Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Identifikácia a analýza repetitívnych sekvencií v nanopórových dátach

Bakalárska práca

2019
Samuel Molčan

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Identifikácia a analýza repetitívnych sekvencií v nanopórových dátach

Bakalárska práca

Študijný program:	Informatika
Študijný odbor:	Informatika
Školiace pracovisko:	Katedra aplikovanej informatiky
Školiteľ:	doc. Mgr. Tomáš Vinař, PhD.

Bratislava, 2019
Samuel Molčan

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Samuel Molčan
Študijný program: informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Identifikácia a analýza repetitívnych sekvencií v nanopórových dátach
Identification and Analysis of Sequence Repeats in Nanopore Data

Anotácia: Analýza repetitívnych sekvencií je dôležitou súčasťou pri analýze osekvenovaných genómov. Repetitívne sekvencie je však veľmi ťažké presne identifikovať v nanopórových dátach, ktoré obsahujú vysoké množstvo chýb. Cieľom práce je vytvoriť nové metódy pre identifikáciu a analýzu repetitívnych sekvencií s priamym použitím surového signálu z nanopórového sekvenovania.

Vedúci: doc. Mgr. Tomáš Vinař, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.

Spôsob sprístupnenia elektronickej verzie práce:
bez obmedzenia

Dátum zadania: 06.11.2018

Dátum schválenia: 06.11.2018 doc. RNDr. Daniel Olejár, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Poďakovanie: Rád by som sa poďakoval svojmu školiteľovi, Tomášovi Vinařovi, za jeho vedenie a trpezlivosť pri našich častých stretnutiach a Bronislave Brejovej, ktorá mi vždy bola ochotná poradiť.

Taktiež mi nedá nepoďakovať mojej rodine, ktorá ma vždy podporuje, či priateľom motivujúcim ma napredovať.

Ďakujem aj sebe za vôľu vydržať a nevzdávať sa.

Abstrakt

V tejto práci sa zaoberáme hľadaním tandemových opakovaní v dátach zo sekvenačnej platformy MinION. Hľadanie tandemových opakovaní v postupnosti báz vytvoreného prekladačom báz je kvôli jeho vysokej chybovosti nepresné. V našej práci navrhujeme spôsob, ktorý v signále sekvenátora MinION identifikuje tandemové opakovania bez využitia prekladača báz. Pri testovaní tento algoritmus dosahuje 2.5-násobné zlepšenie v aspoň jednej z vlastností (specificite alebo senzitivite) podľa kvality čítaní a štruktúry príslušného DNA reťazca.

Kľúčové slová: DNA sekvenovanie, MinION, repetitívna DNA

Abstract

In this thesis we examine the problem of searching data from MinION sequencing platform for tandem repeats. Searching for tandem repeats in sequences created by a base caller is inaccurate due to its high error rate. In this bachelor thesis, we present a new method for finding tandem repeats which works without base calling the signal. During the testing, this algorithm reaches 2.5 times improvement in at least one of the features, specificity or sensitivity, depending on the quality of sequence read and structure of a particular DNA sequence.

Keywords: DNA sequencing, MinION, repetitive DNA

Obsah

Zoznam obrázkov

Zoznam tabuliek

Cieľom tejto práce je navrhnúť metódu na hľadanie repetitívnych sekvencií v dátach nanopórového sekvenátora MinION.

Genetická informácia je v DNA skladovaná ako postupnosť štyroch dusíkatých báz (adenín, cytozín, guanín, tymín). Už niekoľko desiatok rokov vznikajú nové postupy na sekvenovanie DNA, čo znamená preklad reálnej vzorky na postupnosť vyššie spomenutých báz.

Existuje viacero spôsobov ako sekvenovať vzorku. Jedným z z najnovších, najlacnejších a zároveň aj rýchlym z nich je nanopórové sekvenovanie. Pri tomto druhu sekvenovania vlákno prechádza nanopórom - veľmi malým otvorom o veľkosti niekoľkých nanometrov - v ktorom tečie elektrický prúd s daným napätím. Sledovaním zmeny napätia, počas prechodu vlákna, môžeme určiť postupnosť báz. Nevýhodou je pomerne veľká chybovosť v porovnaní so skutočnosťou.

Veľkú časť ľudského genómu, ale aj genómu iných organizmov, tvoria repetitívne oblasti, miesta, ktoré sú tvorené takmer rovnakou postupnosťou dusíkatých báz. U človeka sú často príčinou rôznych genetických chorôb (napr. Huntingtonova choroba [3]). Za všetky spomenieme *teloméry*, koncové časti eukaryotických buniek. Pri každej replikácii bunky sa teloméry skracujú.

V bakalárskej práci sa pokúsime o vytvorenie metódy, ktorá v dátach nanopórového sekvenátora MinION [10] nájde repetetívne oblasti bez toho, aby sme celé dáta museli pretransformovať na postupnosť báz.

V kapitole 1 popíšeme spôsob sekvenovania pomocou nástroja MinION, zadefinujeme pojem repetitívneho opakovania a vysvetlíme spôsob fungovania jedného známeho programu na hľadanie takýchto opakovaní. Kapitola 2 bude venovaná rozboru nami riešeného problému a návrhu riešenia. V kapitole 3 implementujeme hľadač repetitívnych sekvencií. Nakoniec v kapitole 4 budeme nami navrhnutú metódu testovať.

Kapitola 1

Definícia problému

1.1 Tandemové opakovania

Tandemová repetitívna DNA je motív opakujúci sa niekoľkokrát v rámci genómu za sebou (priamo alebo invertovane) a tvorí markantnú časť DNA eukaryotických organizmov. Napriek tomu, ich význam nie je jasný. Niektoré majú rôzne regulačné a evolučné funkcie, ale existujú aj také, ktoré nemajú žiadnu známu významnú rolu [3].

V praxi sa tandemové opakovania využívajú na určovanie predkov organizmov. Napríklad, jeden organizmus môže niesť tri tandemové opakovania v určitom géne, zatiaľ čo iný organizmus môže mať pätnásť, bez zjavných rozdielov medzi týmito dvoma jednotlivcami, ktoré sa dajú vystopovať na danom géne. Pri určovaní predka potom hľadáme taký genóm, ktorý sa čo najviac zhoduje v počte opakovaní v tandemových oblastiach [7].

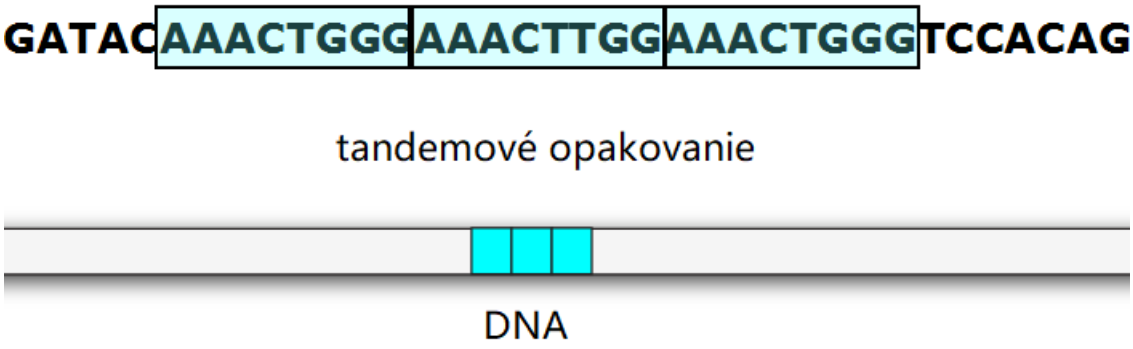
Pomáhajú taktiež pri diagnostike rakoviny a pri posúdení jej progresu. V nádorových bunkách, ktoré majú poškodenú kontrolu replikácie, môžu opakovania, takzvané mikrosatelity[1], pribúdať alebo ubúdať pri každom procese mitózy, čím sa začne značne odlišovať od hostiteľského tkaniva. Sú teda významným laboratórnym a analytickým nástrojom.

Ďalším dôvodom, prečo venujeme tandemovým opakovaniam pozornosť je, že spôsobujú u človeka genetické choroby, za všetky spomeňme Huntingtonovu chorobu [3], ktorá vedie k odumieraniu mozgových buniek.

1.2 Nanopórové sekvenovanie

Metódy sekvenovania sa začali vytvárať v druhej polovici dvadsiateho storočia. Technológia odvtedy pokročila, a tak sa sekvenovanie stalo lacnejšie a prístupnejšie pre širšiu akademickú verejnosť. Rôzne sekvenačné technológie produkujú rôzne dĺžky čítaní,

[1] mikrosatelit - tandemové opakovanie dĺžky jeden až šesť nukleotidov



Obr. 1.1: Tandemové opakovania

partícií DNA, z ktorých sa vybuduje celá štruktúra na základe vzájomných prekryvov.

V sekvenátore, prístroji určenom na sekvenovanie DNA vzorky, v polymérovej membréne oddeľujúcej dve komory sa vytvorí otvor veľkosti niekoľkých nanometrov, ktorý je tak malý, že ním prejde len jedno vlákno DNA. Nanopórom tečie elektrický prúd s napätím ovplyvňovaným bázami vlákna DNA, ktoré ním prechádzajú, pričom každá báza ovplyvňuje napätie inak. Zaznamenáva sa hodnota napätia v čase a z príslušných zmien vieme určiť výslednú postupnosť. V skutočnosti na nanopór vplýva *k-mer*, teda podpostupnosť veľkosti k báz, pričom k je prirodzené číslo. Membrána obsahuje niekoľko desiatok až stoviek nanopórov, vďaka čomu vieme sekvenovať niekoľko vlákien DNA súčasne. Zvyčajne $k = 5$, no rôzne pätice báz môžu mať takmer totožné stredné hodnoty napätia.

Výhodou takéhoto sekvenovania sú dlhé čítania, vysoká rýchlosť, výsledky v reálnom čase a nízka cena. Nevýhodou je pomerne vysoká chybovosť, spôsobená viacerými faktormi, ako napríklad:

1. nekonštantná rýchlosť pohybu vlákna prechádzajúceho nanopórom - z toho aj rôzny čas pôsobenia bázy na napätie,
2. kontaminácia vzorky vláknom DNA iného organizmu,
3. vplyv predchádzajúcich vlákien, ktoré boli sekvenované pomocou tej istej membrány.

Výstupné surové dáta vzniknú vytvorením číselnej postupnosti, ktorá predstavuje nameranú hodnotu v čase. Zariadenie urobí v priemere osem meraní na bázu [10], takže dĺžka surových dát je priamo úmerná dĺžke sekvenovanej DNA. Takto vytvorené dáta v prípade tandemových repetícií budu opakovane ovplyvňované rovnakými *k-mermi* a teda aj výsledný signál bude vykazovať repetitívne charakteristiky.

V bakalárskej práci využívame dáta zo sekvenátora MinION vytvoreného firmou Oxford Nanopore Technologies [10]. Zariadenie využíva na sekvenovanie DNA technológiu nanopórov.

1.3 Hľadanie tandemových opakovaní v nanopórových dátach

Výsledkom sekvenovania sú **surové sekvenačné dáta**, teda dvojice čas a nameraná hodnota elektrického napätia. Ukladáme ich do súboru typu $.fast$.

Existujúce algoritmy a nástroje na hľadanie repetitívnych sekvencií [3] používajú postupnosti báz A, C, G, T. Takéto špecifické opakujúce sa postupnosti nazývame *repetitívne motívy*. Do takejto formy ich pretransformujeme pomocou *prekladača báz*.

1.3.1 Prekladač báz

Preklad signálu vytvoreného sekvenátorom do postupnosti báz sa vykonáva pomocou prekladača báz (basecaaler) [6]. Najlepšie moderné prekladače báz majú presnosť okolo 90%, pričom výsledná presnosť môže byť zvýšená skombinovaním niekoľkých nezávislých sekvenovaní tej istej vzorky.

Väčšina dnešných moderných prekladačov používa neurónové siete, čo znamená, že sa musia natrénovať na skutočných dátach [15]. Presnosť prekladu tak závisí od trénovacej množiny.

Chyby prekladača báz vieme rozdeliť do dvoch kategórií. Prvou sú chyby *náhodné*, teda také, ktoré nastanú na náhodnom mieste. Napríklad, ak máme čítania s 80% presnosťou, ktorých chyby sú náhodné, tak výsledný konsenzus môže byť presný na 100%. Druhou skupinou sú chyby *systematické*, vyskytujúce sa vždy za určitých podmienok. V takomto prípade aj perfektné čítania s 98% presnosťou môžu vytvoriť konsenzus maximálne s 98% presnosťou [14].

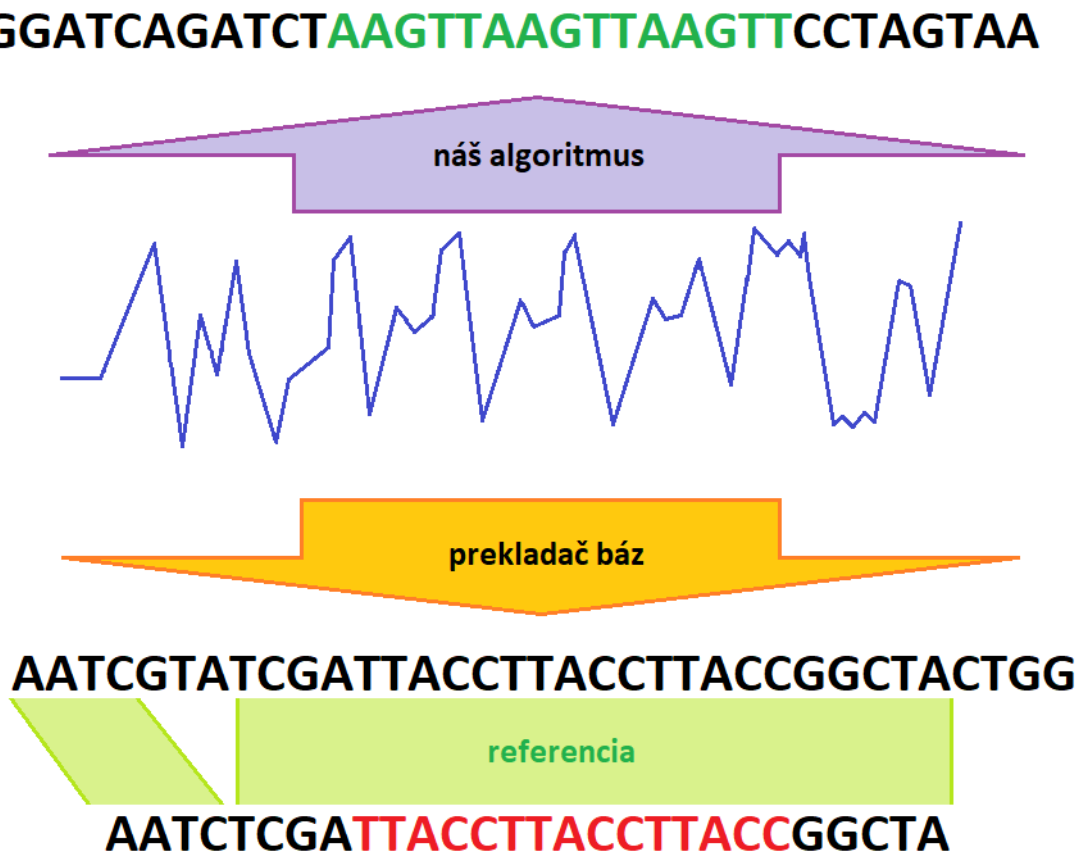
Porovnanie metód hľadania tandemových opakovaní

Tradičným spôsobom hľadania tandemových opakovaní je najprv osekvenovať danú vzorku, potom použiť prekladač báz, ktorý sekvenačné dáta dokáže interpretovať ako reťazec nad abecedou A, C, G, T a následne v tomto reťazci hľadať opakujúce sa sekvencie metódami, ktoré popíšeme v kapitole 2. Výstupom takéhoto hľadania sú informácie o repetitívnych motívoch spolu s polohami, kde sa vyskytujú a počtom výskytu.

V prípade dát pochádzajúcich zo sekvenátora MinION je tento proces naviac komplikovaný tým, že prekladač báz má vysokú chybovosť a táto chybovosť je dokonca

vyššia v repetitívnych úsekoch, prípadne niekedy v takýchto úsekoch prekladač báz zlyhá úplne.

V našej práci sa preto snažíme vytvoriť nástroj, ktorý by dokázal detekovať tandemové opakovania priamo v surovom sekvenačnom signále. V kapitole 1.2 sme popísali, ako surový signál súvisí so sekvenciou, ktorej sekvenovaním vznikol a je zjavné, že sekvenovanie repetitívnej sekvencie vedie ku repetitívnemu signálu. Našou úlohou je teda v surovom signále identifikovať polohu tandemových opakovaní, motívy a počty opakovaní, podobne ako je to pri hľadaní tandemových opakovaní v reťazcoch (obr. 1.2).



Obr. 1.2: Vizuálne porovnanie originálneho a nášho hľadania tandemových opakovaní

Môžeme vidieť, že *signál zo sekvenátora* je preložený pomocou prekladača báz na *postupnosť báz*, ktorá je zarovnaná s *referenciou*. V samotnej referencii sú vyznačené vyhľadané tandemové opakovania, nájdené pomocou niektorého zo štandardných nástrojov. K signálu sú zarovnané tandemové opakovania nájdené našou metódou.

Kapitola 2

Metódy hľadania tandemových opakovaní v reťazcoch

Úloha hľadania tandemových opakovaní je tradične definovaná na reťazcoch. V tejto kapitole sa preto budeme zaoberať metódami, ktoré hľadajú tandemové opakovania v reťazcoch a v nasledujúcej kapitole budeme riešiť rozšírenie týchto metód na nanopórové signály.

2.1 Základný prístup ku hľadaniu tandemových opakovaní

V prípade, že poznáme dĺžku tandemového opakovania, stačil by na hľadanie jednoduchý postup. Systematicky by sme prechádzali reťazec zľava doprava a ak je aktuálny úsek dostatočne podobný s úsekom posunutým o d pozícií, tak sa jedná o tandemové opakovanie.

V skutočnosti sa jednotlivé metódy na hľadanie tandemových opakovaní líšia v tom, ako sa hľadá vzdialenosť d a v rozhodovaní, či sa dve kópie dostatočne zhodujú.

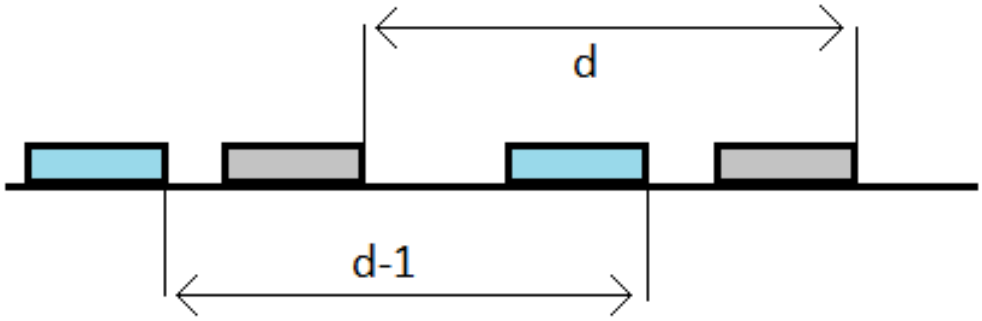
V ďalších podkapitolách popíšeme, ako k riešeniu týchto dvoch podproblémov pristupujú Tandem Repeat Finder [3] a *tantan* [9], ktoré sú jedny z najpoužívanejších bioinformatických nástrojov na tento účel.

2.2 Jednoduchý model a Tandem Repeat Finder

Vyhľadávač tandemových opakovaní - *Tandem Repeat Finder* (ďalej len TRF) využíva nasledujúci model:

- predpokladáme, že dĺžka opakovania je d,

- ak sme mali sekvenciu x_1, x_2, ... x_n a na pozícii $n+1$ začína ďalší výskyt opakovania, tak $x_{n+i} = x_{n-d+i}$ s pravdepodobnosťou P_m, kde P_m je parametrom modelu,
- navyše, s pravdepodobnosťou P_i, ktorá je taktiež parametrom modelu, môže nastať inzercia alebo delécia, ktorá posunie hodnotu d o $+1$ alebo -1.



Obr. 2.1: Variabilita vzdialenosti pre predpoklad d

Keďže možných začiatkov tandemových opakovaní a možných hodnôt d je veľa, nie je možné efektívne prehľadať všetky možnosti. Z tohto dôvodu sa za účelom zrýchlenia TRF pracujeme s tzv. $k - sondami$, čo sú páry presných zhôd vo vzdialenosti k a ich relatívnymi vzdialenosťami. Tieto páry (i, j) predstavujú časť reťazca, kde $x_i = x_j, x_{i+1} = x_{j+1}, ..., x_{i+k} = x_{j+k}$ a vzdialenosť medzi týmito potencionálnymi opakovaniami je $j - i$.

V rámci tandemového opakovania dĺžky d očakávame veľa takýchto párov s dĺžkami v blízkom okolí d, nie však nutne presne dĺžky d, kvôli možným inzerciám či deléciám.

Rozdelenie možnej dĺžky medzi jednotlivými výskytmi nám pomáha odlíšiť tandemové od rozptýlených opakovaní. Ak si predstavíme, že opakovania vo vzdialenosti d rozdelíme na nadväzujúce intervaly, tak pre tandemové sekvencie budú rozdelené v intervaloch rovnomerne, naopak, pri rozptýlených opakovaniach budú viac na pravej strane intervalu.

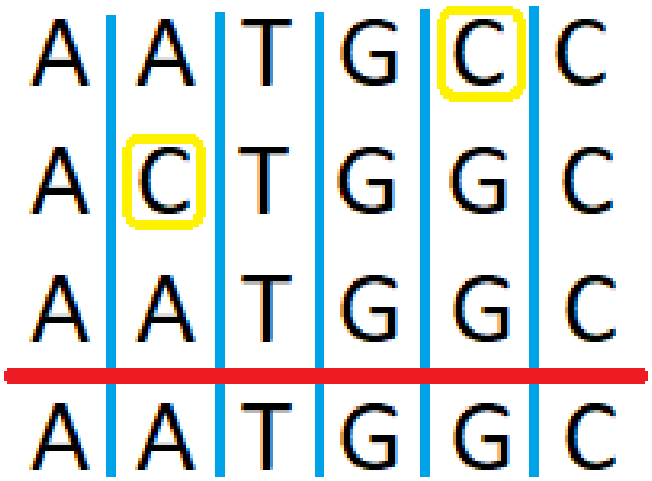
Pre dané parametre P_m a P_i tak vieme určiť kritériá, pre ktoré má zmysel podrobnejšie preskúmať existenciu tandemového opakovania konkrétnej dĺžky d od konkrétnej pozície i na základe veľkého množstva k-sond v tejto oblasti (pre vhodne zvolené k) [3].

Zvolené kandidátske pozície a potencionálne hodnoty d preskúmame pomocou zložitejšieho algoritmu, ktorý sa snaží nájsť segmentáciu na jednotlivé opakovania a ich zarovnanie tak, aby pravdepodobnosť takéhoto zarovnania vzhľadom k parametrom P_m a P_i bola najlepšia možná.

Analytická fáza nastáva len vtedy, keď sú splnené kritériá opísané vyššie. V tejto chvíli máme kandidáta na vzor pre tandemové opakovania na pozícii $j + 1$ až i. Táto podpostupnosť je vybraná a dynamickým programovaním zarovnaná s okolím (tento drahý krok prebieha len pre potencionálne riešenia), pričom na zníženie času behu algoritmu obmedzíme výpočet len na úzky pás diagonály. Táto diagonála sa periodicky presúva na najaktuálnešiu pozíciu postupnosti k zhôd. Ak sú detekované aspoň dva výskyty tohoto vzoru besprostredne za sebou, našli sme tandemové opakovanie.

V prípade, že sa v oblasti tandemových opakovaní vyskutuje veľa kópií, môže program ohlásiť rôzne veľkosti vzoru. Napríklad pre veľkosť vzoru 15 môže TRF nahlásiť aj veľkosť vzoru aj 30, 45 atď. Zamedzenie veľkého množstva redundatných výsledkov nie je obmedzené maximálnou alebo minimálnou dĺžkou opakovania (pretože neraz môžu mať väčšie veľkosti lepšie skóre zarovania, ako malé, a naopak), ale maximálnym počtom možných potencionálnych dĺžok, v tomto programe je hranica stanovená na tri.

Výsledný vzor h však ešte nemusí byť najlepší možný. Pre zlepšenie skóre výsledného zarovnania sa používa pravidlo väčšiny, **konsenzus**, podľa všetkých kópií h, čím vznikne h_f. Daná oblasť sa opätovne zarovná a až toto zarovnanie s vzorom h_f je výstupom programu.



Obr. 2.2: Zarovnanie všetkých kópií a ich konsenzus

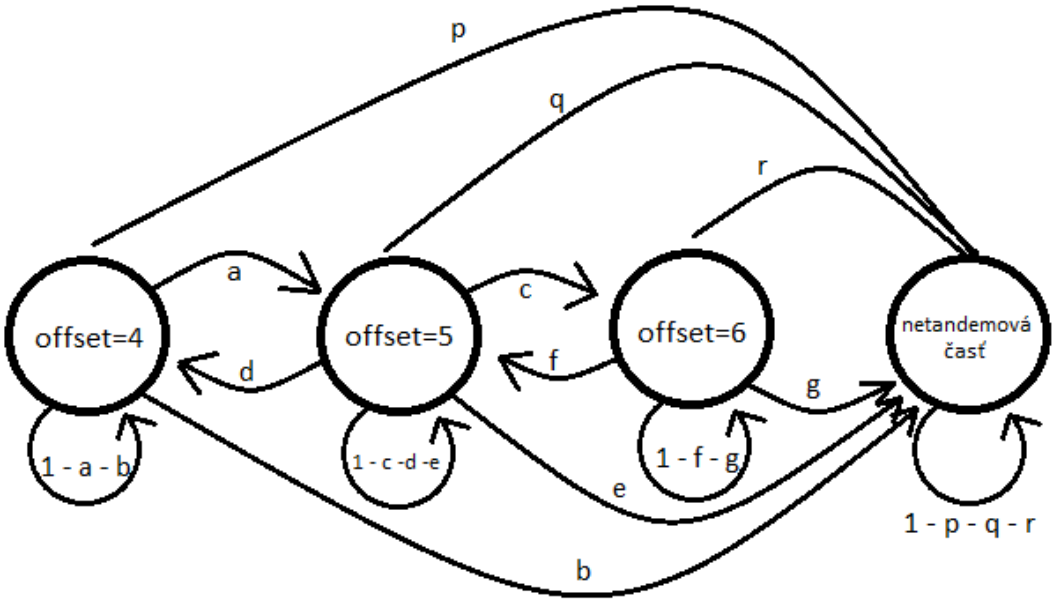
2.2.1 Modelovanie pomocou skrytých Markovovských modelov (*tantan*)

Program *tantan* používa na detekciu tandemových opakovaní prístup modelovania na základe skrytých Markovovských modelov (ďalej len HMM).

Jedná sa o *generatívny model*, ktorý sa vyznačuje možnosťou v každom stave vygenerovať jedno písmeno reťazca spolu s jeho anotáciou, pričom pod pojmom anotácia chápeme priradenie, či je písmeno časťou tandemového opakovania, alebo nie.

Ďalšou vlastnosťou takéhoto modelu je, že vygenerovaná dvojica písmeno, stav môže závisieť od predošlých dvojíc - napríklad stavy označené ako *offset=k* sa pozerajú na znak vygenerovaný k krokov dozadu. Tým vieme zvýšiť pravdepodobnosť vygenerovania písmena, ktoré je identické ako to s posunom k pozícií dozadu. Takéto stavy teda generujú tandemové opakovania, zatiaľ čo ostatné stavy generujú zvyšok sekvencie. Pravdepodobnosť prechodov modelujú diskretizáciu dĺžok jednotlivých častí sekvencie.

Pri analýze danej sekvencie hľadáme najpravdepodonejšiu cestu cez HMM, ktorá mohla takúto sekvenciu vygenerovať, čím vlastne každému znaku priradíme jeho anotáciu.



Obr. 2.3: Príklad Skrytého Markovovského Modelu (HMM)
Súčet pravdepodobností prechodov vychádzajúcich z jedného stavu musí byť presne jedna.

Na nájdenie najpravdepodobnejšej cesty cez daný HMM používame *Viterbiho algoritmus* [8]. Časová zložitosť tohto algoritmu je $O(T * S^2)$, kde T predstavuje dĺžku sekvencie, ktorú chceme anotovať a S je počet stavov modelu.

Jedná sa o flexibilnejší model v porovnaní s TRF, najmä vďaka možnosti určenia väčšieho množstva stavov, ktoré ovplyvňujú generovanie aktuálnej pozície. Môžu to byť stavy predstavujúce rôzne k pre offset, alebo napríklad také, kde na vygenerovanie aktuálneho písmena a stavu vplýva viacero predchádzajúcich pozícií.

2.2.2 Vhodnosť použitia modelu pre veľké abecedy

V ďalších častiach práce sa ukáže, že je pre nás dôležité, aby model dokázal pracovať nad veľkými abecedami.

TRF v tomto prípade vyžaduje na hashovanie s $k-ticami$ veľa pamäte a neposkytuje flexibilitu pri špecifikácii podobných znakov. Dokáže hľadať tandemové opakovania len v postupnosti základných dusíkatých báz, a teda len nad pomerne malou abecedou. To sú jeho najväčšie nevýhody pre využitie v našej práci.

Naopak *tantan* má oproti TRF hneď niekoľko výhod. Dokáže hľadať tandemové opakovania aj v postupnosti základných aminokyselín, čo znamená väčšiu abecedu. Na takéto hľadanie používa maticu s názvom BLOSUM62 (2.4). Tá obsahuje skóre pre každú dvojicu aminokyselín, ktoré predstavuje mieru zhody takejto dvojice. Túto maticu nazývame *skórovacia matica* a *tantan* umožňuje používateľovi zadať vlastnú skórovaciu maticu ako vstupný parameter.

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	−1	5																		
Asn	−2	0	6																	
Asp	−2	−2	1	6																
Cys	0	−3	−3	−3	9															
Gln	−1	1	0	0	−3	5														
Glu	−1	0	0	2	−4	2	5													
Gly	0	−2	0	−1	−3	−2	−2	6												
His	−2	0	1	−1	−3	0	0	−2	8											
Ile	−1	−3	−3	−3	−1	−3	−3	−4	−3	4										
Leu	−1	−2	−3	−4	−1	−2	−3	−4	−3	2	4									
Lys	−1	2	0	−1	−3	1	1	−2	−1	−3	−2	5								
Met	−1	−1	−2	−3	−1	0	−2	−3	−2	1	2	−1	5							
Phe	−2	−3	−3	−3	−2	−3	−3	−3	−1	0	0	−3	0	6						
Pro	−1	−2	−2	−1	−3	−1	−1	−2	−2	−3	−3	−1	−2	−4	7					
Ser	1	−1	1	0	−1	0	0	0	−1	−2	−2	0	−1	−2	−1	4				
Thr	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	1	5			
Trp	−3	−3	−4	−4	−2	−2	−3	−2	−2	−3	−2	−3	−1	1	−4	−3	−2	11		
Tyr	−2	−2	−2	−3	−2	−1	−2	−3	2	−1	−1	−2	−1	3	−3	−2	−2	2	7	
Val	0	−3	−3	−3	−1	−2	−2	−3	−3	3	1	−2	1	−1	−2	−2	0	−3	−1	4

Obr. 2.4: Skórovacia matica BLOSUM62

Kapitola 3

Diskretizácia nanopórových signálov

Ako sme ukázali v predchádzajúcej kapitole, hľadanie tandemových opakovaní je dobre známym problémom na reťazcoch. Jednou z možností, ako hľadanie tandemových opakovaní viesť na úrovni nanopórového signálu, bez potreby prekladu signálu do sekvencie, je vytvoriť z nanopórových dát postupnosť znakov (bez potreby basecallingu) a na takýto reťazec aplikovať známe metódy popísane v predošlej kapitole. Nami vytvorený postup sa pri hľadaní tandemových opakovaní skladá z troch fáz:

1. Vyhladenie signálu za účelom čiastočného odstránenia šumu, nepresností merania.
2. Diskretizácia úrovní signálu, ktorá rozdelí úrovne signálu do pevného množstva kategórií a pomocou týchto kategórií preložíme signál do reťazca nad malou abecedou.
3. Použitie existujúceho algoritmu na nájdenie tandemových opakovaní vo výslednom reťazci.

3.1 Diskretizácia signálu

Snaha vyhnúť sa prekladu surového signálu do postupnosti báz pomocou basecallera, nás priviedla k myšlienke diskretizácie signálu, vytvoreniu istého počtu kategórií, ktorým je priradený názov v podobe jedného písmena. Následne, po rozdelení zaznamenaného signálu do príslušných kategórií, vznikne textový reťazec, na ktorý môžeme aplikovať známe metódy hľadania tandemových opakovaní.

Na vytvorenie kategórií budeme hľadať *hranice*, ktoré nám jednoducho určia kategóriu pre príslušné meranie (signálu v časovom bode). Hranice budú tvoriť rastúcu postupnosť čísel a na klasifikovanie merania m nájdeme dvojicu po sebe idúcich hraníc h_i a h_{i+1} takých, že $h_i \leq m < h_{i+1}$. Každá takáto dvojica hraníc ohraničuje jednu kategóriu[1].

[1] v špecifickom prípade bližšie vysvetlíme spôsob určovania

Na takéto rozdelenie sme vytvorili niekoľko rôznych metód:

- kategórie obsahujú rovnaké množstvo meraní,
- kategórie obsahujú rovnaké množstvo úrovní signálu,
- dynamické rozdelenie hraníc pre kategórie.

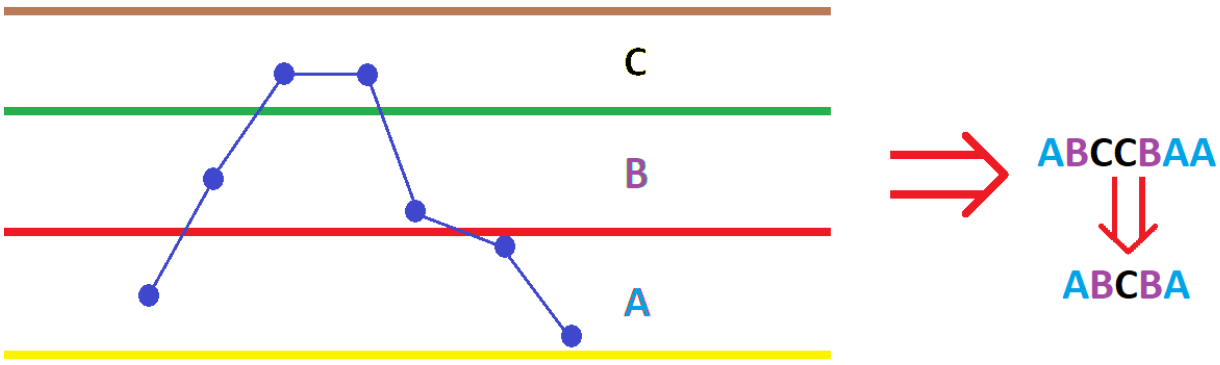
3.1.1 Vytvorenie reťazca, kompresia a oscilácia

Pred tým, než sa budeme venovať jednotlivým metódam, musíme objasniť niektoré vlastnosti nanopórových signálov, na základe ktorých sme navrhli spoločný postup pre vytvorenie reťazca za predpokladu, že už máme určené hranice jednotlivých kategórií.

Vytvorenie a kompresia reťazca

Na základe hraníc kategórií môžeme každému meraniu priradiť jedno písmeno a vytvoriť tak textový reťazec. V tomto reťazci následne zlúčime po sebe idúce rovnaké písmená (obr. 3.1), pretože rýchlosť prechodu nanopórom je nestála a tak niektorým bázam prislúcha viac meraní, ako iným [3], čím docielime, že výsledný reťazec by nemal závisieť od lokálnej rýchlosti prechodu nanopórom.

Týmto krokom však v niektorých prípadoch zlúčime do jedného znaku aj po sebe idúce merania, ktoré zodpovedajú viacerým bázam pôvodnej DNA sekvencie. To je spôsobené tým, že signál závisí od petice báz v kontexte DNA, no rôzne pätice báz môžu mať takmer totožné stredné hodnoty napätia. Preto kompresiou reťazca môžeme vytvoriť falošné pozitívne výsledky.



Obr. 3.1: Príklad vytvorenia a kompresie reťazca

Oscilácia signálu

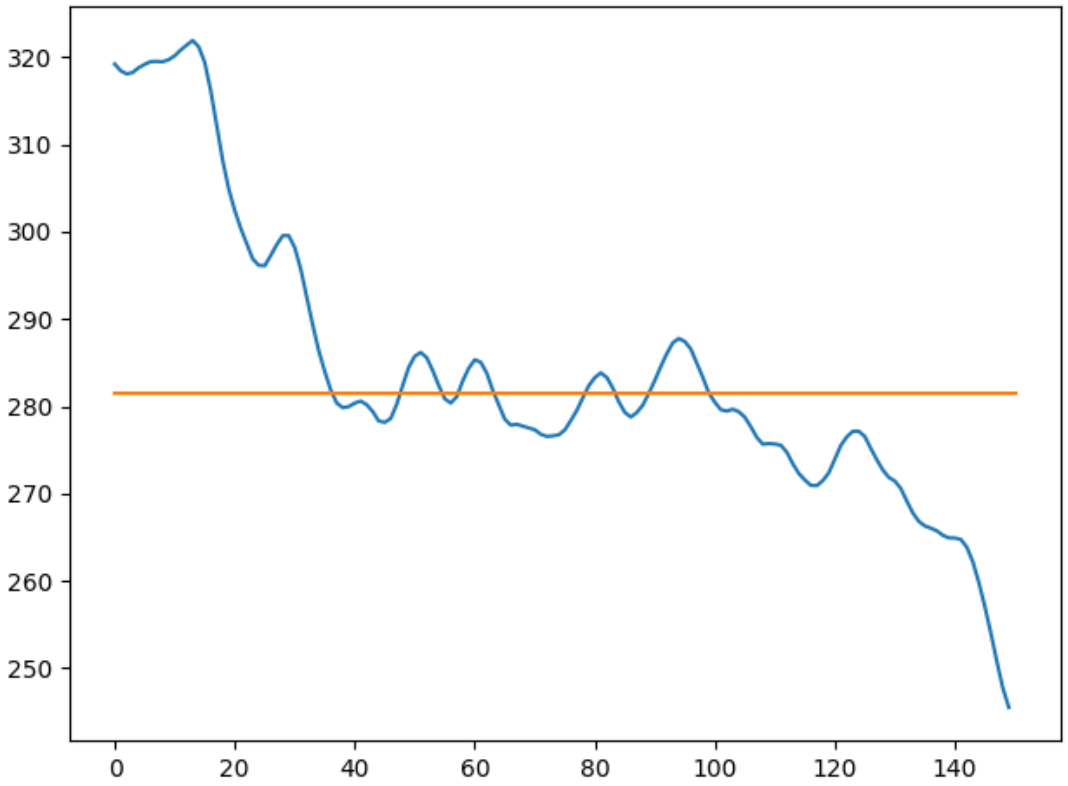
Ďalším problematickým javom pri dikretizácii je *oscilácia* signálu okolo niektorej z hraníc. Jedná sa o úsek signálu, ktorého po sebe idúce merania nadobúdali hodnoty blízke niektorej z jeho hraníc h_i v oboch smeroch, čiže hodnoty menšie aj väčšie ako h_i (obr. 3.2).

V takomto prípade sa teda rôzne inštancie rovnakého tandemového opakovania môžu líšiť počtom znakov zodpovedajúcich tomu istému kontextu DNA, ak jednotlivé merania oscilovali okolo hranice (obr. 1.2).

Napríklad si predstavme pre hranicu $h_i = 100$, ktorá rozdeľuje kategórie A, B a dve takéto oscilujúce časti inštancií rovnakého tandemového opakovania:

$$i_1 = 99, 99, 99, 101, 99, 99, 99, i_2 = 99, 101, 99, 101, 99, 101, 99.$$

V takomto prípade po skomprimovaní dostaneme reťazce $K(i_1) = ABA$ a $K(i_1) = ABABABA$, kde K je vytvorenie a kompresia reťazca.



Obr. 3.2: Príklad oscilácie signálu

3.1.2 Rovnaké množstvo úrovní signálu

Najjednoduchší spôsob určenia hraníc pre n kategórií spočívajúci v nájdení minimálneho a maximálneho napätia vrámci nameraného signálu, zistení rozdielu d medzi mi-

nimom a maximom a na určení rozsahu jednej kategórie c ako podiel d/n. Následne postupnosť hraníc určíme ako:

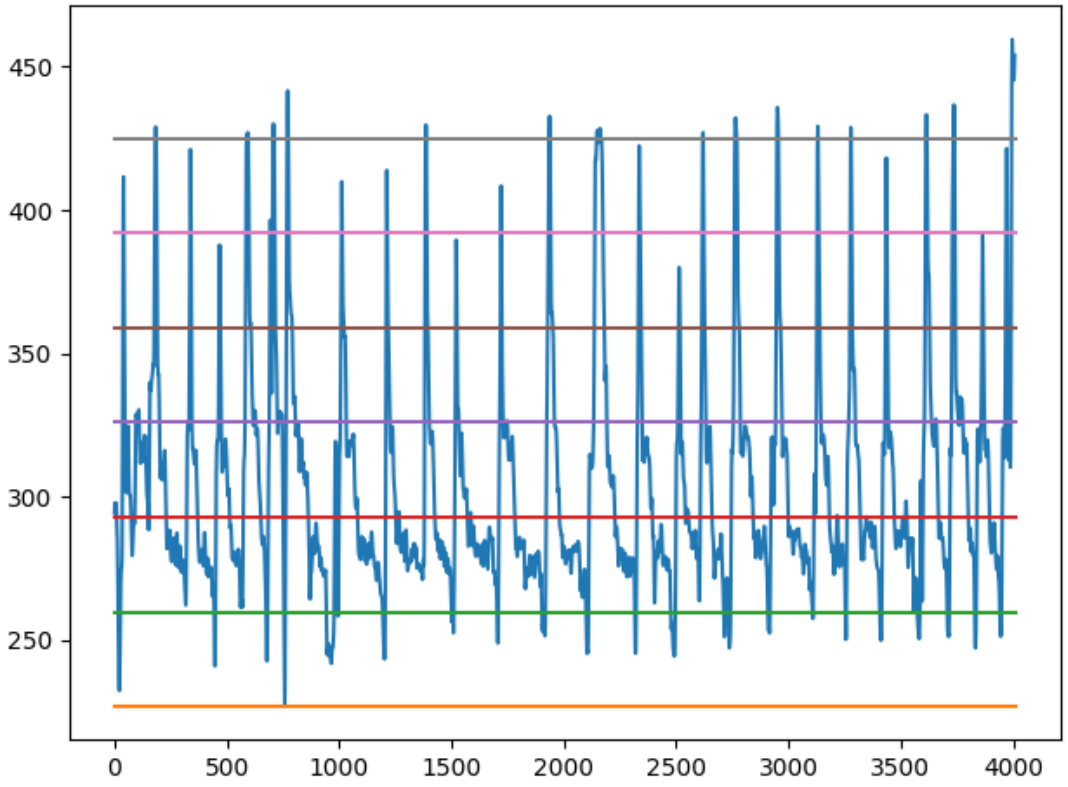
$$min = min + c * 0, min + c * 1, ..., min + c * n = max$$

.

V tomto prípade každá dvojica hraníc určuje jednu kategóriu ako $[h_i, h_{i+1})$ (obr. 3.3). Táto metóda spĺňa podmienku *intervalovej disjunktnosti*.

Intervalová disjunktnosť je vlastnosť postupnosti hraníc, ktorá hovorí, že nenastáva prípad $h_{X1} \geq h_{Y1} \geq h_{X2}$, kde X_1, X_2 sú hranice patriace kategórii X a Y_1 je hranica kategórie Y.

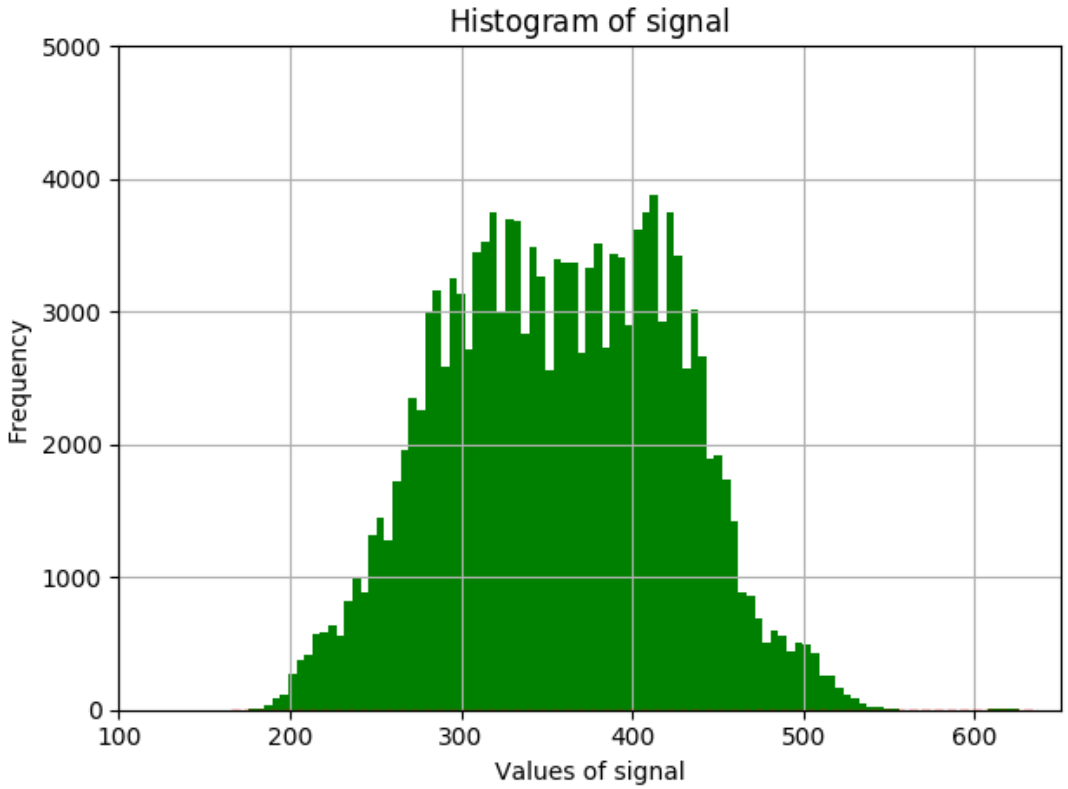
Problémom takéhoto určenia hraníc je neuniformné množstvo meraní v jednotlivých kategóriách, pretože môžu existovať kategórie s malým množstvom meraní. Zároveň vôbec neberieme do úvahy problém oscilácie.



Obr. 3.3: Rozdelenie na sedem kategórií podľa množstva úrovní signálu

3.1.3 Rovnaké množstvo meraní

Druhá navrhovaná metóda adresuje problém s neuniformnými kategóriami. Na obrázku 3.4 môžeme vidieť rozdelenie početnosti signálu na jednotlivých úrovniach.



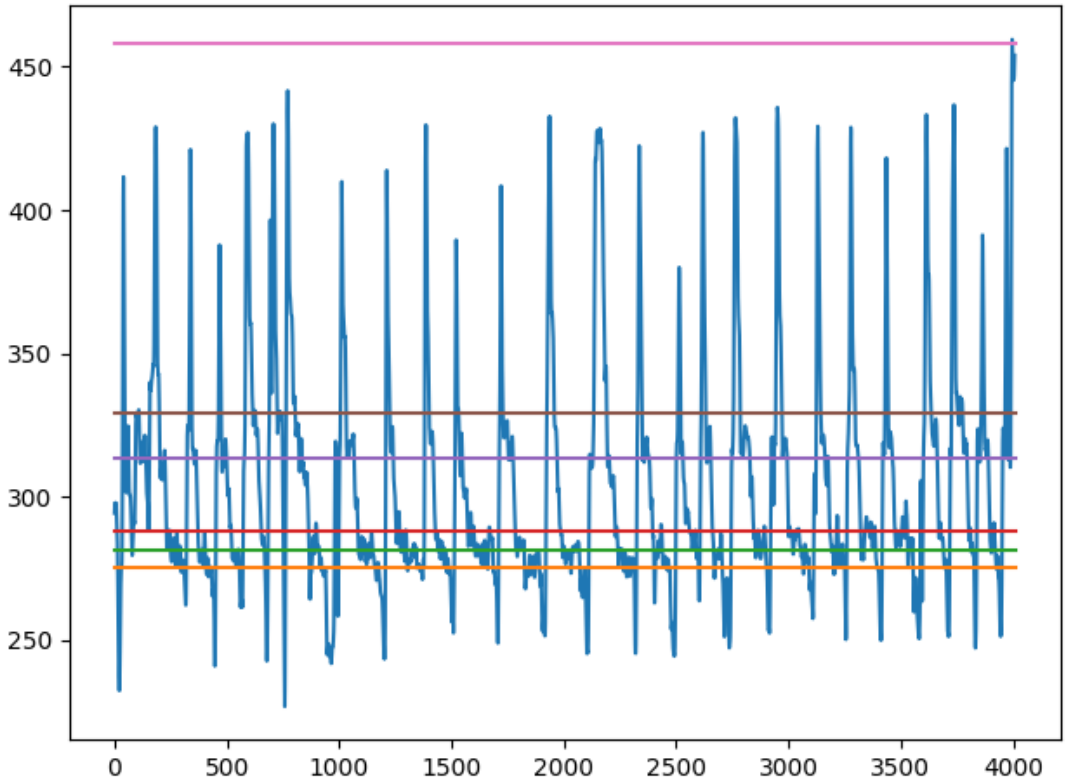
Obr. 3.4: Histogram zobrazujúci početnosť signálu na danej úrovni

Konkrétne, hranice kategórií určíme tak, aby z celkového množstva nameraných hodnôt prislúchalo každej z n kategórií rovnaké množstvo meraní. Na vytvorenie tohoto rozdelenia hraníc najprv spočítame množstvá meraní s rovnakou hodnotu a vytvoríme tak dvojice (úroveň signálu, počet meraní) a následne ich zoradíme podľa veľkosti nameraného napätia. Nakoniec prechádzame zoradenie od najmenšieho po najväčšie napätie a určujeme hranice tak, aby vzniknuté kategórie obsahovali rovnaký počet prvkov.

Niekedy počty prvkov v jednotlivých kategóriách nebudú identické. Napríklad, ak chceme rozdeliť dvojice (100, 1) a (50, 2) do dvoch kategórií s rovnakým počtom prvkov, pre zachovanie korektnosti, čím myslíme intervalovú disjunktnosť, bude jedna kategória obsahovať sto meraní a druhá len päťdesiat.

Dvojice po sebe idúcich hraníc určujú jednu kategóriu. Špeciálnymi prípadmi sú kategórie pre prvú a poslednú hranicu h_1 a h_n. V tomto prípade jednoducho určíme interval príslušnej kategórie ako $(-\infty, h_1]$ a (h_n, ∞) (obr. 3.5).

Problémom je jav oscilácie, ktorý ani v tomto prípade neberieme do úvahy.



Obr. 3.5: Rozdelenie na sedem kategórií podľa množstva meraní

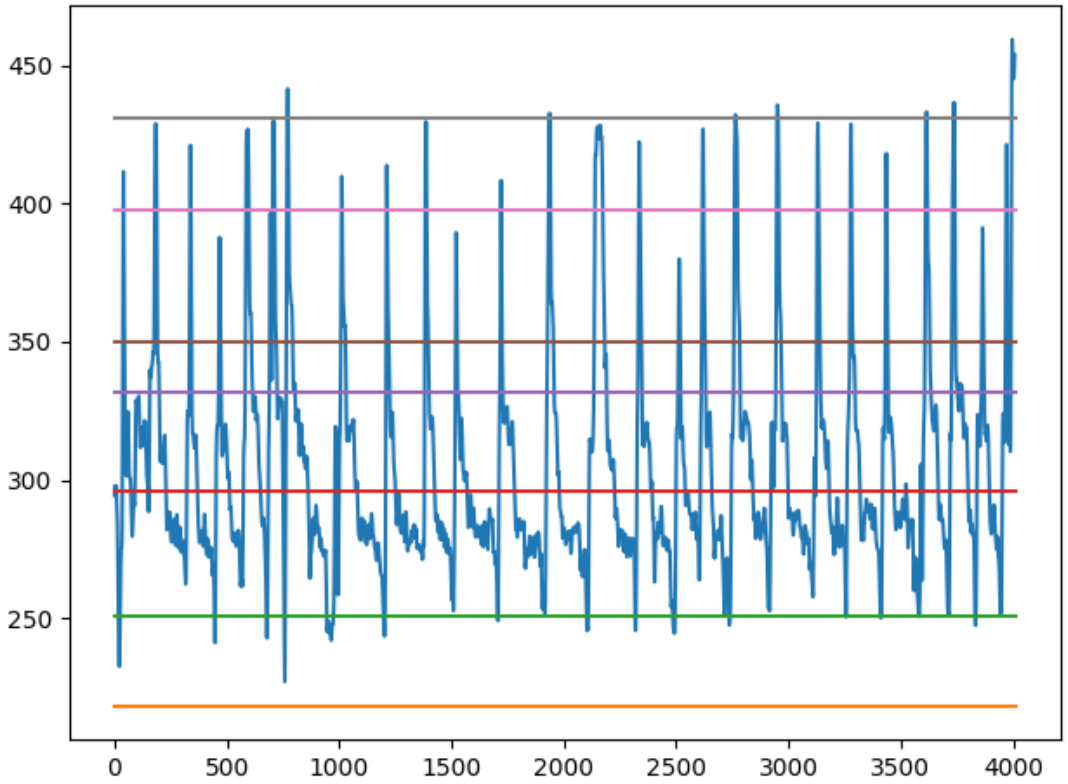
3.1.4 Dynamické rozdelenie hraníc

Spôsob, popísaný v tejto časti, sa snaží optimalizovať rozmiestnenie hraníc tak, aby v čo najväčšej miere eliminoval problém oscilácie, z čoho prihliadajúc na spôsob kompresie popísaného v kapitole 3.1.1, vytvorí čo najkratší reťazec.

Využívame hranice signálov z rozdelenia na kategórie s rovnakým množstvom úrovní signálu, ktoré ďalej upravujeme. Každú hranicu h posúvame v intervale $< h-x, h+x >$ po krokoch k, kde x a k sú parametrami rozdelenia. Pri každej možnosti h_i sme stanovili, že každý signál menší ako h_i bude patriť do jednej kategórie a naopak, väčší alebo rovný ako h_i do druhej. Následne aplikujeme kompresiu prvej úrovne a určíme dĺžku výslednej postupnosti.

Nakoniec vyberieme také h_i, pri ktorom je dĺžka sekvencie najkratšia. Takto postupujeme pre každú hranicu (obr. 3.6). Dynamické rozdelenie hraníc adresuje problém oscilácie, keďže vyberáme polohu hranice, pre ktorú má reťazec najmenšiu dĺžku. V prípade, že by sa problém oscilácie nebral do úvahy, vytvorený reťazec by bol dlhší, napriek možnosti predísť takejto situácii.

Keďže toto rozdelenie je optimalizáciou predošlých dvoch, z hľadiska tvorby reťazca kategórií poskytuje najlepšie výsledky. Z tohto dôvodu náš algoritmus používa dynamické rozdelenie hraníc.



Obr. 3.6: Dynamické rozdelenie hraníc pre desať kategórií

3.1.5 Určenie počtu kategórií

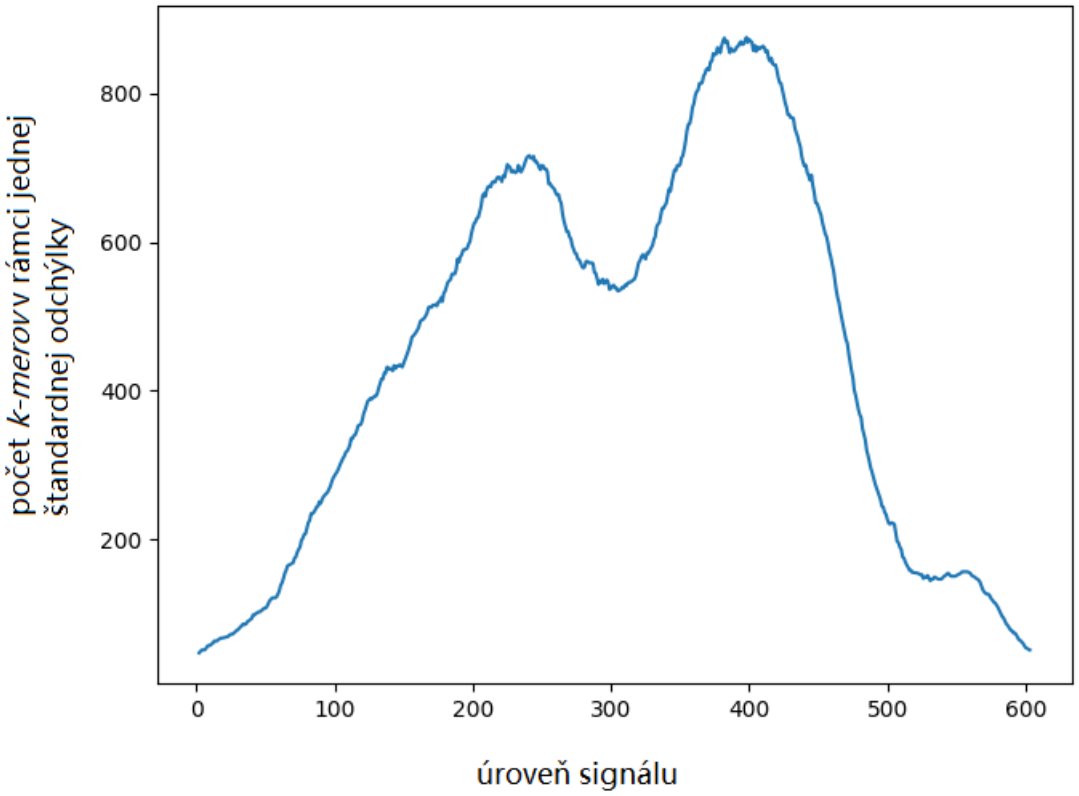
Priemer a štandardná odchýlka merania pre jednotlivé $k-tice$ je známy a je možné ho získať odchýlkou zariadenia. Ako sme popísali v kapitole 1.2, signál u nanopórového sekvenovania priamo závisí od kontextu $k = 5$ báz DNA sekvencie a je zašumený. Pre rôzne kontexty DNA existuje $4^5 = 1024$ rôznych úrovní merania, no rôzne pätice báz môžu mať takmer totožné stredné hodnoty napätia.

Najlepšími hranicami pre naše kategórie by boli *diery v úrovniach napätia*. Diery v úrovniach napätia sú také hodnoty, ktoré nenadobúda žiadna pätica báz a vytvárajú nám prirodzené skupiny signálov. V prípade, že by existovali, umožnilo by nám to efektívnejšie vytvoriť hranice kategórií.

Obrázok 3.7 ukazuje histogram, ktorý ilustruje ako veľmi sa prekrývajú možné namerané hodnoty pre $k-tice$. Hodnoty napätí a štandardnej odchýlky sme preškálovali na interval od nula po šesťsto, vytvorili sme pole f dĺžky šesťsto jeden s hodnotami nula na každej pozícii a následne pre každú päticu s priemerným signálom s_m zvýšili hodnotu na každom indexe i poľa f, pričom $i = s_m - x, s_m - x + 1, ..., s_m + x - 1, s_m + x$ a x je štandardná odchýlka.

Priebeh na obrázku 3.7 ukazuje, že neexistujú diery v úrovniach napätia signálu.

Algoritmy na nájdenie hraníc sme preto implementovali pre ľubovoľný, nami zvo-



Obr. 3.7: Rozdelenie úrovní napätia pre pätice báz (preškálované na rozsah 0 až 600)

lený počet kategórií. Napríklad pre *počet kategórií* = *8*, by bol výsledný reťazec tvorený ôsmimi kategóriami, pomenovanými napr. A, B, C, D, E, F, G. Keďže je takéto delenie jemnejšie, výsledky by mali byť v niektorých sitáciách presnejšie.

3.2 Úprava surového signálu

Výstup sekvenovacieho zariadenia MinION je postupnosť nameraného elektrického napätia počas času. Ako sme už spomínali, ten je často skreslený rôznymi vplyvmi (obr. 1.2), výsledný šum spôsobuje nepresnosti v signále a teda implicitne aj v našej diskredizácii.

Na zmiernenie týchto nežiaducich situácií, akou je napríklad oscilácia, alebo extrémne pozorovania (jedna hodnota napätia, ktorá je signifikantne odlišná od susedných hodnôt, či dokonca všetkých hodnôt v signále), sme implementovali techniky na vyhladzovanie signálu a odstránenie extrémnych pozorovaní.

3.2.1 Centrovaný kĺzavý priemer

V tomto prípade sa nová hodnota S_t vypočíta ako priemer hodnôt z prvkov x_{t-k}, x_{t-k+1}, ... x_{t+k-1}, x_{t+k}. Na výpočet sa využíva vzorec:

$$S_t = (x_{t-k+1} + x_{t-k+2} + ... + x_{t+k-1} + x_{t+k})/2 * k, t = k, ...n - k$$

kde k je parametrom algoritmu a $k \in 0, 1, ...\lceil n/2 - 1 \rceil$.

Rozdiel medzi klasickým a centrovaným kĺzavým priemerom je, že kým klasický počíta novú hodnotu na mieste t ako priemer k hodnôt z miest $t - k + 1$ až t (ktoré nadobúdala originálna časová postupnosť do hodnoty t), naopak centrovaný priemer berie k hodnôt v okolí bodu t (napr. pre k nepárne od pozície $t - \lfloor k/2 \rfloor$ po $t + \lfloor k/2 \rfloor$.

Z tohto dôvodu sme vybrali na implementáciu centrovaný kĺzavý priemer.

3.2.2 Eliminácia extrémnych hodnôt

Napriek vyrovnávacím technikám, výsledná postupnosť obsahovala extrémne pozorovania (obr. 3.8). Pre korektnosť vytvárania kategórií sme však potrebovali takéto záznamy odstrániť.

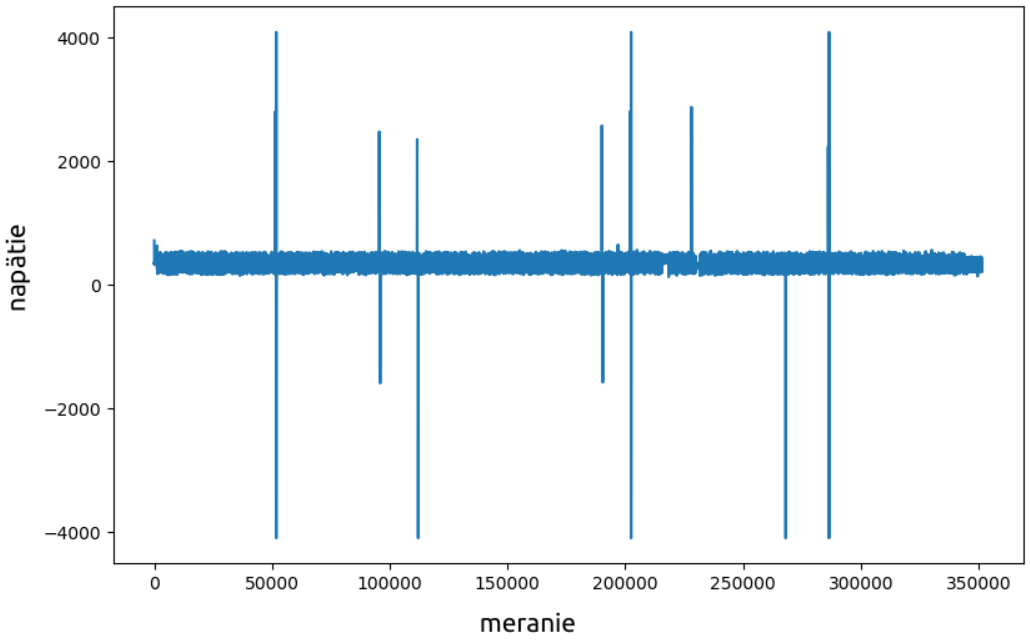
Na tento účel sme použili funkciu knižnice $numpy-percentile$. Tá na vstup dostane číslo p z intervalu $< 0, 100 >$, pole f a na výstup vráti číselnú hodnotu h predstavujúcu takú hranicu, že $p\%$ prvkov poľa f je menších ako h. Funkciu použijeme dvakrát, na nájdenie hornej hranice h_u a dolnej h_d. Následne z postupnosti vylúčime všetky záznamy nepatriace intervalu $< h_d, h_u >$.

Experimentálne sme dolnú hranicu určili na 0.01% a hornú hranicu na úroveň 99.99%. Odstránime tak desatinu promile najväčších a najmenších hodnôt.

3.3 Hodnotiace komponenty

Po zostavení reťazca používame algoritmus *tantan* (kapitola 2.2.1) na nájdenie tandemových opakovaní v tomto reťazci. Dôvod výberu nástroja *tantan* je jeho výhoda oproti TRF v počte kategórií, s ktorými dokáže pracovať. LRF vie pracovať len so štyrmi základnými bázami. Naopak nástroj *tantan* dokáže hľadať tandemové opakovania nielen v DNA postupnosti, ale taktiež v postupnosti základných aminokyselín, ktorých je dvadsať [9].

Algoritmus *tantan* hľadá tandemové opakovania na základe *skóre*, bodového ohodnotenia prislúchajúceho istej časti vstupnej postupnosti. Ak je skóre dostatočne veľké, *tantan* určí danú oblasť ako výskyt tandemových opakovaní. V našom algoritme tieto hodnotiace komponenty upravujeme tak, aby zodpovedali našim potrebám.



Obr. 3.8: Extrémne hodnoty v nameranom signále

3.3.1 Skórovacia matica

V časti 2.2.2 sme objasnili spôsob hľadania tandemových opakovaní v postupnosti aminokyselín s použitím matice BLOSUM62 (obr. 2.4). Tá popisuje vzťahy pre podobnosti aminokyselín, ktoré nemajú žiaden súvis s našou interpretáciou kategórií.

Spomenuli sme však aj možnosť zadať vlastnú skórovaciu maticu používateľom, vďaka čomu sme mohli vytvoriť maticu, ktorá odpovedala našej interpretácii. Dôležité bolo určiť, ako má v súvislosti s naším algoritmom vyzerať. Popisovať by mala vzdialenosť jednotlivých kategórií a teda ich možnú *podobnosť*, čo sa dosiahlo tým, že na diagonálu matice, mieste, kde je skóre v prípade úplnej zhody, vložíme najväčšie skóre a smerom od diagonály bude skóre klesať.

Zároveň sme však museli *podobnosť* obmedziť len na určitý počet susedných kategórií. Nezadefinovaním takého obmedzenia budeme vo výsledku detekovať ako tandemové opakovania aj časti reťazca, ktoré nemajú repetitívny charakter, z dôvodu vysokého skóre dosiahnutého reťazcom. Nakoniec sme sa rozhodli, že podobnosť, v podobe kladného skóre, určíme len pre dve susedné kategórie v jednom aj v druhom smere (ak existujú). To znamená, že ak máme kategóriu A s hranicami h_1, h_2, B s h_2, h_3, C s h_3, h_4 a D s h_4, h_5, tak budeme mať príslušné skóre dvojíc s kategóriou A nasledovné:

$$(A, A) = x, (A, B) = y, (A, C) = z, (A, D) = w, x > y \geq z \geq 0 > w$$

,

čo nazývame ako zhoda, skóre pre prvého a druhého suseda, nezhoda (obr. 3.9).

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	J	Z	X	*
A	5	3	1	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
R	3	5	3	1	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
N	1	3	5	3	1	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
D	-5	1	3	5	3	1	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
C	-5	-5	1	3	5	3	1	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
Q	-5	-5	-5	1	3	5	3	1	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
E	-5	-5	-5	-5	1	3	5	3	1	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
G	-5	-5	-5	-5	-5	1	3	5	3	1	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
H	-5	-5	-5	-5	-5	-5	1	3	5	3	1	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
I	-5	-5	-5	-5	-5	-5	-5	1	3	5	3	1	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
L	-5	-5	-5	-5	-5	-5	-5	-5	1	3	5	3	1	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
K	-5	-5	-5	-5	-5	-5	-5	-5	-5	1	3	5	3	1	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
M	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1	3	5	3	1	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
F	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1	3	5	3	1	-5	-5	-5	-5	-5	-5	-5	-5	-5
P	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1	3	5	3	1	-5	-5	-5	-5	-5	-5	-5	-5
S	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1	3	5	3	1	-5	-5	-5	-5	-5	-5	-5
T	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1	3	5	3	1	-5	-5	-5	-5	-5	-5
W	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1	3	5	3	1	-5	-5	-5	-5	-5
Y	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1	3	5	3	1	-5	-5	-5	-5
V	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1	3	5	3	1	-5	-5	-5
B	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1	3	5	3	1	-5	-5
J	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1	3	5	3	1	-5
Z	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1	3	5	3	1
X	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1	3	5	3
*	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1	3	5

Obr. 3.9: Príklad matice pre hodnoty $zhoda = 5, 1.sused = 3, 2.sused = 1, inak = -5$

Posledným, nemenej dôležitým faktorom bolo určenie hodnôt x, y, z a w. Presné určenie v procese implementácie by mohlo spôsobiť horšiu *senzitivitu*, mieru nájdenia tandemového opakovania, alebo zlú *specificitu*, určenie oblasti ako tandemové opakovanie napriek tomu, že v skutočnosti to nie je pravda (bližšie tieto vlastnosti popíšeme v kapitole 4).

Z tohto dôvodu sme implementovali funkciu, ktorá na vstupe dostane hodnoty x, y, z, w a ako výstup vráti vytvorenú maticu s daným skóre na príslušných miestach. To nám umožnilo v testovaní skúšať viacero možností a vybrať tak len tie najlepšie.

Názvy kategórií

Pri skórovacej matici ešte musíme spomenúť priradenie jednopísmenových názvov jednotlivým kategóriám, pretože úzko súvisia s názvami aminokyselín v matici BLOSUM62.

Nástroj *tantan* očakáva, že matica, ktorú na svojom vstupe dostane, bude vo formáte BLOSUM matice (skóre sa môže, samozrejme, líšiť). Z tohto dôvodu sme sa rozhodli pomenovať kategórie rovnako, ako sú pomenované aminokyseliny. Tých je dvadsať, čo nám umožňuje vytvoriť maximálne dvadsať kategórií. V skutočnosti to môže

byť ešte o jednu viac, pretože BLOSUM definuje skóre aj pre neznámu aminokyselinu.

Ak vo výslednom algoritme vytvárame n kategórií, zoberieme na pomenovanie prvých n aminokyselín z matice BLOSUM (napr. pre $n = 5$, vezmeme A, R, N, D, B)

3.3.2 Ohodnotenie zhôd, nezhôd, inzercií a delécií

Tandemové opakovania nemusia byť úplne identické. Na to, aby sme mohli prehlásiť časť reťazca za oblasť tandemových opakovaní, musia byť jednotlivé inštancie takmer rovnaké, no existuje priestor na variáciu medzi nimi. Tento jav zachytáva druhý hodnotiaci komponent, ktorý popisuje skóre v prípade, kde jedna inštancia obsahuje na niektorom mieste o niekoľko báz viac alebo menej, ako iné, čo nazývame *inzercia* a *delécia*.

Inzercie a delécie môžeme často zaznamenať v prípade oscilujúceho signálu (obr. 3.2). Kým v jednej inštancii opakovania sa oscilácia nevyskytne a celá oblasť je reprezentovaná jedným znakom, v druhej, oscilujúcej inštancii, nám tú istú oblasť bude reprezentovať dlhšia sekvencia alterujúcich znakov zodpovedajúcich susedným kategóriám.

Z tohto dôvodu musíme inzerciu a deléciu penalizovať miernejšie. V kapitole 4 preskúmame viacero možností a vyberieme tú s najlepším výsledkom (z hľadiska senzitivity a specificity) pre náš algoritmus.

3.4 Priebeh algoritmu a zisťovanie parametrov

V predchádzajúcich častiach práce sme opísali jednotlivé komponenty algoritmu a ich implementácie. V tejto časti popíšeme, ako tieto komponenty navzájom spolupracujú a objasníme priebeh algoritmu.

Vstupom je súbor obsahujúci hodnoty napätia v čase, signál zo sekvenátora MinION pre jedno čítanie. Na tieto dáta aplikujeme techniku vyrovnania, čím zmiernime nežiaduci šum. Následne spustíme algoritmus na dynamické určenie počtu kategórií (teda zistenie hraníc). Dĺžka trvania tohto kroku sa môže výrazne líšiť v závislosti od počtu kategórií a dĺžky čítania.

Po tom, čo máme určené hranice, prejdeme celým signálom a každý záznam priradíme do príslušnej kategórie a priebežne vytvárame textový reťazec tvorený názvami kategórií, písmenami. Vykonáme kompresiu opísanú v časti 3.1.1.

Výsledný reťazec spolu s upravenou skórovacou maticou, ohodnotením zhôd, nezhôd, inzercií a delécií sú vstupom pre nástroj *tantan*. Ten spôsobom popísaným v kapitole 2.2.1, detekuje tandemové opakovania, ktorých pozície a konsenzus vzor vráti ako výstup.

3.5 Optimalizačné obmedzenia algoritmu

Za účelom zlepšenia senzitivity a specificity nášho algoritmu, sú prítomné určité optimalizačné obmedzenia, ktoré zamedzujú vznik nesprávnych riešení.

Prvým z nich je stanovenie maximálneho počtu kategórií. V mnohým miestach tejto kapitoly sme spomínali problém oscilácie, ktorý sme sa pokúšali odstrániť dynamickým rozdelením hraníc.

V situácii, kde zvolíme veľký počet kategórií, sa tento problém prejaví vo výraznejšej miere a hľadanie tandemových opakovaní robí veľké chyby. Zároveň je delenie až príliš jemné, čo vytvára veľa falošných detekcií. Uvedieme si príklad. Signál stúpa od hodnoty x_1 po x_2 a vystúpi späť, pohyb sa zopakuje dvakrát a následne sa začne signál správať iným spôsobom (neovplyvňujúcim tento príklad). V prípade, že signály z intervalu $[x_1, x_2]$ spadajú do dvoch kategórií a nie je prítomná žiadna oscilácia, dostaneme podpostupnosť $p = k_1, k_2, k_1, k_2, k_1$. Ak sú tandemové opakovania detekované až od skóre S a máme prísnu skórovaciu schému (za zhodu neprirátame veľký počet bodov), tak skóre celého podreťazca p nemusí byť dostatočné. Pri väčšom počte kategórií pokrývajúcich rovnaký interval, však podreťazec p výrazne narastie a jeho výsledné skóre môže presiahnuť hranicu S.

Práve z tohto dôvodu nám program ohlási veľa falošných tandemových opakovaní. Z sme experimentov určili, že funkčný počet kategórií môže nadobúdať hodnoty z množiny $\{7, 8, 9, 10, 11\}$.

Druhým obmedzením je dĺžka konsenzu vzoru detekovaných opakovaní. Na obrázku 3.10 možno vidieť oblasť signálu, ktorá predstavuje tandemové opakovania. Vidíme, že signál sa ustálil a rozsah jeho celkových hodnôt bol nižší, ako rozsah hodnôt celého vstupného signálu.

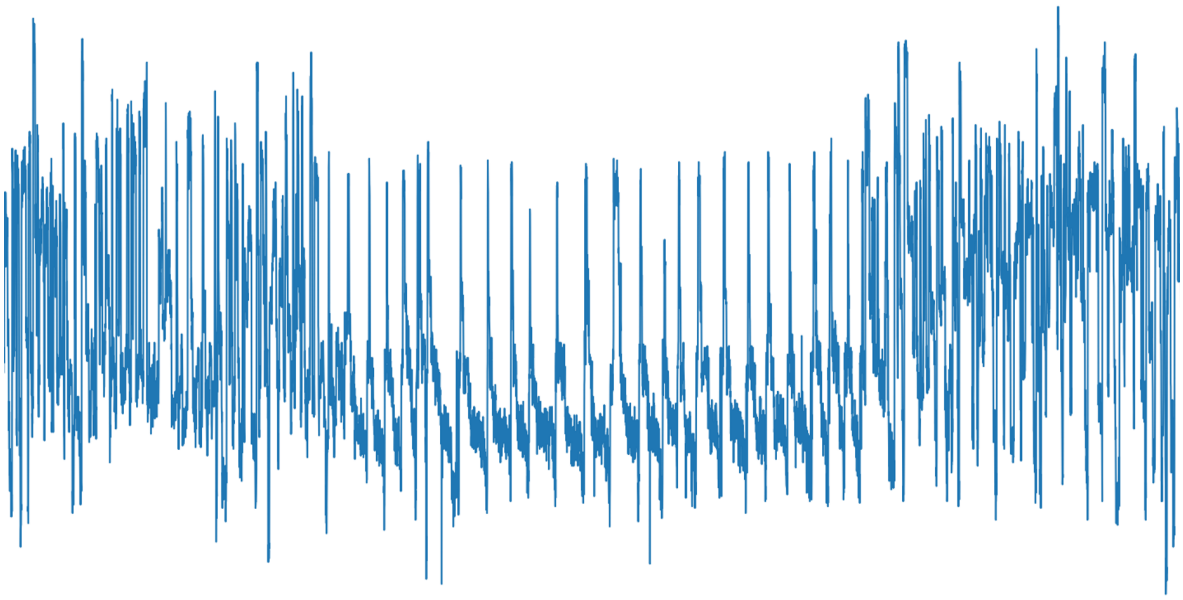
Výsledný konsenzus, vzor celej oblasti, by po úspešnej eliminácii väčšiny oscilácie dynamickým rozdelením hraníc, nemal dosahovať veľkú dĺžku. Experimentálne sme určili maximálnu dĺžku vzoru na dvadsaťpäť.

3.6 Chyby algoritmu

Pri analyzovaní signálu prekladač báz uplatňuje sofistikované techniky prekladu. V mnohých prípadoch vie aspoň čiastočne vyfiltrovať príliš zašumené merania, určiť, či sa rýchlosť prechodu vlákna nanopórom nezmenila [6]. Nedokáže to vždy, ale jeho výsledný reťazec je presnejším obrazom referenčnej sekvencie (skutočnou postupnosťou báz pre skvenovanú vzorku), ako nanopórový signál.

Náš algoritmus signál analyzuje, no len kvôli minimalizovaní oscilácie. Nerobí korekciu iných chýb. To spôsobuje nepresnosť pri hľadaní tandemových opakovaní.

Problémom sú aj opakovania so signálom ustáleným do takej miery, že všetky in-



Obr. 3.10: Ustálený signál oblasti s tandemovými opakovaniami

štancie pokrýva jedna kategória. Vo výslednom skomprimovanom reťazci takúto oblasť reprezentuje len jeden znak, ktorého skóre nedosiahne hranicu pre detekovanie (bližšie tento jav popíšeme v kapitole 4.3.1.

Kapitola 4

Návrh experimentov a výsledky

V tejto kapitole popíšeme spôsob identifikácie najvhodnejších parametrov pre náš algoritmus na hľadanie tandemových opakovaní v nanopórových dátach, implementáciu hľadania takýchto parametrov a porovnanie s hľadaním tandemových opakovaní v reťazci z prekladača báz.

Pri hľadaní parametrov signál upravíme pomocou metód z kapitoly 3.2 a na vytvorenie kategórií použijeme dynamické rozdelenie hraníc z kapitoly 3.1.4. Parametre nášho algoritmu, ktoré sme hľadali, boli:

- počet kategórií, do ktorých signál rozdelíme,
- hodnoty skórovacej matice pre tantan - štyri hodnoty (zhoda, prvý sused, druhý sused, inak),
- skóre začiatku a rozšírenia medzery,
- maximálna dĺžka konsenzu vzoru detekovaných opakovaní.

Tandemové opakovania nájdené naším algoritmom porovnáme s tandemovými opakovaniami, ktoré nájdeme v referencii a zistíme ich správnosť a kvalitu. Správnosť riešenia budeme hodnotiť percentuálne v troch kategóriách, pričom sto percent je najlepšie možné skóre pre každú z nich.

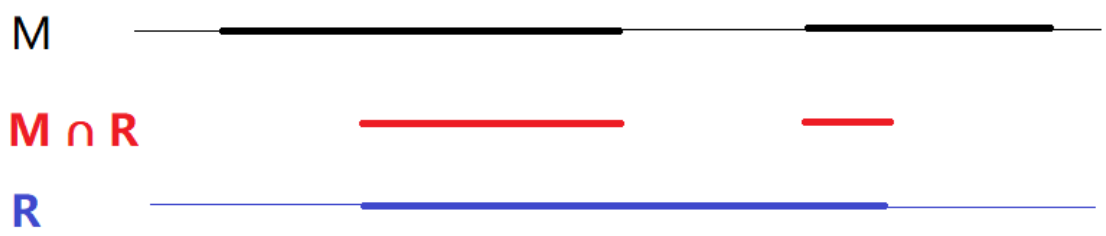
Senzitivita

Senzitivita je miera, ktorá hovorí o tom, koľko percent z tandemových opakovaní nájdených v referencii bolo detekovaných aj naším algoritmom. Pri tejto vlastnosti je naším cieľom zistiť *intervalový prekryv* výstupov.

Označíme intervaly zo signálu, ktoré boli naším algoritmom ohlásené ako tandemové opakovania, ako M a tie nájdené v referencii pomocou *tantanu* ako R (obr. 4.1), senzitivitu vyjadríme ako:

$$S_e = \frac{M \cap R}{R} * 100.$$

Vynásobením hodnotou sto získame percentuálne vyjadrenie senzitivity.



Obr. 4.1: Vizualizácia vytvárania množiny $M \cap R$

Specificita

Ďalším hodnotiacim kritériom je specificita, popisujúca koľko percent z tandemových opakovaní detekovaných naším algoritmom je správne detekovaných vzhľadom na referenciu a aká časť je falošne označená za tandemovú oblasť.

Pri rovnakom značení množín intervalov M, R môžeme túto vlastnosť vyjadriť ako:

$$S_p = \frac{M \cap R}{M} * 100.$$

Aj v tomto prípade nás zaujíma len *intervalový prekryv*.

Zhoda počtu opakovaní

Kým pri predošlých dvoch vlastnostiach nám šlo o intervalový prekryv, v prípade zhody počtu opakovaní pozorujeme zhodu počtu opakovaní v miestach intervalového prekryvu nášho algoritmu a referencie.

Spôsob priradenia tandemových intervalov z referencie (referenčný interval) jednému tandemovému intervalu nájdenému naším algoritmom (náš interval) sme určili nasledovne. Pre každý náš interval vezmeme všetky referenčné intervaly, s ktorými má neprázdny prienik. Vytvoríme tak jeden *súvislý komponent*.

Vo výstupných dátach z *tantan-u* sa nachádza pre každú tandemovú oblasť aj jej konsenzus vzor, počet jeho výskytov, ktorý nemusí byť celé číslo. Tieto hodnoty máme pre každý interval zo súvislého komponentu.

Pre každý referenčný interval, ktorý celý leží v našom intervale, vezmeme všetky jeho výskyty. Naopak, pre referenčné intervaly ležiace v našom intervale len čiastočne, napríklad p percent z celkovej dĺžky, zoberieme len p percent z počtu výskytov takéhoto tandemového opakovania. Počet opakovaní v referenčných intervaloch v jednom komponente je rovný súčtu počtu jednotlivých výskytov, ktoré získame takýmto spôsobom.

Ak si počty opakovaní v našom intervale označíme ako M a súčet počtu opakovaní v príslušných referenčných intervaloch R_S, tak samotnú hodnotu tejto vlastnosti získame ako:

$$I_m = \frac{min(M, R_S)}{max(M, R_S)} * 100$$

Hodnota I_m je vždy z intervalu $< 0, 100 >$.

Trénovacia množina

V tejto kapitole využívame množinu dát, pomocou ktorej zistíme najlepšiu sadu parametrov pre náš algoritmus. Skladá sa z dvoch hlavných častí. Prvou je niekoľko súborov typu $.fast5$ predstavujúcich jednotlivé čítania, obsahujúce signál (zmena napätia počas času), reťazec vytvorený pomocou prekladača báz a informácie o jeho vytváraní (*udalostiach*), teda presné priradenie časti signálu, ktorý vytvoril každú bázu v reťazci. Druhou časťou je *.txt* súbor obsahujúci informáciu o relevantných častiach reťazcov z prekladača báz nachádzajúcich sa v jednotlivých $.fast5$ súboroch vzhľadom na referenciu (bližšia špecifikácia *relevantnej časti* v kapitole 4.2). Základné vlastnosti testovacej množiny sú zobrazené v tabuľke 4.1.

	dĺžka preloženého reťazca	dĺžka signálu	počet báz v tandemových opakovaniach	% báz v tandemových opakovaniach
read 110	145 478	14 252	547	3.84%
read 2849	172 338	14 435	647	4.48%
read 643	159 576	16 971	491	2.89%
read 5432	215 609	20 192	681	3.37%
read 9835	249 753	24 056	846	3.52%
read 9967	294 931	28 789	618	2.15%
read 2096	351 503	33 879	756	2.23%
read 9723	60 696	5 509	465	8.44%
spolu	1 649 884	158 083	5051	3.2%
priemer	206 235	19 760	631	-

Tabuľka 4.1: Základné vlastnosti trénovacej množiny

4.1 Mapovanie diskretizovaného reťazca na signál

Samotným vytvorením textového reťazca pomocou našej diskretizácie a následným vyhľadaním tandemových opakovaní nevieme, kde sa v signále nachádzajú korešpondujúce pozície.

Z tohto dôvodu sme si museli pri vytváraní reťazca pamätať, kde sa v signále nachádza príslušná kategória. Mali sme teda dvojice kategória, pozícia a pri kompresii sme s nimi pracovali, pričom sme komprimovali podľa prvej zložky dvojice. Vo výsledku sme teda vedeli presne určiť miesto v signále, odkiaľ kategória pochádza.

Na určenie časti signálu, ktorý je tandemovým opakovaním, sme vo výsledkoch z *tantan-u* určili polohy začiatku a konca oblasti a v zozname dvojíc predstavujúcom bázy komprimovaného reťazca sme zistili príslušné miesto v signále, na ktoré sa mapuje daná kategória.

4.2 Mapovanie reťazca z prekladača báz na referenciu

Náš diskretizovaný reťazec vieme namapovať na signál. Pomocou informácií o vytváraní preloženej sekvencie (*udalosti*) zo súboru o čítaní vieme presne určiť polohu každej kategórie v tomto reťazci.

Na korektné porovnávanie senzitivity a specificity potrebujeme vedieť, ktorým častiam referencie patrí tandemové opakovanie nájdené naším algoritmom. Dôvodom je nekonštantná rýchlosť prechodu vlákna cez nanopór (kapitola 1.3.1), čo spôsobuje, že rovnako dlhé úseky signálu nemusia reprezentovať rovnaký počet báz pôvodnej DNA.

Našli sme korektný spôsob mapovania reťazca z prekladača báz na referenciu vďaka programu [11], ktorý nájde v reťazci preložených báz časti výrazne zhodné s referenciou (nemusia byť identické z dôvodu chýb a pod. [2]). Tak určíme, ktoré časti reťazca z prekladača báz sú *relevantné*. Hľadanie tandemových opakovaní budeme v signále robiť len na príslušnej relevantnej časti a hodnotenie riešenia vytvoríme len vzhľadom na ňu.

Bijektívne zobrazenie medzi referenciou a preloženým reťazcom vieme dostať pomocou algoritmu *minimap*2 [12], ktorý sa využíva v programe [11]. Malou úpravou sme vo výstupe dostali informáciu o zobrazení. Vďaka nej vieme každú bázu z preloženého reťazca presne namapovať na referenciu. V prípade, že program [11] ohlási viacnásobnú detekciu jednej časti preloženého reťazca v referencii, vyberieme vždy tú s najväčším skóre a zvyšok prekrývajúci sa s touto časťou ďalej nepoužívame.

Vizualizácia procesov z tejto a predchádzajúcej kapitoly 4.1 je znázornená na obrázku 1.2 (avšak namiesto jednosmerných zobrazení máme bijekcie).

4.3 Nájdenie najlepšej sady parametrov a porovnanie metód

Po určení hodnotenia správnosti riešenia a spôsobu jeho získania, sme dané metódy implementovali a začali s testovaním. Skúšali sme každú prípustnú kombináciu pre počet

kategórií, skórovaciu maticu, vznik, rozšírenie medzery a maximálnej dĺžky konsenzus vzoru.

Hodnoty z jednotlivých čítaní sme spojili a získali senzitivitu, specificitu a zhodu v počte opakovaní pre náš algoritmus s danou sadou parametrov. Výsledky mali podobu zobrazenú v tabuľke 4.2, ktorá je výberom 31 104 rôznych kombinácii parametrov, ktoré sme vyskúšali.

Úplná zhoda	1. sused	2. sused	Nezhoda	Začiatok medzery	Rozšírenie medzery	Počet kategórií	Maximálna dĺžka vzoru	Senzitivita (%)	Specificita (%)	Zhoda iterácií (%)
5	4	1	-7	6	5	7	14	3.33	4.69	50.73
7	4	1	-8	6	5	8	8	7.97	2.14	39.19
7	3	2	-9	6	4	10	9	7.71	3.17	36.54
6	3	2	-8	6	5	7	10	21.05	3.34	41.76
7	3	2	-7	6	5	10	18	25.39	2.61	50.26
6	4	3	-9	5	4	7	20	31.38	4.41	33.22
5	3	2	-7	5	4	9	24	33.74	2.45	57.84
6	2	1	-6	6	4	7	22	48.78	3.40	38.04
7	4	3	-7	5	3	7	15	26.05	14.11	50.52
7	4	3	-6	5	3	7	11	26.28	14,73	45.30
priemerné hodnoty								18.6	3.7	39.8
maximálne hodnoty								59.0	14.7	60.9
detekcia v preloženom reťazci								10.55	15.94	54.88

Tabuľka 4.2: Vybrané výsledky hľadania parametrov (vzhľadom na referenciu)

Posledným krokom bol výber najlepšej sady parametrov a porovnanie detekčnej schopnosti nášho algoritmu (senzitivita, specificita, zhoda v počte opakovaní) s hľadaním tandemových opakovaní na reťazci z prekladača báz. Z tohto dôvodu sme potrebovali vybudovať podobný systém mapovania na referenciu, ako pri našom algoritme. Pri priradení nášho reťazca referencii sme náš reťazec mapovali na signál, z neho na preložený reťazec a napokon na referenciu. Z tohto postupu v tomto prípade potrebujeme len posledný krok.

Pre hľadanie tandemových opakovaní v preložených reťazcoch trénovacej množiny v porovnaní s referenciou sme dostali výsledok zobrazený v tabuľke 4.2, ktorý bude podmieňovať výber najlepšej sady parametrov.

Vidíme, že maximálna hodnota specificity nášho algortimu je menšia ako specificita vyhľadania v preloženom reťazci, čo nás viedlo k prvej podmienke výberu, teda čo najvyššej specificite. Následne nás zaujímala senzitivita algoritmu a napokon zhoda v počte opakovaní.

Zároveň sme sledovali aj pomer týchto hodnôt, aby sa nestalo, že pri (relatívne) vysokej specificite aj najväčšia senzitivita bude dosahovať nízke hodnoty. Uveďme si jednoduchý príklad. Specificita jednej sady parametrov je 13% a jej senzitivita je 10%. Druhá sada má specificitu 12%, no jej senzitivita je 20%. Lepšia pre nás bude druhá sada.

Takýmto spôsobom sme vyfiltrovali najlepšie kombinácie parametrov a z nich vybrali tie, ktoré sa vyskytovali v najväčšom počte, avšak s inou hodnotou maximálnej dĺžky konsenzu.

Vybrali sme sady, kde specificita bola väčšia ako 10% a súčet senzitivity a specificity bol väčší ako 40 (na túto hodnotu sme stanovili na základe pozorovaných dát) a zhoda v počte iterácií opakovaní bola väčšia ako 45%. Najlepšou sadou sa ukázala kombinácia parametrov vyznačená v tabuľke 4.2 (zároveň sa ukázalo, že všetky sady, ktoré splnili dané kritériá sú takmer identické, líšili sa len dĺžkou konsenzu vzoru a v jednej hodnote z parametrov ohodnotenia nezhody, začiatku alebo konca medzery, pričom ich výsledky boli takmer zhodné - z tohto dôvodu v kapitole 4.4 testujeme len sadu s najlepšími výsledkami).

Takáto kombinácia parametrov dosahovala **senzitivitu = 26.28%**, **specificitu = 14,73%** a **zhodu v počte opakovaní = 45.30%**. V porovnaní s hľadaním v preloženom reťazci sme mali podobné hodnoty specifcity a zhode počtu opakovaní, no senzitivita bola 2.5-násobne vyššia.

Musíme upozorniť na to, že aj napriek takmer rovnakej specificite nášho algoritmu a specificite hľadania tandemových opakovaní z reťazca vytvoreného prekladačom báz je hodnota specificity (v oboch prípadoch) pomerne nízka.

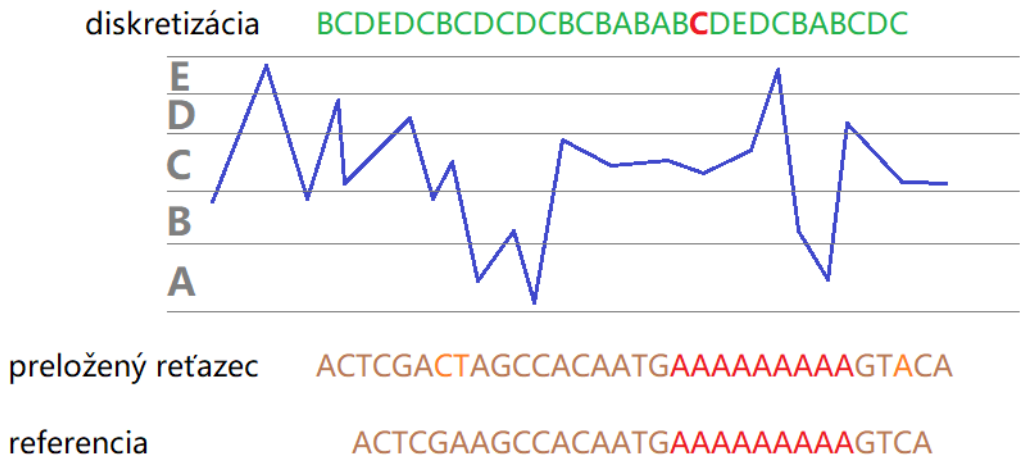
4.3.1 Detekčná schopnosť algoritmu

Pri porovnávaní metód sme narazili na niekoľko opakujúcich sa situácií, ktoré ukazujú silné aj slabé stránky algoritmu, ktorý sme vytvorili.

Zhoršenie detekčnej schopnosti

Na obrázku 4.2 môžeme vidieť, že náš algoritmus nedokáže identifikovať tandemové opakovania, ktoré sú predstavované časťou signálu, ktorý nadobúda hodnoty z *úzkeho* intervalu napätia v porovnaní z celkom (3.6).

Kým komprimácia nám v iných prípadoch pomáha, ako napríklad pri rôznej rýchlosti prechodu vlákna nanopórom (kapitola 3.1.1), v tomto prípade bude tandemová



Obr. 4.2: Príklad signálu nadobúdajúceho hodnoty napätia z úzkeho intervalu Červenou farbou sú vyznačené detekované tandemové opakovania.

oblasť reprezentovaná len jedným písmenom reťazca, ktoré nedosiahne dostatočné skóre na to, aby ho *tantan* ohlásil. Tandemové opakovania vyznačujúce sa takýmto signálom náš program nedokáže detekovať a spôsobujú zhoršenie senzitivity algoritmu.

Výrazným problémom pre náš algoritmus je aj situácia, kedy signál nejaví repetitívny charakter, no na príslušnom mieste v referencii sa nachádza tandemové opakovanie (obr. 4.3). V prípade, že chceme takéto tandemové opakovania detekovať, musíme použiť parametre, ktoré spôsobia veľké množstvo nesprávnych označení signálu za repetitívnu oblasť.

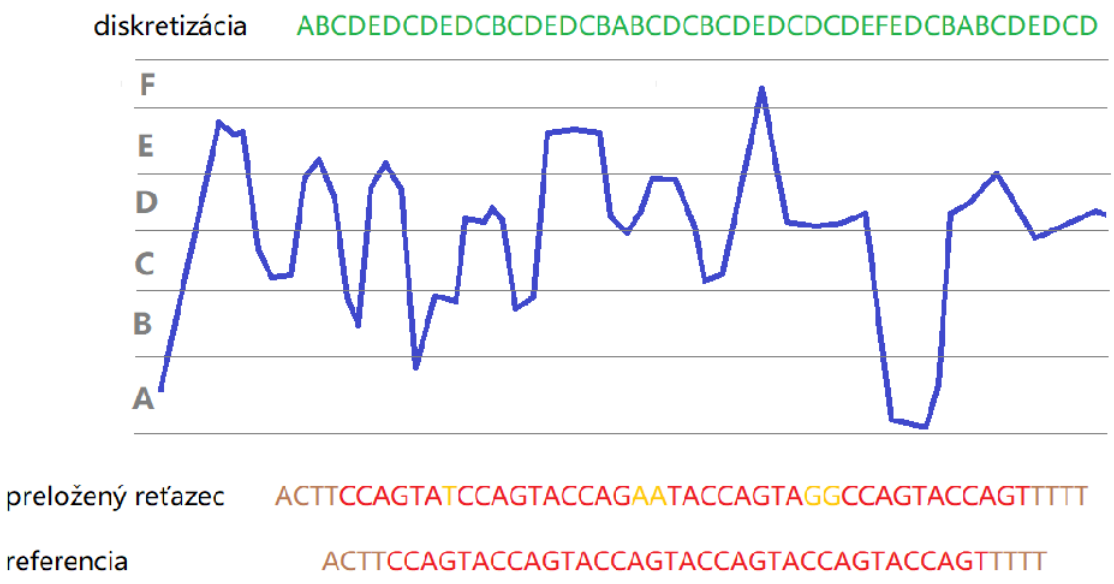
Zlepšenie detekčnej schopnosti

Naopak, existovali časti podobné ako na obrázku 4.4, pre ktoré signál jasne vykazoval repetitívne charakteristiky, avšak hľadanie v postupnosti báz z prekladača báz relevantnej podľa referencie nebolo úspešné pri ich detekcii.

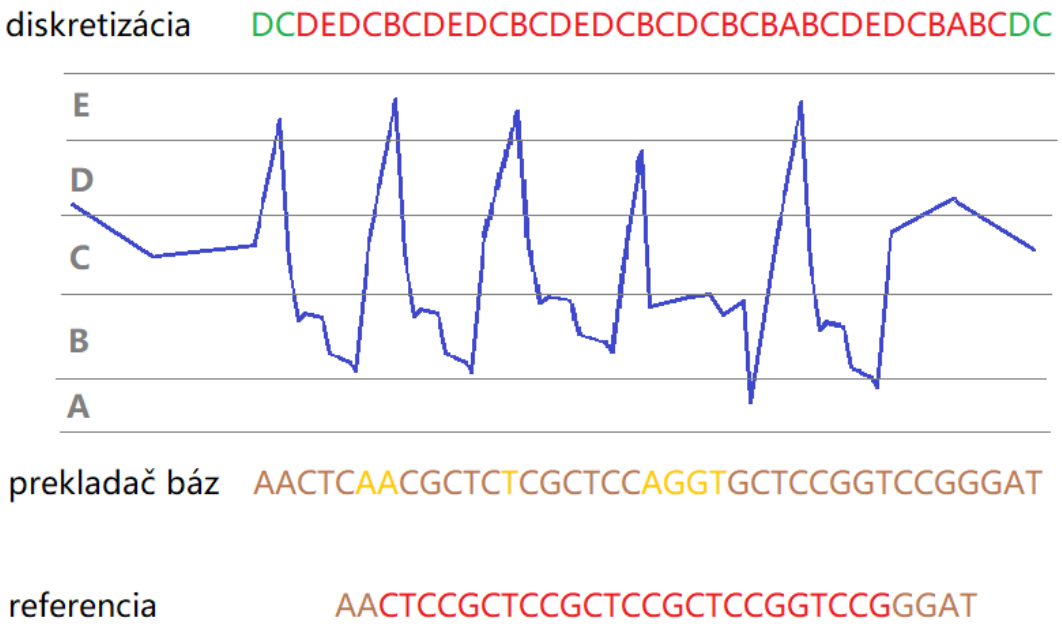
Samozrejme, takéto opakovania môžu byť po preložení zo signálu do báz variabilné do takej miery, že sú *tantan-om* nedetekovateľné. Naša práca sa ale zameriava práve na hľadanie repetícií v signále, a tak považujeme takéto označenie za správne. Tento jav zvyšuje specificitu algoritmu.

4.4 Finálne overenie funkcionality algoritmu

Na záver sme otestovali náš algoritmus na nezávislej testovacej sade čítaní a ich referencie, čo slúžilo na overenie správnosti algoritmu. Nezávislá množina čítaní rovnako ako trénovacia množina pozostávala z 51 čítaní v podobe $.fast5$ súborov, ich referencie a



Obr. 4.3: Príklad signálu s nerepetitívnym charakterom
Červenou farbou sú vyznačené detekované tandemové opakovania.



Obr. 4.4: Príklad zlepšenia detekčnej schopnosti
Červenou farbou sú vyznačené detekované tandemové opakovania.

súboru s informáciou o relevantných častiach preložených reťazcov jednotlivých čítaní vzhľadom na referenciu.

Algoritmus sme spúšťali so sadou parametrov, ktoré sme vyhodnotili ako najlepšie

v kapitole 4.3. Výsledky sú znázornené v tabuľke 4.3. Môžeme usúdiť, že riešenie, ktoré nájdeme, silno závisí od konkrétnej vzorky a kvality nameraného signálu. Zároveň máme potvrdenú existenciu prípadov, kde oproti vyhľadávaniu tandemových opakovaní v reťazci z prekladača báz je miera senzitivity rovnaká a specificita je niekoľkonásobne lepšia.

metóda	senzitivita	specificita	zhoda počtu opakovaní
hľadanie naším algoritmom v signále	12.16%	34.23%	24.38%
štandardný algoritmus v preloženom reťazci	6,25%	19,75%	44.08%

Tabuľka 4.3: Výsledky finálneho testovania

Záver

Cieľom tejto práce bolo navrhnúť nový algoritmus na identifikáciu tandemových opakovaní v surových dátach zo sekvenačného prístroja MinION, ktorý by pri detekovaní nepotreboval signál preložiť na postupnosť báz pomocou prekladača báz. Takúto identifikáciu sme chceli dosiahnuť na základe pozorovateľných repetitívnych oblastí signálu.

Navrhli a implementovali sme algoritmus na hľadanie tandemových opakovaní v signále. Úspešnosť nášho algoritmu je v prípade jasne repetitívneho signálu vysoká, v niektorých prípadoch lepšia ako hľadanie v reťazci báz vytvoreného prekladačom báz. Naopak v prípadoch, kde nie je pozorovateľný repetitívny charakter signálu spôsobený štandardnou odchýlkou merania a vplyvmi prostredia, je jeho detekčná schopnosť malá. Napriek tomu nepovažujeme výsledný algoritmus za neúspešný, pretože technikou diskretizácie opakovania predstavované nepravidelným signálom detekujeme s ťažkosťami.

Algoritmus, ktorý sme navrhli, by bolo možné v budúcnosti rozšíriť dvoma spôsobmi. Prvým z nich je korekcia detekcií, ktorá by v časti označenej našim algoritmom ako tandemové opakovania spustila analýzu na odstránenie nesprávnych detekcií. Druhou ja dodatočné hľadanie repetitívnych častí s nepravidelným signálom. V oboch prípadoch by sme chceli využiť metódy strojového učenia alebo neurónových sietí.

Hlavným prínosom tejto bakalárskej práce je algoritmus, ktorý nám v pomerne krátkom čase nájde časti signálu predstavujúce tandemové opakovania. Algoritmus má pre rôzne čítania porovnateľnú specifcitu a lepšiu senzitivitu ako hľadanie tandemových opakovaní v reťazci báz z prekladača báz.

Literatúra

[1] Dávid Barbora. Multiple alignment and visualization of nanopore sequencing signals. Bakalárska práca, Univerzita Komenského v Bratislave, 2018.

[2] Eduard Batmenjdin. Identifikácia variantov v dátach nanopórového sekvenovania. Bakalárska práca, Univerzita Komenského v Bratislave, 2018.

[3] Benson and Gary. Tandem repeats finder: a program to analyze dna sequences, 1999.

[4] Vladimír Boža, Broňa Brejová, and Tomáš Vinař. Deepnano: Deep recurrent neural networks for base calling in minion nanopore reads. *PLoS One*, 12(6):e0178751, 2017.

[5] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[6] Yaniv Erlich, Partha P Mitra, W Richard McCombie, Gregory J Hannon, et al. Alta-cyclic: a self-optimizing base caller for next-generation sequencing. *Nature methods*, 5(8):679, 2008.

[7] Luzitano Brandão Ferreira, Celso Teixeira Mendes, Cláudia Emília Vieira Wiezel, Marcelo Rizzatti Luizon, and Aguinaldo Luiz Simões. Genomic ancestry of a sample population from the state of sao paulo, brazil. *American Journal of Human Biology: The Official Journal of the Human Biology Association*, 18(5):702–705, 2006.

[8] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[9] Martin C Frith. A new repeat-masking method enables specific detection of homologous sequences. *Nucleic acids research*, 39(4):e23–e23, 2010.

[10] Ian P Gent, Christopher Jefferson, and Ian Miguel. Minion: A fast scalable constraint solver. In *ECAI*, volume 141, pages 98–102, 2006.

[11] Andrej Korman. *Nástroj na mapovanie basecallového reťazca na referenciu*, 2019 (accessed April 30, 2019).

[12] Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 2018.

[13] Sharma, Deepak, Issac, Biju, G. P. S., and Ramaswamy. Spectral repeat finder (srf): identification of repetitive sequences using fourier transformation, 2004.

[14] Ryan R Wick, Louise M Judd, and Kathryn E Holt. Comparison of oxford nanopore basecalling tools. *January. https://doi. org*, 10, 2018.

[15] Ryan R Wick, Louise M Judd, and Kathryn E Holt. Performance of neural network basecalling tools for oxford nanopore sequencing. *bioRxiv*, page 543439, 2019.