tvar molekuly v priestore. RNA sa však vyskytuje hlavne v jednovláknovej forme, pričom sa vlákno spája vodíkovými väzbami samo so sebou, čo prináša veľkú variabilitu v tvare molekuly. Platí, že vodíkovými väzbami sa navzájom viažu nukleotidy cytozínu a guanínu, adenínu a uracylu, guanínom a uracylom.

1.2 Druhy a funkcie RNA

Okrem prenosu genetickej informácie pri syntéze proteínov, ktorý pozostáva z replikácie DNA, transkripcie DNA do RNA a nakoniec translácie z RNA do proteínu, pričom sa využívajú rôzne typy RNA, zastáva RNA aj iné funkcie. Slúži napríklad na uchovávanie genetickej informácie niektorých jednoduchých organizmov ako sú vírusy. Tie môžu na uchovanie genetickej informácie používať jednovláknovú RNA, dvojitú RNA a v prípade retrovírusov špeciálny typ RNA, ktorý je schopný prepisovať genetickú informáciu z RNA do DNA procesom reverznej transkriptázy a vložiť tak svoju genetickú informáciu do génu napadnutej bunky. Medzi tieto vírusy patrí napríklad známy vírus HIV. Krupovic a kol. (2018)

Prehľad niektorých typov RNA:

- kódujúca (2%)
 - mediátorová RNA (mRNA)
- nekódujúca (98%)

- ribozomálna RNA (rRNA)
- prenosová RNA (tRNA)
- funkcionálna RNA (fRNA)
- mikro RNA (miRNA)
- malá interferujúca (small interfering) RNA (siRNA)
- jadrová (nuclear) RNA (snRNA)
- jadriková (nucleolar) RNA (snoRNA)
- vírusová RNA (vRNA)
- dlhá nekódujúca RNA (lncRNA)
- ďalšie...

Messenger RNA (mRNA) vzniká pri prepise (transkripcii) DNA v jadre bunky. Najprv je vytvorená pre-mRNA, ktorá obsahuje aj nekódujúce úseky. V ďalšom kroku sa z nej ešte v jadre bunky pomocou procesu nazývaného splicing, odstránia exóny (nekódujúce úseky) za pomoci snRNA, ktorá rozpoznáva sekvenciu báz AGGU označujúcu prechod medzi intrónom a exónom. Následne mRNA putuje von z jadra jeho pórami do cytoplazmy, kde sa naviaže na ribozóm. Alberts B (2002)

Ribozóm obsahuje ribozomálnu RNA, ktorá sa zúčastňuje translácie (prekladu) mRNA na kódovanú bielkovinu. Okrem toho je to typicky najčastejšie sa vyskytujúca RNA v bunke, pričom jej dĺžka môže byť niekoľko tisíc nukleotidov.

Prenosová tRNA sa nachádza v cytoplazme bunky a jej funkcia spočíva v dopravení správnej aminokyseliny do procesu translácie.

Niektoré typy RNA plnia regulačnú funkciu. Napríklad mikro RNA (miRNA) zabráňuje procesu translácie mRNA tým, že sa na ňu naviaže a zabráni jej spojeniu s ribozómom. snoRNA zas hrá úlohu pri modi

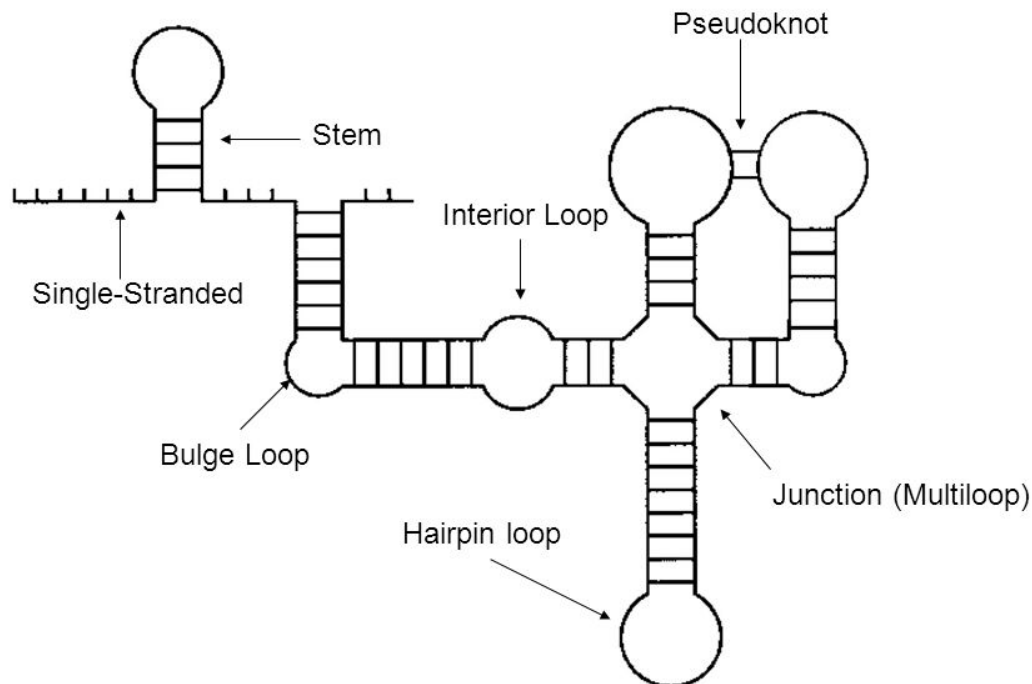
kácii ostatných typov RNA - hlavne rRNA, tRNA a snRNA.

Sekvence lncRNA mávajú typicky dĺžku okolo 200 nukleotidov a jeden zo známych zástupcov je gén XIST, ktorý sa uplatňuje pri procese inaktivácie chromozómu X. ??

1.3 Reprezentácia a práca s RNA

Fyzicky je RNA v bunke vlastne len mnoho atómov vodíka, kyslíka, uhlíka, fosforu usporiadaných v priestore vďaka chemickým a fyzikálnym väzbám a vlastnostiam atómov. Pre to, aby sme ich mohli spracovávať pomocou počítača potrebujeme vhodnú reprezentáciu štruktúry. Typ reprezentácie závisí od toho, aké informácie o štruktúre chceme mať k dispozícii a takisto aké informácie sme schopní získať. Principiálne môžeme rozdeliť reprezentáciu RNA štruktúr na nasledujúce štyri úrovne:

- Primárna
- Sekundárna
- Terciárna



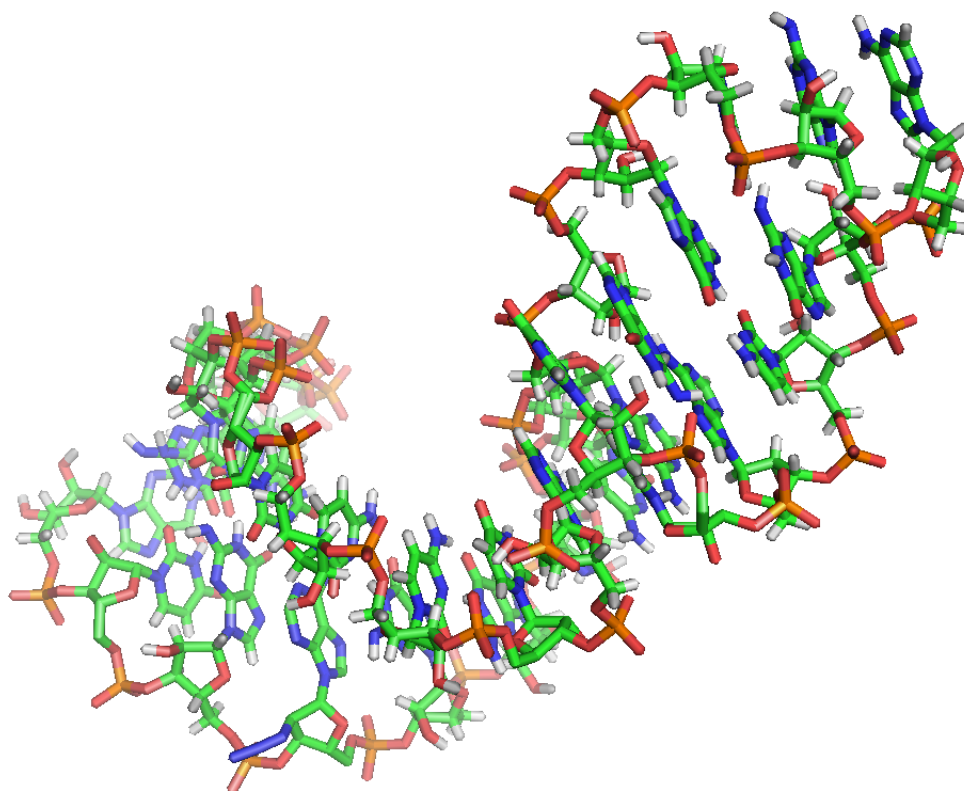
Obrázek 1.1: Príklad nakreslenia sekundárnej štruktúry RNA. Eddy (2004)

- Kvarciárna

Primárna štruktúra je najviac zjednodušená reprezentácia RNA. Určuje len poradie a typ jednotlivých nukleotidov v RNA štruktúre a ďalej ju budeme v práci tiež označovať ako sekvenciu. V počítači ju reprezentujeme ako textový súbor typu fasta, ktorý má na prvom riadku identifikáciu štruktúry (názov, chain) a v ďalších riadkoch sú to len písmena A, G, C, U určujúce presné poradie nukleotidov. V prípade, že typ nukleotidu na niektorej pozícii je neznámy používame písmeno X alebo N. Primárna sekvencia RNA takisto slúži ako jeden zo vstupov pre náš prediktor a definuje sekvenciu štruktúry, ktorú cheme získať.

Sekundárna štruktúra zachytáva vodíkové väzby medzi jednotlivými nukleotidmi vo vlákne RNA. Dva nukleoidy, ktoré sú spojené vodíkovou väzbou označujeme ako base pair. Vďaka spájaniu jednotlivých nukleotidov vieme v sekundárnej štruktúre pozorovať rôzne podštruktúry, ako napríklad helix, loop, pseudoknot, hairpin loop, internal loop, branch loop, stem a ďalšie 1.1. Sekundárnu štruktúru budeme v tejto práci používať na pomoc pri predikcii terciárnej štruktúry, nakoľko nám dáva informáciu o nukleotidoch, ktoré sú spojené vodíkovou väzbou a teda sa nachádzajú blízko pri sebe. Sekundárnu štruktúru molekuly budeme reprezentovať ako textový súbor, kde bodka značí, že na nukleotid danej pozícii nie je spárovaný žiadnou väzbou, base pair spojený vodíkovou väzbou je značený ako validne uzátvorkovanie jednoduchými zátvorkami a pseudoknot býva reprezentovaný hranatými zátvorkami.

Zmyslom terciárnej štruktúry je popísať presné rozloženie jednotlivých atómov v trojdimenzionálnom priestore za pomoci koordinátov. V našej práci je hlavným cieľom tieto koordináty určiť za predpokladu znalosti primárnej sekvencie a terciárnych štruktúr ďalších RNA molekúl, ktoré sa pokúšame pri predikcii použiť ako vzory.



Obrázek 1.2: Príklad terciárnej štruktúry RNA nachádzajúcej sa v baktérii *escherichia coli*.

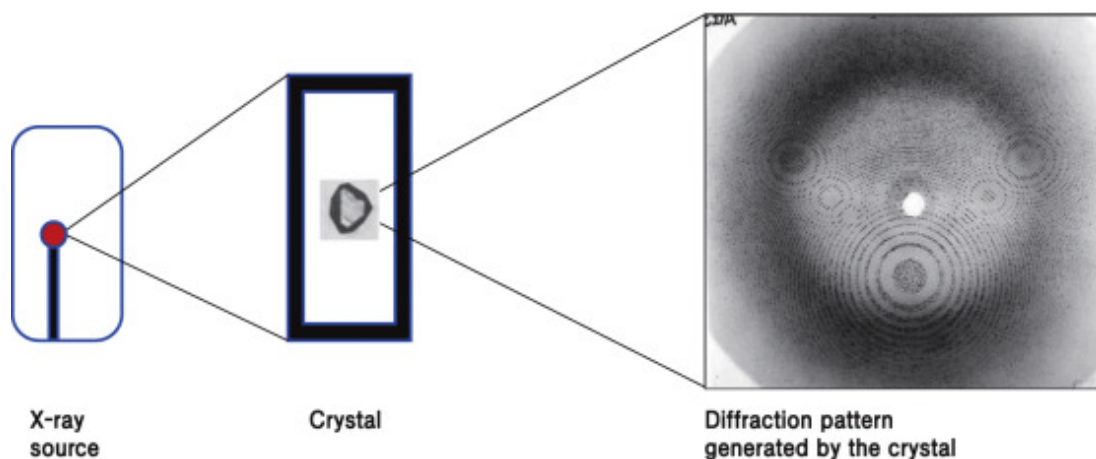
Kvarciárna štruktúra RNA popisuje vzťahy medzi celými molekulami RNA - napríklad interakcie medzi jednotlivými molekulami RNA v ribozónoch Noller (1984) a taktiež vzťahy medzi RNA a molekulami bielkovín.

1.4 Význam a získavanie terciárnej štruktúry

Štruktúra RNA priamo súvisí s funkciou, ktorú vykonáva. Tvar štruktúry 1.2 určuje, ktoré enzýmy sa na ňu dokážu pripájať a prípadne ju modifikovať a s ktorými bielkovinami a nukleovými kyselinami sa dokáže viazať. Bolo preukázané, že väčšina časti RNA štruktúry, ktoré sú schopné sa viazať s inými molekulami nie sú súčasťou žiadneho base pair-u a teda sú v sekundárnej štruktúre označené ako nespárované Schudoma C. (2010). Zmena terciárnej štruktúry, ktorá vedie ku strate pôvodnej funkcie molekuly, sa nazýva denaturácia.

Bolo vyvinutých viacero experimentálnych metód, pomocou ktorých môžeme získať terciárnu štruktúru RNA Felden (2007):

- metódy s vysokou presnosťou
 - X-ray crystallography
 - Cryo-electron microscopy
 - Nuclear Magnetic Resonance (NMR) spectroscopy
- metódy s nižšou presnosťou



Obrázek 1.3: Princíp rentgenovej kryštalografie. Ryu (2017)

- Mass spectrometry
- Chemical probing
- Thermal denaturation
- RNA engineering

X-ray crystallography (rentgenova kryštalografia) funguje principiálne tak, že sa molekula najprv zkrýštalizuje a následne sa nasvieti rentgenovým lúčom. Z kryštálu je lúč odrazený a pritom rozdelený na viacero lúčov. Zmeraním týchto uhlov intenzity odrazených lúčov je následne možné určiť pozície jednotlivých atómov v molekule. Momentálne je to jasná z najpoužívanejších metód na získanie mnohých makromolekulárnych štruktúr. Rozlíšenie získanej štruktúry sa pohybuje okolo 2.0 Å. 1.3

Cryo-electron microscopy metóda využíva zmrazenie molekuly v substancii, ktorá je následne pozorovaná elektrónovým mikroskopom. Princíp metódy je známy približne od roku 1970, ale až donedávna pomocou nej nebolo možné získať tak presné výsledky, ako pomocou rentgenovej kryštalografie. Na druhej strane dĺžka skúmanej štruktúry nie je pri tejto metóde tak limitujúcim faktorom. V roku 2017 bola udelená nobelová cena za chémiu J. Dubochetovi, J. Frankovi a R. Hendersonovi za vyvinutie metódy, ktorou sa dá získať atómová štruktúra molekuly s vysokým rozlíšením.

Metóda Nuclear Magnetic Resonance je založená na pôsobení statického magnetického poľa na jadrá atómov v molekule. Je vhodná hlavne na získavanie kratších štruktúr.

Experimentálne prístupy sa od seba navzájom líšia presnosťou výsledku, dĺžkou štruktúry s ktorou sú schopné pracovať, ale ich hlavným problémom je, že sú stále časovo náročné a drahé. Pretože získanie primárnej štruktúry RNA a proteínov je oveľa ľahšia úloha, začali byť skúmané aj možnosti, ako predikovať sekundárnu a terciárnu štruktúru za pomoci počítača, čomu sa budeme v našej práci venovať.

2. Metódy výpočetnej predikcie

Cieľom výpočetnej predikcie RNA štruktúry je dokázať algoritmicky modelovať terciárnu alebo sekundárnu štruktúru zo znalosti primárnej sekvencie RNA molekuly. Pri takejto predikcii je dôležité, aby sme dostali čo najpresnejší výsledok v porovnaní s experimentálnymi metódami, ale aby výpočové nároky a čas boli výrazne nižšie, ako v prípade experimentálnej rezolúcie štruktúry. Aby malo zmysel sa pokúšať o predikciu štruktúry zo sekvencie, potrebujeme vedieť, že terciárna a teda aj sekundárna štruktúra je do veľkej miery jednoznačne určená štruktúrou primárnou.

Túto otázku môžeme zodpovedať vďaka znalostiam zo skladania (foldingu) proteínov, ktorých výskumu sa venovalo viacej úsilia. Platí, že skladanie bielkovín a RNA prebieha veľmi podobne a preto poznatky o štruktúrach a sekvenciách bielkovín môžeme použiť aj pri RNA. Moore (1999)

Existujú dve hlavné pozorovania, ktoré nám umožňujú štruktúry makromolekúl modelovať Jenny Gu (2009):

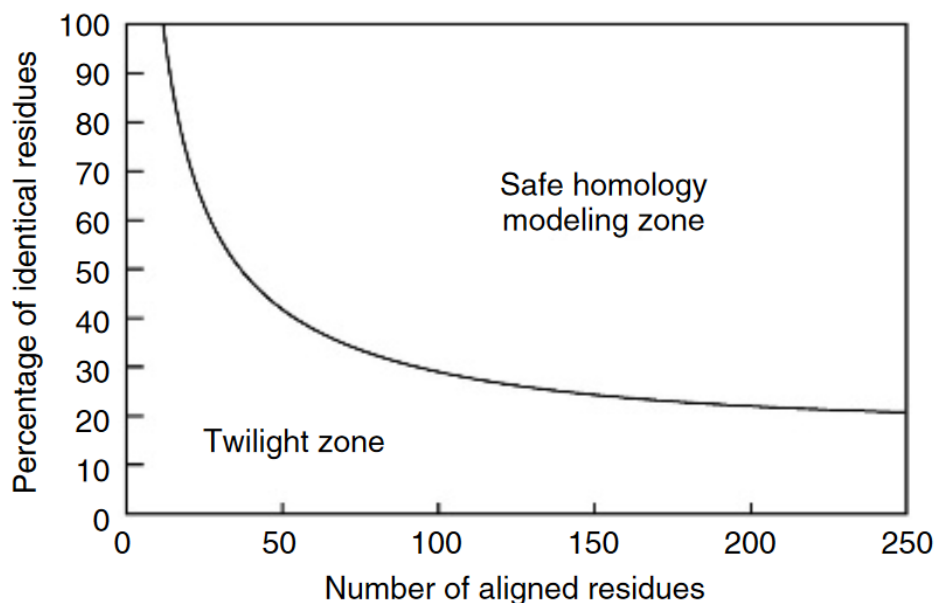
- Štruktúra proteínu je unikátne určená sekvenciou aminokyselín.
- Štruktúra sa zachováva aj pri určitých zmenách v sekvenciách a teda platí, že napriek odlišnosti v sekvenciách sú štruktúry veľmi podobné. Je to spôsobené tým, že počas evolúcie štruktúra stále plní podobnú úlohu a preto sa jej tvar ostáva nezmenený aj napriek mutáciám sekvencie. Vďaka rozširujúcej sa databázi makromolekúl (Protein Data Bank) boli získané vzťahy, ako podobné si musia byť rovnako dlhé sekvencie, aby sme mohli predpokladať, že aj ich štruktúry sú podobné. Tento vzťah zobrazuje obrázok 2.1.

2.1 Ab initio predikcia

Pri ab initio predikcii štruktúry vychádzame iba z primárnej sekvencie a chemicko-fyzikálnych vlastností, vďaka ktorému sa v reálnom svete štruktúra skladá do stabilného tvaru. Algoritmus postupne vytvára kandidátske štruktúry tak, že sa snaží sa minimalizovať funkciu predstavujúcu voľnú energiu (energia, ktorá je ľahko dostupná v systéme). Následne z takto vygenerovaných kandidátov musí vybrať najprirodzenejšiu štruktúru. Najväčším problémom tohoto prístupu je mnoho lokálnych miním vo funkciách predstavujúcej voľnú energiu a preto aj výpočetná zložitosť.

Tieto komplikácie sa dajú čiastočne riešiť viacerými spôsobmi. Jedna cesta je zvýšiť výpočetný výkon - použitie superpočítača, alebo distribuovať výpočet na mnoho výpočetných staníc. Ďalšia je pokus o zmenšenie vyhľadávacieho priestoru a efektívnejšie vyhľadávať kandidátske štruktúry. Jedna metóda je označovaná ako coarse-grained reprezentácie, kde nie sú reprezentované všetky atómy. Využívajú sa taktiež heuristické a pravdepodobnostné metódy na zmenšenie prehľadávaného priestoru.

Stále však platí, že takáto metóda je pre dlhšie štruktúry nepoužiteľná. Napriek tomu, že súčasný state-of-the art umožňuje predikovať štruktúry celkom



Obrázek 2.1: Vzťah dĺžky štruktúr a percentuálneho pomeru identických residuí v sekvenciách určujúce predpoklad, že štruktúry takýchto sekvencií sú podobné. Jenny Gu (2009)

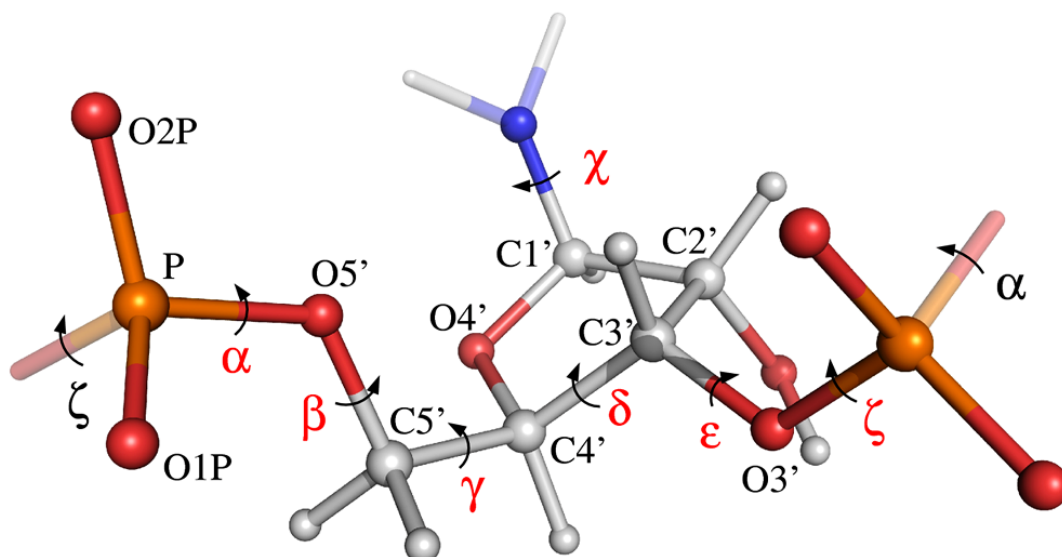
presne s rastúcou dĺžkou sekvencie neúmerne rastie výpočetná náročnosť. Ako príklad uvidíme pokus predikovať štruktúru dlhú 112 nukleotidov, pričom výsledná štruktúra sa líšila od experimentálne získanej len minimálne, výpočet však stál viac ako 100 000 hodín CPU. Qian a kol. (2007)

2.2 Knowledge based de novo predikcia

Princíp tohoto typu predikcie je veľmi podobný, ako ten v ab initio metóde ale namiesto samplovania možných usporiadaní atómov používa knižnicu krátkych úsekov štruktúry (väčšinou dĺžky 2-5 nukleotidov). Algoritmus následne vytvára kandidátske štruktúry tým, že kombinuje jednotlivé krátke úseky štruktúr z knižnice do kandidátskych štruktúr a takisto minimalizuje voľnú energiu modelu. Výhodou je hlavne zrýchlenie generovanie kandidátskych štruktúr oproti ab initio predikcii. Aj tak je však predikovanie dlhých štruktúr príliš pomalé. Mnohé nástroje preto umožňujú vložiť sekundárnu štruktúru predikovanej sekvencie a tak zmenšiť prehľadávaný priestor.

Ďalší spôsob, ako znížiť prehľadávaný priestor je použitie internej reprezentácie štruktúry. V prípade, že atómy reprezentujeme v súradnicami v trojdimenzionálnom priestore, ich síce vieme dobre zobrazit, ale takáto reprezentácia má $3 \times \text{počet atómov}$ stupňov voľnosti. Výhodnejšie je štruktúru reprezentovať napríklad pomocou reprezentácie uhlov medzi nukleotidmi. 2.2

Nástroj FARFAR Das R. (2010), ktorý používame v našej predikcii patrí tiež medzi knowledge based modelovacie metódy.



Obrázek 2.2: Reprezentácia RNA fragmentu pomocou siedmych uhlov. Frellsen a kol. (2009)

2.3 Alignment sekvencií

Zarovnanie dvoch sekvencií slúži na získanie informácie o tom, či su dané sekvencie nejako evolučne, štrukturálne, alebo funkčne príbuzné. Existuje viacero druhov algoritmov zarovnania - napríklad jednoduchý dot plot vhodný na jednoduchú vizualizáciu zarovnania, heuristické metódy ako FASTA a BLAST určené na čo najrýchlejšie porovnanie sekvencie s rozsiahlou databázou ďalších sekvencií, alebo metódy počítajúce najlepšie zarovnanie určené skórovacím systémom za pomoci dynamického programovania.

V tejto práci budeme využívať semiglobálne zarovnanie pomocou algoritmu Needleman–Wunsch Needleman S. B. (1970) implementovaného v programe EM-BOSS ?. Ako vstup algoritmus dostáva dve sekvencie dĺžiek m a n , ktoré chceme zarovnať a hodnoty parametrov gap open (penalizácia v skóre za otvorenie medzery v zarovnaní) a gap extend (penalizácia v skóre za predĺženie medzery v zarovnaní). Algoritmus následne za pomoci dynamického programovania 2.3 vypočíta zarovnanie s najnižším skóre v čase aj priestore $O(nm)$. Výstupom algoritmu sú zarovnané sekvencie a skóre zarovnania. V zarovnaní na určitej pozícii môžu nastať 3 prípady, a to zarovnanie dvoch rovnakých reziduí (match), zarovnanie dvoch odlišných reziduí (mismatch) a nakoniec zarovnanie rezidua na medzeru (gap) vloženú do druhej sekvencie. Nami používaná implementácia algoritmu nepenalizuje za medzery v zarovnaní nachádzajúce sa na začiatku, alebo na konci zarovnania, preto je možné ňou zmysluplne zarovnať krátku štruktúru na časť oveľa dlhšej štruktúry.

Okrem globálneho poznáme aj presné lokálne zarovnanie vyriešené algoritmom Smith–Waterman Smith T. F. (1981). Tento algoritmu pracuje taktiež na princípe dynamického programovania a vyhľadáva zarovnanie dvoch subsekvencií s najlepším skóre. Používa sa na nájdenie podobných regiónov medzi dvomi sekvenciami.

Needleman-Wunsch

match = 1

mismatch = -1

gap = -1

		G	C	A	T	G	C	U	
		0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5	
A	-2	0	0	1	0	-1	-2	-3	
T	-3	-1	-1	0	2	1	0	-1	
T	-4	-2	-2	-1	1	1	0	-1	
A	-5	-3	-3	-1	0	0	0	-1	
C	-6	-4	-2	-2	-1	-1	1	0	
A	-7	-5	-3	-1	-2	-2	0	0	

Obrázek 2.3: Needleman-Wunsch algorithm (2014) Wikipedia dostupné na https://en.wikipedia.org/wiki/Needleman-Wunsch_algorithm 27.05.2019. Příklad jednoho z troch nejlepších zarovnaní dvou sekvencí:

GCATG-CU

G-ATTACA

2.4 Homológne modelovanie

Tvrdenie zo začiatku kapitoly, ktoré hovorí, že štruktúra si zachováva podobný tvar aj napriek tomu, že jej sekvencia postupne mutuje, umožňuje zmysluplne predikovať štruktúru na základe vzoru.

Homológne modelovanie používa na modelovanie neznámej štruktúry zo sekvencie ešte jednu vzorovú sekvenciu (template), ktorej štruktúra je známa, teda získaná za pomoci nejakej experimentálnej metódy. Predikovanú štruktúru zvykne nazývať cieľ (target).

Prvým krokom je teda určenie vhodnej template štruktúry, pomocou ktorej budeme predikovať target štruktúru. Druhým krokom je globálne zarovnanie oboch sekvencií a získanie konzervovaných úsekov, teda úsekov v ktorých by mali byť obe štruktúry veľmi podobné. Konzervované úseky môžu byť po nejakých úpravách prenesené do cieľovej štruktúry. Z princípu vyplýva, že čím podobnejšie sekvencie budú mať target a template štruktúry, tým viac konzervovaných úsekov bude existovať a tým jednoduchšia a presnejšia by mala predikcia byť.

V treťom kroku musia byť dopredikované nekonzervované (chýbajúce úseky) cieľovej štruktúry. Existujú viacero prístupov. Jedným z nich je knižnica fragmentov, kde sa do chýbajúcej medzery v cieľovej štruktúre snažíme vhodne umiestniť fragment štruktúry z knižnice, ďalším je napríklad dopredikovanie medzery ab initio alebo de novo algoritmami.

Takto hotový model sa nakoniec môže optimalizovať použitím algoritmu na minimalizovanie voľnej energie, alebo sa riešia kolízie medzi jednotlivými nukleotidami.

Hlavnou výhodou homológneho modelovania je, že je možné ho použiť na dlhé štruktúry. Problémom môže byť vybrať správnu template štruktúru a dopredikovanie nekonzervovaných úsekov. 2.4

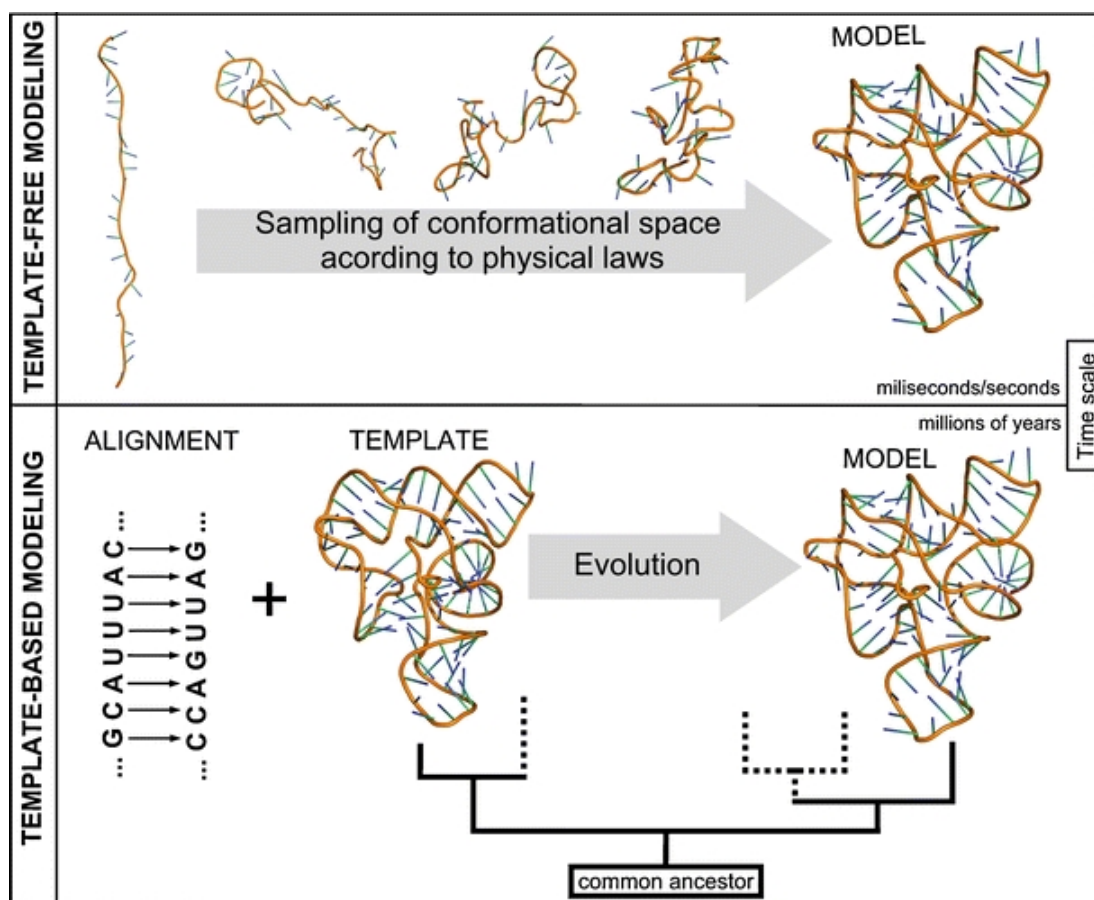
2.5 Prehľad existujúcich nástrojov

Vďaka tomu, že počet dostupných primárnych sekvencií stále rastie rýchlejšie, ako počet experimentálne zistených terciárnych štruktúr, vzniklo mnoho nástrojov na predikciu terciárnej štruktúry RNA. 2.1

2.6 ModeRNA

ModeRNA je implementácia algoritmu komparatívneho modelovania RNA s ktorým sme porovnávali nami vyvinutý algoritmus. Je dostupná ako ModeRNA server a ponúka službu kompletnej predikcie submitovanej sekvencie (nájdanie vhodného template, zarovnanie sekvencií a vytvorenie modelu terciárnej štruktúry). Okrem toho je možné stiahnuť jej zdrojové kódy (Python) a nainštalovať a používať ju lokálne pre automatické hromadné spracovanie.

Ako vstup ModeRNA požaduje zarovnanie template a target sekvencií spolu so súradnicami jednotlivých atómov template štruktúry. Dodané zarovnanie ModeRNA nijako nemodifikuje a od jeho kvality a zvoleného template závisí výsledná presnosť predikcie. Zjednodušený algoritmus, ktorým ModeRNA predikuje štruktúru Rother K (2011):



Obrázek 2.4: Porovnanie princípu de novo a template based predikcie Rother K (2011)

- Skopírovanie zarovnaných nukleotidov.
- Substitúcia nukleotidov, ktoré boli zarovnané na iný nukleotid.
- Modelovanie indelov vložení fragmentov štruktúr z knižnice obsahujúcej 131 316 fragmentov dĺžky 2-19 nukleotidov. ModeRNA najprv rýchlym filtrovaním podľa vzdialeností prekryvajúcich sa atómov fragmentu a template- u vyberie 50 najvhodnejších kandidátov, pokúsi sa ich vložiť do medzery a pre každého kandidáta spočíta skóre, pričom vyberie jediného s najlepším skóre a vloží ho do medzery.
- V prípade, že predikovaná štruktúra (backbone) nie je spojitá ModeRNA sa ju pokúsi opraviť.

2.7 FARFAR

FARFAR je algoritmus de novo predikcie RNA implementovaný spolu s ďalšími bioinformatickými algoritmi a nástrojmi v balíčku Rosetta. My ho používame na predikovanie krátkych nekonzervovaných úsekov v našom algoritme.

Nástroj je schopný predikovať štruktúru iba z primárnej sekvencie.

Algoritmus pracuje tak, že nedeterministicky generuje kandidátske štruktúry z ktorých vyberie tú s najnižšou voľnou energiou. Keďže sa jedná o algoritmus typu Monte Carlo dve rôzne spustenie algoritmu môžu generovať rôzne výsledky a platí, že čím viac kandidátskych štruktúr vygenerujeme, tým zvyšujeme šancu na vygenerovanie čo najlepšej štruktúry.

Z pohľadu výkonnosti platí, že čím viac nukleotidov predikujeme (včetně tých pevne daných) tým predikcia dlhšie trvá. Pre presnejšiu predstavu sme urobili porovnanie, pričom sme predikovali nekonzervovaný úsek dlhý 9 nukleotidov. Pri zahrnutí zvyšných 129 konzervovaných nukleotidov do predikcie trvalo vygenerovanie jednej štruktúry približne 13 minút. Pri totožných podmienkach a pa-

Názov	Info	Referencia
MacroMolecule Builder	komparatívny modeling RNA	Flores a kol. (2011)
ModeRNA	komparatívna predikcia s knižnicou databázových fragmentov na predikovanie medzier	Rother a kol. (2011)
SimRNA	corase-grained model s Monte Carlo samplingom štruktúr	Boniecki a kol. (2015)
FARFAR	knowledge based de novo prediktor knowledge-based automatizovaná	Das R. (2010)
RNAComposer	predikcia štruktúry RNA s využitím sekundárnej štruktúry	Biesiada a kol. (2016)
iFoldRNA	de novo predikcia RNA založená na corase-grained model	Sharma a kol. (2008)

Tabulka 2.1: Prehľad niektorých programov určených na predikciu RNA s informáciou o type použitého algoritmu.

rametroch s jedinou zmenou a to, že sme do predikcie vybrali len 41 okolitých konzervovaných nukleotidov trvala predikcia jednej kandidátskej štruktúry v priemere menej ako 6 minút.

Očakávaná presnosť predikcie je priamo úmerná dĺžke neznámeho predikovaného úseku. Autori algoritmu uvádzajú, že pri predikcii štruktúr dĺžky 6 až 13 nukleotidov je priemerná RMSD menšia ako 2 Å. Pri štruktúrach dlhých 13-23 nukleotidov predstavovala priemerná RMSD už 6,5 Å.

Je však takisto možné dať mu na vstup pdb súbor s koordinátami niektorých nukleotidov a zakázať mu tieto nukleotidy modifikovať. Takisto je možné mu dodať pdb súbor s nukleotidmi a dovoliť mu, aby ho algoritmus bral ako fixovaný kus štruktúry, ktorým môže ľubovoľne pohybovať oproti zvyšku štruktúry. Posledná pre nás využiteľná možnosť je dodať algoritmu sekundárnu štruktúru target molekuly. Touto štruktúrou sa potom algoritmus pri predikcii riadi a zmenšuje sa tak prehľadávaný priestor.

3. Algoritmus z bakalárskej práce

V tejto práci naväzujeme a ďalej vylepšujeme algoritmus, ktorý bol vytvorený v rámci bakalárskej práce. Preto v tejto kapitole uvedieme princípy fungovania a stav implementácie algoritmu tak, ako bol popísaný v bakalárskej práci. Jedná sa o algoritmus založený na princípe homológneho modelovania, čo znamená, že predikujeme terciárnu RNA štruktúru na základe primárnej sekvencie molekuly označovanej ako target a známej terciárnej štruktúry a sekvencie inej RNA molekuly označovanej ako template.

3.1 Používané typy súborov

V algoritme opakovane pracujeme s určitými typmi textových súborov. Sú to súbory s príponami fasta, secstr, pdb a aln.

Súbory typu fasta slúžia na ukladanie sekvencií. Pozostávajú z dvoch riadkov, v prvom je identifikátor sekvencie pozostávajúci z jej názvu a chain-u (jedna sekvencia máva často viacero chains) a v ďalších riadkoch sú za sebou zoradené jednotlivé nukleotidy A, C, G, U. V prípade, že je nejaký nukleotid v rade neznámy, bežne namiesto jeho typu úvádza písmeno N alebo X.

Súbory secstr nám slúžia na ukladanie informácií o sekundárnej štruktúre molekuly. Sú tvorené kombináciou rôznych typov zátvoriek, ktoré popisujú sekundárnu štruktúru tak, že medzi nukleotidmi odpovedajúcimi zátvorkám existuje chemická väzba - takéto dva nukleotisy sa tiež nazývajú base pair. To znamená, že sa v terciárnej štruktúre budú nachádzať blízko pri sebe. Bodka v sekundárnej štruktúre znamená, že nukleotid netvorí base pair so žiadnym ďalším nukleotidom. Rôzne typy zátvoriek ako `[]`, `{}`, `<>` reprezentujú pseudouzly.

Súbory typu pdb uchovávajú okrem iného informácie o jednotlivých atómoch molekuly. V každom riadku sú uložené informácie o presných koordinátoch atómu v 3D priestore, typ atómu v rámci nukleotidu, chain do ktorej atóm patrí a index nukleotidu, ktorému patrí. Pdb súbory často nie sú kompletne, chýbajú v nich atómy alebo celé nukleotidy. Takisto sa stáva, že indexy nukleotidov v pdb súboroch a fasta súboroch nesúhlasia.

Súbory s príponou aln označujú výstup zarovnania dvoch sekvencií z programu EMBOSS Needle. Súbor obsahuje presné zarovnanie sekvencií, skóre zarovnania, percentuálny pomer medzier v zarovnaní (gaps) a percentuálny pomer korektne zarovnaných nukleotidov označený ako similarity.

3.2 Kostra algoritmu

V nasledujúcom zozname uvádzame postupnosť hlavných krokov algoritmu.

1. Predpríprava a validácia vstupných súborov: template sekvencia, target sekvencia a štruktúra.
2. Alignemnt: Zarovnanie target a template sekvencií.
3. Sliding window: Algoritmus posuvného okienka na zarovnaní.

4. Treating indels: Vyríšenie medzier v zarovnaní.
5. Kopírovanie a mapovanie konzervovaných nukleotidov z target štruktúry do predikovanej template štruktúry.
6. Vyčlenenie predikcie príliš dlhých medzier v target štruktúre.
7. Príprava vstupu pre FARFAR.
8. Predikcia nekonzervovaných úsekov pomocou algoritmu FARFAR.
9. Zloženie predikovaných úsekov a dlhých medzier do finálnej štruktúry.

3.3 Popis algoritmu

Ako vstup algoritmus dostane target sekvenciu a template sekvenciu aj štruktúru. Na výstupe očakávame terciárnu štruktúru target molekuly RNA.

Ako prvý krok algoritmus skontroluje, či sú sekvencia vo fasta súbore a štruktúra v pdb súbore rovnako indexované. Nukleotidy v pdb súbore sú očíslované, ale vo fasta súbore číslo nukleotidu odpovedá pozícií nukleotidu v súbore. Kontrolujeme to prechodom cez pdb súbor tak, že indexom nukleotidu z pdb zaindexujeme do fasta súboru a typ nukleotidu musí byť v oboch súboroch na tejto pozícií zhodný. V prípade, že zhodný nie je skúsime ešte posunúť fasta sekvenciu pridaním dummy nukleotidov na začiatok sekvencie (pre prípad, že by začiatok sekvencie v súbore chýbal). Ak sa nám nepodarí ani takýmto spôsobom dosiahnuť aby sa typy nukleotidov v rovnakých indexoch zhodovali, označíme target za nevhodný pre predikciu a algoritmus končí neúspechom.

V druhom kroku urobíme globálne zarovnanie (alignment) target a template sekvencií v programe Emboss Needle. Na vytvorené zarovnanie použijeme algoritmus posuvného okienka (sliding window) a pre každú pozíciu určíme percentuálnu mieru okolitých úspešne zarovnaných nukleotidov spadajúcich do okienka. V prípade, že získaná hodnota je vyššia ako parametrom určená hranica, označíme príslušnú pozíciu v zarovnaní ako konzervovanú.

V treťom kroku sa zaoberáme medzerami (indels), ktoré vznikli v target alebo template sekvencií pri zarovnaní. Inak povedané, do oboch sekvencií algoritmus zarovnania mohol vložiť ľubovoľne medzery, tak aby získal zarovnanie s čo najlepším skóre, prípadne na seba mohol zarovnať nezhodujúce sa nukleotidy 3.1.

To znamená, že medzery v inak konzervovanom úseku template sekvencie by vo výsledku nenechali miesto na doplnenie nukleotidov z target sekvencie zarovnaných oproti týmto medzerám z template sekvencie. Naopak, medzery v target sekvencií zarovnané oproti nukleotidom v template sekvencií v inak konzervovanom úseku by mohli spôsobiť medzeru v predikovaní štruktúre, nakoľko by sme

Sekvencia	konzervované	nekonzervované	gap	gap
template	G	A	-	U
target	G	G	C	-

Tabulka 3.1: Prehľad štyroch situácií, ktoré môžu nastať na každej pozícií v zarovnaní dvoch sekvencií.

z fragmentu konzervovanej štruktúry len odmazali nejaké nukleotidy a ničím ich nedoplnili. 3.1

Oba tieto problémy riešime tak, že nukleotidy v určitom okolí takýchto úsekov označíme za nekonzervované a budú dopredikované algoritmom FARFAR. Taktiež označíme za nekonzervované tie nukleotidy, ktoré boli zarovnané na nezhodujúci sa typ nukleotidu.

V štvrtom kroku skopírujeme konzervované časti template štruktúry do predikovanej target štruktúry. Vzhľadom na to, že v zarovnaní môžu byť rôzne vložené medzery do target aj template sekvencie musíme premapovať indexy nukleotidov z template štruktúry tak, aby odpovedali nukleotidom, na ktorých miesta sú vložené v target sekvencií. Toto urobíme jednoducho vďaka informáciám zo zarovnania. Takto získame target štruktúru s medzerami, ktoré potrebujeme dopredikovať.

V piatom kroku identifikujeme dlhé nekonzervované úseky a vyčleníme ich následnu predikciu do samostatných behov algoritmu FARFAR. Prvý dôvod je, že takto sa môže FARFAR zamerať iba na predikciu dlhého úseku a tým znížime celkovú výpočetnú náročnosť. Takisto môžeme zmeniť jeho parametry, ako napríklad zvýšiť počet sámplovaných modelov, prípadne zvýšiť celkový čas predikcie. Ďalší dôvod prečo dopredikovanie nekonzervovaných úsekov takto delíme je, že algoritmus FARFAR sa nedokáže dobre vysporiadať s predikciami príliš dlhých štruktúri, aj keď je časť nukleotidov pevne daná. Z toho dôvodu rozdeľujeme dlhé štruktúry na úseky dĺžky 300 nukleotidov na základe ich poradia v sekvencii. Takéto delenie spôsobuje ďalší problém a to, že nukleotidy, ktoré sú od seba vzdialené v sekvencií môžu byť blízko pri sebe v terciárnej štruktúre. Naše riešenie teda vybere tieto dlhé nekonzervované úseky spolu s okolitými nukleotidmi, ktoré ležia v guli so stredom určeným úsečkou spájajúcou posledný konzervovaný nukleotid pred nekonzervovaným úsekom s prvým konzervovaným nukleotidom za nekonzervovaným úsekom vzhľadom na ich poradie v sekvencii. Polomer tejto gule je určený experimentálne ako 0,75 násobok dĺžky úsečky, kedy by mala obsiahnuť všetky relevantné nukleotidy. 3.2

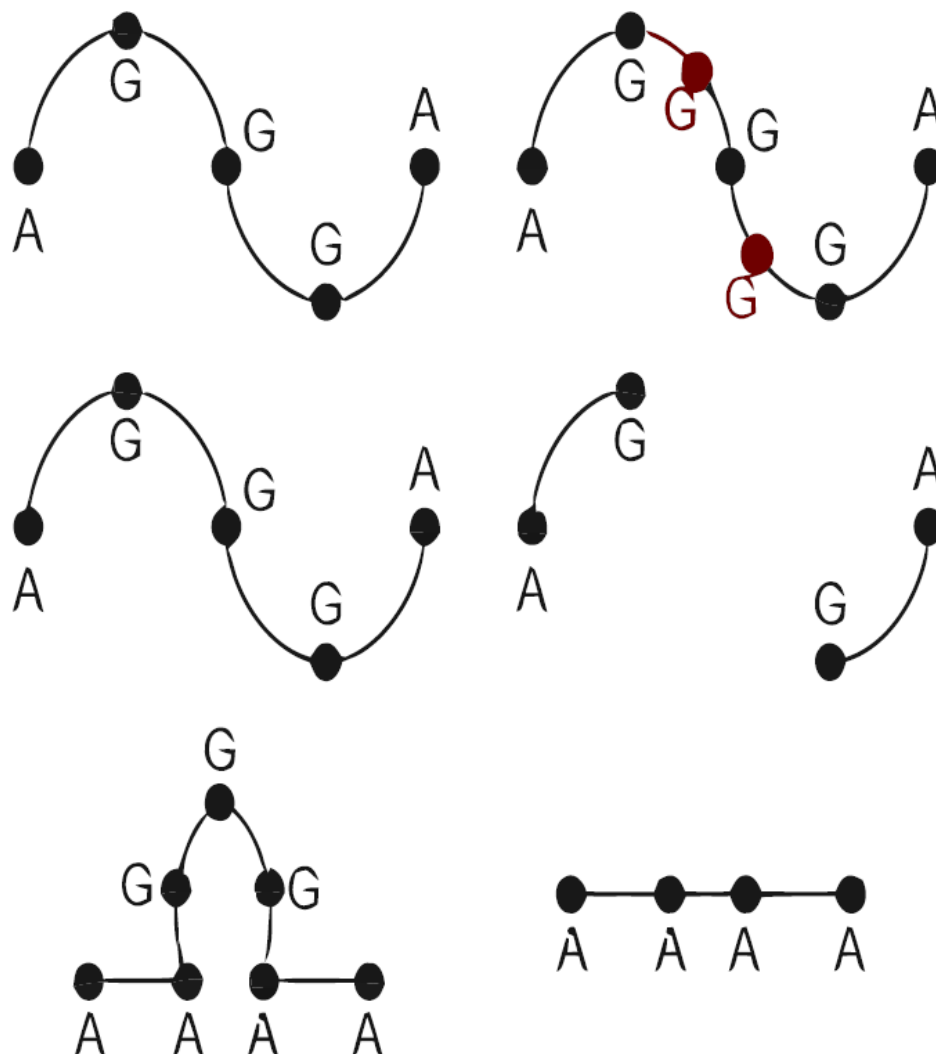
V šiestom kroku pripravíme vstupné dáta pre algoritmus FARFAR. To zahŕňa prípadné rozdelenie na úseky po 300 nukleotidov spomenuté v predchádzajúcom odstavci a prepísanie informácií o tom, ktoré nukleotidy sú pevne dané a ktoré treba dopredikovať do vstupného súboru. Takisto tu určíme parametre pre jednotlivé predikcie ako napríklad počet vygenerovaných štruktúr.

V siedmom kroku všetky takto pripravené časti predikcie spustíme a počkáme na výsledok. Toto je najpomalšia časť algoritmu, kedy FARFAR potrebuje čas minimálne pár hodín až niekoľko desiatok hodín, aby dokázal predikovať dlhšie nepredikované úseky. Tie sú napriek tomu najväčšou slabinou nášho algoritmu podľa výsledkov získaných v bakalárskej práci.

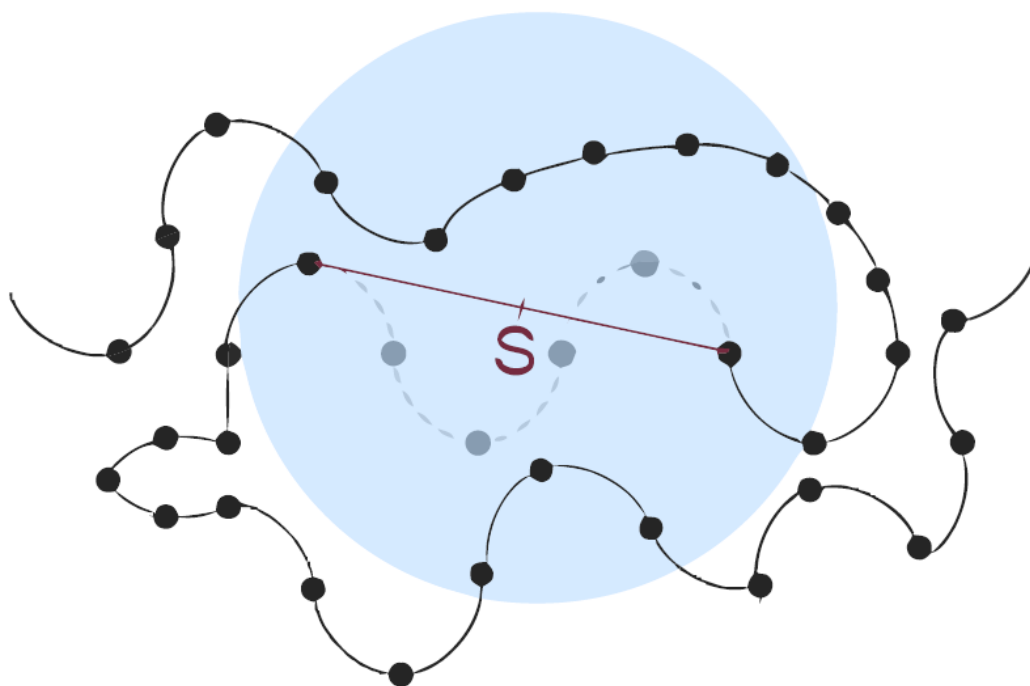
V poslednom ôsmom kroku najprv konvertujeme úseky z internej reprezentácie FARFARu do klasických pdb súborov a tie spojíme do výsledku.

3.4 Popis implementácie

Algoritmus bol implementovaný prevažne v programovacom jazyku Python 2.7. s využitím knižnice BioPython, ktorá zjednodušuje prácu so štandardnými súbormi používanými v bioinformatike, ako napríklad pdb a fasta. Okrem toho



Obrázek 3.1: Problémy, ktoré môžu nastať v štruktúre pri vložení medzier do target alebo template časti zarovnania. Prvý riadok zobrazuje komplikácie pri pokuse vložiť nukleotidy do celistvej štruktúry (teda v zarovnaní boli pridané medzery do target sekvencie). Druhý riadok ukazuje opačný problém a to vynechanie dvoch nukleotidov a roztrhnutie štruktúry (zodpovedá to vložení medzier do target sekvencie). Tretí riadok odpovedá situácií rovnakej situácií ako druhý, ale odstránenie nukleotidov zo štruktúry nespôsobuje problém, pretože odstránené nukleotidy tvorili loop, ktorý môžeme bez problémov odobrať.



Obrázek 3.2: Schématické nakreslenie sféry so stredom v bode S, ktorý je stredom úsečky spájajúcej dve krajné konzervované nukleotidy medzery v štruktúre. Všetky nukleotidy, ktoré padnú do bledomodrej gule, budú použité pri predikcii daného úseku.

používame bash scripty na manipuláciu so súbormi a spustenie predikcie vo FARFAR.

Algoritmus bol rozdelený na tri časti. Prvá pozostávala zo spustenia predikcie dopredu pripravených dvojíc na lokálnom PC s operačným systémom Window. To obsahovalo algoritmus po siedmy krok, teda bola pripravená target štruktúra do stavu, kedy treba dopredikovať nekonzervované úseky algoritmom FARFAR. Následne boli takto pripravené vstupy pre FARFAR skopírované na servery organizácie Metacentrum používajúci operačný systém unixového typu s dávkovým spracovaním úloh, kde mohli bežať paralelne viaceré predikcie za pomoci FARFAR naraz. To je veľmi dôležité, pretože de novo predikcia nekonzervovaných úsekov bola najdlhšie trvajúca časť algoritmu a predikcia jednej štruktúry mohla obsahovať niekoľko takýchto de novo predikcií. Po skončení FARFAR predikcií nekonzervovaných úsekov boli výsledky skopírované späť na lokálny PC a tam boli v treťom kroku vyhodnotené výsledky.

Časová náročnosť celého algoritmu, je závislá hlavne od nekonzervovaných úsekov, ktoré treba predikovať. Časť predikcie po algoritmus FARFAR beží v rádoch desiatok sekúnd. Pre FARFAR sme okrem pár problémových predikcií, používali obmedzenie predikcie časom 24 hodinami prípadne 100 štruktúrami. Počet a kvalita kandidátskych štruktúr, ktoré za tento čas algoritmus stihne vygenerovať závisí na počte nekonzervovaných úsekov, ich dĺžke (problematické sú hlavne dlhé nekonzervované úseky) a dĺžke celej predikovanej štruktúry vrátane konzervovaných nukleotidov. Závrečné získanie výsledkov a porovnanie s experimentálne získanými štruktúrami prebieha opäť v rádoch desiatok sekúnd.

3.5 Hlavné problémy algoritmu a jeho implementácie

Najväčším problémom v našom algoritme, ktorý sme identifikovali na základe výsledkov bakalárskej práce je predikcia dlhších nekonzervovaných úsekov ??. Preto by sme potrebovali minimalizovať takéto úseky, prípadne pomôcť algoritmu FARFAR zmenšiť prehľadávaný priestor pri generovaní kandidátskych štruktúr. To by potom mohlo pomôcť rýchlosti predikcie kandidátskych štruktúr FARFAR-om a zlepšiť jeho presnosť.

Takisto by sme chceli do väčšej miery automatizovať predikciu, presúť ju celú na jedno prostredie a zbaviť sa tak nutnosti manuálneho kopírovania súborov. Okrem toho by sme chceli predstaviť alternatívnu možnosť vstupných parametrov, kedy by nebolo treba určiť target aj template molekuly, ale stačilo by určiť target sekvenciu a algoritmus by sám našiel vhodnú štruktúru, ktorá by mohla slúžiť ako template.

4. Automatizácia predikcie a porovnanie s ModeRNA

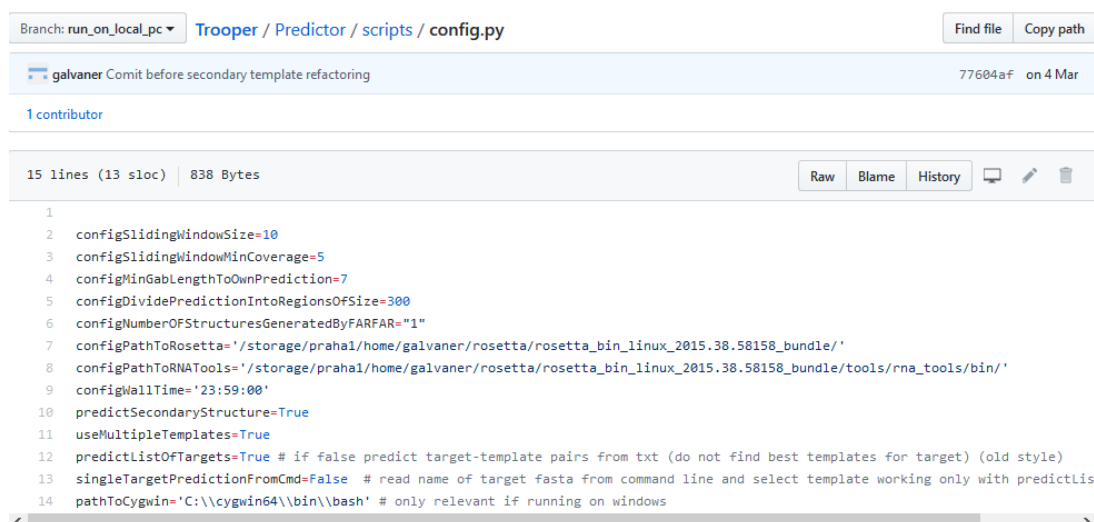
V tejto kapitole sa budeme venovať ďalšej automatizácii algoritmu a zjednodušeniu predikcie. Takisto sme doplnili funkcionality, vďaka ktorej algoritmus nájde vhodnú template štruktúru pre zadaný target. Okrem toho sme náš algoritmus porovnali s ďalším nástrojom na predikciu RNA štruktúr ModeRNA, ktorý funguje na podobnom princípe ako náš algoritmus - teda na princípe komparatívneho modelovania. Ďalej sme sa pokúsili riešiť problémy, ktoré vznikajú vďaka nezhode fasta sekvencií a sekvencií extrahovaných z odpovedajúcich pdb štruktúr. Pre implementáciu máme vytvorený repozitár na GitHub-e s adresou <https://github.com/galvaner/Trooper>.

4.1 Automatické vyhľadanie template štruktúry

V pôvodnej implementácii algoritmus očakáva pripravený súbor target a template dvojíc, ktoré bude predikovať. To je vhodné pre účely testovania algoritmu, kedy nám ide o to zadať vzťah medzi target a template sekvenciou (dĺžka, podobnosť, počet medzier v zarovnaní). V prípade, že by sme našu implementáciu chceli poskytnúť užívateľom tak, aby mohli pomocou nej napredikovať neznámu štruktúru očakávame, že užívateľ nebude vedieť, akú štruktúru je vhodné použiť ako target. Preto sme naimplementovali možnosť, zadať na vstup, iba target sekvenciu a náš algoritmus sám určí vhodnú sadu target sekvencií, podľa ktorých by bolo možné cieľovú sekvenciu predikovať čo najlepšie.

Algoritmus sme naimplementovali v metóde `TemplateSelector.SelectTemplate` a pozostáva z toho, že v cykle postupne globálne zarovnáваме target sekvenciu so sekvenciou z databázy sekvencií (v našom prípade zložka s fasta súborami). Následne zarovnanie validujeme podľa minimálnej/maximálnej podobnosti template sekvencií, ktorá nás zaujíma. Ak zarovnanie prejde validáciou, štruktúru si označíme ako vhodnú pre predikciu a pokračujeme od začiatku. Štruktúru taktiež považujeme za vhodnú ako target iba vtedy, ak žiadnu veľmi podobnú štruktúru už takto označenú nemáme (podobnosť maximálne 90%). Ukončovacie kritérium pre cyklus je buď prejdenie všetkých štruktúr a vrátenie požadovaného počtu najlepších, alebo nájdenie prvých x štruktúr splňujúcich parametre hľadania.

Takto môže užívateľ predikovať štruktúru podľa viacerých odlišných template štruktúr vo viacerých nezávislých predikciách a výsledku manuálne porovnať a vybrať ten najlepší. Nevýhodou vyhľadávania vhodných template štruktúr takýmto spôsobom je časová náročnosť, kedy vyhľadanie v našej databáze obsahujúcej 1770 fasta štruktúr môže trvať niekoľko minút, čo je výrazné spomalenie. Bolo by to možné riešiť použitím rýchleho heuristického zarovnania typu BLAST alebo FASTA ale vzhľadom na trvanie predikcie FARFAR a predpokladu, že táto funkcionality nebude využívaná na hromadnú predikciu štruktúr sme to nepovažovali za prioritu.



```
Branch: run_on_local_pc Trooper / Predictor / scripts / config.py Find file Copy path
galvaner Commit before secondary template refactoring 77604af on 4 Mar
1 contributor
15 lines (13 sloc) 838 Bytes Raw Blame History
1
2 configSlidingWindowSize=10
3 configSlidingWindowMinCoverage=5
4 configMinGapLengthToOwnPrediction=7
5 configDividePredictionIntoRegionsOfSize=300
6 configNumberOfStructuresGeneratedByFARFAR="1"
7 configPathToRosetta='/storage/prahal/home/galvaner/rosetta/rosetta_bin_linux_2015.38.58158_bundle/'
8 configPathToRNATools='/storage/prahal/home/galvaner/rosetta/rosetta_bin_linux_2015.38.58158_bundle/tools/rna_tools/bin/'
9 configWallTime='23:59:00'
10 predictSecondaryStructure=True
11 useMultipleTemplates=True
12 predictListOfTargets=True # if false predict target-template pairs from txt (do not find best templates for target) (old style)
13 singleTargetPredictionFromCmd=False # read name of target fasta from command line and select template working only with predictlis
14 pathToCygwin='C:\\cygwin64\\bin\\bash' # only relevant if running on windows
```

Obrázek 4.1: Príklad konfiguračného súboru.

4.2 Automatizácia predikcie

V pôvodnej implementácii bežala časť algoritmu na OS Windows a časť na UNIXe. Medzi operačnými systémami bolo treba manuálne kopírovať súbory, čo je zbytočne zdĺhave. Kvôli tomu sme algoritmus testovali iba na skupine vybraných dát

Preto sme sa rozhodli presunúť beh celého algoritmu na unixový systém. Dôvod, prečo sme sa rozhodli pre Unix a nie Windows je kvôli paralelizácii predikcie FARFAR v Metacentre a taktiež kvôli tomu, že Rosetta nie je na Windows podporovaná. Algoritmus bol upravený tak, že pre na predikciu stačí dodať vstupné dáta spustiť jeden skript na prípravu predikcie bodov 1-7 3.2 (teda až do spustenia FARFAR predikcie). Táto časť trvá v základnej verzii algoritmu pár sekúnd pre jednu target štruktúru v závislosti na jej dĺžke. Následne treba spustiť druhý skript, ktorý spustí predikciu FARFAR. Bolo by možné automaticky spustiť tento skript hneď po prvom, ale vzhľadom na to, že v prípade, že by prvá časť skončila s chybami, máme možnosť to prekontrolovať, a nezahltiť tak metacentrum naplánovaním zbytočných úloh. Trvanie druhého skriptu sa štandardne pohybuje okolo 24 hodín, ak nie je nastavené inak. Po dobehnutí predikcie nekonzervovaných úsekov, je treba pustiť tretí skript, ktorý konvertuje výstupy FARFAR z internej reprezentácie, to pdb súborov a štvrtý skriptm ktorý ich zmerguje do výslednej štruktúry. Oba skripty bežia v rádoch sekúnd pre jednu štruktúru. Nakoniec v prípade, že v priečinku s experimentálne získanými štruktúrami existuje predikovaná target štruktúra, porovná s ňou napredikovanú štruktúru pomocou programu PyMol a výsledok uloží.

V našom prediktore existuje mnoho konfigurovateľných parametrov, ktoré nechceme vždy zadávať na vstupe, niektoré sa nachádzajú v shell scriptoch a niektoré v python scriptoch. Preto sme vytvorili konfiguračný súbor Predictor/scripts/config.py, ktorý obsahuje všetky relevantné parametre, ktoré sú tak prehľadne na jednom mieste 4.1.

Kroky potrebné vykonať pre kompletnú predikciu štruktúry:

1. Dodať vstupný súbor target template párov

2. Uistiť sa, že potrebné pdb a fasta súbory sa nachádzajú v rovnomenných priečiinkoch.
3. Spustiť skript `prepare_rosetta_prediction.sh` a uistiť sa, že dobehol.
4. Spustiť skript `startFARFAR.sh`.
5. Počkať, kým predikcia nekonzervovaných úsekov dobehne.
6. Spustiť skript `extract_pdbsh.sh`.
7. Spustiť skript `concat_pdbsh.sh`

Pre otestovanie algoritmu sme použili všetky stiahnuté štruktúry sekvencie dĺžky 50 a viac nukleotidov. Najprv sme ich rozdelili do priehradok podľa dĺžky (50-100nt, 101-500nt, 500-viac nt). Štruktúry v priehradkách sme následne spárovali každú s každou ako target - template dvojice a tieto dvojice rozdelili do skupín podľa podobnosti a pomeru medzier vzhľadom na zarovnanie 4.1. Z bakalárskej práce sme vedeli, že predikovať sekvenciu na základe štruktúry s podobnosťou menšou ako 60% nie je vhodné, čo sa nám potvrdilo aj pri tomto pokuse. Naopak, pri podobnosti sekvencií väčšej ako 60% sme schopní dosahovať presnosť predikcie okolo 10Å. Tieto výsledky sme publikovali v článku Galvanek a kol. (2016).

4.3 Porovnanie s ModeRNA

V bakalárskej práci, sme algoritmus porovnávali s algoritmom FARFAR, ktorý z princípu nedokáže v rozumnom čase predikovať dlhšie štruktúry. Teraz sme sa rozhodli porovnať náš algoritmus s komparatívnou predikciou ModeRNA. Pre porovnanie sme sa rozhodli napredikovať rovnaké sekvencie za pomoci rovnakých target štruktúr a porovnať výsledky z ModeRNA s výsledkami nášho algoritmu. Napísali sme skript `ModeRNA/moderRNA.py`, ktorý sa pokúša predikovať target - template páry poskytuté na vstupe. Predikcia v ModeRNA je výrazne rýchlejšia a vytvorenie jednej štruktúry sa väčšinou pohybuje v rádoch minút. Výsledky oboch algoritmov sme porovnali 4.2, pričom sme uvažovali iba predikcie, ktoré úspešne vytvorili oba algoritmy. V tomto prípade sme rozlišovali dĺžky štruktúr a to 50-100 nukleotidov a 101-500 nukleotidov. Kritérium na podobnosť template

Podobnosť (%)	Gap (%)	Priemer RMSD (Å)	Smerodatná odchýlka (Å)
30-45	30-45	32,05	6,09
45-60	30-45	32,30	4,78
60-75	0-15	11,88	7,81
60-75	15-30	9,63	2,33
60-75	30-45	8,80	7,20
75-90	0-15	6,02	4,37
75-90	15-30	6,93	4,45

Tabulka 4.1: Výsledky predikcie pre target template páry dĺžky 50-500 nukleotidov.

štruktúry bolo pre všetky dĺžky rovnaké, a to 60-90%. Z výsledkov vidno, že ModeRNA bola úspešnejšia v predikovaní dlhších štruktúr, pričom naopak náš algoritmus si viedol lepšie pri predikovaní kratších štruktúr. Na druhej strane ModeRNA dokázala napredikovať celkovo viac štruktúr oproti nášmu algoritmu, pretože dokázala lepšie zvládnuť nedokonalé vstupné dáta.

4.4 Porovnanie predikcie dlhých štruktúr s ModeRNA

Algoritmy založené na princípe komparatívneho modelovania majú vďaka vysoko konzervovaným ribozomálnym RNA štruktúram potenciál predikovať prakticky neobmedzene dlhé štruktúry, ak pre ne existuje dosť dobrá template štruktúra. Predikcia štruktúr dlhých niekoľko tisíc nukleotidov samozrejme prináša problémy ako dlhšie nekonzervované úseky a celkovo dlhší čas predikcie. Skúšali sme predikovať 4 rôzne target-template páry aj pomocou nášho algoritmu aj pomocou ModeRNA algoritmu. Pravdepodobne kvôli tomu, že ModeRNA používa na vyplnenie nekonzervovaných úsekov knižnicu fragmentov, sa v takto dlhej štruktúre našiel nekonzervovaný úsek, ktorý ModeRNA nedokázala napredikovať. Náš algoritmus má síce s dlhými úsekmi problémy tiež, ale vďaka ich de novo predikcií dokázal napredikovať aj takýto úsek 4.3.

4.5 Úprava vstupných dát

Fasta a pdb súbory sme stiahli zo stránok ProteinDataBank <https://www.rcsb.org> Berman (2000). Sú to experimentálne získane štruktúry, ktoré často nie sú kompletne, alebo obsahujú chyby. Problém pre náš algoritmus nastáva hlavne v prípade, kedy sekvencia fasta súboru neseď so sekvenciou extrahovanou z pdb súboru. Takto upravené vstupné dáta sme použili na testovanie posledného vylepšenia algoritmu, teda s použitím viacerých template štruktúr.

Dĺžka (nt)	ModeRNA RMSD (Å)	Trooper RMSD (Å)
50-100	5,77	8,34
101-500	8,25	4,29
50-500	6,21	7,65

Tabulka 4.2: Porovnanie nášho algoritmu s ModeRNA.

Target(len)	Template(len)	Sim(%)	Trooper	ModeRNA
3DG0(2904)	2O45(2880)	68	13,65	No fragments candidates.
4JI1(1522)	4V4Q(1542)	71,6	14,5	No fragments candidates.
4V6W(1995)	4V6X(1869)	68,3	32,7	Sequences do not match.
3DG5(1542)	3J2G(1533)	99,4	9,6	9,6

Tabulka 4.3: Porovnanie výsledkov predikcie vybraných dlhých štruktúr pomocou nášho algoritmu a pomocou ModeRNA.

túr. Generovanie je naimplementované v priečinku s relatívnou cestou Predictor/MyTools/CreateFastasFromPdb.

Principiálne problémy s rozdielnymi sekvenciami v našom algoritme nastávajú v troch konkrétnych prípadoch:

1. Pri hľadaní vhodnej template štruktúry pre predikciu (funkcionalita automatického vyhľadávania template štruktúry).
2. Pri mapovaní template štruktúry na konzervované úseky plynúce zo zarovnania sekvencií. 5
3. Pri hľadaní a mapovaní sekundárnych template štruktúr.

V prípade že nastane takáto situácia algoritmus nedokáže pokračovať v predikcii. Preto sme to doteraz riešili tak, že sme sa pokúsili template fasta sekvenciu posunúť pridaním dummy reziduí na jej začiatok (predpoklad, že z FASTA sekvencie chýba na začiatku pár reziduí a ďalej sa už bude poradie nukleotidov zhodovať s tým v pdb súbore) a v prípade, že sa nám niečo takéto nepodarilo nájsť danú štruktúru sme ako template nepoužili.

Rozhodli sme sa skúsiť eliminovať tento problém tak, že sme si vygenerovali fasta sekvencie z pdb súborov. Tu ale nastáva problém s častou nekompletnosťou pdb súborov, kedy býva časté, že začiatok, prípadne koniec štruktúry chýba. Chýbajúce úseky znova nahradzujeme dummy reziduami (písmeno X). Takéto štruktúry môžu byť následne bez problémov použité ako template alebo sekundárny template pri predikcii, pretože pridaný dummy nukleotid X sa nezarovná na target sekvenciu a bude neskôr dopredikovaný. Z princípu sekencia obsahujúca takéto dummy nukleotidy nemôže byť predikovaná, pretože nevieme, aký nukleotid máme na dané miesto vlastne predikovať.

Okrem problému s mapovaním štruktúry a sekvencie sme riešili aj duplicitu rôzne pomenovaných, ale pritom rovnakých štruktúr. V pôvodnej implementácii nám to nerobilo problém, pretože sme si dopredu testovacie dáta rozdelili podľa podobností zarovnaní ich sekvencií a tie vzájomne spárovali. Teda ako template sme používali len dosť odlišné template sekvencie od target sekvencie, nakoľko predikovať sekvenciu pomocou inej so skoro 100% zhodou by bolo triviálne. V prípade, že hľadáme vhodný template sa však chceme vyhnúť zbytočnému zarovnávaniu sekvencií a preto sme pri generovaní fasta súborov rovno podobné odfiltrujeme. Docielime to jednoducho tak, že pri vygenerovaní novej fasta sekvencie ju globálne zarovnáme na všetky už vygenerované fasta sekvencie a ak podobnosť presiahne istú konštantnú hranicu (použili sme 97,5%), tak takúto sekvenciu nepridáme, medzi už vygenerované sekvencie.

5. Predikcia sekundárnej štruktúry

Ako prvé opatrenie, na zlepšenie rýchlosti a presnosti predikovania nekonzerovvaných úsekov sme navrhli najprv riešiť jednoduchší problém a to predikciu sekundárnej štruktúry targetu a pomocou nej neskôr predikovať terciárnu štruktúru nekonzerovvaných úsekov, vďaka čomu predpokladáme, že by sa mohol výrazne zmenšiť prehľadávaný priestor pri generovaní kandidátskych štruktúr algoritmom FARFAR.

5.1 Príprava dát

5.2 Predikcia sekundárnej štruktúry

5.3 Integrácia do existujúceho algoritmu

5.4 Experiment

5.5 Výsledky

6. Použitie viacerých template štruktúr pri predikcií

Ako druhé opatrenie, na zlepšenie rýchlosti a presnosti predikovania nekonzervovaných úsekov sme navrhli skúsiť použiť viacero template štruktúr na predikciu jednej template štruktúry. Očakávali sme pritom, že sa nám podarí zmenšiť počet a dĺžku nekonzervovaných úsekov v predikovanej štruktúre a tým výrazne zjednodušiť ich predikciu algoritmom FARFAR.

6.1 Varianty a prístupy

6.2 Výber vhodných štruktúr

6.3 Algoritmus

6.4 Integrácia do existujúceho algoritmu

6.5 Experiment

6.6 Výsledky

Závěr

Seznam použité literatury

- ALBERTS B, JOHNSON A, L. J. A. K. (2002). *Molecular Biology of the Cell*. 4th edition. Garland Science, New York. ISBN 0-8153-3218-1.
- BERMAN, H. M., W. J. F. Z. G. G. B. N. W. H. A. S. I. N. (2000). The protein data bank. *Nucleic Acids Research*, pages 235242, number = 28.
- BIESIADA, M., PURZYCKA, K. J., SZACHNIUK, M., BLAZEWCZ, J. a ADAMIĄK, R. W. (2016). *Automated RNA 3D Structure Prediction with RNA-Composer*, pages 199–215. Springer New York, New York, NY. ISBN 978-1-4939-6433-8. doi: 10.1007/978-1-4939-6433-8_13. URL https://doi.org/10.1007/978-1-4939-6433-8_13.
- BONIECKI, M. J., LACH, G., DAWSON, W. K., TOMALA, K., LUKASZ, P., SOLTYSINSKI, T., ROTHER, K. M. a BUJNICKI, J. M. (2015). SimRNA: a coarse-grained method for RNA folding simulations and 3D structure prediction. *Nucleic Acids Research*, **44**(7), e63–e63. ISSN 0305-1048. doi: 10.1093/nar/gkv1479. URL <https://doi.org/10.1093/nar/gkv1479>.
- CRICK, F. (1970). Central dogma of molecular biology. *Nature*, **227**(2), 561–563.
- DAS R., KARANICOLAS J., B. D. (2010). Atomic accuracy in predicting and designing noncanonical rna structure. *Nature methods*, (7), 291–294.
- EDDY, S. (2004). Genome regulation by long noncoding rnas. *Nature Biotechnology*, (22).
- FELDEN, B. (2007). Current opinion in microbiology. *Current opinion in microbiology*, (10), 286–291.
- FLORES, S., SHERMAN, M., M BRUNS, C., EASTMAN, P. a ALTMAN, R. (2011). Fast flexible modeling of rna structure using internal coordinates. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, **8**, 1247–57. doi: 10.1109/TCBB.2010.104.
- FRELLSEN, J., MOLTKE, I., THIIM, M., MARDIA, K. V., FERKINGHOFF-BORG, J. a HAMELRYCK, T. (2009). A probabilistic model of rna conformational space. *PLOS Computational Biology*, **5**(6), 1–11. doi: 10.1371/journal.pcbi.1000406. URL <https://doi.org/10.1371/journal.pcbi.1000406>.
- GALVANEK, R., HOKSZA, D. a PÁNEK, J. (2016). Template-based prediction of rna tertiary structure. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1897–1900. doi: 10.1109/BIBM.2016.7822808.
- JENNY GU, P. E. B. (2009). *Structural Bioinformatics*. 2th edition. Wiley-Blackwell, New Jersey. ISBN 978-0-470-18105-8.
- KRUPOVIC, M., BLOMBERG, J., COFFIN, J. M., DASGUPTA, I., FAN, H., GEERING, A. D., GIFFORD, R., HARRACH, B., HULL, R., JOHNSON, W.,

- KREUZE, J. F., LINDEMANN, D., LLORENS, C., LOCKHART, B., MAYER, J., MULLER, E., OLSZEWSKI, N. E., PAPPU, H. R., POOGGIN, M. M., RICHERT-PÖGGELER, K. R., SABANADZOVIC, S., SANFAÇON, H., SCHÖELZ, J. E., SEAL, S., STAVOLONE, L., STOYE, J. P., TEYCHENEY, P.-Y., TRISTEM, M., KOONIN, E. V. a KUHN, J. H. (2018). Ortervirales: New virus order unifying five families of reverse-transcribing viruses. *Journal of Virology*, **92**(12). ISSN 0022-538X. doi: 10.1128/JVI.00515-18. URL <https://jvi.asm.org/content/92/12/e00515-18>.
- MOORE, P. B. (1999). *The RNA World*. 2th edition. Cold Spring Harbor Laboratory, New Haven. ISBN ISBN 0-87969-561-7.
- NEEDLEMAN S. B., W. C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, (48), 443–453.
- NOLLER, H. (1984). Structure of ribosomal rna. *Annual Review of Biochemistry*, (53), 119–162.
- QIAN, B., RAMAN, S., DAS, R., BRADLEY, P., MCCOY, A. J., READ, R. J. a BAKER, D. (2007). High-resolution structure prediction and the crystallographic phase problem. *Nature*, **450**(7167), 259. doi: 10.1038/nature06249. URL <https://app.dimensions.ai/details/publication/pub.1001896118andhttp://europepmc.org/articles/pmc2504711?pdf=render>. Exported from <https://app.dimensions.ai> on 2019/05/26.
- ROTHER, M., ROTHER, K., PUTON, T. a BUJNICKI, J. M. (2011). RNA tertiary structure prediction with ModeRNA. *Briefings in Bioinformatics*, **12** (6), 601–613. ISSN 1477-4054. doi: 10.1093/bib/bbr050. URL <https://doi.org/10.1093/bib/bbr050>.
- ROTHER K, ROTHER M, B. M. P. T. B. J. (2011). Rna and protein 3d structure modeling: similarities and differences. *Journal of Molecular Modeling*, (10), 2325–2336.
- RYU, W.-S. (2017). Chapter 2 - virus structure. In RYU, W.-S., editor, *Molecular Virology of Human Pathogenic Viruses*, pages 21 – 29. Academic Press, Boston. ISBN 978-0-12-800838-6. doi: <https://doi.org/10.1016/B978-0-12-800838-6.00002-3>. URL <http://www.sciencedirect.com/science/article/pii/B9780128008386000023>.
- SCHUDOMA C., MAY P., N. V. W. D. (2010). Sequence-structure relationships in rna loops: establishing the basis for loop homology modeling. *Nucleic Acids Research*, (38), 970–980.
- SHARMA, S., DING, F. a DOKHOLYAN, N. V. (2008). iFoldRNA: three-dimensional RNA structure prediction and folding. *Bioinformatics*, **24**(17), 1951–1952. ISSN 1367-4803. doi: 10.1093/bioinformatics/btn328. URL <https://doi.org/10.1093/bioinformatics/btn328>.
- SMITH T. F., W. M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, (147), 195–197.

Seznam obrázků

1.1	Príklad nakreslenia sekundárnej štruktúry RNA. Eddy (2004) . . .	6
1.2	Príklad terciárnej štruktúry RNA nachádzajúcej sa v baktérii esche- richia coli.	7
1.3	Princíp rentgenovej kryštalografie. Ryu (2017)	8
2.1	Vzťah dĺžky štruktúr a percentuálneho pomeru identických residuí v sekvenciách určujúce predpoklad, že štruktúry takýchto sekvencií sú podobné. Jenny Gu (2009)	10
2.2	Reprezentácia RNA fragmentu pomocou siedmych uhlov. Frellsen a kol. (2009)	11
2.3	Needleman–Wunsch algorithm (2014) Wikipedia dostupné na https://en.wikipedia.org/wiki/Needleman\T1\textendashWunsch_algorithm 27.05.2019. Príklad jedného z troch najlepších zarovnaní dvoch sek- vencií: GCATG-CU G-ATTACA	12
2.4	Porovnanie princípu de novo a template based predikcie Rother K (2011)	14
3.1	Problémy, ktoré môžu nastať v štruktúre pri vložení medzier do tar- get alebo template časti zarovnania. Prvý riadok zobrazuje kom- plikácie pri pokuse vložiť nukleotidy do celistvej štruktúry (teda v zarovnaní boli pridané medzery do target sekvencie). Druhý ri- adok ukazuje opačný problém a to vynechanie dvoch nukleotidov a roztrhnutie štruktúry (zodpovedá to vložení medzier do tar- get sekvencie). Tretí riadok odpovedá situácií rovnakej situácií ako druhý, ale odstránenie nukleotidov zo štruktúry nespôsobuje pro- blém, pretože odstránené nukleotidy tvorili loop, ktorý môžeme bez problémov odobrať.	20
3.2	Schématické nakreslenie sféry so stredom v bode S, ktorý je stre- dom úsečky spájajúcej dve krajné konzervované nukleotidy med- zery v štruktúre. Všetky nukleotidy, ktoré padnú do bledomodrej gule, budú použité pri predikcii daného úseku.	21
4.1	Príklad konfiguračného súboru.	24

Seznam tabulek

2.1	Prehľad niektorých programov určených na predikciu RNA s informáciou o type použitého algoritmu.	15
3.1	Prehľad štyroch situácií, ktoré môžu nastať na každej pozícii v zarovnaní dvoch sekvencií.	18
4.1	Výsledky predikcie pre target template páry dĺžky 50-500 nukleotidov.	25
4.2	Porovnanie nášho algoritmu s ModeRNA.	26
4.3	Porovnanie výsledkov predikcie vybraných dlhých štruktúr pomocou nášho algoritmu a pomocou ModeRNA.	26

Seznam použitých zkratek

A. Přílohy

A.1 První příloha