VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ
DEPARTMENT OF BIOMEDICAL ENGINEERING

PŘÍMÉ SESTAVOVÁNÍ GENOMOVÝCH SIGNÁLŮ ZE SEKVENACE NANOPÓREM
DIRECT ASSEMBLY OF GENOME SIGNALS FROM NANOPORE SEQUENCING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Inna Karmazinová
AUTHOR

VEDOUCÍ PRÁCE — Mgr. Ing. Karel Sedlář
SUPERVISOR

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Biomedicínská technika a bioinformatika**
Ústav biomedicínského inženýrství

Studentka: Inna Karmazinová ***ID:*** 186665
Ročník: 3 ***Akademický rok:*** 2017/18

NÁZEV TÉMATU:

Přímé sestavování genomových signálů ze sekvenace nanopórem

POKYNY PRO VYPRACOVÁNÍ:

1) Zpracujte literární rešerši o základních přístupech využívaných pro sestrojení genomového sestavení (assembly) ve vztahu k různým sekvenačním technologiím. 2) Seznamte se s nativním signálovým formátem platformy Oxford Nanopore (squiggle space) a prozkoumejte základní charakteristiku signálů. 3) Navrhněte metodu pro vyhledávání překryvů mezi jednotlivými signály. 4) Metodu implementujte v libovolně zvoleném jazyce do podoby uživatelsky přívětivého balíčku. 5) Na vhodném testovacím souboru statisticky vyhodnoťte úspěšnost metody. 6) Metodu porovnejte se stávajícími technikami pracujícími se znakovými sekvencemi.

DOPORUČENÁ LITERATURA:

[1] UTTURKAR, Sagar M. et al. Evaluation and validation of de novo and hybrid assembly techniques to derive high-quality genome sequences. Bioinformatics. 2014. 30 (19): s. 2709-2716.

[2] LOOSE, Matthew, Sunir MALLA a Michael STOUT. Real-time selective sequencing using nanopore technology. Nature Methods. 2016, 13(9), 751-754.

Termín zadání: 5.2.2018 ***Termín odevzdání:*** 25.5.2018

Vedoucí práce: Mgr. Ing. Karel Sedlář
Konzultant:

prof. Ing. Ivo Provazník, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:
Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Bakalářská práce se zabývá hledáním překryvů mezi signály ze sekvenace nanopórem z přístroje MinION verze R9. Teoretická část se věnuje metodám sestavování genomu – hladovým algoritmům, dále grafovým overlap-layout-consensus (OLC) a de Bruijnovým grafům. Novým přístupem je sekvenování nanopórem a sestavování genomu z těchto dat. Oxford Nanopore Technologies představili přístroj MinION, který zjednodušuje sekvenování s využitím změny proudu při průchodu DNA nanopórem. Chybovost přístroje je stále vysoká, problém nastává při překladu signálu do nukleotidů. S využitím rozdílového signálu, případně i dynamického borcení časové osy, je možné nalézt překryvy mezi jednotlivými signály. Sestavování genomu s využitím původního signálu z MinION, by mohlo zlepšit přesnost metody.

Klíčová slova

de Bruijnovy grafy; dynamické borcení časové osy; hladový algoritmus; sestavování genomu; MinION; nanopór; overlap-layout-consensus; rozdílový signál

Abstract

The aim of this bachelor thesis is to search for overlaps between signals from nanopore sequencing using MinION device version R9. The theoretical part deals with methods used for genome assembly - greedy algorithm, overlap-layout-consensus (OLC) and de Bruijn graphs. Oxford Nanopore Technologies introduced the MinION device, which simplifies sequencing using the current change, which occurs while the DNA is passing through the nanopore. The error rate of the device is still high, the accuracy problem occurs during the base-calling. Using the difference signal, possibly also the dynamic time warping, it is possible to find overlaps between the individual signals. Signal analysis and genome assembly using the MinION signal could provide better accuracy.

Keywords

de Bruijn graphs; difference signal; dynamic time warping; genome assembly; greedy algorithm; MinION; nanopore; overlap-layout-consensus

Bibliografická citace:

KARMAZINOVÁ, I. *Přímé sestavování genomových signálů ze sekvenace nanopórem*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 62s. Vedoucí práce: Mgr. Ing. Karel Sedlář.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Přímé sestavování genomových signálů ze sekvenace nanopórem jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestně právních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne **25. května 2018** ……

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Mgr. Ing. Karlu Sedlářovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne **25. května 2018** ……

podpis autora

Obsah

Seznam obrázků

Seznam tabulek

ÚVOD

Sekvenování a sestavování genomu představuje důležitý krok pro poznání nových organismů, studium genetické variability nebo analýzu genomu jedinců. Různé platformy pro sekvenování DNA poskytují čtení o různých délkách od desítek bp až po tisíce bp. Genomy organismů jsou však mnohonásobně větší, lidský genom obsahuje 3 miliardy bp, což představuje problém pro sestavování genomu z osekvenovaných čtení. Mezi požadavky na sestavovací softwary patří především finanční a časová náročnost a také přesnost sekvenace. Sekvenační platformy tedy poskytují čtení, která byla vytvořena namnožením a rozdělením genomu na mnohem kratší úseky. Cílem sestavování genomu je tato čtení pomocí různých algoritmů na základě jejich překryvů spojit do výsledné DNA sekvence.

Stále se vyvíjí nové metody pro sekvenování DNA, což vede i k vyvíjení nových přístupů k sestavování genomu. Mezi třetí generaci sekvenování patří přístroj od Oxford Nanopore Technologies (ONT) MinION. Oproti jiným platformám, je MinION levnější, rychlejší a jeho použití je mnohem jednodušší. Při průchodu molekuly DNA nanopórem vznikají charakteristické změny elektrického proudu. Tyto změny proudu jsou převedeny na krátké nukleotidové sekvence, které jsou dále analyzovány.

Největší úskalí sekvenace pomocí přístroje MinION je přeložení signálu do nukleotidové sekvence. Úseky signálu jsou nejdříve překládány do nukleotidů a poté následuje hledání překryvů mezi sekvencemi pomocí metod pro genomová sestavení. Analýza signálu a hledání překryvů mezi signály, místo hledání překryvů mezi nukleotidy, může poskytnout nové informace a zlepšit přesnost metody.

Cílem práce je navrhnout metodu pro hledání překryvu mezi čteními. K hledání překryvů lze využít rozdílového signálu, kdy je známo, že rozdílový signál je nejmenší u nejvíce podobných signálů. Na základě těchto poznatků lze vypočítat rozdílový signál všech signálů navzájem. Reálná data získaná sekvenací přístrojem MinION mohou mít až stovky GB, z toho důvodu je potřeba vytvořit omezující podmínky, aby nedocházelo k porovnávání všech vzorků všech signálů navzájem, což by bylo časově neefektivní. Toho lze dosáhnout například porovnáváním pouze takových signálů, které obsahují alespoň jeden vzorek, který je shodný s prvním vzorkem jiného signálu. Dva signály, jejichž rozdílový signál od shodného vzorku bude nejmenší, budou označeny jako navazující se vzájemným překryvem.

Signály lze před samotným výpočtem rozdílového signálu zarovnat pomocí dynamického borcení časové osy, což je metoda, která nelineárně zarovnává dva časově závislé signály. Toto zarovnání by mohlo mít vliv na přesnost metody a snížit rozdílový signál. Soubory získané ze sekvenace obsahují také již přeložené nukleotidové

sekvence. Po nalezení překryvů mezi signály lze ověřit správnost přiřazení pomocí lokálního zarovnání známých nukleotidových sekvencí. Skóre lokálního zarovnání je nejvyšší u nejvíce podobných sekvencí.

1 METODY SESTAVOVÁNÍ GENOMU

Sestavování genomu je hierarchická struktura, která mapuje sekvenační data k zpětné rekonstrukci pro vytvoření původní sekvence DNA. Seskupuje čtení do kontigů a kontigy následně do scaffoldů. Kontigy poskytují vícenásobná zarovnání sekvencí čtení a konsensuální sekvenci. Scaffoldy definují pořadí kontigů, orientaci a velikost mezer mezi kontigy. Scaffoldy mohou být jednoduché nebo mohou tvořit síť. Nejpoužívanější formát pro sestavování je FASTA formát, kde je kontig konsensuální sekvence reprezentován písmeny: A, C, G, T a dalšími písmeny předem definovaného významu označující například purinové nebo pyrimidinové báze apod. [1].

Čtecí úseky jsou mnohem kratší než vůbec nejmenší genomy, a to představuje problém pro sestavovací softwary. Repetice v genomu, které mají dokonalou shodu, mohou být nerozlišitelné, a to především v případě, kdy jsou repetitivní úseky delší než samotná čtení [2].

Při sestavování genomu rozlišujeme *de novo* metody, které rekonstruují genom, který nelze přirovnat k žádnému dosud osekvenovanému organismu, a komparativní metody, které v průběhu sestavování využívají sekvence již osekvenovaných blízkých organismů. Tato metoda má tedy omezené využití pouze pro malý počet genomů, jejichž referenční sekvence jsou předem dostupné a známé. Dále se využívá při výzkumu variability genomů a pro modelové organismy. *De novo* sestavování je matematicky mnohem náročnější proces [3]. Během komparativního sestavování se podle referenčního genomu zarovnávají pouze nová čtení, části, které se výrazně liší od referenčního genomu, však musí být rekonstruovány již pomocí *de novo* technik.

De novo metody lze rozdělit do tří hlavních skupin – Overlap-layout-consensus (OLC) je metoda založena na grafech překryvů, de Bruijn grafové metody (DBG) pracují s k-mer grafy a poslední skupina popisuje hladový algoritmus, který je znakovou a nejjednodušší metodou [1]. Většina softwarů navržena pro zpracování dlouhých čtení získaných ze Sangerova sekvenování, Roche 454 sekvenování nebo PacBio používají k sestavování překrývající se shody – OLC [3]. OLC metoda však není vhodná pro krátká čtení, právě kvůli časové náročnosti, která roste kvadraticky s počtem čtení, i přes to však některé sestavovací softwary používají OLC i pro krátká čtení a to např. Edena [4] a SGA [5], problém s časovou náročnosti je řešen pomocí vhodného indexování, jež tyto metody používají [6].

Graf překryvů, který je využíván u OLC metody, představuje čtení a jejich překryvy. Překryvy musí být předem vypočítané pomocí zarovnání. Výpočet překryvů je časově velmi náročný. Graf je tedy složen z vrcholů, které reprezentují čtení a z hran, které představují překryvy. Graf může mít různé prvky nebo atributy, pro

odlišení 5’ a 3’ konce čtení, dopředného vlákna a reverzního komplementu, délku čtení a délku a typ překryvu. Cesta grafu je potenciálním kontigem a může být převedena na sekvenci.

Metody pro *de novo* sestavování krátkých sekvencí lze rozdělit do dvou skupin – prodlužovací metody a de Bruijn grafové algoritmy [7]. Mezi prodlužovací metody patří SSAKE [8] a JR-Assembler [6]. Tyto metody jsou výpočetně velmi účinné, ale jsou citlivé na sekvenační chyby, opakující oblasti a na četné nukleotidové polymorfismy [6].

Nejpoužívanější metody pro sestavování krátkých čtení jsou založeny na de Bruijnových grafech, kde jsou čtení rozdělena na k-mery (podřetězce sekvence o délce k), které poté tvoří vrcholy grafu a jsou spojeny pří sdílení (k-1)-mer [2]. Často používané sestavovací softwary jako SOAPdenovo [9], ALLPATHS-LG [10], ABySS [11] a Velvet [12] vychází právě z de Bruijnových grafických algoritmů. Existují také hybridní sestavovací postupy, jako například Atlas [13], Ray [14] a MaSuRCa [15], kombinující výhody různých algoritmů a používající data z více sekvenačních technologií [16].

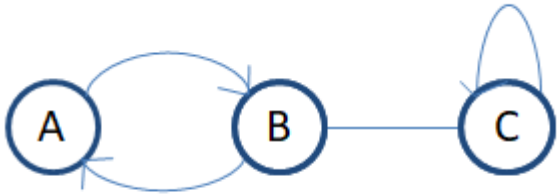
1.1 Teorie grafů

Graf je uspořádanou dvojící vrcholů a hran. Jedná se o matematickou strukturu znázorňující vztahy mezi vrcholy a hranami [17]. Počet vrcholů a hran je konečný. Každá hrana musí být definována dvěma vrcholy. Graf lze tedy popsat dle vztahu [17]:

$$G = (V, H), \tag{1.1}$$

kde V označuje vrchol a H označuje hranu.

Hrany mohou být neorientované nebo orientované, kdy lze stanovit počáteční a koncový vrchol. Hrany mohou být také násobné nebo vytvářet smyčku (Obrázek 1. 1). Jednoduchý graf, je takový graf, jenž neobsahuje žádné smyčky, a dva různé vrcholy jsou spojeny nanejvýš jednou hranou [18]. Stupeň vrcholu vypovídá o tom, kolik hran s daným vrcholem interaguje, v případě jednoduchého grafu, jehož počet vrcholů je N, je maximální stupeň vrcholu N-1. Pokud je dosaženo maximálního stupně u každého vrcholu, pak je graf úplný. [17]



Obrázek 1. 1: Znázornění násobné hrany a smyčky

Mezi vrcholy A a B je násobná hrana, vycházející z vrcholu A do vrcholu B a z vrcholu B zpět do vrcholu A. Vrchol C obsahuje smyčku, hrana vychází i končí ve stejném vrcholu C.

Střídavá posloupnost vrcholů a hran se nazývá sled. Sled, kde se nevyskytuje žádný vrchol více než jednou, se označuje jako cesta. Cesta, která prochází všemi vrcholy, se označuje jako hamiltonovská cesta nebo kružnice, pokud cesta začíná i končí ve stejném vrcholu [18]. Sled, kde se nevyskytuje žádná hrana více než jednou, se označuje jako tah. Tah, který prochází všemi hranami, se nazývá eulerovský. Souvislý graf je takový, pro jehož každé dva vrcholy existuje cesta. Komponenta souvislosti je maximální souvislý podgraf grafu G. [17]

Souvislý graf, jehož vrcholy jsou sudého stupně kromě maximálně dvou, je nazýván jako eulerovský [18]. Vychází-li tah ze stejného vrcholu, ve kterém následně končí, pak jsou všechny vrcholy sudé. V případě, že tah začíná a končí v různých vrcholech, pak je právě první a poslední vrchol lichého stupně.

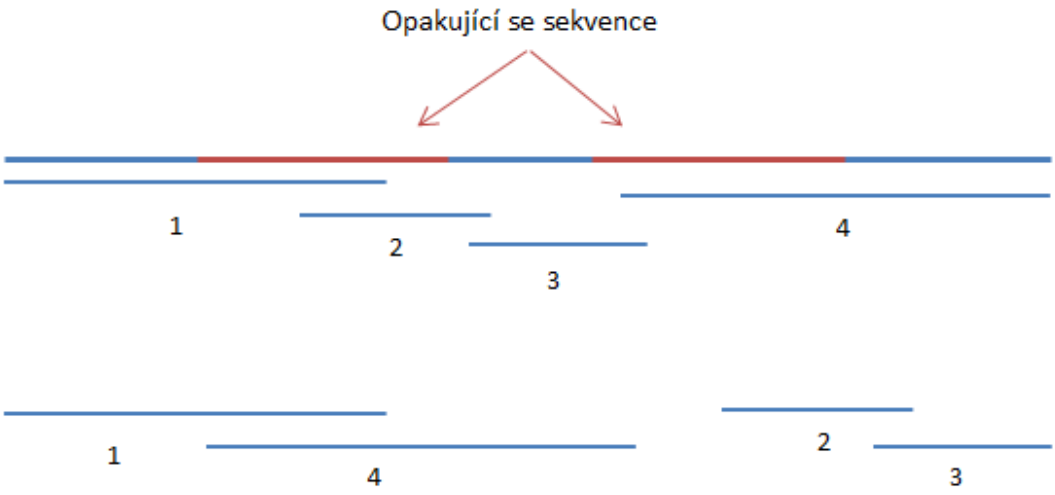
Strom je souvislý graf, který neobsahuje žádnou kružnici a každé dva vrcholy spojuje pouze jedna cesta, počet hran je tedy vždy o jednu nižší, než je jeho počet vrcholů [18]. Jeho využití je široké.

Hamiltonovskou cestu, kružnici či cyklus lze nalézt po splnění určitých podmínek, graf musí být souvislý, při hledání hamiltonovské kružnice, musí být každý vrchol alespoň druhého stupně. Hledání hamiltonovské cesty, kružnice nebo cyklu se řadí mezi obtížné úlohy neboli NP-úplné úlohy [19].

1.2 Hladový algoritmus

Hladový algoritmus představuje nejjednodušší cestu k sestavení genomu. Jednotlivá čtení jsou spojována do kontigů podle nejvyššího dosaženého skóre překryvu. Překryvy jsou určovány podle shody koncové části jednoho čtení (sufix) se začátkem druhého čtení (prefix), kvalita překryvu závisí na několika parametrech, a to na délce překryvu, pokrytí a na procentuálním zastoupení shodných párů bází v překrývajícím se úseku [3].

Sestavovací softwary používají pro výpočet překryvů variaci na Smith-Watermanův algoritmus [20]. Tento algoritmus vždy upřednostňuje překryvy s nejvyšším dosaženým skóre, což může mít za následek špatně seřazené opakující se (repetitivní) sekvence (Obrázek 1. 2). Hladový algoritmus je tak vhodný pro použití pouze u jednoduchých genomů, jejichž repetitivní úseky nejsou delší než čtecí rámec [21].



Obrázek 1. 2: Příklad nesprávného přiřazení čtení

Repetitivní sekvence jsou vyobrazeny červeně. Čtení 1 a 4 jsou spojeny, díky opakující se sekvenci mají nejvyšší skóre, i když spolu nesousedí a neměly by být přiřazeny k sobě.

Jiným postupem při použití hladového algoritmu, který je využíván u novějších sestavovacích softwarů, je vybrání čtení, které bude tvořit začátek kontigu. Tento kontig je následně opakovaně prodlužován čteními, která se překrývají s 3' koncem tohoto kontigu do té doby, než není možné přiřadit žádný další úsek [3]. Stejný proces je aplikován i v případě opačného směru, kdy se využívá reverzního komplementu původního kontigu a nová čtení jsou přiřazovány k 5' konci. Čtení jsou hodnocena podle hloubky pokrytí [22] nebo kombinací různých faktorů. Aby se během sestavování předešlo chybnému přiřazení čtení, prodlužování je ukončeno ve chvíli, kdy například dvě nebo více sekvencí, které by mohly řetězec prodloužit, se navzájem nepřekrývají [3].

1.3 Overlap-Layout-Consensus

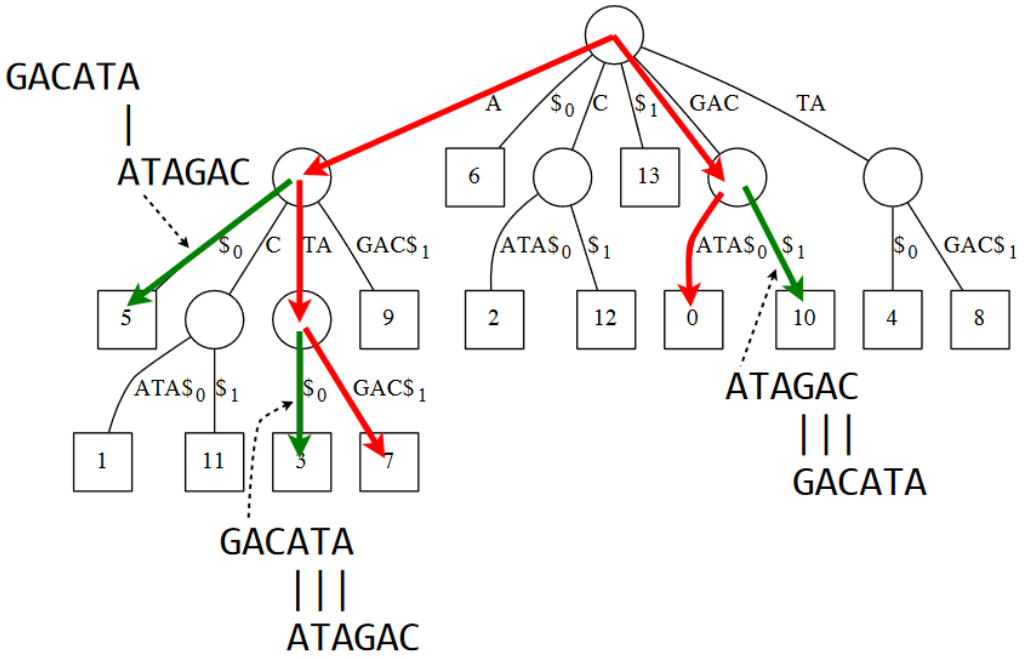
Tato metoda, jak již z názvu vyplývá, sestává ze 3 kroků. V prvním kroku probíhá sestavení grafu překryvů. Všechna čtení jsou navzájem porovnána, jako v případě

hladového algoritmu, pro zjištění jejich vzájemných překryvů. Poté je z těchto překryvů sestrojen graf. Každé čtení je v grafu vyobrazeno jako vrchol a dva vrcholy jsou spojeny hranou v případě, že mezi těmito čteními byl nalezen jejich vzájemný překryv. Poté následuje vytvoření konsensuální sekvence z jednotlivých kontigů. [3]

1.3.1 Překryvy (Overlap)

Při hledání překryvu mezi čteními, musí být porovnána všechna čtení navzájem. Tento proces je výpočetně náročný. Každé čtení je v grafu znázorněno vrcholem a překryvy jednotlivých čtení jsou vyobrazeny jako hrany. Z jednoho vrcholu může být vedeno i více hran.

Hledání překryvů je nejpomalejší část celého algoritmu a pro nalezení může být použito více metod [23]. Jednou z metod je „naivní“ metoda, která bere v potaz pouze absolutní shodu. Porovnává všechna čtení navzájem a je tak velmi pomalá. Další metodou pro hledání překryvů jsou sufixové stromy, kde je vytvořen graf znázorňující všechny překryvy začátku jedné sekvence *x* s koncovou částí druhé sekvence *y* (Obrázek 1. 3). Každý vnitřní vrchol odpovídá prefixu jedné sekvence a sufixu druhé sekvence. Každá subsekvence ve stromě je ukončena znakem $. Každá cesta označuje sufix a hodnoty ve vrcholu označují začínající pozici korespondujícího sufixu. [24]



Obrázek 1. 3: Sufixový strom [24]

Graf znázorňující sufixový strom pro sekvence GACATA a ATAGAC. Červené šipky následují cestu původních dvou sekvencí. Zelené šipky znázorňují překryvy sekvencí. Čísla označují pozici sufixu v původní sekvenci.

Tato metoda je poměrně rychlá, ale nezaznamenává mezery a neshody. Celková délka je $N = dn$, kde d je počet čtení a n je jejich délka. Čas potřebný k sestavení generalizovaného sufixového stromu je $O(N)$, čas potřebný k projití červených cest je $O(N)$, k nalezení překryvu je potřeba čas $O(a)$, a tedy celkový čas je $O(N+a)$, kde a označuje počet překrývajících se párů. [24].

Poslední metodou pro hledání překryvů, je dynamické programování, kdy se využívá globálního zarovnání a skórovací matice. Vytvořená matice zarovnání je definována [24]:

$$D[i,j] = min\begin{cases} D[i-1,j] + s(x[i-1],-) \\ D[i,j-1] + s(-,y[j-1]) \\ D[i-1,j-1] + s(x[i-1],y[j-1]) \end{cases}, \quad (1.2)$$

kde D je nově vytvořená matice zarovnání a s je předem daná skórovací matice (Obrázek 1. 4), i a j jsou indexy matice.

	A	C	G	T	-
A	0	4	2	4	6
C	4	0	4	2	6
G	2	4	0	4	6
T	4	2	4	0	6
-	6	6	6	6	0

Obrázek 1. 4: Příklad skórovací matice

Shodné báze mají nejmenší ohodnocení, zatímco inzerce nebo delece mají ohodnocení nejvyšší.

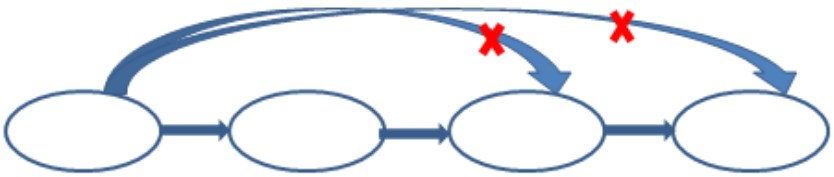
V případě, že provádíme zarovnání dvou sekvencí A a B s tím, že sekvence A označuje řádky matice a sekvence B zase sloupce matice, první sloupec nově vytvořené matice je poté vyplněn 0, první řádek zase ∞. Tato inicializace zajišťuje zarovnání sufixu A k prefixu B, 0 v prvním sloupci zajišťuje, že je možný jakýkoliv sufix sekvence A a ∞ v prvním řádku zase zajišťuje to, že musí být prefixem B [24]. Zpětná cesta začíná na nejnižší hodnotě posledního řádku. Dynamické programování je oproti sufixovému stromu výhodnější z důvodu započítávání mezer a neshod, je však

časově náročnější. Celkový počet překryvů, které je potřeba otestovat je $O(d^2)$, velikost vytvořené matice je $O(n^2)$ a celkově $O(d^2n^2)=O(N^2)$, v porovnání se sufixovým stromem $O(N+a)$ [24].

Po nalezení překryvu je tedy sestrojen graf čtení a jejich překryvů [25], který je v této fázi nepřehledný se spoustou hran, které musí být odstraněny, pro nalezení vhodné cesty.

1.3.2 Layout

Během layout fáze dochází ke zjednodušení a upravení grafu tak, aby bylo možné spojit čtení do kontigů a najít cestu skrz celý graf. Ideálním případem je, pokud je nalezena cesta, která vede přes všechny vyobrazené vrcholy právě jednou – hamiltonovská cesta [23]. V této fázi dochází také k odstranění hran, které mohou být odvozeny z jiných a jsou tak nadbytečné (Obrázek 1. 5) [24].

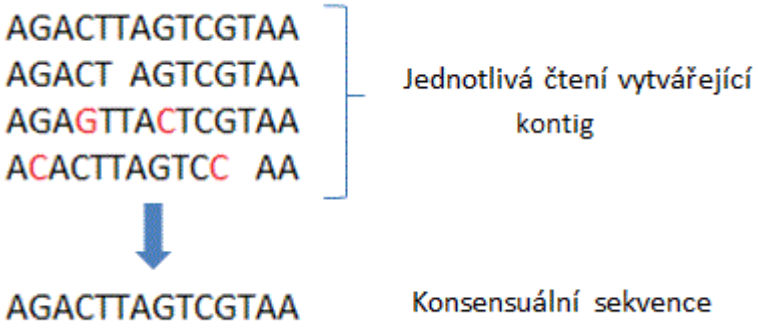


Obrázek 1. 5: Odstranění nadbytečných hran

Po odstranění přebytečných hran, již lze rozeznat a spojit jednotlivé kontigy. Kontigy, které nevytváří další větve, jsou vynechány [2]. Graf musí mít také prvky nebo atributy k rozeznání 5' a 3' konce sekvencí, vedoucího a komplementárního řetězce, délky čtení a délky a typu překryvu [2].

1.3.3 Konsensuální sekvence (Consensus)

V poslední fázi je ze všech překrývajících se sekvencí, které tvoří kontigy, vytvořena konsensuální sekvence (Obrázek 1. 6).



Obrázek 1. 6: Vytvoření konsensuální sekvence

Čtení jsou zarovnána a na každé pozici nové sekvence je vybrán nukleotid, který se ve všech čteních na stejné pozici vyskytuje nejčastěji.

1.3.4 Lander Watermanův model

Lander-Watermanův model byl prvním matematickým modelem pro sestavování sekvencí [23]. Je založen na ideálních sekvenačních datech. V případě, že délka překryvu dvou čtení je větší než T, jsou tato čtení sloučena do kontigu [26]. Tento proces pokračuje do té doby, dokud nejsou k dispozici žádná další čtení, která by mohla být přiřazena.

Pomocí tohoto modelu, tak lze vypočítat výsledný počet kontigů. Počet kontigů závisí na délce čtení, překryvu T, hloubce sekvenování a velikosti genomu. Podle Landera a Watermana, při rozdělování DNA do fragmentů např. při shot-gun sekvenování, podléhají tyto fragmenty DNA Poissonovu rozdělení [26]. Pomocí těchto poznatků, lze stanovit pokrytí, kterého lze dosáhnout [26]:

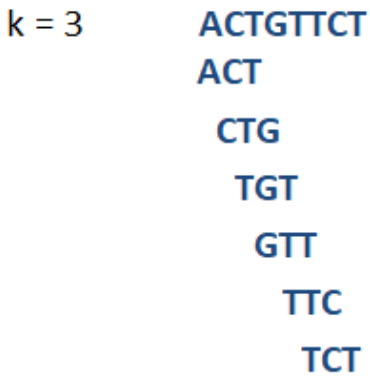
$$C = \frac{L \cdot N}{G}, \tag{1.3}$$

kde C je pokrytí, L je délka čtení, N je počet čtení a G je délka genomu.

1.4 De Bruijn

Metoda založena na de Bruijnových grafech rozděluje čtení na kratší úseky délky k (k-mery) [3]. K-mer graf je jedním z de Bruijnových grafů, v případě těchto grafů, není potřeba porovnávat překryvy všech čtení navzájem, což je výhodou oproti OLC metodě a hladovému algoritmu. Vrcholy představují subsekvence délky *k-1* a hrany jsou reprezentovány sekvencemi délky k [27]. Každý k-mer je rozdělen na levý a pravý (k-1)-mer. Každý k-mer se v genomu vyskytuje, a tak je potřeba najít takový tah skrz celý graf, který využívá všech hran v grafu – eulerovský tah.

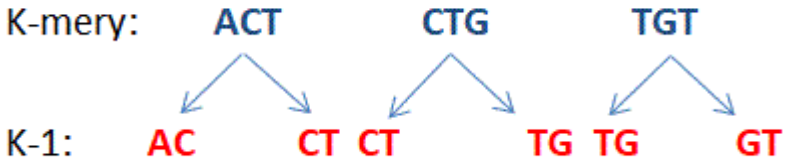
Tuto metodu lze také rozdělit do 3 kroků, kde v prvním kroku dochází k rozdělení čtení na jednotlivé k-mery o délce *k* (Obrázek 1. 7).



Obrázek 1. 7: Rozdělení čtení na k-mery

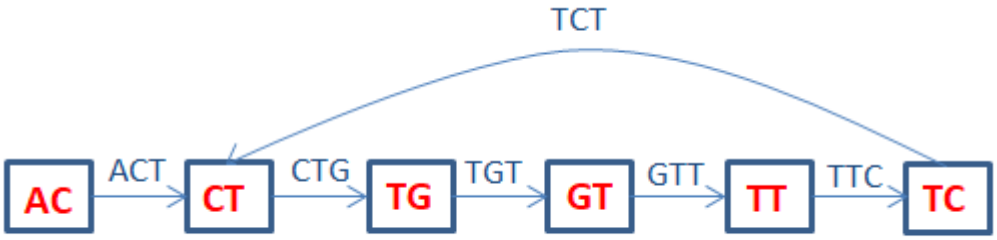
Čtení délky 8 je rozděleno na 6 k-mer délky 3, obsahující všechny nukleotidy z původní sekvence.

V dalším kroku je každý k-mer rozdělen na pravý a levý (k-1)-mer (Obrázek 1. 8). Každý (k -1)-mer je v grafu vyobrazen pouze jednou. Velikost *k* je volena podle stupně pokrytí. V případě, že je pokrytí nízké, je voleno i nízké *k*, pro vytvoření většího počtu překrývajících se čtení a vyšší citlivost, pokud je naopak pokrytí vysoké, je voleno i vysoké *k* [28].



Obrázek 1. 8: Rozdělení k-mer na (k-1)-mery

V grafu jsou jako vrcholy vyobrazeny (k-1)-mery a hrany jako odpovídající k-mery (Obrázek 1. 9). Dva vrcholy jsou spojeny hranou v případě, že jejich (k-1)-mery se navzájem překrývají délkou *k-2* [3]. Nalezení eulerovského tahu spočívá v tom, že tah prochází každou hranou grafu (každým k-merem) právě jednou.



Obrázek 1. 9: Příklad de Bruijnova grafu

Výhodou oproti metodě OLC je, že není potřeba vypočítávat překryvy všech čtení navzájem, pro nalezení eulerovské cesty existují efektivní algoritmy, zatímco u OLC je nalezení hamiltonovské cesty, kde musí cesta procházet každým vrcholem právě jednou, mnohem náročnější [28]. Problémem je však to, že v grafu lze najít exponenciální počet odlišných eulerovských tahů, což představuje mnoho způsobů, jak genom rekonstruovat [3]. Rozdělením čtení na k-mery jsou ztraceny informace o pozici v genomu a repetitivní části genomu nemusí být rozpoznány a rekonstruovány tak dobře, jako v případě OLC. Metoda de Bruijnových grafů vyžaduje vysokou paměť, celý graf, který byl vytvořen, musí být uložen v paměti pro následné sestavování, s délkou a počtem čtení rostou požadavky na paměť, a tak většina sestavovacích softwarů využívající de Bruijnovy grafy nezvládají sestavovat velké genomy při jejich omezené paměti [6]. Software ABySS dokázal sestavit lidský genom za 87 hodin [11], software SOAPdenovo sestavil lidský genom za 40 hodin [29].

Tabulka 1. 1: Přehled metod pro sestavování genomu

	Hladový algoritmus	**OLC**	**De Bruijn**
Typ	Znakový	Grafový	Grafový
Software	SSAKE, VCAKE	Celera, Newbler, Edena	Velvet, SOAPdenovo, Euler
Vhodné genomy	Malé genomy	Malé genomy	Velké a komplexní genomy
Délka čtení	Krátká čtení	Krátká i dlouhá čtení (lepší pro dlouhá)	Krátká čtení
Časová náročnost	-	$O(N+a)/O(N^2)$	O(N)

2 OXFORD NANOPORE

Oxford Nanopore patří mezi třetí generaci sekvenačních metod. Přístroj MinION je v současné době nejmenší přenosný DNA sekvenátor, který je schopen sekvenovat jednotlivé molekuly DNA. Jeho výhodou je snížení nákladů na sekvenaci, a použití v podstatě kdekoliv s minimální přípravou, MinION využívá DNA fragmenty bez nutnosti amplifikace [30]. Je schopen sekvenovat dlouhé fragmenty DNA bez ztráty kvality. Aby bylo možné sekvenovat oba řetězce DNA, byla vytvořena knihovna, zahrnující přípravu před sekvenováním z dvouřetězcové DNA (dsDNA) s protokolem podobným tomu, který se využívá pro krátká čtení u platformy druhé generace. [31]

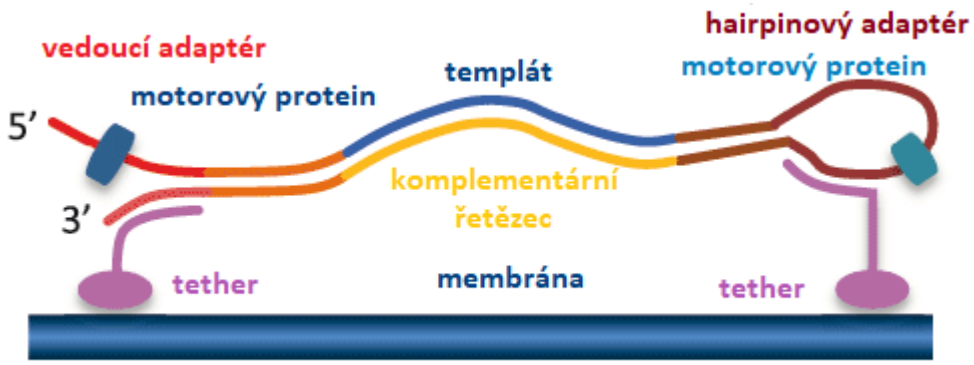
2.1 MinION

MinION je momentálně nejmenším zařízením na sekvenaci DNA (Obrázek 2. 1). Lze ho připojit přímo do počítače pomocí USB portu. Sekvenace probíhá pomocí softwaru MinKNOW [32]. MinION dokáže klasifikovat DNA báze na základě změny elektrického proudu, který je generován ve chvíli, kdy řetězec DNA prochází nanopórem přístroje [33].

Obrázek 2. 1: Přístroj MinION [34]

2.1.1 Sekvenace verze R7

Chemie R7 s knihovnou - SQK-MAP005 a SQK-MAP005.1 používá dva různé adaptéry, které se vážou na DNA řetězec. První, vedoucí adaptér, je složen ze dvou oligopeptidů s částečnou komplementaritou utvářející tvar písmene Y. Druhý adaptér, hairpinový adaptér, je jednořetězcový oligopeptid s vnitřní komplementaritou utvářející hairpinovou strukturu. Oba tyto adaptéry obsahují motorový protein k pohánění DNA pórem a zároveň zajišťují nasednutí na oligopeptidy (tethery) na polymerové membráně. Sekvenace začíná na 5' konci vedoucího adaptéru (Obrázek 2. 2). Ve chvíli, kdy se začíná sekvenovat komplementární část vedoucího adaptéru, motorový protein vedoucího adaptéru rozvolňuje dsDNA a templátový řetězec proniká nanopórem. Nanóporem použitým v přístroji MinION je α-hemolysin. Ve stejnou chvíli je senzorem měřen iontový proud. Po dosažení hairpinového adaptéru se do nanopóru dostává i komplementární řetězec. Vzorkovací frekvence je v řádu tisíců Hz [32]. Data jsou poté předána na mikročip (ASIC – application-specific integrated circuit) a poté jsou již zpracována pomocí softwaru MinKNOW. [30]



Obrázek 2. 2: Sekvenace pomocí dvou adaptérů [30]

Na membránu s nanopóry je přiloženo napětí, které pohání jednořetězcovou DNA (ssDNA) přes nanopór a může tak být změřena změna proudu, která je následně klasifikována. Proudová změna, která vzniká při kontaktu s nanopórem je charakteristická pro každou bázi [32].

MinION obsahuje 512 kanálu, kde může probíhat sekvenace DNA. Což znamená, že v jednu chvíli může být sekvenováno až 512 odlišných DNA molekul [30]. Každý jednotlivý kanál je připojen ke čtyřem nádržím. Každá nádrž může obsahovat nanopór v elektricky izolované dvojvrstvě. Každý kanál vykazuje jinou aktivitu během sekvenování, některé póry jsou aktivnější než jiné [32].

Každá čtveřice nádrží je testována pro jednotlivý kanál pomocí operace „Mux“, během toho je každá nádrž ze čtveřice ohodnocena podle aktivity. Nejaktivnější nádrž je přiřazena do skupiny g1, druhá nejaktivnější do skupiny g2, třetí do skupiny g3 a čtvrtá do skupiny g4. Každý aktivní kanál získává ze začátku data ze skupiny g1 po dobu 24 hodin, dalších 24 hodin kanál získává data z 3 zbývajících skupin postupně [32].

2.1.2 Výstupní data

Výstupní data z MinION jsou ve formátu FAST5. Tento formát má hiearchické uspořádání a je variací na HDF5 standard. Každé jednotlivé čtení, které vzniká v jednom z 512 kanálů je uchováváno ve FAST5 formátu spolu s metadaty a charakteristickými informacemi pro každý kanál [30].

Pro zmírnění šumu jsou raw data jsou konvertována do událostí (events), přičemž každá sekvence je charakterizována průměrnou hodnotou proudu, odchylkou a délkou trvání [32]. Událost by měla odpovídat k-meru (krátká nukleotidová sekvence), s tím že více různých událostí může tvořit jeden stejný k-mer. Události, které jsou identické, by měly reprezentovat stejný k-mer. Raw data, která jsou vyobrazena v závislosti na čase, se nazývají „squiggle plot“. Base-calling, neboli přiřazování bází, je prováděno pomocí 5-mer nebo 6-mer (úseky DNA délky 5 nebo 6), přičemž existuje 4^5 případně 4^6 kombinací. Každá kombinace k-meru má modelovou průměrnou hodnotu, na základě které, lze daný k-mer identifikovat a přiřadit k němu odpovídající posloupnost nukleotidů [35]. 1D base-caling se provádí zvlášť pro templátový a komplementární události a poté jsou použity pro 2D base-caling.

2.1.3 HDF5 (Hiearchical Data Format version 5)

HDF5 je formát sloužící k ukládání velkého množství číselných dat. Podporuje velké množství různých datových typů. Mezi funkce, které HDF5 nabízí, patří pojmenované datasety, hierarchicky organizované skupiny a uživatelem definovaná metadata, respektive atributy, které lze přiřadit k datasetům a skupinám [36].

Pro práci s formátem HDF5 slouží standardizované knihovny vytvořené společností HDF Group. Softwarové knihovny jsou napsány v C, C++ a Javě. Nejznámější Python rozhraní, PyTables a h5py, oba využívají knihovnu psanou v programovacím jazyce C [36]. Formát HDF5 lze číst, upravovat a vytvářet téměř v každém programu. Například MATLAB využívá HDF5 jako defaultní formát pro svoje soubory s koncovkou .mat.

2.1.4 Přiřazování bází (Base-calling)

Base-calling neboli přiřazování bází probíhá mimo počítač, ke kterému je přístroj připojen, probíhá na Amazon cloud pomocí cloud-based Metrichor (který se využívá pro sekvenování pomocí nanopóru). FAST5 soubory, které jsou vytvořeny sekvenovacím softwarem, jsou následně analyzovány pomocí systému Metrichor. Base-calling ve verzi R7 je založen na algoritmu skrytého Markovova modelu (HMM) [33], jedná se o konečný model, který popisuje pravděpodobnost distribuce mezi nekonečným počtem možných sekvencí [37], s Viterbi dekodérem pro volání bází z událostí. V nové verzi chemie R9 byl HMM nahrazen rekurentní neuronovou sítí [38]. Pozice a počet nukleotidů, které ovlivňují hodnoty proudu, závisí na konkrétním nanopóru. Specifické změny proudu, které by dokázaly rozlišit jednotlivé nukleotidy, nejsou obvykle dostatečně vysoké, aby je bylo možné odlišit od šumu [39]. Proto se při volání bází využívá 5 nebo 6-mer. Hodnoty proudu jsou poté analyzovány, aby se zjistilo, které hodnoty odpovídají templátovému a komplementárnímu řetězci. Poté je pomocí statistických modelů Oxford Nanopore Technologies (ONT) vytvořen vztah mezi 5 nebo 6-mery a hodnotami proudu. Je vypočítán rozdíl mezi modelovými a reálnými hodnotami. Následně jsou vytvořeny dvě 1D sekvence pro každou molekulu a pokud je to možné, je vytvořena i 2D sekvence.

Pro každý fragment DNA procházející nanopórem, lze vytvořit tři typy čtení – templátové, komplementární a obousměrné [40]. Pokud se pro volání bází využívá informací pouze z jednoho řetězce – templátového nebo komplementárního, jedná se o 1D čtení, v případě, že se využívá jak templátového, tak komplementárního řetězce a z nich je poté vytvořena konsensuální sekvence, hovoříme o 2D obousměrném čtení. 2D sekvence je vytvořena pouze v případě dosažení dobrých výsledku z templátového a komplementárního řetězce. S využitím 2D čtení je dosaženo lepších výsledků [32].

Po skončení base-callingu je každá sekvence rozdělena do skupiny, na základě kvality, podle toho, zda prošla či nikoliv (pass/fail). Všechny tyto informace získané pomocí Metrichor během volání bází jsou uloženy ve FAST5 souboru, který lze následně stáhnout uživatelem [40].

Kromě systému Metrichor, který byl prvním softwarem pro base-calling, existují i další programy pro volání bází. A to buď vytvořené přímo od ONT nebo i ostatními uživateli. Vybrání vhodného postupu při volání bází ovlivňuje přesnost sekvenovaní. Nové metody pro volání bází se stále vyvíjí pro zlepšení přesnosti. Nejnovější algoritmy pro volání bází se snaží vynechat rozdělení signálu do událostí a překládat do nukleotidů přímo.

Právě volání bází představuje největší překážku pro správné sekvenování pomocí nanopóru. I přesto, že jsou známé průměrné hodnoty a délky trvání jednotlivých

modelových k-mer, ve skutečnosti to tak jednoduché není a reálné k-mery neodpovídají přesně těm modelovým. Každý nanopór totiž poskytuje mírně odlišné hladiny proudu. I při průchodu stejné sekvence stejným nanopórem, nejsou hodnoty proudu totožné.

2.1.5 FASTQ formát

Nukleotidové sekvence po ukončení base-calling jsou ve finálním FAST5 souboru uloženy ve formátu FASTQ. Tento formát obsahuje hlavičku (*header*), která slouží jako identifikátor sekvence, samotnou sekvenci a kvalitativní zhodnocení dané sekvence. Zhodnocení kvality je prováděno pomocí znaků. Offset je nastaven tak, že minimální kvalita, odpovídající phred score x je nastavena jako symbol „!“ a symbol „~” označuje naopak nejvyšší kvalitu a odpovídá phred y (skóre 40), což odpovídá přesnosti čtení 99,99 %.

FASTQ soubor vytvořený během base-calling obsahuje v hlavičce identifikátor a název souboru, ze kterého lze vyčíst číslo kanálu a čtení (Obrázek 2. 3).

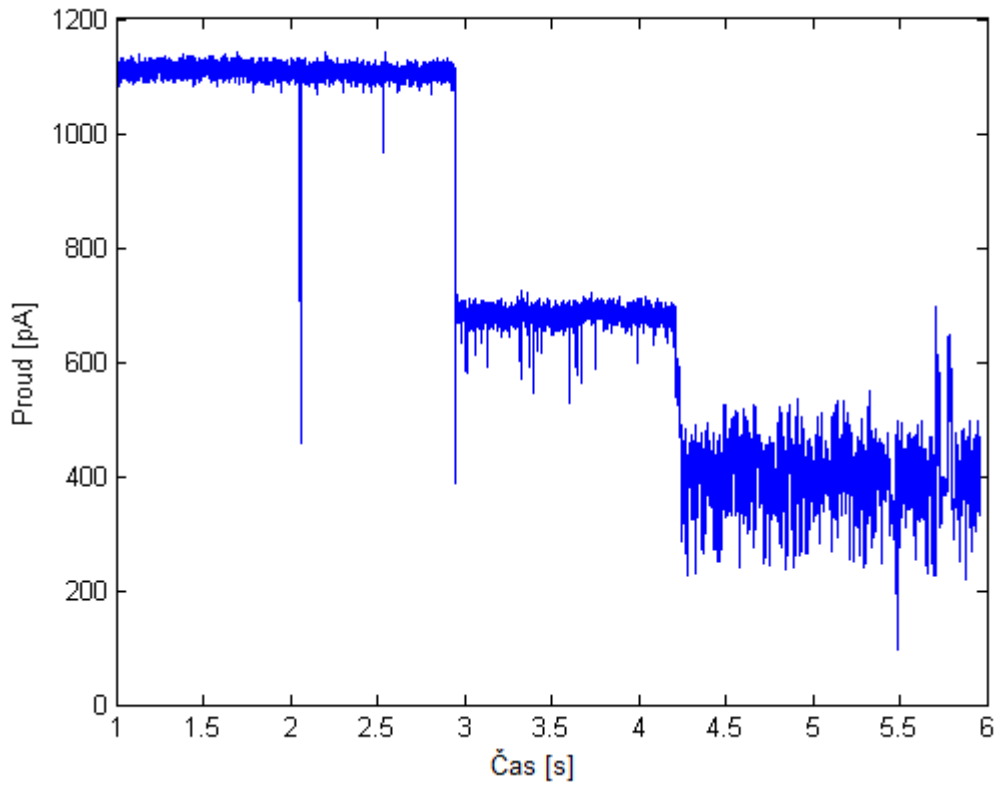
Obrázek 2. 3: Ukázka FASTQ souboru získaného z FAST5 formátu

2.1.6 R7 vs. R9

V nejnovější verzi R9 a výše je na pór přidána CsgG membrána a HMM je nahrazena rekurentní neuronovou (RNN) sítí pro zlepšení base-callingu [38]. HMM měla výhody při použití na krátké vzdálenosti, jako přesun z jednoho k-meru na druhý, ale nedokázala uchovávat závislosti na větší vzdálenosti [41]. Chemie R9 má větší přesnost a sekvenace probíhá mnohem rychleji. V případě R7 byla rychlost ~70 bp za sekundu, ve verzi R9 je to ~250 bp za sekundu [38]. Oproti verzi R7, nová verze pracuje kromě 2D také s 1D metodami, což umožňuje sekvenovat pouze jeden řetězec DNA, při použití pouze 1D metod není využíván hairpinový adaptér a komplementární sekvence se do nanopóru nedostává. Výhodou používání pouze 1D metod je delší životnost póru a rychlejší průběh. Chybovost, která je dána inzercí, delecí a nezavoláním báze,

poklesla ve verzi R9 z původních 9,1 % na 7,5 % pro 2D čtení a chybovost pro templátový řetězec také poklesla z 26,7 % na 14,5 % [38].

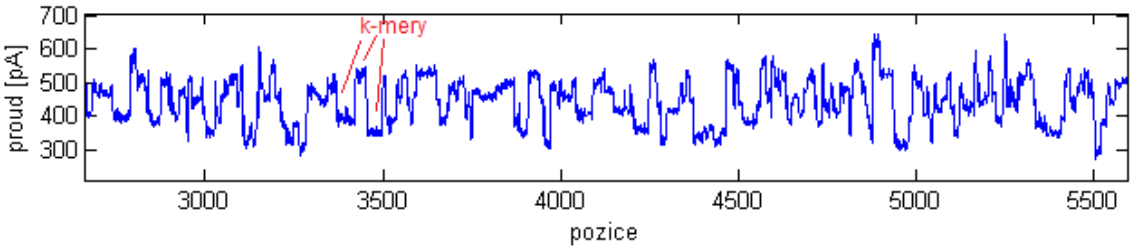
Rozptyl hodnot elektrického proudu je u R9 až dvakrát širší. Rozdíl je také v samotném ukládání dat ve FAST5 formátu. V chemii R7 ve verzi 1.0 byla templátová a komplementární data uložena ve skupině *Basecall_2D_00*, ve verzi 1.1 byla tato data přesunuta do skupiny *Basecall_1D_000*. Chemie R9 ukládá raw data do úplně nové skupiny */Raw*, ze které lze také získat původní (raw) signál. Ve starší verzi chemie R7 se signál do FAST5 formátu neukládal, ale musel být rekonstruován ze středních hodnot a délky trvání, což mělo za následek ztrátu některých informací. Z FAST5 formátu lze také získat informace o vzorkovací frekvenci, která je u obou chemií odlišná. Čtení, která byla využita pro zobrazení (Obrázek 2. 4, Obrázek 2. 6), mají vzorkovací frekvenci 4 kHz pro chemii R9 a 5 kHz pro chemii R7. Vysoká vzorkovací frekvence zajišťuje snímání proudu při procházení každého nukleotidu.



Obrázek 2. 4: Zobrazení raw signálu v čase z verze R9.

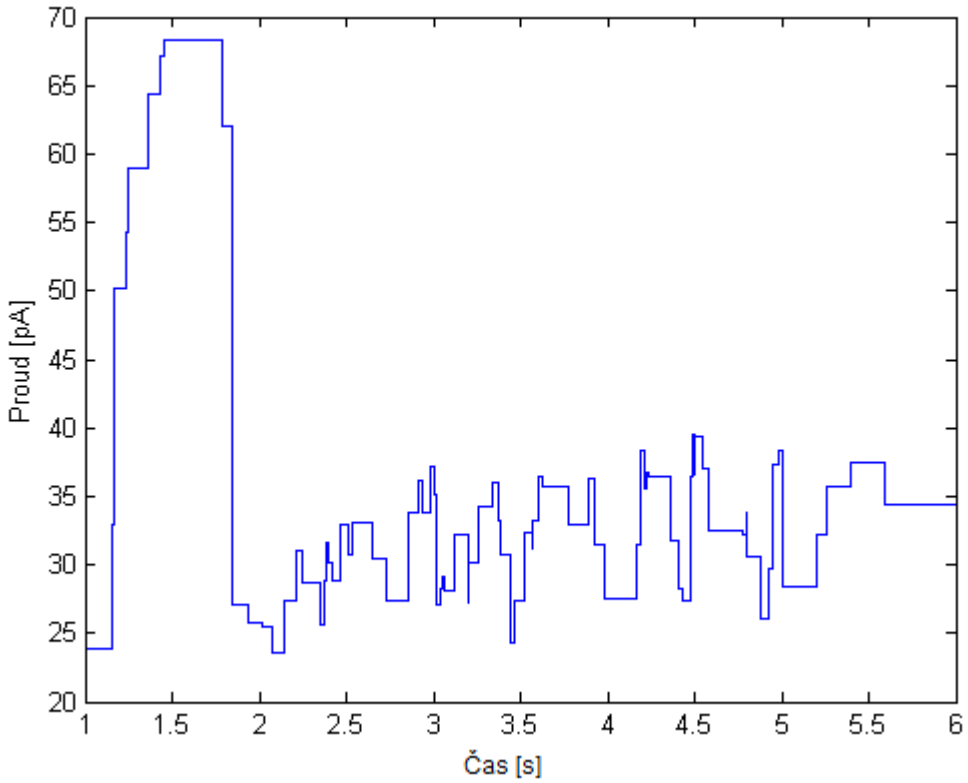
Signál je rozdělen na události, které charakterizují jednotlivé k-mery. Jednotlivé změny hladiny v grafu vyznačují rozdílný k-mer (Obrázek 2. 5). V tomhle případě se jedná o 6-mery. Každá změna hladiny se tedy liší o jeden nukleotid. V případě sekvence

ACTGACCT by první hladina označovala k-mer ACTGAC, druhá hladina CTGACC a třetí TGACCT.



Obrázek 2. 5 Signál se znázorněnými k-mery

Každá hladina signálu označuje jeden k-mer složený ze šesti nukleotidů, k-mery jsou charakteristické svým průměrem a také délkou trvání, toho je dále využíváno pro volání bází (base-calling). Pomocí těchto známých hodnot lze také zrekonstruovat signál ze starší verze R7.



Obrázek 2. 6: Zrekonstruovaný signál z verze R7

Zrekonstruovaný signál se oproti raw signálu z verze R9 liší především tím, že jsou vyobrazeny pouze průměrné hodnoty v závislosti na jejich trvání. Raw signál z verze R7 nelze získat. Průběh proudu je tak velmi odlišný od chemie R9. Signál je tvořen pouze ostrými hranami, což pro práci se signálem není vhodné, z důvodu ztráty velkého množství informací.

3 NÁVRH ŘEŠENÍ

Přístroj MinION poskytuje na výstupu soubor signálů, které lze následně zpracovávat, či analyzovat. Původní metody zpracování dat z MinION nepracují s raw signálem, ale rovnou s přeloženými nukleotidy v 5 nebo 6-merech. Zpracování raw signálu a hledání překryvů pro genomové sestavení přímo ze signálů by mohlo poskytnout nové informace a zlepšit tak přesnost sekvenování i samotné sestavování genomu. Při klasickém sekvenování pomocí MinION dochází k největším chybám při volání bází. Cílem je tak nejdříve jednotlivé raw signály spojit, na základě nalezení jejich vzájemných překryvů a až poté překládat do nukleotidů.

Jednotlivá čtení by tak mohla být spojena pomocí jejich překryvů v signálu a následné sestavení genomu by mohlo probíhat na základě OLC metody. Metody založené na OLC jsou vhodné především pro dlouhá čtení. Použití metody založené na de Bruijnových grafech by kvůli hledání překryvů mohlo vést ke ztrátám informací ze signálu. Pokud by byl signál rozdělen na kratší úseky a následně analyzován, ztratily by se informace o překryvech a o pozicích těchto kratších úseků v signálu a signál by se tím znehodnotil. Hledání překryvů na takto krátkých úsecích signálu, by nebylo vhodné, protože při porovnávání pouze krátkých úseků, by mohlo docházet k nalezení shody na více částech signálu, i když spolu ve skutečnosti vůbec nesousedí. Pro přesnější hledání shody v překryvu je lepší použít co nejdelší možné úseky.

Použití rekonstruovaného signálu z verze R7 by mohlo vést ke špatným výsledkům, z důvodu ztráty informací, kdy jsou dostupné pouze střední hodnoty úseku signálu po určitý čas. Ke hledání překryvů tedy budou použity pouze raw signály získané pomocí nové verze chemie R9, kde je charakteristický průběh mnohem viditelnější a pro potřeby hledání překryvu je tento signál vhodnější.

Signál má v některých částech schodovitý průběh, což by mohlo být také výhodou při hledání překryvů.

3.1 Předzpracování signálů

Před samotným hledáním překryvů je potřeba signály upravit pro zlepšení hledání shody. Ve většině signálu jsou na různých místech výrazné známky rušení, které by mohly přesnost algoritmu snížit. Pro odstranění tohoto typu rušení je vhodné použít mediánový filtr. Jelikož jsou pro signál charakteristické jeho k-mery, které mají malé výchylky proudu, přílišná filtrace by mohla tyto elementy poškodit nebo ztratit.

3.2 Hledání překryvů

Při hledání překryvů je důležité, aby se sufixová část jednoho úseku signálu překrývala s prefixovou částí druhého signálu. Každý bod ze začátku druhého signálu tak musí být zároveň součástí prvního signálu. Když vezmeme první bod druhé sekvence, tak musí být také obsažen v první sekvenci. Nalezneme tak v první sekvenci všechny pozice, kde se tento bod vyskytuje, z tohoto místa prodloužíme úsek signálu až do konce (Obrázek 3. 1). Pro porovnání nebudou použity signály, jejichž délka od shodného bodu je kratší než délka hlavního signálu od jejich společného bodu. Takový signál by ten hlavní, ke kterému je hledán jeho navazující, nijak neprodloužil a tyto signály by algoritmus pouze zpomalovaly.

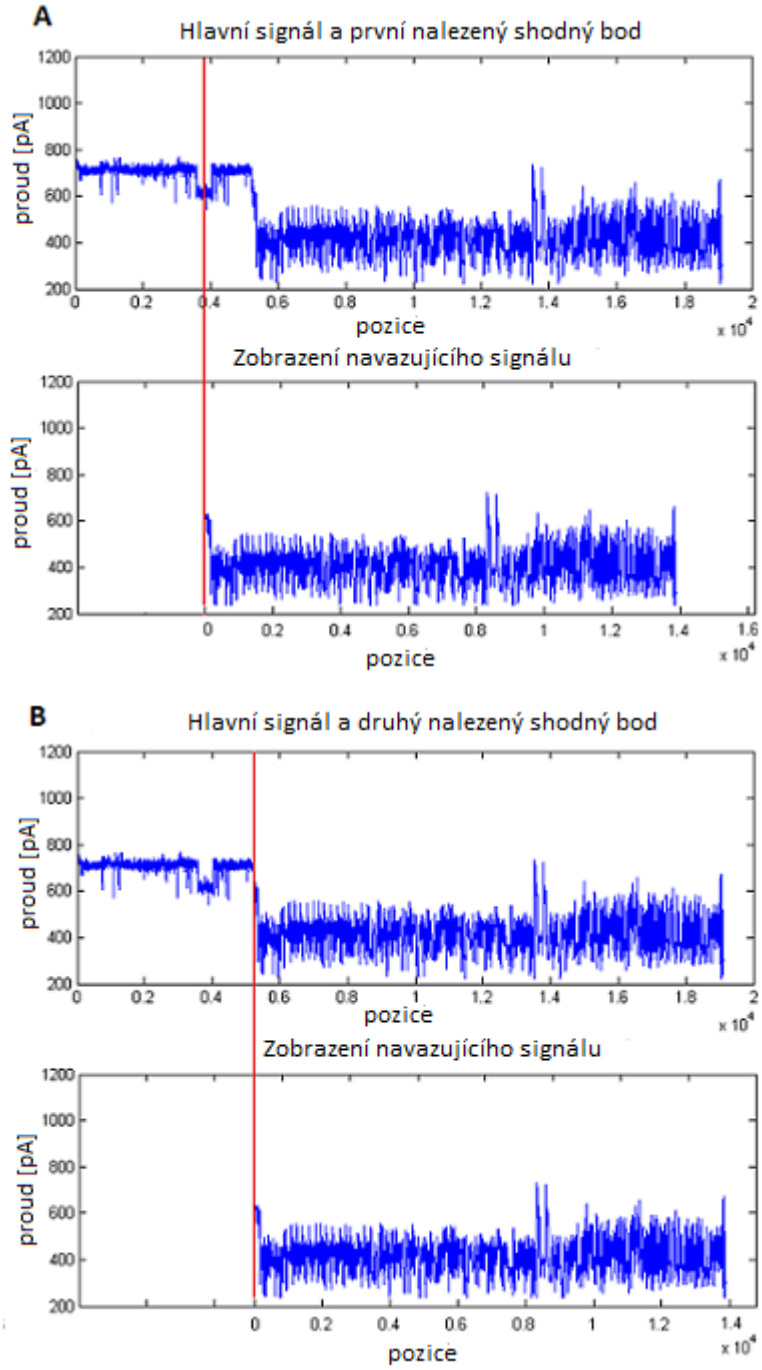
Jelikož signály nejsou stoprocentně totožné, nelze shodu stanovit pouze podle počtu shodných bodů. K hledání překryvů tak bude využita diference signálu. Jejich překryv tak bude určen podle rozdílů těchto dvou nalezených úseků od jejich společného bodu podle vzorce:

$$koef = \frac{\sum_{i=1}^{L} |main\_select_i - other\_select_i|}{L}, \tag{3.1}$$

kde *main_select* je vybraná část hlavního signálu od společného bodu navazujícího signálu, *other_select* je část navazujícího signálu od vybraného shodného bodu až po délku hlavního signálu tak, aby délka *main_select* a *other_select* byla totožná. *L* je počet vzorků signálu v *main_select* a *other_select*.

Rozdílový koeficient *koef* bude vypočítán pro všechny nalezené shodné body. Při použití diference signálů, je potřeba, aby výsledná suma vydělena počtem vzorků byla rovna 0 (pro totožné signály) nebo, aby se 0 blížila. Tahle podmínka bude kontrolována a pouze od určité hranice bude signál označen jako možný překryv. Podle hranice a dalších statistických veličin, bude určen signál, který nejlépe navazuje na první signál. Výsledkem by bylo spojení dvou signálů, jejichž překryv bude charakterizován nejnižším rozdílovým koeficientem.

Ke každému signálu je známá také nukleotidová sekvence ve FASTQ formátu, kterou lze získat z původního FAST5 souboru. Tato nukleotidová sekvence může být využita k ověření správnosti spojení dvou signálů. S využitím lokálního zarovnání nukleotidových sekvencí a podle dosaženého skóre lze ověřit, zda určené signály byly přiřazeny správně, či nikoliv.



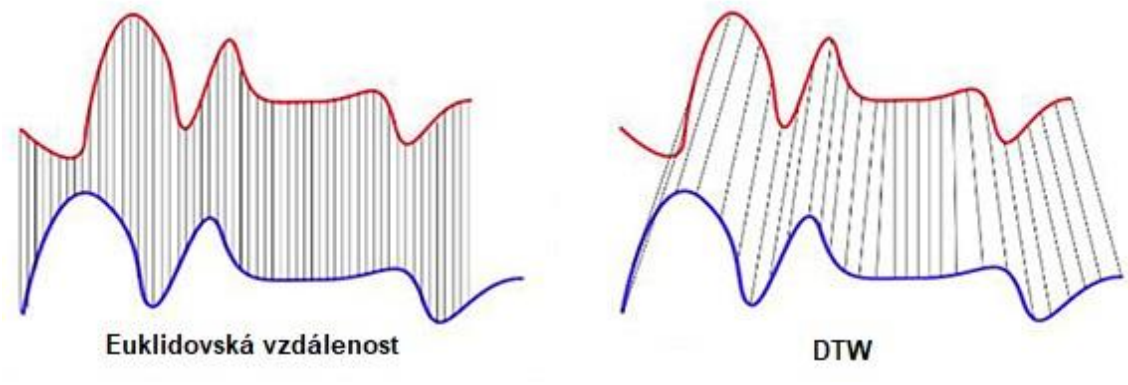
Obrázek 3. 1: Znázornění hledání překryvu na základě shodných bodů

Na obrázku je znázorněn hlavní signál, kde byly nalezeny dva shodné body (vyznačeny červeně) s možným navazujícím signálem. V tomto případě se vypočítá nejdříve koeficient pro první shodný bod (A), poté se vypočítá koeficient pro druhý shodný bod (B). Ve druhém případě bude mít vypočítaný koeficient daleko nižší hodnotu – signály jsou v tomto posunutí podobnější.

3.3 Úprava pomocí dynamického borcení časové osy

Metodu dynamického borcení časové osy by bylo možné aplikovat pro zlepšení přesnosti algoritmu. Algoritmus popsaný výše, lze tak pouze upravit a před výpočtem koeficientu *koef* jsou úseky *main_select* a *other_select* nejprve zarovnány a až poté je vypočten koeficient. Dále algoritmus probíhá stejně jako v předchozím případě.

Dynamické borcení časové osy (DTW) je metoda, která je hojně využívána pro časově závislé signály. Původně byla používána k rozpoznání řeči. Metoda hledá nejlepší možné zarovnání pro dva časově závislé signály. Dochází tak k nelineárnímu zarovnání těchto signálů (Obrázek 3. 2).



Obrázek 3. 2: Porovnání dvou metod pro vzdálenost signálů [61]

Obrázek vlevo znázorňuje euklidovskou vzdálenost pro dva časově závislé signály, kdy se počítá vzdálenost mezi každým odpovídajícím bodem prvního a druhého signálu. Na obrázku vpravo je znázorněno zarovnání pomocí metody DTW. Zarovnání těchto dvou signálů je nelineární a lépe odpovídá podobnosti dvou časově závislých signálů.

Nejprve je vypočítána matice distancí mezi prvky signálů. Tato matice je označována jako matice kumulativních vzdáleností. Je získána podle vzorce [61]:

$$D(n,m) = \min\{D(n-1,m-1), D(n-1,m), D(n,m-1)\} + c(x_n, y_m), \quad (3.1)$$

kde D je matice kumulativních vzdáleností, $n \in (1,N\rangle$ a $m \in (1,M\rangle$, kde N je délka jednoho signálu a M je délka druhého signálu, c označuje nákladovou funkci. Jedna z možných nákladových funkcí je podle vzorce [61]:

$$c \in R: c(x_n, y_m) = |x_n - y_n|, \quad (3.2)$$

kde c je nákladová funkce a x_n,y_m jsou jednotlivé vzorky signálu.

Inicializace prvního řádku a prvního sloupce matice kumulativních vzdáleností je následující [61]:

$$D(n,1) = \sum_{k=1}^{n} c(x_k, y_1), \;\; n \in \langle 1, N \rangle, \tag{3.3}$$

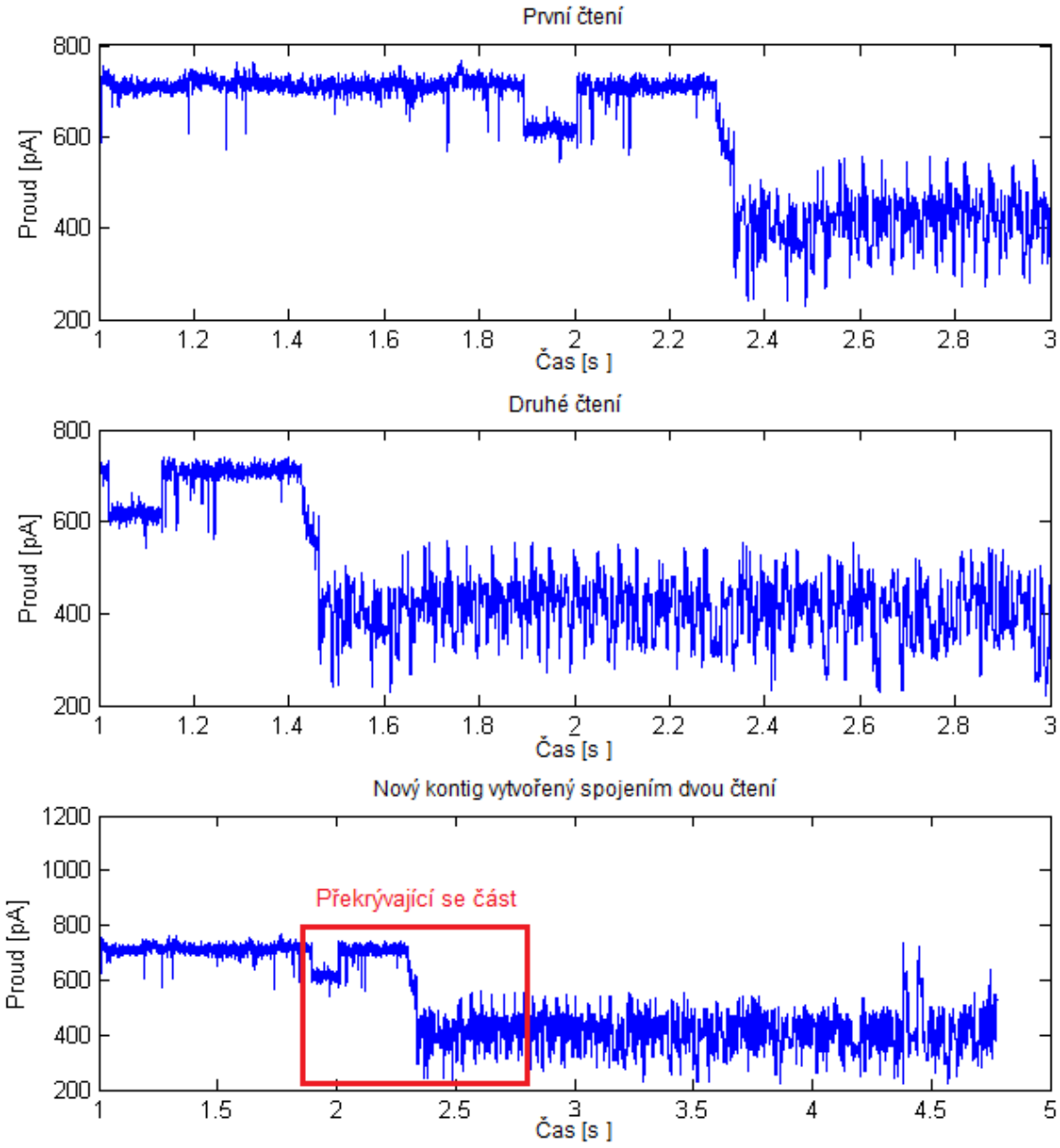
$$D(1,m) = \sum_{k=1}^{m} c(x_1, y_k), \;\; m \in \langle 1, M \rangle, \tag{3.4}$$

kde *D* je matice kumulativních vzdáleností, *N* je délka jednoho signálu a *M* je délka druhého signálu, *c* označuje nákladovou funkci.

Po získání matice kumulativních vzdáleností, je následně rekonstruována zpětná cesta. Cesta je hledána od pravého horního rohu pomocí nejmenších kumulativních vzdáleností až na počátek k levému spodnímu rohu. Jelikož s růstem matice kumulativních vzdáleností a tedy s růstem velikosti signálů, roste počet zpětných cest exponenciálně, je metoda DTW časově velmi náročná. Časová náročnost je *O(MN).*

Metoda zarovnává signály od jejich začátku až po konec, jedná se tedy o globální zarovnání. Pro porovnávání signálů ze sekvenace nanopórem není žádoucí zarovnat signálu od začátku až do konce, ale pouze zarovnat jejich odhadovaný vzájemný překryv. Proto lze tuto metodu použít v algoritmu před samotným výpočtem koeficientu a vypočítat tak až rozdílový koeficient zarovnaných signálů. Vybrané úseky *main_select* a *other_select* jsou tak nejdříve zarovnány pomocí DTW a až poté je ze zarovnaných signálů vypočten koeficient (3.1).

Po nalezení překryvu mezi signály, lze tyto dva signály spojit (Obrázek 3. 3)

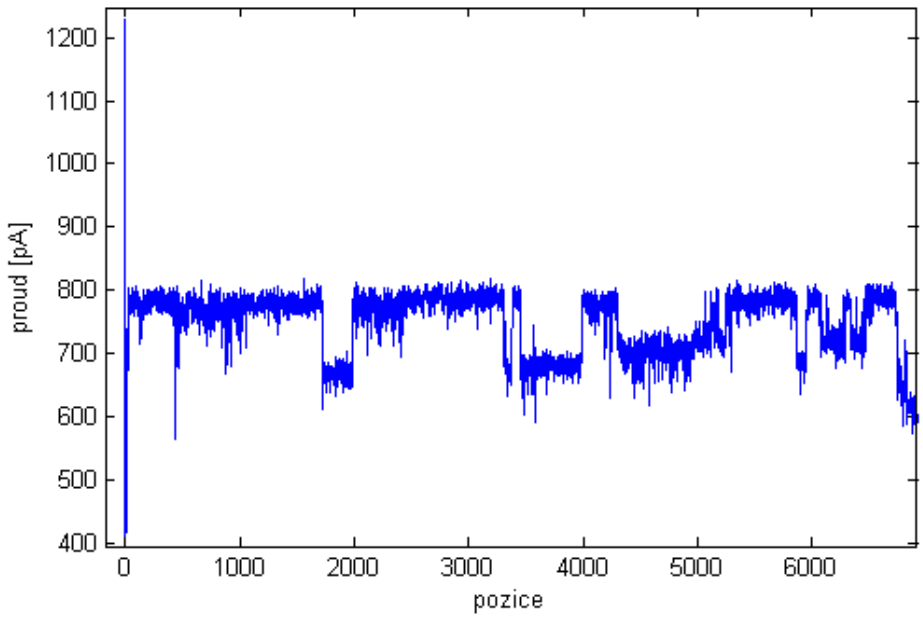


Obrázek 3. 3: Sestavení nového kontigu ze dvou překrývajících se čtení

První a druhé čtení se navzájem překrývají. Koncová část prvního čtení od přibližně 1,8 s se překrývá se začínající částí druhého čtení až po přibližně 2 s.

4 REALIZACE

Pro hledání překryvů mezi signály byly vytvořeny dvě funkce v prostředí MATLAB. První funkce find_overlap využívá rozdílového signálu bez zarovnání, zatím co druhá funkce find_overlap_dtw nejdříve zarovnává pomocí DTW a až poté probíhá výpočet rozdílového koeficientu. Obě funkce obsahují dva vstupy – všechny raw signály uložené v jedné proměnné a všechny nukleotidové sekvence uložené v další proměnné. Funkce *find_overlap* byla opatřena i třetím vstupem, který reprezentuje filtraci, kdy nula znamená vynechání filtrace. Funkce vždy vybere jeden signál jako hlavní (*main*) a hledá k němu navazující druhý signál (*other*). Ze signálu *other* je vybrán padesátý vzorek a ten je porovnáván se všemi vzorky hlavního signálu. Výsledkem tohoto porovnávání je seznam pozic, na kterých se hlavní signál s tímto vybraným vzorkem shoduje. Vybrání padesátého vzorku signálu je dáno tím, že velká část signálů obsahuje v prvních několika vzorcích výrazné skoky a píky (Obrázek 4. 1), které výsledek zkreslovaly. Tyto výrazné píky mohou být způsobeny přechodnými jevy při nasednutí DNA na nanopór a jsou pro vyhodnocování nežádoucí. Vybrání padesátého vzorku k porovnávání tento problém eliminoval.



Obrázek 4. 1: Zobrazení začátku signálu s viditelným rušením

Dále byly smazány pozice, které se nacházely na konci hlavního signálu (v poslední třetině). Porovnávání takhle krátkých úseků vedlo opět ke zkreslení výsledků. Při nalezení shodného vzorku v posledních několika vzorcích signálu, je

vybraná část k porovnávání příliš krátká. Koeficient (3.1) byl u těchto krátkých úseků často velmi nízký, a to z důvodu nedostatečného počtu vzorků k rozeznání překryvu.

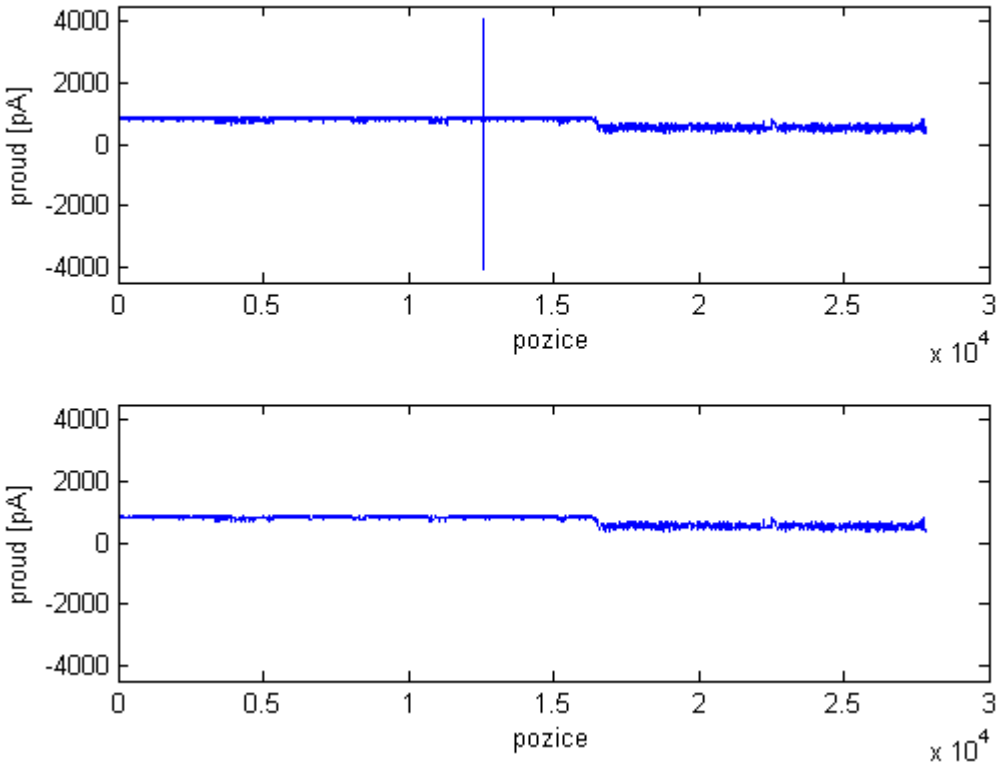
Pro zjednodušení byly dále odstraněny pozice, ve kterých byla délka hlavního signálu od tohoto shodného bodu delší, než délka druhého signálu, což znamená, že druhý signál by mohl být součástí hlavního signálu, ale dále by tento signál nijak neprodloužil.

4.1 Předzpracování mediánovou filtrací

Všechny signály před začátkem porovnávání byly upraveny mediánovou filtrací. Kvůli výraznému rušení, vyskytující se v téměř všech signálech, docházelo k častějšímu přiřazování nevhodných signálů.

Délka okna pro mediánovou filtraci byla nastavena na hodnotu 5. K filtraci byla použita funkce v prostředí MATLAB *medfilt1*. Při této délce okna jsou ještě stále zachovány průběhy k-mer, ale většina výrazného rušení je odstraněna (Obrázek 4. 2).

Jelikož jsou překryvy v signálu hledány na úrovni k-mer, není vhodné signály příliš filtrovat, mohlo by tím docházet ke ztrátě jednotlivých k-mer.



Obrázek 4. 2: Odstranění výrazného rušivého elementu ze signálu č. 1

Signál č. 1 obsahoval přibližně v polovině rušivý prvek, s hodnotou proudu od -4000 až do 4000 pA. Takový rušivý element velmi ovlivňuje výsledky rozdílového koeficientu.

4.2 Využití rozdílového signálu

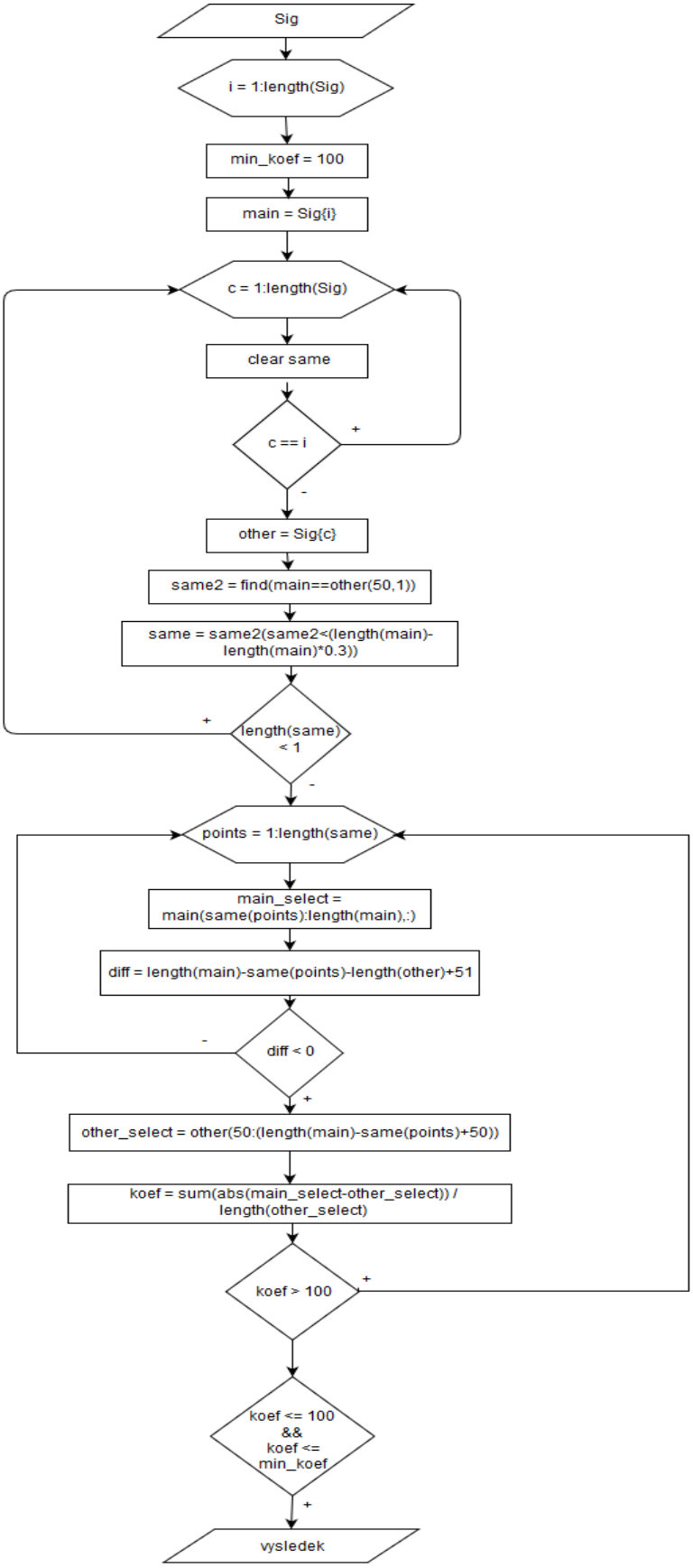
Ve funkci *find_overlap* byly od každého shodného bodu vytvořeny dva vektory. Jeden pro hlavní signál od shodného bodu až po konec hlavního signálu (*main_select*) a druhý od dvoustého vzorku s délkou odpovídající vybranému úseku *main_select* (*other_select*). Dále byl získán diferenční signál těchto dvou úseků. Všechny vzorky diferenčního signálu byly sečteny a vyděleny počtem vzorků (3.1). Tímto byla získána hodnota rozdílového koeficientu *koef*. Nižší koeficient vždy značí vyšší podobnost. Pro totožné signály by hodnota koeficientu byla rovna nule.

Pro přiřazení správného signálu je nejdůležitější právě hodnota tohoto koeficientu, proto všechny výsledné hodnoty koeficientu vyšší než stanovená minimální hodnota 100, jsou nevyhovující. Tato hodnota byla stanovena experimentálně tak, aby byl ke každému signálu přiřazen signál navazující. Pokud je tak hodnota koeficientu nižší než 100, je druhý signál přiřazen jako možný navazující signál.

Po přiřazení prvního navazujícího signálu je změněna hranice koeficientu, při kterém lze druhý signál považovat za navazující. Tato hodnota je nastavena na hodnotu koeficientu pro právě přiřazený signál. Tímto způsoben se snižuje hranice a zároveň zvyšuje přesnost pro přiřazené úseky. Jako navazující signál lze nyní označit pouze úsek s menší hodnotou rozdílového koeficientu, než hodnota koeficientu předcházejícího přiřazeného signálu. Tento postup je opakován pro všechny nalezené shodné vzorky signálu a pro všechny signály navzájem.

Výstupem je matice hodnot, která v prvním sloupci obsahuje hlavní signál a ve druhém sloupci obsahuje jeho možný navazující signál, ve třetím sloupci je zapsána pozice (vzorek) hlavního signálu, od které nastává překryv. Ve čtvrtém sloupci je dále zapsán vypočítaný koeficient *koef* pro danou dvojici signálu. V pátém sloupci je zobrazeno skóre z lokálního zarovnání nukleotidových sekvencí pomocí funkce *swalign*, s penalizací mezer 50 a penalizací prodlužování mezer 25.

Vývojový diagram vytvořeného algoritmů je znázorněn níže (Obrázek 4. 3).



Obrázek 4. 3: Vývojový diagram vytvořeného algoritmu

4.3 Využití dynamického borcení časové osy

Původní funkce *find_overlap* využívající pouze rozdílového signálu byla modifikována a před výpočet koeficientu bylo vloženo zarovnání úseků *main_select* a *other_select*. Dále pokračoval algoritmus stejně jako v původní verzi. Tímto způsobem byla vytvořena funkce *find_overlap_dtw*. Pro výpočet DTW byla použita funkce *samplealign*, vstupem funkce jsou dva signály *X* a *Y*, které jsou použity pro porovnání a výstupem jsou dva vektory, kde první obsahuje indexy ze signálu *X* shodující se signálem *Y* a druhý vektor obsahuje indexy signálu *Y* shodující se s indexy signálu *X*.

Pro zrychlení výpočtu lze ve funkci nastavit různé parametry. Parametr *band* omezuje maximální dovolenou vzdálenost mezi dvěma signály, omezuje tak počet potenciálních shod mezi danými signály. Parametr *band* byl nastaven na hodnotu 35. Dalším parametrem je parametr *quantile*, označuje hodnotu mezi 0 a 1. Využívá se pro výpočet penalizace mezer. Tento parametr byl pro výpočet nastaven na hodnotu 0.5. Pro zarovnání obou signálů, byly použity výstupy funkce *samplealign* a pomocí interpolace byly úseky signálu zarovnány. Následně byl vypočten rozdílový koeficient těchto zarovnaných úseků stejně, jak bylo popsáno v předešlé kapitole o využití rozdílového signálu. Zbývající kroky algoritmu byly stejné jako při použití rozdílového signálu bez zarovnání.

4.4 Doplňkové funkce

Pro práci se signály a nukleotidovými sekvencemi uloženými ve formátu FAST5 v prostředí MATLAB bylo potřeba vytvořit doplňkové funkce. Kromě hlavní funkce pro hledání překryvu tak byla vytvořena funkce *load_data*, která z FAST5 souboru dokáže získat požadované raw signály do jedné proměnné. Každé čtení získané během sekvenace, a tedy i každý signál, je uloženo jako samostatný FAST5 soubor. Funkce načítá všechny soubory ze složky s koncovkou .fast5. Výstupem této funkce je buňkové pole obsahující v každém řádku jeden raw signál. Vytvořená funkce *load_data* využívá funkcí pro práci s HDF5 formátem *h5info* a *h5read*. Cesta vedoucí k signálu je tvořena */Raw/Reads/Read_N/Signal*, kde *N* označuje jednotlivá čtení a je různé pro každý signál.

Dále byla vytvořena funkce *load_seq* pro získání sekvencí nukleotidů pro každý signál také z FAST5 souboru. Nukleotidové sekvence jsou zde uložené ve FASTQ formátu. Sekvence lze ve FAST5 souboru získat z datasetu *Fastq* pomocí cesty */Analyses/Basecall_1D_000/BaseCalled_complement/Fastq*. Výstupem funkce je opět buňkové pole, které v každém řádku obsahuje jednu sekvenci pro daný signál.

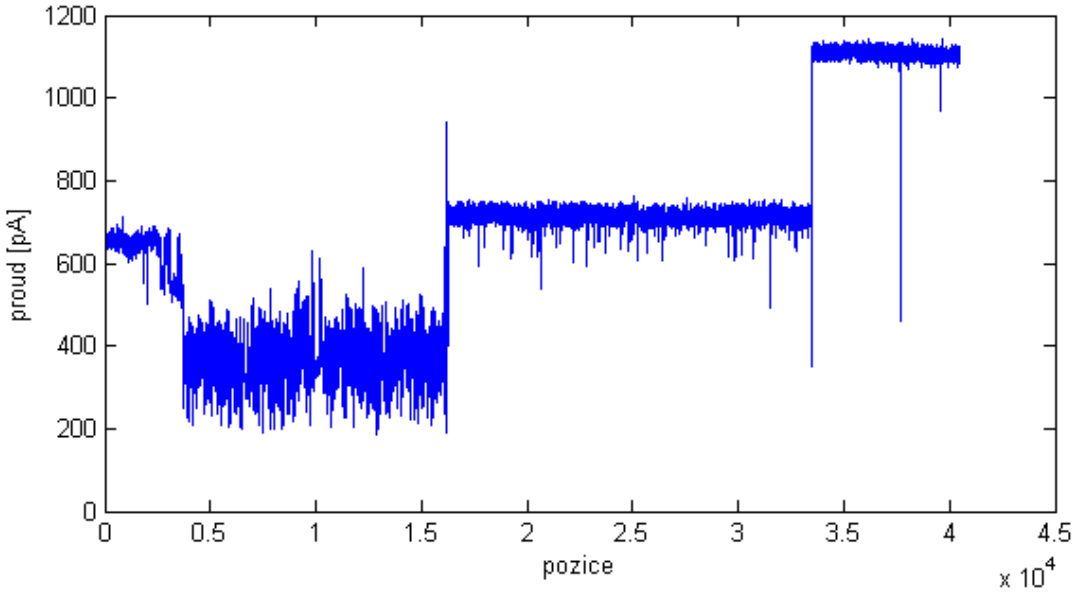
5 VYHODNOCENÍ ALGORITMU

Protože z reálných dat nelze jednoznačně určit, zda algoritmus funguje správně a opravdu přiřazuje signály, které jsou podobné a mají vzájemný překryv, vyhodnocení a funkčnost algoritmu bude ukázáno na modelových datech, kde jsou jednotlivé překryvy známé, a poté i na datech reálných.

Vyhodnocení je věnováno algoritmu pouze s použitím rozdílového signálu – funkce *find_overlap*. Upravenému algoritmu s použitím DTW (*find_overlap_dtw*) je věnována poslední podkapitola.

5.1 Modelová data

Pro ověření správnosti algoritmu byla vytvořena modelová data spojením několika náhodných signálů (Obrázek 5. 1) a následným rozdělením spojeného signálu do pěti různě dlouhých úseků. Tímto způsobem vzniklo 5 signálů, která na sebe plně navazovala. Modelová data byla vytvořena proto, že je zde stoprocentní překryv a lze tak ověřit, zda algoritmus funguje opravdu správně. U reálných dat nelze jednoznačně určit, zda je určený překryv opravdu správný.



Obrázek 5. 1: Modelový signál spojený z pěti různých signálů

Modelový signál byl vytvořen tak, aby obsahoval velké množství různých skoků a rušivých elementů. Překryvy byly nastaveny tak, aby se nacházely mezi vzorky 4000 až 6000.

Překryvy byly nastaveny tak, že signál č. 1 se překrývá se signálem č. 3, signál č. 2 se překrývá se signálem č. 4, signál č. 3 se překrývá se signálem č. 2 a poslední signál č. 4 se překrývá se signálem č. 5 (Tabulka 5. 1).

Tabulka 5. 1: Očekávaný výstup algoritmu

Hlavní signál	**Očekávaný navazující signál**
1	3
2	4
3	2
4	5

Pro úseky jedna až čtyři byly nalezeny odpovídající překryvy. K signálu č. 5 nebyl přiřazen žádný překryv, jelikož se jedná o poslední úsek signálu a hodnota koeficientu pro signál č. 5 a jakýkoliv další signál nebyl nikdy menší než stanovené minimum (100) nebo nebyl nikdy nalezen shodný bod s ostatními signály.

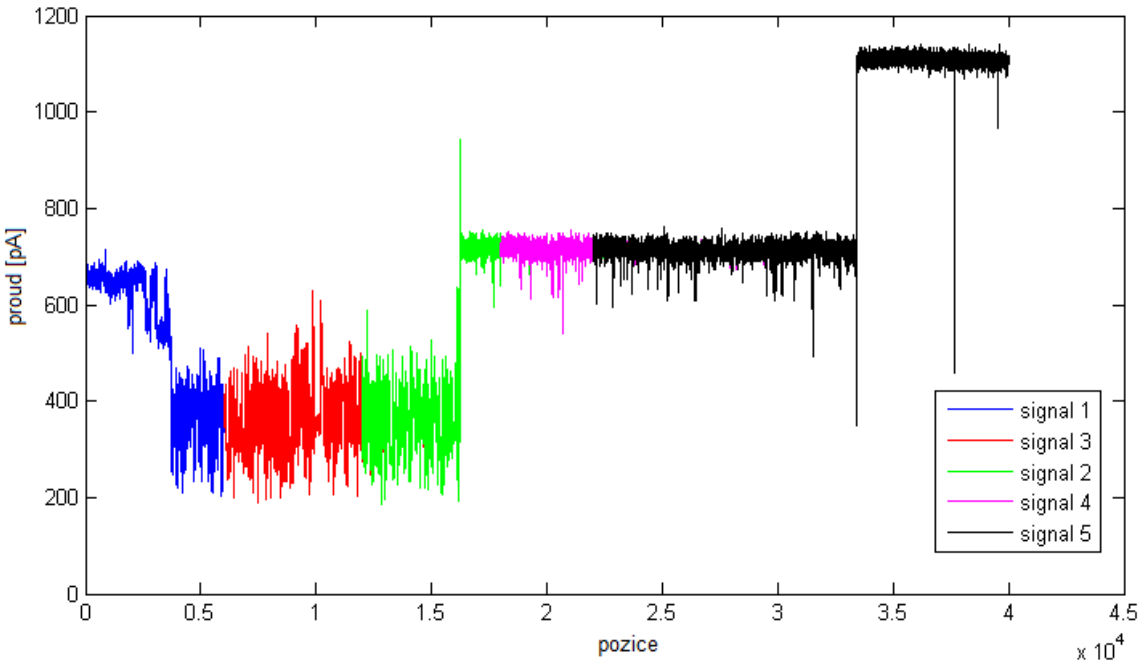
Výsledky dopadly dle očekávání a k hlavním signálům byly správně přiřazeny všechny navazující signály. Minimální koeficient byl vždy u překrývajících se úseků roven nule, což značí identické signály. Ke každému signálu bylo přiřazeno více možných navazujících signálu, ale s koeficientem rovným nule byl vždy pouze jeden.

Tabulka 5. 2: Výstup algoritmu na modelových datech

Hlavní signál	**Navazující signál**	**Počátek překryvu**	**Koeficient**
1	3	6 000	0
2	4	6 001	0
3	2	6 001	0
4	5	4 001	0

Algoritmus správně vyhodnocuje signály, jejichž překryv je naprosto totožný. Proto lze předpokládat, že při použití na reálných datech, kde překryvy nejsou nikdy shodné, bude algoritmus přiřazovat signály správně na základě jejich podobnosti.

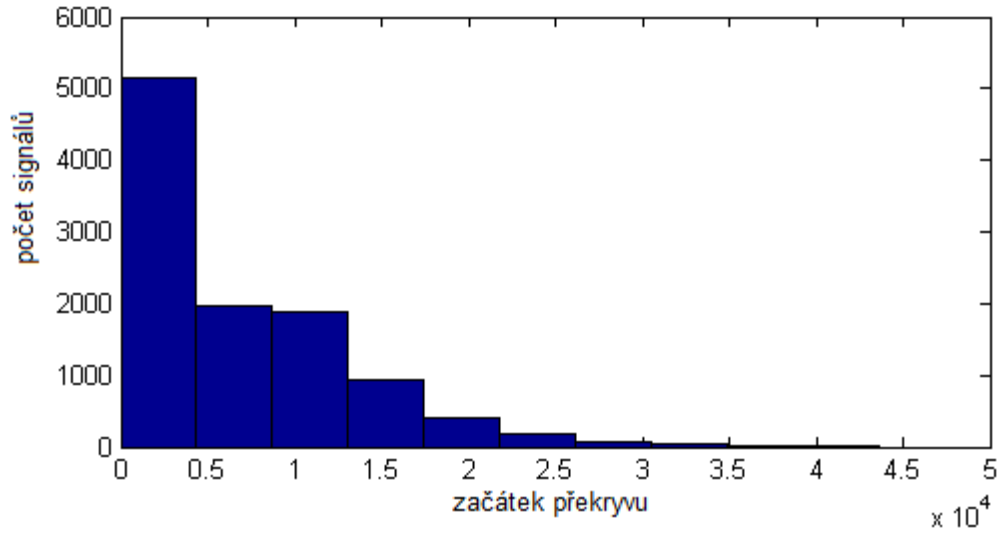
Dílčí modelové signály je tak možné spojit díky známým pozicím překryvu do původní podoby signálu před jeho rozdělením (Obrázek 5. 2).



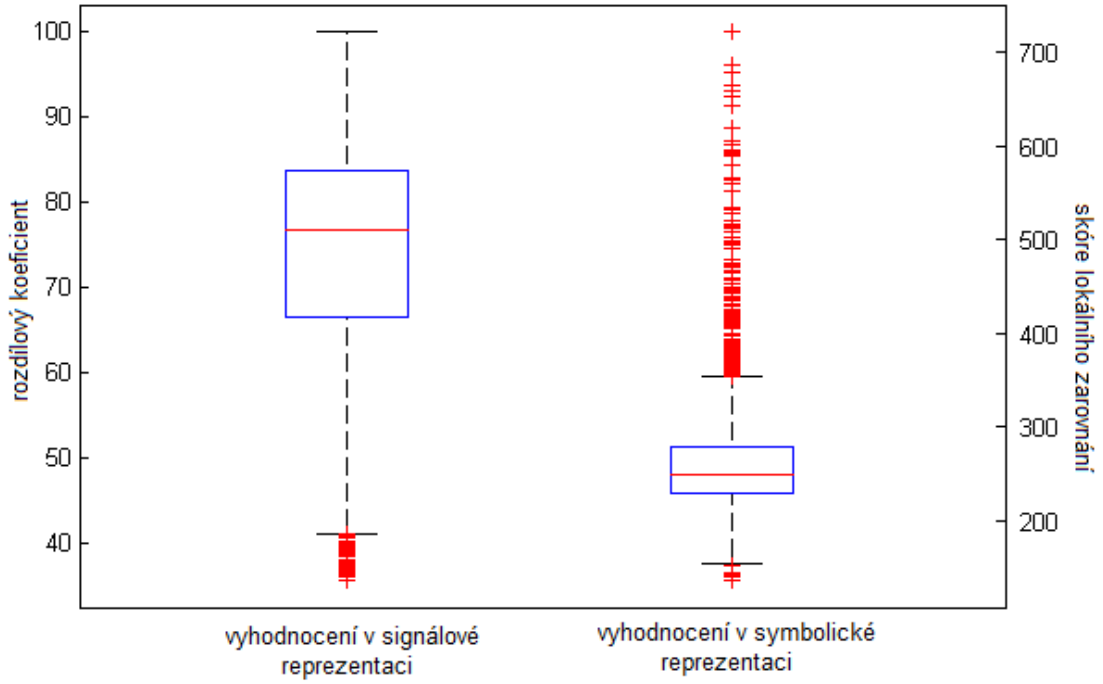
Obrázek 5. 2: Spojení dílčích modelových signálů do jednoho celku

5.2 Reálná data

K testování byly použity signály získané ze sekvenace nanopórem z verze MinION R9. Jedná se o sekvenaci viru Zika, která probíhala v rámci Zibra projektu (dostupné z: zibraproject.org/data). Volání bází probíhalo pomocí systému Metrichor. Vzorků signálů použitých pro vyhodnocení bylo celkem 674. Ke každému hlavnímu signálu z testovacího souboru bylo přiřazeno průměrně 16 možných navazujících signálů. Ke každému hlavnímu signálu byl nalezen překryv s jiným signálem. Překryv nastával nejčastěji v prvních 5000 bodech hlavního signálu (Obrázek 5. 3). Průměrná hodnota vypočítaného koeficientu (3.1) byla 74,82 (Obrázek 5. 4).



Obrázek 5. 3: Histogram znázorňující počátek překryvu

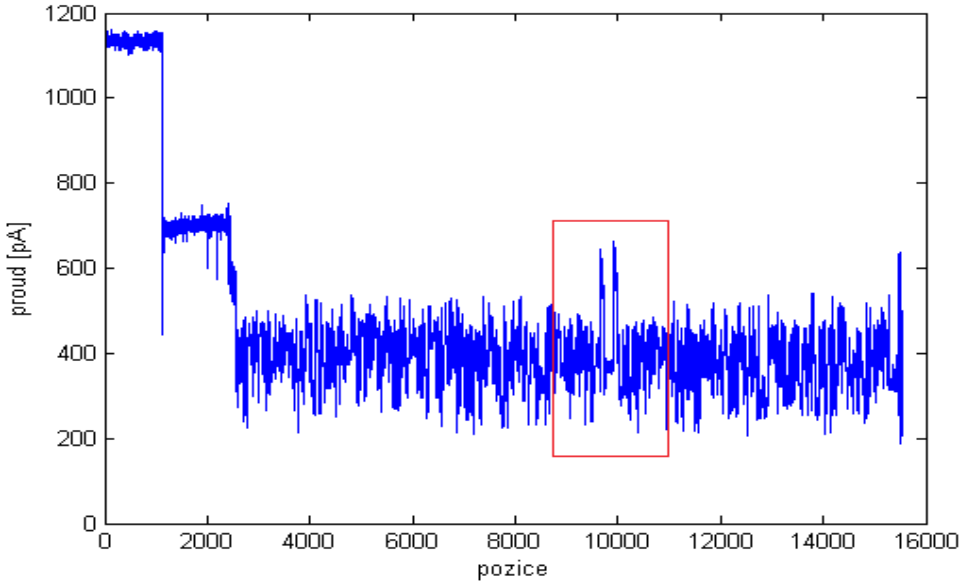


Obrázek 5. 4: Znázornění koeficientu a lokálního skóre

Na obrázku 5. 4 označuje první krabicový graf vypočtený koeficient a druhý krabicový graf označuje skóre z lokálního zarovnání nukleotidových sekvencí. První a poslední vodorovná černá čára (vousy) krabicového grafu znázorňuje rozpětí hodnot, které je pro koeficient 40 až 100 a pro skóre lokálního zarovnání je rozpětí hodnot asi 150 až 350. Spodní okraj modrého boxu označuje první kvartil, červená vodorovná čára

znázorňuje medián, který byl pro koeficient 76,8 a pro skóre lokálního zarovnání 249,3. Horní okraj modrého rámečku označuje třetí kvartil. Odlehlé hodnoty jsou znázorněny červeně.

Všechny porovnávané signály obsahují ve své druhé polovině artefakt neznámého původu, připomínající písmeno M (Obrázek 5. 5), který komplikuje vyhodnocování překryvů.

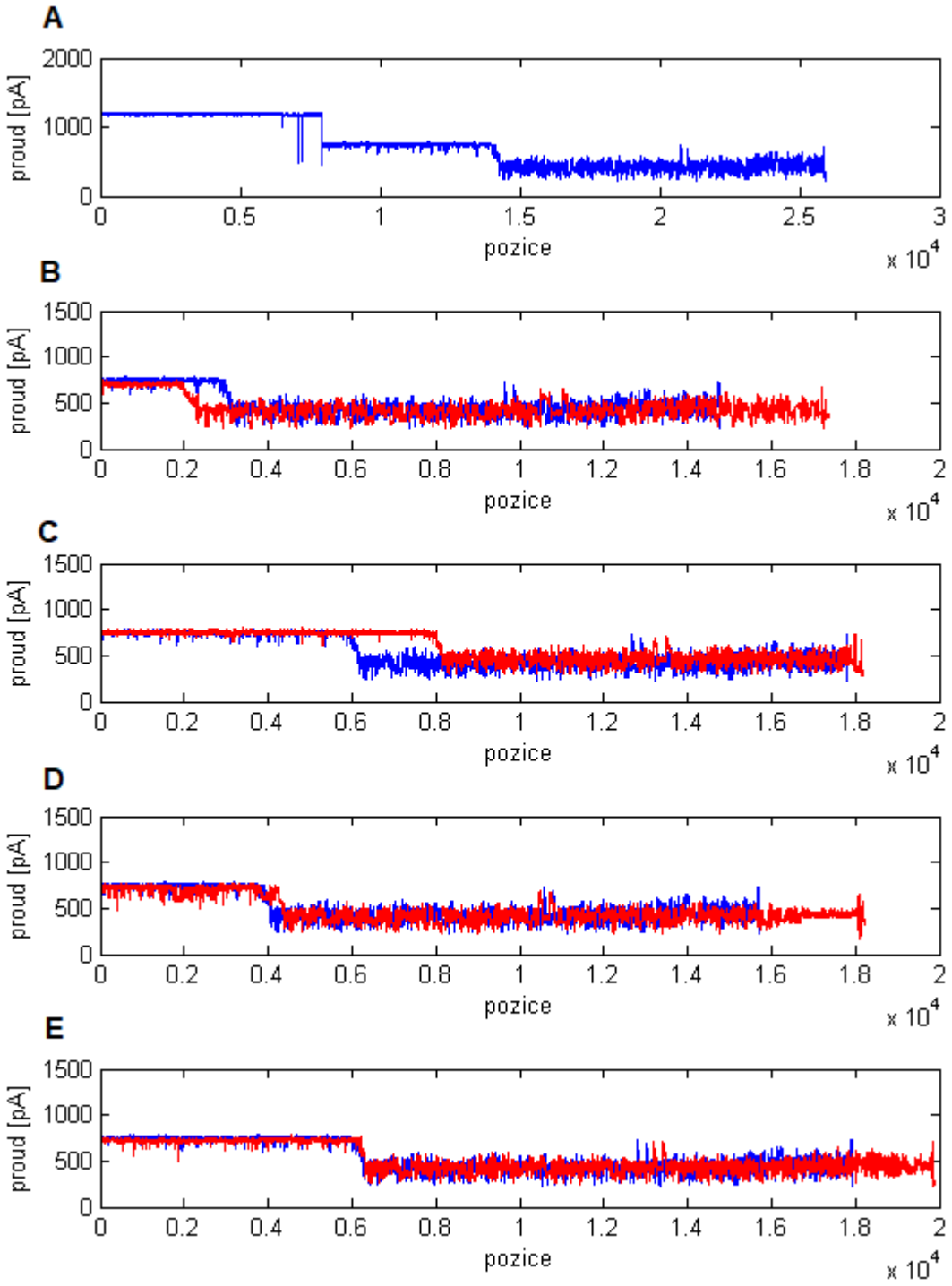


Obrázek 5. 5: Zobrazení artefaktu v signálu

5.3 Vyhodnocení konkrétního příkladu signálu

Pro názornost a ukázání překryvu byl vybrán signál č. 10 a všechny jeho možné navazující signály. Hlavní signál je vždy vyobrazen modře, navazující signály jsou znázorněny červeně. V prvním grafu A (Obrázek 5. 6) je vyobrazen celý hlavní signál č. 10, ke kterému byl hledán překryv. V druhém grafu B (Obrázek 5. 6) je zobrazen signál č. 10 a signál č. 16 od jejich společného vzorku. Koeficient byl v tomto případě roven 91,98. Jejich vzájemný překryv nastal od vzorku 11143 hlavního signálu. Ve třetím grafu C (Obrázek 5. 6) je zobrazen překryv se signálem č. 20. Jejich koeficient byl 85,84. Překryv nastal od vzorku 8098 hlavního signálu. V grafu D (Obrázek 5. 6) je zobrazen překryv se signálem č. 58 s koeficientem 70,25. Počáteční vzorek překryvu hlavního signálu byl 10200. V posledním grafu D je zobrazen překryv se signálem č. 73 (Obrázek 5. 6), vypočtený koeficient byl roven 57,78 a překryv nastal od vzorku 7980.

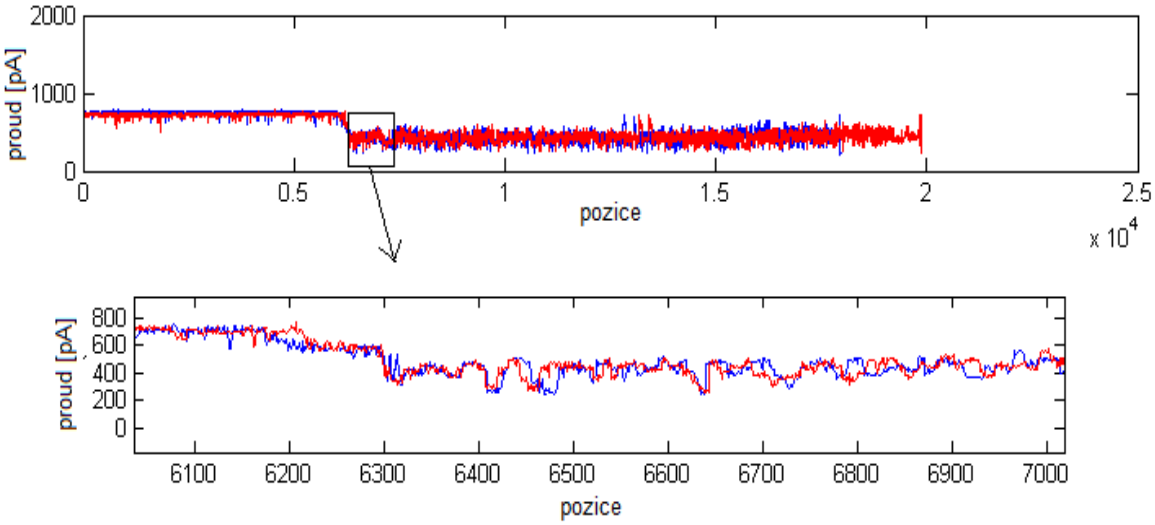
Na základě porovnání koeficientu, který je nejmenší u signálu č. 73, vychází jako nejlepší překryv právě tento poslední přiřazený signál. I po vizuálním zhodnocení, lze označit signál č. 73 jako nejlépe navazující.



Obrázek 5. 6: Zobrazení hlavního signálu a všech jeho možných překrývajících signálů

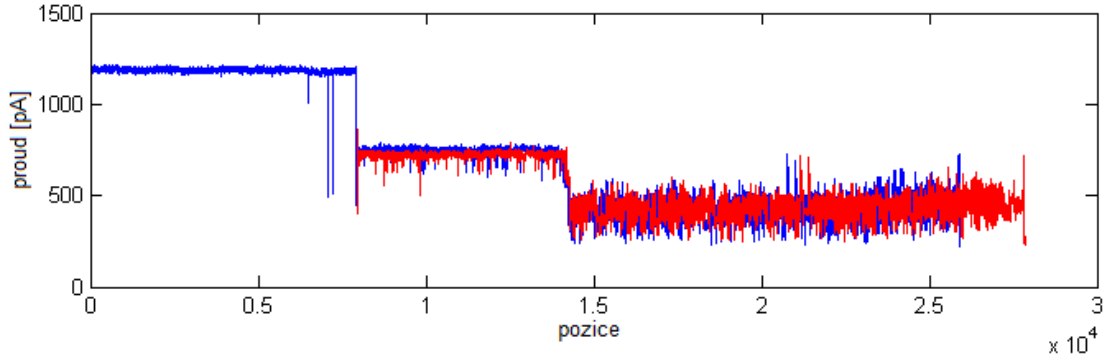
Hlavní signál je znázorněn modře, navazující signály jsou znázorněny červeně. Graf A znázorňuje hlavní signál č. 10, graf B zobrazuje hlavní signál se signálem č. 16, v grafu C je zobrazen hlavní signál se signálem č. 20, v grafu D je zobrazen hlavní signál a signál č. 58. V posledních grafu E je zobrazen hlavní signál se signálem č. 73. Ze všech zobrazených signálů nejvíce vyhovuje signál č. 73 z grafu E.

Po odmyšlení rušivého M-elementu (popsán výše), lze na první pohled určit, že signál č. 73 nejlépe navazuje na hlavní signál č. 10. Signál č. 58 vyšel jako možný navazující signál s druhým nejnižším koeficientem, nejspíše právě kvůli M-elementu, jehož jedná část je v tomto případě dobře zarovnaná. Při zvětšení vybraného úseku překryvu signálu č. 73, lze pozorovat poměrně vysokou shodu obou signálů na úrovni jednotlivých k-mer (Obrázek 5. 7). Nelze však očekávat stoprocentní překryv z důvodu šumu, a také toho, že hodnoty proudu při procházení nanopórem se i při průchodu stejné sekvence liší.



Obrázek 5. 7: Zvětšení části překryvu mezi signály č. 10 a č. 73

Při zobrazení celého signálu č. 10 od začátku a č. 73 od shodného bodu, lze pozorovat, že signál č. 73 velmi dobře kopíruje průběh hlavního signálu č. 10 (Obrázek 5. 8).



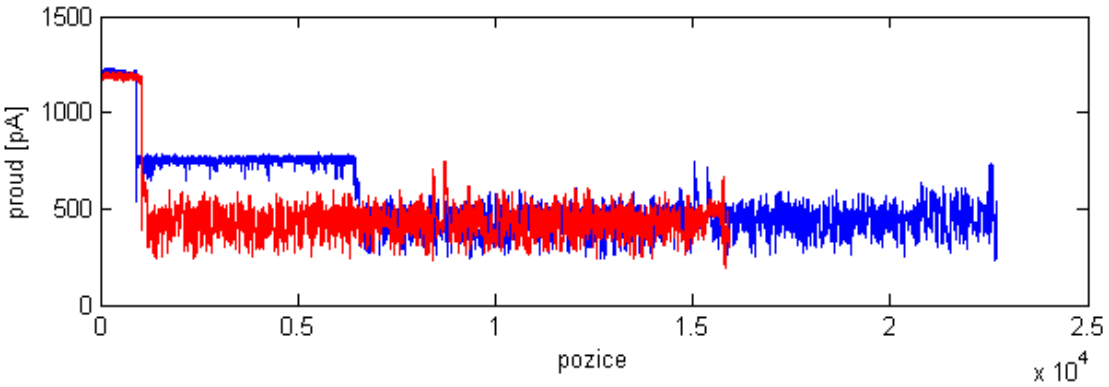
Obrázek 5. 8: Výsledné spojení signálů č. 10 a č. 73

Spojení hlavního signálu č. 10 a navazujícího signálu č. 73. Modře je znázorněn hlavní signál, červeně je znázorněn navazující signál.

Tabulka 5. 3: Navazující signály k signálu č. 10

Hlavní signál	Navazující signál	Počátek překryvu	Koeficient
10	16	11 143	91,98
10	20	8 100	85,84
10	58	10 200	70,25
10	73	7 980	57,78

Pro názornost jsou níže zobrazeny dva signály, které na sebe nenavazují (Obrázek 5. 9).



Obrázek 5. 9: Nesprávně přiřazené signály

5.4 Porovnání se znakovými metodami

Nejjednodušší metodou porovnání správnosti přiřazení signálu by bylo na základě vzájemného posunu signálu. Nukleotidové sekvence by se tak posunuly stejně, jako jsou posunuty signály. Tento postup však není možný, protože nelze stanovit, kolik vzorků signálu kóduje jeden nukleotid. Každý k-mer má totiž odlišnou délku trvání a neexistuje tedy jednoznačný vztah mezi délkou signálu a délkou nukleotidové sekvence.

Pro porovnávání sekvencí nukleotidů byla použita metoda lokálního zarovnání – Smith-Waterman. Pro toto porovnání byla použita funkce *swalign* v MATLAB. Lokální zarovnání je v tomhle případě výhodnější oproti globálnímu, které zarovnává sekvence po celé jejich délce od začátku až po konec. Jelikož je hledán překryv mezi sekvencemi a nepředpokládá se, že by sekvence byly shodné po celé jejich délce, metoda lokálního zarovnání tak hledá nejpodobnější úseky o různých délkách.

Jako skórovací matice byla použita NUC44, penalizace mezery byla nastavena na 50 a prodlužování mezery na 25. Penalizace mezer byla nastavena na vysokou hodnotu proto, aby se předešlo vkládání mezer do sekvencí, což je v tomhle případě nežádoucí.

Bylo provedeno lokální zarovnání sekvence č. 10 a všech možných navazujících sekvencí. Nejnižší skóre bylo u sekvence č. 20, jejich společné skóre s hlavním signálem bylo 235. Dále byla porovnávaná sekvence 58 a 16, jejich společné skóre se sekvencí č. 10 bylo 238,7 a 239. Nejvyšší skóre 271,7 měla hlavní sekvence se sekvencí č. 73. Sekvence č. 73 by tak měla nejlépe navazovat na hlavní sekvenci č. 10. Tato skutečnost vyplývá i z vypočteného koeficientu, který byl pro signál č. 73 nejnižší a nejlépe tak na tento signál navazoval.

Tabulka 5. 4: Porovnání skóre lokálního zarovnání

Hlavní signál	**Navazující signál**	**Skóre zarovnání**	**Koeficient**
10	16	239,0	91,98
10	20	235,0	85,84
10	58	238,7	70,25
10	73	271,7	57,78

5.5 Přesnost algoritmu

Celkem bylo získáno 10668 výsledků. Za správně přiřazený signál je považován ten, jehož vypočtený koeficient je nejnižší a zároveň skóre lokálního zarovnání nukleotidových sekvencí je nejvyšší. Za těchto podmínek bylo správně přiřazeno 221 signálů.

Jako falešně pozitivní výsledky jsou označeny takové, jejichž skóre lokálního zarovnání není nejvyšší, ale hodnota koeficientu je nejnižší. Falešně pozitivních výsledků bylo 453. Falešně negativní výsledky jsou ty, jejichž skóre lokálního zarovnání je nejvyšší, ale vypočtený koeficient není nejnižší. Falešně negativních bylo také 453 a všechny ostatní výsledky, jejichž skóre lokálního zarovnání nebylo nejvyšší a zároveň hodnota vypočteného koeficientu nebyla nejnižší, byly označeny jako pravdivě negativní.

Tabulka 5. 5: Vyhodnocení výsledků

TP	221
FP	453
TN	9 541
FN	453

TP – pravdivě pozitivní, FP – falešně pozitivní, TN – pravdivě negativní, FN – falešně negativní

Tabulka 5. 6: Přesnost algoritmu

	[%]
Senzitivita	33
Specificita	95
Přesnost	92

Senzitivita je dána jako *TP/(TP+FN)*, specifita je vypočítána jako *TN/(FP+TN)*. Přesnost (accuracy) byla vypočítána jako *(TP+TN)/(TP+FP+FN+TN)*.

Nízká hodnota senzitivity značí, že větší část výsledků, která by měla být označena jako pozitivní je chybně označena za negativní, což může být způsobeno tím, že signály, které mají nejvyšší hodnotu skóre z lokálního zarovnání, ale nemají nejnižší hodnotu koeficientu, obsahují více rušivých elementů (M-element, přechodové jevy na začátku signálu, rušení), které zkreslují vypočtený koeficient. Dále by mohl být vysoký počet falešně pozitivních výsledků způsoben tím, že volání bází (base-calling) je část zpracování sekvenačních dat, ve které nastává nejvíce chyb a nepřesností. Sekvence získané z výsledného FAST5 souboru tak nemusí být úplně správně přeloženy a mohou tak zkreslovat výsledky. Pro verzi R9 je chybovost stále 7,5 % [38].

Rozdíly koeficientů mezi nejnižším a druhým nejnižším koeficientem jsou velmi malé, jedná se často desetiny, v 60 případech měl signál s druhým nejnižším koeficientem nejvyšší hodnotu skóre lokálního zarovnání.

5.6 Porovnání rozdílového signálu a DTW

Protože je metoda DTW časově velmi náročná, nebyla použita pro celý soubor signálů, ale pouze pro prvních 60 signálů, a bude porovnávána s metodou rozdílového signálu také pouze pro prvních 60 signálů. Minimální koeficient, který byl určen v případě použití rozdílového signálu na hodnotu 100, byl pro účel porovnání s metodou DTW z časových důvodů snížen na hodnotu 70. Lze tedy předpokládat, že nemusí být nalezeny shody se všemi 60 signály. Snížení minimálního koeficientu na hodnotu 70 bylo provedeno i u metody rozdílového signálu bez zarovnání, aby měly obě metody stejné počáteční podmínky.

Tabulka 5. 7: Porovnání výsledků obou metod

	Rozdílový signál	**DTW**
TP	21	17
FP	15	15
TN	370	213
FN	15	15

TP – pravdivě pozitivní, FP – falešně pozitivní, TN – pravdivě negativní, FN – falešně negativní

Při použití algoritmu pouze s rozdílovým signálem byla nalezena shoda celkem s 36 signály, při použití algoritmu se zarovnáním pomocí DTW a následným výpočtem rozdílového koeficientu byla nalezena shoda ve 32 případech. To znamená, že pro 24 signálů v případě metody bez zarovnání a 28 signálů pro metody se zarovnáním, neměl žádný signál hodnotu koeficientu nižší než stanovená minimální mez 70. Lze předpokládat, že při zvýšení koeficientu na původní hodnotu 100 nebo více, by bylo nalezeno více shod.

Tabulka 5. 8: Porovnání přesnosti obou metod

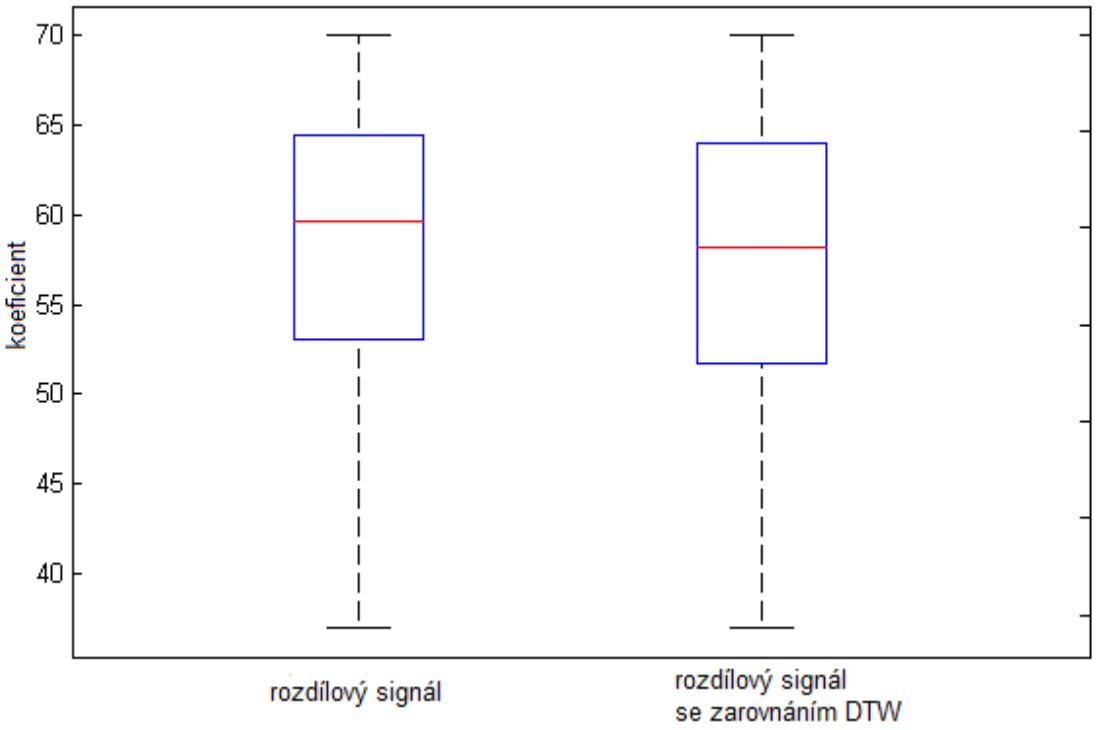
	Rozdílový signál	**DTW**
	[%]	
Senzitivita	58	53
Specificita	96	93
Přesnost	93	88

Senzitivita je dána jako *TP/(TP+FN)*, specifita je vypočítána jako *TN/(FP+TN)*. Přesnost (accuracy) byla vypočítána jako *(TP+TN)/(TP+FP+FN+TN)*.

Rozdíly v senzitivitě, specificitě a přesnosti při použití rozdílového signálu bez zarovnání a se zarovnáním pomocí DTW nejsou příliš výrazné. Ale i přesto jsou výsledky rozdílového signálu bez zarovnání přívětivější. Z toho důvodu lze říci, že metoda použití rozdílového signálu je stejně efektivní i bez zarovnání pomocí DTW, které je časově mnohem náročnější a na omezeném vzorku signálů nepřineslo žádné zlepšení.

Senzitivita je opět nižší než specificita, je však mnohem vyšší než při vyhodnocení celého souboru signálu, což je způsobeno právě omezeným vzorkem signálů.

Rozdílové koeficienty se u obou metod mírně lišily (Obrázek 5. 10). Koeficienty v případě metody s použitím DTW byly nižší než u jednodušší metody bez zarovnání. Nižší koeficient značí větší podobnost signálů a zarovnání má tedy vliv na rozdílový koeficient.



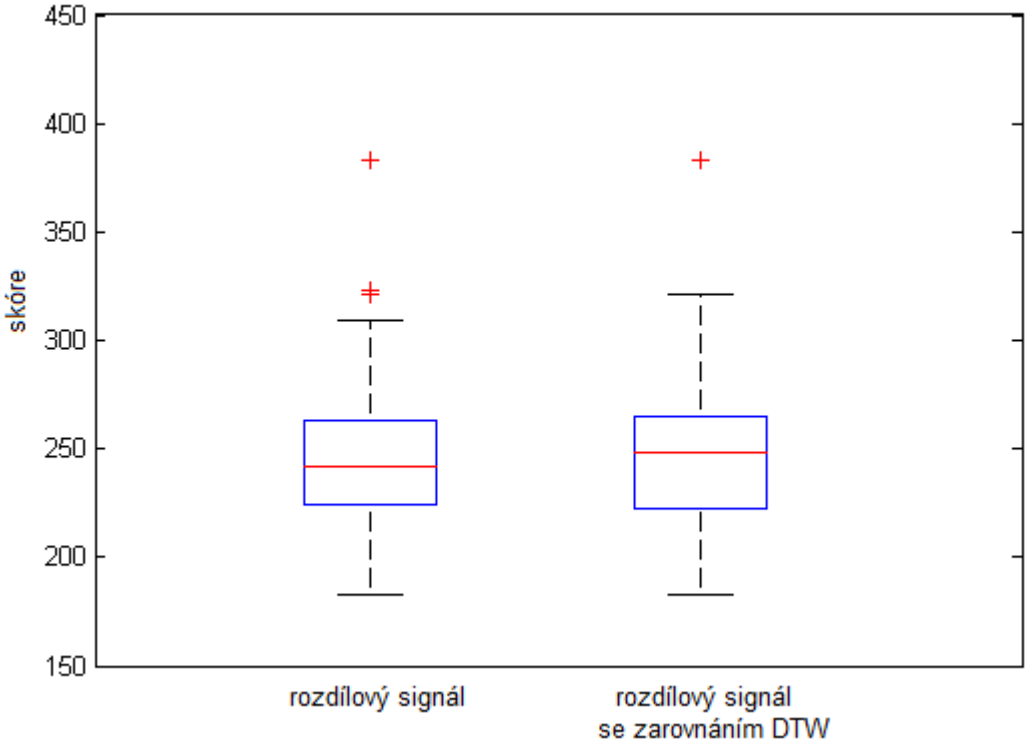
Obrázek 5. 10: Porovnání koeficientů při použití obou metod

Na obrázku 5. 10 označuje první krabicový graf koeficient pro rozdílový signál a druhý krabicový graf označuje koeficient se zarovnáním pomocí DTW před výpočtem samotného koeficientu. První a poslední vodorovná černá čára (vousy) krabicového grafu znázorňuje rozpětí hodnot, které je pro koeficient rozdílového signálu i rozdílového signálu se zarovnáním DTW asi 35 až 70. Spodní okraj modrého boxu označuje první kvartil, červená vodorovná čára znázorňuje medián, který byl pro koeficient rozdílového signálu 59,6 a pro koeficient rozdílového signálu se zarovnáním DTW 59,0. Horní okraj modrého rámečku označuje třetí kvartil. Odlehlé hodnoty jsou znázorněny červeně.

Z krabicového grafu lze pozorovat nižší hodnotu mediánu, prvního a třetího kvartilu pro vypočtený koeficient u metody s použitím DTW, což je způsobeno právě zarovnáním signálů, jejichž koeficient je při správném zarovnání nižší.

Pro obě metody bylo také počítáno skóre lokálního zarovnání ze známých nukleotidových sekvencí pomocí funkce *swalign*, opět s penalizací mezer 50 a

penalizací prodlužování mezer 25. Při zhodnocení skóre lokálního zarovnání ze všech výsledků lze říci, že při použití zarovnání pomocí DTW byly hodnoty skóre lokálního zarovnání vyšší (Obrázek 5. 11).



Obrázek 5. 11: Porovnání skóre lokálního zarovnání nukleotidových sekvencí

Na obrázku 5. 11 označuje první krabicový graf skóre lokálního zarovnání pro nukleotidové sekvence při použití pouze rozdílového signálu a druhý krabicový graf označuje skóre lokálního zarovnání pro nukleotidové sekvence při použití rozdílového signálu se zarovnáním pomocí DTW před výpočtem samotného koeficientu. První a poslední vodorovná černá čára (vousy) krabicového grafu znázorňuje rozpětí hodnot, které je pro skóre lokálního zarovnání pro rozdílový signál i skóre lokálního zarovnání rozdílového signálu se zarovnáním DTW téměř stejný, DTW má pouze vyšší horní hranici. Spodní okraj modrého boxu označuje první kvartil, který je v druhém případě nižší než u použití pouze rozdílového signálu. Červená vodorovná čára znázorňuje medián, který byl v prvním případě 241,7 a v druhém případě 248. Horní okraj modrého rámečku označuje třetí kvartil. Odlehlé hodnoty jsou znázorněny červeně.

Skóre lokálního zarovnání požadujeme co nejvyšší, pří zarovnání pomocí DTW došlo ke zvýšení mediánu pro prvních 60 vzorků signálů i rozpětí hodnot je lehce vyšší, což je pro vyhodnocení žádoucí.

Koeficient je v případě rozdílového signálu se zarovnáním DTW nižší a skóre lokálního zarovnání naopak vyšší, avšak při výpočtu přesnosti algoritmu zarovnání

DTW nepřináší žádné zlepšení. Je možné, že je to způsobené použitím omezeného vzorku a použití této metody na celém souboru dat by vedlo ke zlepšení. Vzhledem však k časové náročnosti dynamického borcení časové osy, není vhodné jeho použití na signálech získaných během sekvenace pomocí přístroje MinION, jelikož na vybraném vzorku signálů nepřináší zlepšení přesnosti oproti použití rozdílového signálu bez zarovnání, které je navíc časově mnohem méně náročné.

Přesto, že získaná senzitivita byla 33 %, sestavování *de novo* dat získaných ze třetí generace sekvenování slouží především pro stanovení kostry genomu, které se pak dále upravují a doplňují mapováním sekvenování příští generace (NGS). Specificita použité metody byla 95 %, což značí správný základ metody, se kterou lze dále pracovat. Pro samotné sestavování genomu by bylo možné použít nejenom signál s nejnižším rozdílovým koeficientem, ale poslední dva nebo tři signály s nejnižším koeficientem a podobně jako u sestavování genomu pomocí OLC metody sestavit graf překryvů, ve kterém by bylo možné nalézt výslednou cestu pro sestavení.

ZÁVĚR

Práce se zabývá metodami pro sestavování genomu, popisuje základní metody. Nejjednodušší znakový hladový algoritmus, grafové OLC a de Bruijnovy grafy patří mezi *de novo* metody pro sestavování genomu. Každá metoda má svoje výhody a je vhodná pro jiný typ sekvenačních dat. Hladový algoritmus a de Bruijnovy grafy jsou vhodnější pro krátká čtení, zatím co OLC je vhodnější pro dlouhá čtení.

Dále se práce věnuje třetí generaci sekvenování a přístroji MinION od Oxford Nanopore Technologies, který představuje jednodušší a levnější cestu k sekvenování a následnému sestavení genomu. Metody určené k sekvenování a sestavování genomu z dat z přístroje MinION využívají úseky průměrných hodnot signálu, které následně porovnávají se známými průměrnými hodnotami nukleotidových sekvencí a následně je překládají. Chybovost přístroje MinION je však stále poměrně vysoká.

Nová verze chemie R9 poskytuje ke stažení původní signál, který představuje nový pohled na věc. Největší chybovost metody nastává během volání bází – při překladu ze signálů do nukleotidů, kdy jsou jednotlivá čtení nejdříve překládána do nukleotidů a poté jsou hledány jejich vzájemné překryvy. Jiným možným řešením by mohlo být signály nejdříve spojit na základě jejich vzájemných překryvů a až poté překládat do nukleotidů. Cílem práce bylo navrhnout a vytvořit algoritmus pro hledání překryvů mezi signály. K hledání překryvů byla vytvořena funkce využívající koeficient vypočítaný pomocí rozdílového signálu. S tím, že nejnižší rozdílový koeficient značí nejvyšší podobnost. Porovnání všech vzorků všech signálů by bylo neefektivní, a tak byly porovnávány pouze úseky, u kterých byl nalezen shodný bod s hledaným signálem.

Správnost algoritmu byla ověřena na modelových datech. Při použití reálných dat bylo porovnáváno 674 signálů získaných ze sekvenace viru Zika. Ke každému signálu byl pomocí rozdílového koeficientu nalezen možný překryv. Za správně přiřazený signál byl považován ten, jehož rozdílový koeficient byl nejnižší a zároveň skóre lokálního zarovnání bylo nejvyšší. Správně přiřazených signálů bylo 221. Senzitivita metody tak vyšla 33 %, specificita vyšla 95 % a přesnost (accuracy) byla 92 %.

Při porovnání metody s využitím rozdílového signálu a s využitím zarovnání signálu (DTW) před samotným výpočtem koeficientu, vyšla metoda bez zarovnání lépe. Metoda byla z důvodu časové náročnosti DTW provedena na prvních 60 signálech. Senzitivita metody bez zarovnání byla 58 %, specificita 96 % a přesnost 93%. Senzitivita metody se zarovnáním byla 53 %, specificita 93 % a přesnost 88 %. Avšak rozdílový koeficient měl tendenci klesat a skóre lokálního zarovnání naopak stoupalo, což je žádoucí. Tyto změny však neměly vliv na přesnost.

Ze získaných výsledků lze říci, že metoda s využitím pouze rozdílového signálu je stejně efektivní jako metoda se zarovnáním pomocí DTW a navíc je časově mnohem méně náročná. Výsledky získané z vytvořené funkce by bylo možné dále použít pro genomové sestavení, kdy lze použít dva nebo tři nejmenší rozdílové koeficienty každého signálu a sestavit tak graf překryvů podobně jako v případě OLC metody.

LITERATURA

[1] EKBLOM, Robert a Jochen B. W. WOLF. A field guide to whole-genome sequencing, assembly and annotation. *Evolutionary Applications*. 2014, **7**(9), 1026–1042. DOI: 10.1111/eva.12178.

[2] MILLER, Jason R., Sergey KOREN a Granger SUTTON. Assembly Algorithms for Next-Generation Sequencing Data. *Genomics*. 2010, **95**(6), 315–327. DOI: 10.1016/j.ygeno.2010.03.001.

[3] POP, Mihai. Genome assembly reborn: recent computational challenges. *Briefings in Bioinformatics*. 2009, **10**(4), 354–366. DOI: 10.1093/bib/bbp026.

[4] HERNANDEZ, David, Patrice FRANÇOIS, Laurent FARINELLI, Magne ØSTERÅS a Jacques SCHRENZEL. De novo bacterial genome sequencing: Millions of very short reads assembled on a desktop computer. *Genome Research*. 2008, **18**(5), 802-809. DOI: 10.1101/gr.072033.107.

[5] SIMPSON, JT a R. DURBIN. Efficient de novo assembly of large genomes using compressed data structures. *Genome Research*. 2012, **22**(3), 549-556. DOI: 10.1101/gr.126953.111.

[6] TE-CHIN, Chu, Lu CHEN-HUA, Liu TSUNGLIN, Greg C. LEE, Li WEN-HSIUNG a Arthur CHUN-CHIEH SHIH. Assembler for de novo assembly of large genomes. *Proc Natl Acad Sci USA*. 2013, **110**(36), 3417–3424. DOI: 10.1073/pnas.1314090110. ISSN 0027-8424

[7] NAGARAJAN, Niranjan a Mihai POP. Sequence assembly demystified. *Nature Reviews Genetics*. 2013, **14**(3), 157-167. DOI: 10.1038/nrg3367. ISSN 1471-0056.

[8] WARREN, R. L., G. G. SUTTON, S. J. M. JONES a R. A. HOLT. Assembling millions of short DNA sequences using SSAKE. *Bioinformatics*. 2007, **23**(4), 500-501. DOI: 10.1093/bioinformatics/btl629. ISSN 1367-4803.

[9] LUO, Ruibang, Binghang LIU, Yinlong XIE, et al. SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *GigaScience*. 2012, **1**(1). DOI: 10.1186/2047-217X-1-18. ISSN 2047-217x.

[10] GNERRE, S., I. MACCALLUM, D. PRZYBYLSKI, et al. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences*. 2011, **108**(4), 1513-1518. DOI: 10.1073/pnas.1017351108. ISSN 0027-8424.

[11] SIMPSON, J. T., K. WONG, S. D. JACKMAN, J. E. SCHEIN, S. J.M. JONES a I. BIROL. ABySS: A parallel assembler for short read sequence data. *Genome Research*. 2009, **19**(6), 1117-1123. DOI: 10.1101/gr.089532.108. ISSN 1088-9051.

[12] BUTLER, J., I. MACCALLUM, M. KLEBER, I. A. SHLYAKHTER, M. K. BELMONTE, E. S. LANDER, C. NUSBAUM a D. B. JAFFE. ALLPATHS: De novo assembly of whole-genome shotgun microreads. *Genome Research*. 2008, **18**(5), 810-820. DOI: 10.1101/gr.7337908. ISSN 1088-9051.

[13] HAVLAK, P. The Atlas Genome Assembly System. *Genome Research*. 2004, **14**(4), 721-732. DOI: 10.1101/gr.2264004. ISSN 1088-9051.

[14] BOISVERT, Sébastien, François LAVIOLETTE a Jacques CORBEIL. Ray: Simultaneous Assembly of Reads from a Mix of High-Throughput Sequencing Technologies. *Journal of Computational Biology*. 2010, **17**(11), 1519-1533. DOI: 10.1089/cmb.2009.0238. ISSN 1066-5277.

[15] ZIMIN, Aleksey V., Guillaume MARÇAIS, Daniela PUIU, Michael ROBERTS, Steven L. SALZBERG a James A. YORKE. The MaSuRCA genome assembler. *Bioinformatics*. 2013, **29**(21), 2669-2677. DOI: 10.1093/bioinformatics/btt476. ISSN 1367-4803.

[16] UTTURKAR, Sagar M., Dawn M. KLINGEMAN, Miriam L. LAND, Christopher W. SCHADT, Mitchel J. DOKTYCZ, Dale A. PELLETIER a Steven D. BROWN. Evaluation and validation of de novo and hybrid assembly techniques to derive high-quality genome sequences. *Bioinformatics*. 2014, **30**(19), 2709-2716. DOI: 10.1093/bioinformatics/btu391. ISSN 1460-2059.

[17] BOLLOBÁS, Béla. *Modern Graph Theory*. New York, NY: Springer New York, 1998. ISBN 9781461206194.

[18] GROSS, Jonathan L. a Jay. YELLEN. *Graph theory and its applications*. 2nd ed. Boca Raton: Chapman & Hall/CRC, 2006. ISBN 9781584885054.

[19] BERTOSSI, Alan A. The edge Hamiltonian path problem is NP-complete. *Information Processing Letters*. 1981, **13**(4-5), 157-159. DOI: 10.1016/0020-0190(81)90048-X. ISSN 00200190.

[20] SMITH, T. F. a M. S. WATERMAN. Identification of common molecular subsequences. *J Mol Biol*. 1981, **147**(1), 195-197.

[21] SCHATZ, M. C., A. L. DELCHER a S. L. SALZBERG. Assembly of large genomes using second-generation sequencing. *Genome Research*. 2010, **20**(9), 1165-1173. DOI: 10.1101/gr.101360.109. ISSN 1088-9051.

[22] JECK, W. R., J. A. REINHARDT, D. A. BALTRUS, M. T. HICKENBOTHAM, V. MAGRINI, E. R. MARDIS, J. L. DANGL a C. D. JONES. Extending assembly of short DNA sequences to handle error. *Bioinformatics*. 2007, **23**(21), 2942-2944. DOI: 10.1093/bioinformatics/btm451. ISSN 1367-4803.

[23] LI, Z., Y. CHEN, D. MU, et al. Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and de-bruijn-graph. *Briefings in Functional Genomics*. 2012, **11**(1), 25-37. DOI: 10.1093/bfgp/elr035. ISSN 2041-2649.

[24] LANGMEAD, Ben. *Overlap Layout Consensus assembly* [online]. [cit. 2018-01-02]. Dostupné z: http://www.cs.jhu.edu/~langmea/resources/lecture_notes/assembly_olc.pdf

[25] MYERS, EUGENE W. Toward Simplifying and Accurately Formulating Fragment Assembly. *Journal of Computational Biology*. 1995, **2**(2), 275-290. DOI: 10.1089/cmb.1995.2.275. ISSN 1066-5277.

[26] LANDER, E. S. a M. S. WATERMAN. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics*. 1988, **2**(3), 231-239.

[27] HUANG, Yao-Ting a Chen-Fu LIAO. Integration of string and de Bruijn graphs for genome assembly. *Bioinformatics*. 2016, **32**(9), 1301-1307. DOI: 10.1093/bioinformatics/btw011. ISSN 1367-4803.

[28] WEI-CHE, Paul. *De novo sequence assembly* [online]. 2015 [cit. 2018-01-02]. Dostupné z: http://lsl.sinica.edu.tw/Services/Class/files/20151117474.pdf

[29] LI, R., H. ZHU, J. RUAN, et al. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research*. 2010, **20**(2), 265-272. DOI: 10.1101/gr.097261.109. ISSN 1088-9051.

[30] IP, Camilla L.C., Matthew LOOSE, John R. TYSON, et al. MinION Analysis and Reference Consortium: Phase 1 data release and analysis. *F1000Research*. DOI: 10.12688/f1000research.7201.1. ISSN 2046-1402.

[31] MIKHEYEV, Alexander S. a Mandy M. Y. TIN. A first look at the Oxford Nanopore MinION sequencer. *Molecular Ecology Resources*. 2014, **14**(6), 1097–1102. DOI: 10.1111/1755-0998.12324

[32] LU, Hengyun, Francesca GIORDANO a Zemin NING. Oxford Nanopore MinION Sequencing and Genome Assembly. *Genomics, Proteomics & Bioinformatics*. 2016, **14**(5), 265-279. DOI: 10.1016/j.gpb.2016.05.004. ISSN 16720229.

[33] TIMP, Winston, Jeffrey COMER a Aleksei AKSIMENTIEV. DNA Base-Calling from a Nanopore Using a Viterbi Algorithm. *Biophysical Journal*. 2012, **102**(10), L37-L39. DOI: 10.1016/j.bpj.2012.04.009. ISSN 00063495.

[34] *Oxford Nanopore Technology* [online]. [cit. 2018-01-02]. Dostupné z: https://nanoporetech.com/.

[35] LOOSE, Matthew, Sunir MALLA a Michael STOUT. Real-time selective sequencing using nanopore technology. *Nature Methods*. 2016, **13**(9), 751-754. DOI: 10.1038/nmeth.3930. ISSN 1548-7091.

[36] COLLETTE, Andrew. *Python and HDF5*. Beijiing: O'Reilly, 2014. ISBN 978-1-449-36783-1.

[37] BLUNSOM, Phil. *Hidden Markov Models* [online]. 2014 [cit. 2018-01-02]. Dostupné z: http://digital.cs.usu.edu/~cyan/CS7960/hmm-tutorial.pdf

[38] JAIN, Miten, John R. TYSON, Matthew LOOSE, et al. MinION Analysis and Reference Consortium: Phase 2 data release and analysis of R9.0 chemistry. *F1000Research*. 2017, **6**, 760-. DOI: 10.12688/f1000research.11354.1. ISSN 2046-1402..

[39] BRANTON, Daniel, David W DEAMER, Andre MARZIALI, et al. The potential and challenges of nanopore sequencing. *Nature Biotechnology*. 2008, **26**(10), 1146-1153. DOI: 10.1038/nbt.1495. ISSN 1087-0156.

[40] LAVER, T., J. HARRISON, P.A. O'NEILL, K. MOORE, A. FARBOS, K. PASZKIEWICZ a D.J. STUDHOLME. Assessing the performance of the Oxford Nanopore Technologies MinION. *Biomolecular Detection and Quantification*. 2015, **3**, 1-8. DOI: 10.1016/j.bdq.2015.02.001. ISSN 22147535.

[41] BOŽA, Vladimír, Broňa BREJOVÁ, Tomáš VINAŘ a Degui ZHI. DeepNano: Deep recurrent neural networks for base calling in MinION nanopore reads. *PLOS ONE*. 2017, **12**(6). DOI: 10.1371/journal.pone.0178751. ISSN 1932-6203.

[42] KAPINCHEV, Konstantin, Adrian BRADU, Frederick BARNES a Adrian PODOLEANU. GPU implementation of cross-correlation for image generation in real time. In: *2015 9th International Conference on Signal Processing and Communication Systems (ICSPCS)*. IEEE, 2015, 2015, s. 1-6. DOI: 10.1109/ICSPCS.2015.7391783. ISBN 978-1-4673-8118-5.

[43] DINOV, Martin, Romy, LORENZ, Gregory, SCOTT, David J. SHARP, Erik D. FAGERHOLM a Robert LEECH. Novel Modeling od Task vs. Rest Brain State Predictability Using a Dynamic Time Warping Spectrum: Comparisons and Contrasts with Other Standard Measures of Brain Dynamics. *Frontiers in Computational Neuroscience.* 2016,**10**. DOI: 10.3389/fncom.2016.0046. ISSN 1662-5188.

[44] MÜLLER, Meinard. Information retrieval for music and motion. Online-Ausg. New York: Springer, 2007. IBSN 9783.

Seznam elektronických příloh

Elektronická verze bakalářské práce

674 FAST5 souborů viru Zika

Zdrojové soubory – *load_data, load_seq, find_overlap, find_overlap_dtw, skript*